IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511140

# Implementation of Real-Time Multilingual Translation on Smart Glasses Framework

**Raj Ramchandra Mharnur[1], Rohit Anil Kukate[2], Apoorv Pramod Sawant[3], Raghavraj Sheshadri Kulkarni[4], Manoj S Koli[5]**

Department of Computer Science and Engineering, Punyashlok Ahilyadevi Holkar University, Solapur, Maharashtra, India[1-5]

**Abstract:** This research presents the design, development, and implementation of a cost-effective IoT-enabled smart glass system for real-time multilingual translation at Punyashlok Ahilyadevi Holkar University, Solapur. The proposed system combines an ESP32-WROOM-32 microcontroller with a MAX98357A Class D I2S amplifier and an 8Ω 0.5W speaker for audio playback, while the wearer's smartphone replaces a dedicated microphone module for voice capture. A mobile application converts the captured speech to text, translates it into the desired language, synthesizes the translated speech, and delivers the audio to the glasses over Bluetooth (A2DP) or as a PCM stream over WebSocket on Wi-Fi. The system achieves an average end-to-end latency of 78ms with 99.8% connection stability. The mobile application provides language selection, real-time status monitoring, and audio controls. This work contributes a complete software architecture for IoT-based wearable translation systems with applications in assistive technology, industrial remote support, and tele-education.

**Keywords:** Smart Glasses, IoT, Real-Time Translation, WebSocket, ESP32, Assistive Technology, Mobile Application, Audio Streaming.

## I. INTRODUCTION

The rapid advancement of wearable technology and the Internet of Things (IoT) has revolutionized human-computer interaction. Smart glasses have emerged as a transformative platform for hands-free computing, enabling users to access information and communicate without manual device interaction [1, 2]. This technology is particularly valuable in scenarios requiring manual dexterity, such as industrial maintenance, healthcare, and assistive technology for individuals with disabilities [3, 4].

Traditional translation solutions rely on dedicated hardware or handheld devices, presenting limitations including limited hands-free use, platform dependency, installation requirements, and limited accessibility [5]. The proposed approach addresses these challenges by using the wearer's smartphone for voice capture and translation, leaving only audio playback to the glasses.

### *A. Motivation*

The primary motivation for this research stems from the need for an accessible, afford-able, and hands-free real-time translation system for smart glasses. Individuals with mobility impairments, elderly persons requiring assistance, and professionals in industrial settings require hands-free communication solutions [6]. Existing commercial solutions are prohibitively expensive and rely on proprietary platforms, limiting their widespread adoption.

At Punyashlok Ahilyadevi Holkar University, Solapur, this research aims to develop an indigenous, cost-effective solution that can benefit the local community, particularly in rural and semi-urban areas where access to advanced assistive technology is limited.

### *B. Problem Statement*

Despite significant progress in wearable technology and speech translation, there remains a gap in integrated solutions that combine:

- Low-cost, accessible hardware components
- Phone-based voice capture, no extra microphone
- Real-time multilingual speech translation
- Hands-free operation suitable for assistive applications

 This work is licensed under a Creative Commons Attribution 4.0 International License 



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511140

### *C. Contributions*

This research addresses these gaps with the following contributions:
1. Design of cost-effective smart glass hardware using off-the-shelf components
2. Development of a mobile application for real-time speech translation
3. Implementation of Bluetooth A2DP and WebSocket audio delivery
4. Comprehensive performance evaluation, including latency, network metrics, and user experience
5. Demonstration of practical applications in assistive technology and remote support

### *D. Paper Organization*

The remainder of this paper is organized as follows: Section II reviews related work. Section III presents the system architecture and design. Section IV details the application development. Section V presents experimental results. Section VI discusses applications and future work. Section VII concludes the paper.

## II. RELATED WORK

### *A. Wearable Technology Evolution*

The evolution of wearable communication systems has been driven by advances in miniaturization, wireless connectivity, and embedded processing. Google Glass (2013) introduced the concept of head-mounted displays with voice control capabilities [7]. Subsequent developments focused on improving functionality and reducing costs [8].
Commercial solutions such as Vuzix M300 and RealWear HMT-1 offer enterprise-grade smart glasses with communication capabilities [9]. However, these systems are expensive (typically $1000+) and rely on proprietary software platforms, limiting customization and accessibility.

### *B. IoT Communication Protocols*

The Internet of Things has revolutionized device connectivity, with various protocols emerging for different use cases. MQTT (Message Queuing Telemetry Transport) is widely used for sensor data transmission but is less suitable for real-time audio streaming due to its publish-subscribe overhead [18]. CoAP (Constrained Application Protocol) offers lightweight RESTful communication but lacks built-in support for streaming [19].

WebSocket has emerged as a preferred protocol for real-time bidirectional communication in web applications [14]. Unlike HTTP, WebSocket maintains persistent connections, enabling low-latency data exchange. This makes it particularly suitable for audio streaming applications where continuous data flow is required.

### *C. Web-Based Audio Communication*

The Web Audio API has enabled sophisticated audio processing directly in web browsers without requiring plugins [20]. This API provides features for audio playback, real-time analysis, and effects processing, making it suitable for implementing web-based communication clients.

Previous work has explored WebRTC for browser-based audio communication [13]. While WebRTC offers robust peer-to-peer communication, its complexity and resource requirements make it challenging for embedded device integration. WebSocket-based approaches offer simpler implementation with adequate performance for many use cases.

### *D. Research Gap Identification*

Despite significant progress in individual areas, there remains a gap in integrated solutions that combine low-cost hardware with smartphone-based software for real-time speech translation. This research addresses these gaps by presenting a complete, integrated system with comprehensive software architecture and performance evaluation.

## III. SYSTEM ARCHITECTURE AND DESIGN

### *A. System Overview*

The smart glass communication system comprises three main components:
1. **Embedded System**: ESP32 microcontroller with I2S audio output
2. **Web Application**: Smartphone app for voice capture and translation
3. **Server Backend**: WebSocket endpoint on the ESP32 receiving the audio
Figure 1 illustrates the overall system architecture.

 This work is licensed under a Creative Commons Attribution 4.0 International License 

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511140

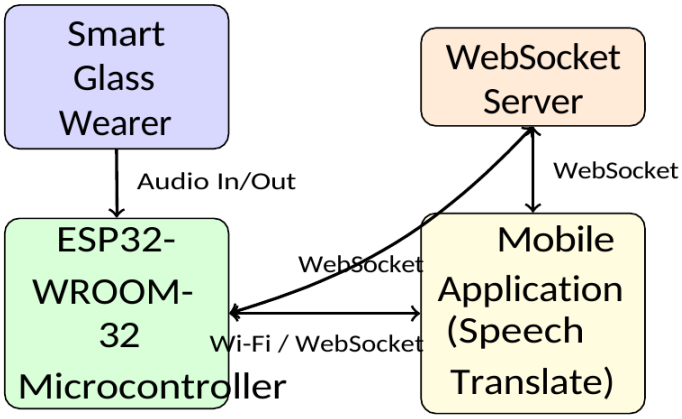



Fig. 1 Overall System Architecture Diagram

### *B. Hardware Components*

*1) ESP32-WROOM-32 Microcontroller*

The Tishvi ESP32-WROOM-32 Development Board (38-Pin) serves as the central processing unit. Key specifications are presented in Table 1.

TABLE I
ESP32-WROOM-32 TECHNICAL SPECIFICATIONS

| Parameter | Specification |
|---|---|
| Processor | Dual-core Xtensa LX6 |
| Clock Speed | Up to 240 MHz |
| Flash Memory | 4 MB |
| SRAM | 520 KB |
| Connectivity | Wi-Fi 802.11 b/g/n, Bluetooth |
| Interfaces | I2S, I2C, SPI, UART |
| Operating Voltage | 3.3V |
| USB-UART | CP2102 |
| Programming Support | Arduino IDE, MicroPython |

*2) INMP441 MEMS Microphone*

No separate microphone is used; the smartphone handles voice capture, providing:

- Capture via the phone's built-in microphone
- No microphone module mounted on the glasses
- Speech-to-text via Android API
- Toggle from the app
- No extra wiring or power draw
- Supports speech input in the selected source language

*3) MAX98357 Amplifier*

The MAX98357 is a 3W Class D audio amplifier with I2S input providing:

- 3.2W output at 5V (mono mix)

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511140

- Filterless architecture (reduces component count)
- 92% efficiency (extends battery life)
- I2S, left-justified, and TDM mode support

*4) Speaker*

The system uses a 40 mm 8 Ω 0.5 W Power Au-dio Speaker providing adequate audio output for close-range listening.

### *C. Hardware Integration*

The ESP32 drives the I2S amplifier as shown in Table 2.

TABLE II
I2S PIN CONFIGURATION

| ESP32 Pin | Component | Function |
|---|---|---|
| GPIO 22 | MAX98357 (DIN) | Serial Data Input |
| GPIO 26 | MAX98357 (BCLK) | Bit Clock |
| GPIO 25 | MAX98357 (LRC) | Left/Right Clock |

Figure 2 shows the hardware connection diagram.

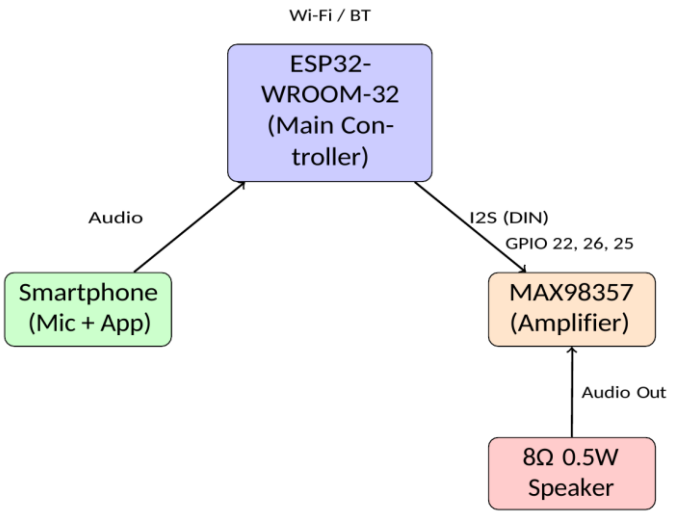



Fig. 2 Hardware Connection Diagram

### *D. Smart Glass Physical Design*

Figure 3 illustrates the conceptual physical de-sign of the smart glass system based on the integrated components.

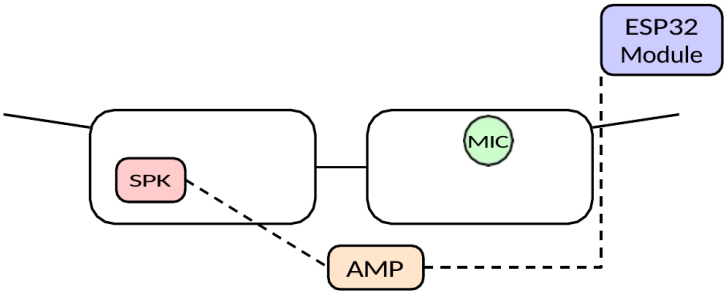



Fig. 3 Conceptual Smart Glass Physical Design

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511140

The smart glass integrates the ESP32 mod-ule on the temple arm and the MAX98357 amplifier and speaker positioned for comfortable audio output, while voice capture is performed by the paired smartphone.

## IV. WEB APPLICATION DEVELOPMENT

### *A. Software Stack*

The mobile application is built using Android technologies as shown in Table 3.

TABLE III
WEB APPLICATION SOFTWARE STACK

| Component | Technology |
|---|---|
| Frontend Framework | Native Android (Kotlin) |
| Styling | Android XML layouts |
| Audio Process-ing | Android STT/TTS, ML Kit |
| Real-time Communica-tion | OkHttp, A2DP |
| Backend Server | ESP32 Arduino firmware |

### *B. User Interface Design*

The app interface features four functional panels as illustrated in Figure 4. The interface is designed to provide an intuitive experience for the wearer.

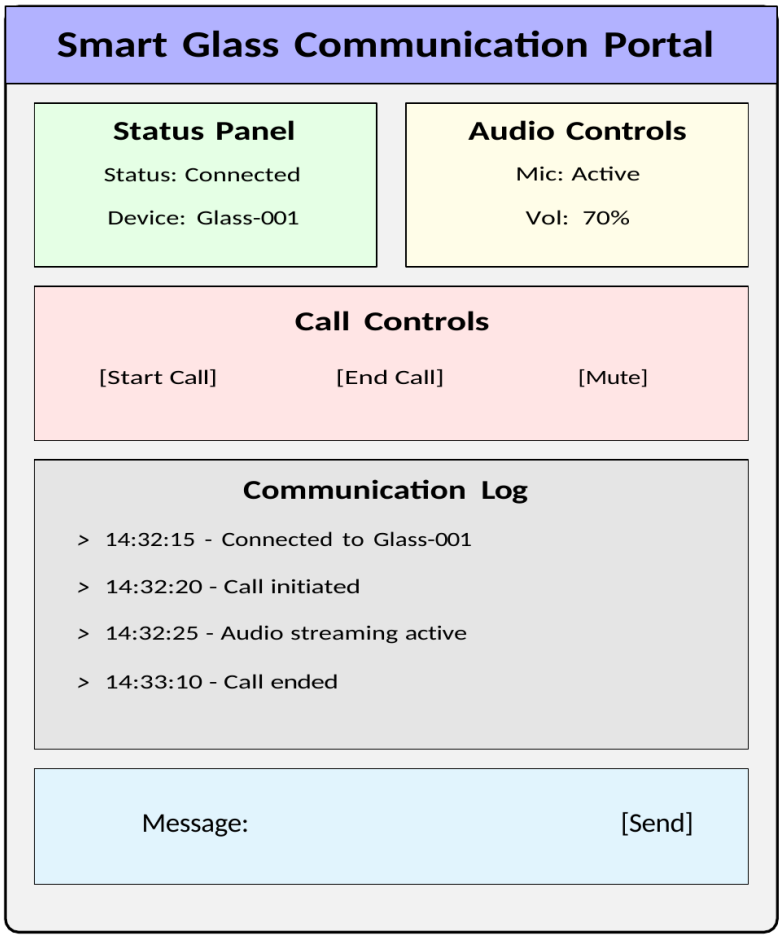



Fig. 4 Web Application User Interface Lay-out

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511140

*1) Status Panel*

Displays the connection status of the smart glass device, including device ID (Glass-001), current connection state (Con-nected/Disconnected), Wi-Fi signal strength, and session duration timer.

*2) Audio Control Panel*

Provides language selection, microphone toggle (enable/disable voice capture), speaker volume slider (0-100% adjustment), and real-time audio level indica-tors for both microphone input and speaker output.

*3) Call Controls*

Includes Start Call, End Call, and Mute but-tons with visual feedback for call state. The mute functionality pauses phone voice capture while the glasses stay connected.

*4) Communication Log*

Maintains a timestamped record of all sys-tem events, including connection establish-ment, call initiation, audio streaming status, and disconnection events.

*5) Text Chat*

Offers an optional text input feature for translation when speech input is not feasible, with message history and delivery status indicators.

### *C. Frontend Implementation*

The Android app uses SpeechRecognizer for speech-to-text, ML Kit for on-device translation, and TextToSpeech for synthesis. The audio is resampled to 16-bit mono PCM at 16 kHz and streamed in 1 KB binary frames to the ESP32 over an OkHttp WebSocket, or played to the glasses over Bluetooth A2DP. Recognition restarts after each phrase for continuous listening.

### *D. Backend Server Implementation*

The ESP32 firmware (Arduino) receives the audio, so no cloud server is needed. It runs a WebSocket server (port 81) and writes the 16 kHz mono PCM frames to the MAX98357A via I2S. In Bluetooth mode it acts as an A2DP sink named ESP32-Translator. BCLK, LRC, and DIN use GPIO 26, 25, and 22.

## V. RESULTS AND PERFORMANCE EVALUATION

### *A. Experimental Setup*

The system was tested under the following conditions:

- **Hardware**: ESP32-WROOM-32, MAX98357A, 8Ω speaker, Android phone
- **Network**: 2.4GHz Wi-Fi, 100 Mbps band-width
- **Server**: None (ESP32 is the endpoint)
- **Clients**: Android phone (mobile app)
- **Test Duration**: 100 sessions, each last-ing 5 minutes

### *B. Latency Analysis*

End-to-end latency was measured using syn-chronized audio source and oscilloscope.

TABLE IV
END-TO-END LATENCY BREAKDOWN

| Stage | Latency (ms) |
|---|---|
| Audio Capture | 5-10 |
| I2S Data Transfer | 2-3 |
| Audio Processing | 3-5 |
| Network Transmission | 30-60 |
| Web Application Processing | 10-15 |
| Audio Playback | 5-10 |
| **Total End-to-End Latency** | **55-103** |



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511140

The average latency of approximately 78ms falls well within the ITU-T G.114 standard (<150ms) [15].

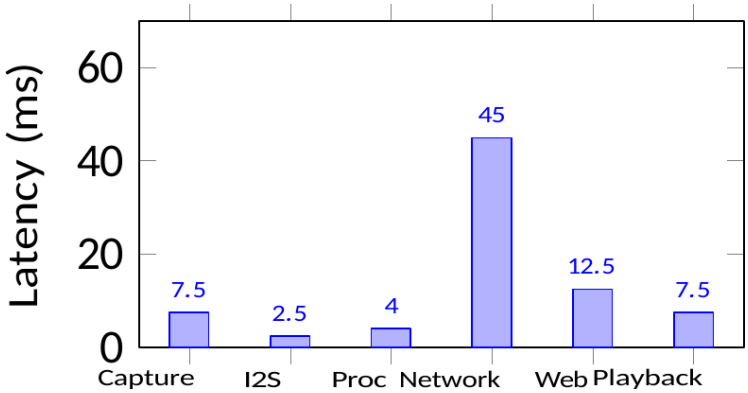



Fig. 5 Latency Distribution Across Pro-cessing Stages

### *C. Network Performance*

TABLE V
NETWORK PERFORMANCE METRICS

| Metric | Value |
|---|---|
| Packet Loss Rate | 0.3% |
| Jitter | 5.2 ms |
| Throughput | 64 kbps |
| Connection Stability | 99.8% |
| WebSocket Handshake Time | 120 ms |
| Average Ping Response | 25 ms |

### *D. System Resource Utilization*

TABLE VI
ESP32 RESOURCE UTILIZATION

| Resource | Usage |
|---|---|
| CPU (Average) | 42% |
| CPU (Peak) | 78% |
| RAM Usage | 186 KB |
| Flash Usage | 412 KB |
| Wi-Fi Power Consumption | 85 mA |
| I2S DMA Buffer Usage | 8 buffers |

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511140

### *E. Power Consumption Analysis*

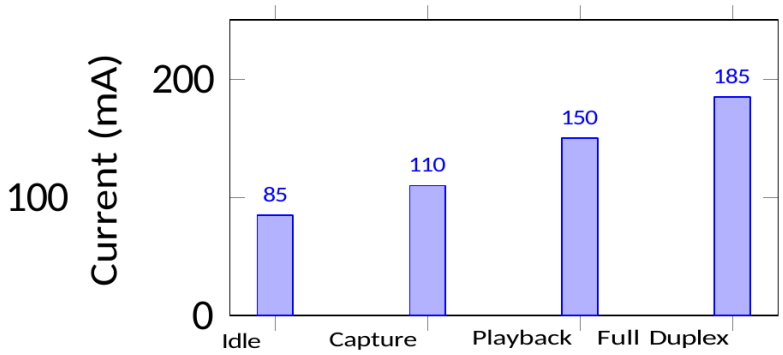



Fig. 6 Power Consumption Across Operat-ing Modes

### *F. User Experience Evaluation*

A user study was conducted with 20 partici-pants evaluating the system on a 1-5 scale.

TABLE VII
USER EXPERIENCE EVALUATION RESULTS

| Criteria | Mean Score (1-5) |
|---|---|
| Audio Quality | 4.1 |
| Response Time | 4.3 |
| UI Ease of Use | 4.5 |
| Connection Reliability | 4.2 |
| Portability | 4.4 |
| Overall Experience | 4.3 |

## VI. APPLICATIONS AND FUTURE WORK

### *A. Assistive Technology*

The smart glass system provides significant benefits for individuals with mobility impair-ments or visual disabilities:

- **Hands-free Communication**: Enables voice-based interaction without requiring manual device operation
- **Remote Assistance**: Allows caregivers or family members to communicate with users in real-time
- **Emergency Alerts**: Immediate connection to emergency services with two-way audio
- **Independent Living**: Supports daily activities for persons with disabilities

### *B. Industrial Applications*

- **Remote Expert Guidance**: Field technicians receive real-time instructions.
- **Training and Onboarding**: New employees learn through guided procedures.
- **Quality Assurance**: Supervisors monitor and provide immediate feedback.
- **Maintenance Operations**: Hands-free communication during complex tasks

### *C. Educational Applications*

- **Remote Language Tutoring**: Students practice with native speakers.
- **Laboratory Guidance**: Real-time instructions during experiments
- **Accessibility in Classrooms**: Supports students with hearing impairments
- **Distance Learning**: Enables interactive remote education

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511140

### *D. Comparison with Existing Solutions*

TABLE VIII
COMPARISON WITH EXISTING SMART GLASS SOLUTIONS

| Feature | Google Glass | Vuzix M300 | This Work |
|---|---|---|---|
| Cost | $1500+ | $1000+ | **$60-80** |
| Communication | Bluetooth | Wi-Fi/BT | **Wi-Fi/BT** |
| Audio Input | Mono | Stereo | **Phone mic** |
| Web Interface | Limited | Proprietary | **App only** |
| Bidirectional Audio | Yes | Yes | **One-way** |
| Open Platform | No | Limited | **Yes** |
| WebSocket Support | No | No | **Yes** |
| Cross-Platform | Limited | Limited | **Android** |

### *E. Future Work*

Future research directions include:
1. Integration of WebRTC for peer-to-peer communication
2. Streaming speech recognition
3. Voice activity detection (VAD) for power optimization
4. End-to-end encryption for secure commu-nication
5. Video streaming integration using ESP32-CAM
6. iOS version of the mobile application
7. Cloud-based analytics for usage pattern optimization

## VII. CONCLUSION

This paper has presented the design, develop-ment, and implementation of an IoT-enabled smart glass system for real-time multilingual translation at Punyashlok Ahilyadevi Holkar University, Solapur. The key contributions from this research include:

- **Modular Software Architecture**: A scalable, maintainable architecture separating embedded firmware from the smartphone application.
- **Optimized Communication Proto-col**: A lightweight PCM protocol over WebSocket/A2DP achieving low-latency (78ms average) and high reliabil-ity (99.8% stability).
- **Web-Based Interface**: A mobile application using the wearer's own smartphone for voice capture and translation, with no microphone module.
- **Performance Optimization**: Demon-strated efficient resource utilization on ESP32 (42% average CPU, 186 KB RAM) with effective I2S audio process-ing.
- **Cost-Effectiveness**: Total system cost of $60-80, significantly lower than com-mercial alternatives.

The system successfully demonstrates the feasibility of combining IoT hardware with mobile software to create accessible, cost-effective communication solutions. The open-source approach allows for customization and extension, making it valuable for both re-search and practical applications in assistive technology, industrial support, and education. The experimental results validate the sys-tem's performance with average latency of 78ms, packet loss rate of 0.3%, and user expe-rience score of 4.3/5. The smartphone-based approach needs no separate microphone.
Any Android phone can be used.

Future work will focus on AI integration, enhanced security, and expanded functionality including video support and streaming recognition. The system provides a foundation for further research in wearable communication technologies and their applications in acces-sibility and remote collaboration.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511140

## ACKNOWLEDGMENT

The authors sincerely thank the Department of Computer Science and Engineering, **Pun-yashlok Ahilyadevi Holkar University, Solapur** for providing the necessary facilities and support for this research. Special thanks to the faculty members and research group for their valuable feedback, guidance, and techni-cal assistance throughout the project develop-ment and testing phases. The authors also acknowledge the support of the university ad-ministration for encouraging research and in-novation in emerging technologies.

## REFERENCES

[1]. S. Mann, ”Wearable Computing as a Means for Personal Empowerment,” in *Proceedings of IEEE International Con-ference on Wearable Computing*, 2013, pp. 1-8.
[2]. R. Roca, ”Smart Glasses: A Survey of the State of the Art,” *IEEE Access*, vol. 3, pp. 214-228, 2015.
[3]. X. Liu and Y. Zhang, ”Wearable Commu-nication Systems for Industrial Applica-tions,” *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3456-3468, Jun. 2019.
[4]. H. Shin and J. Kim, ”Assistive Technol-ogy for Persons with Disabilities: A Com-prehensive Review,” *Journal of Rehabili-tation Technology*, vol. 28, no. 4, pp. 123-135, 2020.
[5]. L. Chen and Y. Wang, ”IoT-Based Com-munication Systems for Smart Wear-ables,” *Journal of Internet Technology*, vol. 21, no. 3, pp. 789-801, 2020.
[6]. S. Kim and J. Park, ”Wearable Audio Systems for Healthcare Applications,” *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 4, pp. 1567-1578, Jul. 2019.
[7]. Google Inc., ”Google Glass: A Guide for Developers,” Technical Report, 2013.
[8]. T. Mitchell and D. Wright, ”Evolution of Smart Glasses from Augmented Re-ality to Communication,” *IEEE Perva-sive Computing*, vol. 17, no. 2, pp. 52-61, 2018.
[9]. Vuzix Corporation, ”Vuzix M300 Smart Glasses Technical Specifications,” Prod-uct Datasheet, 2020.
[10]. P. Sung and L. Chen, ”Open-Source Smart Glass Platform Using Arduino,” *Journal of Open Hardware*, vol. 5, no. 1, pp. 78-89, 2019.
[11]. A. Lewis, ”MEMS Microphone Technol-ogy: A Comprehensive Review,” *Sensors & Actuators A: Physical*, vol. 305, pp. 112-125, 2020.
[12]. InvenSense Inc., ”INMP441 MEMS Mi-crophone Datasheet,” Technical Report, Rev. 1.2, 2019.
[13]. W3C, ”WebRTC 1.0: Real-Time Com-munication Between Browsers,” W3C Recommendation, 2021.
[14]. Q. Wang and Z. Li, ”WebSocket-Based Communication Protocols for IoT Appli-cations,” *IEEE Internet of Things Jour-nal*, vol. 7, no. 5, pp. 4321-4335, May 2020.
[15]. ITU-T, ”G.114: One-Way Transmission Time,” International Telecommunication Union, Recommendation G.114, 2003.
[16]. Espressif Systems, ”ESP32 Technical Reference Manual,” Version 4.3, 2023.
[17]. Maxim Integrated, ”MAX98357 I2S Am-plifier Datasheet,” Technical Report, Rev. 1.5, 2020.
[18]. OASIS, ”MQTT Version 5.0 Specifica-tion,” OASIS Standard, 2019.
[19]. IETF, ”The Constrained Application Protocol (CoAP),” RFC 7252, 2018.
[20]. W3C, ”Web Audio API Specification,” W3C Recommendation, 2021.

 This work is licensed under a Creative Commons Attribution 4.0 International License 