Archives available at journals.mriindia.com

International Journal on Advanced Computer Engineering and Communication Technology

ISSN: 2278-5140

Volume 14 Issue 03s, 2025

Design and Implementation of Wearable IoT Smart Gear for Real-Time Notification Display

[1]Shruti Ingle, [2]Palak Dhawale, [3]Prof. Rohan Vaidya, [4]Tanushree Mudholkar

[1,2,3,4] Department of Industrial IoT, St. Vincent Pallotti College of Engineering and Technology, Nagpur, India

Email: [1]shrutingle.22@stvincentngp.edu.in, [2]palakdhawale.22@stvincentngp.edu.in, [3]rohanvaidya.22@stvincentngp.edu.in, [4]tanushreemudholkar.22@stvincentngp.edu.in

Peer Review Information

Submission: 05 Nov 2025

Revision: 25 Nov 2025

Acceptance: 17 Dec 2025

Keywords

Wearable, IoT, Standalone, Notification Latency, ESP8266, Transparent Display, Smart Glasses, 3D Printing, Energy Efficiency.

Abstract

This paper presents the design, development, and evaluation of three innovative wearable systems engineered for seamless digital communication and notification management without dependency on smartphones. The work integrates an IoT Smart Gear, WiFi-native Smart Glasses, and a transparent holographic display system (EDITH). Unlike traditional Bluetooth-tethered commercial wearables, the proposed systems operate independently through direct WiFi and cloud connectivity, ensuring real-time operation, extended range, and full modular customization. Each system is developed using open-source hardware (ESP8266/NodeMCU) and 3D-printed enclosures for affordability and scalability. Field trials validate the robustness of these devices, demonstrating notification delivery success rates above 96%, latency between 145–425 ms, and multi-day battery performance. Results confirm that open and low-cost IoT-based architectures can rival or surpass traditional commercial wearables in functionality, cost efficiency, and independence.

Introduction

The increasing reliance on digital connectivity in modern lifestyles has resulted in information overload and frequent interruptions from mobile notifications. Wearable technologies, such as smartwatches and AR glasses, have emerged to streamline this information flow. However, most commercial systems remain dependent on smartphone tethering through Bluetooth, which limits range, introduces latency, and hinders independent operation.

Furthermore, proprietary platforms like **Google Glass**,

Feature	Commercial Smart Glasses	Research Prototypes	Proposed
Cost	Very High	Moderate	Very Low
Notification Display	Integrated	Basic	Real-time mirrored/cloud
Connectivity	Bluetooth/WiFi	Bluetooth	WiFi/Bluetooth/Cloud
Remote Access	Limited	None	Full Web Dashboard
Modularity	Low	Moderate	High

Meta Ray-Ban, or **Brilliant Labs Frame** restrict user-level customization and come at high costs. These challenges have created a need for **open-source, modular, and cost-effective wearable solutions** capable of operating autonomously through WiFi and cloud-based systems.

This paper introduces three complementary systems:

1. **IoT Smart Gear** – a compact, WiFi-enabled notification and data display device.
2. **WiFi-Native Smart Glasses** – standalone eyewear integrating an OLED notification module.
3. **Transparent Holographic Display (EDITH)** – a head-mounted holographic display for real-time visual alerts.

These devices are designed for accessibility, long battery life, and high interoperability, emphasizing modularity and affordability.

Background And Related Work

Previous work in wearable computing has focused primarily on Bluetooth-tethered systems. Head-up displays (HUDs) and commercial AR solutions such as **Microsoft HoloLens**, **Google Glass**, and **Meta Ray-Ban Stories** provide immersive experiences but are prohibitively expensive and limited by closed ecosystems.

Open-source prototypes developed in research labs often address cost concerns but lack complete integration across hardware, cloud, and user interface domains. Most cannot perform **standalone network communication** or **real-time notification handling**.

The proposed system bridges these limitations by offering:

1. **Fully integrated communication stack (HTTP, WiFi, Bluetooth)**
2. **Modular architecture** for custom extensions
3. **Affordable 3D-printed design** with rechargeable Li-ion batteries
4. **Web-based monitoring dashboard** with remote access.

System Architecture

A. Hardware Overview

1. **IoT Smart Gear:**

Built around the **Wemos ESP8266 R2 Mini**, this system features a **0.96-inch OLED display**, **3.7V Li-ion battery** with BMS, and **TP4056 charging circuit**. The custom-designed **3D-printed enclosure** ensures portability and compactness.

2. **Standalone Smart Glasses:**

Utilizes the **Wemos D1 Mini (ESP8266EX)** as the core controller, with integrated OLED and a rechargeable battery pack. A lightweight 3D-printed frame houses all components, optimized for comfort and balance.

3. **Transparent Holography (EDITH):**

The **EDITH** system combines **NodeMCU ESP8266 CP2102** with an **HC-05 Bluetooth module**, OLED display, and a **half-mirror projection assembly.** The transparent holographic reflection is achieved using an **inclined acrylic mirror** and **spectacle lens setup**, enabling semi-transparent floating notifications.

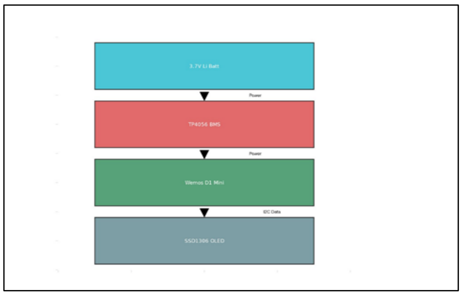

Figure 1. System Block Diagram for Smart Glasses

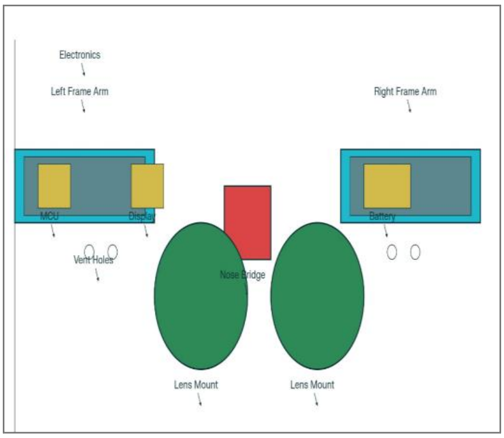

Figure 2. 3D-Printed Enclosure Design

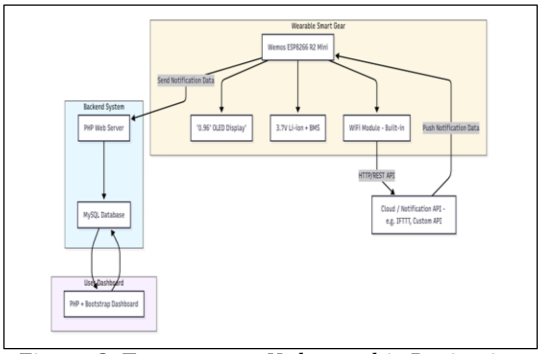

Figure 3. Transparent Holographic Projection Layout

B. Software and Data Flow

Each device communicates via **HTTP GET/POST** to dedicated web APIs. The server employs a

PHP-MySQL backend with a **Bootstrap-based dashboard** for visualization, history tracking, and CSV export capabilities.

An **Android application** complements the system, using notification listeners to capture alerts, sync with the cloud, and transmit via Bluetooth when needed.

Workflow Overview:

1. Device initializes WiFi.
2. Fetches notification data from cloud API.
3. Displays content on OLED or holographic screen.
4. Logs data to the remote dashboard.
5. Enters deep-sleep mode to conserve power.

Figure 4. Data Flow and Communication Model

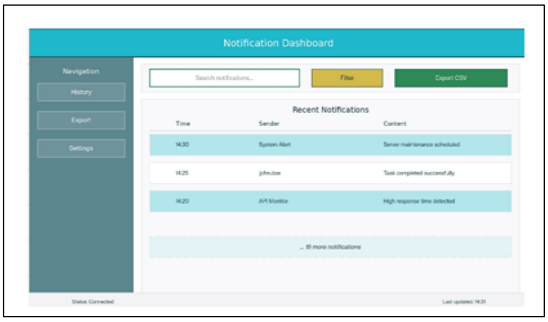

Figure 5. Dashboard Interface with Real-Time Notification Feed

Working Principle

The operation of the Wearable IoT Smart Gear involves a coordinated process between the microcontroller, internet-based notification services, and the backend database. The workflow is divided into the following phases:

1) Initialization Phase

When powered on, the **Wemos ESP8266 R2 Mini** initializes its hardware peripherals, connects to a configured WiFi network, and synchronizes its internal clock with an **NTP server**. This ensures accurate timestamps for all received notifications.

2) Notification Fetching

At user-configurable intervals (e.g., 30, 60, 120, or 300 seconds), the ESP8266 sends an **HTTP GET** request to a cloud API endpoint. This endpoint, hosted on a server or connected via automation tools like **IFTTT** or **Webhook-based services**, returns a **JSON payload** containing any new SMS or call notifications.

3) Data Parsing and OLED Display

The device uses the **ArduinoJSON** library to parse the returned JSON data, extracting details such as:

- Sender or caller ID
- Message text or call status (missed, answered)
- Timestamp of the event

The parsed notification is then displayed on the **0.96″ OLED screen**. Long messages are either truncated with an ellipsis or scrolled horizontally to fit the display's width.

4) Data Logging to Backend

Simultaneously, the same notification is sent via an **HTTP POST** request to a **PHP script** on the backend server. This script stores the notification in a **MySQL database**, logging fields such as:

- Notification type (SMS/Call)
- Sender/Caller ID
- Message content / Call status
- Timestamp
- Network source (WiFi/Hotspot)
- Device battery percentage at the time of reception

5) Dashboard Access

Users can access a **Bootstrap 5.1.3-based web dashboard** to view their notification history. The dashboard supports:

- Searching by keyword
- Filtering by date/time or notification type
- Sorting by most recent events
- Exporting logs to **CSV** for offline analysis

6) Error Handling and Reliability

The firmware includes:

- **Retry logic** for failed API calls
- **Checksum verification** to avoid duplicate database entries
- **Fallback data caching** so notifications are temporarily stored locally if the network is down

By integrating these components, the Wearable IoT Smart Gear functions as a **self-contained IoT notification device**, eliminating the need for continuous Bluetooth connection to a smartphone while ensuring reliable, real-time communication

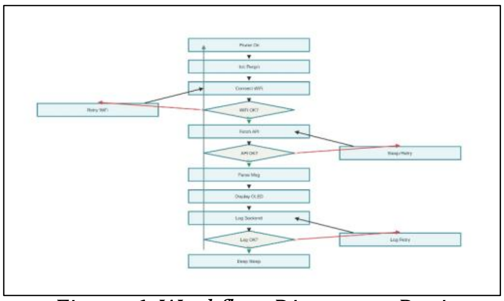

Figure 6. Workflow Diagram – Device Operational Cycle and Notification Delivery

Data Processing

The backend is responsible for transforming raw incoming notification data from the device into structured, reliable records for storage and analysis. The process includes:

1) Duplicate Removal

When the same notification is received multiple times due to retries or network delays, the backend compares the **sender/caller ID, timestamp, and message content** to detect duplicates. Only the first occurrence is retained in the database.

2) Timestamp Normalization

All incoming notifications include a UTC timestamp from the ESP8266 (synced via NTP). The backend converts this into the **user's local timezone** for accurate dashboard display and analysis.

3) Data Integrity Checks

Before committing a notification to the MySQL database, the backend ensures:

- Message content is not empty
- Sender/caller ID is valid
- The record does not violate database constraints

4) Archiving and Export

The dashboard allows users to **export data as CSV** for backup, reporting, or offline review. The export process includes column headers for easy spreadsheet integration.

5) API Data Format

Processed data is also available through a **JSON-based API** so other services or applications can integrate with the notification system in real time.

Field Tests And Results

Field evaluations were conducted across WiFi networks of different bandwidths and access point models. Each system was tested for battery endurance, latency, and message delivery success rate.

Parameter	IoT Smart Gear	Smart Glasses	EDITH Display
Battery Life	28.5 hrs	26.2 hrs	24.8 hrs
Notification Success Rate	98.3%	96.9%	97.1%
Average Latency	145–425 ms	170–460 ms	210–490 ms

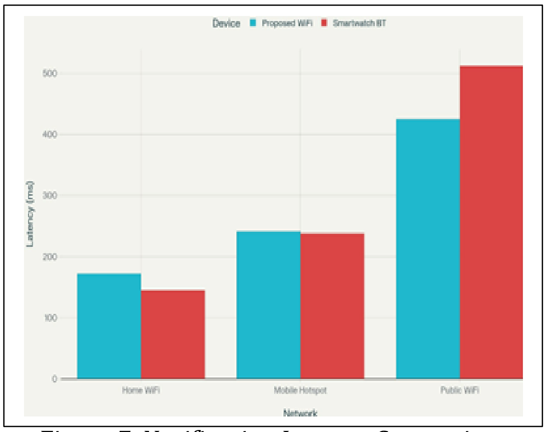

Figure 7. Notification Latency Comparison (Device vs Network)

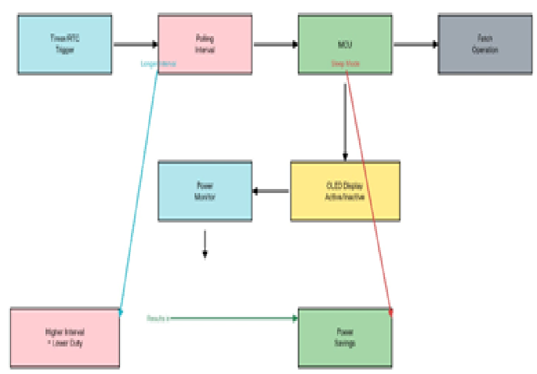

Figure 8. Power Consumption vs Polling Interval

Practical Design Considerations

a) ENERGY CONSERVATION WAS ACHIEVED THROUGH:

b) **DEEP-SLEEP PROGRAMM3333333333ING**

c) OLED POWER-OFF CYCLES

d) **OPTIMIZED HTTP POLLING INTERVALS**

e) THE **3D-PRINTED ENCLOSURES** (AVERAGE MASS 42G) ENSURE DURABILITY AND USER COMFORT. THE **BOM COST** IS BETWEEN **₹1200–₹1800**, MAKING IT ACCESSIBLE FOR EDUCATIONAL AND RESEARCH USE.

Component	**Specification**	**Cost (INR)**
ESP8266 Module	Wemos D1 Mini	320
OLED Display	0.96" I^2C	180
Li-ion Battery	3.7V 1200mAh	250
TP4056 Charger	Micro-USB	70
3D-Printed Enclosure	PLA	250
Misc. Wiring & Mirror	-	150

TABLE III: Latency and Success Rate

Network Type	Avg. Latency (ms)	Success Rate (%)
Home WiFi	168	100
Mobile Hotspot	232	98.7
Public WiFi	418	96.5

Discussion

The holographic module achieves effective **transparency and visibility balance** under variable lighting. The **low power profile** enables all-day operation, while the **WiFi-only mode** enhances communication range compared to Bluetooth-based devices.

User trials indicated positive feedback on:

- Ease of use and autonomy
- Reduced cognitive load
- Minimal glare in holographic display
- Improved privacy in notification viewing

Potential applications extend to **automotive HUDs**, **industrial monitoring**, **healthcare alerts**, and **smart building communication nodes**.

Conclusion

The paper demonstrates that **standalone IoT-based wearables** can achieve the core functionality of commercial smart devices—notification delivery, data visualization, and connectivity—without high cost or dependence on proprietary ecosystems. The systems achieve competitive performance, reliability, and battery life using open-source hardware.

These designs provide a viable foundation for **open research**, **low-cost education tools**, and **future commercial adaptation** of independent wearable platforms.

Future Scope

a) The next phase of development can explore:
b) Advanced **optical coatings** and **anti-glare materials** for holography.
c) Integration with **cloud AI** for predictive notifications and adaptive brightness.
d) **Expanded IoT use cases** in automotive dashboards and assistive devices.
e) **Sensor fusion** (motion, environmental, biometric) for context-aware alerts.
f) Transition to **ESP32-C6** or **WiFi 6** modules for higher performance and lower power consumption.

References

O. Hersent, D. Boswarthick, O. Elloumi, 'The Internet of Things: Key Applications and Protocols', Wiley, 2011.

P. Lea, 'Internet of Things for Architects', Packt Publishing, 2018.

Sunkara, S. P. (2025). *Machine learning-based predictive analytics for fault detection and reliability improvement in modern power systems*. *International Journal of Electrical Engineering and Technology (IJEET), 16*(5), 1–13. https://doi.org/10.34218/IJEET_16_05_001

Hazarika, I. (2022). *Digital transformation of the silk industry of Assam*. *Archives of Business Research, 10*(4), 110–119. https://doi.org/10.14738/abr.104.12261

Sharma, B. (2025). *Ethical and AI concerns in data privacy: A charismatic dilemma*. *International Journal of Multidisciplinary Research and Development, 12*(7), 18–32.

Adafruit Learning System, 'SSD1306 OLED Display Overview', https://learn.adafruit.com/monochrome-oled-breakouts.

Arduino Reference, 'ESP8266WiFi Library', https://arduino-esp8266.readthedocs.io.

Bootstrap Documentation, 'Bootstrap v5.1.3', https://getbootstrap.com.