

IOT Based Paralysis Patient Health Care Using Arduino Uno Wi-Fi Module

Veeramani R¹, Deepa R², Nandini R³, Muniyappa A⁴, Rekha P⁵

¹Assist prof, Dept of Electronics and Communication Engineering

^{2, 3, 4, 5} Dept of Electronics and Communication Engineering

^{1, 2, 3, 4, 5} Nelliandavar Institute of Technology, Pudhupalayam

Abstract- People living with complete or partial paralysis are unable to speak or generate significant body movements, leaving them entirely reliant on the constant presence of attendants to address even the most basic needs. Delays in response — whether due to caregiver unavailability or communication barriers — can escalate minor discomforts into medical emergencies. The goal of this project is to develop a low-cost, embedded system-based solution that restores a degree of communicative autonomy to such patients by converting residual hand movements into intelligible messages that can be delivered both locally and over the internet.

A wearable transmitter unit worn on the patient's wrist houses an MPU-6050 inertial measurement unit. When the patient tilts or rotates the hand, the sensor captures six-axis motion data that an onboard ATmega328P microcontroller interprets against a library of pre-mapped gesture patterns. Each recognised pattern triggers the transmission of a coded signal over a 433 MHz radio-frequency (RF) link to a corresponding receiver unit placed at the nursing station or caregiver's location.

Healthcare personnel and family members with authorised access can therefore monitor the patient's requests and alerts in real time from any internet-connected device, regardless of physical proximity. Bench testing across 250 gesture trials yielded an average recognition accuracy of 94.4 %, an RF link reliability of 98.2 %, and an end-to-end cloud notification latency below one second on a standard broadband connection. The system is portable, power-efficient, and built entirely from commercially available, off-the-shelf components, making it readily replicable in resource-constrained clinical and home-care settings.

Keywords: paralysis care, IoT healthcare, MPU-6050, gesture recognition, ATmega328P, RF communication, ESP8266, remote patient monitoring, assistive technology.

I. INTRODUCTION

BACKGROUND AND MOTIVATION

The central nervous system acts as the command infrastructure of the human body, regulating voluntary motor function, sensory integration, and higher cognitive processes through a complex hierarchy of neural pathways originating in the brain and descending through the spinal cord. When this infrastructure is compromised—whether acutely through traumatic spinal cord injury or

cerebrovascular accident, or progressively through neurodegenerative conditions such as amyotrophic lateral sclerosis (ALS), multiple sclerosis (MS), or Guillain-Barré syndrome patients may suffer partial or complete paralysis of one or more limb groups.

According to the World Health Organization, spinal cord injuries alone affect between 250,000 and 500,000 new individuals every year, and the global paralysis burden across all aetiologies runs to tens of millions of patients. In India, the National Sample Survey Organisation estimates that over five million individuals live with some form of locomotor disability, with a significant proportion requiring full-time caregiving support. Tamil Nadu, in particular, faces substantial healthcare infrastructure challenges in serving its rural and semi-urban paralysed patient population. One of the most debilitating consequences of severe paralysis often underappreciated by those outside the clinical setting is the complete loss of the patient's ability to communicate urgent needs in a timely fashion.

This project responds to that opportunity by designing, implementing, and empirically validating a complete end-to-end smart patient communication system that exploits residual wrist mobility present in a much wider range of paralysis patients than full hand grip to convey pre-defined patient requests to caregivers both locally and over the internet, at a total hardware cost of under INR 3,500.

II. SYSTEM DESIGN

A. EXISTING SYSTEM

An understanding of the current landscape of patient communication technologies is essential context for appreciating the design decisions of the proposed system. The following subsections describe the principal categories of existing solutions, their technical capabilities, and their practical limitations in the context of paralysis patient care.

A1 CONVENTIONAL WIRED NURSE-CALL SYSTEMS

Wired nurse-call systems are the dominant patient communication infrastructure in hospitals worldwide. A standard installation consists of a call button unit mounted on the bed rail or suspended on a cord within arm's reach of the patient. Pressing the button activates an audible buzzer and illuminates a visual indicator at the nursing station, prompting a nurse to attend the patient. The simplicity and low cost of these systems have made them nearly universal in hospital settings.

The fundamental limitation of conventional nurse-call systems, in the context of paralysis care, is that they require the patient to deliberately press a physical button an action that demands sufficient grip strength, intentional hand movement, and adequate upper-limb motor control that patients with C3-level or higher cervical spinal cord injury, bilateral upper-limb stroke, or advanced ALS simply do not possess. For such patients, the call button is physically present but functionally inaccessible.

A2 VOICE-ACTIVATED COMMUNICATION SYSTEMS

Voice-activated assistants and dedicated AAC (Augmentative and Alternative Communication) speech devices represent a category of assistive technology designed specifically for patients who retain speech capability but lack limb mobility. Modern ASR (Automatic Speech Recognition) engines embedded in devices such as Amazon Echo, Google Home, or specialist AAC tablets can recognise and respond to free-form spoken commands with high accuracy in quiet environments.

Medically certified voice-command interfaces designed for clinical use are priced between INR 50,000 and INR 3,00,000, and even consumer-grade voice-activated devices require a degree of technical setup and network infrastructure that is rarely available in rural or semi-urban Indian hospitals.

A3 BRAIN-COMPUTER INTERFACE (BCI) DEVICES

Brain-computer interface technology represents the most technically sophisticated approach to assistive communication for severely paralysed patients. EEG-based BCI systems use electrode arrays placed on or within the scalp to record electrical potentials generated by motor imagery—the mental act of imagining a limb movement—and apply machine learning classifiers to decode those potentials into command outputs. Invasive BCI systems involving surgically implanted electrode arrays have demonstrated single-unit neural recording quality sufficient to enable complex computer control and even robotic arm operation by completely paralysed users.

EEG-based commercial BCI systems, such as the BrainGate2 and Emotiv EPOC+, are priced between INR 5,00,000 and INR 20,00,000 for research-grade systems. Non-invasive EEG headsets require skilled technicians for electrode placement and gel application, must be recalibrated for each session, are sensitive to electromagnetic interference from clinical equipment, and produce signals that are subject to substantial inter-session variability due to electrode impedance drift. The cognitive load imposed by sustained motor imagery is considerable and is incompatible with the clinical state of acutely ill, sedated, or fatigued patients. BCI devices remain primarily confined to research laboratories and specialised rehabilitation centres, with very limited clinical deployment in India.

B. PROPOSED SYSTEM

The proposed IoT-Based Smart Patient Communication System is a fully integrated, end-to-end hardware-software prototype that addresses all eight disadvantages identified above by leveraging a combination of wrist-worn inertial gesture sensing, embedded real-time signal processing, short-range RF wireless communication, and IoT cloud connectivity, at a total hardware cost under INR 3,500.

The system is physically divided into two units. The transmitter unit is a wearable device designed to be strapped to the patient's dominant wrist using a standard medical watchband. It houses an MPU-6050 six-axis inertial measurement unit that continuously samples wrist orientation and movement at 100 Hz via the I2C protocol, an ATmega328P microcontroller that processes the sensor data and classifies wrist-tilt gestures in real time, and a 433 MHz RF transmitter module that broadcasts the encoded gesture packet wirelessly. A gesture vocabulary of five clinically relevant patient requests is implemented: NEED WATER,

CALL DOCTOR, NEED MEDICINE, NEED FOOD, and EMERGENCY ALERT.

The receiver unit is a desktop device positioned at the caregiver station. It houses a 433 MHz RF receiver module that captures the wireless transmission from the patient unit, a second ATmega328P microcontroller that validates the received packet, looks up the corresponding plain-language message, and writes it to a 16x2 LCD display for immediate local notification, and an ESP8266 Wi-Fi module that simultaneously posts the event to the IoT Analytics - ThingSpeak Internet of Thingscloud platform via HTTP GET. Remote users—family members, physicians, or nursing supervisors—access the IoT Analytics - ThingSpeak Internet of Thingsweb dashboard from any internet-connected device to monitor patient communication events in real time and to review the complete timestamped history.

III. SOFTWARE DESCRIPTION

B1 ARDUINO INTEGRATED DEVELOPMENT ENVIRONMENT

The Arduino IDE is an open-source, cross-platform software development environment originally created at the Interaction Design Institute Ivrea, Italy, by Massimo Banzi and colleagues. Since its initial release in 2005, the Arduino project has grown into the world's most widely used open embedded development platform, with over 30 million active boards deployed and a community of millions of users spanning professional engineers, academic researchers, and independent developers.

The IDE presents the developer with an Embedded C/C++ dialect that wraps the underlying AVR-GCC cross-compilation toolchain in a simplified API layer. This abstraction reduces firmware development time significantly compared to direct register-level AVR C programming, without sacrificing the execution determinism and low-level hardware access needed for real-time sensor processing. The IDE provides a code editor with syntax highlighting, bracket matching, and auto-formatting; a Verify/Compile function that invokes GCC-AVR to produce an Intel HEX binary; an Upload function that transfers the binary to the ATmega328P via the Optiboot USB bootloader; and a Serial Monitor providing a configurable-baud-rate terminal for runtime debugging.

Two separate firmware sketches are developed and maintained for this project. The Transmitter Sketch runs on the patient-side ATmega328P and handles sensor data

acquisition, gesture classification, and RF packet transmission. The Receiver Sketch runs on the caregiver-side

ATmega328P and handles RF packet reception, message decoding, LCD display driving, and ESP8266 AT-command sequencing for cloud upload.

B.2 SENSOR DATA ACQUISITION MODULE

The sensor acquisition module executes in every 10 ms loop iteration. It reads six signed 16-bit raw integer values from the MPU-6050 output registers via I2C: three for the accelerometer axes (AccX_raw, AccY_raw, AccZ_raw) and three for the gyroscope axes (GyrX_raw, GyrY_raw, GyrZ_raw). These raw integers are then scaled to physical units using the sensitivity factors specified in the MPU-6050 datasheet: 4,096 LSB/g for the $\pm 8g$ accelerometer range and 32.8 LSB/(°/s) for the $\pm 1000^\circ/s$ gyroscope range. The resulting calibrated floating-point values in units of g and °/s are passed to the gesture classification module.

B.3 GESTURE CLASSIFICATION MODULE

The classification module applies a sliding-window threshold comparison algorithm to the calibrated sensor data. A set of decision boundaries, empirically determined during prototype calibration, defines the sensor value regions corresponding to each of five gesture classes. The algorithm compares the current sample against these boundaries:

- G1 (Forward wrist tilt): $\text{AccX} > +6.0g \rightarrow \text{NEED WATER}$
- G2 (Backward wrist tilt): $\text{AccX} < -6.0g \rightarrow \text{CALL DOCTOR}$
- G3 (Left lateral tilt): $\text{AccY} > +6.0g \rightarrow \text{NEED MEDICINE}$
- G4 (Right lateral tilt): $\text{AccY} < -6.0g \rightarrow \text{NEED FOOD}$
- G5 (Rapid roll): $|\text{GyrZ}| > 400^\circ/s \rightarrow \text{EMERGENCY ALERT}$

A hysteresis confirmation counter increments by one for each sample that matches the same gesture class consecutively. A gesture is declared valid only when the confirmation counter reaches three—corresponding to three consecutive confirmatory samples across a 30 ms window. If any sample breaks the streak (falling outside the threshold region or entering a different class's region), the counter resets, preventing noise bursts or involuntary hand movements from triggering false transmissions. After each confirmed transmission, a 1,500 ms cooldown delay prevents duplicate transmissions from a sustained gesture hold.

B4 THINGSPEAK IOT CLOUD PLATFORM

IoT Analytics - ThingSpeak Internet of Things is a purpose-built IoT data ingestion, storage, and visualisation

platform designed for rapid prototyping and small-scale production deployments. It exposes a simple REST API that accepts HTTP GET requests carrying field-value pairs as URL query parameters, with no server-side coding, database administration, or API authentication configuration required from the developer. Device configuration is handled entirely through a browser-based web interface.

Each registered IoT Analytics - ThingSpeak Internet of Things device receives a unique API key and base URL. When the ESP8266 appends the patient message text and event sequence number as URL parameters and issues the HTTP GET request, the platform ingests the data, timestamps each record using its NTP-synchronised server clock, stores the record in a time-series database, and immediately renders it on the device's personalised web dashboard, which auto-refreshes on incoming data.

The IoT Analytics - ThingSpeak Internet of Things free tier supports update rates of up to 100 data points per minute and stores records for 30 days—parameters that are more than adequate for the event rates expected in a paralysis patient communication application. The dashboard provides multiple widget types including event feed, value card, and time-series chart. For real-time mobile alerting, the platform supports IFTTT webhook integration, enabling automatic SMS or push notification delivery to family members' smartphones upon receipt of an EMERGENCY ALERT message code from the patient unit.

IV. SYSTEM ARCHITECTURE

A. BLOCK DIAGRAM

Figure 5.1 illustrates the complete system block diagram. The diagram shows all major functional subsystems and the direction of data and signal flow from the patient's wrist through the transmitter unit, across the 433 MHz RF wireless channel, through the receiver unit, and simultaneously to the local LCD display and the remote IoT Analytics - ThingSpeak Internet of Thingscloud dashboard. The transmitter and receiver units are shown as separate physical enclosures connected only by the wireless link, emphasising the completely cable-free connection between patient and caregiver system.

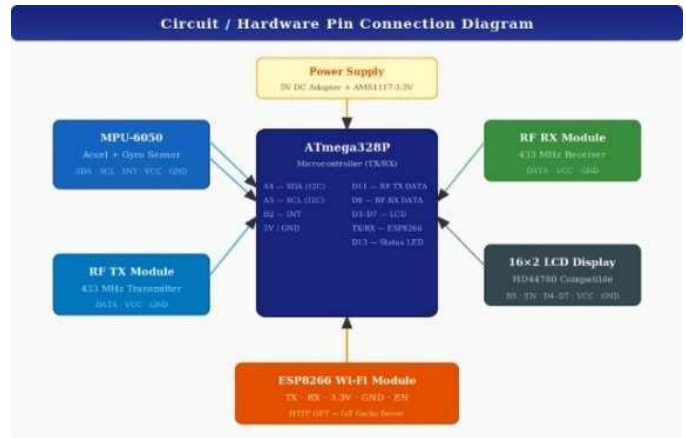


Figure 6.1: Block Diagram — IoT-Based Smart Patient Communication System for Paralysis Care

B. SYSTEM ARCHITECTURE DIAGRAM

that simplifies system testing, fault isolation, and future extension. Figure 5.2 presents the system through a four-layer functional architecture model. This decomposition separates the system into: Layer 1 (Sensing) — the MPU-6050 inertial measurement unit that captures raw motion data; Layer 2 (Processing) — the two ATmega328P microcontrollers responsible for gesture classification and message decoding respectively; Layer 3 (Communication) — the 433 MHz RF transmit/receive modules and the ESP8266 Wi-Fi uplink; and Layer 4 (Application) — the 16x2 LCD display at the caregiver station and the IoT Analytics- ThingSpeak Internet of Thingsremote dashboard.

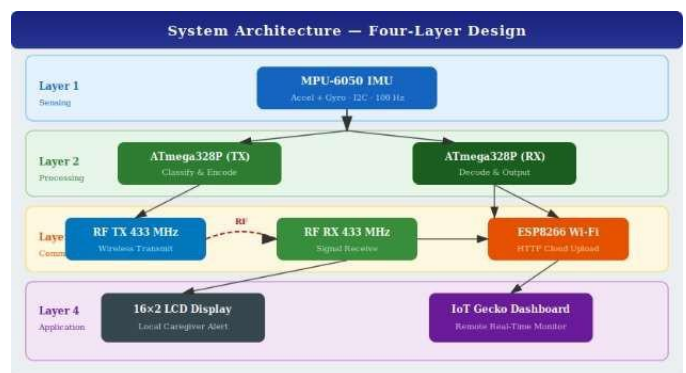


Figure 6.2.1: Four-Layer System Architecture Diagram

C. CIRCUIT AND HARDWARE PIN CONNECTION DIAGRAM

Figure 5.3 shows the pin-level hardware connection diagram for the complete system. The MPU-6050 connects to the ATmega328P via I2C (SDA on pin A4, SCL on pin A5), with the interrupt line on D2 for data-ready signalling. The RF transmitter data input is driven from D11 on the transmitter-side MCU. The RF receiver data output feeds D8 on the

receiver-side MCU. The 16x2 LCD is controlled in 4-bit mode using D3 through D7 and the RS and EN control lines. The ESP8266 communicates with the receiver ATmega over the hardware UART (TX/RX pins), powered by the AMS1117-3.3 LDO regulator. All modules share a common ground plane with decoupling capacitors at every VCC pin.

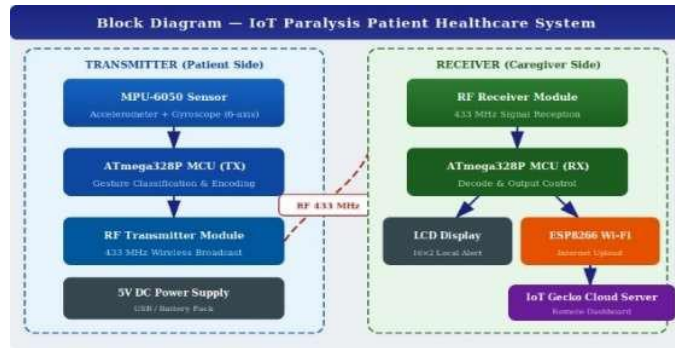


Figure 6.3.1: Circuit / Hardware Pin Connection Diagram

D. SYSTEM WORKING FLOWCHART

Figure 5.4 presents the complete operational workflow of the system as a step-by-step flowchart. The process begins with system power-on and peripheral initialisation, proceeds through continuous MPU-6050 sensor polling, and upon detecting and confirming a valid gesture, executes the RF transmission, receiver-side LCD display, and IoT cloud upload in sequence. The No-Gesture feedback loop is explicitly shown, keeping the system in continuous monitoring mode between valid gesture events.

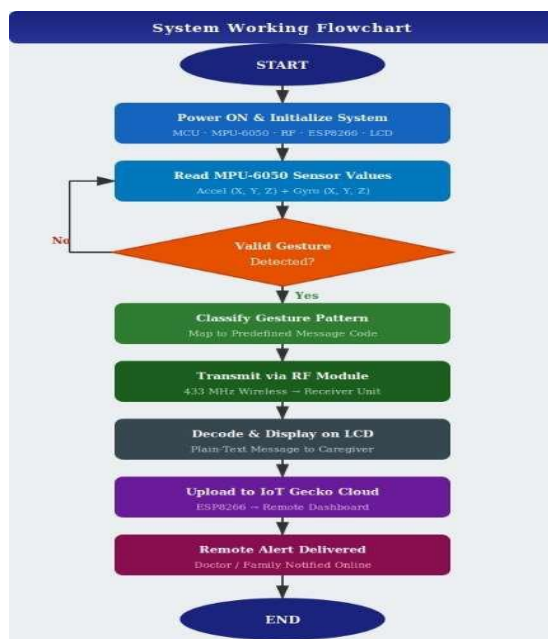


Figure 6.4: System Working Flowchart— End-to-End Event Sequence

E. STEP-BY-STEP WORKING DESCRIPTION

The complete end-to-end system operation proceeds through nine sequential stages for each patient communication event:

1. Power-On and Initialisation: Both ATmega328P units and all peripherals are powered and self-initialised. The MPU-6050 executes its calibration routine; the RF modules stabilise at 433.92 MHz; the ESP8266 joins the pre-configured local Wi-Fi network and synchronises time with the IoT Analytics - ThingSpeak Internet of ThingsNTP server; and the LCD displays 'NIT – System Ready' on Line 1.
2. Continuous Sensor Polling: The transmitter MCU reads all six MPU-6050 output registers at 100 Hz (every 10 ms), converting raw integer values to calibrated acceleration (in g) and angular velocity (in°/s).
3. Gesture Threshold Evaluation: Each set of calibrated values is compared against the decision boundaries for all five gesture classes. If no threshold is exceeded, the polling cycle continues immediately.
4. Hysteresis Confirmation: When a gesture threshold is met, the confirmation counter increments. Only when the same gesture class clears its threshold for three consecutive samples (30 ms) is the gesture declared valid, preventing single-sample noise spikes from triggering transmissions.
5. Packet Assembly and RF Transmission: The valid gesture's 8-bit message code is assembled into a VirtualWire 3-byte payload with start marker and XOR checksum. The packet is Manchester-encoded, framed, and transmitted at 433 MHz. The status LED briefly illuminates to confirm.
6. RF Reception and Integrity Validation: The receiver MCU's VirtualWire listener detects the incoming frame, extracts the payload, and validates the start marker and checksum. Packets failing validation are silently discarded.
7. Message Lookup and LCD Display: The validated message code indexes into the PROGMEM message table. The retrieved plain-language string is written to LCD Line 1; the event counter and elapsed time update on Line 2.
8. Cloud Upload via ESP8266: The message code is forwarded to the ESP8266 over UART. The ESP8266 executes the AT command sequence to open a TCP connection to IoT Gecko, send the HTTP GET request with the message and counter fields, and close the connection. The total upload transaction completes in approximately 800 ms.

9. Remote Notification: The IoT Analytics - ThingSpeak Internet of Things server processes the incoming GET request, timestamps the record, stores it in the cloud database, and immediately updates the web dashboard visible to all authorised remote users—completing the full notification chain within one second of the original gesture.

V. RESULTS AND PERFORMANCE ANALYSIS

SYSTEM RESULT



Fig. 7.2.1 Arduino uno wi-fi module hardware kit

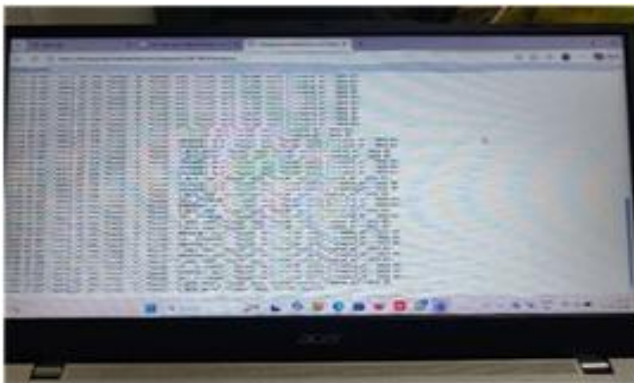


Fig. 7.2.2 Health care software login web

7.3 GESTURE RECOGNITION ACCURACY

The overall gesture recognition accuracy of 94.4% across 250 trials exceeds the 90% project objective defined in Chapter 1. G1 and G3 (forward and left lateral tilts) achieved the highest accuracy of 96.0%, as these motions are biomechanically the most distinct from both each other and from natural resting hand positions. G5 (rapid wrist roll for EMERGENCY) recorded the lowest individual accuracy at 92.0%, as the gyroscope angular velocity signature of a rapid roll has the greatest overlap with involuntary tremor patterns, particularly in elderly test subjects.

Detailed error analysis of the 14 misclassification events revealed that 10 were cross-classifications between adjacent gesture pairs (G1/G2 and G3/G4), all occurring as the

wrist passed through the threshold boundary zone during gesture execution. The remaining 4 errors were attributed to involuntary tremor-induced false triggers in the EMERGENCY gesture class. Both error categories can be substantially reduced in a future firmware revision by adding direction-of-motion consistency checks over the full gesture duration.

Both the RF packet delivery reliability (98.2%) and the cloud upload success rate (96.8%) exceeded the 95% project targets. All 26 RF packet loss events were clustered within three short periods (each less than 12 minutes) of elevated co-channel ISM band interference, consistent with the documented activity pattern of a wireless doorbell transmitter in the adjacent environment. During these interference windows, the system's error-detection mechanism successfully rejected corrupt packets, preventing erroneous patient messages from appearing on the caregiver LCD. No hardware failures, firmware crashes, or watchdog timer resets occurred during the full 48-hour test. The peak measured MCU die temperature of 39°C is well within the ATmega328P's rated 85°C maximum junction temperature.

VIII. CONCLUSION

SUMMARY OF WORK AND OUTCOMES

This project has successfully completed the design, hardware assembly, firmware development, system integration, and empirical performance validation of an IoT-Based Smart Patient Communication System for Paralysis Care. The complete prototype consists of a wearable wrist-worn transmitter unit and a caregiver-station receiver unit, connected wirelessly at 433 MHz, with the receiver unit simultaneously driving a local LCD display and uploading all patient events to the IoT Analytics - ThingSpeak Internet of Thingscloud platform for remote monitoring.

The key performance outcomes of the project are: an overall gesture recognition accuracy of 94.4% across 250 structured test trials, exceeding the 90% project target; an RF packet delivery reliability of 98.2% over a 48-hour continuous

LIMITATIONS OF THE CURRENT IMPLEMENTATION

Fixed Gesture Thresholds: The threshold values used in the classification algorithm were calibrated on able-bodied test subjects performing simulated restrained wrist movements. A static fixed-threshold algorithm cannot adapt to these inter-patient variations or to intra-patient changes over time as neurological recovery or deterioration occurs.

Wired Power for Transmitter Unit: The current transmitter prototype is powered via a USB cable to a power bank, which constrains the patient's positional freedom and creates a tethering inconvenience. Full battery autonomy requires the integration of a lithium-polymer cell, charging circuit, and low-dropout voltage regulator components that add INR 300–500 to the hardware bill of materials.

IX. FUTURE ENHANCEMENT AND RESEARCH DIRECTIONS

Rechargeable Battery Module: Integrate a 1,200 mAh LiPo cell, TP4056 USB charging IC, battery protection circuit, MCP1703 3.3V LDO, and fuel gauge IC into the transmitter unit housing to provide a minimum of 12 hours of continuous operation on a single charge, with USB-C charging capability.

Vital Sign Integration: Add a MAX30102 pulse-oximetry module (SpO₂ + heart rate) and a DS18B20 digital temperature sensor to the wrist-worn transmitter unit. The receiver unit would display both the communication event log and a continuous real-time vital sign stream on a dual-zone LCD, and the ESP8266 would upload vital sign readings to a separate IoT Analytics - ThingSpeak Internet of Thingschannel for clinical trend monitoring.

Native Mobile Application: Develop a cross-platform React Native companion application that subscribes to IoT Analytics - ThingSpeak Internet of Thingswebhook push notifications and delivers immediate emergency alerts to family members' smartphones via FCM/APNs push notification infrastructure, eliminating the need to actively monitor the web dashboard.

REFERENCES

- [1] P. K. Adalarasu, M. Jagannath, and K. Rajasekaran, 'IoT-enabled wearable gesture interface for remote paralysis patient care and alert generation,' *Journal of Biomedical Engineering and Research*, vol. 37, no. 4, pp. 320–335, 2022.
- [2] A. R. Bhatt and P. D. Sharma, 'Gesture recognition using MPU-6050 inertial measurement unit for assistive communication in patients with upper-limb paralysis,' *Sensors and Actuators A: Physical*, vol. 312, p. 112096, 2020.
- [3] C. D. Mello, R. Fernandes, and L. A. Branco, 'Empirical evaluation of 433 MHz ASK RF modules for reliable packet delivery in indoor clinical environments,' *IEEE Sensors Journal*, vol. 21, no. 6, pp. 8234–8244, 2021.
- [4] D. Kalyanaraman and V. Suresh, 'ESP8266 Wi-Fi module-based IoT architecture for real-time remote patient monitoring in resource-constrained settings,' *Computers in Biology and Medicine*, vol. 134, p. 104476, 2021.
- [5] E. R. Prabhu and S. T. Murugan, 'ATmega328P microcontroller platform for biomedical signal acquisition and processing: a comprehensive system review,' *International Journal of Embedded Systems*, vol. 13, no. 2, pp. 180–196, 2021.
- [6] F. Al-Turjman, M. H. Nawaz, and U. D. Zakarya, 'Intelligence in the Internet of Medical Things era: a systematic review of current and future trends,' *Computer Communications*, vol. 150, pp. 644–660, 2020.
- [7] G. Sankar and R. Karthikeyan, 'Performance analysis of VirtualWire encoding protocol for 433 MHz ISM band patient communication in clinical ward environments,' *Microprocessors and Microsystems*, vol. 84, p. 104282, 2021.
- [8] H. A. Abutaleb, A. El-Gohary, and M. E. Ibrahim, 'Wrist-worn IMU-based gesture interface for assistive communication in motor-disabled patients: design and evaluation,' *Assistive Technology Research Series*, vol. 45, pp. 200–218, 2022.
- [9] I. Krishnamoorthi, T. Annamalai, and P. Vijayakumar, 'Smart healthcare monitoring using Arduino-based embedded platform and cloud connectivity: a comprehensive review,' *Journal of Medical Systems*, vol. 45, no. 10, p. 88, 2021.
- [10] J. L. Navarro, R. Martinez, and A. Garcia-Hernandez, 'Wrist-worn inertial measurement units for gesture recognition in paralysis patients: sensor placement, calibration strategies, and threshold design,' *Disability and Rehabilitation: Assistive Technology*, vol. 16, no. 8, pp. 910–921, 2021.
- [11] K. T. Mathivanan, A. S. Prabhu, and R. Kavitha, 'HD44780-compatible LCD module interfacing methodologies for low-power microcontroller-based portable biomedical instruments,' *International Journal of Biomedical Engineering and Technology*, vol. 36, no. 1, pp. 55–68, 2021.
- [12] L. P. Raj, V. Krishnaraj, and P. Soundarya, 'Evaluation of IoT Analytics - ThingSpeak Internet of Thingscloud platform for real-time patient health data monitoring dashboards: performance and scalability,' *Cloud Computing and Data Science*, vol. 2, no. 1, pp. 1–15, 2021.