

Morse Code Translation System

Tanishka Pankaj Jadhav, Akshar Rohidas Chaoghule, Lavanya Vijay Dani, Mrs. P. V Javkar

Students, Computer Technology

Guide, Computer Technology

Sou. Venutai Chavan Polytechnic, Pune, Maharashtra, India

Abstract: *This project presents a Low-Cost Morse Code Translation System designed as an assistive communication interface for individuals suffering from severe motor disabilities, such as Locked-in Syndrome (LIS) and ALS. Traditional eye-tracking systems are often prohibitively expensive and computationally demanding; this research proposes a non-invasive, wearable alternative using Infrared (IR) Reflectometry.*

The hardware architecture utilizes a Witty Fox IR Sensor mounted on a pair of safety goggles to detect eye-blink durations, which are processed by an Arduino Nano microcontroller. Using a temporal logic algorithm, the system distinguishes between "Dots" and "Dashes" to translate Morse patterns into English text. The translated data is transmitted via a serial bridge to a high-contrast p5.js visual dashboard, providing real-time feedback for caregivers.

Results indicate a high degree of accuracy in indoor environments, offering a functional, power-efficient, and accessible communication tool that costs a fraction of commercial medical-grade alternatives.

Keywords: *Low-Cost Morse Code*

I. INTRODUCTION

1.1 Background

This project falls under the domain of Assistive Technology (AT) and Human-Computer Interaction (HCI), specifically focusing on rehabilitative engineering for patients with severe speech and motor impairments. In the medical field, conditions like Amyotrophic Lateral Sclerosis (ALS), brainstem strokes, and traumatic spinal cord injuries often lead to "Locked-In Syndrome." In this state, a patient remains fully conscious and cognitively intact but suffers from total paralysis of nearly all voluntary muscles, except for vertical eye movements and blinking. Modern engineering seeks to bridge this gap by treating biological signals—such as the infrared reflectance of an eyelid—as digital inputs. By integrating embedded systems (Arduino) with web-based visualization (p5.js), we can transform a simple physiological reflex into a sophisticated, binary-coded communication linguistic tool.

1.2 Problem Statement

The primary challenge in the current healthcare landscape is the "Accessibility Gap" created by high-cost medical infrastructure. While advanced eye-tracking systems exist, they typically cost thousands of dollars, require high-end computing hardware, and are notoriously sensitive to ambient lighting and head movement, leading to "calibration drift." Furthermore, many assistive devices are invasive or require complex electrode placement (EOG), which can cause skin irritation and patient discomfort over time. There is a critical lack of low-cost, non-invasive, and portable communication aids that can function in resource-constrained environments. This project addresses the need for a simplified, "plug-and-play" hardware solution that does not require expensive cameras or proprietary software, ensuring that the basic human right to communicate is not limited by a patient's financial status.

1.3 Objective

The primary objective of this project is to design and implement a wearable, Infrared-based Morse Code Translation System that is both affordable and highly accurate. The project aims to:

Copyright to IJARSCT
www.ijarsct.co.in



DOI: 10.48175/IJARSCT-32566



449

Develop a wearable "Smart Goggle" frame that houses an Arduino Nano and an IR reflectance sensor.
Engineer a robust C++ algorithm capable of distinguishing between natural involuntary blinks and intentional Morse "Dots" and "Dashes" based on temporal duration.
Create a high-contrast, real-time p5.js dashboard that provides immediate visual feedback to caregivers.
Achieve a system cost-point of under \$20 to ensure global accessibility for paralyzed patients.

1.4 Importance

In real-world clinical and home-care settings, the ability to express basic needs—such as "hunger," "pain," or "discomfort"—can drastically improve the quality of life and mental health of a paralyzed patient. This project matters because it democratizes assistive technology. By using open-source components and the universal language of Morse code, we provide a reliable "voice" to those who are digitally silenced. Furthermore, its low power consumption and portability mean it can be deployed in rural clinics or areas with unstable electricity, where expensive medical computers cannot operate. Ultimately, this work serves as a proof-of-concept that high-impact medical solutions do not always require high-complexity hardware; they require empathetic, efficient engineering.

II. LITERATURE REVIEW

2.1 Evolution of Assistive Communication Interfaces

The academic foundation of this project is built upon decades of research into Augmentative and Alternative Communication (AAC). Early literature in the 1980s and 90s focused primarily on Electromyography (EMG) and Electrooculography (EOG), which recorded the electrical potential of muscle movements or eye rotations using skin-contact electrodes. While these studies proved that biological signals could be digitized, they highlighted significant drawbacks, including signal drift due to skin perspiration and the invasive nature of adhesive sensors. Modern research has shifted toward Non-Invasive Opto-electronics, leveraging the reflective properties of the human eye to minimize patient discomfort while maintaining high data integrity.

2.2 Infrared Reflectometry vs. Video-Based Tracking

A critical area of current study involves the comparison between Video-Based Eye Tracking (VET) and Infrared (IR) Reflectometry. Research by Shih et al. (2010) demonstrates that while high-speed cameras offer high precision for "gaze-point" detection (identifying exactly where a patient is looking on a screen), they suffer from high computational overhead and are sensitive to ambient light interference. In contrast, IR-based systems—like the one utilized in this project—operate on the principle of Photometry. By measuring the difference in infrared reflectance between the moist surface of the cornea (low reflectance) and the skin of the eyelid (high reflectance), a binary state (Open vs. Closed) can be established with minimal processing power. This makes Arduino-based IR sensing a superior choice for "Binary Communication" (Morse Code) over complex computer-vision algorithms.

2.3 Morse Code as a Linguistic Protocol

The use of Morse code in assistive technology is supported by the "Cognitive Load Theory." Studies in Human-Computer Interaction suggest that for patients with limited motor control, a "Single-Switch" input (a blink) combined with a rhythmic code is more sustainable than "Gaze Dwelling." Morse code provides a structured linguistic framework that allows for a theoretically infinite vocabulary using only two timing states: the Dot and the Dash. Recent experiments, such as Google's "Hello Morse" (2018), have validated that patients can achieve significant communication speeds once muscle memory is established, proving that Morse remains a robust and relevant protocol for modern rehabilitative engineering.



2.4 Summary of Gaps in Existing Literature

While the literature extensively covers high-end medical devices, there is a distinct "Research Gap" regarding low-cost, platform-independent solutions. Most existing studies assume the availability of a dedicated Windows PC or specialized medical monitors. This project addresses this gap by utilizing a Cross-Platform Serial Bridge (p5.js), allowing the hardware to interface with any modern web browser. By synthesizing established IR sensing techniques with contemporary web technologies, this research provides a scalable model for affordable assistive engineering.

III. METHODOLOGY

The methodology of this project follows a Modular Systems Engineering approach, where the hardware acquisition, signal processing, and visual representation are treated as distinct yet interconnected layers. The core logic relies on the transduction of physical eyelid movement into digital timing intervals.

3.1 System Architecture (Block Diagram)

The system architecture is designed as a linear data pipeline. It begins with the Physical Layer (the user's eye), moves through the Processing Layer (the microcontroller), and terminates at the Application Layer (the web dashboard).

Data Acquisition: The Witty Fox IR Sensor emits a continuous 950nm beam. The reflection is captured by a phototransistor.

Signal Conditioning: An onboard LM393 comparator converts the analog reflection into a clean 5V (HIGH) or 0V (LOW) digital signal.

Logic Processing: The Arduino Nano uses a "Polling Routine" to measure how long the signal remains HIGH.

Communication Bridge: The translated character is sent via USB-Serial to the p5.js server.

Visual Output: The web interface renders the character in a high-contrast format for the caregiver.

3.2 The Proposed Algorithm

The system operates on a Temporal Threshold Algorithm. This approach is superior to simple "State Detection" because it allows the system to ignore natural biological noise while identifying intentional communication patterns.

Initialize: Setup Digital Pin 2 for input and open the Serial port at 9600 Baud.

Detection: Continuously poll the sensor. If the state changes from LOW to HIGH, start a timer using millis().

Measurement: When the state returns to LOW, calculate the total duration (Ttotal).

Classification: * If $400\text{ms} < T$

total

$< 1200\text{ms}$, record a DOT.

If $1200\text{ms} < T$

total

$< 2500\text{ms}$, record a DASH.

If T

total

$< 400\text{ms}$, discard as NOISE (reflex blink).

Timeout: If no signal is detected for 3000ms, trigger the Translate() function.

Mapping: Search the Morse Look-up Table for the recorded pattern and send the corresponding ASCII letter to the PC.

3.3 Flow of the System

The operational flow ensures that the device remains "Real-Time" and "Non-Blocking." The flowchart below illustrates the decision-making process of the Arduino Nano as it manages incoming eye-blink data.



Phase 1: Hardware Calibration

Before usage, the 10k potentiometer is adjusted to set the reference voltage. This ensures the "Trigger Point" is perfectly aligned with the user's specific skin reflectance and the ambient light of the room.

Phase 2: Signal Transduction

As the user blinks, the IR sensor generates a pulse-width modulated (PWM) signal where the width of the pulse represents the "On-Time" of the blink.

Phase 3: Data Serialization and Display

The translated byte is sent through the Serial port. The p5.serialcontrol bridge picks up this byte and pushes it to the browser. The browser then executes a "Redraw" command, updating the screen instantly so the caregiver can read the patient's intent without delay.

IV. IMPLEMENTATION

The implementation phase involved translating the theoretical framework into a functional hardware-software prototype. This chapter details the technical stack, the modular breakdown of the code, and the physical execution of the communication bridge.

4.1 Tools and Technologies Used

To ensure the system is both lightweight and cross-platform, a hybrid stack of embedded C++ and web technologies was selected:

Arduino IDE (Embedded C++): Used for developing the firmware for the ATmega328P (Arduino Nano). This environment was chosen for its robust timing libraries and low-level hardware control.

p5.js (JavaScript Library): A specialized library for "creative coding" used to build the visual dashboard. It allows for high-contrast, scalable UI rendering that can run in any modern web browser.

p5.serialcontrol: A desktop application that serves as the Serial-to-WebBridge. It bypasses browser security sandboxes to allow the website to read data directly from the USB port.

Witty Fox IR Module: The primary hardware sensing unit, chosen for its onboard LM393 comparator which simplifies the digital signal processing.

4.2 Project Modules

The project is divided into four critical functional modules that operate in a synchronized loop:

Module 1: The Sensing & Calibration Unit

This hardware module consists of the IR emitter/receiver pair and the 10k Potentiometer.

Task: The potentiometer sets the "V-Ref" (Reference Voltage).

Output: A clean digital pulse sent to the Arduino's interrupt-capable pins.

Module 2: The Temporal Analysis Engine (Firmware)

Running on the Arduino Nano, this module is the "Brain" of the system.

Task: It calculates the pulse width using millis(). It filters out "noise" (blinks < 400ms) and categorizes dots and dashes based on the Timing Window Algorithm.

Module 3: The Morse Mapping & Translation Module

This module contains the Parallel Array Structure discussed in Section 4.3.

Task: Once a sequence is finished (3-second timeout), it performs a linear search.

Result: It identifies the ASCII character (e.g., 'S') and pushes it into the Serial buffer.

Module 4: The Visual Dashboard (UI)

The final module is the web-based interface.



Task: It listens for incoming bytes, stores them in a string, and renders them in Extra-Large (150pt) font.

Feature: It includes a "Clear Screen" function to reset the conversation once a sentence is finished.

4.3 System Implementation Screenshots

The following visual documentation represents the system during active operation and testing.

1. The Serial Bridge Interface:

This screenshot shows the p5.serialcontrol application active and connected to COM3, confirming that data packets are flowing from the goggles to the laptop.

2. The Visual Dashboard in Action:

This screenshot displays the final web interface. Notice the high-contrast dark mode and the large "Last Character" display designed for caregiver visibility.

3. Hardware Integration on Goggles:

This image captures the physical implementation, showing the Arduino Nano and IR sensor securely mounted on the safety goggles, with the potentiometer accessible for real-time tuning.

4.4 Code Implementation Logic

The implementation relies on "Non-Blocking" code. By avoiding the delay() function in C++, the Arduino remains responsive to the sensor at all times. If the user blinks while the system is "Thinking," the software will still catch the start of that blink, ensuring no data loss during high-speed communication sessions.

V. RESULTS AND DISCUSSION

This chapter evaluates the functional performance of the Morse Code Translation System. The results are based on empirical testing conducted to measure the accuracy, latency, and usability of the integrated hardware and software stack.

5.1 Prototype Evaluation and Results

The final prototype successfully achieved the objective of a wearable, lightweight communication aid. The integration of the Arduino Nano onto the safety goggles resulted in a stable platform that maintains a consistent distance between the Witty Fox IR Sensor and the user's eyelid.

5.1.1 Accuracy and Environmental Testing

To determine the reliability of the system, testing was performed under varying light conditions. The "Bit Accuracy" refers to the system's ability to correctly identify an individual Dot or Dash, while "Translation Accuracy" refers to the successful mapping of a full Morse sequence to a letter.

Lighting Condition	Bit Accuracy (%)	Translation Accuracy (%)	Latency (ms)
Indoor (Fluorescent)	98%	96%	< 50ms
Dim Light (Night)	99%	98%	< 50ms
Direct Sunlight	88%	82%	< 50ms

The results indicate that the system is most stable in indoor and dim-light environments, which are typical for hospital wards or home-care settings. The drop in accuracy during direct sunlight was successfully mitigated by manually adjusting the 10k Potentiometer to increase the threshold voltage



5.1.2 Signal Stability and Noise Filtering

A key result of the implementation was the effectiveness of the 400ms software filter. During testing, it was observed that natural, involuntary "micro-blinks" lasted between 150ms and 250ms. Because the algorithm ignores any signal shorter than 40

VI. CONCLUSION

The development of the Morse Code Translation System successfully demonstrates that high-impact medical solutions can be engineered using affordable, open-source components. By synthesizing Infrared Reflectometry with a custom Temporal Logic Algorithm, this project achieved its primary objective: providing a reliable, non-invasive, and low-cost communication "voice" for individuals with severe motor disabilities. The final prototype, consisting of an Arduino Nano and a high-contrast p5.js dashboard, proved capable of translating intentional eye-blinks into digital text with a bit-accuracy of over 95% in controlled environments.

Throughout this project, I gained significant technical expertise in hardware-software integration, particularly in managing real-time serial data and hardware debouncing to filter biological noise. Beyond the technical scope, I learned the importance of User-Centric Design in assistive technology—recognizing that ergonomics and visual accessibility are just as critical as the underlying code. While limitations like ambient light sensitivity exist, the project serves as a robust proof-of-concept for democratizing healthcare technology. It confirms that with efficient engineering, the gap between a paralyzed patient and their caregiver can be bridged for less than the cost of a standard medical consultation.

VII. FUTURE SCOPE

While the current prototype successfully translates eye-blinks into Morse code, there is significant potential for enhancing the system's intelligence, usability, and integration. Future iterations of this project can focus on the following key areas:

7.1 Adaptive Machine Learning Algorithms

The current software relies on "Hard-Coded" timing windows (e.g., a Dash is exactly >1200ms). A major improvement would be the implementation of Adaptive Calibration. Using a simple machine learning model, the Arduino could "learn" a specific user's blinking rhythm during a 30-second initialization phase. This would allow the system to automatically adjust its thresholds for patients with slower reflexes or muscle fatigue, significantly reducing "Input Errors."

7.2 Predictive Text and Word Completion

To increase the Words-Per-Minute (WPM) rate, the p5.js Dashboard can be integrated with a T9-style Predictive Text library. Instead of blinking every letter of a word like "WATER," the user could blink "W-A" and select the full word from a generated list of suggestions. This "Shorthand" approach could improve communication speed by as much as 300%, making conversations more natural for the patient.

7.3 Wireless and IoT Integration

Transitioning from a wired USB connection to a Wireless Architecture would greatly enhance patient mobility.

Bluetooth/ESP32: Replacing the Nano with an ESP32 module would allow the goggles to connect wirelessly to a smartphone or tablet.

Smart Home Control: The system could be linked to an IoT Hub. Specific Morse sequences (e.g., "... --- ...") could automatically trigger a bedside alarm, turn on the lights, or send an emergency SMS to a doctor via a cloud server.



7.4 Binocular Redundancy

Future hardware designs should include a second IR Sensor for the other eye. Binocular Sensing would allow the system to differentiate between an intentional communication blink (both eyes) and a natural reflex or "eye twitch" (one eye). This redundancy would solve the current limitation of "False Positives" and provide a much more stable signal in high-stress environments.

7.5 Mobile Application Development

Developing a dedicated Android/iOS Application to replace the p5.js web dashboard would make the system more portable. A mobile app could utilize the phone's text-to-speech (TTS) engine to "speak" the translated Morse code aloud, allowing a paralyzed patient to have a literal voice in public spaces or family gatherings.

REFERENCES

- [1] C. H. Shih, L. A. Breedin, and T. P. Moran, "A New Assistive Communication System for People with Multiple Disabilities using IR Reflectometry," *Journal of Medical and Biological Engineering*, vol. 30, no. 5, pp. 225-231, 2010.
- [2] J. P. Hansen, "A Survey of Eye-Gaze Tracking Techniques and Applications in Assistive Technology," *IEEE Transactions on Human-Machine Systems*, 2018.
- [3] S. S. S. P. Rao and K. R. Rao, "Implementation of Morse Code Based Communication for Paralyzed Patients using Microcontrollers," *International Journal of Embedded Systems and Applications*, vol. 4, no. 2, 2014.
- [4] F. B. Felton and J. Williams, "Comparing Morse Code vs. Gaze Dwelling: A Study in Cognitive Load for Locked-In Patients," *IEEE Conference on Rehabilitation Robotics (ICORR)*, 2019.
- [5] M. Margolis, *Arduino Cookbook: Recipes to Begin, Expand, and Enhance Your Projects*, 3rd ed., Sebastopol, CA: O'Reilly Media, 2020.
- [6] J. Noble, *Programming Interactivity: A Designer's Guide to Processing, Arduino, and OpenFrameworks*, 2nd ed., Beijing: O'Reilly, 2012. (Used for p5.js and Serial communication logic).
- [7] Atmel Corporation, "ATmega328P 8-bit AVR Microcontroller with 32K Bytes In-System Programmable Flash," *Datasheet, Rev. 8161D-AVR-10/09*, 2025.
- [8] Texas Instruments, "LM393 Dual Differential Comparators," *Technical Specifications, SLCS007K*, 2024.
- [9] International Telecommunication Union (ITU), "Recommendation ITU-R M.1677-1: International Morse Code," [Online]. Available: <https://www.itu.int/rec/R-REC-M.1677-1-200910-I/en>.
- [10] The p5.js Foundation, "p5.serialport library documentation," 2026. [Online]. Available: <https://p5js.org/libraries/>.
- [11] Google Creative Lab, "Hello Morse: An Experiment in Accessible Communication," 2018. [Online]. Available: <https://experiments.withgoogle.com/morse>

