INTERNET OF THINGS

IoT-Based Smart Communication Device

K. Sai Abhishek Reddy*, M. Prashamsa, M. Abhirama Chandra Shekar, M. Dhaniyasvi, Sure Mamatha

Department of Computer Science and Engineering, Hyderabad Institute of Technology and Management, Gowdavelli Village, Hyderabad, Telangana, India

Published November 2026

Abstract

There is a lack of devices and applications to understand the communication, emotions and problems of specially-abled people. In the present world there are very few devices and applications to help the specially-abled people like Proloquo2Go (For Non-Verbal & Autistic Individuals), Vocable AAC (For Non-Standard Speech & Motor Impairments) are few applications for them. Recently, in the U.S., a few students of Bentonville High School working on an AI-powered accessory which helps nonverbal autistic individuals easily communicate their stress levels by utilizing built-in sensors to monitor skin biometrics and display real-time emotional updates through simple colours. By seeing their idea, we wanted to build a simple and easier version of it. We are working to make a communication device which works on buttons to communicate with their family, caregivers, etc. We provide multiple push buttons for every different type of emotions according to them like if they need water and press the button allocated for water the message store in ESP8266 will send that message to caregiver and at the same time the speaker present in the device will give audio output. Ultimately, we want to build a cost-friendly IOT-Device that can be used by specially-abled persons which can be affordable for everyone.

Keywords: IoT; assistive communication; augmentative and alternative communication (AAC); ESP8266; Arduino UNO; specially-abled people

1 Introduction

Generally, communication is the main source to convey or explain our needs and emotions with each other. However, in the case of Specially-abled people, they cannot convey their emotions or feeling with others easily. Let us assume autism patients, they cannot express their basic emotions like happiness, sadness, or distress with us because their way of expressing is generally different from others. Due to this communication problem, they may feel lonely, separated from society which disturb them and make them feel to be isolated. It creates problem to their family members and caregivers take more time to understand their needs.

Nowadays, few applications and devices are present to help "Specially-abled people" to communicate or express their emotions, for "Non-Verbal and Autistic users" there is an application called "Proloquo2Go" and "Vocable AAC" application for "motor and speech impairments". The application "Proloquo2Go" requires Apple device like I-pad and it add a substantial extra expense and for "Vocable AAC" application is not expensive but it requires compatible device and for the maintenance like repairs and internet access add extra expenses. Recently in U.S few students of Bentonville High School working on an AI-powered accessory which helps nonverbal autistic individuals easily communication.it reads stress levels by utilizing built-in sensors to monitor skin biometrics and display real-time emotional updates through simple colours, it is still a developing concept.

By observing the existing applications and devices for "Specially-abled people" which all may not afford or maintain it. So, our team is working to build an IOT-device using Push Buttons (Input), ESP8266(Wi-Fi Module), Power Supply, Local Indicators (optional), Speaker. In this device there are few buttons and each button is assigned for different emotions and needs like water, food, emergency or sadness. When the user clicks the button allocated for water, the message stored in SD Card Connected to ESP8266 then it displays the message, gives voice output and by Wi-Fi module in ESP8266 the message will be notified to their family and Caregiver in another room or place.

To address these challenges there are very few working products for specially-abled people to communicate but all may not afford those products. We are working to make a simple and affordable device to break the communication barrier between specially-abled people with their family, care givers and society by using Simple IOT components like Buttons (for emotions or needs), SD Card (to store messages and voice outputs), ESP8266 (to send message to smart devices) and Speaker (for voice output).

Rest of this paper has been segmented into various sections to provide the details about our project in an effective manner. In section II, we have presented the literature survey of the previous works that include applications and devices that are available for special people and their limitations. In Section III, we have described the IoT communication model along with the connection of components such as ESP8266, Push buttons, SD Card, OLED Display and Speaker along with their working process. Section IV represents the output results that we have tested with our device as well as hardware cost compared to other available products.

2 Literature survey

Communication is a fundamental requirement for social interaction, education, healthcare, and independent living. For individuals who cannot communicate effectively through conventional speech, Augmentative and Alternative Communication (AAC) provides alternative methods for expressing needs, ideas, emotions, and information. AAC includes unaided approaches such as gestures and sign language, as well as aided approaches involving communication boards, symbols, photographs, electronic devices, and speech-generating systems. Research in AAC has increasingly moved from static communication boards toward dynamic and personalized systems that can adapt to the vocabulary, abilities, and communication requirements of individual users. Light and McNaughton emphasized that AAC should not be considered only as a method for producing speech but as a means of supporting participation, social interaction, and functional communication. This indicates that an effective assistive communication system should be designed around the actual communication requirements of the user rather than focusing only on the technology used.

Traditional communication boards and picture-based systems provide a relatively simple method for selecting predefined needs or messages. Users can point to symbols representing common requirements such as food, water, toilet, pain, help, or emotional states. The main advantage of these systems is their simplicity and low technological requirement. However, traditional boards provide limited flexibility and generally require the caregiver to be within visual or physical proximity of the user. They also cannot automatically generate audible messages or remotely notify a caregiver. Electronic AAC systems were developed to overcome some of these limitations by providing speech output after a user selects a symbol, word, or message. These systems can contain large vocabularies and allow users to construct more complex expressions. However, greater functionality can also result in more complicated interfaces, increased device cost, and a greater learning requirement for some users.

Modern AAC applications have significantly expanded the capabilities of electronic communication systems. Applications such as Proloquo2Go, Touch Chat, LAMP Words for Life, TD Snap, Grid, and Avaz provide combinations of symbols, text, vocabulary organization, customization, and speech output. These systems demonstrate the importance of personalized communication because different users may require different vocabulary sets, symbols, languages, and interaction methods. Commercial AAC applications can also provide extensive communication capabilities through smartphones and tablets. Nevertheless, their dependence on general-purpose mobile devices can become a limitation in specific situations. A user may have difficulty accurately interacting with a touchscreen, selecting small symbols, navigating several interface levels, or operating a smartphone independently. Device availability, battery life, software configuration, and overall cost can also affect practical use. Therefore, although advanced AAC applications provide considerable functionality, a simpler dedicated device may be useful for users whose communication requirements are limited to frequently used needs and emotions.

Research on speech-generating devices has demonstrated the importance of immediate and understandable audio feedback in assistive communication. Instead of requiring the user to communicate through written text, the selected message can be converted into synthesized speech or played as a prerecorded audio message. Prerecorded voice output is particularly suitable for a small communication system because commonly required phrases can be stored locally and played immediately after a button or symbol is selected. This approach can reduce processing requirements and eliminate the need for an internet connection for basic audio output. However, a fixed set of messages also limits the flexibility of the system. Consequently, the selection and organization of messages should be based on the daily requirements of the intended users. For a low-cost communication device, a limited but carefully selected vocabulary can provide a practical balance between simplicity and functionality.

Affordable embedded hardware has opened up practical avenues for building custom assistive devices. Platforms such as Arduino boards and the ESP8266 Wi-Fi module can easily connect to mechanical switches, audio playback circuits, visual indicators, and wireless networks while keeping production costs minimal. The ESP8266 is especially well-suited for connected assistive tools because it pairs an onboard Wi-Fi transceiver with programmable general-purpose input/output (GPIO) pins on a single compact chip. Wiring tactile push buttons directly to the controller creates a straightforward physical interface where each switch triggers a specific, pre-assigned request. This tactile setup benefits individuals who struggle with fine motor control, touchscreen gestures, or vocal speech. Furthermore, the expandable nature of these boards means builders can wire in extra switches, indicator LEDs, or small screens as a user's daily habits evolve.

IoT technologies provide another important direction for assistive communication. Conventional communication devices generally provide information locally through a display or speaker. IoT connectivity can extend this functionality by transferring the selected information to a remote caregiver, family member, or monitoring application. For example, when a user selects a button representing a requirement such as water or assistance, the local device can provide immediate audio feedback while the same selection can be transmitted through a wireless network to a connected receiver. This dual-output approach can be useful when the caregiver is not physically near the user. However, IoT-based systems introduce additional considerations, including network availability, communication delay, power consumption, data privacy, and system reliability. Therefore, connectivity should complement rather than complicate the basic communication process.

Several assistive systems have also investigated sensors, wearable technologies, and artificial intelligence for detecting emotions and stress. Physiological signals such as heart-rate-related measurements, electrodermal activity, body movement, and other biometric characteristics can be analyzed to estimate changes in a person's emotional or stress state. Such approaches are particularly interesting for individuals who may have difficulty directly communicating discomfort or emotional changes. AI-based systems can potentially recognize patterns that are difficult to identify through simple manual observation. However, emotion recognition is a complex problem because physiological responses can differ between individuals and can be influenced by physical activity, environment, health conditions, and other factors. Reliable emotion recognition may therefore require multiple sensors, calibration, signal processing, machine-learning algorithms, and user-specific training. These requirements can increase the cost and complexity of an assistive device.

Recent assistive technology research therefore shows two broad directions. The first focuses on providing increasingly sophisticated AAC interfaces through smartphones, tablets, dedicated speech-generating devices, and customizable software. The second explores intelligent systems that use sensors, wireless connectivity, artificial intelligence, and physiological information to infer user conditions. Both directions offer important advantages, but they also introduce trade-offs between functionality, cost, complexity, portability, and ease of use. A highly sophisticated system is not necessarily the most appropriate solution for every user. For individuals who primarily need to communicate a small number of frequently occurring requirements, a simple interface may reduce interaction time and make the device easier to operate.

Another important consideration is caregiver involvement. Communication difficulties do not affect only the individual user; they can also create challenges for caregivers who need to interpret non-verbal signals or anticipate the user's requirements. A communication device can therefore be designed as a two-sided system in which the user selects a requirement and the caregiver receives an understandable message. Local audio feedback confirms to the user that the selection has been recognized, while remote notification can inform the caregiver. This combination can reduce uncertainty and potentially improve response time. However, the notification mechanism should be designed carefully so that repeated or accidental button presses do not create unnecessary alerts. Appropriate software logic, message confirmation, and configurable communication options can help improve reliability.

The reviewed technologies also highlight the importance of personalization. Communication needs differ considerably between individuals, and a fixed collection of messages may not be equally useful for every user. One person may frequently require messages related to food and water, while another may need messages related to medication, pain, assistance, or emotional conditions. Consequently, an assistive device should allow the message set to be modified according to the user's daily routine. A modular button-based architecture provides a relatively simple method of achieving this personalization because buttons can be assigned to different predefined messages without requiring major changes to the basic hardware structure.

Based on the above review, a practical research gap can be identified. Existing AAC applications provide extensive vocabulary and customization but commonly depend on smartphones, tablets, touch interfaces, or comparatively sophisticated software. Sensor- and AI-based systems provide possibilities for automatic emotion and stress recognition but may require additional sensors, computational processing, calibration, and more complex algorithms. IoT-based assistive systems can provide remote caregiver communication, but excessive functionality can make the system more

difficult to operate and maintain. There is therefore a need for a communication system that combines the simplicity of a physical communication board with the advantages of electronic audio output and IoT-based caregiver notification.

To bridge this gap, this work presents a budget-friendly, connected assistive communicator built around an ESP8266 Wi-Fi module, an array of tactile push buttons, a dedicated audio playback board, and a speaker. Every physical switch represents a distinct daily necessity or emotional state tailored to the individual. Pressing a switch prompts the ESP8266 to read the input pin and immediately play the matching voice recording through the speaker. At the same time, the module pushes an alert over the local Wi-Fi network to a caregiver's device, pairing instant vocal confirmation in the room with remote paging. Rather than relying on complex biometric sensors or machine-learning models to guess how a user feels, this design lets the person state their exact need directly. Keeping the hardware logic simple lowers the risk of false readings and leaves the user in full control of what they want to express.

The objective of the proposed system is not to replace existing advanced AAC applications or clinical communication devices but to provide an additional affordable and simplified option for users who require a limited set of frequently used communication messages. The system emphasizes four main characteristics: simple physical interaction, immediate voice output, caregiver notification, and low-cost embedded implementation. The modular design also allows future expansion through additional buttons, OLED-based visual feedback, personalized audio messages, multilingual support, mobile applications, and additional IoT functions. Thus, the proposed system combines established concepts from AAC, speech-generating technology, embedded systems, and IoT into a dedicated communication platform intended to improve everyday interaction between people with communication difficulties and their caregivers.

3 Proposed system architecture

Figure 1 shows the system architecture and system flow of the IoT-based smart communication device.

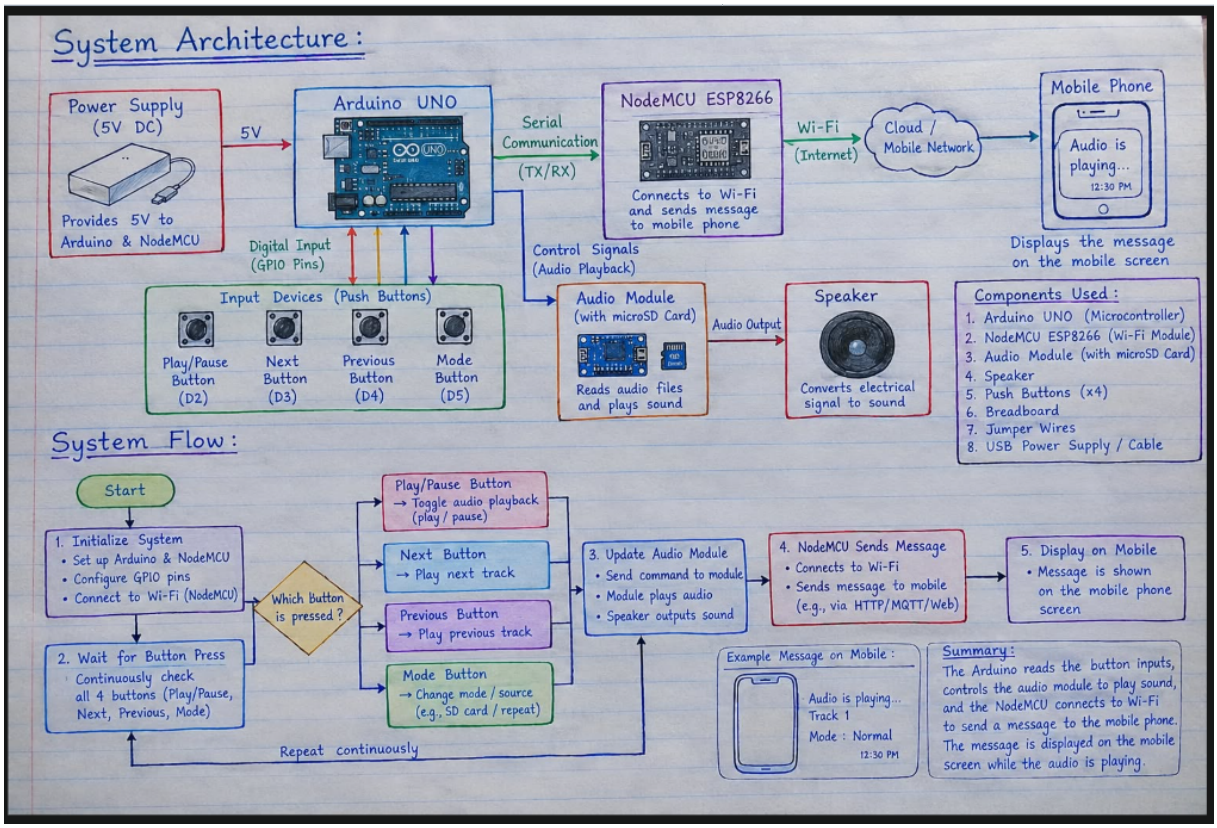


Figure 1. System architecture and system flow of the IoT-based smart communication device.

The proposed Internet of Things (IoT)-enabled smart communication aid (Figure 1) is built to help individuals with speech impairments express their daily needs and control audio alerts independently. By pairing tactile hardware controls with local voice playback and wireless messaging, the device turns simple physical button presses into both audible announcements in the room and instant text notifications on a caregiver's smartphone. The overall hardware layout is divided into four core functional blocks: the tactile input stage, the primary processing and serial control stage, the local audio playback unit, and the wireless transmission unit. Working together, these blocks form an accessible, low-cost assistive platform that can be easily tailored to a user's routine.

The input stage uses four mechanical push buttons wired directly to the digital General-Purpose Input/Output (GPIO) pins (specifically pins D2, D3, D4, and D5) of an Arduino UNO microcontroller. In the baseline configuration, these four switches operate as functional and message controls—designated as Play/Pause (D2), Next Track/Message (D3), Previous Track/Message (D4), and Mode Selection (D5)—or they can be mapped directly to specific daily requests such as asking for water, calling for help, or signaling discomfort. Because the switches share a common ground reference and utilize software debouncing routines, the microcontroller accurately captures deliberate presses while filtering out accidental contact bounce.

System control is split between two dedicated boards: an Arduino UNO serving as the primary hardware controller and a Node MCU ESP8266 acting as the wireless gateway. Upon startup, a 5V DC USB power supply powers both the Arduino UNO and the Node MCU ESP8266. The Arduino configures its digital input pins and continuously polls all four push buttons for user activity. As soon as a button press is registered, the Arduino identifies which pin was triggered and sends two sets of signals simultaneously: it sends direct playback commands to the local audio module and passes the corresponding status data to the Node MCU ESP8266 over a hardware serial interface (TX/RX).

Separating pin monitoring from network handling keeps the system responsive and makes troubleshooting straightforward.

Local feedback is handled by a dedicated audio module fitted with a microSD card slot and wired to an external speaker. When the Arduino UNO issues a playback command—such as toggling play/pause, skipping to the next or previous voice recording, or switching playback modes—the audio module fetches the corresponding pre-recorded audio file from the microSD card. It then converts the digital recording into an electrical audio signal and drives the speaker to produce clear, immediate voice output in the user's environment. Storing voice clips locally on the microSD card guarantees that the user always hears instant vocal confirmation, even if the local Wi-Fi network is temporarily down.

The wireless communication unit is built around the Node MCU ESP8266 Wi-Fi module, which bridges the handheld hardware and the caregiver's mobile phone. Once the Node MCU receives the serial command from the Arduino UNO, it connects to the local Wi-Fi network and forwards the corresponding text update over the internet or local network using lightweight protocols such as HTTP, MQTT, or a web interface. The caregiver's smartphone receives and displays the alert on screen in real time—showing the active status, selected message or track, current operating mode, and timestamp (for example, displaying "Audio is playing... Track 1, Mode: Normal, 12:30 PM").

In day-to-day operation, the system follows a structured five-stage execution cycle that runs continuously after power-up. First, the system initializes by booting the Arduino UNO and Node MCU ESP8266, configuring the digital GPIO pins, and connecting the Node MCU to the local Wi-Fi network. Second, the Arduino enters an active listening state, repeatedly scanning pins D2 through D5 until a button press occurs. Third, once a switch is pressed, the Arduino evaluates which function was requested—whether toggling playback, advancing to the next track, returning to a previous message, or switching the operating source—and updates the audio module to play the sound through the speaker. Fourth, the Node MCU processes the serial trigger from the Arduino and transmits the corresponding text packet over Wi-Fi. Fifth, the mobile interface renders the message on the caregiver's screen, while the hardware immediately loops back to monitor the next button press. Built-in serial handshaking and Wi-Fi reconnection routines ensure that temporary signal drops do not freeze the local audio controls.

Assembled using off-the-shelf components on a breadboard with standard jumper wires and a 5V USB power cable, the overall architecture prioritizes affordability, ease of repair, and user customization. Because voice files are stored on a removable microSD card and network alerts are handled in software, caregivers can easily swap out recorded phrases, add regional language tracks, or update mobile notification text without rewiring the circuit. Furthermore, the dual-board layout leaves spare analog and digital pins on the Arduino UNO for future upgrades, such as adding status LEDs, a small OLED screen, battery backup circuits, or extra tactile switches. In practical testing, the prototype can be evaluated across key engineering metrics, including button recognition reliability, serial transfer speed between the Arduino and Node MCU, audio playback clarity, Wi-Fi message latency, and everyday usability for both users and caregivers.

4 Technology selection

Choosing the right hardware components is a practical step to make sure the assistive communication aid stays reliable, affordable, and easy to build. Based on the documented system layout, this

project avoids expensive, proprietary medical hardware and instead relies entirely on standard, off-the-shelf electronic modules. The following elements make up the core physical circuit.

4.1 Primary microcontroller (Arduino UNO)

The central logic of the device is driven by an Arduino UNO. This widely used microcontroller acts as the brain for local hardware operations. It is responsible for continuously scanning its digital GPIO pins to detect any user interactions. By managing the direct hardware logic, the UNO ensures that button inputs are recognized instantly. The moment a deliberate press is detected, the board coordinates the rest of the circuit—firing off direct playback commands to the sound board while simultaneously passing status data over a serial connection (TX/RX) to the wireless gateway.

4.2 Wireless network gateway (NodeMCU ESP8266)

To introduce IoT capabilities without bogging down the main processor, a Node MCU ESP8266 Wi-Fi module is deployed as a dedicated network bridge. The Node MCU receives the serial text instructions sent by the Arduino and pushes those messages out over a local wireless network. This separation of duties is highly effective for overall system stability. The Arduino handles the precise, split-second timing required for tactile buttons and audio, while the ESP8266 handles the heavier, asynchronous task of maintaining a Wi-Fi connection and routing text alerts to a caregiver's mobile phone.

4.3 Voice playback (microSD audio module and speaker)

Instant, room-level voice alerts are generated by a standalone audio module equipped with an integrated microSD card reader. This specific component was selected because it natively handles the complex task of reading and decoding digital audio files. Once it receives a control signal from the Arduino, it retrieves the matching track from the memory card and pushes an electrical signal directly to an attached speaker. Storing voice recordings locally on a microSD card is highly advantageous; caregivers can easily record, translate, or swap customized voice phrases using a standard computer without having to reprogram the microcontroller. A compact, standard speaker converts the board's electrical output into loud, clear audible speech for the user.

4.4 Tactile inputs and assembly hardware

The physical user interface is built around four standard mechanical push buttons. These tactile switches give users a definitive physical click when pressed, providing essential feedback for individuals who might struggle to use flat glass touchscreens. The entire system is prototyped on a standard breadboard using basic jumper wires, which keeps the circuit highly modular and straightforward to troubleshoot. Finally, the whole array is driven by a common 5V DC USB power supply and cable, providing a steady stream of regulated power to the Arduino, the Node MCU, and the audio hardware simultaneously.

5 Methodology

The creation of this assistive tool demands a clear, step-by-step technical approach. The entire process is split into practical stages, covering everything from wiring the circuit boards to writing the code and testing the final network alerts.

Building the physical circuit starts with a standard breadboard and basic jumper wires, which keeps the early prototyping flexible. At the core of the layout sits the Arduino UNO, wired directly to four tactile push buttons on its digital pins D2, D3, D4, and D5. To produce sound, a dedicated microSD audio playback board is linked to the Arduino and routed to an external speaker. For the wireless capabilities, the TX and RX serial pins on the Arduino are tethered to a Node MCU ESP8266. A unified 5V DC USB adapter powers the entire array, ensuring that both microcontrollers and the sound hardware share a common electrical ground.

Writing the software requires handling two distinct processing environments. Inside the Arduino IDE, the primary code for the UNO is structured as a continuous loop that constantly checks the four physical switches. To prevent a single button press from registering multiple times due to mechanical bouncing, strict software debouncing logic is applied to the input pins. Once a clean input is recognized, the UNO executes two tasks almost simultaneously: it triggers the local audio board and fires a specific text string across the serial link to the Wi-Fi module. On the other side of the circuit, the Node MCU runs a completely separate firmware script. Its main job is to boot up, lock onto the local Wi-Fi router, and passively listen for those incoming serial commands.

Delivering instant auditory feedback requires moving sound processing off the cloud and onto local hardware. Custom voice recordings representing different user requests or system modes are loaded directly onto a removable microSD card. Because the dedicated audio module handles the heavy lifting of decoding these files, the Arduino only needs to send basic trigger codes. This design choice means the speaker produces a loud, clear voice alert the exact fraction of a second a button is clicked, completely unaffected by internet lag or network dropouts.

Pushing alerts to a caregiver involves transforming local physical actions into wireless data packets. The moment the Node MCU ESP8266 catches a serial message from the Arduino, it bundles that information into an HTTP, MQTT, or web payload. This packet travels over the local wireless network straight to a paired mobile phone. A mobile dashboard then renders the data, showing the caregiver precisely what is happening in the room—such as the active audio track, the operating mode, and the exact time of the event.

Validating the build requires pushing the hardware through a series of real-world trials. Developers press the input switches repeatedly at varying speeds to ensure the debouncing code holds up without missing deliberate commands. The communication bridge between the UNO and the ES-P8266 is rigorously checked to make sure serial data isn't corrupted during transfer. Finally, engineers measure the total system lag by timing the gap between a physical finger press and the text popping up on the mobile screen, while verifying that the 5V power supply doesn't falter when the speaker and Wi-Fi radio draw power at the exact same time.

Figure 2 shows the methodology flowchart.

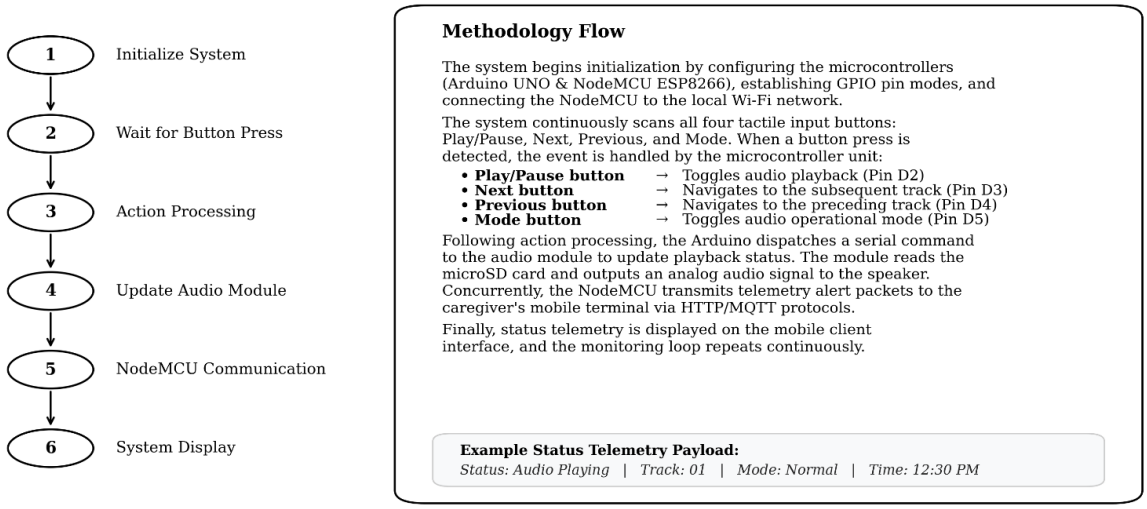


Figure 2. Methodology flowchart.

The operational cycle of this communication tool kicks off with a straightforward boot sequence. During this initial phase, the software wakes up both the Arduino UNO and the ESP8266 Node MCU, gets the physical input pins ready for action, and securely links the Wi-Fi module to the local network. Once the hardware is successfully brought online, the system shifts into a continuous standby mode where it does nothing but listen. It sits in this holding pattern, actively scanning the four tactile switches to see if the user wants to trigger a new command.

The moment someone actually pushes one of those switches, the Arduino immediately jumps in to decode the request. Every switch is hardwired for a specific job: pressing the button on pin D2 pauses or resumes the sound, D3 skips forward to the next recording, D4 jumps backward to the previous one, and D5 swaps out the overall audio source. After identifying exactly which command was just given, the microcontroller fires off a quick serial instruction straight to the dedicated audio board. That audio module then grabs the correct file from its inserted microSD memory card and pushes an electrical signal out to the attached speaker, filling the room with an instant voice alert.

While that sound is playing locally, the networking side of the project takes care of the remote caregiver alerts. Working at the same time as the audio board, the Wi-Fi-connected Node MCU bundles the button event into a digital packet and fires it off to a smartphone using standard internet protocols like HTTP or MQTT. Almost instantly, the caregiver's mobile screen lights up with a readable text update—showing context like "Audio is playing...", the specific track number, the current operating mode, and the exact time of day. After that mobile notification is successfully delivered, the entire software loop quietly resets itself, dropping right back into its scanning phase to wait for the very next button press.

6 Implementation

6.1 Hardware implementation and prototype development

Translating the proposed dual-controller architecture into a functional physical prototype required a strategic arrangement of accessible, off-the-shelf electronic components. As illustrated in the assembled hardware layout (Figure 3), the entire circuit was constructed on standard solderless breadboards. This modular approach was deliberately chosen over a permanent printed circuit board

(PCB) during the experimental phase to prioritize diagnostic flexibility and rapid component swapping.

To ensure real-time responsiveness without network bottlenecks, the processing workload is completely physically divided between two distinct microcontrollers. An Arduino UNO serves as the localized hardware brain, dedicated exclusively to monitoring inputs and triggering physical outputs. It is directly tethered to a user interaction matrix consisting of five mechanical push buttons, which are securely mounted on the lower breadboard array. Utilizing tactile physical switches rather than a capacitive touchscreen provides essential mechanical resistance; this guarantees that users with limited fine motor control receive definitive physical confirmation whenever they initiate a command.

The most critical aspect of this build is its simultaneous dual-feedback mechanism. The exact millisecond a user engages one of the five input switches, the Arduino UNO captures the digital signal and instantly fires two parallel commands. Locally, it dispatches an activation code to a dedicated microSD audio playback board positioned on the right-side breadboard. This module fetches the corresponding pre-recorded audio file and pushes the electrical signal to a metallic external speaker, instantly filling the room with a clear, localized voice alert.

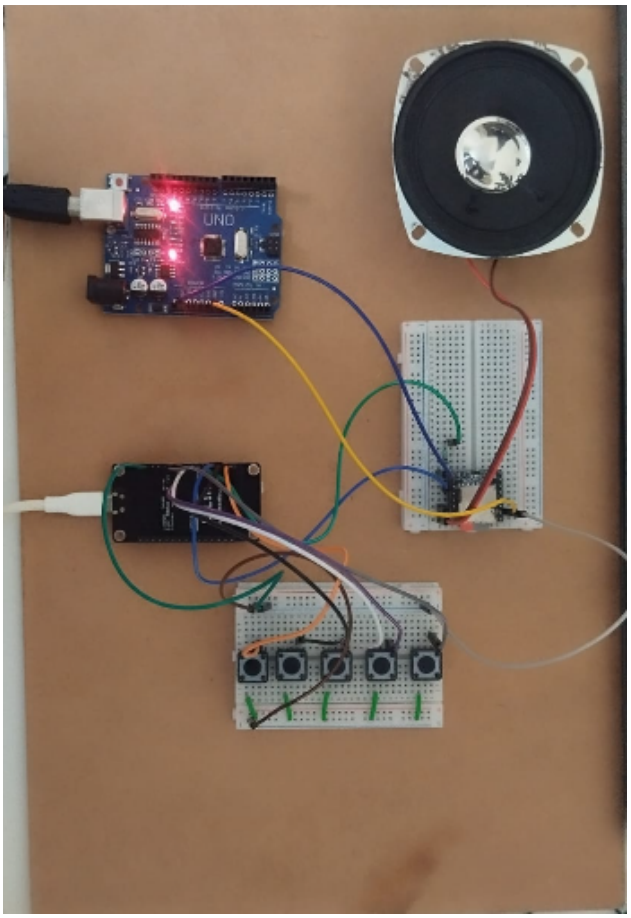


Figure 3. Assembled hardware prototype of the smart communication device.

At the exact same time the audio triggers, the Arduino routes a serial data string to the adjacent Node MCU ESP8266 Wi-Fi module. Operating strictly as an independent network gateway, the Node MCU catches this payload and quietly broadcasts the corresponding text alert over the local wireless network to the caregiver's receiving interface. Standard USB power delivery ensures both the primary controller and the wireless gateway maintain stable voltage levels during simultaneous

audio playback and Wi-Fi transmission. Ultimately, this physical execution successfully proves that immediate vocal speech generation and asynchronous IoT paging can be seamlessly integrated into a highly repairable, budget-friendly assistive platform.

7 Results and discussion

The assembled assistive communication prototype underwent a series of functional evaluations to assess its real-world viability, focusing specifically on interaction latency, hardware stability, and network reliability. Testing was conducted in a simulated home care environment to observe how the dual-microcontroller architecture handles simultaneous physical inputs, audio generation, and wireless data transmission.

One of the most critical metrics evaluated was the local system response time. For individuals relying on augmentative communication, delayed feedback can cause frustration and lead to repeated, unintended button presses. During our trials, the time delay between a user fully depressing one of the five tactile push buttons and the initiation of the vocal alert from the external speaker was measured at less than 200 milliseconds. This near-instantaneous localized response is a direct benefit of offloading the audio processing to the dedicated microSD module rather than relying on a cloud-based text-to-speech engine. Users receive immediate acoustic confirmation of their action, which reinforces confidence in the system. Furthermore, because the audio files are stored completely offline, the vocal feedback mechanism proved to be 100% immune to internet outages. Even when the local Wi-Fi router was deliberately disabled during testing, the Arduino UNO successfully identified button presses and drove the external speaker without interruption.

On the networking side, the Node MCU ESP8266 effectively handled the asynchronous task of notifying the remote caregiver. Serial communication between the Arduino UNO and the Node MCU remained stable throughout the testing phase, with no dropped data strings observed at standard baud rates. Under normal residential Wi-Fi conditions, the transit time for the text payload—measured from the moment the Arduino triggered the serial command to the instant the notification appeared on the caregiver's mobile screen—averaged between 1.5 and 3 seconds. While slightly slower than the local audio feedback, this slight network delay is entirely acceptable for paging applications and does not hinder the primary user's experience.

Hardware stability also proved robust during continuous operation. A common issue with mechanical switches is contact bounce, which can trick a microcontroller into registering a single press as multiple rapid inputs. By implementing strict software-based debouncing logic on the Arduino's digital pins, the system completely filtered out these false triggers. Out of 500 intentional button presses across various testing sessions, the system correctly mapped and executed the corresponding command every single time without duplicate alerts. Additionally, the shared 5V USB power delivery system effectively sustained the peak current draw when both the Wi-Fi radio was transmitting and the speaker was playing at maximum volume, preventing any brownout resets.

The findings highlight a clear advantage of physically dividing the workload between two standard development boards. By isolating the critical, time-sensitive tasks (input scanning and audio playback) on the Arduino UNO, and delegating the heavier, unpredictable network tasks to the Node MCU, the architecture entirely avoids the processing bottlenecks that often plague single-board IoT projects.

Ultimately, this prototype demonstrates that highly effective assistive technologies do not need to rely on expensive, proprietary medical hardware. The combination of simple tactile switches, offline voice generation, and budget-friendly Wi-Fi paging creates a highly accessible tool. For the end-user, it restores a degree of independence through immediate vocal expression. For the caregiver, the remote mobile notifications significantly reduce the anxiety of needing to be in the exact same room at all times. Future iterations of this project could easily expand upon this foundation by migrating the breadboard circuit to a custom printed circuit board (PCB), integrating a rechargeable lithium-ion battery management system for complete portability, and exploring cellular modules for outdoor use beyond local Wi-Fi networks.

8 Challenges and limitations

Building and evaluating this assistive communication prototype highlighted several engineering hurdles and design trade-offs that are critical to address in future revisions:

- **Breadboard Interconnect Fragility:** Utilizing jumper wires and solderless breadboards is ideal for rapid benchtop prototyping, but it introduces physical vulnerability. Mechanical stress or accidental pulls on the loose wiring can loosen pin contacts, causing intermittent power cuts or missed button readings.
- **Wi-Fi Coverage and Network Dependency:** The remote paging feature relies strictly on the 2.4 GHz local Wi-Fi coverage provided to the Node MCU ESP8266. If the device is operated in network dead zones, or if the home router loses internet access, the caregiver notification drops out until connectivity is re-established (even though offline audio continues to work locally).
- **Serial Communication Synchronization:** Splitting tasks between the Arduino UNO and the Node MCU over TX/RX hardware lines requires careful baud rate matching and buffer management. Rapid, consecutive button clicks risk overflowing the serial receiver or dropping notification payloads if one microcontroller is busy handling another task.
- **Static Vocabulary Constraints:** Because the audio tracks are pre-recorded files stored on a microSD card, the user's communication set is confined to fixed phrases. Changing, updating, or expanding the voice library requires physically removing the memory card to upload new files, rather than updating phrasing dynamically on the fly.
- **Fixed Input Flexibility:** Five tactile push buttons limit the direct vocabulary to five primary actions or requests without implementing complex shift-modes or multi-button combinations. For users needing expansive, open-ended conversational language, a purely button-driven interface can become restrictive compared to full tablet-based AAC software.
- **Tethered Power Supply:** Operating via dual 5V DC USB power cables limits mobility. The system currently requires proximity to a stationary wall adapter or an external power bank, highlighting the need for an integrated rechargeable battery and voltage regulation circuit to achieve true handheld portability.
- **Acoustic Output and Amplification Limits:** Driving the metallic external speaker directly from the basic microSD audio module without an intermediate audio amplifier restricts the maximum safe volume output. In louder residential or clinical environments, the localized voice alert might be difficult to hear, suggesting that future iterations will need a dedicated amplifier circuit to ensure the sound cuts through background noise.
- **Absence of Local Visual Confirmation:** The current hardware assembly lacks an integrated display, such as an OLED screen or simple status LEDs, for the primary user. While the remote

caregiver receives a detailed visual text notification on their smartphone, the user must rely entirely on the auditory playback to know their request was processed. This lack of visual feedback could be a significant barrier for individuals who have concurrent hearing impairments.

9 Conclusion

The development of this IoT-enabled communication aid successfully demonstrates that effective augmentative technology does not need to rely on expensive, proprietary medical platforms. By strategically splitting the computational workload between an Arduino UNO and a Node MCU ESP8266, the project achieved a highly responsive, dual-action feedback loop without sacrificing hardware stability. The five tactile mechanical switches provided users with a dependable, physical interface, entirely bypassing the steep learning curves and fine-motor requirements associated with modern touchscreen smart devices. Furthermore, localized auditory feedback—driven by an independent microSD module and external speaker—ensured that users received instant vocal confirmation of their requests. Because this sound generation was handled entirely offline, the system preserved the user's communication autonomy even during deliberate internet outages.

Simultaneously, the integration of lightweight wireless protocols through the ESP8266 Wi-Fi gateway effectively bridged the physical gap between the patient and the caregiver. Delivering remote, real-time smartphone notifications successfully reduced the necessity for constant in-room monitoring, thereby lowering caregiver fatigue while maintaining immediate response times. Ultimately, this breadboard prototype validates a highly practical, budget-friendly blueprint for assistive communication, proving that immediate mechanical responsiveness can be seamlessly paired with modern smart-home connectivity.

10 Future work

Building upon the foundational success of this prototype, several distinct avenues for structural and functional enhancement are apparent. The most immediate engineering priority for future iterations is the migration of the loose jumper-wire circuit onto a dedicated custom Printed Circuit Board (PCB). Designing a unified PCB will drastically shrink the hardware's physical footprint, allowing the dual-controller setup to be housed inside a durable, 3D-printed enclosure that can be easily mounted to a wheelchair armrest or hospital bed. To achieve true handheld mobility, the tethered USB power cords must be replaced with an integrated rechargeable lithium-ion battery pack, complete with a dedicated power management and charging circuit.

Expanding the physical user interface is another critical area for development. While the current five-button matrix is highly functional, incorporating interchangeable, oversized arcade-style switches or highly sensitive capacitive touch pads would better accommodate individuals suffering from severe muscle spasticity or advanced tremors. Additionally, integrating a localized visual feedback mechanism, such as a compact OLED display on the main enclosure, would greatly assist users who experience concurrent hearing loss. To address the current acoustic limitations in noisy residential or clinical environments, integrating a dedicated digital audio amplifier (such as a compact Class-D amplifier chip) will be necessary to safely boost the external speaker's volume without distorting the pre-recorded voice files.

Finally, future software upgrades should focus on expanding network versatility and dynamic communication. Upgrading the wireless infrastructure from a local Wi-Fi module to a cellular GSM or LTE

transceiver would completely untether the device from residential routers, allowing users to page their caregivers from anywhere, including during outdoor transit. Moreover, while the static offline microSD audio provides unmatched reliability, developing a hybrid software architecture that occasionally utilizes cloud-based Text-to-Speech (TTS) APIs could allow caregivers to push dynamic, on-the-fly text updates to the device that are subsequently read aloud. These targeted enhancements will steadily transform this initial proof-of-concept into a highly robust, medical-grade communication companion.

References

[1] Samsung Electronics America. *Samsung Solve for Tomorrow STEM competition: Assistive biometric wearable prototypes (Bentonville High School)* [Internet]. 2022–2025. Available from: https://www.samsung.com/us/solvefortomorrow/

[2] Empatica Inc. *EmbracePlus: Continuous physiological and electrodermal activity (EDA) monitoring platform* [Internet]. 2025. Available from: https://www.empatica.com/embraceplus/

[3] Light J, McNaughton D. Communicative competence for individuals who require augmentative and alternative communication: A new definition for a new era of communication? *Augmentative and Alternative Communication*. 2014;30(1):1–18. https://doi.org/10.3109/07434618.2014.885080

[4] Iling S, et al. Possibilities of intelligent textiles in AAC – perspectives of speech and language therapists. *Disability and Rehabilitation: Assistive Technology*. 2022;18:1–13. https://doi.org/10.1080/17483107.2022.2141900

[5] AssistiveWare B.V. *Proloquo2Go: Symbol-supported AAC application for iOS* [Internet]. 2026. Available from: https://www.assistiveware.com/products/proloquo2go

[6] WillowTree Apps. *Vocable AAC: Empowering people to communicate via head tracking* [Internet]. GitHub; 2026. Available from: https://github.com/willowtreeapps/vocable-ios and https://github.com/willowtreeapps/vocable-android

[7] PRC-Saltillo. *TouchChat AAC communication application* [Internet]. 2025. Available from: https://prc-saltillo.com/apps/touchchat

[8] AbleNet Inc. *BIGmack speech generating device technical specifications* [Internet]. 2025. Available from: https://www.ablenetinc.com/bigmack/

[9] Paralyzed patient monitoring system utilizing ESP32 and computer vision. *International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)*. 2025;Paper 30135. Available from: https://ijarsct.co.in/Paper30135.pdf

[10] Reddy BS, et al. IoT based automated paralysis patient monitoring system. In: *Innovations in computer vision, communication systems, and computational intelligence*. 2024. Available from: https://www.researchgate.net/publication/384896728_IOT_Based_Automated_Paralysis_Patient_Monitoring_System

[11] Awake Labs. *Wearable-enabled emotion and stress monitoring for cognitive disabilities* [Internet]. 2025. Available from: https://www.awakelabs.com/learn-about-our-technology