International Journal of Research in Engineering, Science and Management
Volume 8, Issue 5, May 2025
https://www.ijresm.com | ISSN (Online): 2581-5792

Hand Gesture Recognition for Voice-Free Communication Using ESP32

J. Daisy Panimariyal[1*], B. S. Shrinidhi[2], R. Niranjana[3], G. Deephiksha[4], Madhuvanthini Rajendran[5], K. Kalaiselvi[6], G. Sundar[7]

[1,2,3,4]*Student, Department of Electrical and Electronics Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, India*
[5]*Assistant Professor, Department of Electrical and Electronics Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, India*
[6]*Associate Professor, Department of Electrical and Electronics Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, India*
[7]*Professor, Department of Electrical and Electronics Engineering, Sri Shakthi Institute of Engineering and Technology, Coimbatore, India*

***Abstract*: A "Hand Gesture recognition for voice-free Communication using ESP32" introduces an intelligent communication aid that converts physical gesture into audible speech, enhancing interaction for speech disturbed individuals. Key components are flex sensors, DF- mini player, speaker, Esp32 microcontroller and a SD card, they work together to detect, process and translate gestures into voice. The flex sensors track finger movements. The microcontroller processes these signals, mapping them to pre- defined speech outputs. It provides an inclusive, user-friendly solution to support non-verbal communication in everyday situations.**

***Keywords*: ESP32 microcontroller, hand gesture, flex sensors, speech system.**

1. Introduction

Communication plays a vital role in daily human interactions. However, individuals with speech and hearing disabilities often face difficulties in conveying their thoughts and emotions to others. While sign language serves as a primary means of communication for them, it requires that both parties understand the language, which is not always practical.

To address this challenge, technological innovations are being developed to convert hand gestures into audible speech. Such systems aim to bridge the communication gap and provide a more inclusive environment for people with disabilities.

In this project, we propose a Hand Gesture to Speech Conversion System using the ESP32 microcontroller. The ESP32 is chosen for its powerful processing capabilities, built-in Wi-Fi and Bluetooth, low power consumption, and real-time data handling, making it ideal for portable assistive devices.

The system works by capturing hand movements through sensors like flex sensors and accelerometers. These sensor readings are processed to recognize specific gesture patterns. Upon successful recognition, the corresponding speech output is generated using a speaker, allowing the user's gestures to be heard clearly and instantly.

This paper discusses the system design, hardware integration, software development, performance evaluation of the proposed model.

2. Literature Review

Various reviews had been held on this project "Hand Gesture Recognition for voice- free communication using ESP32" and notable ones are explained below:

Smart Glove for Hand Gesture Recognition – by Kumar et al., The authors proposed a smart glove system embedded with flex sensors to detect hand gestures. The main objective was to recognize gestures in real-time and convert them into speech using microcontrollers, enhancing communication for speech-impaired users.

"Gesture Controlled Voice Output System for Dumb and Deaf" – by Sharma and Mehta. This work focused on using accelerometer sensors to identify hand movements. The system aimed to map predefined gestures to voice outputs, allowing users with speech disabilities to communicate effectively through synthesized speech.

"Real-Time Hand Gesture Recognition Using Machine Learning" – by Rani and Patel. The study utilized machine learning techniques to classify hand gestures based on sensor data. The objective was to improve the accuracy of gesture recognition and ensure faster processing for real-time speech conversion applications.

"Assistive Communication Device Using ESP32" – by Das and Roy. This paper explored the use of the ESP32 microcontroller for wireless gesture-to-speech translation. The primary goal was to design a portable, low-cost device capable of accurately translating physical gestures into audio messages.

"IoT-Based Sign Language Interpreter" – by Bhatia and Singh. The authors introduced an IoT- based system integrating cloud connectivity with gesture recognition. The aim was to enable remote communication and data storage while converting hand signs into voice for improved accessibility.

"Real-Time Hand Gesture to Speech Conversion System" – by Patel and Singh. The paper describes a real-time gesture recognition system integrated with a speech synthesis module.

*Corresponding author: daisypotes93@gmail.com

The core aim was to create an accurate and responsive system using accelerometer data and machine learning techniques.

3. Circuit Components

A. ESP32

Fig. 1.

The ESP32 is a powerful and versatile low-cost microcontroller developed by Espressif Systems. It features a dual-core Tensilica LX6 processor, operating at up to 240 MHz, with integrated Wi-Fi and Bluetooth capabilities. This makes it ideal for a wide range of IoT and embedded applications requiring wireless communication and real-time processing.

The ESP32 includes a variety of built-in peripherals such as ADCs, DACs, UARTs, SPI, I2C, PWM, capacitive touch sensors, and multiple GPIOs. Its high performance, low power consumption, and rich feature set allow for efficient sensor interfacing, data processing, and communication in real- time systems.

B. Flex Sensor

Fig. 2.

A Flex sensor is a thin, bendable sensor that measures the amount of bending or flexing. It is made of a conductive material whose resistance changes based on the amount of bend: the more the sensor bends, the higher its resistance. This change in resistance can be easily measured using a microcontroller to determine the angle of flex.

Flex sensors are lightweight, flexible, and easy to integrate into wearable applications. They are commonly used in glove-based gesture recognition systems, where the bending of fingers causes measurable resistance changes, helping detect specific hand gestures.

In hand gesture to speech conversion systems, flex sensors are attached along the fingers to capture the degree of finger movements accurately. The sensor outputs are processed to recognize different hand shapes or signs, which are then converted into corresponding speech outputs.

C. DF-Mini Player

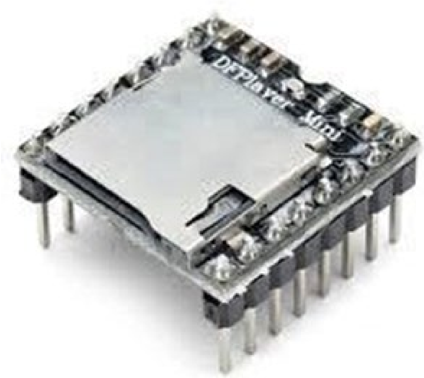

Fig. 3.

The DFPlayer Mini is a small, low-cost MP3 module that can play audio files directly from a microSD card. It is a standalone module with a built-in amplifier, capable of driving small speakers without the need for additional external circuits. The module can be controlled through serial communication (UART) with microcontrollers like the ESP32.

The DFPlayer Mini supports MP3 and WAV audio formats and offers simple commands to play, pause, stop, and control volume. It operates on a 3.2V to 5V supply, making it highly suitable for embedded and portable applications.

In a hand gesture to speech conversion system, the DFPlayer Mini plays a critical role by storing pre-recorded voice outputs corresponding to specific gestures. When the ESP32 identifies a gesture, it sends a command to the DFPlayer Mini to play the related audio file, allowing users to hear the converted speech in real time.

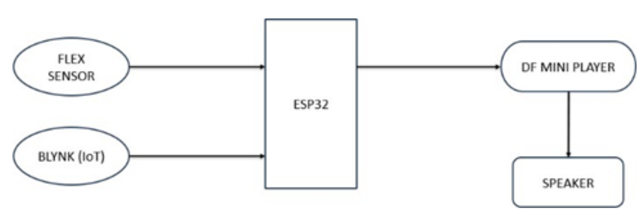

Fig. 4. Block diagram

The block diagram illustrates the detailed workflow of the Hand Gesture to Speech Conversion System based on the ESP32 microcontroller. It highlights the interaction between different hardware modules for real- time gesture recognition and speech output generation.

The first block is the Flex Sensor, which is mounted on the user's fingers. The flex sensor acts as an input device that detects the degree of bending when the user performs a hand gesture. As the fingers bend, the sensor's resistance changes proportionally, which is critical for understanding the user's intended movement.

The analog signals produced by the flex sensors are sent directly to the ESP32 microcontroller. The ESP32 plays the role of a central processing unit that collects these signals, digitizes them using its internal ADC (Analog to Digital Converter), and processes the data to recognize specific gesture patterns based on predefined thresholds or algorithms.

The flex sensor detects the bending of a finger or hand movement. As it bends, the resistance changes, producing a varying voltage signal. This signal is sent to the ESP32, which interprets the degree of bending to recognize specific gestures.

The Blynk app serves as a wireless interface, allowing users to send commands to the ESP32 over the internet. It provides control options through a smartphone, making the system remotely accessible and customizable in real-time.

The ESP32 acts as the central processing unit of the system. It receives input from both the flex sensor and the Blynk app. Based on these inputs, it makes decisions and sends commands to the DF Mini Player to play specific audio files.

The DF Mini Player is an audio playback module that stores and plays MP3 files from an SD card. When it receives a command from the ESP32, it selects and plays the corresponding audio track.

The speaker is connected to the DF Mini Player and outputs the sound. As the DF Mini Player plays the audio, the speaker delivers it audibly to the user, completing the process of gesture-controlled or remotely triggered audio output.

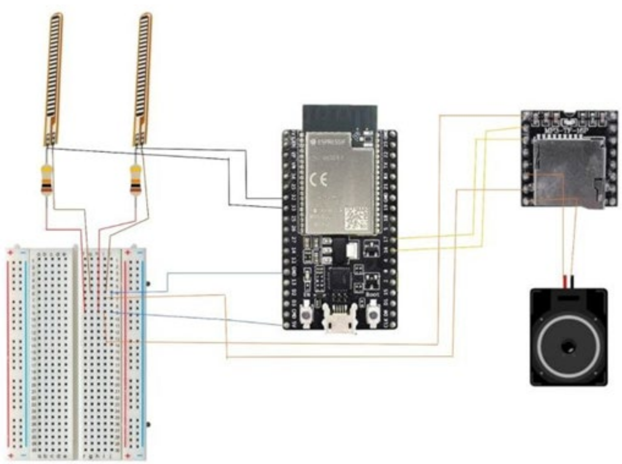

Fig. 5. Circuit diagram

4. Conceptual Diagram and Work

The circuit begins with two flex sensors connected to a breadboard. Each flex sensor has one leg connected to the 3.3V power rail, and the other leg connected in series with a resistor. This forms a voltage divider that helps measure the varying resistance when the sensor bends. The other side of the resistor is connected to ground, and the middle point between the flex sensor and the resistor is connected to an analog input pin on the ESP32.

The ESP32 board is placed centrally in the setup. It receives analog signals from the flex sensors. These signals indicate how much each sensor is bent. The ESP32 reads these inputs and processes them to detect specific gestures or movements based on the voltage levels coming from the voltage dividers. Audio playback depending on the detected gestures or commands.

Flow:

This module is wired to the ESP32 using serial communication lines—TX and RX. The module also shares a common ground with the ESP32, and is powered using the 5V supply line. The ESP32 sends commands to this module to control audio playback depending on the detected gestures or commands.

A speaker is connected to the DF Mini Player. It receives the audio output and plays the corresponding sound stored in the DF Player's SD card. The player manages playback directly and sends the audio signal to the speaker.

The process starts when a user bends one or both of the flex sensors. As the sensor bends, its resistance changes, altering the voltage at the point where it's connected with the resistor. This change in voltage is sent to the analog input pins of the ESP32, allowing the microcontroller to detect how much the sensor is bent.

The ESP32 constantly reads these analog voltage values from the flex sensors.

Based on pre-defined threshold values or patterns, it determines whether a specific gesture or input condition has been met. This decision logic is programmed into the ESP32 through its firmware.

Once the ESP32 recognizes a valid gesture or receives an input via the Blynk app (if included in the system), it sends a serial command to the DF Mini Player.

This command tells the DF Player which specific audio file to play from its microSD card.

The DF Mini Player receives the command and starts playing the corresponding MP3 file. It sends the audio signal directly to the speaker connected to it. The speaker then produces the sound, which corresponds to the detected gesture or user action.

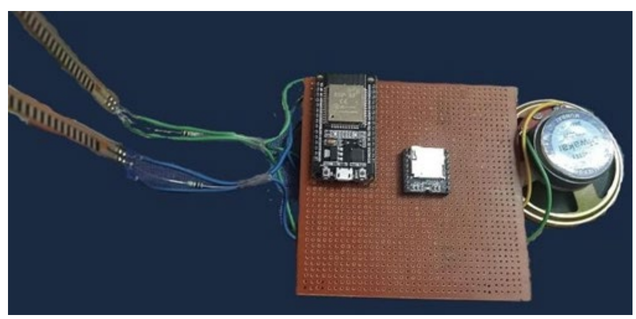

Fig. 6. Hardware setup

5. Conclusion

The monitoring system for the solar panel project has proven effective in optimizing energy generation. By automatically adjusting the panels to track sunlight and further rotating them during rain, the system ensures that the panels operate at peak efficiency throughout different weather conditions. The implementation of such dynamic adjustments enhances the energy capture during periods of low sunlight and minimizes the risk of damage during adverse weather conditions.

Data from the monitoring system has shown increased energy yield, particularly in areas with frequent cloud cover or rainfall. Additionally, the system's adaptability contributes to a more reliable and resilient solar energy infrastructure, reducing maintenance needs and extending the lifespan of the panels.

Overall, the integration of weather-responsive technology in solar panel systems holds significant promise for improving the efficiency and sustainability of solar energy, making it a valuable advancement for both residential and commercial solar energy solutions.

This project successfully demonstrates a simple yet effective gesture-controlled audio system using flex sensors, an ESP32 microcontroller, a DF Mini Player, and a speaker. The flex sensors serve as intuitive input devices that detect finger or hand movements, which are then translated into electrical signals. These signals are read by the ESP32, allowing it to interpret specific gestures in real time.

The ESP32 plays a central role by processing the sensor inputs and sending appropriate commands to the DF Mini Player. This integration enables dynamic control over audio playback without any mechanical switches or buttons, making the system more user-friendly and responsive. Additionally, the DF Mini Player efficiently handles audio playback and outputs the sound through a connected speaker, providing immediate feedback based on the user's gestures.

The use of the Blynk IoT platform adds remote control capability, enabling the system to be monitored or activated via a smartphone. This enhances the versatility of the setup, making it suitable for applications where both local gesture control and remote interaction are desired.

Overall, the circuit offers a compact, low- power, and low-cost solution for implementing gesture-based audio interaction. It demonstrates the potential of combining simple sensor technologies with powerful microcontrollers and IoT platforms to create interactive systems.

This approach can be extended to applications such as assistive communication tools for differently-abled individuals, interactive teaching aids, or smart home control systems.

References

[1] Ranjani Aruna A., Jeyalakshmi V., Srivarshini R., Shibi Sharan K., Tharun S., Sethu Madhavan S. (2023). "Flex Sensor-Based Hand Gesture Detector." *Journal of Embedded Systems and Processing*, vol. 8, no. 2.

[2] Patel Kathan, Ronak Makwana, Dhruvil Bhavsar, Sandeep Chauhan, Shriji Gandhi (2023). "IoT Based Hand Gesture Control Robot." *International Journal of Engineering Research & Technology*, vol. 12, no. 9.

[3] Joel Richard G., Shurvesh S. S., Sree Sankar J. (2023). "IoT Based Hand Gesture Recognition System." *International Research Journal on Advanced Science Hub*, vol. 5, no. 5.

[4] C. V. Suresh Babu, Rithika Purushothaman, K. Anusha, Shri Sakthi (2023). "Smart Gesture Controlled Systems Using IoT." In *AI-Based Digital Health Communication for Securing Assistive Systems*, pp. 114–133.

[5] Santosh Kumar J., Vamsi, Vinod, Madhusudhan, Tejas (2021). "Design and Development of IoT Device that Recognizes Hand Gestures using Sensors." *International Journal for Modern Trends in Science and Technology*, vol. 7, pp. 29–34.