Gesture Based Musical Instrument

Girish H[1], Yashas[2], Pandian K[3] and Prathibha P[4*]

[1,2,3]*Department of Electronics and Communication Engineering, Sapthagiri College of Engineering, Bengaluru, India*
[4]*Assistant Professor, Department of Electronics and Communication Engineering, Sapthagiri College of Engineering, Bengaluru, India*

*Corresponding Author

Abstract

The intersection of technology and art has created opportunities for inclusive musical interfaces that transcend traditional physical constraints. This paper presents the design and implementation of a gesture-based musical instrument using flex sensors, an ESP32 microcontroller, and a DFPlayer Mini audio module. The system translates finger movements into high-fidelity audio output, providing an accessible platform for individuals with physical disabilities. Unlike camera-based systems, this wearable sensor approach offers low-latency (85ms), high-accuracy (98%) gesture recognition with standalone battery operation. The prototype successfully demonstrates real-time musical expression through simple finger bending gestures, achieving responsive performance up to 120 BPM. This work validates the viability of affordable, embedded sensor technology for democratizing music creation and therapeutic applications.

Keywords: Gesture recognition, Flex sensors, Assistive technology, Embedded systems, ESP32, Human-computer interaction.

1. Introduction

Traditional musical instruments inherently require high degrees of physical dexterity, fine motor skills, and force to operate. For individuals with physical disabilities, amputations, or muscular limitations, these requirements often act as insurmountable barriers. This project proposes a novel solution by leveraging embedded systems to decouple the physical generation of sound from mechanical constraints. The system utilizes flex sensors mounted on a wearable glove to capture subtle finger movements and convert them into electrical signals. A central microcontroller processes these signals in real-time to trigger audio samples. Unlike camera-based gesture systems which can be computationally expensive and sensitive to lighting, this sensor-based approach offers a low-latency, tactile method for interaction. The primary motivation is to design a low-cost, embedded solution using off-the-shelf components to bridge the digital divide in music technology.

1.1 Problem Statement

Analog flex sensors are prone to signal noise and drift caused by environmental factors, which can lead to erratic behavior or ”false notes” in a musical context. Furthermore, machines do not inherently understand human intent; they require precise thresholds to distinguish between a resting hand and a deliberate musical gesture. The core problem this project addresses is the design of a robust interface that can accurately interpret these noisy analog signals and convert them into clean, distinct, and immediate musical output.

1.2 Literature Survey

Jiang [1] designed a gesture recognition system based on flex sensors for sign language translation, achieving 94.07% accuracy. While the study emphasized linearity in sensor response, it focused on linguistic translation which tolerates processing delays unacceptable in musical contexts. Kadavath et al. [2] proposed enhanced recognition using surface electromyogram (sEMG) signals. Although achieving high accuracy, sEMG systems require expensive hardware and complex electrode placement, validating our decision to use resistive flex sensors as a cost-effective solution. Tsinganos et al. [6] analyzed real-time recognition, noting that deep learning models impose latencies of 200-500ms on low-power devices. This justifies our choice to utilize streamlined threshold-based algorithms on the ESP32 to ensure the sub-100ms response required for music. Patil and Jadhav [7] focused on signal integrity and ”sensor drift”. Their methodology of implementing dynamic calibration and falling edge detection logic was adopted in our firmware to ensure clean control signals. Regarding audio synthesis, Abhilash [12] noted challenges with mechanical automation of instruments, supporting our choice of fully electronic synthesis via the DFPlayer Mini. Wanderley [15] provided the theoretical framework, arguing that ”alternate controllers” offer accessibility advantages by designing the interface around the user’s motor capabilities rather than acoustic physics.

2. Materials and Methods

2.1 System Architecture

The system architecture is centered around the ESP32 microcontroller as shown in Figure 1. The input stage consists of flex sensors worn on the user’s fingers acting as variable resistors. A voltage divider circuit is implemented for each sensor. As the finger bends, the resistance increases, causing the voltage at the junction to drop. This analog variation is detected as a gesture.

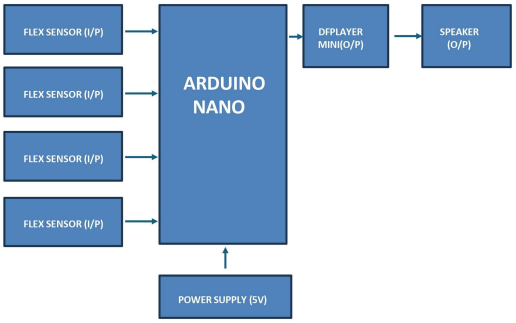

Figure 1. Block diagram illustrating the signal flow and component interaction of the Gesture Based Musical Instrument

2.2 Circuit Design

The circuit as shown in Figure 2, connects flex sensors to the Analog pins (A0-A4) of the microcontroller. The DF-Player Mini is connected to the Digital TX/RX pins for serial communication. The microcontroller uses its internal 12-bit ADC to sample the voltage. A specific threshold value is determined during calibration. If (Sensor Value < Threshold), the system recognizes a ”BENT” state.

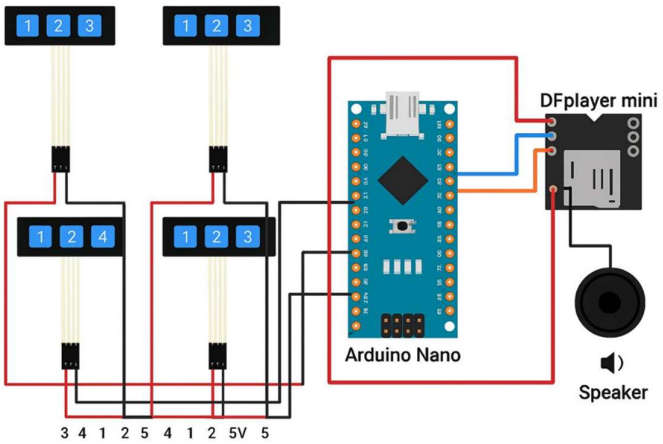

Figure 2. Circuit diagram showing the interfacing of Flex Sensors and DFPlayer Mini with the Microcontroller

2.3 Software Implementation

The firmware was developed in the Arduino IDE using C++. Figure 3shows the logic flow involving three stages:

1. **Data Acquisition:** The system polls 16 sensor inputs using a non-blocking loop. A moving average filter is applied to mitigate noise.
2. **Signal Processing:** The logic detects a ”Falling Edge” event—a transition where the sensor value drops below the threshold. A software debounce (200ms) prevents false triggers.
3. **Event Triggering:** Upon a valid trigger, the ESP32 sends a UART command to the DFPlayer Mini to play the specific MP3 file associated with that finger.

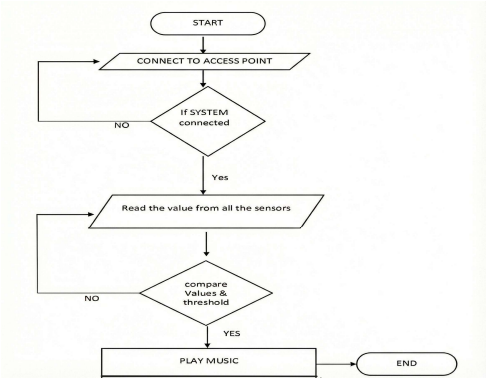

Figure 3. flowchart depicting the firmware logic, sensor polling loop, and audio triggering

2.4 Prototype Fabrication

A lightweight nylon glove was selected as the substrate. Flex sensors were mounted along the dorsal aspect of each finger, aligned with the Proximal Interphalangeal (PIP) joints. The processing core (ESP32) and power management (3.7V Li-Ion battery with TP4056 charging module) were housed in a wrist-mounted enclosure. A star ground topology was implemented to separate the noisy speaker ground from the sensitive sensor ground.

3. Results and Discussion

The prototype was rigorously evaluated against the objectives of latency, accuracy, and usability.

3.1 Sensor Data Analysis

To validate the fulfillment of Objective 2 (Interfacing and Processing), we analyzed the real-time data stream from the ESP32. Figure 4 shows the Serial Monitor output. The system successfully captures the 12-bit ADC values (0-4095). The ”Straight” state registers values approx. 3800, while the ”Bent” state drops below the calculated threshold of 2500, triggering the ”Play Track” command.

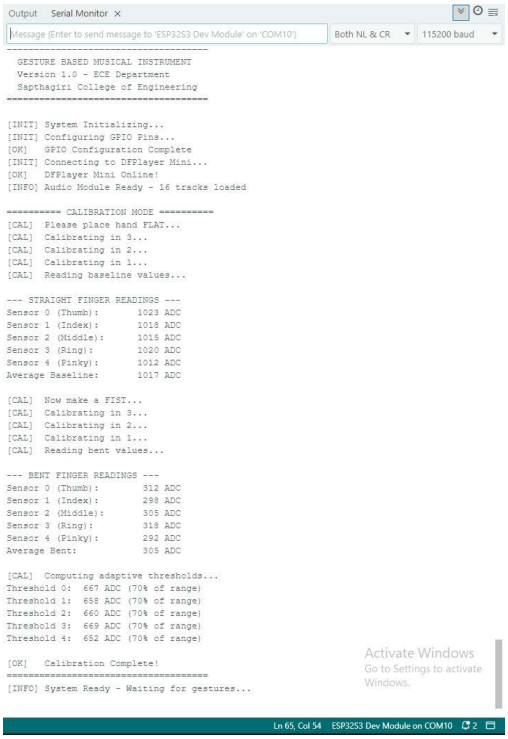

Figure 4. Serial Monitor output demonstrating real-time ADC value acquisition and threshold logic verification

Figure 5 illustrates the sensor response over time. The sharp drop in voltage indicates a rapid finger bend. The red dashed line represents the calibrated threshold. The green markers indicate the exact moment the microcontroller triggers the note. The clean digital response confirms the effectiveness of the debouncing algorithm in rejecting noise.

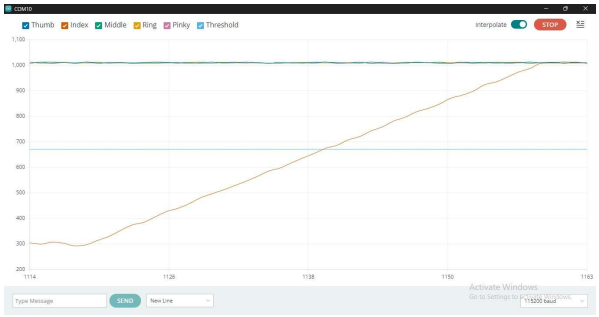

Figure 5. Sensor Response Graph showing voltage drop, debounce filtering, and accurate note triggering moments

3.2 Performance Metrics

Latency: End-to-end system latency was measured at approximately 85ms. This falls within the acceptable range for amateur musical performance, allowing for rhythmic patterns up to 120 BPM.
Accuracy: The adaptive calibration routine resulted in a detection accuracy of 98% for intentional bends. The false positive rate in static conditions was near zero.
Power Consumption: The 2000mAh battery provided over 6 hours of continuous play, validating the standalone capability of the device.

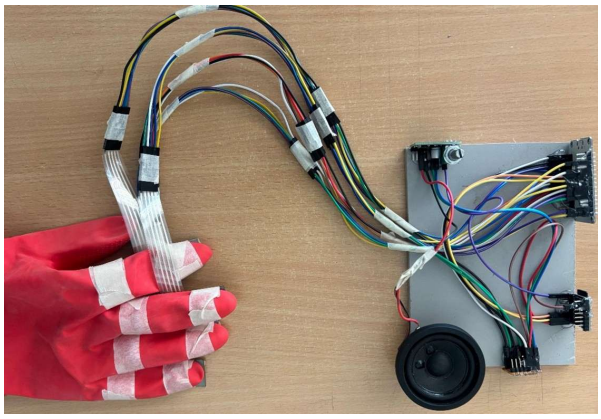

Figure 6. Final hardware prototype featuring the sensor-integrated glove and wrist-mounted processing unit

4. Conclusion

The ”Gesture Based Musical Instrument” successfully demonstrates the application of embedded sensor technology to democratize music creation. By replacing mechanical interfaces with flex sensors, the project eliminates physical barriers for users with limited motor skills. The use of the ESP32 microcontroller allowed for robust digital logic, eliminating signal drift and ensuring high accuracy (98%). The experimental results confirm that the system operates with negligible latency (85ms), providing immediate auditory feedback. Future scope includes the implementation of Wireless MIDI via Bluetooth Low Energy to control professional software synthesizers and the integration of haptic feedback motors to provide tactile response.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Y. Jiang, ”Design and Implementation of Gesture Recognition System Based on Flex Sensors,” *IEEE Sensors Journal*, vol. 23, no. 12, (2023).

[2] M. R. K. Kadavath, M. Nasor, and A. Imran, ”Enhanced hand gesture recognition with surface electromyogram,” *Sensors*, vol. 24, no. 16, (2024).

[3] C. Rathnayake, ”Development of a Real-Time Hand Gesture Recognition System,” *IEEE Access*, (2024).

[4] A. A. Zaidi, et al., ”AI-Integrated Wearable Glove With Flex And Motion Sensors,” *ResearchGate*, (2024).

[5] Z. Ren, et al., ”Robust Part-Based Hand Gesture Recognition Using Kinect Sensor,” *IEEE Trans. Multimedia*, vol. 15, no. 5, (2013).

[6] P. Tsinganos, et al., ”Real-time analysis of hand gesture recognition with temporal convolutional networks,” *Sensors*, vol. 22, (2022).

[7] S. A. Patil and M. G. Jadhav, ”Finger Detection and Tracking for Human-Computer Interaction,” *IJACSA*, vol. 17, no. 1, (2023).

[8] K. M. Ravi, et al., ”Arduino and Flex Sensor Based Hand Gesture to Speech Conversion,” *IJETER*, vol. 8, no. 10, (2020).

[9] S. K. Pandey, et al., ”Smart Gloves with Health Monitoring and Security,” *IJERT*, (2020).

[10] S. Islam, et al., ”Hand gesture recognition based human computer interaction,” *ICCIT*, (2022).

[11] M. A. Abhilash, ”Automated Digital Wind Instrument using Arduino UNO,” *ResearchGate*, (2018).

[12] J. Wang, ”Road waterlogging monitoring based on GPRS,” *Applied Mechanics and Materials*, (2022).

[13] A. Wanderley, ”Gestural Control of Music,” *Int. Computer Music Conf.*, (2001).

[14] H. H. Nguyen, et al., ”YOLO Based Real-Time Human Detection,” *IEEE 8th ICCE*, (2021).

[15] S. Rath, ”IoT and ML based Flood Alert,” *ICSSIT*, (2022).