IOT Based Smart Air Mouse and Voice Home Automation

[1]A.S.Mane, [2]D.J.Daiv, [3]T.N.Mali, [4]A.A.Naik, [5]I.R.Rajmane

[1,2,3,4,5]Electrical Engineering, DKTEs YCP, Ichalkaranji, Kolhapur, Maharashtra, India

E-mail: [1]abhijitsamane@gmail.com, [2]daivdhanashri@gmail.com, [3]tejaswinimali07@gmail.com, [4]ankitanaik2302@gmail.com, [5]ishwarirajmane31@gmail.com

Abstract - **This project presents the design a development of an offline voice-controlled wireless device control system using the AI-Thinker VC-02 voice recognition module and ESP32microcontrollers. The main objective of the project is to control electrical devices through voice commands without the use of internet connectivity, ensuring fast response, reliability, and enhanced privacy. The system utilizes embedded technology and wireless communication to provide a compact and efficient automation solution. The proposed system consists of two ESP32 microcontrollers communicating through the ESP-NOW wireless protocol. The transmitter section includes the VC-02 offline voice recognition module interfaced with an ESP32. This module is capable of recognizing predefined voice commands and transmitting corresponding hexadecimal command signals through serial communication. The ESP32 receives these commands and sends them wirelessly to another ESP32 using ESP-NOW, a low-power peer-to-peer communication protocol that does not require a Wi-Fi router or internet connection. The receiver ESP32 processes the received command and controls connected output devices such as an LED and a 12V DC fan using 2N2222 transistor switching circuits. The use of transistors instead of relay modules ensures faster switching, compact design, and reduced power consumption. The system demonstrates efficient wireless communication and reliable device control based on voice commands. This project offers a cost-effective, secure, and scalable solution for smart automation systems. It can be applied in smart homes, industrial automation, and assistive technologies where hands-free device control is required. The system can be further expanded to include multiple devices, display units, and additional automation features, making it suitable for real-time and practical applications in modern embedded systems.**

Keywords: IoT, Smart Air Mouse, Voice Control, Home Automation, Motion Sensors, Hand Gesture Control, Smart Home, Wireless Communication, User-Friendly System, Energy Efficiency.

I. INTRODUCTION

In recent years, the demand for smart and contactless control systems has increased significantly due to rapid advancements in embedded electronics and wireless communication technologies. Voice-controlled systems are becoming important part of modern automation because they provide a convenient and efficient way to operate devices without physical interaction. Such systems are especially useful in smart homes, industrial environments, and assistive technologies for elderly or physically challenged individuals.

This project focuses on the design and development of a voice-controlled wireless device control system using Thinker VC-02 offline voice recognition module and ESP32 microcontrollers. The main aim of this project is to control electrical devices using voice commands without requiring internet connectivity. The system operates the AI completely offline, making it reliable, fast, and secure.

The proposed system consists of two ESP32 microcontrollers communicating with each other using the ESP-NOW wireless protocol. The first ESP32 is connected to the VC-02 voice recognition module, which recognizes predefined voice commands and sends \corresponding hexadecimal signals to the microcontroller through serial communication. These commands are then transmitted wirelessly to another ESP32 using ESP-NOW technology. The second ESP32 receives the command and controls output devices such as an LED and a 12V DC fan through transistor switching circuits. ESP-NOW is a low-power, peer-to-peer communication protocol that allows ESP32 devices to exchange data directly without the need for a Wi-Fi router. This makes the system efficient and suitable for real-time applications. The use of an offline voice recognition module ensures that the system does not depend on cloud services, resulting in faster response time and enhanced privacy. This project demonstrates an innovative approach to wireless automation by integrating voice recognition, microcontroller programming, and wireless communication. It provides a simple, cost-effective, and scalable solution for controlling devices remotely using voice commands. The system can be

further expanded to control multiple appliances and can be applied in smart home automation, industrial control systems, and assistive technologies.

II. RELEVANCE

This project presents the design and development of an offline voice-controlled wireless device control system using the AI-Thinker VC-02 voice recognition module and ESP32 microcontrollers. The main objective of the project is to control electrical devices through voice commands without the use of internet connectivity, ensuring fast response, reliability, and enhanced privacy. The system utilizes embedded technology and wireless communication to provide a compact and efficient automation solution. The proposed system consists of two ESP32 microcontrollers communicating through the ESP-NOW wireless protocol. The transmitter section includes the VC-02 offline voice recognition module interfaced with an ESP32. This module is capable of recognizing predefined voice commands and transmitting corresponding hexadecimal command signals through serial communication. The ESP32 receives these commands and sends them wirelessly to another ESP32 using ESP-NOW, a low-power peer-to-peer communication protocol that does not require a Wi-Fi router or internet connection. The receiver ESP32 processes the received command and controls connected output devices such as an LED and a 12V DC fan using 2N2222 transistor switching circuits. The use of transistors instead of relay modules ensures faster switching, compact design, and reduced power consumption. The system demonstrates efficient wireless communication and reliable device control based on voice commands. This project offers a cost-effective, secure, and scalable solution for smart automation systems. It can be applied in smart homes, industrial automation, and assistive technologies where hands-free device control is required. The system can be further expanded to include multiple devices, display units, and additional automation features, making it suitable for real-time and practical applications in modern embedded systems.

III. LITERATURE REVIEW

Recent advancements in embedded systems and Internet of Things (IoT) technology have enabled the development of intelligent wireless control systems using microcontrollers such as ESP32. Many researchers and developers have explored voice-controlled automation, wireless device communication, and gesture-based human–machine interfaces.

Voice recognition based control systems are widely used in smart home and industrial automation applications. Traditional systems rely on cloud-based speech processing, which requires internet connectivity and results in latency. To overcome this limitation, offline voice recognition modules

such as Ai-Thinker VC-02 have been introduced. These modules process predefined voice commands locally without requiring internet access, ensuring faster response time and higher reliability in industrial and domestic environments. Offline voice recognition systems are especially useful where continuous internet connectivity is not available or where privacy is a concern.ESP32 has become one of the most popular microcontrollers for IoT applications due to its inbuilt Wi-Fi and Bluetooth communication, high processing capability, allows power consumption. Researchers have widely used ESP32 in home automation, industrial monitoring, and wireless control systems. Its dual-core architecture allows simultaneous execution of communication and control tasks, making it suitable for realtime automation projects. Wireless communication between microcontrollers is another important research area. Traditional communication methods include Bluetooth, Wi-Fi, ZigBee, and RF modules. Recently, ESP-NOW protocol developed by Espressif has gained popularity because it allows direct peer-to-peer communication between ESP32 devices without using a router. ESP-NOW provides fast data transmission, low latency, and reduced memory usage compared to traditional Wi-Fi communication, making it ideal for embedded control systems. Gesture-based human interface systems such as air mouse technology are also gaining importance. These systems allow users to control devices using motion or simple hand gestures instead of traditional switches or remotes. Combining gesture control with voice recognition provides a more flexible and user-friendly interface for modern automation systems. Several experimental systems have demonstrated that local voice processing on ESP32-bas devices can achieve reliable command recognition when limited to predefined commands. Community projects show that offline voice recognition combined with ESP32 enables smart home control without relying on cloud services, improving privacy. However, researchers also highlight that performing full speech recognition directly on ESP32 is computationally intensive, so using dedicated offline voice modules improves performance and reliabile.

Reference [1] – M. R. Prayogi and T. Tommy (2025) Prayogi and Tommy designed and developed an air mouse using the ESP32 microcontroller in combination with the MPU6050 sensor. Their work focuses on translating hand gestures into precise cursor movements on a computer. The integration process of accelerometer and gyroscope data with Bluetooth communication is detailed thoroughly. The authors highlight the ESP32's efficiency, low cost, and portability for IoT-based human-computer interaction devices. Testing demonstrates smooth, low-latency performance during real-time gesture control. This study provides a foundational methodology for building ESP32-based air mouse systems.

Reference [2] – R. Krisna (2025) Krisna's project article presents a comprehensive guide to creating an air mouse with ESP32 and MPU6050. The system uses Bluetooth to communicate wirelessly with computers, ensuring portability. The tutorial covers both hardware interfacing and software implementation for motion detection. Real-time hand gesture tracking is emphasized, demonstrating the feasibility of gesture controlled devices. The article is useful for practical implementations and educational purposes. It supports the application of ESP32 and MPU6050 in wireless air mouse projects.

Reference [3] – N. Rithik (2025) Rithik's Instructables tutorial demonstrates the creation of an air mouse with additional media control features. The design uses the MPU6050 sensor for motion detection along with the ESP32-C3 microcontroller. Step-by-step assembly and programming instructions make the project accessible for hobbyists and researchers. The study highlights the importance of sensor calibration for accurate gesture recognition. Wireless communication via Bluetooth ensures convenience and usability. This reference provides a practical approach to combining air mouse and media control functionalities.

Reference [4] – T. Qiuhong, L. Shaobo, and J. Shaoquan (2017) Qiuhong, Shaobo, and Shaoquan discuss the fundamental design and production of air mouse devices. They focus on integrating inertial measurement units (IMUs) for gesture detection and motion tracking. The study addresses challenges in achieving accurate real-time cursor control using hand gestures. Ergonomic and usability considerations are also highlighted to improve user experience. Their framework lays the groundwork for modern ESP32 and MPU6050-based implementations. The paper is significant for understanding the basic principles of air mause design. Al-Fahoum and Abadir propose a modified Madgwick filter for improved quaternion-based orientation estimation using AHRS systems. The modifications enhance accuracy and reduce drift in IMU sensor readings. Such improvements are crucial for precise gesture tracking in air mouse devices. Their research demonstrates better stability in real-time motion control applications. This work highlights advanced filtering techniques to optimize sensor data for reliable performance. It provides a strong foundation for improving ESP32 and MPU6050-based air mouse system.

Reference [5] – A. S. Al-Fahoum and M. S. Abadir (2018) Al-Fahoum and Abadir propose a modified Madgwick filter for improved quaternion-based orientation estimation using AHRS systems. The modifications enhance accuracy and reduce drift in IMU sensor readings. Such improvements are crucial for precise gesture tracking in air mouse devices. Their research demonstrates better stability in real-time motion control applications. This work highlights advanced filtering techniques to optimize sensor data for reliable performance. It provides a strong foundation for improving ESP32 and MPU6050-based air mouse system.

IV. THE PROPOSED WORK

Problem statement

The proposed work focuses on designing and developing an intelligent human-machine interaction system using the ESP32 microcontroller. The system combines gesture-based computer control and offline voice-controlled wireless automation into a single compact device. The project is divided into two major units: the Air Mouse Controller Unit and the Wireless Automation Control Unit.

In the Air Mouse Controller Unit, an ESP32 is interfaced with the MPU6050 motion sensor to detect hand movements. The accelerometer and gyroscope data are processed to control the computer cursor through Bluetooth communication. Push buttons are integrated to perform mouse operations such as left click, right click, copy, and paste. This enables wireless cursor control without the need for a traditional wired mouse. For appliance automation, an offline voice recognition module AI-Thinker VC-02 is connected to another ESP32 microcontroller. The VC-02 recognizes predefined voice commands without requiring an internet connection. When a command is detected, it sends a specific hex code to the ESP32 via serial communication. The ESP32 transmitter then sends the decoded command wirelessly using the ESP-NOW protocol to another ESP32 receiver unit. The receiver ESP32 processes the received data and controls output devices such as LEDs and DC fans through transistor driver circuits or relay modules. The rapid growth of digital systems has increased the need for intuitive and touch-free human-computer interaction devices. Conventional input devices such as wired mice and touchpads restrict user mobility and require flat surfaces for operation, making them inconvenient for presentations, smart classrooms, gaming, and assistive applications. Additionally, existing air mouse devices available in the market are often expensive and lack customization options. There is a need for a low-cost, portable, and efficient motion-controlled device that can provide seamless wireless communication with computers and smart devices. The challenge lies in accurately capturing hand gestures, processing motion data in real time, and transmitting precise cursor control signals without noticeable delay. Furthermore, ensuring stable Bluetooth connectivity, low power consumption, and user-friendly design adds to the complexity of development. Therefore, the problem is to design and develop a compact, cost-effective Smart Air Mouse using ESP32 and motion sensors that enables reliable gesture-

based cursor control, enhances user flexibility, and improves overall interaction experience while maintaining accuracy, responsiveness, and affordability.

Objective

The objective of the ESP32-based Air Mouse project is to develop a smart, motion-controlled pointing device that converts hand movements into accurate cursor movements using MPU6050 sensor data. The system will implement a real-time motion-to-cursor mapping algorithm that processes accelerometer and gyroscope data to ensure smooth, stable, and precise cursor control by reducing noise and drift. The project also aims to integrate left-click and right-click functionalities using physical buttons or gesture-based triggers to enhance user interaction. To provide seamless performance, the ESP32 will be configured in Bluetooth HID mode to ensure low-latency and reliable communication with minimal delay between motion and cursor response.

Additionally, the device will be designed as a compact, lightweight, and ergonomic handheld prototype suitable for long-duration use. The air mouse will be compatible with multiple operating systems including Windows, macOS, Linux, and Android devices supporting Bluetooth HID. Power consumption will be optimized through sleep modes and efficient programming to extend battery life for portable use. Furthermore, the system will include calibration and sensitivity adjustment features, allowing users to customize cursor responsiveness and maintain accuracy under different operating conditions.

V. METHODOLOGY

The methodology of this project involves the design, development, and implementation of a wireless voice-controlled device control system using the VC-02 offline voice recognition module and ESP32 microcontrollers. The system is developed by dividing it into two main sections: the transmitter section and the receiver section. A systematic approach is followed to ensure reliable communication and accurate device control. The first step in the methodology is the selection of suitable hardware components. The ESP32 microcontroller is chosen due to its built-in Wi-Fi capability, fast processing speed, and multiple communication interfaces. The AI-Thinker VC-02 offline voice recognition module is selected for its ability to recognize predefined voice commands without requiring internet connectivity. Additional components such as LEDs, a 12V DC fan, 2N2222 transistors, resistors, and diodes are used to implement and demonstrate device control. In the transmitter section, the VC-02 voice recognition module is interfaced with the ESP32 using UART serial communication. The module is programmed and trained to recognize specific voice commands such as turning the light

and fan ON or OFF. When a user speaks a trained command, the VC-02 processes the voice locally and sends a corresponding hexadecimal signal to the ESP32. The ESP32 continuously monitors the serial port and reads the received command data. After receiving the command from the VC-02 module, the transmitter ESP32 processes the data and sends it wirelessly to the receiver ESP32 using the ESP-NOW communication protocol. ESP-NOW is configured in peer-to-peer mode by registering the MAC address of the receiver ESP32. This allows direct and fast communication between the two ESP32 devices without using a router or internet connection. In the receive r section, another ESP32 is configured to operate in ESP-NOW receiver mode. It continuously listens for incoming data from the transmitter ESP32. When accommand is received, the ESP32 decodes the data and performs the required action based on the command. For demonstration purposes, an LED representing a light and a 12V DC fan are connected to the ESP32 through 2N2222 transistor switching circuits. The transistors act as electronic switches to control the devices safely and efficiently. The system is programmed using Arduino IDE, and the code is developed to ensure smooth communication between modules and proper device control. Serial monitoring is used during testing and debugging to verify command transmission and reception. The entire system is tested for different voice commands to evaluate response time, accuracy, and reliability. This methodology ensures a structured implementation of the project by integrating voice recognition, wireless communication, and embedded control techniques. The developed system demonstrates efficient and reliable operation, making it suitable for practical automation applications.
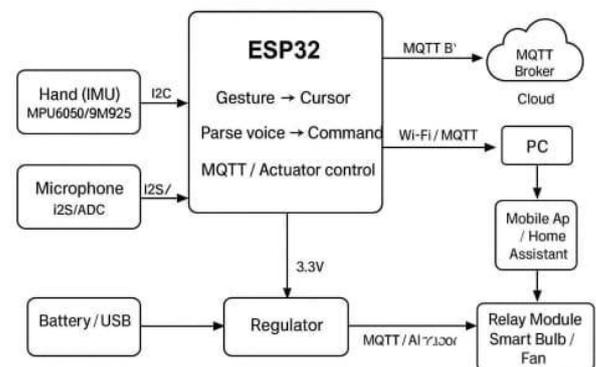

Figure 1: Block diagram of Smart Air Mouse and Voice Home Automation

1. ESP32 Development Boards (3 Units)

The ESP32 microcontroller is the main controller used in this project. It has built-in Wi-Fi and Bluetooth capabilities

which make it suitable for wireless communication and IoT applications.

One ESP32 is used for the air mouse system to control the computer cursor through Bluetooth.

The third ESP32 is used as a receiver to control output devices like LED and DC fan.

2. MPU6050 Gyroscope and Accelerometer Sensor

The MPU6050 is a motion tracking sensor used to detect hand movement. It contains a 3-axis accelerometer and 3-axis gyroscope which measure tilt, motion and rotation.

Function in project:

Detects hand motion
Converts movement into cursor movement
Communicates with ESP32 via I2C

3. AI-Thinker VC-02 Offline Voice Recognition Module

The VC-02 is an offline voice recognition module capable of recognizing predefined voice commands without internet connection.

Function in project:

Recognizes stored voice commands
Sends hex command data through UART serial
Controls appliances through ESP32

4. Push Buttons (4 Units)

Push buttons are used for mouse click operations.

Functions:

Left click
Right click
Copy command
Paste command

These buttons are connected to ESP32 GPIO pins using internal pull-up resistors.

5. 2N2222 NPN Transistor (2 Units)

The 2N2222 transistor is used as a switching device to control external loads.

Function:

Drives LED (light)
Drives 12V DC fan
Works as relay alternative

6. LED

LED is used to represent light control in the automation system.

Function:

Turns ON/OFF using voice command
Indicates output status

7. 12V DC Fan

A small DC fan is used as a demonstration load for appliance control.

Function:

Controlled using voice command
Operates through transistor driver

8. Resistors

Different resistors are used in the circuit.

Types:

220Ω for LED
1kΩ for transistor base
10kΩ pull-up resistors (if required)

9. Power Supply

The system requires proper power supply for operation.

Types:

5V supply for ESP32
3.3V internal for sensors
12V supply for DC fan
Rechargeable battery or adapter

10. Jumper Wires and Breadboard

Used for circuit connections and prototyping.

Function:

Easy wiring
Testing and debugging
Temporary connections

VI. FUTURE SCOPE

The proposed ESP32 based air mouse with offline voice-controlled wireless system can be further enhanced and expanded in many ways to improve functionality, performance and real-world usability In future, the system can be upgraded by adding rechargeable lithium battery and charging module,

making the air mouse completely portable and compact. A battery level indicator or OLED display can also be added to show system status and power level. The accuracy and smoothness of cursor movement can be improved by using advanced motion sensors such as MPU9250 or BNO055, which provide better orientation tracking and filtering. This will result in more stable and precise air mouse control. Additional gesture recognition features can be implemented using accelerometer and gyroscope data. Users can perform gestures such as shake, tilt or tap to execute commands like scroll, zoom, volume control or application switching. The voice control system can be expanded by training more commands in the VC-02 module to control multiple electrical appliances. The system can be integrated with relay modules or smart switches to control home or industrial devices such as lights, fans, motors and security systems A display unit (OLED or LCD) can be added to show received commands, connection status and device operation. This will improve user interaction and debugging. Future versions can include Wi-Fi or Bluetooth based mobile app control, allowing users to configure settings, sensitivity and commands from a smartphone. This will make the system more user-friendly and customizable. Security features such as password-protected activation or voice authentication can be implemented to prevent unauthorized access and improve safety. The project can be expanded into a complete smart home automation system where multiple ESP32 nodes communicate with each other to control various appliances using voice, gesture and mobile control. Integration with AI-based voice assistants and cloud platforms can also be explored to provide hybrid online-offline functionality while maintaining privacy and reliability. This system can also be adapted for industrial automation, presentation systems, gaming control and assistive technology for physically challenged individuals. With further development and miniaturization, the project has strong potential to become a commercial smart control device for modern automation applications.

VII. RESULT

The ESP32 based Air Mouse with offline voice controlled wireless automation system was successfully designed and implemented. The system performed all the intended operations efficiently and reliably. The air mouse section worked smoothly with the MPU6050 sensor. Hand movements were accurately detected and converted into cursor motion on the computer screen through Bluetooth communication. The cursor response was fast and stable with minimal delay. The push buttons connected to the ESP32 performed mouse operations such as left click, right click, copy and paste successfully.

The offline voice recognition system using the AI-Thinker VC-02 module demonstrated accurate recognition of pre-trained commands. When the user spoke commands like light ON, light OFF, fan ON and fan OFF, the module correctly generated the corresponding HEX values. These commands were received by the transmitter ESP32 through serial communication without any data loss. Wireless communication between the two ESP32 boards using ESP-NOW worked efficiently. The transmitter ESP32 successfully sent command data packets to the receiver ESP32. The receiver decoded the received command and controlled the output devices accordingly. The LED representing the light and the 12V DC fan operated correctly as per the received voice commands. The use of 2N2222 transistors as switching devices worked effectively for controlling the reliable. The entire system operated without the need for an internet connection, making it suitable for offline a low power application.

Table 1: Time schedule

Month	Work Schedule
Aug 25–Sept 10,2025	Finding a problem in a searching place (industry, conference room, offices, etc...)
Sept 11–Sept 25, 2025	Discussion on effective problems and identification of most real problem.
Sept 26–Oct 10, 2025	Final selection of the problem.
Oct 11 – Oct 25, 2025	Collect references (books, journals, research papers, online sources).
Oct 26–Nov 10,2025	Fixing a suitable project title.
Nov 11–Dec 10, 2025	Literature review (study of past work, existing solutions, gap).
Dec 11–Dec 31, 2025	Discussion on costing of project and estimation of budget (~25k).
Jan 1– Jan 20, 2026	Preparation of block diagram (input–process–output).
Jan 21 –Feb 20, 2026	Methodology and flow chart preparation.
Feb 21–Mar 15, 2026	Draft report preparation (Intro, Problem statement, Literature, Costing, Block diagram, Methodology)
Mar 16–Apr 10, 2026	Correction, editing, and final report writing
April 11–Apr 26, 2026	Final submission, viva and presentation.

VIII. CONCLUSION

In conclusion, this project shows how smart technology can make our daily life easier and more comfortable. The IoT-based Smart Air Mouse and Voice Home Automation system helps users control their computer by moving their hand in the air and control home appliances by simply speaking commands. It reduces the need to touch devices or move around to switch appliances on and off. This makes the system very useful for elderly people, physically challenged individuals, and even busy people who want quick and easy control. By using internet connectivity, the system allows devices to respond quickly and can even be controlled from a distance. The project combines hand gesture control and voice control in one simple system, making it practical and user-friendly. It also helps in saving time and reducing physical effort in daily activities. Overall, this project proves that technology can be simple, helpful, and meaningful when designed for human comfort. It supports the idea of a smart home and shows how modern solutions can improve the quality of everyday life in a safe and convenient way.

REFERENCES

[1] Espressif Systems, ESP32 Technical Reference Manual, *Available at:* https://www.espressif.com

[2] Espressif Systems, ESP-NOW User Guide, *Available at:* https://docs.espressif.com

[3] Ai-Thinker, VC-02 Offline Voice Recognition Module Datasheet, *Ai-Thinker Technology Co., Ltd*.

[4] InvenSense, MPU6050 6-Axis Accelerometer and Gyroscope Datasheet, *TDK Corporation*.

[5] Arduino Official Documentation, Arduino Programming and IDE Guide, *Available at:* https://www.arduino.cc

[6] Neil Kolban, ESP32 BLE and IoT Development Guide, *Available online at:* https://github.com/nkolban

[7] Random Nerd Tutorials, ESP32 ESP-NOW Communication Guide, *Available at:* https://randomnerdtutorials.com

[8] Electronics Hub, Interfacing MPU6050 with ESP32, *Available at:* https://www.electronicshub.org

[9] Ai-Thinker Official Documentation, Voice Recognition Module User Manual, *Available at:* https://docs.ai-thinker.com

[10] IEEE Research Papers on IoT and Smart Home Automation Systems.

[11] Embedded Systems Design by Peter Marwedel, *Springer Publication*.

[12] Internet of Things: A Hands-On Approach by Arshdeep Bahga and Vijay Madisetti.

[13] M. R. Prayogi and T. Tommy, "Design and Development of Air Mouse Using ESP32 and MPU6050 Sensor," *Journal of Artificial Intelligence and Engineering Applications,* vol. 4, no. 3, pp. 1678–1686, Jun 2025.

[14] R. Krisna, "ESP32 Air Mouse using Bluetooth and MPU6050," *Circuit Digest,* Apr. 22, 2025. Krisna's article provides a comprehensive guide on building an air mouse using the ESP32 and MPU6050.

[15] N. Rithik, "DIY Air Mouse with Media Controls", Instructables 2025 This tutorial outlines the creation of air mouse incorporating media control functionalities.

[16] T. Qiuhong, L. Shaobo, and J. Shaoquan, "Design and Production of Air Mouse," *Topics in Intelligent Computing and Industry Design,* vol. 1, no. 1, pp. 164–166, 2017.

[17] A.S. Al-Fahoum and M. S. Abadir, "Design of a Modified Madgwick Filter for Quaternion-Based Orientation Estimation Using AHRS," *Int. J. of Computer and Electrical Engineering*, vol. 10, no. 3, pp. 174–186, 2018.

****