Grenze International Journal of Engineering and Technology, June Issue

Hands Free, Mind Full: The Gesture - Controlled Presentation Revolution

Lahari Valluri[1], Kranthi Madala[2], Thanuja Nadikattu[3], Eswar Avsetti[4] and Mehaetab Shaik[5]
[1,3-5]Department of EIE, V.R. Siddhartha Engineering College, Vijayawada, India
Email: laharivalluri18@gmail.com, nadikattuthanuja9@gmail.com, avisettieswar@gmail.com, mehaetab@gmail.com
[2]Asst.Professor, Department of EIE, Siddhartha Academy of Higher Education, Deemed to be University, Vijayawada, India
Email: kranthimadala16@gmail.com

***Abstract*— The proposed system is based on gestures; features an IoT project which allows for remote control of your presentations via Bluetooth or Wi-Fi through an ESP8266 microcontroller with Python software running on a computer. The ability to navigate presentation slides wirelessly with different hand gestures is available, so that there is no need for the use of keyboards, remote controls, etc. The ESP8266 is used to transmit the gesture commands received from the sensor to the receiver unit by Wi-Fi through a computer, where a Python-based software utilizes this data and can perform the required tasks (e.g., go to the next slide, go back to the previous slide, start and stop the presentation). The hands-free, contactless and inexpensive solution provided by this system enhance interactivity in presentations as well as increase mobility for presenters. Because the system is wireless, it allows presenters to move around freely while having total control of their presentations. This proposed system will not only allow presenters to use the system with ease, but also provide scalability and flexibility to add future improvements to the system. This system could be enlarged to accommodate the use of more gestures, voice commands and/or use machine learning to recognize gestures with high levels of accuracy. Therefore, this system has the potential to be integrated into smart classrooms, smart offices.**

***Index Terms*— Gesture Control, Internet of Things (IoT), Wi-Fi Communication, Python, Wireless Presentation System.**

I. Introduction

As a result of changes in technology, human beings have experienced a significant shift in the way they interact with electronic systems during this new digital age. Human-Computer Interaction (HCI) now encompasses input devices that were once considered to be traditional, such as keyboard and mouse inputs and remote controls, and provides new means of input through more natural human-based controls on touch, voice, gesture-based control, etc. One of these means, gesture recognition, has received considerable attention and has gained recognition in that it operates touchlessly, is easy to use, and creates a better user experience. Meanwhile, the rapid growth of the Internet of Things (IoT) has enabled the control of communications between various IoT Devices and has made the execution of automated processes much more intelligent, faster, and efficient. Presentations are critical components of education, business, educational seminar, and conferences. Traditional presentation control methods, which rely on wired connections and/or handheld remotes, limit a presenter's ability to move around and interact with his/her audience.

Grenze ID: 01.GIJET.12.2.351_24

Common technical issues may arise, such as range limitations, battery life, and the necessity to maintain proximity to his/her computer, resulting in the potential for an interruption to the flow of a presentation. An alternative presentation control method needs to be developed to provide more flexibility and wirelessly/hand free to eliminate these issues. The proposed project aims to develop a gesture enabled remote presentation hardware that will utilize the NodeMCU ESP8266 microcontroller along with a gesture recognition sensor and the use of the Python programming language for integration into the IoT environment. The proposed device will allow users to manage their presentation slides with only hand movements which will be transmitted via wireless communication (Wi-Fi). By using this system, presentations will be more productive for both the presenter and the audience, as well as allowing greater freedom of movement by the presenter due to the absence of touch points on control devices. The gesture based controlled presentation system involves the use of transmission (gesture recognition through the NodeMCU), and reception (processed by a computer running Python code). The gesture recognition sensor sends the user's hand movement to a NodeMCU, which interprets the user's hand movement into digital commands. The digital command signals are transmitted wirelessly from the NodeMCU to another NodeMCU that is connected to a computer running the Python control program. The Python control program receives and processes the digital command signals for appropriate actions (navigation of individual slides and complete control over the presentation). Overall, this project represents a reasonable price point and a good value for the presenter. The project demonstrates actual application of both the IoT and gesture-recognition technologies and allows for future developments of these technologies, including multiple gestures, voice command integration, and machine-learned automatic settings to help improve accuracy and efficiency of usage of these technologies.

II. Literature Survey

“This section reviews previously published literature and articles and conducts an online search for relevant information, laying the groundwork and foundation for future work”

Gupta et al.(2017) showed how a MEMS-based sensor could detect hand gestures and facilitate human-computer interaction (HCI). The study aimed to identify the movements of the hand for converting into commands to operate electronic devices. It identified multiple types of gesture recognition, with those based on sensors having lower latency and computational complexity. However, the need for users to wear the sensor made it somewhat inconvenient to use. Despite this inconvenience, the study concluded that MEMS-based gesture interactions should be considered reliable, cost-effective and practicable for real-time embedded systems [1].

B. R. Sharma et al. (2018), “Gesture controlled presentation system with the computer vision”. Sharma et al. came up with a gesture-controlled presentation system based on web camera and image processing method. The movements are detected by processing the camera images with the OpenCV and were used to navigate the slides using the hand gestures. The system facilitated next slide, previous slide and pointer movement. The system used was very sensitive to light conditions and background noise despite the fact that it used hardware sensors which were eliminated. The authors have found that presentation control with the use of vision is efficient but demands a great amount of processing power and controlled conditions [2].

C. K. Patel (2019) “IoT-based wireless control of devices with the use of the Node MCU ESP8266)” Researchers Patel and associates developed an Internet of Things (IoT), with wireless control capabilities, for devices using the NodeMCU ESP8266 microcontroller. They designed the control mechanism to employ communication with Wi-Fi technology to transmit the control signals wirelessly. The authors described the ESP8266 as reliable, low power consumption and inexpensive. The results of their experimentation demonstrated that the NodeMCU is appropriate for real-time IoT applications, including smart home management, smart switches and remote-control management systems [3].

D. S. Mehta "Design of touchless gesture-based control system using APDS-9960 sensor" According to Mehta et al., touchless control can be created through the utilization of an infrared gesture sensor (APDS-9960) and a gesture controller for interfacing. The gesture controller was programmed to provide a method of recognizing left, right, up, and down movements. A microcontroller was utilized to connect the system together and to execute commands in real-time. Additionally, the group was able to demonstrate small size, low-cost development, and also develop an infrared gesture sensor so they could use the sensor without physically touching it; therefore it could be used in situations with high amounts of traffic. However, the group experienced difficulties in accuracy due to light and distance limitations. The accuracy of the sensor was determined to be accomplished by using calibration methods [4].

E. P. Nair et al. (2021), Automation system utilizing Python and a variety of library resources that allow them to use the Serial Communication protocol interface and Keyboard and Mouse Control protocol interface to interact with external hardware on computers via a number of different outputs. The hardware developed by Nair converts output data from the microcontrollers controlling the devices into the equivalent Keyboard / Mouse operations so that they can control Applications in Real Time. The research performed by these authors demonstrates that Python is a very versatile programming language and supports the integration and execution of automation systems with a very high degree of speed. Their overall conclusion was that Python is a viable programming environment in which to develop automation applications from a PC to control Embedded Systems and IoT Devices.[5]

R. K. Verma and S. Tiwari (2019) In this research paper, researchers have evaluated several different techniques for human-computer interaction that involve the recognition of hand gestures. Both the authors analysed various vision-based (and sensor-based) as well as hybrid gesture recognition techniques; discussing their advantages and disadvantages. The conclusions drawn from their analyses implied that although vision-based gesture recognition was more flexible than using a sensor-based technique, a sensor based approach would be better because of its reliability and lower computational overheads. Therefore sensor-based techniques can be used successfully for real-time applications [6].

A new machine learning-based hand gesture recognition system that is able to operate in real time for intelligent control applications was proposed by M. Al-Hammadi and colleagues (2020). The authors indicate that machine learning systems dramatically outperform traditional hard-coded non-machine learning approaches relative to the recognition of gesture data; however, these systems require large amounts of data and significant computing capabilities in order to achieve a true real time, real world performance level [7].

S.Banerjee and Paul (2019), have created a cost-effective IoT-based system utilizing the ESP8266 NodeMCU as a means to wirelessly monitor devices. As a result of developing this system, both authors supply significant evidence that shows NodeMCU supports reliable two-way communication via Wi-Fi and is a high-value solution due to its programming simplicity and reliable wireless communication capabilities [8].

H. Y. Lee and K. S. Kim (2020) The authors reviewed infrared-based gesture sensing technologies to control embedded systems with accuracy, but with minimal sensitivity to the environment. They concluded that infrared gesture detection has greater reliability than gesture detection using vision-based systems in terms of variations in lighting; therefore, infrared sensed gesture systems can be used reliably in real-time control systems [9].

P. Mishra et al. (2018) proposed wireless presentation control system, which was designed using embedded technology, allows users to control the movement of slides from a distance. The design for the wireless presentation control system has successfully implemented some basic controls such as transition effects to advancing slides over a wireless connection. Although the system is known to be quite effective, it has been noted that the scalability and flexibility of the gestures used on the controller hinder the incorporation of additional more advanced sensing and automation technologies [10].

J. R. Fernandez and M. Lopez (2019) researchers conducted an in-depth examination of several different forms of interaction between people and machines. They also discussed how to improve the usability of interaction with machines by creating interfaces that use natural movements (gestures) for those who are connecting to the machine and those who are connecting to the interface. They indicated that gesture-based human/computer interfaces offer users an improved level of usability; however, they will only be effective if less than 5% of the people using the gesture-based interface do not understand its design or feel fatigued when using it [11].

G. Singh & P. Kaur (2020) described an IoT-based automation system that utilizes a NodeMCU as well as provides for cloud-computing communication capabilities has been reviewed. It allows users to control their devices remotely via the Internet. The authors noted that the various IoT platforms provide an integral component of modern-day smart automation systems through their ability to transmit real-time data to/from these systems [12].

L.Brown, et al. (2020) investigated how to use the APDS-9960 sensor in a gesture-based control environment. They concluded that utilizing an APDS-9960 sensor for gesture recognition results in better accuracy than other sensors with respect to gesture recognition and that the APDS-9960 is particularly suitable for use in small devices where efficient power consumption is an important factor [13].

M. Das and R. Agrawal (2020) developed a clever class framework utilizing microcontrollers and gestural capability. Their study found that pupils could connect with the framework through gestures rather than touch. They noted that employing gestural control would improve participant engagement as well as facilitate hygienic conditions, especially in areas shared by multiple individuals like at the university and conference Centre [14].

The exploration into the automation capabilities of Python and its associated libraries was conducted by S. Patel and R. Mehra (2021) using a study of their use in interfacing with people. One area of focus was the use of Python to automate system functions such as keyboard and mouse activity and the advantages this provided for interacting with machines; the authors found that this type of automation provided flexibility, ease, and efficient development for HCI systems. [15].

III. PROPOSED METHOD

The proposed work brings forth an IoT-based gesture-controlled presentation system that allows users to navigate presentation slides remotely without the use of gestures. The presentation system is designed to enhance user interactions as it does not use traditional presentation systems like remotes and keyboards at all; rather, it implements a fully gesture-controlled interface. The presentation system is divided into a dual- module design comprising a transmitter and receiver module that comprises Node MCU ESP8266 microcontrollers. The dual-module design enables perfect wireless communication via Wi-Fi algorithms, ensuring that there are no lag problems during presentation. In the transmitter module, a gesture sensor similar to the MIN/APDS-9960 is utilized to detect directional gestures and relay them to a digital signal that is processed by the Node MCU microcontroller to decode the signal into a command that corresponds to a gesture and transmit the command to the receiving Node MCU microcontroller wirelessly.

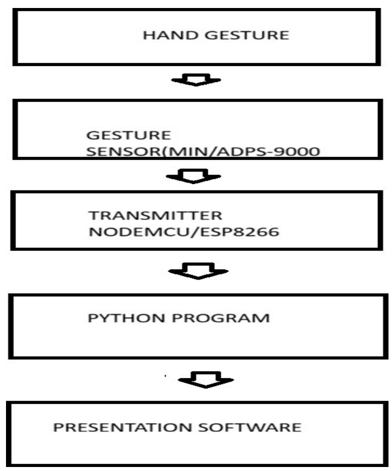

Figure 1: Block Diagram of Proposed Model

In the portable transmitters, a battery pack is utilized as a power source and a push button that acts as a resetting/calibrating button The transmitted command is captured by the receiver unit and then sent to any connected computer via serial communication. The script running on the computer, based on Python, translates the received command into keyboard actions controlling the presentation software, such as Microsoft PowerPoint or Google Slides. The approach results in smooth and intuitive presentation while increasing both mobility and user engagement. The modularity will also permit further upgrading, multi-gesture recognition, voice integration, or even machine-learning-based accuracy enhancement. Figure 1 shows the block diagram of proposed model.

IV. METHODOLOGY

The design of IoT-based gesture-controlled presentation systems follows a structured theoretical and experimental plan that includes integration of hardware interfacing using wireless technology and developing software to automate the process of navigating slides by means of gestures. The process begins with choosing and configuring a gesture sensor (e.g., MIN/APDS-9960), which is able to detect hand gestures in multiple directions, and connecting it to a transmitter Node MCU ESP8266 via I²C protocol. The microntroller continually reads the gestures, clears out any extraneous noise, identifies the gesture made, and correlates that gesture with a previously defined command (e.g., NEXT, PREV, START, or STOP). The transmitter Node MCU packages the commands into lightweight packets and transmits them wirelessly to the receiver Node MCU using Wi-Fi communication (UDP/HTTP). The receiver Node MCU will always be listening (in station or access-point mode) for a valid command packet. When a command is received, the receiver Node MCU will send the command data to the associated computer, via USB serial connection.

A. Working Steps

- Power-On and Initialization of System: On turning on the system, the gesture sensor as well as the two Node MCU modules initialize their communication channels. The transmitter Node MCU initializes its communication channel with the gesture sensor through its I²C interface, and it also establishes its Wi-Fi connection to communicate with the receiver Node MCU.
- Gesture Detection by Sensor: The user does a hand gesture (left, right, up, or down) in front of the APDS-9960 or MIN gesture sensor. The sensor is able to detect the direction of movement, which results in an output signal transmitted to the transmitter Node MCU for processing.
- Processing and Command Generation: The Node MCU in the transmitter detects the gesture data and identifies the type of gesture, and then it translates it into a command like NEXT, PREV, START, or STOP. The commands are then packaged into a wireless message that needs to be transmitted.
- Wireless Transmission: The transmitter broadcasts the gesture signal through Wi-Fi, which uses UDP/HTTP communication protocols to transfer the signal to Node MCU. communication ensures the presenter moves freely without any physical impediment.
- Command Reception at Receiver Node MCU:The receiver Node MCU is constantly waiting to receive the packets. As soon as the command is received, it transfers the information related to the gesture to the computer via the USB serial connection.
- Python Script Explanation: There is a Python automation script executing on the computer that interprets the arriving serial data and assigns it to a keyboard command using modules such as keyboard.
- Slide Control Execution: Python code sends a keystroke signal (Right Arrow, Left Arrow, F5 keys, or Esc key) for execution, which will control PowerPoint Presentation software directly. The speaker caneasily move to a next slide using this solution.
- Continuous Monitoring and Real-Time Execution The system continuously goes through these steps, allowing real-time recognition of gestures and real-time control over the presentation if the gestures are within the range covered by the sensor.

B. Modules

Software Modules

- Transmitter Firmware (Node MCU Code): This code is written in the Arduino IDE. The code takes the gesture data from the motion sensor and turns it into command signals (NEXT, PREV, START, STOP) for the transmitter's Wi-Fi connection. The command signals are then sent over a Wi-Fi wireless network using the receiver firmware (Node MCU Code): This code also monitors for incoming Wi-Fi signal broadcasts, decodes the motion gesture and sends the decoded signal as serial data to a computer.
- PC Automation Script in Python: PC automation is accomplished from the computer through a script that listens for incoming serial data transmitted from the wireless transmission node (Node MCU). In order to accomplish this type of automation, a number of libraries are utilized; these libraries allow the automation script to convert the command signals sent from the controlling node into simulated keyboard events or commands that can be interpreted by presentation software.
- Wi-Fi Communication Protocol: This protocol is responsible for managing the communication of data over the Wi-Fi connection between the transmitter and receiver. It may use either UDP or HTTP as a means of communication.

Hardware Modules

- Gesture Sensor (MIN/APDS-9960): A sensor shown in Figure 2 that detects the direction of your hand movement, such as left/right/up/down. It connects to the Node MCU via I²C protocol; it acts as an input device.

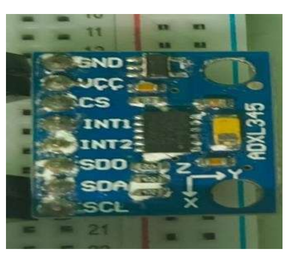

Figure 2: MIN sensor

- Transmitter Node MCU ESP8266: Is shown in Figure 3 receives gesture data from the sensor, sends the command codes via Wi-Fi to the receiver module using a processed method of encoding.

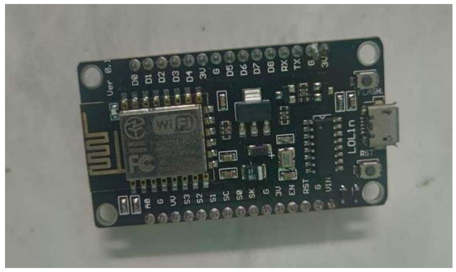

Figure 3: NODEMCU ESP8266

- Receiver Node MCU ESP8266: Receives gesture command codes and Wi-Fi packets of transmitted data from the transmitter Node MCU module; relays data to the computer via USB cable Power Supply
- Battery Module: Supplies power to the transmitter and node; Li-ion battery or regulated 5V source provides stable power supply to the transmitter node.
- Push Button Module: Set on transmitter side; can be used for resetting/calibrating/mode-selection related to gesture sensor.
- Computer / Presentation Device: Connected to the receiver Node MCU with USB cable; runs program that translates gesture command received for presentation via Python.

V. Results and Discussion

The outputs displayed in Table I of the gesture-based presentation system include all the actions are performed by your presentation software (e.g., PowerPoint or Google Slides) based on the user-gestures captured through the touchscreen controller. To illustrate, when the touchscreen controller detects your gesture, it sends an instruction signal via its transmitter (Node MCU), which is then sent to another Node MCU (the receiver) to relay the command to your computer. MCU (the receiver) to relay the command to your computer. Your computer's automation program in Python receives the instruction from the receiver and responds by issuing a separate instruction that produces a keystroke event based on that first instruction. By using this system, you can enjoy seamless slide transitions, regardless of whether you physically interact with your computer. The system will output the command NEXT when the user uses a right swipe action, which thereby changes to the next slide of the presentation program. Left swipe action of the user generates the command PREV, which thereby changes to the previous slide of the program. Upward action of the user may activate the program, while downward action may terminate the program. These results prove the ability of the system to correctly and consistently respond to any user gesture seamlessly without requiring the use of human hands. The final output showcases efficient wireless communication with low latency and precise mapping from gestures to actions, thereby generally proving its utility.

TABLE I: THE OUTPUTS OF GESTURE-BASED PRESENTATION SYSTEM

Gesture Performance	Detected Command	Transmitter Action	Receiver Action	Final Output on PC / Presentation
Right	NEXT	Gesture encoded & sent via Wi- Fi	Received	Next Slide
Left	PREV	Gesture encoded & sent via Wi- Fi	Received	Previous Slide
Up	START	Gesture converted to START command	Received	Starts Slideshow
Down	STOP	Gesture converted to STOP command	Received	Stop Slideshow
No Gesture / Idle	NONE	No data transmitted	No action	No Effect

TABLE II: COMPARISON OF EXISTING SYSTEM AND PROPOSED SYSTEM

Feature / Parameter	Existing Systems (Vision / Remote Based)	Proposed Gesture-Controlled System
Control Method	Remote control / Camera-based gestures	Sensor-based hand gestures
Hardware Requirement	Webcam / Remote device	NodeMCU ESP8266 + APDS-9960 sensor
User Interaction	Requires holding device or staying in camera view	Fully hands-free interaction
Accuracy	Affected by lighting and background noise	More stable (sensor-based detection)

Computational Requirement	High (image processing using OpenCV)	Low(microcontroller-based processing)
Cost	Moderate to high	Low-cost implementation
Reliability	Less reliable in dynamic environments	More reliable and consistent

VI. CONCLUSIONS

This system uses the IoT and allows for a new and modern way of interacting with a presentation via touch free gestures to move through a slide show. The system described utilizes a gesture sensor and Node MCU ESP8266 modules to create a wireless gesture-controlled automation script in Python, enabling seamless wireless communication and smooth navigation through slides without the need for several traditional remotes or keyboards. With this design, there is an improvement in mobility and participation, while reducing the number of physical interactions. This solution is well-suited to smart class settings, office environments, seminars, and interactive presentation environments. The project demonstrates the successful integration of the internet-of-things technology with human-computer interaction in creating a user-centered solution for productivity and efficiency. This project also represents an innovative and effective solution for creating an automated and intelligent presentation system. And the comparison of existing systems and proposed systems are mentioned in the Table 2.

REFERENCES

[1] A. Gupta, “Hand gesture recognition for human–computer interaction using MEMS sensors,” International Journal of Human–Computer Interaction, vol. 33, no. 4, pp. 250–258, 2017.

[2] B. R. Sharma, “Gesture controlled presentation system using computer vision,” Proceedings of the International Conference on Image Processing and Analysis, pp. 112–118, 2018.

[3] C. K. Patel, “IoT-based wireless device control using Node MCU ESP8266,” Journal of Internet of Things and Embedded Systems, vol. 6, no. 2, pp. 45–52, 2019.

[4] D. S. Mehta, “Design of touchless gesture-based control system using APDS-9960 sensor,” International Journal Sensor Technologies, vol. 8, no. 1, pp. 30–37, 2020.

[5] E. P. Nair, “Python-based automation for human–computer interaction,” International Journal of Computer Applications, vol. 179, no. 22, pp. 15–21, 2021.

[6] R. K. Verma and S. Tiwari, “A review on hand gesture recognition techniques for human– computer interaction,” International Journal of Computer Engineering and Applications, vol. 12, no. 3,pp. 90–98, 2019.

[7] M. Al-Hammadi et al., “Real-time hand gesture recognition using machine learning for smart control applications”, IEEE Access, vol. 8, pp. 196000–196010, 2020.

[8] S. Banerjee and A. Paul, “Development of a low-cost IoT system using ESP8266 for wireless device monitoring,” International Journal of IoT and Smart Technology, vol. 5, no. 2, pp. 25–33, 2019.

[9] H. Y. Lee and K. S. Kim, “Infrared gesture sensing technologies for embedded control systems,” Sensors and Actuators A, vol. 301, pp. 111–119, 2020.

[10] P. Mishra et al., “Wireless presentation control using embedded systems,” Proceedings of the International Conference on Embedded Computing, pp. 45–50, 2018.

[11] J. R. Fernandez and M. Lopez, “Human–computer interaction through gesture-based input devices,” Journal of Interaction Science, vol. 7, no. 1, pp. 1–12, 2019.

[12] G. Singh and P. Kaur, “IoT-based automation using Node MCU and cloud communication,” International Journal of Advanced Networking and Applications, vol. 11, no. 6, pp. 4557–4563, 2020.

[13] L. Brown et al., “Evaluating APDS-9960 performance for gesture-driven interfaces,” International Journal of Sensor Networks, vol. 15, no. 4, pp. 210–218, 2020.

[14] M. Das and R. Agrawal, “Gesture-enabled smart classroom system using microcontrollers,” Journal of Educational Technology Systems, vol. 49, no. 1, pp. 72–85, 2020.

[15] S. Patel and R. Mehra, “Python automation for interactive systems: A review of libraries and applications,” International Journal of Computer Science Trends and Technology, vol. 9, no. 3, pp. 14–22, 2021.