Journal of Artificial Intelligence and Engineering Applications

Website: https://ioinformatic.org/

15th June 2025. Vol. 4. No. 3; e-ISSN: 2808-4519

Design and Development of Air Mouse Using ESP32 and MPU6050 Sensor

Muhammad Ryan Prayogi[1]*, Tommy[2]

[1,2]Informatics Engineering Study Program, Faculty of Engineering and Computers
Universitas Harapan Medan
ryanprayogi776@gmail.com [1*]

Abstract

In the era of modern technology, innovations in digital input devices are growing, one of which is an air mouse based on ESP32 and MPU6050 sensors. The air mouse allows users to control the cursor on the computer screen with just a hand movement, which provides more comfort and flexibility compared to a conventional mouse. This research aims to design and develop an air mouse device using the ESP32 microcontroller and MPU6050 sensor, and test its performance in transforming hand movements into responsive cursor movements. The methods used include hardware design, installation of sensors and modules connected to the ESP32, and testing to ensure the system functions according to specifications

The results of this study show that the ESP32 and MPU6050-based air mouse is able to detect hand movements and move the cursor on the computer screen accurately. The MPU6050 sensor, which functions to detect acceleration and rotation, provides good performance in cursor control, although it needs to be calibrated against noise and latency. The use of Bluetooth Low Energy (BLE) technology enables stable and efficient communication between the device and the computer. Tests show that the device works well, although there is still room for improvement in terms of responsiveness and accuracy.

Keywords: Air Mouse, ESP32, MPU650, Bluetooth Low Energy, Motion Sensor

1. Introduction

In this modern era, technological developments have had a significant impact on various aspects of human life, including the way we interact with digital devices. One technological innovation that is gaining more attention is the use of motion sensors and microcontrollers in the creation of input devices, such as the air mouse. An air mouse, as a device that allows users to control a computer cursor with hand movements in the air, offers flexibility and convenience that a conventional mouse cannot provide. This technology is particularly useful in presentations, education, games, and other applications that require dynamic and intuitive interaction. The ESP32 microcontroller is a microcontroller designed by the Espressif Systems company in Shanghai China. The ESP32 microcontroller is equipped with an Xtensa dual core LX6 processor, with a voltage of 3.3 Volts, and has Wi-Fi and BLE (Bluetooth Low Energy) support (WK et al., 2021). With its abundant features and relatively low cost, the ESP32 allows developers to create innovative solutions that are accessible to a wide range of people. In addition, the use of the MPU6050 sensor, which is capable of detecting acceleration and rotation, makes it particularly suitable for monitoring hand movements and translating them into cursor movements on a computer screen. The MPU-6050 sensor is a sensor that functions to track the movement of an object, the MPU-6050 sensor has a DMP (Digital Motion Processing) feature which aims to get a value from the sensing results, the MPU-6050 sensor can read the yaw, pitch, and roll values from the sensing results that have been carried out by the user and then processed into a cursor on a laptop. (Darmawan et al., 2019).

The use of BLE (Bluetooth Low Energy) Technology is done because it is seen from several aspects, first is the aspect of convenience and integration, where the ESP32 module already has a BLE module that is integrated directly, so there is no need for integration with a separate BLE module, this is also certainly related to the second aspect, namely costs that will decrease, compared to the integration of a separate module (Jonas et al., 2022).

While this technology promises many conveniences and advantages, the development of an air mouse based on ESP32 and MPU6050 sensors still faces technical challenges, especially in terms of processing and transmitting motion data in real-time with minimal latency. A smooth and responsive user experience depends on the accuracy of interpreting the motion data obtained from the sensors, as well as processing the data in real time. This is the main focus of this research to overcome these challenges and optimize the performance of the air mouse device.

In the context of education and presentations, the use of an air mouse can offer very significant benefits. In a classroom or seminar, for example, a presenter can move freely while still controlling their presentation, without needing to rely on traditional input devices such as a mouse or pointer. This not only improves the interaction between the presenter and the audience, but also makes the presentation more dynamic and engaging. As an aid in the learning process, the air mouse enables more innovative and efficient teaching (Pratiwi & Handoko, 2019).

The development of this ESP32-based air mouse is expected to contribute to the advancement of wireless input device technology, which has the potential to be used in various applications other than education, such as in entertainment, sports, and industrial control. One of the interesting things is how this technology can be integrated into a wider system, for example in the automation and control of other devices in smart environments (Musa, 2023).

Despite its great potential, the development of an air mouse based on ESP32 and MPU6050 faces a number of technical challenges. One of them is how to transfer motion data in real-time with minimal latency so that the user experience remains smooth and responsive. Therefore, this research is important to overcome these challenges and optimize the performance of air mouse devices.

In the context of education and presentations, the use of an air mouse can add significant value. For example, in a classroom or seminar environment, presenters can move freely while still controlling their presentation without depending on traditional input devices. This not only improves the interaction between the presenter and the audience but also makes the presentation more dynamic and engaging. Although this technology is still in the early stages of development, a successful implementation could pave the way for a variety of innovative applications in the future.

2. Methods

System Design Analysis

System analysis is the first step in understanding and designing the system to be developed. At this stage, identification of needs, system boundaries, and relationships between components in the system to be built are carried out. This process aims to describe in detail each component in the system and describe the interactions between these parts. In this context, the analysis aims to evaluate existing problems and identify solutions or improvements needed to develop an ESP32 and MPU6050-based mouse tool. This analysis includes describing the flow and processes that exist in the device, which will produce outputs according to user needs.

Problem Analysis

Along with the development of technology, various innovations have emerged to facilitate human interaction with digital devices. One such innovation is the air mouse, which allows users to control a computer cursor with just hand movements in the air, without requiring physical touch on conventional input devices. To develop this system, an ESP32 microcontroller and an MPU6050 motion sensor are used as the main components. Although it offers great potential, there are several challenges that need to be overcome, such as hardware and software integration, and ensuring smooth user interaction. Therefore, it is important to analyze and overcome these technical challenges so that the developed air mouse can function effectively and responsively according to the user's expectations...

System Block Diagram

The system design uses UML modeling language which consists of Use case Diagram, Class Diagram, Activity Diagram, and Sequence Diagram.

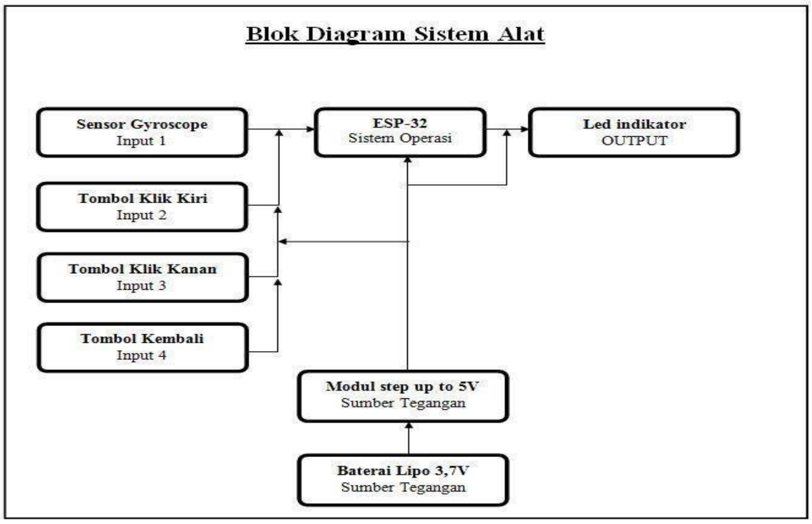

Figure 1. System Block Diagram

The system block diagram in Figure 1 illustrates the main components that interact with each other in an input device system. The Gyroscope sensor (Input 1) detects the motion and orientation of the device, while the left-click button (Input 2), right-click button (Input 3), and back button (Input 4) provide navigation command or menu selection input to the ESP-32 microcontroller. The ESP-32 (Operating System) serves as the brain of the system, processing all inputs and outputting signals to control the indicator LEDs (Outputs), which provide visual feedback to the user. The power source of the system comes from a 3.7V LiPo battery, which is supplied voltage through a step-up module to 5V, ensuring components such as the ESP-32 operate properly.

Flowchart

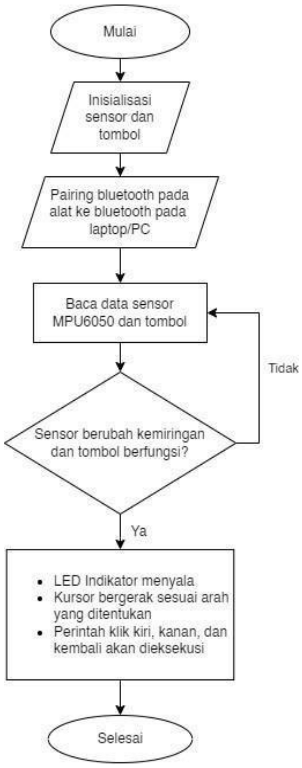

Figure 2. Flowchart of ESP-32-based Air Mouse

The flowchart in Figure 2 illustrates the process of operating an ESP-32 Mouse device that uses an MPU6050 sensor and a button to control the cursor via Bluetooth connection with a laptop or PC. The process starts with initialization of the sensor and button, followed by Bluetooth pairing to connect the device to the computer. The device then reads data from the sensor and button continuously, checking for any change in tilt or button pressed. If there is a change, the device moves the cursor and sends a click command according to the button pressed, such as left-click, right-click, or back button. This process repeats continuously to ensure the device can control the cursor and execute commands responsively.

Tool Design Process

To assemble a protoype requires a design that must be done so that a tool can work properly. At this stage, designing tools for installing MPU6050, LED Indicators, buttons & push buttons to pins found on the ESP-32 microcontroller using jumper cables so that the tool and its functions can run properly. The tool design process is carried out by designing electronic circuits and framework designs using the Fritzing application. With Fritzing, the circuit diagram can be made clearly and structured, making it easier in the physical assembly process.

a. MPU6050 Sensor Installation to ESP32

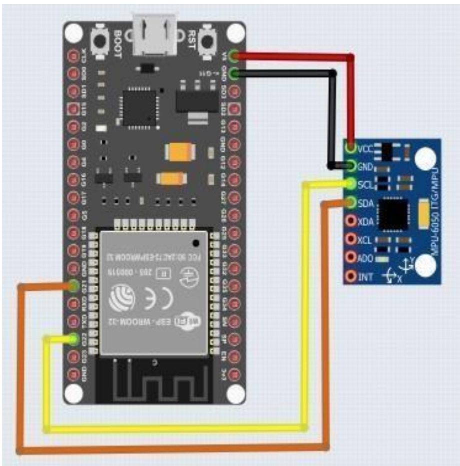

Figure 3. Installation circuit of MPU6050 to ESP32

Figure 3 shows an illustration of connecting the MPU6050 with the ESP32, the VCC pin of the MPU6050 is connected to the VIN pin of the ESP32 for power, and the GND pins of both devices are connected to complete the circuit. The SCL (Serial Clock Line) pin of the MPU6050 is connected to GPIO pin 21 on the ESP32, while the SDA (Serial Data Line) pin is connected to GPIO pin 22. In the integration steps, first use jumper cables to connect the ESP32 and MPU6050 pins according to the schematics mentioned. Then, install the Adafruit_MPU6050, Wire, and BleMouse libraries on the Arduino IDE.

b. MT3608 Step Up Module Installation

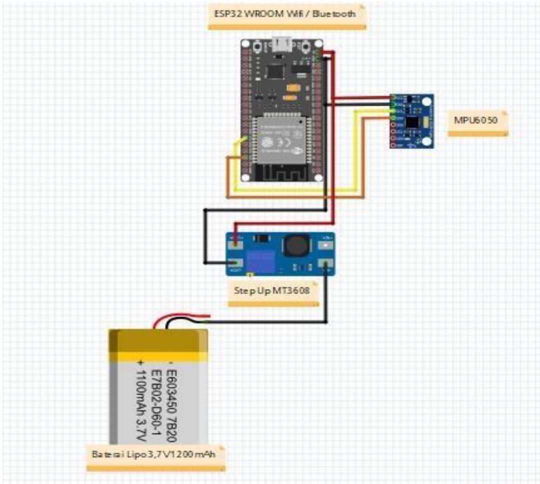

Figure 4. MT608 Step Up Module & battery installation circuit

Figure 4 shows the installation of the Step Up MT3608 module with ESP32 and LiPo battery. The LiPo battery with a nominal voltage of 3.7V is connected to the input pin of the Step Up MT3608 module to provide initial power. The output pin of the Step Up module is then connected to the VIN pin of the ESP32 to provide the 5V voltage required by the module. The GND pins of the battery, Step Up module, and ESP32 are also connected together to complete the circuit and ensure a steady flow of current. To switch the system on or off, an ON/OFF switch is installed in the path between the battery and the Step Up module. A DC Power Female socket is included as an alternative power option from an external source. The integration steps involve using jumper cables to connect the input pins of the Step Up module to the battery, and the output pins to the ESP32, according to the given schematics. Buttons on the schematic can be used to provide manual inputs to the ESP32, such as triggering certain actions.

c. Overall Hardware Design

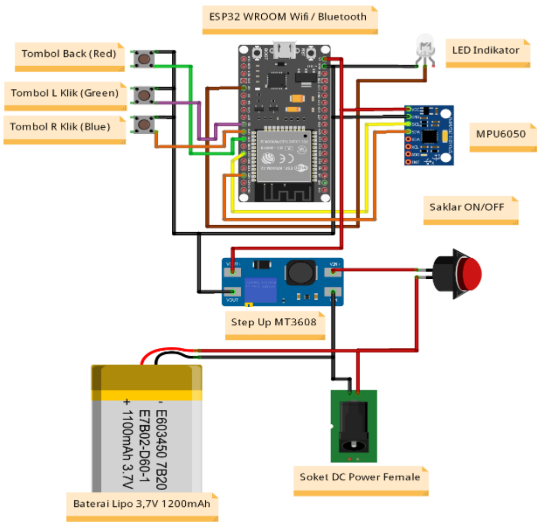

Figure 5. Hardware circuitry

After designing each hardware, the next stage is the design stage for the entire hardware. Hardware such as the MPU6050 Sensor, MT3608 Step Up Module, Button, On/Off Push Button, LED Indicator, and LiPo battery are connected into a single unit with the ESP32 Microcontroller through predetermined pins. At this stage, the entire hardware unit is realized into a prototype so that it can be concluded directly. Figure 3.5 shows the hardware circuit of the ESP32-based air mouse with MPU6050 sensor, where the ESP32 serves as the main microcontroller that manages the entire system, including sending motion data to the computer via Wi-Fi or Bluetooth. The MPU6050 is used as a sensor module that combines an accelerometer and gyroscope to detect movement and rotation, which is translated into mouse movements. Three buttons (back, left-click, and right-click) are used to provide click command input, while indicator LEDs provide visual status about the device, such as whether it is connected or active. The device also comes with an ON/OFF switch to turn the device on or off, as well as a DC power female socket that allows recharging or connecting to an external power source. The MT3608 Step Up Module is used to increase the voltage from the 3.7V 1200mAh LiPo battery to the voltage required by the ESP32 and other components, ensuring the system gets stable and sufficient power.

Table 1. Wiring Connections in the Circuit

Component	Pin	Connects to ESP32 Pins
MPU6050	VCC	3V3
	GND	GND
	SDA	GPIO 21 (SDA)
	SCL	GPIO 22 (SCL)
Back Button (Red)	Pin 1	GPIO 32
	Pin 2	GND
L Click Button (Green)	Pin 1	GPIO 33
	Pin 2	GND
R Click Button (Blue)	Pin 1	GPIO 25
	Pin 2	GND
LED Indicator	Anode (+)	GPIO 2
	Cathode (-)	GND
ON/OFF switch	Pin 1	VIN (via Step Up MT3608)
	Pin 2	Positive battery
Step Up MT3608	VIN+	Positive battery
	VIN-	Battery negative
	VOUT+	ESP32 VIN
	VOUT-	GND
LiPo Battery	Positive	VIN (+) Step Up MT3608
	Negative	VIN (-) Step Up MT3608

Table 1 describes the cable connections in the ESP32-based air mouse device circuit. The MPU6050 is connected to the ESP32 via power lines (VCC to 3V3 and GND to GND), as well as I2C data lines (SDA to GPIO 21 and SCL to GPIO 22). The back (red), left-click (green), and right-click (blue) buttons are connected to GPIOs 32, 33, and 25 respectively to detect inputs, with GND as the circuit completion. The indicator LED is connected to GPIO 2 to control the LED flame, with GND as circuit completion. The ON/OFF switch connects power from the battery to the MT3608 Step Up module, which then steps up the voltage from the battery and supplies power to the ESP32 via VIN and GND. All connections ensure system operation with stable power flow and proper data transmission between components. 3.7V 1200mAh LiPo battery and MT3608 Step Up module. The positive connection from the battery (Positive) is connected to VIN (+) on the Step Up MT3608, which serves to supply main power to the Step-Up converter module. While the negative connection from the battery (Negative) is connected to VIN (-) on the Step Up MT3608, which serves to complete the ground path in the circuit. With this connection, the Step Up MT3608 module can increase the voltage from the battery to meet the power requirements for the ESP32 and other components.

3. Tool Implementation and Testing

3.1. System Implementation

The implementation and testing of the ESP32 and MPU6050-based air mouse system consists of several stages. The first stage is the preparation of hardware, which consists of components such as ESP32 microcontroller, MPU6050 sensor, push button switch, LED indicator, 3.7V 1200mAh LiPo battery, MT3608 Step Up module, DC power female socket, jumper cables, and a box to accommodate all components. All of this hardware is designed to work in an integrated manner, with the ESP32 microcontroller as the control center that manages communication between devices and data processing from sensors. In the implementation stage, the hardware is installed and connected according to the schematic that has been designed, allowing the system to function properly.

Once the hardware was assembled, the next stage was system testing. The air mouse device is turned on via the on/off switch, which then initializes the ESP32 and turns on the flashing indicator LED to show that the device is in the process of searching for a Bluetooth connection. Once connected to a laptop or PC device, the indicator LED will light up continuously. Further testing is done by tilting the device in various directions to ensure that the cursor on the device moves according to the movement of the device. After the movement test is successful, the buttons on the device, such as the left-click, right-click, and back buttons, are tested to ensure they are functioning properly. If all functions run as expected, the tool is declared successfully used and ready for further operation.

3.2. Discussion Results

Based on the design that has been made before. In this section, testing is carried out first on the ESP32 and MPU6050-based Mouse water tool so that it can run as desired. The test carried out is testing several components in the design of the ESP32 and MPU6050-based Mouse water tool. Testing is done to ensure that the components used work properly or do not experience errors. The following are the results of designing an ESP32 and MPU6050-based Mouse water tool using a Bluetooth connection in Figure 6.

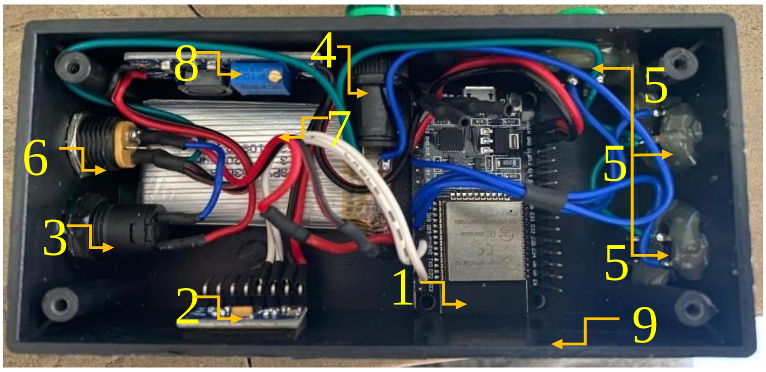

Figure 6. Overall Tool Set

Description:
1) The ESP32 microcontroller serves as the center of data processing and where other components are connected.
2) MPU6050 functions as a sensor that detects certain tilt conditions with a gyroscope.
3) The On/Off switch serves to connect or disconnect electricity. With this, users can manually turn the device on or off.
4) The Indicator LED serves as a notification light that the bluetooth connection is connected or not.
5) The Push Button functions for buttons such as left-click, right-click, and back button as well as the Mouse tool in general.
6) The DC Power Female socket serves as an alternative power option from an external source so that the device can be charged using a 5-volt adapter
7) The 3.7V 1200mAh LiPo battery is useful as a portable power source for the device. This battery provides enough power to operate the device for a period of time without requiring a direct electrical connection.
8) The MT3608 Step Up Module functions as a voltage converter that increases the voltage from the 3.7V battery to a higher voltage (up to 5v) required by other components in the device. This ensures all components receive sufficient power.
9) The box is useful for protecting and housing all the components inside. The box provides physical protection from damage as well as preventing direct contact with the outside environment that could compromise the device's functionality.

Tool System Testing

This tool test aims to see the results and find out whether the circuit and design of the ESP32 and MPU6050-based Air Mouse tool can work in accordance with the design designed by the author. This test is carried out to find out the work of each hardware used can work properly.

a) Testing Bluetooth Connectivity on ESP32

In this test, the first thing to do is check the bluetooth connectivity on the ESP32 which can be seen from the serial monitor in the Arduino IDE.

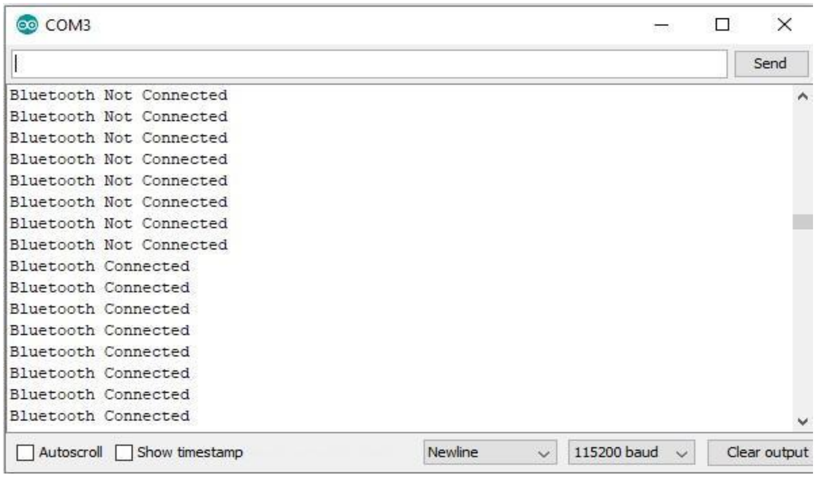

Figure 7. Bluetooth Connectivity Output on ESP32

In Figure 7 it can be seen that Bluetooth connectivity can function properly. In standby mode, the lights on the indicator LED will flash then the message "Bluetooth Not Connected" appears on the serial monitor indicating that Bluetooth is on and ready to be connected to the device you want to use. After the bluetooth on the tool and the device you want to connect are connected and the light on the indicator LED is no longer blinking, a message will appear on the serial monitor "Bluetooth Connected".

Push Button Testing

In this Push Button test, the device is already on and the bluetooth connection on the device is connected to the connected device. This test is used to ensure that each button functions correctly and provides the appropriate input.

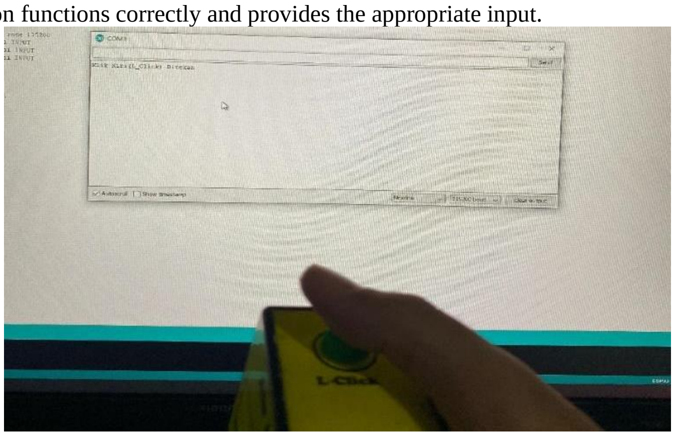

Figure 8. Serial Monitor Button L_Click

Figure 8 shows that when the L_click push button (green button) is pressed, the message "Left Click (L_Click) Pressed" will appear on the serial monitor in the Arduino IDE.

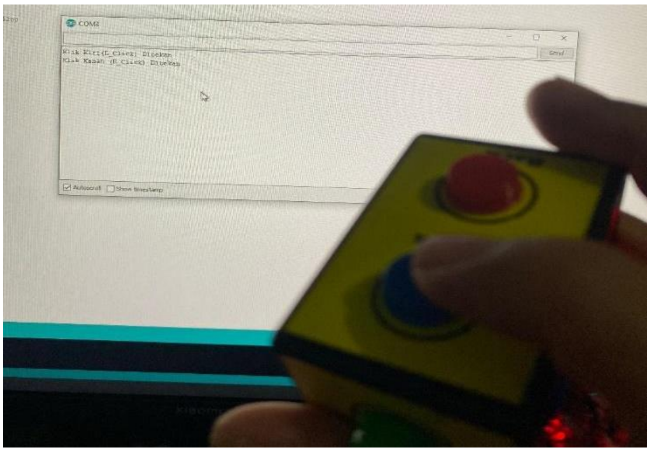

Figure 9. Serial Monitor Button Back

Figure 9 shows that when the R_click push button (blue button) is pressed, the message "Right Click (R_Click) Pressed" will appear on the serial monitor in the Arduino IDE.

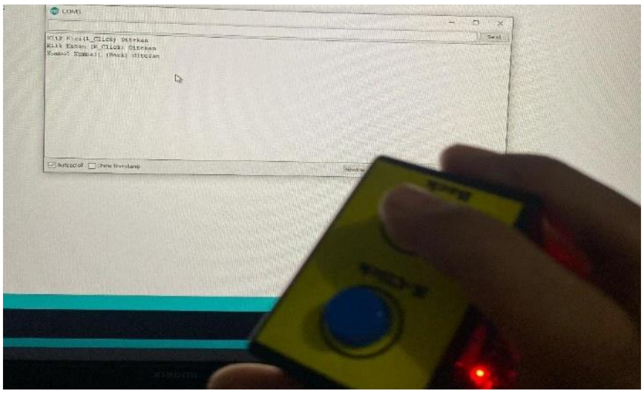

Figure 10. Serial Monitor Button Back

Figure 10 shows a description that when the "Back" push button (red button) is pressed, the message "Back button pressed on the serial monitor in the Arduino IDE will appear. All of these push buttons have the same function as a mouse in general.

Tilt and Latency Calibration

After testing the ESP32 and MPU6050-based Air Mouse tool, there are several shortcomings such as noise when the tool is placed on a flat plane, besides that there are also shortcomings such as the latency of movement that is so fast that the tool is difficult to control. To reduce noise and latency that occurs, a calibration is carried out which is expected to make the tool have no noise and get the appropriate latency.

Table 2. Tilt & Latency Testing Results

NO.	Conditions	Slope (Before Calibration)		Latency (Before Calibration)	Slope (After Calibration)		Latency (After Calibration)
		X	Y		X	Y	
1.	Flat	0	0	2002ms	0	0	0
		0	0	2002ms	0	0	0
		-1	0	2002ms	0	0	0
		0	0	2002ms	0	0	0
		0	1	2002ms	0	0	0
		0	0	2002ms	0	0	0
2.	Tilt to the Right	1	1	2002ms	1	0	1820ms
		2	0	2002ms	2	0	1820ms
		3	0	2002ms	3	0	1820ms
		4	1	2002ms	4	0	1820ms
		5	0	2002ms	5	0	1820ms
		6	0	2002ms	6	0	1820ms
3.	Tilt to the Left	-1	0	2002ms	-1	0	1820ms
		-2	0	2002ms	-2	0	1820ms
		-3	1	2002ms	-3	0	1820ms
		-4	0	2002ms	-4	0	1820ms
		-5	0	2002ms	-5	0	1820ms
		-6	0	2002ms	-6	0	1820ms
4.	Above (Y)	-1	2	2002ms	-1	1	1820ms
		0	3	2002ms	0	2	1820ms
		0	4	2002ms	0	3	1820ms
		0	5	2002ms	0	4	1820ms
		0	6	2002ms	0	5	1820ms
		0	7	2002ms	0	6	1820ms
5.	Downward (X)	2	-1	2002ms	1	0	1820ms
		3	0	2002ms	2	0	1820ms
		4	0	2002ms	3	0	1820ms
		5	0	2002ms	4	0	1820ms
		6	0	2002ms	5	0	1820ms
		7	0	2002ms	6	0	1820ms

Table 2 shows the difference in slope and latency values before and after calibration. Before calibration, there is noise when the tool is placed on a flat plane so that the cursor moves by itself. Latency before calibration looks faster than after calibration, which can make it difficult for users to use the tool.

To resolve the noise in this case, calibration is carried out with the results that can be seen in Table 2 that when the tool is placed on a flat plane, the X and Y values remain at (0,0) and do not change if it is not moved. This can happen because the program code adds a "limit" variable. The variable is used to set the maximum range of acceleration values generated by the MPU6050 sensor on the X and Y axes, which are then used to move the cursor. The main function of this parameter is to stabilize the cursor movement so that it is not too fast or excessive, so that the cursor control becomes smoother and more accurate. In the program code, the "limit" variable is applied using the "constrain()" function which limits the acceleration value within a certain range. After experimenting with several values, a value between -5 to 5 suits this case. This helps to reduce the effects of noise and small fluctuations that may be captured by the sensor, so that the cursor does not move due to unwanted changes in value. By applying the "limit" variable, the user has the flexibility to adjust the cursor sensitivity according to a typical mouse. In cases where high precision is required, the "limit" value should be reduced to reduce the movement sensitivity. The main advantages of using this parameter are ease of customization, improved accuracy and control, and reduced operational errors caused by uncontrolled cursor movement. Thus, the use of limits can effectively improve the user experience and functionality of tools that utilize the MPU6050 and ESP32 sensors for cursor control via Bluetooth.

After performing noise calibration on the ESP32 and MPU6050-based air mouse devices, latency calibration was next performed. Mouse movement latency calibration involves adjusting the delay time between the sensor data reading and the cursor movement response. In the program using the MPU6050 and ESP32 sensors, the latency is set with the delay() function, which delays the execution of the main loop. Initially, the program uses delay(5), which means there is a 5 millisecond delay between each iteration of the loop. When using a loop with delay(5), the sensitive cursor moves quickly, making it difficult for users to use. Changing the value of delay(5) to delay(15) will improve the latency of the system, the drawback is that it will make the cursor less responsive to changes in position detected by the sensor. Increasing the delay from 5 milliseconds to 15 milliseconds makes the cursor position update occur less frequently per second, which reduces the responsiveness of the cursor to user movements. This is also done to reduce noise and provide comfort and avoid too fast cursor movement, which can make it difficult for users to control the device. The result of the tool can be seen in Figure 11.

Figure 11. Results of the ESP32 and MPU6050 Based Air Mouse Tool

3. Conclusion

This research has successfully designed and developed an air mouse device using ESP32 microcontroller and MPU6050 sensor. The designed ESP32 and MPU6050-based air mouse is able to detect hand movements and transform them into cursor movements on the computer screen well. The use of MPU6050 sensor to detect acceleration and rotation provides good enough accuracy for cursor control purposes. Calibration of the noise and latency of the sensor is very important to improve the performance of the air mouse. By adjusting the boundary value and delay, the air mouse is able to reduce noise and improve user comfort in controlling the cursor. The use of BLE (Bluetooth Low Energy) technology enables stable and efficient communication between the air mouse and laptop/PC devices. This makes the device easy to integrate without the need for additional modules. The test results show that the device can function properly on a laptop/PC device, with a latency level that is still within the user's tolerance...

As a suggestion, for further development of ESP32 and MPU6050-based air mouse tools, we recommend adding components that are useful for scrolling like a mouse in general. Further research is expected to improve the accuracy and responsiveness of the air mouse such as the use of more efficient data processing methods or algorithms.

Reference

[1] Awati, R. (2023). Lithium polymer battery (LiPo). TechTarget. https://www.techtarget.com/whatis/definition/lithium-polymer-battery-LiPo
[2] Darmawan, B., Syauqy, D., Hannats, M., & Ichsan, H. (2019). View controller design using accelerometer sensor in bus simulator game as a means of user experience (UX) based on embedded system. Development of Information Technology and Computer Science, 3(2), 1883-1891.
[3] Eka, R. A. (2022). Thing View is ready to use. Diploma Thesis, Indonesia Digital Technology University, 34-41.
[4] Harmayani, Apdilah Dicky, Mapilindo, Oktopanda, & Hutahaean Jeperson. (2021). FullBook+Application+Computer-compressed.
[5] Jonas, D., Supriyono, I. A., & Junianto, H. (2022). Design of ESP32-based Motor Vehicle Theft Prevention System at PT Suwarna Dwipa Maju. Technomedia Journal, 7(2), 216-230. https://doi.org/10.33050/tmj.v7i2.1748

[6] Mahanin Tyas, U., Apri Buckhari, A., Information Technology Education Studies, P., & Technology and Vocational Education Studies, P. (2023). Implementation of Arduino Idea Application in Digital System Course. TEKNOS: Journal of Information Technology Education, 1(1), 1-9.

[7] Muliadi, Imran, A., & Rasul, M. (2020). Development of Smart Trash Can Using Esp32. Jurnal Media Elektrik, 17(2), 2721-9100. https://ojs.unm.ac.id/mediaelektrik/article/view/14193

[8] Musa, P. (2023). Book - May 2023 (Issue May).

[9] Pratiwi, H. I., & Handoko, P. (2019). Arduino Uno Application on Human Robot Module as a Mediation of Design, Narration and Simulation Teaching. Widyakala Journal, 6(2), 150. https://doi.org/10.36262/widyakala.v6i2.196

[10] Rozzak Ruwandi. (2022). Analysis of Angle and Angular Acceleration Using Mpu6050 Gyroscope Sensor for Fall Detection in Humans.

[11] Silmi Nurul Utami, S. G. (2021). No Title. KOMPAS.COM. https://www.kompas.com/skola/read/2021/06/24/120000369/flowchart--definisi-fungsi-jenis-dan-contohnya

[12] Sri Hartanto, A. R. (2023). ELECTRICAL ENERGY GENERATION DESIGN USING 20 THERMOELECTRIC TRANSDUCERS WITH 300W INVERTER. 11(1).

[13] Sulaeman, W., Alimudin, E., Sumardiono, A., Cilacap, P. N., Elekronika, T., & Cilacap, K. (2022). Locker security system using face detection. 03(02), 117-122.

[14] WK, M. R. N., Ansori, A. S. R., & Dinimaharawati, A. (2021). Development of Virtual Mouse Using Imu Mpu-9250 and Esp32 Sensors Using DMP (Digital Motion Processor) Algorithm. EProceedings ..., 8(5), 6429-6441.

[15] Zalukhu, A., Swingly, P., & Darma, D. (2023). Flowchart Learning Application Software. Journal of Technology, Information and Industry, 4(1), 61-70. https://ejurnal.istp.ac.id/index.php/jtii/article/view/351

[16] Riski, M. D. (2019). Designing an Automatic Light Tool in the Cargo Compartment of an Arduino-Based Airplane Using a Push Botton Switch as Learning at the Surabaya Aviation Polytechnic (already). Proceedings of the National Seminar on Aviation Technology Innovation (SNITP), 1-9.

[17] Razor, A. (2020). Arduino Jumper Cable: Definition, Functions, Types, and Prices. Available: https://www. aldyrazor. com/2020/04/kabel-jumper-arduino. Html.

[18] Husain, A. (2023). Gaming Mouse Selection Decision Support System Using Analytical Hierarchy Process and Profile Matching Methods (Doctoral dissertation, Gadjah Mada University).