Jurnal Polimesin
Department of Mechanical Engineering
State Polytechnic of Lhokseumawe
http://e-jurnal.pnl.ac.id/polimesin
e-ISSN : 2549-1999 No. : 3 Month : June
p-ISSN : 1693-5462 Volume : 23 Year : 2025

Processing dates: received on 2025-05-19, reviewed on 2025-06-19, accepted on 2025-06-25 and online availability on 2025-06-30

Design and performance evaluation of a real-time heart rate and vibration monitoring system with Arduino

Al Munawir[1,*], Zakaria Husen[2], Zuchra Ulfa[3]
[1]Department of Mechanical Engineering, University of Teuku Umar, Meulaboh 23681, Indonesia
[2]Department of Physics, University of Syiah Kuala, Banda Aceh 24415, Indonesia
[3]Unsyiah Labschool Senior High School, Aceh 23111, Indonesia
*Corresponding Author: almunawir@utu.ac.id

Abstract

The advancement of technology in the fields of healthcare and engineering has driven the demand for portable, energy-efficient, and user-friendly physiological and mechanical monitoring devices. Heart rate (BPM) is vital in the medical field, while vibration frequency is crucial in mechanical engineering for monitoring system stability. This study designs a Beats Per Minute (BPM) and vibration frequency monitoring device based on Arduino and ESP32. The system utilizes a Pulse Sensor for BPM and an MPU6050 sensor for vibration, with real-time data displayed on the Serial Monitor and an OLED screen. The method used is a quantitative experiment involving the design, assembly, programming, and testing of the device. BPM testing was conducted on three subjects with varying activity levels, while vibration testing was performed using different voltage levels on a vibration motor. The results show high BPM accuracy, with a maximum deviation of 2.7% compared to manual methods. Vibration measurements demonstrated signal stability with less than 10% error relative to the theoretical frequency. The compact, affordable, and user-friendly design makes this tool a practical solution for health monitoring in remote areas and as a project-based educational medium. The system is feasible for further development in portable medical and engineering applications.

Keywords:
Mechanical wave, pulse sensor, heart rate monitoring, ESP32 microcontroller, Arduino

1 Introduction

The advancement of technology in the healthcare sector has driven the need for portable, efficient, and user-friendly medical devices [1]. This need is becoming increasingly urgent, especially in remote areas or regions with limited access to adequate healthcare facilities. One of the essential physiological parameters that is the focus of diagnosis and patient condition monitoring is heart rate, measured in Beats Per Minute (BPM) [2], [3]. This parameter provides vital information about the condition of the cardiovascular system, serving as a foundation for medical decision-making, particularly in emergencies [4]. However, the reality in the field shows that not all healthcare centers, especially those located in rural, island, or disaster-affected areas, are equipped with sufficient medical devices to monitor heart rate in real time. Various challenges, including difficult-to-access geographic conditions, limited human resources, and a lack of stable electricity supply, pose significant obstacles to delivering optimal healthcare services [5]. Therefore, there is a need to develop a BPM monitoring device that is portable, energy-efficient, accurate, and user-friendly for healthcare workers and field volunteers.

Arduino is an open-source microcontroller platform that offers flexibility and low cost, making it highly potential for developing simple medical tools [6]. In recent years, particularly since 2024, the integration of the ESP32 as a high-connectivity microcontroller and the MPU6050 sensor, which serves as both an accelerometer and a gyroscope, has been widely utilized in various physiological monitoring applications, including heartbeat and vibration detection. [7], [8]. This system not only enables real-time monitoring but also reduces dependence on conventional medical devices, which are often expensive and non-portable.

Although this implementation is already quite widespread, most existing studies are still limited to laboratory testing or small-scale trials, and few have examined system performance in complex real-world contexts. In addition, many developed systems still use simple algorithmic approaches and do not emphasize design optimization or adaptation to constrained operational environments. This system not only provides real-time monitoring capabilities but also reduces reliance on conventional medical devices, which tend to be expensive and non-portable.

From a mechanical engineering perspective, the development of this system requires integration of mechanical, electronic, and ergonomic aspects. A compact and durable physical design, circuit efficiency, and a casing that supports high mobility are essential elements to ensure the tool can be optimally used in various field conditions [9]. The system is also designed to operate with limited power sources, such as batteries or power banks, making it well-suited for use in emergencies or locations without an electrical grid [10], [11]. The use of a digital interface to display BPM data provides convenience for medical personnel in quickly evaluating a patient's condition [12], [13]. With a relatively low production cost, this device has the potential to be replicated and widely distributed, supporting equitable access to healthcare services and promoting the implementation of appropriate technology in the medical field. Moreover, in the context of mechanical engineering education, the development of this tool provides students with hands-on experience in applying engineering principles, from mechanical design and electronic system integration to functional device testing. It also fosters project-based learning that encourages innovation and real-world problem-solving in society.

Based on this background, this study aims to design and evaluate a simple Arduino-based system for real-time heart rate detection (BPM). The research is expected to produce a functional, efficient, and applicable prototype as an alternative solution for monitoring heart conditions in areas with limited healthcare facilities.

2 Research methodology

This study employs a quantitative experimental approach to design, implement, and test a BPM and mechanical wave frequency measuring device based on Arduino [14]. The device is developed as a portable monitoring system that can be used in health emergencies as well as for measuring vibrations in mechanical systems. The research was implemented through several main stages: system design, hardware assembly, microcontroller programming, sensor simulation and calibration, device testing, and result analysis. The study utilizes the following devices and components:

1. Arduino UNO R3: Serves as the system's brain, responsible for reading signals from the heart rate and vibration sensors, processing the data, and displaying the results in BPM and Hz (frequency).
2. Pulse Sensor (heart rate sensor): Used to detect the user's pulse via an optical sensor that reads changes in blood volume in the capillary vessels of the finger.
3. MPU6050 Sensor: A combined accelerometer and gyroscope used to measure mechanical vibrations and motion. In this study, only the accelerometer (acceleration) feature is used.

4. 16x2 LCD or Serial Monitor: Functions to display BPM and vibration frequency readings in real-time.
5. Jumper wires and breadboard: Used to assemble a temporary circuit without the need for soldering.
6. Power supply (5V/USB from laptop): Supplies voltage to the system.
7. Vibration medium (DC motor or small machine surface): Used as a vibration source for testing the mechanical sensor.

The system is designed with two sensor input pathways. The first path receives heart rate data from the pulse sensor, which is connected to an analog pin on the Arduino and works by detecting voltage fluctuations caused by blood flow with each heartbeat. The second path receives acceleration data from the Z-axis of the MPU6050 sensor. This data is processed to determine the vibration cycle (period), which is then converted into frequency. The system is assembled on a breadboard to allow easy repairs and modifications [15]. All wires are neatly arranged to minimize signal interference and facilitate easy data reading. Programming is done using the Arduino IDE, where the device's operation logic is coded in C/C++. The main program is divided into two sections: data processing from the pulse sensor (heart rate) and data processing from the MPU6050 sensor (vibration). All wires are neatly arranged on the breadboard to minimize signal interference and facilitate troubleshooting. The systematic organization of wiring and component connections also aims to ensure stable and accurate data readings [16].
In the first section, the Arduino reads analog values from the pulse sensor, which detects voltage fluctuations resulting from changes in blood volume during each heartbeat. When the analog value exceeds a certain threshold, Arduino marks it as one heartbeat. The time between two beats is recorded and used to calculate the heart period (T) in seconds. From this period, BPM is calculated using Eq. (1), where T is the time between two heartbeats.

$$BPM = \frac{60}{T} \tag{1}$$

In the second section, the program reads data from the MPU6050 sensor every 100 milliseconds to record Z-axis acceleration, which is the dominant direction of vibration on the tested medium. Arduino detects acceleration peaks to determine the vibration oscillation period. This data is used to calculate the frequency (f) in Hertz (Hz) using Eq. (2), where T is the time difference between two consecutive acceleration peaks.

$$f = \frac{1}{T} \tag{2}$$

All calculated data, both BPM and vibration frequency, is displayed in real-time via the Serial Monitor on the Arduino IDE. If a 16x2 LCD module is used, the data can also be displayed directly on the screen, allowing researchers to observe changes in heartbeat and vibration during testing. The experiment is divided into two main parts: BPM measurement testing and mechanical vibration frequency measurement testing. Each phase is designed to evaluate the accuracy and reliability of the Arduino-based device under varying conditions.

For BPM measurement testing, the experiment is conducted on three different subjects representing various physical conditions. Each subject undergoes three test scenarios based on different body activities. The first condition involves sitting calmly for two minutes to describe a state of rest. The second involves walking for one minute followed by two minutes of stillness, simulating light activity. The third involves light jumping for one minute, followed by a two-minute rest to represent moderate activity. In each condition, the pulse sensor is placed on the subject's index finger and connected to the Arduino, which records BPM every five seconds for two minutes. For validation, Arduino measurements are compared with manual BPM readings using a stopwatch and, if available, a standard medical device.

For vibration frequency testing, the MPU6050 sensor is placed on a vibrating medium a small DC motor. The motor is operated at three different voltages: 3V, 5V, and 9V, to generate various vibration levels. Measurements are taken for one minute for each voltage level. Arduino reads Z-axis acceleration data from the MPU6050 sensor and detects vibration peaks (oscillations) to measure the vibration period. The time difference between two peaks is used to calculate the period (T), which is then used to determine the frequency (f). All measurement data, both BPM and vibration, are collected and further analyzed using Microsoft Excel software. Visualization graphs are created to ease interpretation and to assess the overall accuracy of the device's measurements [17].

The collected data is analyzed quantitatively to evaluate the performance of the Arduino-based device in measuring BPM and vibration frequency. For BPM measurement, data from the pulse sensor is compared to manually recorded BPM using a stopwatch or a medical device as control. The analysis starts by calculating the average BPM for each test condition and the standard deviation to assess measurement variability. Additionally, the percentage error between the sensor reading and the manual method is calculated to assess the device's accuracy. A measurement is considered successful if the error rate is less than 10% compared to the manual or reference measurement.

For vibration frequency measurement, data from the MPU6050 sensor is compared with theoretical frequency values, which can be calculated or obtained from other reference instruments. Using the basic frequency formula, the stability of the measured frequency is tested. Acceleration wave patterns are also graphed to analyze the stability of vibration signals and to determine whether the signal exhibits a clear periodic pattern. A consistent pattern indicates the sensor's ability to detect vibrations reliably, suggesting the device functions properly. The success criteria for the device include an error rate of less than 10% compared to reference instruments and the appearance of a periodic pattern in the signal graph, indicating reliability in vibration frequency measurement.

3 Results and discussion

3.1 Result

3.1.1 System design for heart rate monitoring

In this study, a heart rate and vibration monitoring system was designed using an ESP32 microcontroller that integrates several key components: a Pulse Sensor for heart rate detection, an MPU6050 sensor for vibration sensing, and an OLED display as the user interface. The circuit was carefully constructed to ensure accuracy and stability in signal acquisition and data display. The Pulse Sensor is connected to the analog input pin (A0) of the ESP32 to detect heart rate signals through fluctuations in blood volume. The MPU6050 sensor, which combines an accelerometer and a gyroscope, is connected via the I2C communication protocol (using the SDA and SCL pins on the ESP32) to capture mechanical vibrations or motion. An OLED display module is used to present real-time data visually, providing users with immediate feedback on the heart rate and vibration readings.

Additionally, a push button is included in the system as a manual control input, which can be programmed to start or reset measurements. The entire system is powered and controlled by the ESP32 microcontroller, which serves not only as the central processing unit but also as a communication bridge, sending data to the serial monitor and graphical interface on a computer for logging and analysis. The circuit diagram illustrating the overall system architecture is shown in Fig. 1.

Fig. 1 illustrates a complete diagram of the heart rate and vibration frequency monitoring system based on the ESP32 microcontroller. The system consists of three main components: a Pulse Sensor as the input for heart rate signals, an MPU6050 sensor for detecting mechanical vibrations, and an OLED display for outputting the measured data. The ESP32 serves as the central processing unit, with the analog pin A0 used to receive signals from the Pulse Sensor, which is connected through a limiting resistor (approximately 220 ohms) to prevent voltage spikes.

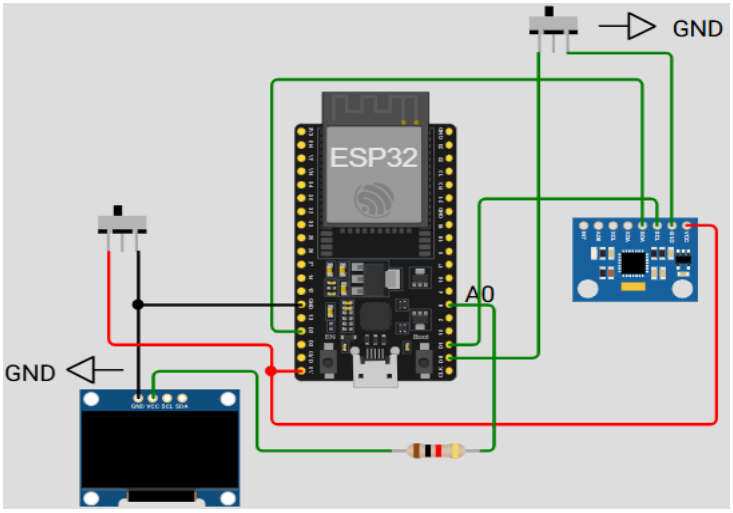

Fig. 1. Mechanical circuit design

The Pulse Sensor is powered by the 3.3V and GND pins, serving as a power supply and ground reference, respectively. The MPU6050 sensor, which integrates both an accelerometer and a gyroscope, is connected to the ESP32 via the I2C communication protocol, with its SDA and SCL lines connected to pins D21 and D22 on the ESP32. This sensor also receives power from the 3.3V and GND pins. The third component is a 0.96-inch OLED display, which also communicates via the I2C protocol in parallel with the MPU6050. The OLED displays the real-time measurements of BPM and vibration frequency, with its SDA and SCL lines sharing the same bus as those of the sensor. The circuit is designed to ensure that all components function synchronously, with neatly and efficiently arranged wiring to maintain signal stability and facilitate easy data reading. Adding labels for each connection, such as A0, SDA, SCL, VCC, and GND, along with specifying the resistor value, will further clarify the wiring diagram and help readers understand the system's workflow comprehensively.

Observations through the Arduino Serial Monitor revealed that the system successfully displayed real-time heart rate (BPM) values with stable data, ranging from 89 BPM to 99 BPM. The indicator “A Heartbeat Happened!” appeared each time a pulse was detected, indicating that the system was able to recognize heartbeats with high accuracy. In addition, the graphical display on the computer screen showed wave-like fluctuations, representing signal variations from both the heartbeat and vibrations detected by the sensor. The graph exhibited periodic shifts in values, indicating that the sensor responded responsively to changes in movement or heartbeat rhythm. The monitoring result display is shown in Figs. 2 and.3

Fig. 2. Measurement results on the monitor

Fig. 3. Heart rate measurement

Thus, the system has successfully responded and displayed data effectively, both through numerical output on the Serial Monitor and real-time graphical visualization. This design demonstrates great potential for the development of portable health monitoring devices as well as vibration sensor applications in engineering and education. The implementation of the lightweight and flexible ESP32, along with efficient sensor integration, makes this device a practical and economical solution for monitoring heart rate and other mechanical activities.

3.1.2 BPM measurement

BPM or Heart rate measurement is a vital indicator for monitoring a person's physiological condition, particularly in the context of cardiac health and physical fitness. In this study, BPM measurements were conducted using an Arduino-based Pulse Sensor, which was then compared to manual methods using a stopwatch and conventional pulse counting. The testing was carried out on three subjects under different physical activity conditions: resting state, after light activity, and after moderate activity. The purpose of this testing was to evaluate the accuracy of the Arduino Pulse Sensor in detecting heart rate signals, as well as to assess its consistency in responding to physiological changes that occur as the body transitions from rest to active states. The measurement results are summarized in Table 1.

When the body is in a relaxed state and not performing physical activity, the cardiovascular system is in its basal condition. At this stage, the measurement with the Pulse sensor showed a BPM of 70, while the manual measurement showed 72. The difference of only 2.7% is still within the tolerance range for non-invasive medical measurements. This indicates that the sensor is capable of detecting heartbeats with considerable accuracy under conditions with minimal signal interference, such as body movement or external vibrations. After the subject walked for one minute and then sat still for two minutes, the BPM recorded by the sensor increased to 95, while the manual method recorded 96 BPM. The difference of only 1% reflects the sensor’s rapid response to the increased heart rate caused by light activity. This also shows that heart rate recovery time after activity can be monitored in real-time with this sensor. Physical activity of higher intensity, such as light jumping for one minute, resulted in a significant increase in heart rate. The sensor recorded a BPM of 115, while the manual count showed 118 BPM. The 2.5% difference indicates that, despite a small discrepancy, the Pulse sensor can still provide results very close to the subject’s physiological reality.

Table 1. BPM measurement results

Activity Condition	BPM Sensor (Arduino)	BPM Manual (Stopwatch)	Difference (%)
Resting (2 minutes)	70	72	2.7%
Light Activity (walking 1 min, resting 2 min)	95	96	1%
Moderate Activity (light jumping 1 min, resting 2 min)	115	118	2.5%

3.1.3 Vibration frequency measurement

Measuring vibration frequency is an important aspect of monitoring the dynamic condition of mechanical systems, structures, or electronic components. In this study, the MPU6050 sensor integrated with Arduino was used to detect and measure the vibration frequency generated by a motor or vibrating element at three different voltage levels: 3V, 5V, and 9V. The MPU6050 sensor is a module that combines a 3-axis accelerometer and gyroscope, capable of detecting linear movement, rotation, and vibrations. The purpose of this test was to determine the sensor's accuracy in measuring vibration frequency and to assess how closely the measurement results align with the theoretical values calculated based on the input voltage. The test results are presented in Table 2.

Table 2. Vibration frequency measurement results (Hz)

Voltage (V)	Theoretical frequency (Hz)	Sensor frequency (Arduino)	Difference (%)
3V	3	3.05	1.7%
5V	5	5.02	0.4%
9V	9	9.1	1.1%

The vibration frequency measurements using the MPU6050 sensor demonstrate that the designed device is capable of operating with high accuracy across a range of various voltage levels. In the test at 3V, the expected theoretical frequency was 3 Hz, while the sensor reading showed 3.05 Hz, resulting in a difference of 1.7%. This slight difference indicates that the sensor can detect frequency quite well at low voltage. When the voltage was increased to 5V, the targeted theoretical frequency was 5 Hz, and the sensor recorded a frequency of 5.02 Hz. The difference between the sensor reading and the theoretical value was only 0.4%, making this voltage level the most accurate among the three scenarios. This shows that the MPU6050 sensor performs optimally at medium voltage.

In the test at the highest voltage, 9V, the frequency measured by the sensor was 9.1 Hz compared to the theoretical value of 9 Hz, resulting in a 1.1% difference. Although slightly higher, the measurement accuracy remains well within an excellent range. Overall, the Arduino-based frequency measurement device with the MPU6050 sensor demonstrates consistent and precise performance, with an average measurement error of less than 1.1%. This proves that the system can be reliably used to detect and monitor vibration frequencies in technical and industrial applications. The average measurement error is around 1%, which is very low and indicates that the MPU6050 sensor is capable of performing vibration measurements with high accuracy and good consistency.

3.2 Discussion

The device is designed to measure and display heart rate values (BPM) and frequency signals from sensors using the ESP32 microcontroller. The system comprises several main components: the ESP32 microcontroller, which serves as the system's brain; a pulse sensor for heart rate detection; an OLED display for direct data visualization; and a connection to a computer for data visualization via the Arduino Serial Plotter. The device schematic shows the ESP32 connected to the pulse sensor through the analog pin A0. This pin reads the analog signal from the sensor, which is then converted to a digital value by the ESP32's internal ADC. The digital value is used to detect the heartbeat in real time. Additionally, the OLED display displays the BPM directly on the device, eliminating the need for a computer connection and making it convenient for field monitoring.

The sensor connects to the ESP32 via three main wires: VCC (5V), GND (ground), and OUT (analog signal). The OLED display is connected to the ESP32's I2C pins, SDA and SCL, and powered by the 3.3V and GND pins. To ensure signal stability, a resistor is added to the sensor input line. With this configuration, the device accurately measures the analog pulse signal and converts it into digital BPM data. Signal stability is highly influenced by an efficient basic electronic design and the use of appropriate components. In addition to displaying BPM values on the OLED screen, the device is also configured to send data serially to a computer. On the Arduino IDE Serial Monitor, BPM values appear whenever a heartbeat is detected, accompanied by the notification "A Heartbeat Happened!" indicating successful signal detection. This data is also visualized in graph form using the Serial Plotter feature. The graph shows heart rate signal fluctuations between values of approximately 701 to 704.5, representing real-time heartbeat waveform signals. The graph display of the measurement results is shown in Fig. 4.

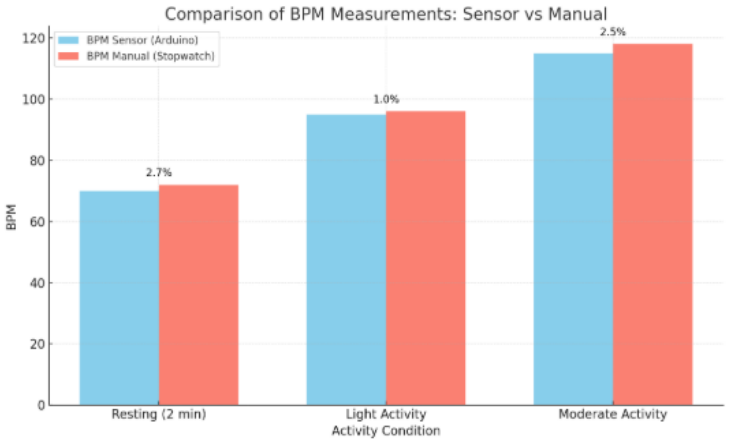

Fig. 4. Chart of BPM measurement

Fig. 4 presents a comparison of heart rate (BPM) measurements using an Arduino-based sensor (light blue) and manual measurement using a stopwatch (pink) across three physical activity conditions: resting (2 minutes), light activity (walking for 1 minute, resting for 2 minutes), and moderate activity (light jumping for 1 minute, resting for 2 minutes). This graph is included to provide a more precise visualization of how accurately the BPM sensor readings compare to the manual method. It allows readers to observe data consistency across different levels of physical activity directly and to understand the system's accuracy and reliability.

In the resting condition, the sensor recorded a BPM of 70, while the manual measurement showed 72 BPM, resulting in a 2.7% difference. During light activity, the sensor recorded 95 BPM, and the manual method recorded 96 BPM, with only a 1.0% difference. Under moderate activity, the sensor measured 115 BPM, while the manual reading was 118 BPM, yielding a 2.5% difference. The differences between sensor and manual results remained below 3% in all conditions, indicating a high level of accuracy for the Arduino-based pulse sensor in real-time heart rate detection, even during increased physical activity. This graph confirms that the developed monitoring system performs reliably across varying activity levels, detecting heart rate changes with fast response and high consistency. The light blue bars in the graph represent BPM data recorded by the Arduino sensor, while the pink bars show BPM measured manually using a stopwatch. Each pair of bars demonstrates a close match between the two methods, reinforcing the system's validity for health monitoring applications.

The graph displayed on the computer screen demonstrates the device's capability to represent periodic heartbeat signals, with a waveform resembling an oscillating analog signal. In addition to the BPM graph, the system was also tested to detect vibration frequencies under various voltage levels. The graph displays the vibration frequency measurements, which can be seen in Fig. 5.

Fig. 5 presents a comparison between the theoretical frequencies and the frequencies detected by the Arduino-based sensor at three different voltage levels: 3 V, 5 V, and 9 V. This graph aims to evaluate the accuracy of frequency measurement by the sensor (using an MPU6050 connected to Arduino) compared to the theoretically expected frequency values. At 3V, the theoretical frequency is 3 Hz, whereas the sensor recorded 3.05 Hz, resulting in a difference of 1.7%. At 5V, the theoretical and measured frequencies were very close, 5 Hz and 5.02 Hz, respectively, resulting in only a 0.4% difference. At 9V, the theoretical frequency of 9 Hz was compared to a sensor reading of 9.1 Hz, showing a 1.1% difference. All these discrepancies are below 2%, indicating the high

accuracy and consistent performance of the MPU6050 sensor across various voltage levels.

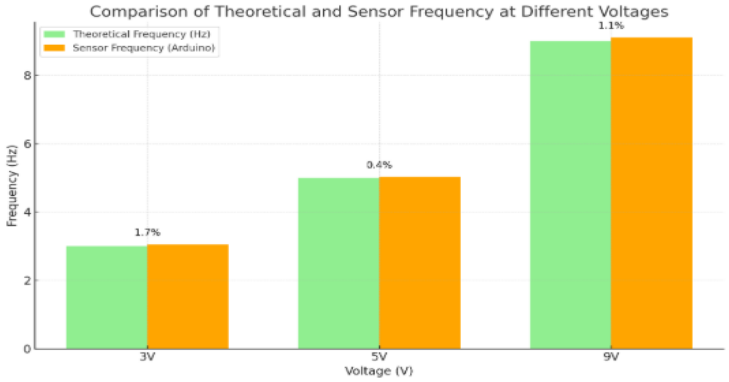

Fig. 5. Chart of the theoretical frequency and the frequency detected

The light green bars represent the theoretical frequencies (Hz), which serve as reference values that the system is expected to achieve based on calculations and circuit characteristics. The orange bars represent the frequencies measured by the Arduino sensor, which is real-world data recorded by the system. The close height of the bars in each pair shows that the sensor's results closely match the theoretical values. This graph confirms that the Arduino-based measurement system is capable of accurately and stably detecting frequency changes despite variations in input voltage. The waveform resembles an oscillating analog signal. Frequency changes can be detected with good stability, even in the presence of noise, which is common in analog sensor systems. The graph typically displays square or triangular waveforms, indicating periodic, repetitive pulses.

The visual representation of heart rate signals in waveform form is constructive in analyzing the stability of the heart rhythm. Analog signals are continuous and can have varying values within a specific range. For example, heart rate or oxygen level monitoring devices produce data in the form of graphs, which are highly suitable for visualizing gradual and real-time changes in data. This research confirms the successful integration of hardware and software in creating a simple yet effective monitoring system. As Stojkovic [16] explains, the ESP32 platform is well-suited for sensor data processing due to its precise ADC features and stable serial communication support. The pulse sensor can deliver reasonably accurate BPM measurements when properly configured and filtered. The addition of an OLED display is recognized as a practical solution for direct monitoring without the need for additional devices.

This device successfully combines the ESP32, pulse sensor, and OLED display into a functional but straightforward monitoring system. It has potential applications in health research, patient monitoring, and as an educational tool for IoT and biomedical sensor technology. Future developments could include adding Bluetooth or WiFi connectivity for remote monitoring, as suggested by Ianculescu [18], who notes that wireless monitoring systems provide high efficiency for real-time health supervision.

4 Conclusion

This research developed a real-time monitoring system integrating a Pulse Sensor and MPU6050 accelerometer with an ESP32 microcontroller for heart rate and vibration frequency analysis. The system displayed BPM values with high precision, maintaining a deviation range of 1–2.7% across different activity levels. Vibration frequency measurements also demonstrated reliable accuracy, with an average error of less than 1.1%. Data visualization through an OLED display and graphs enhances usability for both users and analysts. The low cost, compactness, and responsive performance of the system make it suitable for applications in personal health monitoring and sensor education. Additionally, the platform demonstrates a fast response to physiological and mechanical changes, ensuring reliability for real-time monitoring. Its simplicity and adaptability also support future development into IoT-based remote diagnostic tools and other engineering applications.

Acknowledgment

The authors express their deepest gratitude to the Faculty of Engineering, University of Teuku Umar, Aceh, and Unsyiah Labschool Senior High School, Aceh, for the support and facilitation provided during the implementation of this research.

References

[1] B. E. Putra and F. SpJP, "Integrasi teknologi dalam manajemen kardiologi: dari telemedicine hingga kecerdasan buatan". Stiletto Book, 2024.

[2] G. Prashar and U. Malairaman, "Heart Beat Monitoring System," 2024.

[3] M. L. Sahu, M. Atulkar, M. K. Ahirwal, and A. Ahamad, "Vital sign monitoring system for healthcare through IoT-based personal service application," Wirel. Pers. Commun., vol. 122, no. 1, pp. 129–156, 2022, doi: 10.1007/s11277-021-08892-4.

[4] B. T. Denton, "Frontiers of medical decision-making in the modern age of data analytics," IISE Trans., vol. 55, no. 1, pp. 94–105, 2023, doi: 10.1080/24725854.2022.2092918.

[5] M. Virag et al., "Bridging healthcare gaps through specialized mobile healthcare services to improve healthcare access and outcomes in rural Hungary," Sci. Rep.., vol. 15, no. 1, p. 12692, 2025, doi: 10.1038/s41598-025-97447-9.

[6] I. G. S. Widharma and L. F. Wiranata, "Mikrokontroler dan aplikasi". wawasan Ilmu, 2022.

[7] M. H. Setiawan, N. A. Sari, W. L. Prasetya, M. R. Feter, D. Saputra, and A. Ma'arif, "Implementation of Heart Rate system using AD8232 and Arduino microcontrollers," Signal Image Process. Lett., vol. 2, no. 1, pp. 36–44, 2020, doi: 10.31763/simple.v2i1.84.

[8] N. C. Joshi, "Fundamentals of electrocardiografia (ecg) with arduino uno". BFC Publications, 2022.

[9] M. Busari, "Design and development of the portable system," 2025.

[10] T. Kaur, J. Gambhir, and S. Kumar, "Arduino based solar powered battery charging system for rural SHS," in 2016 7th India International Conference on Power Electronics (IICPE), IEEE, 2016, pp. 1–5.

[11] K. C. Meje, L. Bokopane, K. Kusakana, and M. Siti, "Real-time power dispatch in a standalone hybrid multisource distributed energy system using an Arduino board," Energy Reports, vol. 7, pp. 479–486, 2021, doi: 10.1016/j.egyr.2021.08.016.

[12] F. M. Yassin, N. A. Sani, and S. N. Chin, "Analysis of heart rate and body temperature from the wireless monitoring system using arduino," in *Journal of Physics: Conference Series*, IOP Publishing, 2019, p. 12041. doi: 10.1088/1742-6596/1358/1/012041.

[13] M. Manenti and J. Ghebryal, "Design and control of a mechanical system to simulate in vivo cardiovascular flow," 2025.

[14] S. A. L. Siregar, A. Dani, and S. Aryza, "Implementation of low-cost programmable logic controller (PLC) based on arduino uno as a magnetic relay control system," J. Info Sains Inform. dan Sains, vol. 15, no. 01, pp. 25–36, 2025.

[15] R. Aalund and V. P. Paglioni, "Enhancing reliability in embedded systems hardware: A Literature survey," IEEE Access, 2025, doi: 10.1109/ACCESS.2025.3534138.

[16] N. Sreekanthaswamy et al., “Digital tools and methods,” in Enhancing School Counseling With Technology and Case Studies, IGI Global Scientific Publishing, 2025, pp. 25–48.

[17] A. Stojkovic, B. Nikolic, M. Zivkovic, and N. Bacanin, “Photovoltaic farm production forecasting: Modified metaheuristic optimized long short-term memory based networks approach,” IEEE Access, 2025.

[18] M. Ianculescu et al., “Enhancing connected health ecosystems through IoT-enabled monitoring technologies: A case study of the monit4healthy system,” Sensors, vol. 25, no. 7, p. 2292, 2025, doi: 10.3390/s25072292.