

# Real-Time Heart Rate and Oxygen Saturation Monitoring System Using Arduino with Blynk Integration

BANDI JANAKI RAM<sup>1</sup>, DUDI PRASAD<sup>2</sup>, YATHAM BUELA<sup>3</sup>, DALLI MRUDULA<sup>4</sup>, KALAPALA BHOGESWARI<sup>5</sup>, S.V.HEMANTH KUMAR<sup>6</sup>, MANDALOGA JYOTHI<sup>7</sup>,

Dr DONTABHAKTUNI JAYAKUMAR<sup>8</sup>

<sup>1,2,3,4,5,6</sup> *Electronics and communication engineering, Lingayas Institute Of Management and Technology, Vijayawada, Andhra Pradesh.*

<sup>7,8</sup> *Assistant Professor, Electronics and communication engineering, Lingayas Institute Of Management and Technology, Vijayawada, Andhra Pradesh.*

\*\*\*

**Abstract** - In recent years, the demand for affordable and portable healthcare monitoring systems has increased significantly, driven by the need for continuous health assessment and early detection of medical conditions. This paper presents the design and implementation of a real-time heart rate and blood oxygen saturation (SpO<sub>2</sub>) monitoring system using an Arduino microcontroller and the MAX30102 sensor. The system is based on the principle of photoplethysmography (PPG), which enables non-invasive measurement of physiological parameters by detecting variations in blood volume through light absorption techniques.

The proposed system integrates the MAX30102 sensor with an Arduino board to acquire and process pulse signals efficiently. The measured data is displayed locally using an LCD module and remotely transmitted via a Wi-Fi-enabled module to the Blynk mobile application, enabling real-time monitoring through smartphones. The use of IoT technology enhances accessibility, allowing users to track their health parameters from any location. The system is designed to be cost-effective, portable, and user-friendly, making it suitable for students, personal healthcare, and basic clinical applications.

Experimental results demonstrate that the system provides reliable and consistent measurements of heart rate and SpO<sub>2</sub> for multiple subjects under normal conditions. Although minor variations may occur due to environmental factors and sensor placement, the overall performance is satisfactory for general health monitoring purposes. The proposed system can be further extended by integrating additional biomedical sensors, cloud-based data storage, and advanced analytics for improved healthcare support.

In conclusion, the developed system offers a simple yet efficient solution for real-time health monitoring, combining embedded systems and IoT technologies to support modern healthcare needs and promote proactive health management.

**Key Words:** Heart Rate Monitoring, SpO<sub>2</sub> Measurement, MAX30102 Sensor, Arduino, Photoplethysmography (PPG), Internet of Things (IoT), Blynk Application, Biomedical Sensors, Real-Time Monitoring, Embedded Systems

## 1.INTRODUCTION

The ability to transmit patient information to a doctor or other healthcare practitioner continuously as well as in an emergency

is what makes remote monitoring such a successful technique of providing urgent care. Remote patient monitoring will transform healthcare in general. In addition to work, home, and leisure pursuits. We can regularly check patients thanks to these modern technology, replacing the requirement for

This project proposes a heart rate monitoring detection system using IoT. Nowadays treatment of most of the heart-related diseases requires continuous as well as long term monitoring. IoT is very useful in this aspect as it replaces the conventional monitoring systems with a more efficient scheme, by providing critical 2 information regarding the condition of the patient accessible by the doctor. In addition, the nurses or the duty doctor available at the hospital can monitor the heart rate of the patient in the serial monitor through the real-time monitoring system.

The system consists of multiple sensors that are placed on the body of the patient, which continuously monitors the heart rate and other vital parameters. The collected data is then transmitted to the cloud-based platform and store the information on the blink app for further analysis and monitoring. The system is designed to detect any abnormal changes in the heart rate and other vital parameters and provide immediate alerts to the healthcare professionals or the concerned family members.

Overall, the IoT based Heart Attack Detection and Heart Rate Monitoring System can play a crucial role in preventing heart-related diseases and improving the quality of life for patients suffering from cardiovascular diseases.

The literature on heart rate monitoring systems highlights significant advancements in the development of efficient, real-time, and portable healthcare solutions using sensors and Internet of Things (IoT) technologies. Over the years, researchers have explored various techniques such as photoplethysmography (PPG) and electrocardiography (ECG) for accurate detection of heart rate, with integration into microcontroller-based systems and mobile applications to enable continuous monitoring and analysis. IoT-based health monitoring systems have gained considerable attention due to their ability to provide remote access to patient data, thereby reducing the need for frequent hospital visits and supporting early diagnosis of cardiovascular conditions. For instance, Prachi et al. (2021) developed an IoT-based patient monitoring system focused on ambulance services, where real-time transmission of vital parameters to hospitals helps doctors prepare treatment in advance and improve emergency response. Similarly, Po-Wei Huang et al. (2020) proposed a non-contact

heart rate monitoring system using remote photoplethysmography (rPPG), which utilizes camera-based detection of subtle facial color variations to estimate heart rate, offering a comfortable and contactless solution particularly useful in driver monitoring systems. Furthermore, R. Priyanka and M. Reji (2019) introduced an IoT-based health monitoring system integrated with the Blynk application, enabling real-time visualization, storage, and alert generation for parameters such as heart rate and temperature, thereby improving accessibility and efficiency in personal healthcare. In addition, Souza et al. (2021) proposed adaptive linear prediction filtering techniques to reduce noise and motion artifacts in ECG and PPG signals, significantly improving peak detection accuracy and ensuring reliable measurements in real-time applications. Earlier work by Ajit Kumar Patro and Bandana Mallick (2016) demonstrated a simple and cost-effective heart rate monitoring system using a fingertip sensor with Arduino and Processing software, which provided real-time visualization of pulse waveforms and served as a foundation for modern embedded health monitoring systems. Collectively, these studies indicate that the integration of IoT, wireless communication, and advanced signal processing techniques has significantly enhanced the performance, reliability, and usability of heart rate monitoring systems, making them highly suitable for both medical and personal healthcare applications.

The proposed system typically consists of a microcontroller (such as Arduino Uno), a pulse sensor (like MAX30102), and a Wi-Fi module (such as ESP8266). The sensor detects the pulse signal from the fingertip, converts it into digital data, and sends it to the microcontroller for processing. The processed data is then displayed on a screen and simultaneously uploaded to an IoT platform for remote monitoring. The contributions of this paper are multifaceted, addressing both technological innovation and practical healthcare applications. Firstly, the paper introduces a cost-effective and portable heart rate monitoring system using Arduino Uno and a fingertip sensor, providing an affordable alternative to existing commercial devices. By leveraging the simplicity and versatility of the Arduino platform, the system is designed to be easily integrated into existing IoT ecosystems. In this project, an intelligent monitoring system is proposed; it is based on cutting-edge technology such as the Internet of Things (IoT). The goal of the Internet of Things (IoT) is to have self-reporting technology that can gather critical information in real time without human intervention, enhancing productivity. This will help prevent congestion and at the long run prevent delay in the hospital. It will also help doctors to give appointment based on empirical evidence. Medical personnel having a real-time information of their patients and also the system categorizing their conditions based on critical and non-critical cases enables the patient book an appointment with the doctor if it's a critical case. Booking an appointment with the doctor who already has a first-hand information about the patient helps prevent delay, as the patient comes to the hospital to see the doctor directly.

## Literature

The literature on heart rate monitoring systems highlights significant advancements in the development of efficient, real-time, and portable healthcare solutions using sensors and Internet of Things (IoT) technologies. Over the years, researchers have explored various techniques such as photoplethysmography (PPG) and electrocardiography (ECG) for accurate detection of heart rate, with integration into

microcontroller-based systems and mobile applications to enable continuous monitoring and analysis. IoT-based health monitoring systems have gained considerable attention due to their ability to provide remote access to patient data, thereby reducing the need for frequent hospital visits and supporting early diagnosis of cardiovascular conditions. For instance, Prachi et al. (2021) developed an IoT-based patient monitoring system focused on ambulance services, where real-time transmission of vital parameters to hospitals helps doctors prepare treatment in advance and improve emergency response. Similarly, Po-Wei Huang et al. (2020) proposed a non-contact heart rate monitoring system using remote photoplethysmography (rPPG), which utilizes camera-based detection of subtle facial color variations to estimate heart rate, offering a comfortable and contactless solution particularly useful in driver monitoring systems. Furthermore, R. Priyanka and M. Reji (2019) introduced an IoT-based health monitoring system integrated with the Blynk application, enabling real-time visualization, storage, and alert generation for parameters such as heart rate and temperature, thereby improving accessibility and efficiency in personal healthcare. In addition, Souza et al. (2021) proposed adaptive linear prediction filtering techniques to reduce noise and motion artifacts in ECG and PPG signals, significantly improving peak detection accuracy and ensuring reliable measurements in real-time applications. Earlier work by Ajit Kumar Patro and Bandana Mallick (2016) demonstrated a simple and cost-effective heart rate monitoring system using a fingertip sensor with Arduino and Processing software, which provided real-time visualization of pulse waveforms and served as a foundation for modern embedded health monitoring systems. Collectively, these studies indicate that the integration of IoT, wireless communication, and advanced signal processing techniques has significantly enhanced the performance, reliability, and usability of heart rate monitoring systems, making them highly suitable for both medical and personal healthcare applications.

## 2. OBJECTIVES OF THE SYSTEM

The primary objective of this system is to develop an efficient and user-friendly heart rate monitoring solution that enables real-time observation, analysis, and management of an individual's cardiovascular condition. One of the key objectives is real-time heart rate monitoring, which allows continuous tracking of the user's pulse without delay. This is essential in modern healthcare, as it provides immediate and accurate information about heart activity, helping individuals stay aware of their health status. The system utilizes a heart rate sensor to detect pulse signals, which are then processed by a microcontroller to calculate the heart rate in beats per minute (BPM) and transmitted wirelessly to a mobile application.

Another important objective is to display heart rate data on a mobile platform such as the Blynk application. The processed data is transmitted through wireless communication technologies like Wi-Fi or Bluetooth and presented in an easy-to-understand format, including numerical values, graphs, or gauge indicators. This user-friendly interface allows individuals, even without technical knowledge, to monitor their heart rate conveniently in real time. Additionally, the application can store previous data, enabling users to observe patterns and trends in their heart activity over time.

The system also focuses on detecting abnormal heart rate conditions and generating alerts. Predefined threshold values are set for normal heart rate ranges, and if the measured value exceeds or falls below these limits, the system immediately

triggers a notification through the mobile application. This feature is particularly beneficial for elderly individuals and patients with cardiac conditions, as it ensures timely intervention and reduces the risk of severe health complications.

Furthermore, the system enables continuous tracking and analysis of heart rate data. By storing data over time and presenting it in graphical formats, users can analyze trends related to physical activity, stress, or lifestyle habits. This long-term monitoring capability supports better health management and assists healthcare professionals in making accurate diagnoses.

### 3. SCOPE OF THE SYSTEM

The scope of this system lies in providing an effective, low-cost, and portable solution for monitoring heart rate in real time. The system is designed to accurately measure pulse rate using a heart rate sensor, which detects changes in blood flow through optical sensing techniques. These variations are converted into electrical signals and processed by a microcontroller to calculate BPM. Accuracy is ensured through proper sensor placement and signal processing techniques that minimize noise and external disturbances, enabling continuous and reliable monitoring.



Fig. Block diagram of system

Another significant aspect of the system is the ability to display heart rate data on a mobile application such as the Blynk app. The system wirelessly transmits processed data to the application, where it is displayed in an intuitive format. This enhances accessibility and allows users to monitor their health anytime and anywhere without requiring specialized medical equipment.

Wireless data transmission plays a crucial role in improving system flexibility and usability. By eliminating the need for physical connections, the system ensures seamless communication between hardware components and the mobile device. This feature enhances portability and allows users to move freely while their heart rate is being monitored. It also supports future scalability, making it easier to integrate with cloud platforms and other smart devices.

The system also includes an emergency alert mechanism that sends instant notifications when abnormal heart rate conditions are detected. These alerts can be delivered as push notifications, warning messages, or alarms through the mobile application, ensuring immediate awareness and response. This feature is particularly useful in critical situations and can be extended to notify caregivers or family members for additional safety.

In addition, the system supports the storage and analysis of historical heart rate data, which helps users track their health trends over time. By presenting past data in graphical formats, users can identify patterns and make informed decisions regarding their lifestyle and health management. This

functionality also assists healthcare professionals in diagnosing potential issues more effectively.

Finally, the system contributes to preventive healthcare and health awareness by providing users with continuous insights into their heart activity. By understanding how factors such as exercise, stress, and sleep affect heart rate, users can adopt healthier habits and take proactive steps toward maintaining overall well-being.

### Context Diagram of the System

The Arduino-based heart rate monitoring system is designed to measure and monitor a person's heart rate and blood oxygen level (SpO<sub>2</sub>) in real time. The system utilizes a MAX30102 sensor to detect pulse signals from the user, which are then processed by the Arduino microcontroller. The processed data is displayed locally and also transmitted wirelessly to a mobile application using a Wi-Fi module. This system is highly beneficial for personal healthcare, remote monitoring, and preventive medical applications, as it enables continuous observation of vital parameters and provides alerts in case of abnormal conditions.

### Power Supply

The power supply is a fundamental component of the health monitoring system, as it provides the necessary electrical energy to all the modules, including the Arduino, MAX30102 sensor, Wi-Fi module (ESP8266), display unit, buzzer, and LED indicators. It ensures that each component operates at the required voltage level for proper functioning. The system can be powered using a battery, USB connection, or an external power adapter. Maintaining a stable power supply is crucial, as voltage fluctuations can affect sensor accuracy and overall system performance. Thus, the power supply acts as the backbone of the system, enabling reliable and uninterrupted operation.

### MAX30102 Sensor (Heart Rate and SpO<sub>2</sub> Measurement)

The MAX30102 sensor is a vital component of the system, responsible for measuring heart rate and blood oxygen saturation levels. It is an optical sensor that operates by emitting red and infrared light into the skin, typically through a fingertip, and measuring the amount of light absorbed by the blood. Based on these variations, the sensor calculates both the pulse rate and SpO<sub>2</sub> levels. Due to its compact size, low power consumption, and high accuracy, it is widely used in wearable and portable healthcare devices. The sensor communicates with the Arduino using I2C protocol, transmitting real-time data for further processing. Overall, it provides a simple, non-invasive, and efficient method for monitoring vital health parameters.

### I2C Communication

I2C (Inter-Integrated Circuit) communication is used in this system to enable data transfer between the MAX30102 sensor and the Arduino microcontroller. It is a two-wire communication protocol consisting of the Serial Data Line (SDA) and Serial Clock Line (SCL). In this setup, the Arduino functions as the master device, while the sensor acts as the slave device. I2C communication is preferred because it minimizes wiring complexity and allows multiple devices to communicate over the same bus. It ensures reliable and synchronized data transmission, making it suitable for real-time health monitoring applications.

### Arduino (Processing and BPM Calculation)

The Arduino serves as the central processing unit of the system and plays a critical role in data handling and analysis. It receives raw signals from the MAX30102 sensor through I2C communication and processes them using embedded algorithms. These algorithms filter noise, detect pulse peaks, and calculate heart rate in beats per minute (BPM), as well as analyze SpO<sub>2</sub> levels. The Arduino also compares the measured values with predefined normal ranges to identify any abnormal conditions. After processing, the data is sent to the display unit for local monitoring and to the Wi-Fi module for remote access. In case of abnormal readings, the Arduino activates alert mechanisms such as a buzzer and LED to notify the user.

### Wi-Fi Module (ESP8266)

The ESP8266 Wi-Fi module is responsible for enabling wireless communication in the system. It is connected to the Arduino through UART communication and allows the system to connect to a wireless network. Once connected, it transmits the processed data, including heart rate and SpO<sub>2</sub> values, to an online platform or mobile application such as the Blynk app. This feature enables users to monitor their health parameters remotely from anywhere. The ESP8266 is widely preferred due to its low cost, low power consumption, and ease of integration with microcontrollers, making it ideal for IoT-based healthcare applications.

### Software and Hardware Requirements

#### Power Supply

The power supply is a fundamental component of the health monitoring system, as it provides the necessary electrical energy to all system modules, including the Arduino, MAX30102 sensor, Wi-Fi module (ESP8266), display unit, buzzer, and LED indicators. It ensures that each component receives a stable and appropriate voltage for efficient operation. The system can be powered using a battery, USB cable, or an external adapter depending on the application requirements. Maintaining a stable power source is crucial because voltage fluctuations can negatively affect sensor accuracy and system performance. Therefore, the power supply acts as the backbone of the system, ensuring continuous and reliable operation.

### MAX30102 Sensor (Heart Rate and SpO<sub>2</sub> Measurement)

The MAX30102 sensor is a key element of the system used to measure heart rate and blood oxygen saturation levels. It operates based on optical sensing technology by emitting red and infrared light into the skin, typically through a fingertip, and measuring the amount of light absorbed by the blood. These variations are used to calculate the pulse rate and SpO<sub>2</sub> levels. The sensor is compact, energy-efficient, and highly accurate, making it suitable for wearable and portable healthcare devices. It communicates with the Arduino using I2C protocol, sending real-time data for further processing. Overall, it provides a non-invasive and efficient method for monitoring vital health parameters.

### I2C Communication

I2C (Inter-Integrated Circuit) communication is used to transfer data between the MAX30102 sensor and the Arduino microcontroller. It is a two-wire communication protocol consisting of the Serial Data Line (SDA) and Serial Clock Line (SCL). In this system, the Arduino acts as the master device, while the sensor functions as the slave device. This communication method reduces wiring complexity and allows multiple devices to share the same communication bus. It

ensures reliable, synchronized, and efficient data transfer, making it highly suitable for real-time health monitoring applications.

### Arduino (Processing and BPM Calculation)

The Arduino serves as the central processing unit of the system and performs all major computational tasks. It receives raw data from the MAX30102 sensor via I2C communication and processes it using programmed algorithms. These algorithms filter noise, detect pulse signals, and calculate the heart rate in beats per minute (BPM), along with analyzing SpO<sub>2</sub> values. The Arduino also compares the measured values with predefined normal ranges to identify abnormal conditions. After processing, the data is displayed locally and transmitted to the Wi-Fi module for remote monitoring. In case of abnormal readings, the Arduino activates alert mechanisms such as a buzzer and LED.

### Wi-Fi Module (ESP8266)

The ESP8266 Wi-Fi module enables wireless communication in the system and plays a crucial role in IoT-based monitoring. It is connected to the Arduino through UART communication and is responsible for transmitting processed data, such as heart rate and SpO<sub>2</sub> values, to the internet. This allows the system to connect to a mobile application like the Blynk app, enabling users to monitor their health parameters remotely. The ESP8266 is widely used due to its low cost, low power consumption, and ease of integration with microcontrollers.

### Blynk App (Mobile Monitoring)

The Blynk application is used as a user interface for real-time monitoring and visualization of health data. It receives data transmitted from the Arduino via the ESP8266 module and displays it in an easy-to-understand format such as numerical values, graphs, or gauges. This enables users to monitor their heart rate and SpO<sub>2</sub> levels anytime and from anywhere using a smartphone. Additionally, the app can generate alerts and notifications when abnormal values are detected, ensuring timely action. Its simple interface makes it suitable even for non-technical users.

### Buzzer and LED Alert System

The buzzer and LED serve as alert mechanisms to notify users in case of abnormal health conditions. When the system detects heart rate or SpO<sub>2</sub> values outside the normal range, the Arduino activates the buzzer to produce an audible alert and turns on the LED for visual indication. This dual-alert system ensures that the user is immediately informed, even if they are not actively monitoring the display or mobile application. It is particularly useful in emergency situations and enhances the safety and reliability of the system.

### Display Unit

The display unit provides a direct and real-time visualization of health parameters such as heart rate (BPM) and SpO<sub>2</sub> levels. It is connected to the Arduino, which sends processed data for immediate viewing. Displays such as LCD or OLED are commonly used due to their simplicity, low power consumption, and ease of interfacing. This feature allows users to check their health data instantly without relying on mobile devices or internet connectivity.

**Working of the System**

The overall working of the health monitoring system is simple and efficient. Initially, the MAX30102 sensor collects heart rate and SpO<sub>2</sub> data when the user places a finger on it. This data is transmitted to the Arduino using I2C communication. The Arduino processes the signals, filters noise, and calculates the required values. The processed data is then displayed on the local display unit for immediate observation. Simultaneously, the Arduino sends the data to the ESP8266 Wi-Fi module through UART communication, which uploads it to the internet. The Blynk mobile application retrieves and displays this data in real time, enabling remote monitoring. If any abnormal condition is detected, the system activates the buzzer and LED to alert the user. In this way, the system ensures continuous monitoring and provides both local and remote feedback.

**Algorithms and flow charts:**

The algorithm and flow of the proposed health monitoring system describe the step-by-step operation involved in measuring, processing, and displaying heart rate (BPM) and blood oxygen level (SpO<sub>2</sub>). The process begins with system initialization, followed by data acquisition, signal processing, and output display, ensuring continuous and real-time monitoring.

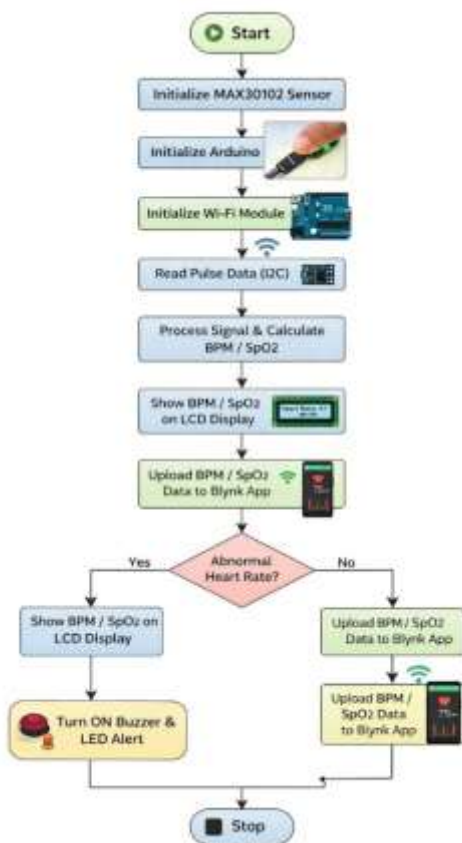


Fig 3 flow charts

**Start and System Initialization**

The operation of the system begins with the start condition, where all components are initialized to ensure proper functioning. Initially, the MAX30102 sensor is activated to enable heart rate and SpO<sub>2</sub> measurement. Then, the Arduino microcontroller is initialized to control the overall system operations, including configuring input/output pins and communication protocols. Following this, the Wi-Fi module is

initialized to establish wireless connectivity for data transmission. Once all components are properly set up, the system proceeds to acquire pulse data from the sensor through I2C communication.

**Initialization of MAX30102 Sensor**

The initialization of the MAX30102 sensor is a critical step that prepares it for accurate data acquisition. During this stage, the Arduino configures the sensor through the I2C interface by setting appropriate registers, enabling LED pulses, selecting the sampling rate, and defining the operating mode for heart rate and SpO<sub>2</sub> detection. The sensor’s red and infrared LEDs, along with the photodetector, are activated to capture physiological signals from the user’s fingertip. Proper initialization ensures stable signal acquisition and reduces noise, thereby improving measurement accuracy.

**Initialization of Arduino**

The Arduino initialization process sets up the microcontroller to manage the entire system. In this stage, all required libraries are loaded, and the input/output pins connected to various components such as the sensor, display, buzzer, LED, and Wi-Fi module are configured. Communication protocols, including I2C for the sensor and serial communication for debugging and data transfer, are also established. This step ensures that the Arduino is fully prepared to receive, process, and transmit data efficiently.

**Initialization of Wi-Fi Module**

The Wi-Fi module (ESP8266) is initialized to enable wireless communication between the system and external devices. This involves supplying the required operating voltage (typically 3.3V), establishing proper TX and RX connections with the Arduino, and configuring network credentials such as SSID and password. Once connected to the wireless network, the module allows the system to transmit data to cloud platforms or mobile applications for remote monitoring.

**Reading Pulse Data (I2C Communication)**

After initialization, the system begins collecting raw pulse data from the MAX30102 sensor using I2C communication. The sensor continuously detects variations in light absorption caused by blood flow and converts these variations into digital signals. These signals are transmitted to the Arduino in real time. The acquired data at this stage may contain noise and requires further processing to obtain accurate results.

**Signal Processing and BPM/SpO<sub>2</sub> Calculation**

The Arduino processes the raw signals using signal processing techniques such as filtering and peak detection. Noise and unwanted disturbances are removed to obtain a clean waveform. The system then identifies peaks in the signal, which correspond to heartbeats. By calculating the time interval between consecutive peaks, the heart rate is determined in terms of beats per minute (BPM). Similarly, SpO<sub>2</sub> values are calculated based on the ratio of red and infrared light absorption. This step ensures accurate and reliable measurement of vital parameters.

**Display of BPM and SpO<sub>2</sub> on LCD**

Once the values are calculated, the Arduino updates the LCD display with the latest readings. The display typically shows heart rate on one line and SpO<sub>2</sub> level on another, allowing the user to view real-time health data directly from the device. This

feature provides immediate feedback without requiring a mobile device or internet connection.

#### Data Upload to Blynk Application

In addition to local display, the system transmits the processed data to the Blynk mobile application using the Wi-Fi module. With the help of the Blynk library and authentication token, the Arduino sends data to virtual pins, where it is displayed using widgets such as graphs, gauges, or numeric indicators. This enables users to monitor their health parameters remotely from anywhere in real time.

#### Continuous Monitoring Loop

The entire process operates in a continuous loop, where data is repeatedly collected, processed, and displayed. If any abnormal condition is detected, such as heart rate or SpO<sub>2</sub> values outside the normal range, the system triggers alerts through a buzzer, LED, and mobile notifications. This continuous operation ensures real-time monitoring, early detection of health issues, and improved user safety.

### 4. DESIGN, IMPLEMENTATION, RESULTS AND DISCUSSION

The design of a heart rate monitoring system using Arduino presents a simple, efficient, and cost-effective solution for personal healthcare monitoring. This system is developed to continuously measure a person's heart rate and provide real-time data, enabling individuals to easily track their cardiovascular health. The system mainly consists of an Arduino microcontroller, a pulse sensor for detecting heartbeats, a display unit such as an LCD or serial monitor, and optional communication modules like Bluetooth or Wi-Fi for integration with mobile applications. The compact nature and affordability of the system make it highly suitable for academic projects and real-world applications.

The working principle of the system is based on detecting changes in blood volume using a pulse sensor, typically placed on the fingertip. The sensor operates using light absorption techniques to sense variations in blood flow. These variations are converted into electrical signals and transmitted to the Arduino for processing. The microcontroller then calculates the heart rate in terms of beats per minute (BPM) and displays the results on a screen or transmits them to a connected mobile application for continuous monitoring. This real-time processing capability enhances the usability of the system in practical healthcare scenarios.

The system offers several important features such as real-time heart rate monitoring, portability, ease of use, and low cost. These characteristics make it highly suitable for students, researchers, and personal healthcare applications. Furthermore, the system can be extended by integrating Internet of Things (IoT) technologies to enable remote health monitoring. Overall, the Arduino-based heart rate monitoring system serves as a practical and beginner-friendly solution that promotes awareness of heart health and assists in the early detection of abnormal conditions.

The MAX30102 sensor plays a vital role in the system by enabling accurate measurement of heart rate and blood oxygen saturation (SpO<sub>2</sub>). Monitoring heart rate is an essential aspect of healthcare, fitness tracking, and biomedical research. With advancements in embedded systems and wearable technology, compact and low-power sensors such as the MAX30102 have gained significant importance. This sensor is an integrated pulse oximeter and heart-rate module developed by Maxim

Integrated (now part of Analog Devices) and is widely used in portable and wearable devices.

The MAX30102 is an optical sensor that integrates red and infrared LEDs, a photodetector, and low-noise analog signal processing components into a single module. It is capable of measuring both heart rate and oxygen saturation levels. The sensor processes optical signals internally and communicates the data digitally to a microcontroller using the I<sup>2</sup>C protocol, making it highly suitable for embedded and IoT-based applications.

The working principle of the MAX30102 is based on photoplethysmography (PPG), a non-invasive optical technique. In this method, the sensor emits red and infrared light into the skin, usually through a fingertip. The amount of light absorbed varies depending on the oxygen level in the blood. Oxygenated blood absorbs more infrared light, while deoxygenated blood absorbs more red light. The photodetector captures the reflected light, and the variations caused by blood flow are analyzed to determine heart rate and SpO<sub>2</sub> levels. This technique provides reliable and continuous monitoring of physiological parameters.

The MAX30102 sensor includes several important features that enhance its performance. It integrates LEDs and a photodetector, reducing the need for external components. The sensor operates at low power, making it ideal for battery-powered devices. It supports I<sup>2</sup>C communication, allowing easy interfacing with microcontrollers such as Arduino. Additionally, it provides high sensitivity, programmable sampling rates, adjustable LED current, and an internal temperature sensor for improved measurement accuracy.

The pin configuration of the MAX30102 module includes VCC, GND, SDA, SCL, and INT pins. The VCC pin supplies power to the sensor, typically in the range of 3.3V to 5V, while the GND pin provides the common ground. The SDA and SCL pins are used for I<sup>2</sup>C communication with the microcontroller, and the INT pin is used to signal the availability of new data. Additional pins such as IRD and RD are associated with infrared and red LEDs, respectively, and are generally not used unless manual LED control is required. This pin configuration ensures easy integration with embedded systems.

The pulse sensor used in the system is compact, lightweight, and designed for continuous contact with human skin. It typically has a small circular shape, making it suitable for wearable applications. The sensor operates within a voltage range of 3V to 5.5V and consumes very low power, making it ideal for battery-operated devices. It generates an analog output signal that varies with blood flow, which is converted into digital values using the Arduino's 10-bit analog-to-digital converter (ADC) for accurate measurement.

The working of the pulse sensor is also based on the principle of photoplethysmography (PPG). It consists of a light-emitting diode (LED) and a photodetector. The LED emits light onto the skin, and the amount of reflected light changes with variations in blood volume during each heartbeat. These variations are captured by the photodetector and converted into electrical signals. The Arduino processes these signals to identify peaks corresponding to heartbeats and calculates the heart rate in BPM. This simple mechanism makes the sensor effective for health monitoring and fitness applications.

The circuit connections between the pulse sensor and Arduino Uno are simple and require only three wires. The power pin (VCC) of the sensor is connected to the 5V pin of the Arduino, while the ground pin (GND) is connected to the Arduino ground. The signal output pin is connected to the analog input

pin (A0) of the Arduino. This setup allows the sensor to transmit pulse signals to the Arduino for processing.

An I<sup>2</sup>C LCD display is used to present the measured values. The LCD is connected using four pins: VCC, GND, SDA, and SCL. The SDA and SCL pins are connected to the A4 and A5 pins of the Arduino, respectively. The use of I<sup>2</sup>C communication reduces wiring complexity and allows multiple devices to share the same communication bus, making the design more efficient and scalable.

The hardware implementation of the system demonstrates the practical working of the proposed model. The pulse sensor is placed on the fingertip, and the Arduino continuously processes the incoming signals. The calculated heart rate and SpO<sub>2</sub> values are displayed on the LCD and can also be transmitted to a mobile application for remote monitoring. The system operates in real time and provides reliable results under normal conditions.

The results obtained from the system show that the heart rate and oxygen saturation levels can be measured accurately and displayed without significant delay. The system performs consistently for different users, and the measurements fall within acceptable ranges when compared with standard medical devices. Minor variations may occur due to external factors such as finger placement, motion artifacts, and ambient light conditions. However, these can be minimized through proper sensor positioning and signal filtering techniques.

In conclusion, the designed system successfully demonstrates an effective approach to real-time heart rate and SpO<sub>2</sub> monitoring using Arduino. The system is simple, cost-effective, and suitable for both educational and practical healthcare applications. The integration of sensors, microcontroller, and communication modules provides a complete solution for personal health monitoring. Future improvements may include advanced signal processing, cloud integration, and wearable device implementation to enhance performance and usability.

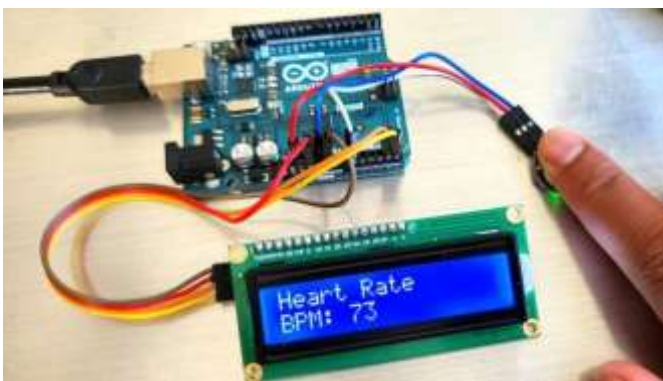


Fig: Hardware implementation with working model

### Library Installation

Several libraries are available for interfacing the MAX30100 sensor with Arduino. In this project, the library developed by OXullo Intersecans is used due to its simplicity and comprehensive functionality. This library provides easy-to-use functions for calculating pulse rate and blood oxygen saturation (SpO<sub>2</sub>), while also exposing most of the internal features of the MAX30100 sensor.

The library can be installed directly through the Arduino IDE Library Manager. To install it, navigate to the **Sketch** → **Include Library** → **Manage Libraries** option. Once opened, the Library Manager will automatically download and update the library index. In the search bar, type “MAX30100” to filter

relevant results. From the displayed list, select the “MAX30100lib” library by OXullo Intersecans and click on the install button. After installation, the library becomes available for use in Arduino programs, enabling efficient communication between the sensor and the microcontroller.

### Blynk Setup

The Blynk platform is used in this project to enable real-time monitoring and visualization of physiological parameters such as heart rate and oxygen saturation. Blynk is a widely used IoT application that allows users to remotely monitor sensor data through a smartphone interface.

The setup process begins by creating a new project within the Blynk application. The user must assign a project name, such as “IoT Pulse Oximeter,” select the appropriate hardware board (e.g., NodeMCU), and choose Wi-Fi as the connection type. After creating the project, an authentication token is automatically generated and sent to the registered email address. This token is essential for establishing communication between the hardware and the mobile application during programming.

Next, the user interface is designed by adding widgets to the project dashboard. By clicking on the “+” icon, gauge and value display widgets are added to visualize the sensor data. Two gauges are typically used: one for displaying heart rate (BPM) and another for oxygen saturation (SpO<sub>2</sub>). For heart rate monitoring, a virtual pin (V1) is assigned, and the display range is set from 0 to 130 BPM. Similarly, a value display widget is configured using the same virtual pin to show numeric values. For oxygen saturation monitoring, another virtual pin (V2) is assigned with a range of 0 to 100%. The gauge and value display widgets are configured accordingly, with an appropriate refresh rate (typically one second) and color settings to enhance visualization. These configurations ensure that real-time sensor data is accurately represented in the application.

Once the setup is complete, the system transmits the measured physiological parameters to the Blynk platform via a Wi-Fi-enabled microcontroller such as ESP8266 or ESP32. The communication is established over the internet, allowing users to monitor their health data from any location. The pulse sensor continuously measures heart rate, and the processed data is sent to the Blynk application, where it is displayed using the configured widgets.

The use of virtual pins in Blynk allows efficient mapping of sensor data to the application interface. Additionally, the platform supports data logging and graphical representation, which helps in analyzing trends over time. This integration significantly enhances the functionality of the system by enabling remote health monitoring, reducing dependency on physical observation, and providing a user-friendly interface for data visualization. Overall, the incorporation of Blynk makes the system more efficient, interactive, and suitable for modern IoT-based healthcare applications.

## 5. RESULTS AND DISCUSSION

The heart rate, oxygen level, humidity, and temperature were successfully measured using the developed system. The measured data was displayed both on the LCD and on the Blynk application for real-time monitoring. The measurement results are shown in the segment below.

Subject Number	Heart Rate (BPM)
1	70.4
2	67.9
3	71.8
4	85.2
5	75.5
6	77.2
7	85.6
8	78.5
9	80.1

Table : readings form different subjects

The experiment was conducted on multiple subjects to evaluate the performance of the system. The observed test results are presented in Table, which demonstrates the effectiveness and reliability of the proposed health monitoring system.

## 6. CONCLUSION AND FUTURE SCOPE

The developed heart rate monitoring system using Arduino and the MAX30102 sensor demonstrates an effective, reliable, and cost-efficient solution for real-time healthcare monitoring. The system successfully measures key physiological parameters such as heart rate (BPM) and blood oxygen saturation (SpO<sub>2</sub>) based on the principle of photoplethysmography. The integration of the MAX30102 sensor with the Arduino microcontroller ensures accurate data acquisition and efficient signal processing. Furthermore, the use of an LCD display and the Blynk application enables clear and real-time visualization of the measured parameters, enhancing user interaction and accessibility. The experimental results obtained from multiple subjects indicate that the system performs consistently and provides satisfactory accuracy for general health monitoring purposes. The incorporation of IoT through the Blynk platform further strengthens the system by enabling remote monitoring, continuous tracking, and easy access to health data through smartphones. Overall, the proposed system offers a simple, portable, user-friendly, and practical approach to personal healthcare monitoring, making it highly suitable for students, fitness enthusiasts, and individuals requiring regular observation of their health status.

Looking toward future advancements, the system can be further enhanced to meet more sophisticated healthcare requirements. Additional biomedical sensors such as ECG, blood pressure, and body temperature sensors can be integrated to develop a more comprehensive multi-parameter health monitoring system. The inclusion of cloud storage and data analytics techniques would enable long-term health record maintenance and facilitate early detection of potential health issues through trend analysis. Moreover, the development of a dedicated mobile or web-based application with advanced features such as real-time alerts, emergency notifications, and improved graphical interfaces can significantly enhance usability and functionality. The system can also be miniaturized into a compact wearable device, such as a smartwatch or wristband, for continuous and convenient monitoring. Incorporating GPS functionality would allow location tracking during emergency situations, particularly for elderly users. Additionally, improvements in power management and battery optimization can further increase the efficiency and portability of the device.

With these enhancements, the proposed system has the potential to evolve into a robust and intelligent IoT-based healthcare solution suitable for real-world medical and remote patient monitoring applications.

## 7. REFERENCES

1. Al Alkeem, E., Shehada, D., Yeun, C. Y., Zemerly, M. J., & Hu, J. (2017). *New secure healthcare system using cloud of things*. Springer Science+Business Media.
2. Kim, Y., Lee, S., & Lee, S. (2014). Coexistence of ZigBee-based WBAN and WiFi for health telemonitoring systems. *IEEE Journal of Biomedical and Health Informatics*. <https://doi.org/10.1109/JBHI.2014.2387867>
3. Baig, M. M., & Gholamhosseini, H. (2013). Smart health monitoring systems: An overview of design and modeling. *Springer Science+Business Media*.
4. Riazul Islam, S. M., Kwak, D., Kabir, M. H., Hossain, M., & Kwak, K. S. (2015). The Internet of Things for health care: A comprehensive survey. *IEEE Transactions*. <https://doi.org/10.1109/TDSC.2015.2406699>
5. Mdhaaffar, A., Chaari, T., Larbi, K., Jmaiel, M., & Freisleben, B. (2017). IoT-based health monitoring via LoRaWAN. In *IEEE EUROCON 2017*.
6. Masud, M. M., Serhani, M. A., & Navaz, A. N. (2015). Resource-aware mobile-based health monitoring. *IEEE*.
7. Bansal, A., Kumar, S., Bajpai, A., Tiwari, V. N., Nayak, M., Venkatesan, S., & Narayanan, R. (2015). Remote health monitoring system for detecting cardiac disorders. *IET Systems Biology*, 9(6), 309–314.
8. Al-Hamadi, H., & Chen, I.-R. (2017). Trust-based decision making for health IoT systems. *IEEE Internet of Things Journal*. <https://doi.org/10.1109/JIOT.2017.2736446>
9. Thangaraj, M., Subramanian, P. P., & Anuradha, A. (2015). Internet of Things (IoT) enabled smart autonomous hospital management system – A real-world healthcare use case with technology drivers. In *IEEE International Conference on Computational Intelligence and Computing Research*.
10. Anil Kumar, M., & Ravi Sekhar, Y. (Year). Android-based health care monitoring system. In *Proceedings of the 2nd IEEE International Conference on Innovations in Information Embedded and Communication Systems (ICIIECS)*.