



A Low-Cost Portable Vital Sign Monitoring System and Mobile Application Customized for Medical Device Designs

Arif Koçoğlu

Biomedical Device Technologies
Department, Baskent University,
arif@baskent.edu.tr

Halid Akkuş

Electrical and Electronics
Engineering, Ankara University,
halidakkus01@gmail.com

Sacit Erdoğan

Electrical and Electronics
Engineering, Ankara University,
saciterd@gmail.com

Cansel Ficici

Electrical and Electronics
Engineering, Ankara University,
ogretmenoglu@ankara.edu.tr

Onur Koçak

Computer & Biomedical
Engineering, Baskent University,
okocak@baskent.edu.tr

Ziya Telatar

Biomedical Engineering, Baskent
University,
ztelatar@baskent.edu.tr

Abstract— This paper presents the design and implementation of a vital signs monitoring system based on embedded electronics and wireless communication. The system aims to acquire real-time physiological data — specifically heart rate, oxygen saturation (SpO₂), and electrocardiogram (ECG) signals — using the MAX30100 and AD8232 biomedical sensors interfaced with an ESP32 microcontroller. The collected data are transmitted wirelessly to a mobile application developed using the Flutter framework and Dart programming language. The system operates over a local Wi-Fi network and employs UDP broadcast for automatic device discovery, eliminating the need for manual IP entry. The mobile application includes responsive charts, dynamic status messages, and ECG signal filtering with real-time R-peak detection. To reduce power consumption, energy-saving strategies were implemented on the microcontroller side. A comprehensive engineering approach was followed: sensor-level signal acquisition, embedded software design, mobile UI development, and real-time visualization. The system provides a low-cost, extensible platform suitable for both academic and practical health monitoring applications, particularly in home-care environments. Project reached accuracy between 95-98% for vital sign results compared with smart watch measurements.

Keywords— ESP32, MAX30100, AD8232, ECG, SpO₂, heart rate, Flutter, Dart, UDP, vital signs monitoring, mobile health, signal processing, simple interface

I. INTRODUCTION

In recent years, the demand for portable and real-time health monitoring systems has increased significantly due to advancements in wireless communication, embedded electronics, and sensor technologies [1]. The rise of wearable devices and remote patient monitoring platforms has opened up new opportunities for personal health tracking, early diagnosis, and chronic disease management [2]. Particularly after the COVID-19 pandemic, the need for non-invasive, cost-effective, and user-friendly biomedical systems has become more prominent in both clinical and non-clinical environments [3].

Recent studies highlight IoT-based and AI-assisted wearables, including a cloud-connected ECG patch with clinical-grade accuracy [4], an IoT-enabled framework for real-time cardiovascular monitoring [5], and a machine learning system for early arrhythmia detection [6]. These developments illustrate the shift toward intelligent, connected, and patient-centered healthcare solutions.

Vital signs such as heart rate (HR), blood oxygen saturation (SpO₂), body temperature, blood pressure, and electrocardiogram (ECG) are critical physiological parameters that reflect the overall condition of the cardiovascular and respiratory systems [7-8]. Continuous and accurate monitoring of these parameters allows for timely detection of anomalies such as arrhythmia, hypoxemia, or cardiac arrest. Traditionally, such measurements require bulky, expensive, and hospital-based equipment, which limits accessibility and early intervention. To overcome these limitations, microcontroller-based wearable systems have emerged as compact alternatives for personal healthcare applications [9].

This paper presents the design and development of a portable, low-cost vital signs monitoring system based on the ESP32 microcontroller [10]. The system integrates the MAX30100 for pulse oximetry and heart rate and the AD8232 for ECG acquisition [11]. Sensor signals are processed and transmitted via Wi-Fi to a mobile device, where they are displayed through a Flutter-based Android application [12].

The system acquires SpO₂, heart rate, and ECG signals in real time, with digital filtering and R-peak detection ensuring accurate analysis. Lead-off detection alerts users to electrode disconnection, preserving data reliability.

Wireless transmission over a local network enables untethered operation and enhances user comfort. The mobile application provides a responsive interface with smooth

animations, theme switching, and automatic device discovery via UDP broadcast, simplifying deployment and improving usability.

This work contributes to mobile health by offering an extensible embedded platform that can be expanded with additional sensors or algorithms, making it suitable for academic research, student training, and personal health monitoring.

II. MATERIAL AND METHODS

The scope of this paper is confined to the design, development, and validation of a prototype-level biomedical monitoring system intended for non-clinical applications, including academic research, personal health tracking, and related investigative purposes. This project does not seek to replace clinical-grade medical devices but rather aims to demonstrate the following:

- The feasibility of integrating embedded systems with mobile platforms to enable continuous health monitoring.
- The implementation of real-time wireless communication between hardware components and software interfaces.
- The provision of both visual and signal-level insights into users' vital signs within a compact system architecture.

Additionally, this project employs a progressive development model, wherein iterative weekly stages of testing, analysis, and optimization were conducted. This approach accounts for practical constraints typical of embedded system design in real-world environments.

The system architecture is organized into five distinct layers to ensure modularity, real-time performance, and clear separation of functionality:

A. Sensor Layer - Biomedical Signal Acquisition

The MAX30100 sensor captures heart rate and peripheral oxygen saturation (SpO₂) using red and infrared photoplethysmography via the I²C interface. Simultaneously, the AD8232 module provides analog electrocardiogram (ECG) signals through differential leads (RA, LA, RL), which are sampled by the ESP32's analog-to-digital converter (ADC) [13]. All components are interconnected on a breadboard to facilitate rapid prototyping [14].

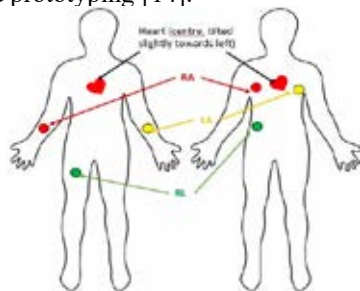


Figure 1. Sensor Positioning Diagram for ECG Measurement

B. Embedded Software Layer – ESP32 Firmware

Firmware development was performed in the Arduino IDE using C++. Key libraries utilized include PulseOximeter, ESP Async Web Server, and WiFi Udp. A state machine orchestrates the measurement sequence: finger detection, stabilization (4 seconds), active measurement (10 seconds), and reset. ECG signals are sampled at approximately 250 Hz, with lead-off detection implemented via LO+/LO- inputs. Sensor data are formatted as JSON and delivered via HTTP, while the ESP32 broadcasts its IP address every 3 seconds over UDP for network discovery.

C. Communication Layer – Wi-Fi Networking

UDP broadcast is employed to enable automatic discovery of the ESP32 by the mobile application. Data retrieval is performed through HTTP polling at intervals of approximately 40 ms, ensuring low-latency acquisition of heart rate, SpO₂, ECG data, and system status to support real-time user interaction. The web server component is implemented on the ESP32 [15-16].

D. Processing Layer

ECG signals are initially processed offline in MATLAB, where raw ADC values are converted to millivolts, then bandpass filtered between 0.5 and 40 Hz. R-peak detection is accomplished using a 5-point sliding window with a 0.5 mV threshold. This detection algorithm was subsequently ported and optimized for real-time execution in Dart [17]. Detected R-peaks are marked on the mobile ECG interface, enabling accurate heart rate visualization under stringent computational constraints.



Figure 2. Bioinstrumentation Module with ECG Leads

E. Mobile Application Layer – Real-Time Visualization

The mobile application is developed using Flutter and Dart frameworks. It supports automatic connection to the ESP32 via UDP discovery and provides real-time graphical displays of heart rate and SpO₂, including animated feedback. The app implements digital ECG filtering within 0.5–40 Hz, real-time R-peak detection, lead-off alerts, and offers dynamic theme switching between dark and beige modes. The interface includes two primary tabs: one dedicated to ECG waveform visualization and another for vital signs such as heart rate and SpO₂.

The design consists of a microcontroller, an AD8232 module for acquiring ECG signals, a MAX30100 sensor for pulse signal detection, and Android applications for visualizing the acquired signals. The critical design block diagram of the system is presented in Figure 4. The electronic system, incorporating an ESP-series microcontroller, was integrated within the Proteus

simulation environment, and the on-board implementation was successfully completed.



Figure 3. Mobile Application for Android



Figure 4. System Block Diagram

III. RESULTS AND DISCUSSION

The processing of electrocardiogram (ECG) signals is a fundamental component of physiological monitoring systems, enabling the detection and analysis of the heart's electrical activity. In this study, ECG signal processing was initially prototyped using MATLAB to develop and validate detection algorithms. Subsequently, the implementation was adapted for real-time operation within a mobile application developed using Flutter. This dual-platform methodology ensured the reliability of signal quality assessment and algorithmic performance prior to deployment.

Analog-to-digital converter (ADC) values were acquired from the ESP32 via HTTP at an approximate sampling rate of 250 Hz. A lightweight high-pass and low-pass filter approximation was applied to the incoming data, and the resulting signal was scaled to enhance the visibility of the ECG waveform [16]. A five-point local maxima detection algorithm was employed to identify R-peaks, which were visually indicated as red markers on the real-time ECG trace. The waveform was rendered using the *fl_chart* library. To optimize performance and memory usage, signal buffers were cleared every five seconds. All processing operations were performed asynchronously to maintain smooth user interface updates. The mobile application environment was designed with minimal computational overhead and visual optimization in mind [18].

The two-stage development approach—initially employing MATLAB for algorithm prototyping and validation, followed by implementation in Flutter for real-time mobile deployment—facilitated a robust and systematic design process for ECG signal processing. The current system demonstrates effective heartbeat detection and smooth waveform

visualization, establishing a strong foundation for future enhancements, including beat-per-minute (BPM) estimation and arrhythmia classification.

Table I. Comparative Summary of MATLAB and Flutter Implementations

Feature	MATLAB	Flutter (Mobile)
Filtering Method	Butterworth Bandpass (0.5–40 Hz)	Simple IIR-style approximation
R-Peak Detection	Local maxima + threshold	Same logic
Visualization	Dual real-time plots	Single graph + peak markers

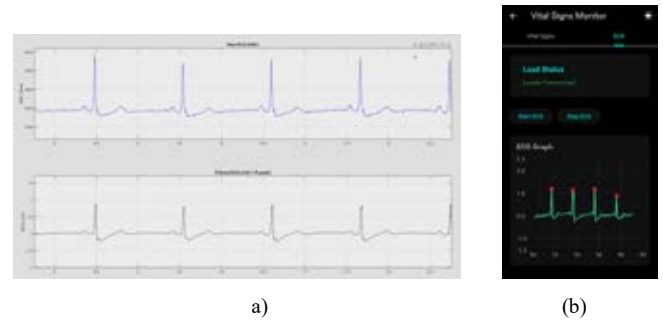


Figure 5. (a) MATLAB Real-Time Processing ECG Signal Plot (b) Mobile App: ECG Tab – ECG Signal Visualization

The Connection Page is the first screen displayed to the user. Its purpose is to automatically detect and connect to the ESP32 device by broadcasting its IP via UDP. Network discovery and connection procedures are managed within the *IPInputPage* module. Incoming UDP datagrams are decoded using the `String.fromCharCode(datagram.data)` method and subsequently parsed by splitting the resulting string at the delimiter “:”. This design eliminates the need for manual static IP configuration, thereby streamlining system setup. The *Vital Signs* tab, integrated into the *HomePage* interface, is dedicated to displaying real-time heart rate and oxygen saturation values acquired from the ESP32 [16, 19].

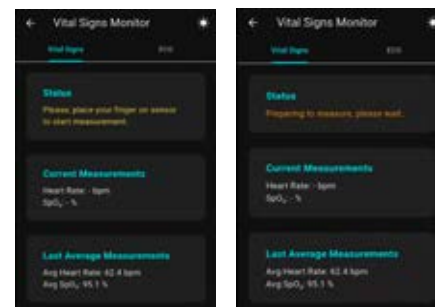


Figure 6. Mobile App: Vital Signs Tab – Status Card pt1

The ECG tab provides real-time visualization of the analog ECG signal processed from the AD8232 module. It performs filtering and basic R-peak detection on the mobile side.

In this study, heart rate and oxygen saturation (SpO₂) values were measured using the *Lifenet Medical™* pulse oximeter available in our laboratory. In addition, measurements were

also obtained from a *Huawei™ Watch D2* smartwatch, and the results were compared accordingly. The acquired ECG signal is presented in Figure 5. This signal was further compared with the 3-channel ECG measurement of a bedside monitor in our laboratory on the same volunteer subject. The ECG waveforms were consistent, and for validation purposes, the heart rate (R-peaks) was counted and compared with the one-minute reference measurement, demonstrating identical results.

SpO₂ validation was performed across three volunteer subjects. Based on one-minute averaged recordings, the smartwatch measurements exhibited a 98% similarity, while the pulse oximeter demonstrated a 95% similarity. Furthermore, R-peak detection results showed 99% agreement. The novelty of this validation process lies in demonstrating that cost-effective wearable devices can achieve high agreement with clinical-grade monitors under experimental conditions, highlighting their potential use in practical biomedical applications.

IV. CONCLUSION AND FUTURE WORKS

The system successfully fulfilled its intended functionalities by measuring heart rate and oxygen saturation using the MAX30100 sensor, acquiring ECG signals from the AD8232 analog front-end module, broadcasting the ESP32's IP address via UDP for automatic network discovery, serving physiological data through an HTTP server in JSON format, and visualizing heart rate (HR), oxygen saturation (SpO₂), and ECG waveforms in real time within a mobile application. Each subsystem, encompassing sensor data acquisition, signal processing, data transmission, and visualization, was seamlessly integrated and operated reliably under typical test conditions.

The MAX30100 sensor consistently provided accurate heart rate and SpO₂ measurements under proper finger placement and stable contact conditions, with a four-second stabilization delay implemented prior to recording to minimize noise. The ECG waveform exhibited clearly distinguishable QRS complexes, particularly when motion was minimized and electrodes were properly positioned, although visual signal quality decreased with hand movement as expected from a non-diagnostic, low-cost sensor. The mobile application, polling the ESP32 every 40 milliseconds, delivered real-time feedback without performance degradation, while UDP-based IP discovery reliably completed within 1–3 seconds after device startup, eliminating the need for manual IP entry. The Flutter-based mobile interface was responsive and user-friendly, supporting both dark and beige/light themes, with a theme toggle feature to enhance readability under varying lighting conditions.

In conclusion, this study presents a technical roadmap for designing a cost-effective vital monitor by integrating embedded electronics, biomedical sensors, and cross-platform mobile software. The system combines real-time physiological feedback with a user-friendly interface, offering both practical healthcare applications and educational value. The novelty of this work lies in providing a structured framework that

translates recent electronic advancements into practical strategies for vital monitoring.

REFERENCES

- [1] How2Electronics, "ECG monitoring with AD8232 ECG sensor and Arduino," 2021. [Online]. Available: <https://how2electronics.com/ecg-monitoring-with-ad8232-ecg-sensor-arduino/>. [Accessed: Aug. 10, 2025].
- [2] K. Tan, A. R. Kavsaoğlu, O. Koçak, and C. Akbay, "Remote monitoring system for incubator data," in *Proc. National Congress on Medical Technologies (TIPTEKNO)*, 2018, pp. 1–4, doi: 10.1109/TIPTEKNO.2018.8597156.
- [3] E. Elif and O. Koçak, "A low cost design non-invasive glucose monitoring experimental setup," in *Proc. National Congress on Medical Technologies (TIPTEKNO)*, 2018.
- [4] J. Lopez, M. Navarro, S. De la Torre, and P. Casaseca, "Wearable patch for continuous electrocardiogram and respiration monitoring: Design, characterization, and clinical evaluation," *IEEE Journal of Biomedical and Health Informatics*, vol. 24, no. 11, pp. 3124–3134, Nov. 2020, doi: 10.1109/JBHI.2020.2991749.
- [5] S. M. R. Islam, A. T. Ghoneim, and K. S. Kwak, "Smart healthcare monitoring framework for IoT-based wearable devices," *IEEE Internet of Things Journal*, vol. 8, no. 21, pp. 15645–15655, Nov. 2021, doi: 10.1109/IJOT.2021.3071564.
- [6] H. T. Nguyen, Y. Zhang, and A. M. Khan, "Early arrhythmia detection using wearable sensors and machine learning: A mobile health approach," *Sensors*, vol. 23, no. 5, p. 2651, 2023, doi: 10.3390/s23052651.
- [7] W. Q. Mok, W. Wang, and S. Y. Liaw, "Vital signs monitoring to detect patient deterioration: An integrative literature review," *International Journal of Nursing Practice*, vol. 21, no. S2, pp. 91–98, 2015.
- [8] U. U. Tarhan, D. Cengiz, and O. Koçak, "Bilgisayar kontrollü damar yeri tespiti cihazı," in *Proc. BPS*, 2014.
- [9] A. T. Tuncer, O. Koçak, and A. Koçoğlu, "PIC microcontroller based RF wireless system design for patient," in *Proc. 7th Int. Advanced Technologies Symp. (IATS)*, 2013.
- [10] Espressif Systems, *ESP32 Datasheet: Wi-Fi & Bluetooth SoC*, 2020. [Online]. Available: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf. [Accessed: Aug. 10, 2025].
- [11] Analog Devices, *MAX30100 Pulse Oximeter and Heart-Rate Sensor for Wearable Health, Datasheet*, 2015. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/MAX30100.pdf>. [Accessed: Aug. 10, 2025].
- [12] Google, *Dart Programming Language Overview*, 2024. [Online]. Available: <https://dart.dev/>. [Accessed: Aug. 10, 2025].
- [13] Circuit Digest, "Interfacing MAX30100 pulse oximeter with Arduino for heart rate & SpO₂ monitoring," 2020. [Online]. Available: <https://circuitdigest.com/microcontroller-projects/interfacing-max30100-pulse-oximeter-with-arduino>. [Accessed: Aug. 10, 2025].
- [14] Analog Devices, *AD8232: Single-Lead, Heart Rate Monitor Front End, Datasheet*, 2016. [Online]. Available: <https://www.analog.com/media/en/technical-documentation/data-sheets/ad8232.pdf>. [Accessed: Aug. 10, 2025].
- [15] GitHub, *ESPAsyncWebServer – Asynchronous Web Server for ESP8266 and ESP32*, 2024. [Online]. Available: <https://github.com/me-no-dev/ESPAsyncWebServer>. [Accessed: Aug. 10, 2025].
- [16] J. Postel, "User Datagram Protocol," *RFC 768*, Internet Engineering Task Force (IETF), 1980. [Online]. Available: <https://datatracker.ietf.org/doc/html/rfc768>. [Accessed: Aug. 10, 2025].
- [17] Google, *Flutter Documentation: Build Beautiful Apps*, 2024. [Online]. Available: <https://docs.flutter.dev>. [Accessed: Aug. 10, 2025].
- [18] MathWorks, *Signal Processing Toolbox: Bandpass Filtering and Peak Detection in Real-Time*, 2024. [Online]. Available: <https://www.mathworks.com/products/signal.html>. [Accessed: Aug. 10, 2025].
- [19] GitHub, *WiFiUDP – Arduino Core for ESP32, Wi-Fi UDP Library for ESP32 Boards*, 2024. [Online]. Available: <https://github.com/espressif/arduino-esp32>. [Accessed: Aug. 10, 2025].