MJFAS
Malaysian Journal of Fundamental and Applied Sciences

RESEARCH ARTICLE

An Embedded System Design for Multi-lead Electrocardiogram On-Device Acquisition and Processing

Thanh-Luan Tran[a,b] , Tien-Thanh Nuyen[a], Quy-Hung Nguyen[a], Vy-Khang Tran[a], Chi-Ngon Nguyen[a], Van-Khanh Nguyen[a,*]

[a]Faculty of Automation Engineering, College of Engineering, Can Tho University, Can Tho city 94000, Vietnam; [b]Faculty of Automotive Technology Engineering, Vinh Long University of Technology Education, Vinh Long province 85000, Vietnam

Abstract Cardiovascular disease (CD) is the leading cause of death in the world. Electrocardiogram (ECG) analysis is an effective method to diagnose CD. In this study, we introduced a microcontroller-based embedded system that allows the collection and processing of ECG signals directly on the device. By integrating low-pass filters (LPFs) and high-pass filters (HPFs), the system can effectively eliminate ultra-low frequency noise, power line noise, and unwanted high-frequency components, thereby improving the fidelity of the signal. R peak detection algorithms (Pan-Tompkins, Hilbert Transform, and Englese & Zeelenberg) have been applied to the acquired ECG of 10 volunteers aged 21–28 years. The results show that these algorithms achieved accuracy of over 99.5%. This study not only proves the ability to improve the reliability of direct ECG collection and analysis based on microcontrollers but also lays the foundation for the development of portable cardiovascular diagnostic devices, supporting early detection and more effective and accessible healthcare.

Keywords: Electrocardiogram, microcontroller, low-pass filter, high-pass filter, R-peak detection.

Introduction

Cardiovascular disease is one of the leading causes of death in most countries worldwide [1]. According to statistics from the World Health Organization (WHO), cardiovascular-related fatalities accounted for 16% of all total deaths in 2023 [2]. This highlights the significant impact of cardiovascular diseases on global health and underscores the urgent need for enhanced solutions to monitor, diagnose, and prevent such conditions early [3]. Typically, the diagnosis and monitoring of cardiovascular diseases rely on the morphology of the electrocardiogram (ECG). An ECG is a graphical representation of the heart's electrical activity over time [4], [5]. ECG signals are collected using electrodes placed on the patient's chest, arms, and legs [6], [7], which are analyzed to detect irregular heart rhythms and conditions related to the myocardium and circulatory system [5], [8], [9]. However, ECG signals exhibit a wide variety of patterns, requiring both specialized medical knowledge and extensive clinical experience by physicians. The speed of analysis is another critical challenge, as many cardiovascular diseases are acute and demand rapid, accurate diagnosis for timely intervention [10]. Consequently, there is an urgent demand for a cost-effective device capable of both acquiring ECG signals and processing them to detect abnormalities associated with cardiovascular diseases.

Numerous studies have focused on the collection and processing of ECG signals using both dedicated and custom-designed circuits. Lv, W. *et al.* [4] proposed a real-time ECG signal acquisition and monitoring system that collects ECG signals and transmits the data via Bluetooth wireless communication, with storage in the cloud. However, this study is limited to data acquisition and does not implement noise filtering or cycle segmentation (QRS detection). In the research conducted by Bui, N. T. *et al*. [8], a real-time ECG signal acquisition model was developed based on the ADS1293 dedicated chip. However, signal filtering and processing algorithms were executed on a computer using MATLAB. Similarly, Guerrero, F. N. [11] proposed a method for ECG data acquisition by combining the MAX 10 Field Programmable Gate Array (FPGA) chip with the ADS131E08 module. This work also focused on

***For correspondence:**
vankhanh@ctu.edu.vn

Received: 13 April 2025
Accepted: 13 August 2025

real-time acquisition without implementing signal processing algorithms on the designed system. Chu, H. *et al*. [12] designed an ECG signal acquisition and processing system that includes a peripheral circuit module for ECG acquisition and a LabVIEW-based processing and display module. However, the study is limited to real-time ECG signal acquisition due to the necessity of further filtering steps in LabVIEW to address signal noise. Ding, J. *et al*. [13] developed an electrode sensor for ECG signal acquisition, which partially reduced signal noise. However, the study did not implement algorithms for ECG signal classification. Lazaretti, G. S. *et al*. [14] proposed an ECG signal acquisition and processing system using BITalino sensors, but the study relied on MATLAB for processing. Similarly, Morello, R. *et al*. [7] designed a system using the NI MyRIO board for ECG acquisition and processing, employing analog filters within the signal acquisition circuit. Jeon, T. *et al*. [15] implemented a real-time ECG signal acquisition and analysis device using ARM microcontrollers (MCUs), but the study did not include algorithms for noise filtering or cycle segmentation. Nguyen, T. N. *et al*. [16] proposed an ECG acquisition system utilizing the ADS1293 chip, displaying the data on a computer via an Arduino Nano integrated with the FT232L chip. However, this study did not address ECG signal filtering or segmentation. Jhang, Y. S. *et al*. [17] developed a small, portable ECG signal acquisition device with Bluetooth communication, designed and integrated with the Nvidia Jetson Xavier NX. Nevertheless, their research did not implement signal classification or cycle segmentation algorithms. Similarly, Su, S. *et al*. [18] proposed an ECG analysis and acquisition system using AD8232 and STM32F103. However, this study also lacked algorithms for signal classification and beat segmentation. These studies demonstrate significant interest in ECG signal acquisition using low-power MCUs. However, post-processing steps such as filtering, QRS detection, and beat segmentation are predominantly executed on computers or mini-computers. This approach poses challenges for the widespread adoption of measurement systems due to high power consumption and associated costs.

ECG signals are often contaminated by noise due to various factors, including patient movement, environmental interference, and other physiological phenomena. Consequently, the application of signal filtering methods is crucial to ensure the acquisition of high-quality ECG signals, which in turn supports accurate diagnosis. Two commonly used filters in ECG signal processing are the low-pass filter (LPF) and high-pass filter (HPF). The LPF is employed to eliminate high-frequency noise originating from sources such as electromagnetic interference, power line disturbances, or patient movement [16], [19], [20]. In contrast, the HPF is used to remove low-frequency noise caused by respiration or slow patient movements [21], [22], [23]. Additionally, other filters have also been applied, such as band-pass filters (BPF) [11], [14], [17], notch filters for suppressing 50 Hz or 60 Hz power lines [4], [5], [24], [25], Wavelet filters [18], adaptive filters [18], and Kalman filters [26]. Although numerous filtering techniques have been implemented to improve ECG signal quality, most have been developed and tested on computers using pre-collected data rather than deployed on systems utilizing MCUs. This limitation can be attributed to two primary reasons. First, while MCUs have significantly advanced, they still face constraints in memory and computational resources. Second, post-processed data is often used for applications requiring diagnostic or AI-based classification, which demand substantial computational and storage capabilities, making direct implementation on MCUs challenging. To enable the widespread adoption of diagnostic systems in the community, it is imperative to implement ECG signal processing algorithms directly on MCUs, potentially followed by artificial intelligence (AI) models. To advance toward fully processing ECG signals on MCUs, this study applies LPF and HPF to denoise ECG signals, ensuring system performance.

Embedded systems are rapidly advancing, with MCUs playing a pivotal role in processing and controlling smart devices. The ability to filter noise and analyze signals directly on MCUs not only improves signal quality but also optimizes system performance and reduces hardware costs. However, implementing signal processing algorithms on MCUs requires careful consideration of their limited resources, including memory and processing speed. In [27], a digital infinite impulse response (IIR) Chebyshev type II filter was implemented to detect fetal heartbeats using infrasonic waves within a filtering frequency range of 2–3 Hz. The filter was designed using MATLAB and deployed on an FPGA Spartan-3E board through hardware description language. The design was based on a bilinear transformation of an analog transfer function, employing Chebyshev polynomials to optimize the desired frequency response. This approach ensured significant attenuation in out-of-band frequencies while maintaining a flat response in the passband. The fourth-order filter achieved a minimum attenuation of 18 dB outside the passband. The study demonstrated that the Chebyshev IIR filter could be effectively implemented on FPGA, meeting technical requirements and paving the way for similar filters to be deployed on various processors and MCUs. In this study, both LPF and HPF filters are implemented directly on MCUs, rather than relying on external design tools. This approach enhances the convenience and applicability of the designed solution, making it a more practical and cost-effective option for embedded systems.

In addition to signal filtering, QRS complex detection and the segmentation of individual heartbeats have garnered significant research interest, as most AI-based diagnostic and classification models require input in the form of single ECG beats. Some QRS detection algorithms were proposed, such as Pan-Tompkins [28], peak threshold and peak filter [29], adaptive thresholds [30], curve-length transform combined with adaptive thresholds [31], combined derivative filtering, Hilbert transform, and Wavelet transform [32], exponential transform (ET) combined with an adaptive threshold based on a proportional derivative (PD) controller [33], and the complex-Pan-Tompkins-Wavelets (CPTW) algorithm [34]. These algorithms were mainly implemented on computers or FPGAs and tested with available online ECG databases. Further study should be conducted to evaluate their real-time performance for integrating on embedded systems. For online QRS detection and beat segmentation, the work in [35] combined the offline Englese & Zeelenberg (E&Z) method with Christov's online adaptive thresholding and tested it with fingertip-measured ECG signals. Tested with bioPLUX commercial device data, the algorithm achieved a mean valid percentage of 94.5%. However, this algorithm was limited to fingertip ECG signals and has not been applied to standard ECG measurements. Similarly, [36] also proposed a real-time detection and segmentation algorithm, but it was also confined to fingertip ECG signals. Finding a suitable QRS algorithm for real-time processing on an embedded system is crucial for developing a stand-alone ECG diagnosis system.

The objective of this research is to develop an independent system capable of collecting, storing, and directly processing ECG signals on an MCU. The system is aimed at capturing 8-lead ECG signals from the sensor, deriving 12-lead ECG signals, and storing long-term data in real-time to support ECG-based studies. Real-time signal processing algorithms, including noise filtering, R peak detection, and cycle segmentation, are implemented directly on the system. The system firmware operates under a multitasking mechanism on a dual-core MCU. This system is designed to be compact, energy-efficient, and cost-effective, with the capability to collect ECG data directly from patients. The performance of all system components will be thoroughly evaluated to demonstrate its potential for practical applications. Experiments show that the system's computational resources are sufficient to integrate AI in the future, enabling it to evolve into an independent diagnostic platform. The developed device is also potentially useful for long-term ECG monitoring, supporting the analysis and detection of cardiovascular conditions such as sleep apnea, atrial fibrillation, etc.

Methodology

Hardware Design

To enable the system to handle real-time tasks related to ECG acquisition, such as real-time measurement of 12-lead ECG signals, noise filtering, beat segmentation, data storage, and displaying relevant information on a screen, a hardware block diagram is proposed, as shown in Figure 1 and Figure 2 is system schematic diagram. The design incorporates the TI ADS1298ECG-FE semiconductor chip for direct processing and acquisition of 8-lead ECG signals, paired with a supporting electrode system. A dual-core ESP32 MCU ensures efficient multitasking, enabling real-time data acquisition and processing. The collected signals are continuously buffered into the MCU's memory and stored on an SD card for later processing. A Thin-Film Transistor (TFT) screen is used to display system-related parameters and sampled signals from the 12-lead ECG.

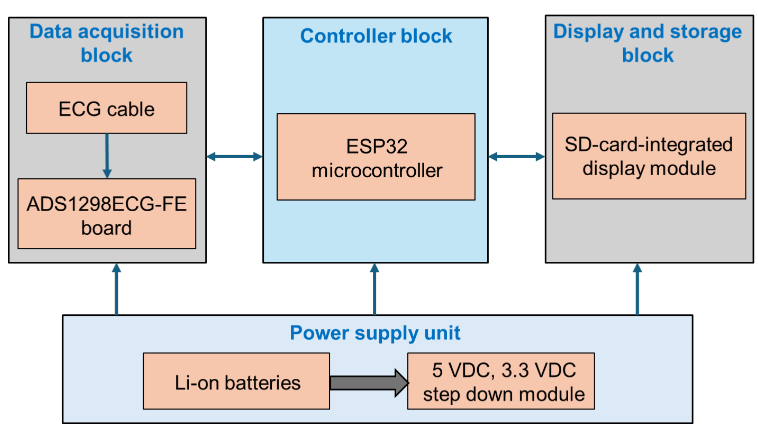


Figure 1. Block diagram of the system

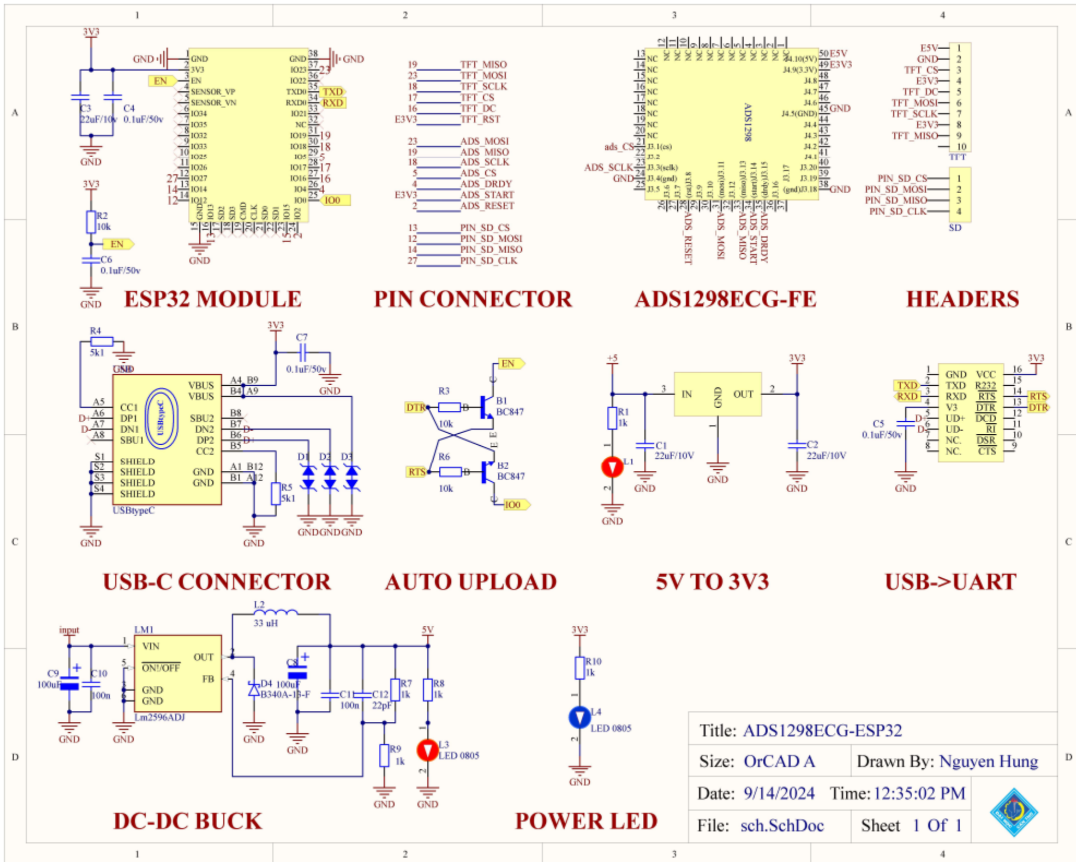


Figure 2. System schematic diagram

Embedded Algorithm

The ESP32 MCU acts as the central scheduler for the system's tasks, ensuring that it simultaneously acquires data from 8-channel ECG measurements directly from the ADS1298ECG-FE at a rate of 500Hz, computes the remaining 4-lead ECG signals, performs noise filtering for the LII lead, and stores all data on the memory card. At the same time, it executes the algorithm for extracting the most recent complete ECG beat. The extracted ECG cycle is downsampled to 360 Hz to be compatible with current databases. Additionally, parameters such as the RR interval, QRS duration, and P-wave are displayed on the TFT screen for real-time monitoring.

Figure 3 depicts the main algorithm of the system, which performs data transmission tasks, including sending configuration data to the ADS1298ECG-FE, SD card, and TFT screen, as well as reading data from the ADS1298ECG-FE. Data transmission and reception are carried out via the serial peripheral interface (SPI). Since the data-ready (DRDY) signal pin of the ADS1298ECG-FE is connected to an external interrupt of the ESP32, the external interrupt function is configured to prevent any data loss. The read data is then converted to voltage values, followed by filtering using high-pass and low-pass filter functions. Simultaneously, the acquired data is displayed on the TFT screen. Finally, the data is stored on the SD card.

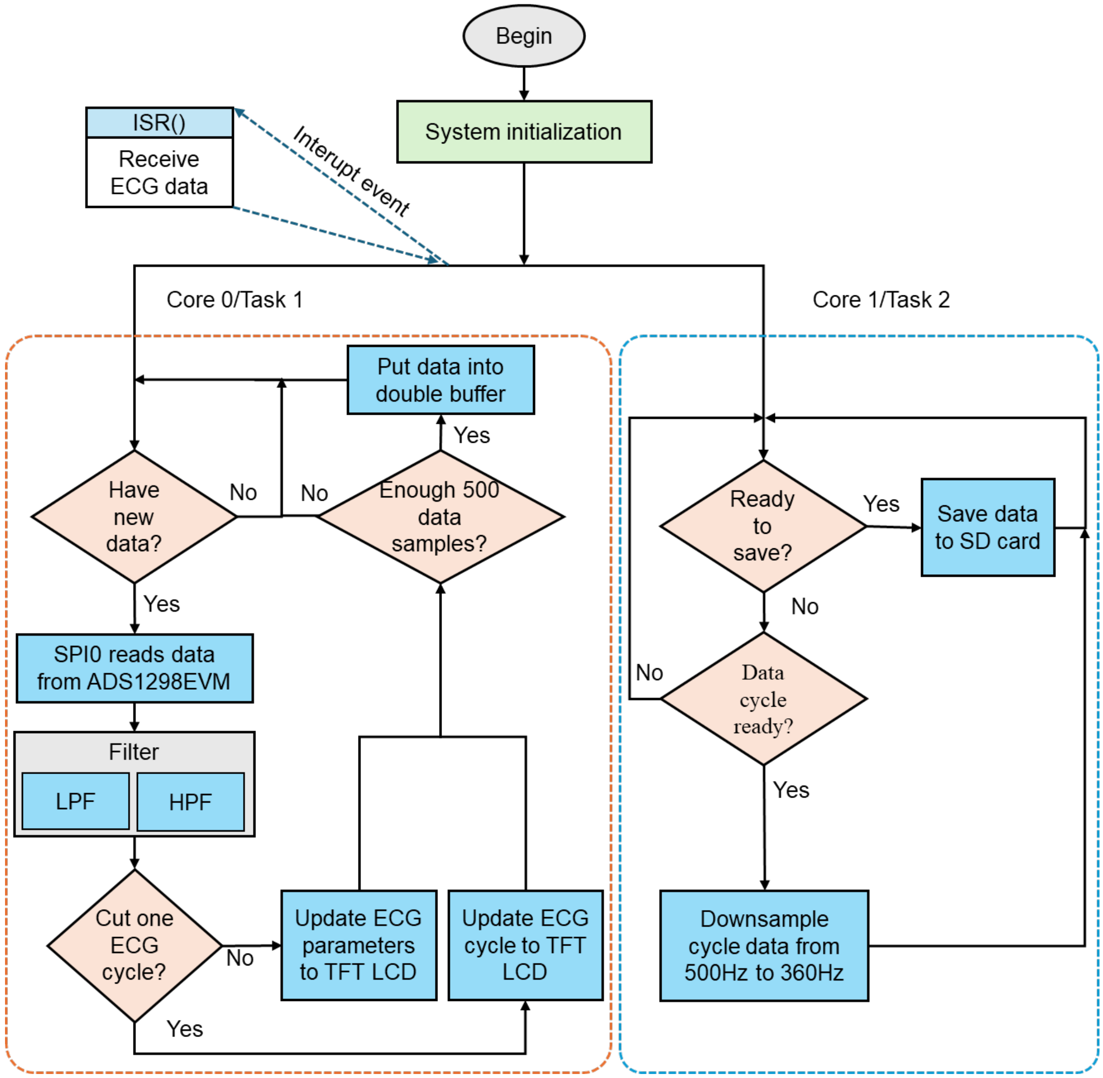


Figure 3. Algorithm flowchart of the system

The firmware algorithm consists of three main tasks: *dataAcquiProcess, dataSave*, and *ISRRoutine*, scheduled as multitasks based on the freeRTOS platform [36]. The *dataAcquiProcess* and *dataSave* tasks are pinned to core 0 and core 1 of the ESP32, respectively, to run in parallel, while the *ISRRoutine* is handled by Core 1. The I*SRRoutine* is triggered by a falling edge of the DRDY signal. Immediately after this, both the *dataAcquiProcess* and *dataSave* tasks are activated to receive data streams from the ADS1298ECG-FE, filter the signals, extract a full LII cycle, display it on the TFT, calculate the indirect channels, and store the data on the SD card. On each *interrupt, dataAcquiProcess* directly receives 8 channels of data from the ADS1298ECG-FE (Lead I, Lead II, V6, V5, V4, V3, V2, and V1) and calculates the remaining 4 channels (Lead III, avR, avL, and avF) using (1), (2), (3), and (4). When 500 data samples are collected on each channel, equivalent to 500 interrupts occurring, a total of 4000 samples will be stored on the SD card.

$$L3 = L2 - L1 \tag{1}$$
$$aVR = -(L1 + L2)/2 \tag{2}$$
$$aVL = (L1 - L2)/2 \tag{3}$$
$$aVF = (2L2 - L1)/2 \tag{4}$$

Beat Segmentation

In ECG signal processing, the extraction of a complete cardiac cycle is typically performed in three main steps—signal preprocessing, QRS complex detection, and cycle segmentation—as illustrated in

Figure 4. In the first stage, the quality of the ECG signal is enhanced by removing noise and unwanted distortions. Both LPF and HPF are applied to eliminate noise from high frequencies, body motion, or power line interference. The cutoff frequencies of the LPF and HPF are set at 40 Hz and 0.5 Hz, respectively. The 4th-order LPF and 2nd-order HPF are applied and designed based on IIR filters, with the transfer functions represented by (5) and (6). The design and implementation of the filters on the MCU are presented in detail in [36].

$$H_{HPF}(z) = \frac{0.995567033 - 1.991134067z^{-1} + 0.995567033z^{-2}}{1 - 1.991114378z^{-1} + 0.991153657z^{-2}} \tag{5}$$

$$H_{LPF}(z) = \frac{0.002077467 - 0.008309868z^{-1} + 0.012464803z^{-2} + 0.008309686z^{-3} + 0.002077467z^{-4}}{1 - 2.717593123z^{-1} + 2.911990841z^{-2} - 1.431263441z^{-3} + 0.270105198z^{-4}} \tag{6}$$

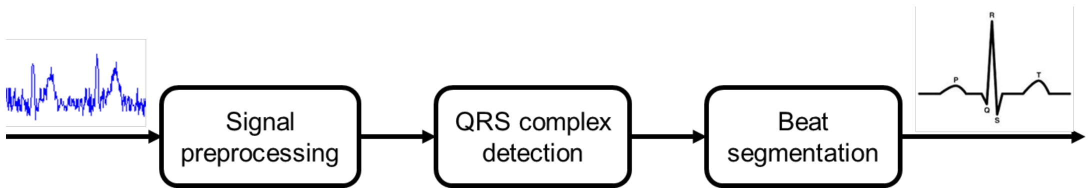


Figure 4. ECG signal processing procedure

After filtering, the QRS complex is detected through the R-wave detection algorithm. Three typical R-wave detection algorithms are utilized, including Pan-Tompkins, Hilbert transform, and E&Z. They are conducted and assessed into identify the optimal algorithm in terms of performance and execution time. The R-peaks are detected based on an adaptive threshold defined by the algorithm proposed in [29]. By initializing a sliding window containing N data samples, the window gradually slides one sample to the right of the signal. Each time the window slides, the maximum value, or the peak, is found, and the threshold is recalculated as in Eq. (7):

$$Threshold = \alpha\gamma Peak - (1 - \alpha)Threshold \tag{7}$$

Where α is the smoothing coefficient, within the range [0, 1], providing flexibility in updating the threshold, and γ is the scaling constant, adjusting the sensitivity of the threshold in relation to the maximum value of the sliding window.

The adaptive threshold applied to the output signal of algorithms such as Pan-Tompkins, Hilbert Transform, and E&Z optimizes the QRS detection process, allowing for faster and more accurate identification of R-peaks. By adjusting the threshold according to each specific signal and scenario, the ability to detect QRS complexes is significantly improved, thereby enhancing the reliability and efficiency of ECG signal processing methods. In an ECG cycle, a data point $x[n]$ is considered the R peak of the complex if the values on both sides of x[n] are smaller than the value itself and all must be greater than the threshold, as described in Eq. (8):

$$x[n] > x[n-1], x[n] > x[n+1], x[n] > Threshold \tag{8}$$

By initializing a sliding window containing 250 samples from the filtered signal, the window will gradually move to the right, one sample at a time. As the window slides, the oldest sample $x[n]$ is discarded, and the newest sample $x[n+250]$ is added to the end of the sliding window. When the newest sample is at a position where an R peak is detected, the current 250 samples in the sliding window will be recorded. Afterward, the sliding window will shift by 250 samples, making the R peak position the oldest sample in the window. Then, 250 new samples will be recorded in the window, forming an ECG cycle with a total of 500 samples.

Signal Downsampling

To ensure that the data collected from the system is compatible with existing online databases like MIT-BIH Arrhythmia, the ECG cycle after segmentation will be further downsampled to 360 Hz. The downsampling algorithm used is based on linear interpolation, a technique that reduces the number of samples while maintaining the important features and characteristics of the ECG signal [37]. Linear interpolation is a method for estimating values between two adjacent known points by the following equation.

$$o_{ecg}[i] = i_{ecg}[index] + frac(i_{ecg}[index + 1] - i_{ecg}[index]) \quad (9)$$

Where o_{ecg} is the downsampling output, i_{ecg} is the input, and *frac* is the ratio of input and output sampling speed. In this study, the ECG is downsampled and returned one beat at a time, with the goal of using it as input for AI models for classification.

Real-time Testing and Data Acquisition

The experiment was conducted at the PLC technology and Industrial IoT laboratory, Can Tho University, Vietnam. The experiment was conducted on a group of volunteers who were students at Can Tho University, consisting of 10 male participants aged between 21 and 28. All participants volunteered to take part and met the study's selection criteria. The ECG signals were collected from electrodes placed at the lead positions on the volunteer's body as shown in Figure 5. The limb leads were clipped at the following positions: right wrist (RA), left wrist (LA), right ankle (RL), and left ankle (LL). The precordial leads had electrodes placed at the following locations: the 4th intercostal space at the right of the sternum (V1), the 4^{th} intercostal space at the left of the sternum (V2), midway between V2 and V4 (V3), the 5^{th} intercostal space along the left midclavicular line (V4), horizontally with V4 along the left anterior axillary line (V5), and horizontally between V4 and V5 along the left midaxillary line (V6). The electrodes were connected to an embedded system to collect, process, and store the signals onto a memory card before sending them to be displayed on a computer for monitoring.

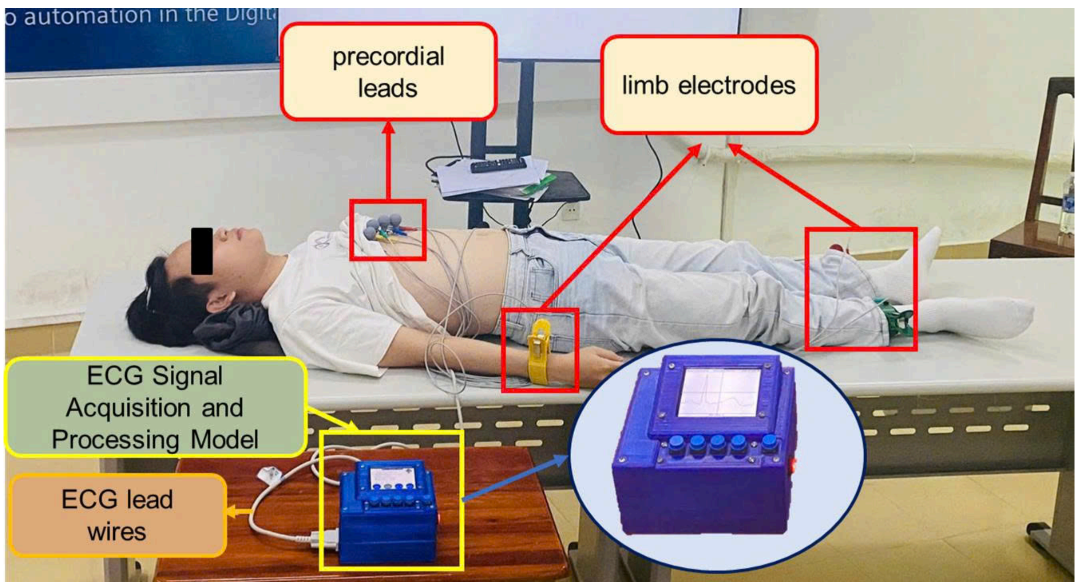


Figure 5. Experimental setup

Each volunteer will have their ECG recorded twice, with each recording lasting 30 seconds. The data collection process follows these steps: (1) the volunteer lies relaxed on the table to have the electrodes connected, (2) The measurement kit is powered on, and the signal is observed on the computer screen, and (3) the measurement begins for 30 seconds, and data is stored if the signal is stable and without noise. If the signal is unstable, step (1) is repeated. During the entire measurement process, the ECG circuit is powered by a battery, and the laptop is not connected to the grid to minimize power supply noise. The volunteer remains relaxed and still throughout the measurement. In addition to short-cycle recordings, some volunteers will be recorded for longer cycles to evaluate the stability of the R peak detection algorithm and beat segmentation.

Evaluation Indicators

The results of real-time detection and segmentation in this study are evaluated using three metrics: *Se* (sensitivity), *+P* (positive predictive value), and *Acc* (accuracy). Their mathematical representations are given by (10), (11), (12), and (13).

$$\mathrm{Se} = \frac{TP}{TP+FN} \times 100\% \quad (10)$$

$$+P = \frac{TP}{TP+FP} \times 100\% \quad (11)$$

$$DER = \frac{FP+FN}{The\ total\ number\ of\ QRS\ completes\ in\ the\ signal\ segment} \times 100\% \quad (12)$$
$$ACC = 100\% - DER \quad (13)$$

Where *TP* is the number of correctly detected ECG cycles with R-peaks within the Q to S interval, *FP* is the number of ECG cycles with R-peaks incorrectly detected outside the QRS complex (including the Q and S peaks), and *FN* is the number of R-peaks missed between QRS cycles.

Results and Discussion

System Performance Analysis

As analyzed, FreeRTOS plays a key role in task scheduling and multitasking in the system. The firmware is responsible for continuously reading the 8-channel ECG data stream from the ADS1298ECG-FE, filtering, and processing the beat segmentation, while simultaneously displaying the data on the screen and saving it to the memory card. Figure 6 illustrates the timeline for the two main tasks and the interrupt service routine (ISR) handling the data stream. The ISR and the *dataSave* task are executed by core 1 of the ESP32, while the *dataAcquiProcess* task runs on core 0.

The use of two cores allows efficient parallel computation, ensuring real-time processing capabilities for the system. When an interrupt occurs, the *dataAcquiProcess* task is scheduled to start immediately after the interrupt. This task operates for approximately 430 µs, then releases core 0 and waits for rescheduling to restart. After every 500 executions of the *dataAcquiProcess* task, the *dataSave* task is scheduled to start and operates for approximately 150,000 µs. During this time, the dataSave task operates concurrently with the *dataAcquiProcess* task. Once completed, it releases core 1 and waits for rescheduling to restart.

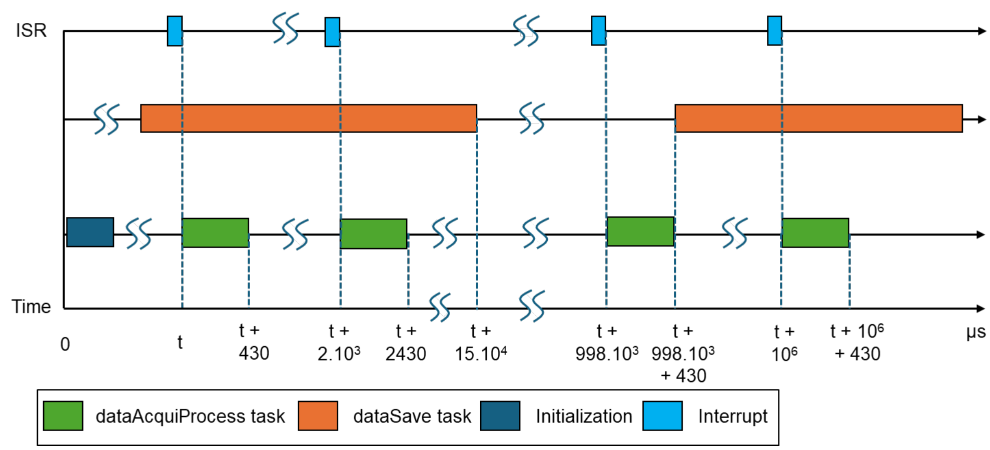
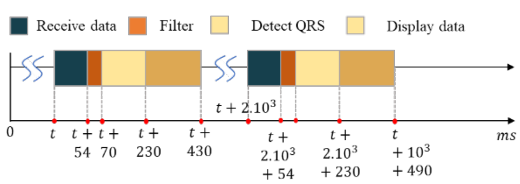


Figure 6. Relationship between tasks and interrupts in freeRTOS

The *dataAcquiProcess* task plays a critical role in ensuring accurate ECG data stream reading and processing. This task is responsible for communicating with the ADS1298ECG-FE board to transmit and receive data using the SPI protocol at a frequency of 4 MHz. The received data consists of 8 leads, including Lead I, Lead II, V6, V5, V4, V3, V2, and V1. Leads III, aVR, aVL, and aVF will be enhanced using formulas, so they are not currently computed on the ESP32. Lead II (LII) is filtered with an HPF and LPF at cutoff frequencies of 0.5 Hz and 40 Hz, respectively, before applying the algorithm to detect and segment the current ECG cycle. This cycle can be used for classification or diagnosis tasks in the future. Filtering and segmentation are only performed on LII for now but can be extended to 8 leads if necessary. All captured data is then saved to the SD card.

The detailed execution time of the *dataSave* and *dataAcquiProcess* tasks is illustrated in Figure 7. The *dataAcquiProcess* task also communicates with the TFT screen via the SPI protocol at a frequency of 40 MHz to display the LII signal and related parameters such as heart rate, RR interval, QRS, and P-wave in real-time. Figure 7(a) shows that the total execution time of *dataAcquiProcess* is approximately 430 microseconds. The *dataSave* task saves the data to the SD card using an SPI speed of 19 MHz. Afterward, to ensure compatibility with databases like MIT-BIH, the segmented ECG cycle is downsampled to 360 Hz using the algorithm described in the above section. The total execution time for *dataSave* is 0.15 seconds, as shown in Figure 7(b). It is proved that core 1 still has more room to integrate additional tasks such as running machine learning models for classification, diagnosis, cloud server communication, etc.



(a) (b)

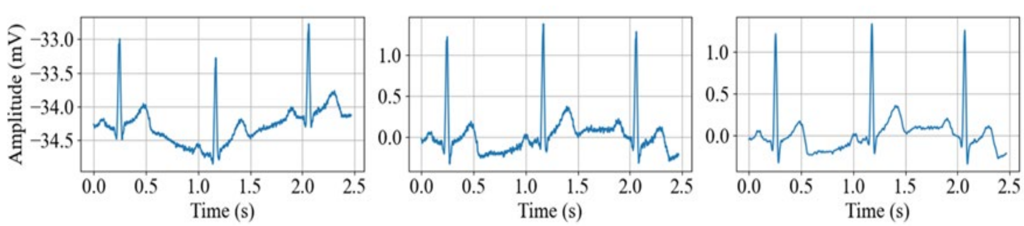
Figure 7. Details the execution time of the (a) *dataSave* and (b) *dataAcquiProcess* tasks

Acquired 12-lead ECG

The ECG measurement system was used for experiments on volunteers with the goal of collecting data for comparison with existing databases. The system was tested on 10 volunteers aged from 21 to 22, with each volunteer undergoing two measurements, each lasting 30 seconds. Each volunteer produced 4 data segments, except for one who had 5 segments. In total, 41 data segments were successfully collected from the volunteers. Only the 8 direct leads (including Lead I, Lead II, V6, V5, V4, V3, V2, and V1) and the data after LPF and HPF of LII were stored. Four other leads are easily derived by equations from (1) to (4) when necessary.

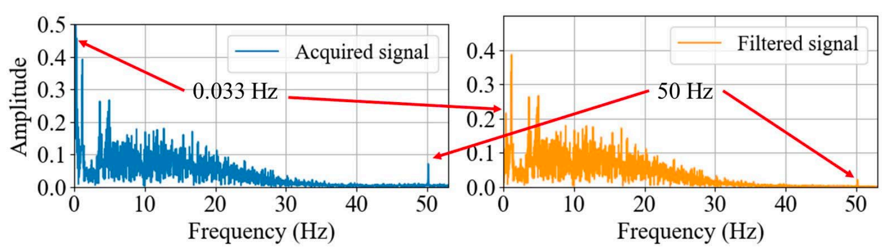
Signal Filter Results

The LII data from all volunteers were filtered and stored directly using the designed circuit. This study applies both LPF and HPF with cutoff frequencies of 40 Hz and 0.5 Hz, respectively. These filters were used to remove ultra-low frequency noise, 50 Hz electrical grid noise, and high frequencies from the data. Figure 8 depicts the filter results. Figure 8(a) shows a raw data segment, which clearly demonstrates the presence of ultra-low frequency components at 0.033 Hz and 50 Hz noise, as shown in the frequency analysis results. After passing through the HPF (Figure 8(b)) and LPF (Figure 8(c)), these noise components were significantly reduced. The amplitude of the signal was attenuated, as demonstrated in the filtered signal's spectrum in Figure 9. In Figure 9(a), the raw signal spectrum has two distinct noise peaks at 0.033 Hz and 50 Hz, but after filtering (Figure 9(b)), these peaks are almost eliminated, leaving the amplitude in the characteristic frequency range of the ECG signal, proving that the noise and high-frequency components have been effectively removed. The filtering was performed continuously on the data read from the ADS1298, with execution times for the LPF and HPF being 5 and 3 microseconds, respectively. The filtered LII records will be used to test the cycle cutting algorithm and measure ECG parameters for comparison with online databases.



(a) (b) (c)

Figure 8. Filtering results (a) raw data, (b) LPF, and (c) HPF



(a) (b)

Figure 9. Frequency spectrum analysis: (a) raw data and (b) filtered data

The parameters of the ECG signal from the volunteers and the MIT-BIH Arrhythmia database were also calculated and statistically analyzed to evaluate the data collected from the designed circuit. The statistical results are presented in Table 1. The results show that the average values of the R-R interval, QRS duration, and P wave duration are consistent with those of the database. The minimum and maximum values also fall within the range of the database values. This indicates that the data collected from the study is initially compatible with the database parameters and can be used for subsequent research, such as anomaly detection or classification.

Table 1. Parameters of ECG data collected in this study compared with MIT-BIH ARRHYTHMIA

No.	Parameter	This study				MIT-BIH			
		N	Min	Mean	Max	N	Min	Mean	Max
1	R-R Interval (s)	1356	0.606	0.805	1.148	1062	0.431	0.808	1.452
2	QRS Duration (s)	1397	0.060	0.078	0.138	1062	0.039	0.092	0.136
3	P Wave Duration (s)	1387	0.026	0.101	0.188	1060	0.017	0.095	0.211
4	Heart Rate (bpm)	1356	52	75	99	1062	41	77	139

Beat Segmentation Result Analysis and Comparison

Real-time segmentation of the ECG signal plays a crucial role in real-time ECG classification. The segmented data are used to implement ECG signal classification algorithms or diagnose cardiovascular diseases. As previously described, the real-time algorithm in this study is based on common QRS complex detection algorithms such as Pan-Tompkins, Hilbert, and E&Z. These algorithms primarily rely on accurately identifying the R peak after processing the ECG with an adaptive threshold algorithm. To simplify real-time processing, a single cardiac cycle is defined as the set of samples to the left and right of the detected R peak.

In this study, the R peak detection algorithms are compared based on two criteria: the deviation rate of the detected R peak compared to the true R peak and the execution time of each algorithm. Figure 10 depicts how to evaluate the peak deviation ratio *n*. Figure 11 illustrates the possible cases when cutting an R peak. Figure 11(a) represents the case where all R peaks are correctly detected, while Figures 11(b) and 10(c) illustrate cases where the R peak is either detected incorrectly or missed. Table 2 presents the statistical results of the error in detecting the R peak compared to the true peak, measured in sample misalignment (i.e., *n* in Figure 10). The results show that applying the Hilbert transform achieved the highest R peak detection rate at 100%, with no error. The Pan-Tompkins and E&Z algorithms both exhibit similar R peak detection accuracy, with an average error of 0.43 samples. However, the standard deviation of the Pan-Tompkins algorithm is larger, at 0.53 samples, compared to 0.51 samples for the E&Z algorithm. The maximum sample deviation for both algorithms is 9 samples. Therefore, in terms of R peak detection accuracy, the Hilbert transform-based algorithm outperforms the others.

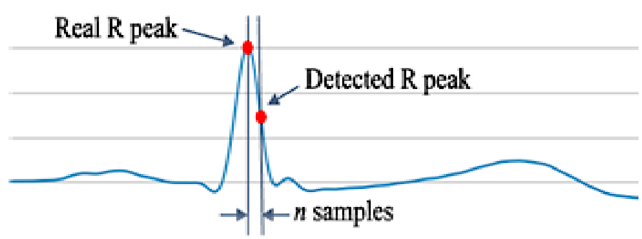


Figure 10. Illustration of the R peak detection results

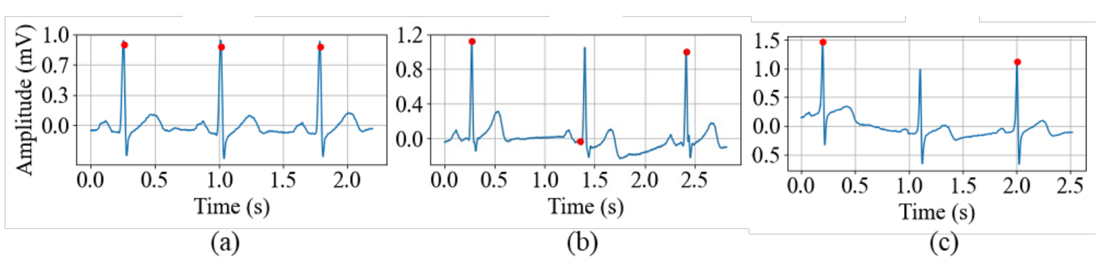


Figure 11. R peak detection results: (a) successful, (b) wrong R peak location, and (c) R peak ignored

Table 2. Peak deviation rate (sample)

No.	Method	Samples	StDev	Mean	Max
1	Hilbert	13050	0	0	0
2	E&Z	13055	0.51	0.43	9
3	Pan-Tompkins	13060	0.53	0.43	9

The real-time cycle cutting times differ mainly in the ECG signal preprocessing phase of the three algorithms presented in Table 3. The algorithm using the Hilbert transform requires more complex signal transformation algorithms, resulting in a processing time of 107 microseconds, while the E&Z and Pan-Tompkins algorithms require only 2 and 3 microseconds, respectively, to complete. The total time for the adaptive thresholding algorithm and cycle cutting for all three algorithms is the same, at 108 microseconds. Therefore, the total time for detecting and cutting the most recent cycle of LII using the Hilbert transform is nearly double that of the other two algorithms 222 microseconds compared to approximately 120 microseconds. However, based on the results, all three algorithms ensure that the system correctly reads data from the ADS1298ECG-FE chip. Thus, if the system requires precise R peak detection, the Hilbert transform-based algorithm should be used. On the other hand, if the system does not prioritize R peak accuracy compared to the true peak, the other two algorithms can be used. Additionally, linear interpolation has been successfully applied to downsample the data from 500 Hz to 360 Hz, ensuring compatibility with online databases. Figure 12 presents the downsampling results for any ECG cycle. The graph shows that the downsampled data fits well with the original data.

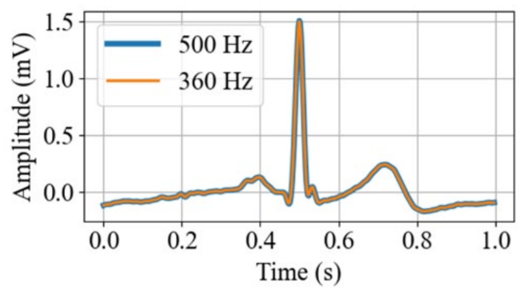


Figure 12. ECG signal downsampling results

Table 3. Real-time beat segmentation algorithms performance

No.	Task	Hilbert	E&Z	Pan-Tompkins
1	Signal pre-processing	107	2	3
2	Thresholding	106	106	106
3	Beat segmentation	2	2	2
4	Total time (μS)	222	117	118

Table 4 presents and evaluates the performance results of accurate QRS complex detection and real-time ECG cycle cutting success rates. A total of 13,069 ECG cycles were tested. The results show that the Pan-Tompkins algorithm achieved the highest accuracy rate of 99.99% in detecting R peaks. The Hilbert transform algorithm had a slightly lower QRS detection accuracy rate compared to the E&Z algorithm, with a difference of about 0.20%.

Additionally, Table 4 compares performance indicators such as *Se, +P*, and *Acc* with previous studies. In the application of the Pan-Tompkins algorithm, [28] achieved a 99.32% accuracy rate in successfully detecting the QRS complex on 116,137 ECG cycles from an online database. In this study, Pan-Tompkins achieved impressive results, testing approximately 1,500 online ECG cycles from volunteers, with a detection rate of 100%. Therefore, if the test sample size is expanded, the system promises good accuracy for QRS detection. Study [29] proposed a relatively lightweight algorithm with good real-time performance, but the QRS detection success rate was quite low. Apart from studies [29] and [32], other studies like [30], [31] and [33] reported QRS detection rates above 99.5%, while [28] and [35] had lower success rates of 99.32% and 99.1%, respectively. The algorithms tested in this study achieved QRS detection rates of 99.5% or higher, although the online test sample size was still much smaller than in the mentioned studies. Therefore, the system needs further testing to confirm its accuracy. Nonetheless, the study has successfully integrated all tasks, such as data collection, processing, and storage of ECG signals, directly on the embedded system. Additionally, the study demonstrated that with multitasking programming, the system can accurately cut out ECG signals in real-time, and the processor still has sufficient room to integrate additional advanced features like ECG anomaly detection, classification based on tiny machine learning models, and enabling server connectivity for building remote monitoring and pre-diagnosis systems.

Table 4. Summary table comparing QRS detection algorithms

No.	QRS detection method	ECG dataset	Total (beats)	TP (beats)	FN (beats)	FP (beats)	Se	+P	Acc
1	Peak threshold [29]	Self-collected data	-	-	-	-	74.62	92.39	70.30
2	Peak filter [29]	Self-collected data	-	-	-	-	83.61	98.38	82.47
3	Algorithm 1 [30]	MIT-BIH	110050	109548	294	215	99.69	99.8	99.53
4	Algorithm 2 [30]	MIT-BIH	110050	109615	240	239	99.74	99.78	99.56
5	[31]	MIT-BIH	109496	109342	154	177	99.86	99.84	99.7
6	MIT-BIH [32]	MIT-BIH	109495	108568	928	856	99.08	99.12	98.19
7	[33]	MIT-BIH	109966	109742	91	224	99.90	99.92	99.82
8	Pan-Tompkins [28]	MIT-BIH	116137	-	277	507	-	-	99.32
9	Complex-Pan-Tompkins-Wavelets [35]	Self-collected data	-	-	-	-	99.57	99.58	99.10
10	Online Ag/AgCl electrode [34]	MIT-BIH	12453	-	-	-	-	-	93.54
11	This study (Pan-Tompkins)	Self-collected data	13069	13055	5	9	100	99.99	99.99
12	This study (Hilbert transform)	Self-collected data	13069	13050	4	15	99.92	99.59	99.51
13	This study (E&Z)	Self-collected data	13069	13060	3	6	100	99.72	99.71

Conclusions

In this study, we successfully designed and implemented a direct ECG signal acquisition and processing system on the MCU. The study delivers four key contributions. First, integrating two digital filters—an HPF at 0.5 Hz and an LPF at 40 Hz—directly on a microcontroller demonstrated effective suppression of ultra-low-frequency drift, 50 Hz mains interference, and unwanted high-frequency components, with execution times of just 3 µs and 5 µs per sample, respectively, meeting near real-time processing requirements on resource-constrained hardware. Second, three R-peak detection algorithms (Pan–Tompkins, Hilbert Transform, and Englese & Zeelenberg) were simultaneously evaluated on data collected from 10 volunteers, achieving QRS detection accuracies of ≥ 99.5%, with Pan–Tompkins reaching 99.99%. Third, the system accurately extracted complete ECG cycles for each heartbeat, providing a reliable foundation for subsequent automated classification and diagnosis. Finally, the fully microcontroller-based implementation paves the way for developing low-cost, portable medical devices that support remote cardiovascular monitoring and diagnosis.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgment

The author would like to express sincere gratitude to the Faculty of Automation Engineering at the University of Technology in Can Tho City, Vietnam, for providing the necessary resources for this research, and also gives special thanks to the volunteers for their valuable support.

References

[1] Raheja, N. & Manocha, A. K. (2023). An IoT enabled secured clinical health care framework for diagnosis of heart diseases. *Biomedical Signal Processing and Control*, *80*.

[2] Tsao, C. W., Aday, A. W., Almarzooq, Z. I., Anderson, C. A. M., Arora, P., Avery, C. L., Smith, C. M. B., Beaton, A. Z., Boehme, A. K., Buxton, A. E., Mensah, Y. C., Elkind, M. S. V., Evenson, K. R., Nliam, C. E., Fugar, S., Generoso, G., Heard, D. G., Hiremath, S., Ho, J. E., Kalani, R., Kazi, D. S., Ko, D., Levine, D. A., Liu, J., Ma, J., Magnani, J. W., Michos, E. D., Mussolino, M. E., Navaneethan, S. D., Parikh, N. I., Poudel, R., Hanna, M. R., Wang, N. Y., Wong, N. D., Wong, S. S., Yaffe, K. & Martin, S. S. (2023). Heart Disease and Stroke Statistics—2023 Update: A Report From the American Heart Association. *Circulation*, *147*(8), e93–e621.

[3] Faruk, N., Abdulkarim, A., Emmanuel, I., Folawiyo, Y. Y., Adewole, K. S., Mojeed, H. A., Oloyede, A. A., Olawoyin, L. A., Sikiru, I. A., Nehemiah, M., Gital, A. Y., Chiroma, H., Ogunmodede. J. A., Almutairi, M. & Katibi, I. A. (2021). A comprehensive survey on low-cost ECG acquisition systems: Advances on design specifications, challenges and future direction. *Biocybernetics and Biomedical Engineering*, *41*(2), 474–502.

[4] Lv, W. & Guo, J. (2021). Real-time ECG signal acquisition and monitoring for sports competition process oriented to the Internet of Things. *Measurement*, *169*.

[5] Gupta, V., Mittal, M., Mittal, V. & Saxena, N. K. (2021). BP Signal Analysis Using Emerging Techniques and its Validation Using ECG Signal. *Sensing and Imaging*, *22*(1), 25.

[6] Singh, A. K. & Krishnan, S. (2023). ECG signal feature extraction trends in methods and applications. *BioMedical Engineering OnLine*, *22*(22).

[7] Morello, R., Ruffa, F., Jablonski, I., Fabbiano, L. & Capua, D. C. (2022). An IoT based ECG system to diagnose cardiac pathologies for healthcare applications in smart cities. *Measurement*, *190*.

[8] Bui, N. T. & Byun, G. S. (2021). The comparison features of ecg signal with different sampling frequencies and filter methods for real-time measurement. *Symmetry*, *13*(8).

[9] Pereira, T. M. C., Conceição, R. C., Sencadas, V. & Sebastião, R. (2023). Biometric Recognition: A Systematic Review on Electrocardiogram Data Acquisition Methods. *Sensors*, *23*(3), 1–31.

[10] Rizwan, A., Priyanga, P., Abualsauod, E. H., Zafrullah, S. N., Serbaya, S. H. & Halifa, A. (2022). A Machine Learning Approach for the Detection of QRS Complexes in Electrocardiogram (ECG) Using Discrete Wavelet Transform (DWT) Algorithm. *Computational Intelligence and Neuroscience*, *2022*(1).

[11] Guerrero, F. N., Oliva, M. J. & Spinelli, E. M. (2023). Real-time noise reduction through independent channel averaging for real-time biomedical signal acquisition. *XI Southern Programmable Logic Conference Proceedings*.

[12] Chu, H., Zhen, W., Wang, H. & Wan, Y. (2023). ECG Signal Acquisition and Processing System. *Journal of Physics:Conference Series*, *2644*(1).

[13] Ding, J., Tang, Y., Chang, R., Li, Y., Zhang, L. & Yan, F. (2023). Reduction in the Motion Artifacts in Noncontact ECG Measurements Using a Novel Designed Electrode Structure. *Sensors, 23*(2).

[14] Lazaretti, G. S., Teixeira, J. P., Kuhn, E. V. & Borghi, P. H. (2022). Android-based ECG Monitoring System for Atrial Fibrillation Detection using a BITalino® ECG Sensor. *In Proceedings of the 15th International Joint Conference on Biomedical Engineering Systems and Technologies (BIOSTEC 2022), 1*, 177–184.

[15] Jeon, T., Kim, B., Jeon, M. & Lee, B. G. (2014). Implementation of a portable device for real-time ECG signal analysis. *BioMedical Engineering OnLine*, *13*(1), 1–13.

[16] Nguyen, T. N., Duong, T. T., Omer, H. & Sulieman, D. A. (2023). The design and construction of a 12-channel electrocardiogram device developed on an ADS1293 chip platform. *Electronics, 12*(11).

[17] Jhang, Y. S., Wang, S. T., Sheu, M. H., Wang, S. H. & Lai, S. C. (2022). Integration design of portable ECG Signal acquisition with deep-learning based electrode motion artifact removal on an embedded system. *IEEE Access, 10*, 57555–57564.

[18] Su S., Zhu, Z., Wan, S., Sheng, F., Xiong, T., Shen, S., Hou, Y., Liu, C., Li, Y., Sun, X. & Huang, J. (2023). An ECG signal acquisition and analysis system based on machine learning with model fusion. *sensors, 23*(17).

[19] Samsudin, N. N., Isaak, S. & Paraman, N. (2022). Implementation of optimized low pass filter for ECG filtering using Verilog. *Journal of Physics: Conference Series, 2312*(1).

[20] Petráš, I. (2022). Novel generalized low-pass filter with adjustable parameters of exponential-type forgetting and its application to ECG signal. *Sensors, 22*(22),.

[21] Abidin, A. Z., Septian, B., Arkan, F. & Ridho, M. F. (2024). Noise suppression of ECG signal using optimized digital butterworth bandpass filter. *The Indonesian Journal of Computer Science*, *13*(4), 5254–5264.

[22] Rajendiran, D. K. J., Babu, C. G., Priyadharsini, K. & Karthi, S. P. (2024). Certain investigation on hybrid neural network method for classification of ECG signal with the suitable a FIR filter. *Scientific reports*, *14*(15087).

[23] Mishra A., Dharahas, G., Gite, S., Kotecha, K., Koundal, D., Zaguia, A., Kaur, M. & Lee, H. N. (2022). ECG data analysis with denoising approach and customized CNNs. *Sensors, 22*(5).

[24] [Jawahar, S., Harish, G., Harsha, V. S., Navialagan, P., & Preethi, D. (2023). Performance analysis of notch filter in ECG signal noise reduction. *International Research Journal on Advanced Science Hub, 5*(4), 137–141.

[25] Amini S. & Tazehkand, B. M. (2022). Design of feedback-structured IIR notch filter with transient suppression using gain variation. *Biomedical Signal Processing and Control, 71*(PA).

[26] Hesar H. D. & Hesar, A. D. (2025). Adaptive dual augmented extended Kalman filtering of ECG signals. *Measurement*, *239*.

[27] Karjadi, M., Salahuddin, N. S., Wibowo, E. P. & Afandi, H. (2016). Digital filter design of infinite impulse response (IIR) Infrasound to detect fetal heart rate. *International Journal of Engineering Research & Science, 2*(3), 25-38.

[28] Pan J. & Tompkins, W. J. (1985). A real-time QRS detection algorithm. *IEEE Transactions on Biomedical Engineering, 32*(3), 230–236.

[29] Utomo, T. P., Nuryani, N. & Darmanto. (2017). QRS peak detection for heart rate monitoring on Android smartphone. *International Conference on Science and Applied Science, 909*(1).

[30] Christov, I. I. (2004). Real time electrocardiogram QRS detection using combined adaptive threshold. *BioMedical Engineering OnLine, 3*(28), 1–9.
[31] Lewandowski, J., Arochena, H. E., Naguib, R. N. G. & Chao, K. M. (2012). A simple real-time QRS detection algorithm utilizing curve-length concept with combined adaptive threshold for electrocardiogram signal classification. *TENCON 2012 IEEE Region 10 Conference*, 1–6.
[32] Madeiro, J. P. V., Cortez, P. C., Marques, J. A. L., Seisdedos, C. R. V. & Sobrinho, C. R. M. R. (2012) An innovative approach of QRS segmentation based on first-derivative, Hilbert and Wavelet Transforms. *Medical Engineering & Physics, 34*(9), 1236–1246.
[33] Chen A., Zhang, Y., Zhang, M., Liu, W., Chang, S., Wang, H., He, J. & Huang, Q. (2020). A real time QRS detection algorithm based on ET and PD controlled threshold strategy. *Sensors, 20*(14), 1–15.
[34] Thurner, T., Hintermueller, C., Blessberger, H. & Steinwender, C. (2021). Complex-Pan-Tompkins-Wavelets: Cross-channel ECG beat detection and delineation. *Biomedical Signal Processing and Control, 66*.
[35] Lourenço, A., Silva, H., Leite, P., Lourenço, R. & Fred, A. (2012). Real time electrocardiogram segmentation for finger based ECG biometrics. *International Conference on Bio-inspired Systems and Signal Processing, 1*, 49–54.
[36] Thomas, A. (2021). *Enclaves in real-time operating systems* (Master's thesis, University of California). Electrical Engineering and Computer Sciences.
[37] Huang M. & Xiang, S. (2022). Image interpolation by direct cross downsampling. *Digital Signal Processing*, *132*.