Design of a nearly linear-phase IIR filter and JPEG compression ECG signal in real-time system

Article *in* Biomedical Signal Processing and Control · May 2021

DOI: 10.1016/j.bspc.2021.102431

8 authors, including:

Thang Bui Ngoc
Pukyong National University
13 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Sumin Park
Pukyong National University
10 PUBLICATIONS **3** CITATIONS
SEE PROFILE

Jaeyeop Choi
16 PUBLICATIONS **6** CITATIONS
SEE PROFILE

Byung-Gak Kim
Pukyong National University
10 PUBLICATIONS **8** CITATIONS
SEE PROFILE

Contents lists available at ScienceDirect

Biomedical Signal Processing and Control

journal homepage: http://ees.elsevier.com

Design of a nearly linear-phase IIR filter and JPEG compression ECG signal in real-time system☆

Ngoc Thang Bui[a], Thi My Tien Nguyen[b], Sumin Park[c], Jaeyeop Choi[a], Thi Mai Thien Vo[a], Yeon-Hee Kang[d], Byung-Gak Kim[e], Junghwan Oh[c,*]

[a] *Interdisciplinary Program of Biomedical Mechanical and Electrical Engineering, Pukyong National University, Busan 48513, Republic of Korea*
[b] *Department of Pediatrics at the Hospital for Tropical Diseases, 764 Vo Van Kiet Street, Ward 1, District 5, Ho Chi Minh City, Viet Nam*
[c] *Department of Biomedical Engineering, Pukyong National University, Busan 48513, Republic of Korea*
[d] *(BK21 Plus) Marine-Integrated Biomedical Technology Program, Pukyong National University, Busan 48513, Republic of Korea*
[e] *College of Future Convergence, Pukyong National University, Busan 48513, Republic of Korea*

ARTICLE INFO

Article history:
Received 1 September 2020
Received in revised form 30 December 2020
Accepted 16 January 2021
Available online xxx

Keywords:
ECG
IIR filter design
Nearly linear-phase filters
JPEG compression
Huffman compression

ABSTRACT

This paper presents the design of an Electrocardiogram (ECG) signal acquisition system based on 32-bit microprocessor platform. In hardware design, we present a complete design similar to the wearable devices used to monitor heart rates over the long term. The firmware is applied by an adaptive data compression method to increase storage space and reduces data transmission time between hardware devices and personal computer (PC). This adaptive algorithm is a combination of Joint Photographic Experts Group (JPEG) compression. The ECG signal was applied to the nearly-linear infinite impulse response (IIR) filter to eliminate noise from ECG signals. There are advantages of nearly-linear IIR filter over traditional IIR that the signal pass through filter is not distorted. Improvements in the JPEG compression algorithm make them suitable for one dimensional signals with a resolution greater than 8 bits and optimize the operation of the Huffman compression as well as the microcontroller unit (MCU) system with low processing speed. Placing the IIR filter at the end of the system promotes the ability to filter while reducing signal distortion after the decompression process. We tested the system with 8 records from the MIT/BIH cardiac arrhythmic database before and after we applied the compression algorithm to the firmware and software in the system that shows the effectiveness of these algorithms. The test results show that the system reached a maximum compression ratio of 8,8. The system deployed on the 32bit MCU (PIC32MZ2048), ECG analog front-end ADS1293 and operation time of ECG device up to 110 h (4.5 days).

1. Introduction

In recent years, the number of deaths related to heart problems has accounted for 31 % of all deaths worldwide [1]. The monitoring ECG signal is the most commonly used method for monotherapy. For the elderly, long term monitoring of ECG signals is an effective solution for detecting cardiac visual abnormalities [2–4]. However, ECG signal is often severely affected by noise from the environment, so effective solutions are needed to eliminate noise [5–7]. There are two main solutions to eliminating these noises: the first one is using noise filters based on hardware designed such as electronic components or signal cables with good noise filtering capabilities; the second one is used signal filters designed by software [8,9]. The method of filtering noise by software is easy to deploy and change when new requirements need to be implemented [10].

In addition, recording signals over a long period of time requires the device to be designed with high capacity memory. The bandwidth of the data link is usually limited especially with Internet of Things (IoTs) devices requiring low power consumption [11,12]. Applying data compression techniques will help solve this problem [13]. A lot of research on ECG data compression has been done. There are two compression methods: lossy compression and lossless compression [14–16]. The lossy compression method is mainly applied to clean, high-quality data to ensure data integrity. In this paper, we worked with a real-time system which collects heavily impacted ECG signals. Therefore, the lossy compression method is used to retain the main components of the signal and eliminate unimportant ones while also reducing the storage space and data transmission time of the signal. In addition, ECG signals include descriptions of various characteristics such as QRS waves, P waves, and T waves [17]. The recording of the ECG signal will ensure accurate identification of these characteristic segments. The IIR filter is applied to eliminate noise from the signal which distort the

☆ This work was supported in part by the Technology development Program (S2829803) funded by the Ministry of SMAs and Startups (MSS, Korea).

* Corresponding author.

E-mail addresses: buingocthang1984@gmail.com (N.T. Bui); bgkim@pknu.ac.kr (B-G Kim); jungoh@pknu.ac.kr (J. Oh)

https://doi.org/10.1016/j.bspc.2021.102431
1746-8094/© 2021.

signal, so we applied a nearly-linear IIR filter to eliminate noise and without distortion the ECG signal in this study [18–20].

Signal compression can allow longer time data to be stored in a small memory size. The ECG signal compression studies focus on lossy and lossless compression. The lossless compression method ensures signal integrity. However, the compression ratio is relatively low and quite complex, requiring a relatively large number of calculations. The lossy compression methods have a much larger compression ratio and often focus on retaining key components of the signal and removing unimportant components of the signal. The lossy compression is a way to permanently remove part of the ECG data. The lossy compression method is mainly based on Fast Fourier Transform (FFT), Discrete Cosine Transform (DCT) and Wavelet transform [21–25]. The DCT compression method is assessed to be simpler than the other two methods, so it is more suitable for MCU systems with low processing speed and resources. Normally, the JPEG algorithm mainly include data pre-processing, DCT and quantization. The compression method in this article is consist of some parts as data pre-processing and quantization. Because the JPEG method is primarily designed to handle 2D signals such as images, the proposed method that work on ECG data is a one-dimensional (1D) signal. Improvements in the proposed compression method focus on these three factors: 1). suitable for 1D signals; 2). reduce computation time during data compression; 3). increase the compression ratio of the algorithm.

There have been many studies about IIR and finite impulse response (FIR) filters. The FIR filters are more accurate than IIR filter but FIR filters require a large number of calculations and delay which are not suitable for sensor applications. In the study of author Nancy Betancourt [18] has described filters clearly. The study compared IIR and FIR filters in eliminating baseline wander and power line interference and the number of orders used between two filters. Another study was conducted by Gandham Sreedevi [19] for the removal noise in ECG signals using IIR and FIR filters. Research has shown the advantages of IIR filter over FIR in eliminating noise from ECG signal. The study of author Aleksandar Milchevski [20] also highlighted the advantages of IIR filter over FIR filter in removing noise from ECG signal. In addition, this study also highlighted the advantages of ring buffer technique in improving filter performance. The linear phase filter has the following characteristics: the filter response is a linear function of frequency. In this paper, we presented design of IIR filters with linear phase based on the classic IIR filters. The time reversal method is a simple method to linearize the phase of IIR filters without changing the filter structure [26]. With this method, data is divided into blocks which are processing data. However, the filter is suitable for digital signal processing systems because of its simple structure and easy to deploy in real-time systems. The ECG signal after applying IIR filter and nearly-linear IIR filters is shown in Fig. 1.

The system proposed in this article has mainly three contributions. First, the IIR linear phase filters are applied to filter noise without distortion the ECG signal [26]. Band-pass filters with a bandwidth of 0.05 Hz–60 Hz are designed and applied in this study. Second, JPEG compression are applied to reduce the storage capacity of the ECG signal and transmission bandwidth so as increase patient follow-up time and to reduce design costs and complexity [16,27]. The data is stored in blocks, with each block containing data for 10 s with a sampling frequency of 356 Hz and a 24-bit resolution [28]. Finally, the above techniques, primarily applied to powerful computing platforms such as PCs are included in the MCU series, which are more suitable for compact, low-power design requirements.

The remainder of this paper is organized as follows. Part II describes the data processing method. In Part III, we described a system overview and design hardware. In Part IV, we presented system implementation. Part V presented the evaluation results. Conclusions are drawn in Part VI.

2. Proposed data processing method

The ECG signal models are described in Fig. 2. The implementation steps of two algorithms are closely related to create the maximum per-

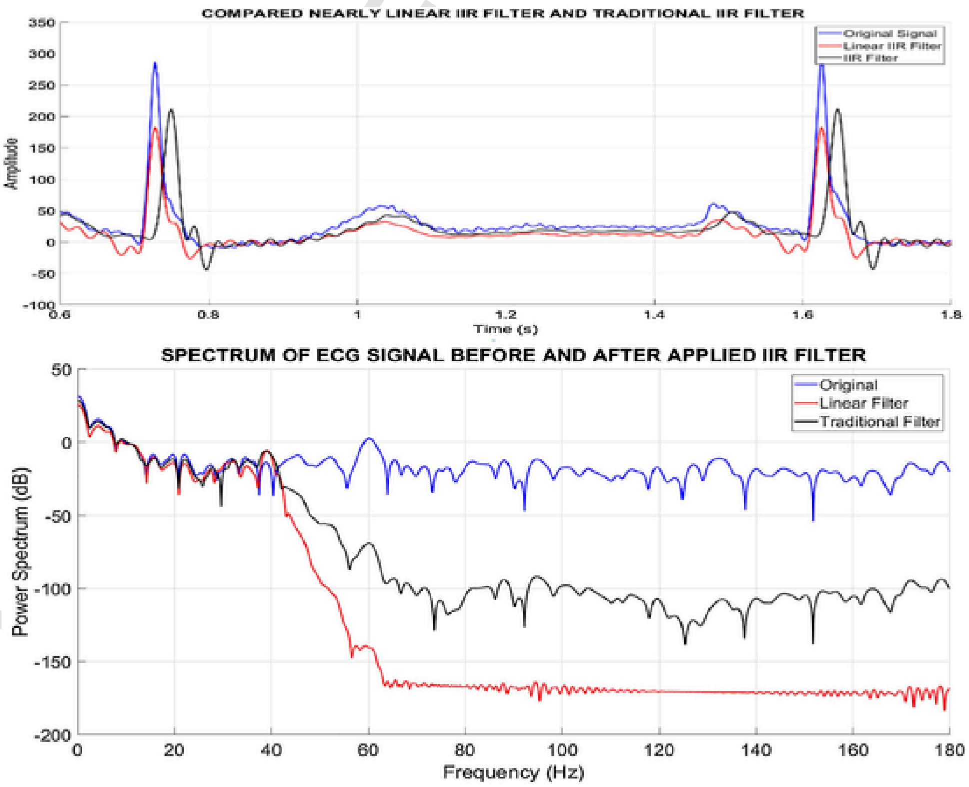


Fig. 1. ECG signal before and after applying LP IIR filter.

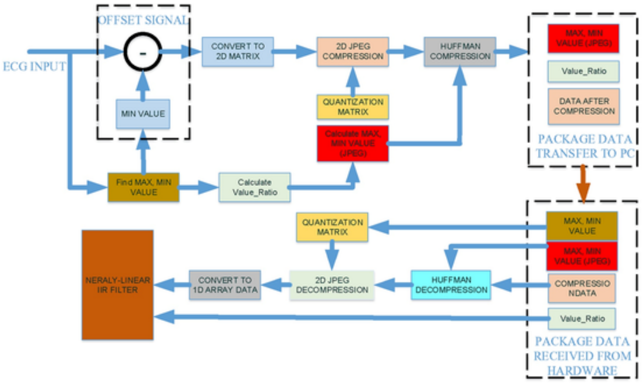


Fig. 2. Compression and decompression method.

formance of the algorithm and to optimize the operating time of processor. The process is carried out through two main steps.

•First, the system divided the ECG signal into frames of a fixed length 1D. Then the frames are convert the 2D matrix of the ECG data to match the quantization matrix of the JPEG method.

•Second, the ECG data need to be converted into an executable format using JPEG compression. The JPEG compression accept positive input data in the range (0–255) 8-bit or (0–4095) 12-bit [29], while ECG data usually has a larger value (0–2047 for MIT-BIH data with a 11-bit resolution [30]). Improvements in the with value range (0–16,777,215) 24-bit resolution of ECG data. The values of data are too large directly affect the response time of the system. So, the system is necessary to have a suitable solution to reduce the computation time of the data compression algorithm. Thereby avoiding data loss or system crashes with systems are operated in real time. The JPEG is a lossy compression algorithm, so reducing the range of data also increases the compression ratio of the algorithm. However, due to the influence of the Right-Leg Driven (RLD) signal, the ECG signal is offset a big value. The important components of the ECG signal are determined in the (Min-Max) range of the signal. The (Min-Max) values is described in Fig. 3. Therefore, the ECG signal needs to be removed after offsetting to reduce the computation time of the system.

2.1. Proposed compression algorithm

The energy of ECG signals is mainly concentrated in the frequency which ranges from 0.1 Hz to 100 Hz, but the signals in the higher frequency range are usually not included in the ECG signal [11].

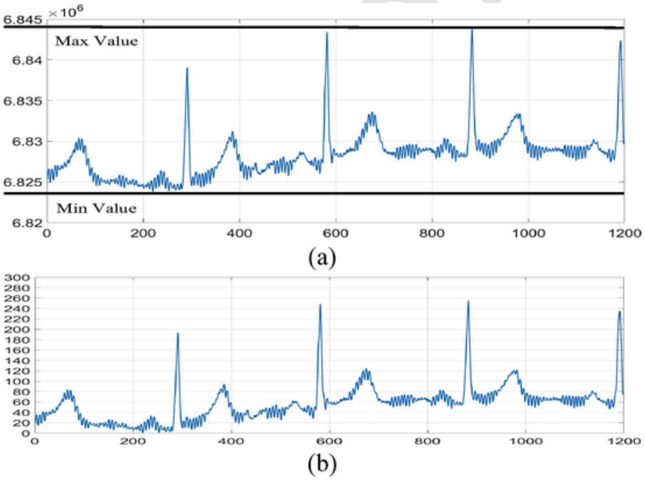


Fig. 3. ECG signal before (a) and after (b) offset value and scaled data.

The ECG signals collected from devices often have an offset, so the value range of signal is raised. Facilitating the calculation and reduction of CPU uptime only needs to be done with the actual values of the ECG signal. In the first step, the algorithm finds the max value of the signal, then performs the offset signal according to the formula and Fig. 3 below.

In the process of implementing data compression algorithms, it is necessary to perform multiplication operations on matrices (DCT process), and find the value (max, min) of the data series. Therefore, the value range of the data directly affects the time that takes to perform calculations on the CPU. The input data is scaled into the range of values (0, 255) to ensure CPU execution time. In the test section, we evaluated the uptime of the CPU with the value range of input data.

Next, we found the max and min values of the data after performing JPEG compression. These values are important in optimizing the number of loops in Huffman compression method. However, by finding this value (Max, Min) can be done at the time of reading data from ADS1293, this reduces the operation time of the algorithm. Data after Huffman compression is packaged into blocks and transmitted to the PC. Data blocks include the maximum and minimum values of the original data and JPEG compression data. The compression ratio depends quite a lot on data and the size of the data. The scanning of the data of Huffman encryption and decryption is performed according to the Zigzag method [30]. The algorithm implementation process is described by us in the following steps:

```
Step 1: Read data from ADS1293, compare with (max, min) and update (max, min).
Step 2: Offset block data: data = data – min. Calculate Ratio_val = (max-min)/255.
Step 3: Re-caculate block data: data = data * Ratio_val.
Step 4: DCT calculate.
Step 5: Huffman compression.
```

2.2. Designed nearly linear-phase IIR filter for ECG system

2.2.1. Introduced time reversal method

The time-reversing method is intended to take the signal over time, then play it back. However, the method is only applied to signals described by the time invariant equation [31]. In this paper, we used the time reversal method to linearize the ECG signal after implementing the IIR filter. The process is also divided into two steps.

- The first step is decoding the ECG signal with a length of N samples (e.g. N_sample = 3600 samples), then dividing signal into M blocks (e.g. M = 5, each block has N = 720 samples) data to perform the filter IIR for each block. Additionally, at the end of each data block, additional N_overlap overlapping samples (e.g. N_overlap = 180) are selected.
- The second step, involves performing a traditional IIR filter with N + N_overlap samples. After completing the IIR filter, we perform a time inversion with the start time as the end of the N + N_overlap samples in order to obtain N_overlap samples with a linearized phase. The implementation process of these steps is described in detail in Fig. 4.

In this method, the linearity of the signal after the time reversal depends on the number of overlapping samples. However, the large number of overlapping samples increases the filter delay. The filter delay is described by the formula below.

The linear phase (LP) IIR filter delay (N_d) in samples equal [32]:

$$N_d = 2*(N + N\_overlap\ ) \tag{1}$$

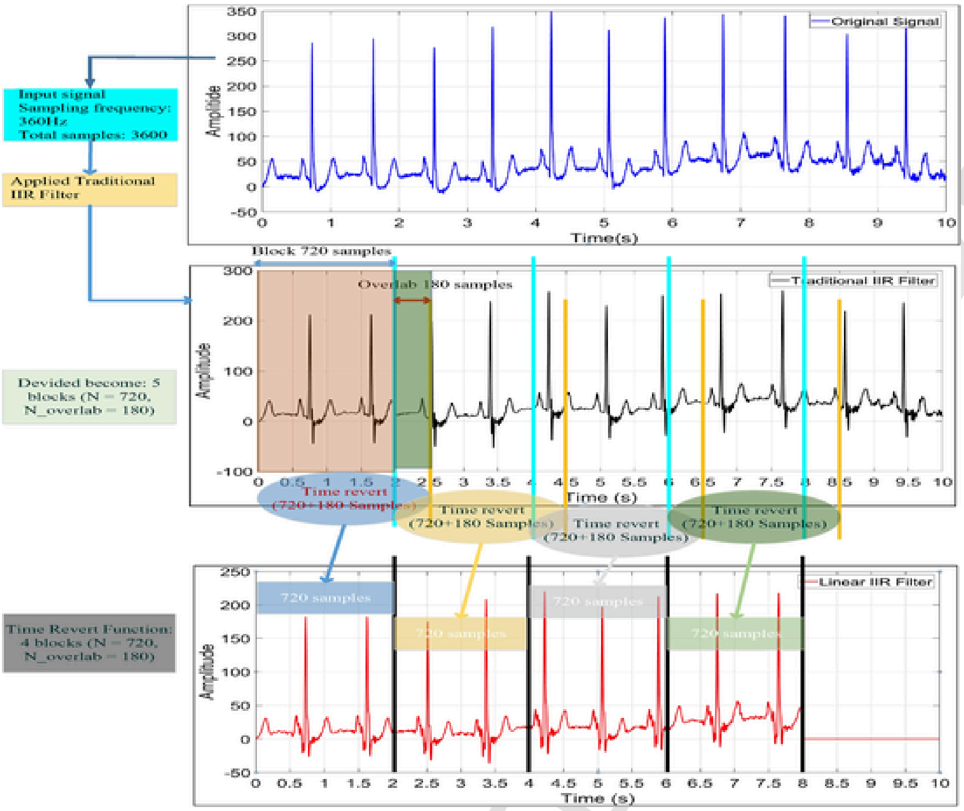


Fig. 4. Sequence of Nearly Linear IIR Filter.

2.2.2. *Chose the method for nearly-linear IIR filter*

There are mainly two types of interference and two types of noise that have the most impact on ECG signals: muscle contraction noise due to a breathing rate of less than 0.05 Hz and the noise affected by the power interference (50/ 60 Hz). Therefore, we designed a band-pass filter in the bandwidth from 0.05 Hz to 60 Hz [20,33] for removing noise form ECG signals.

In Table 1, we compared the properties of each method applied to IIR filters.

Based on the comparison results in the Table 1, we can see the Elliptical IIR filter have order number smaller than other methods. Thus, we chose the Elliptical method for design implementation on our system. Although, the filter is implemented via a software, with the Elliptical method, we can easily implement this filter effectively on the firmware for MCU [10].

3. System overview and design

The overall system of the ECG signal acquisition system is shown in Fig. 5. The overall system design can be divided into four different sections: system overview, hardware design, firmware program, and software display ECG signal.

Table 1
Comparison of Band-with IIR Method.

Type	Pass band	Stop band	Roll-off	Step response
Butterworth	Flat	Monotonic	Poor	Good
Chebyshev 1	Rippled	Monotonic	Good	Poor
Chebyshev 2	Flat	Rippled	Very good	Good
Elliptical	Rippled	Rippled	Best	Poor

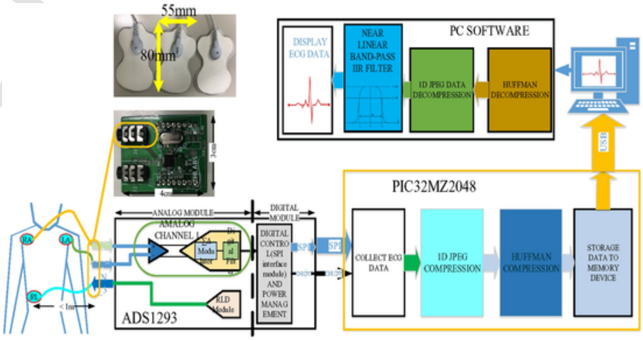


Fig. 5. Overall ECG system.

3.1. *System overview*

In the proposed scheme, the electrodes are affixed to the patient's body, then connected to device via a wire cable. The ECG device is divided into two parts: the ECG front-end chip is used to convert the analog signal to digital and processor chip is used to process signals and transmit to the PC via UART communication which used conversion cable USB2COM. On the PC, the software receives signals through communication with the COM port, decompresses the signals, filter and display results.

The ECG signal acquisition and processing module is built on two main chips, ADS1293 [28] and PIC32MZ2048 [34]. The front-end ECG chip has a 24-bit resolution and RLD module which eliminates common-mode and increases signal acceptability. The main processor chip PIC32MZ has a speed of up to 200 MHz capable of implementing some algorithms in real-time. In addition, PIC32MZ can also save data to external USB devices.

Software on PC collects processing and displays ECG signals every 10 s, and the signals are displayed in real-time for easy observation.

3.2. Designing analog to digital module of ECG system

In this study, we proposed the analog front-end design as shown in Fig. 6 using a front-end ECG chip as ADS1293. The central processing chip is the PIC32MZ2048 which collects data about ECG signals from the ADS1293 chip through Serial Peripheral Interface (SPI). The DRDYB signal of the ADS1293 chip determines the sampling frequency of the main chip, the sampling frequency can be configured from 50 Hz to 2133 Hz.

Therefore, the ECG signal may be affected by noise and common-mode signal. To overcome this phenomenon, Right-Leg Driver circuit is designed for ADS1293 chip. The design of the RLD circuit is described in Fig. 6 [28,35].

With the RLD circuit design described above on the ECG signal before entering the ADC module in the ADS1293 chip as follows Fig. 6.

The ADC value of the ECG signal is calculated using the following formula [28]:

$$ADC_{OUT} = \left[\frac{3.5(V_{INP} - V_{INM})}{2{*}V_{REF}} + \frac{1}{2}\right]{*}ADC_{MAX} \quad (2)$$

where $V_{REF} = 2.4V$, ADC_{MAX} depends on the sampling frequency of ADS1293.

4. System implementation

In this section, we described the process of hardware execution as well as firmware programming for the ECG device in this article. The main goal is to design a wearable device that collects ECG signals in real time. We focus on the main characteristics as follows. Ability to work continuously and collect ECG signals in real time, avoiding the possibility of data failure. To accomplish those two goals we presented quite in detail about the hardware. We design the important modules in the firmware. In terms of hardware, we chose 32-bit processors with fast processing capabilities to be able to perform operations in data compression algorithms. The firmware should be able to handle the data carefully in order to avoid possible conflicts between data collection, processing and transmission. The proper distribution of the data in the firmware by using the ring buffer technique is effectively applied in this firmware.

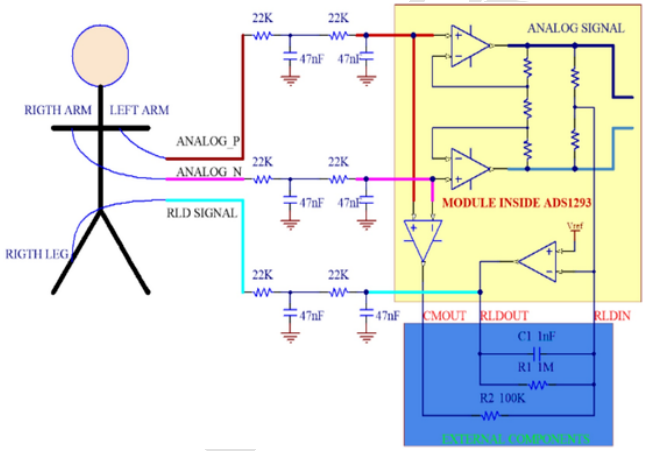


Fig. 6. RLD design and external components.

4.1. Implementation embedded system platform

We implemented a hardware design that used the algorithms described above. Hardware details are depicted in the Fig. 7 and Table 2, including three main blocks: power block, MCU block, and ECG analog front-end block. The power unit includes battery charging circuit and 3.3 V generator circuit. The power block uses 5 V input power supply. When an external 5.0 V power supply is disconnected, the 3.7 V (full chage around 4.2 V) battery will generate 3.3 V and 5.0 V power supply for the whole system. The system works with 18,650 battery that can operate continuously for 110 h. Restore processing using ic PIC32MZ2048 performs algorithm and transfer data to PC via USB2COM (FT232). The ECG analog front-end block uses the ADS1293 with serial peripheral interface (SPI) to transmit data to the PIC32MZ2048 chip. The highlight of this design is that the ADS1293 has a built-in RLD circuitry to remove noise from power interference. The test details are presented in the section below.

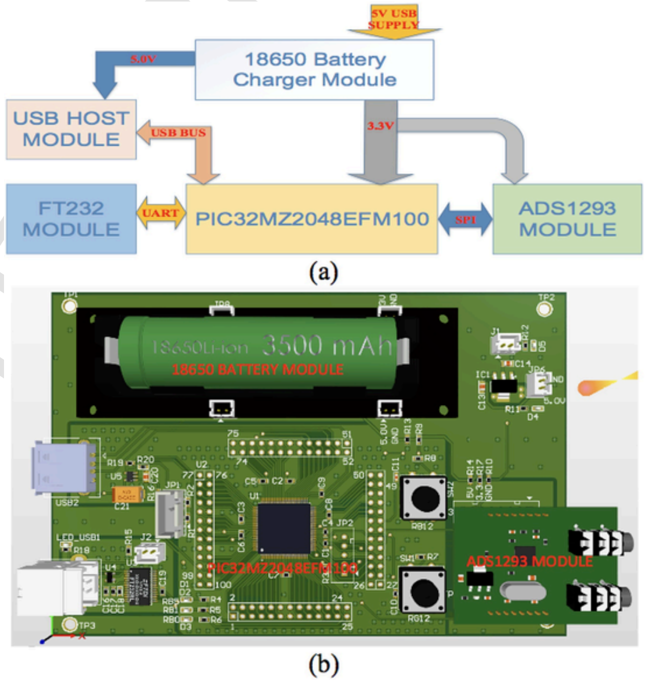


Fig. 7. (a) Block Hardware design based on PIC32MZ, (b) 3D model of hardware implement on Altinum Designer.

Table 2
Pin connection between PIC32MZ and other modules.

Module	Signal Name	Pin of PIC32MZ	ADS1293	FT232	USB TYPE A
ADS1293 MODULE	CS	81	16		
	SCK	76	17		
	SDO	78	18		
	SDI	47	19		
	RST	21	25		
	DRDY	39	20		
FT232	TXD	7		5	
	RXD	8		1	
USB HOST	D +	55			3
	D-	54			2

4.2. Implementation firmware for Pic32MZ

Implementing firmware for PIC32MZ to work with 1D JPEG and Huffman compression is one of the most important tasks in our study. The maximum speed of PIC32MZ is 200 MHz, equivalent to 330MIPS. The modules in the firmware are shown in Fig. 8.

Firmware is implemented with three main modules. The first is to collect ECG data from the front-end ECG chip via SPI protocol, the second is the JPEG compression module, and the last module's function is transfer data to the PC.

4.3. ECG data collection module

The operation description of data collection module is described based on flow chart in Fig. 9. The module is responsible for collecting ECG signals based on the external interrupt pin of PIC32MZ.

Because the design system runs in real-time, the sampling process that filters, compresses data, and communicates with the PC is done sequentially and continuously; we used a ring buffer structure to show the module, as above. Details of the buffer structure are shown in Fig. 10. The ring buffer technique helps the process of reading, processing and transmitting data sequentially and avoids blanking that leads to data errors.

of MCU starts to read data from ADS1293 chip after receiving the DRDYB signal. The initial sequence of the ADS1293 and the process of reading 24-bit data from channel 1 of the ADS1293 are described in Fig. 11. CPU sends 0 × 50 data to ADS1293 chip to read 3 bytes in the ADS1293 buffer. The results are shown below.

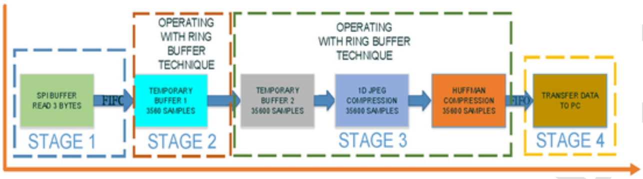

OPERATING WITH RING BUFFER TECHNIQUE
OPERATING WITH RING BUFFER TECHNIQUE
SPI BUFFER READ 3 BYTES
FIFO
TEMPORARY BUFFER 1 3560 SAMPLES
TEMPORARY BUFFER 2 35600 SAMPLES
1D JPEG COMPRESSION 35600 SAMPLES
HUFFMAN COMPRESSION 35600 SAMPLES
FIFO
TRANSFER DATA TO PC
STAGE 1
STAGE 2
STAGE 3
STAGE 4

Fig. 8. Sequence of all module working on PIC32.

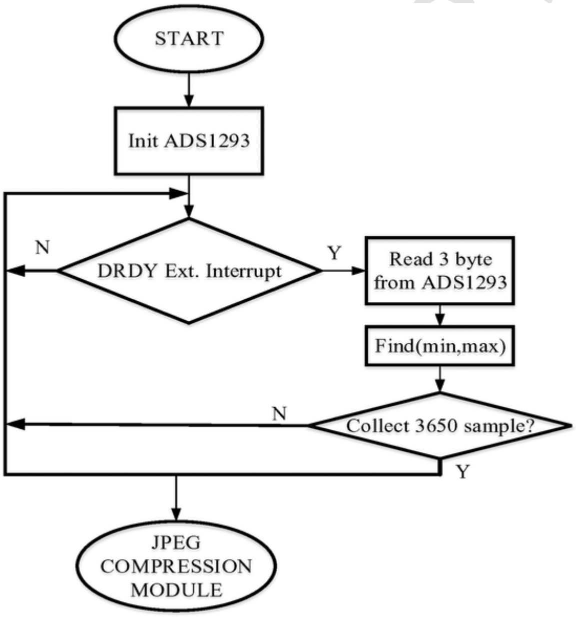


Fig. 9. Flowchart of ECG data collection module.

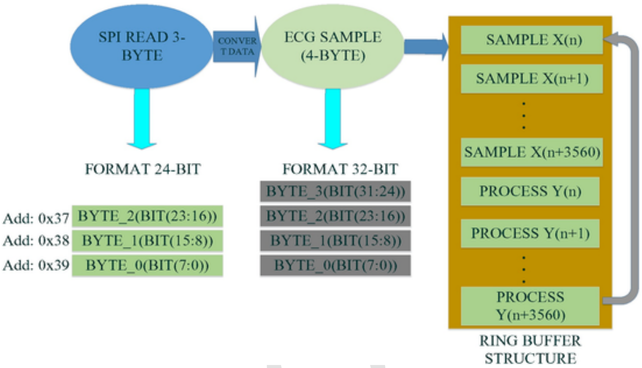
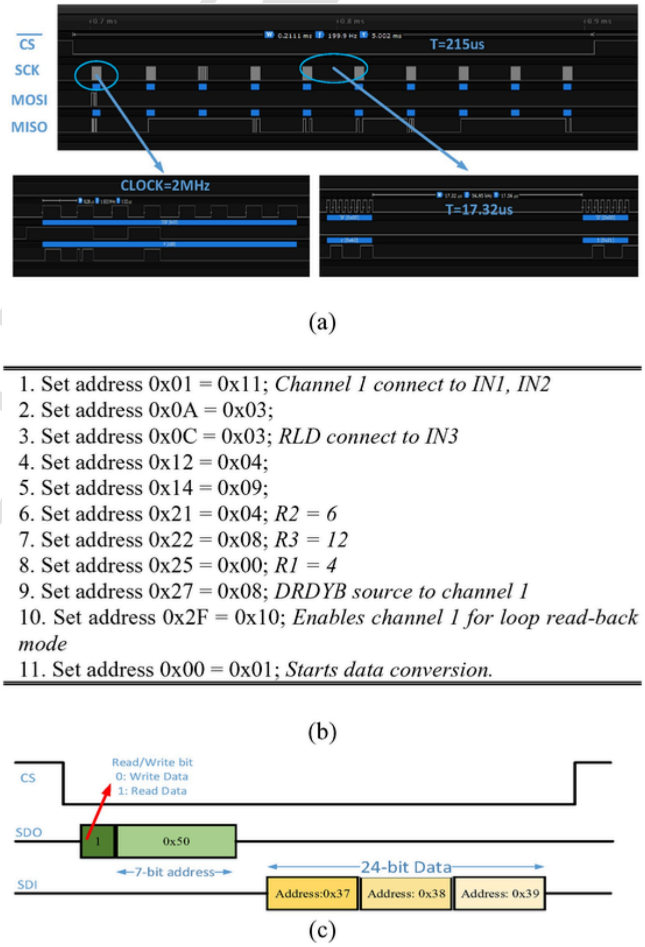


Fig. 10. Ring buffer structure with ECG sample and data after processing.We configured the clock of the SPI module to 2 MHz and operated in mode 0. The CPU



1. Set address 0x01 = 0x11; *Channel 1 connect to IN1, IN2*
2. Set address 0x0A = 0x03;
3. Set address 0x0C = 0x03; *RLD connect to IN3*
4. Set address 0x12 = 0x04;
5. Set address 0x14 = 0x09;
6. Set address 0x21 = 0x04; *R2 = 6*
7. Set address 0x22 = 0x08; *R3 = 12*
8. Set address 0x25 = 0x00; *R1 = 4*
9. Set address 0x27 = 0x08; *DRDYB source to channel 1*
10. Set address 0x2F = 0x10; *Enables channel 1 for loop read-back mode*
11. Set address 0x00 = 0x01; *Starts data conversion.*

Fig. 11. (a) SPI signal and processing time, (b) Sequence of initial ADS1293, (c) Sequence read 24-bit data of channel 1.

1. Set address 0 × 01 = 0 × 11; *Channel 1 connect to IN1, IN2*
2. Set address 0 × 0A = 0 × 03;
3. Set address 0 × 0C = 0 × 03; *RLD connect to IN3*
4. Set address 0 × 12 = 0 × 04;
5. Set address 0 × 14 = 0 × 09;
6. Set address 0 × 21 = 0 × 04; *R2 = 6*
7. Set address 0 × 22 = 0 × 08; *R3 = 12*
8. Set address 0 × 25 = 0 × 00; *R1 = 4*
9. Set address 0 × 27 = 0 × 08; *DRDYB source to channel 1*

10. Set address 0 × 2F = 0 × 10; Enables channel 1 for loop read-back mode
11. Set address 0 × 00 = 0 × 01; *Starts data conversion.*

4.4. Data compression module

The data compression module includes JPEG compression and Huffman compression; JPEG compression has a more complex calculation than Huffman compression. Although they are both implemented based on the computing power of the CPU. In the firmware, data compression is performed for each block of data (3560 * 8 samples) corresponding to data collection time of 10 * 8 = 80 (seconds). The process for performing data compression is described in Fig. 12.

4.5. Display software

The data display software in this paper is implemented using MATLAB software. After receiving the compressed data transmitted via USB2COM port, the software will extract and interpret data. The decompression and filter algorithm is presented in Fig. 13 below.

Data processing software consists of two main modules that are data decompression module and band-pass IIR filter. The data decompression module works similarly to the data compression module described in the firmware. The band-pass IIR filter module is implemented with the same ring buffer technique as the firmware.

The last part of the data display software is the display module with the interface shown in Fig. 14.

4.6. Data parking format for transfer

On the other hand, to facilitate the reception of data transferred from hardware to PC and processed later. We export the data frame according to the model described in Fig. 15. In this frame, the values needed for the decompression process are transmitted directly from the hardware to help save time for the decompression and signal recovery process.

5. Experimental and results

After completing worked with the hardware, firmware, and software, we conducted tests to evaluate the effectiveness of this system. The test was carried out in four parts: Part 1 evaluated the viability of hardware in this design. Part 2 evaluated the data compression capabilities of the algorithms to be exported above when performing on the firmware. Part 3 experimented with ECG signals to evaluate filter performance and JPEG data compression modules as well as signal accuracy after filtering and decompressing signals with BIT/ MIT. Part 4 measured ECG signals on volunteers to assess the accuracy and the response of the filter in real time. The hardware settings for these tests are depicted in Fig. 16. The firmware was written by the software suit MPLAB X v5.2 with the compiler MPLAB XC32 v2.05 and the MPLAB Harmony v2.06.

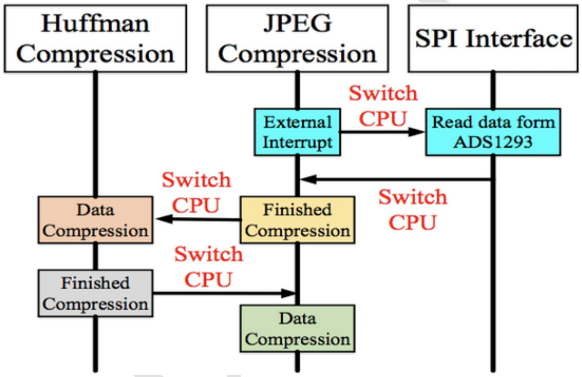


Fig. 12. CPU Task with Data Compression.

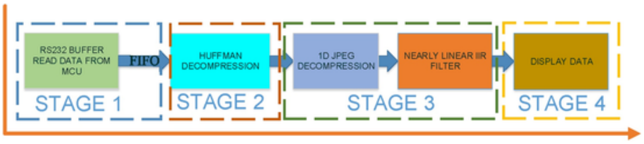
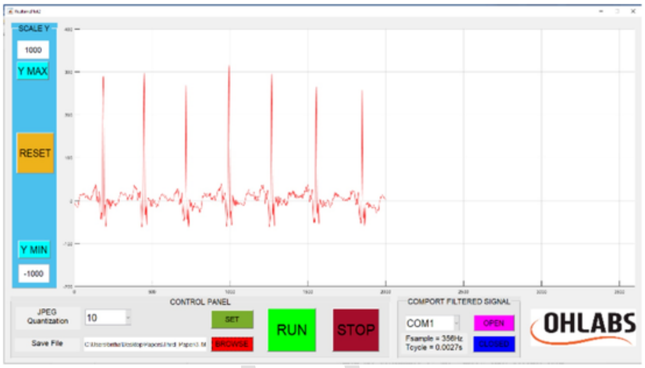


Fig. 13. Sequence of all module working on display software.



1000
-1000
JPEG Quantization 10
Save File
COM1
Fsample = 356Hz
Tcycle = 0.0027s

Fig. 14. Software display real-time ECG signal.

Start Frame	Ratio_val	Max_val (JPEG)	Min_val (JPEG)
{	3 byte	3 bytes	3 bytes
Data after Compression		Check Sum	End Frame
		1 byte	}

Fig. 15. Data Parking Format.

5.1. Performance evaluation of the hardware

After completing the hardware design, soldering the components to the circuit board as well as fully implementing the modules in the firmware, we conducted a review of the design of the board above. Fig. 16 describes the full range of components in this system. In this test, we used the Agilent Technologies U3606A (Multimeter and DC Power Supply) to generate a 3.3 V voltage to the device and measure the current consumption of the circuit board. In normal working mode, the ECG device consumes a current about 30 mA. In addition, the ECG device is designed for using with ion rechargeable battery (18,650 Li-ion) with a capacity of 3500 mA h, which means that the device can be used continuously for 110 h (4.5 days).

5.2. Performance evaluation of the firmware

The fraction of the compressed signal size of the original signal size is called the compression ratio (*CR*). It delivers all the information and ignores the unnecessary data. By reducing the *CR* ratio, the data bits required for storing or transmitting are obviously reduced:

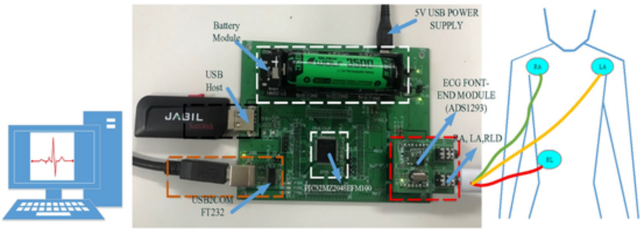


Fig. 16. Real system for monitoring ECG signal.

$$CR = \frac{Original}{Compression} \tag{3}$$

where *Original* denotes the bits used in the original data and *Compression* denotes the number of bits after compression.

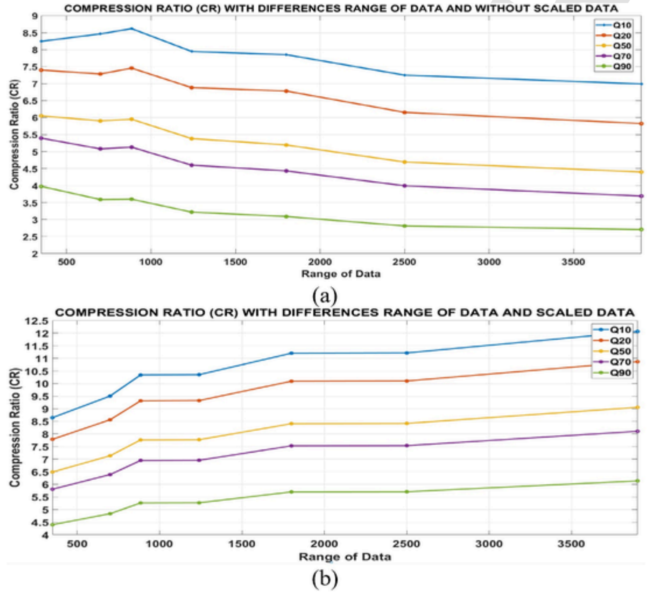
As we mentioned above, the firmware implemented on the PIC32MZ series has limited resources and data processing capabilities. In this section, we assess the performance of the firmware with different data. The results of this evaluation are important in selecting other MCU lines that can replace PIC32MZ in other designs. Our test includes evaluating the compression time data of compression ratio with different data ranges. We used input data including 3600 sample ECG signals (10 s). The result is described in Fig. 17, Fig.18.

Based on the results described in Fig. 17. We can clearly see that the CR depends on the range of the data. If the value of the input data does not return to the range (0, 255), then increasing the value of the data will decrease the compression ratio. On the other hand, the value of input data is about (0, 255), then increasing the value of the data will increase the compression ratio.

Based on the results described in Fig. 18. We can clearly see that the operating time of MCU depends on the range of the data. If the value of the input data does not return to the range (0, 255), then the system only works with range data (0, 700). On the other hand, the systems do not work stably with a larger range of input values. However, when the quantization value is increased, the CPU uptime is significantly increased, especially with the Q90 values. Because during the Huffman compression algorithm implementation, too many loops need to be performed, the DCT execution time only includes operations performed on the matrix. Therefore, the process of finding the max-min value of the data series before performing compression algorithm is performed at the process of collecting ECG data from ADS1293 chip. This process is described in Fig. 9.

In Table 3 below, we compared the bits of input data before and after performing the data compression algorithm.

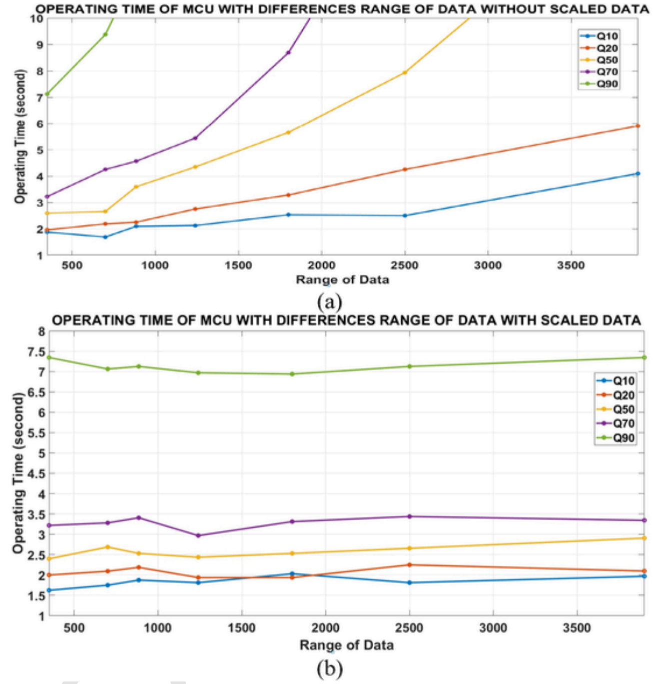
Based on the data in Table 3, we can clearly see the number of bits after implementing data compression algorithm as constant for different quantization values. This is important in ensuring the bandwidth of the transmission line as well as reducing the data transmission time.



(a)

(b)

Fig. 17. Compared CR with differences range of data: (a) Data without scaled data to range (0, 255), (b) Data with scaled data to range (0, 255).



(a)

(b)

Fig. 18. Compared operating time of MCU with differences range of data: (a) Data without scaled data to range (0, 255), (b) Data with scaled data to range (0, 255).

Table 3
Comparison Number of Bits Before and After Compression.

Range of Data	Original (bits)	Q10 (bits)	Q50 (bits)	Q90 (bits)
350	36,115	4178	5566	8211
700	39,715	4178	5566	8211
900	43,201	4178	5566	8211
1240	43,252	4178	5566	8211
1800	46,801	4178	5566	8211
2500	46,852	4178	5566	8211
3900	50,401	4178	5566	8211

5.3. Evaluation of compression method

In the second test, we measured the ECG signal on the volunteers and compared the signal obtained from the hardware with the software decoded signal. To get the ECG signal of the device before performing the compressions module, we used the ICD4 debugger tool in conjunction with IDE MPLABX v5.2 to directly observe variables in the firmware that we have implemented. These variables are saved as files and redrawn (original signal) using MATLAB v.2018 software. In addition, these signals also apply linear phase filter IIR (filtered signal). The LP IIR structure is implemented on MATLAB with bandwidth structure (frequency band 0.05 Hz–40 Hz) in order of 10, and Astop is 80 dB and Apass of 3 dB, sample weight is 720 samples and superimposed cross sample content of 180 samples.

Testing the components of the firmware including performance evaluation and usage resources for each module. The main modules in the firmware include: data acquisition module from ADS1293, quantum compression DCT, Huffman data compression module and data transfer module on PC.

Fig. 19 described the LA-RA signals we made using this method. These signals include those directly sampled from volunteer signals after passing through the compression/ decompression module with dif-

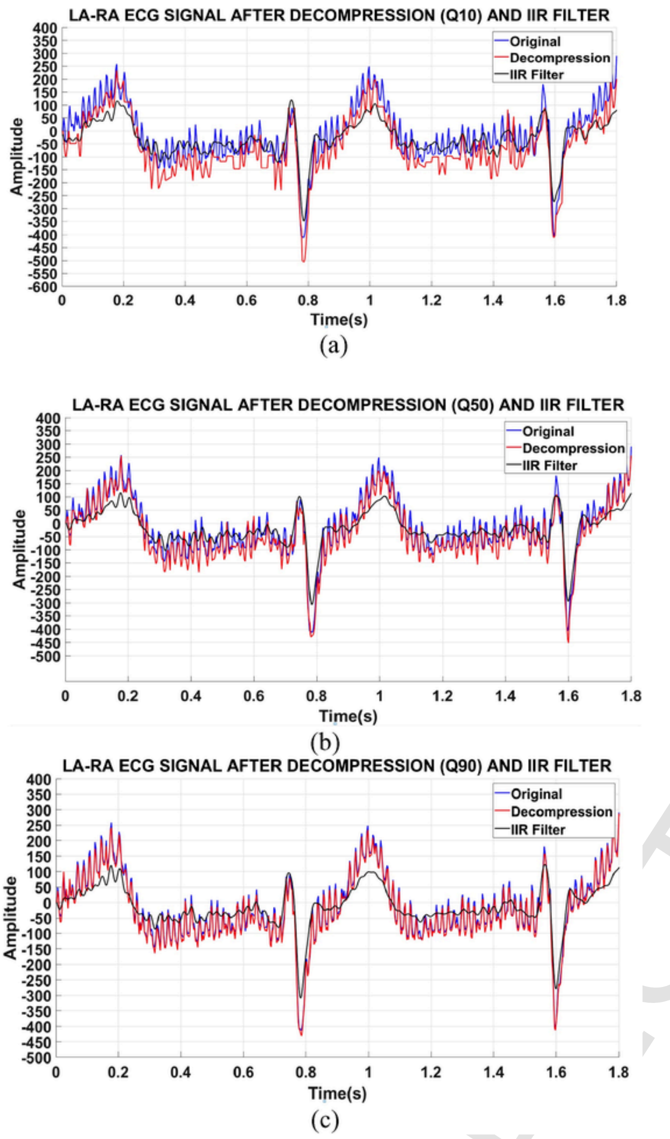


Fig. 19. Volunteer ECG signal with differences quality: (a) Quality = 10, (b) Quality = 50, (c) Quality = 90.

ferent quality coefficients and signals after going through compression/ decompression and IIR filters.

Based on the signal in Fig. 19 (a, b, c), we see a big difference between the red signal (after decompressing JPEG with different quality coefficients (Q10, Q50, Q90)) and the green signal (original signal). The black signal (after extracting JPEG with different quality coefficients and performing IIR filter) is not very different from different quality coefficients (Q10, Q50, Q90). Therefore, in this design, we can use it effectively with a low-quality coefficient to increase data storage space.

In addition to assessing the accuracy of the signal as well as the noise filtering ability of the IIR filter, we conducted a calculation of the signal to noise ratio for the signals presented above. The filter design process removes signals with frequencies less than 0.05 Hz and frequencies greater than 40 Hz. All signals within the frequency range of 0.05 Hz–40 Hz are considered ECG signals. Signal to Noise Ratio (SNR) is calculated according to these parameters. The result is described in Table 4.

Table 4
Compared SNR between Original, Compression/ Decompression and IIR Filter with Real-Tested Data.

JPEG Quantization	10 %	50 %	90 %
SNR Original	9.67	9.67	9.67
SNR Decompression	3.62	3.63	3.65
SNR IIR Filter	384.89	314.32	327.32

Based on the data in Table 4, we can clearly see that, after implementing the distorted signal data compression algorithm, there is an increase in SNR ratio. The IIR filter has eliminated most of the noise, so the SNR ratio increased about 40 times higher than the original signal.

On the other hand, we also conducted data tests from the BIT/MIT [36] database with a total of 8 records (NSR # 100 -> # 107) with a modern compression algorithm which is lossless compression (a lossless compression method based on Golomb-Rice Coding) [13]. Both methods of data compression are implemented on MCU platform. The comparison results are described in Table 5.

Based on the results described in Table 5, we see that the JPEG method has an advantage over the lossless compression method because of the higher compression ratio. Data from the BIT/MIT facility is highly accurate and relatively clean. For signals that do not require high precision, the JPEG compression method is more effective.

To evaluate the error of the JPEG compression algorithm, we conducted experiments with 3600 samples from the MIT/BIH database with different quality coefficients. The error shown in the Table 6 is the average error after unpacking of these 3600 samples.

Based on the data in Table 6, we can see that the error of the signal with a 90 % quality coefficient (Q90) is only about 9% compared to the original signal. The error increases to 25 % if the quality coefficient drops to 10 % (Q10). For standard signals, this is a relatively large error that distorts the signal. However, in our design, the ECG signal is heavily affected by noise, so this ratio does not affect the actual signal but aims to eliminate the noise in the ECG signal.

In addition, we also evaluated the average error of data after performing IIR filter and data after decompression and implementation of IIR filter with a total sample of 3600 taken from MIT/ BIH database.

The data in Table 7 shows that the error of the ECG signal after performing decompression and the ECG signal after decompression and implementation of the IIR filter is not too large. The amplitude and key components of the ECG signal are fee.

Table 5
Compared Compression Ratio between JPEG and Lossless Compression Method.

Record	Ref method [13]	CR(Q10)	CR(Q50)	CR(Q90)
100	2.972	8.644	6.489	4.4
101	3.202	8.86	6.61	4.431
102	2.879	8.634	6.38	4.225
103	2.826	8.513	6.1	4.212
104	2.736	8.771	6.57	4.419
105	2.86	8.8	6.81	4.678
106	2.697	8.374	6.22	4.132
107	2.556	8.572	6.429	4.343

Table 6
Average Error of Data before Decompression.

JPEG Quantization (Q)	10 %	20 %	50 %	70 %	90 %
Average Data Error	25.974	21.627	17.161	10.884	9.378

Table 7
Average Error of Data before and after Decompression and IIR.

JPEG Quantization	10 %	20 %	50 %	70 %	90 %
Average Data Error before JPEG(%)	15.912	15.912	15.912	15.912	15.912
Average Data Error after IIR(%)	21.556	22.694	21.025	17.997	18.573

6. Conclusion

This article has introduced an improved compression method that explores the potential features of the JPEG method to compress ECG signals. The comparison results show the strength of the system in creating a very high CR of 8.8, especially since, depending on the signal quality, we can change the CR to avoid signal distortion. Based on the results of error evaluation in Table 6, we see that the final signal of the system does not have too many errors (9.3 %) compared to the uncompressed signal. The error of the signal after decompression does not exceed 10 % compared to the original signal. Therefore, a new set of quality matrices or strategies is required to address the low energy part of the ECG signal.

One of the main contributions of this study is to modify the 1D (ECG signal) signal in a way that makes it suitable for 2D compression algorithms such as JPEG. The modification consists of two phases: First, the 1D ECG signal is divided into frames of a fixed length and the arrangement of the frames is accepted in 2D. Second, ECG signal data often has an abnormal signal amplitude, so it is important to estimate the max and min values of the signal over a period of time to adjust accordingly JPEG compression. There are coefficients of JPEG and Huffman compression algorithms that are recalculated and optimized to achieve the best performance in terms of execution time and in accordance with the fluctuations of ECG signals in real time. The improvements to the JPEG compression algorithm have increased the compression ratio, reducing algorithm execution time.

Another contribution is the LP IIR filter which is also suitable for interpolation. The calculation efficiency is greatly improved compared to the use of FIR filter. The linear phase filter also shows its effectiveness when the ECG signal after the filter has not distorted and still accurately identifies the QRS, T wave and R wave. In addition to reducing signal distortion after passing the filter, we can adjust the number of overlap samples; however, this will increase filter delay and duration. One of the other important applications of LP IIR filters is that it can be applied to other types of 1D signals such as audio signals to increase the performance of the signal processing device.

Real-time ECG signal acquisition is based on simple devices and low noise resistance. The process of compressing and filtering noise has great significance in reducing the effective signal storage space, reducing data transmission time and eliminating noise to make the signal clearer to observe and evaluate. Important information via clean and undistorted ECG signals are important in the diagnosis of cardiovascular problems.

Furthermore, a PIC 32bit-based MCU was used to build a programmable embedded platform for signal processing that was applied to a wearable ECG monitoring system. In our design, there are a harmonious combination of hardware design as well as data processing in firmware and software filtering. Parts of the design must be considered together to achieve the best performance.

We compared and evaluated aspects of JPEG compression for ECG signals in real time performed on PIC32MZ. The results help the reader to evaluate the use and replace the PIC32MZ chip with other popular MCU types such as MSP430 or STM32. This is also one of the important factors that helps reduce the time spent researching and implementing designs with commercial significance.

Sources of funding

This work was supported in part by the Technology development Program (S2829803) funded by the Ministry of SMAs and Startups (MSS, Korea).

Ethical approval

Our study do not involves to human subjects Ethical approval: Not required

Declaration of Competing Interest

The authors declare no conflicts of interest.

References

[1] Organization, World Health, Cardiovascular Diseases (CVDs) 17 May 2017.

[2] Xunde Dong, Cong Wang, Wenjie Si, ECG beat classification via deterministic learning, Neurocomputing 240 (May 21) (2017) 1–12.

[3] Shanshan Chen, Wei Hua, Zhi Li, Jian Li, Xingjiao Gao, Heartbeat classification using projected and dynamic features of ECG signal, Biomed. Signal Process. Control 31 (January) (2017) 165–173.

[4] Christopher Beach, Sammy Krachunov, James Pope, Xenofon Fafoutis, Robert J. Piechocki3,Ian Craddock3,Alexander J. Casson, An Ultra Low Power Personalizable Wrist Worn ECG Monitor Integrated With IoT Infrastructure. 10 August 2018. 6: p. 44010-44021.

[5] Alka S. Barhatte, Rajesh Ghongade, Sachin V. Tekale, Noise analysis of ECG Signal using fast ICA, 2016 Conference on Advances in Signal Processing (CASP), 9–11 June, 2016.

[6] Syed Khairul Bashar, Eric Ding, Allan J. Walkey, D. Mcmanus David, H. Chon Ki, Noise detection in electrocardiogram signals for intensive care unit patients, IEEE Access 7 (July) (2019) 88357–88368.

[7] Nevi Jain, DevendraKumar Shakya, Denoising Baseline Wander Noise from Electrocardiogram Signal using Fast ICA with Multiple Adjustments, Int. J. Comput. Appl. 99 (August 2) (2014) 0975–8887.

[8] JianqiangLi Ming, Genqiang Deng, Wei Wei, Huihui Wang, Zhong, Design of a real-time ECG filter for portable mobile medical systems, IEEE Access 5 (October) (2016) 696–704.

[9] Brian Pisani, Digital Filter Types in Delta-Sigma ADCs May 2017.

[10] Ngoc Thang Bui, DucTri Phan, ThanhPhuoc Nguyen, Giang Hoang, Jaeyeop Choi, QuocCuong Bui, Oh Junghwan, Real-time filtering and ECG signal processing based on dual-core digital signal controller system, IEEE Sensor 20 (12) (2020) 6492–6503.

[11] Ngoc Thang Bui, Tan Hung Vo, Byung-Gak Kim, Oh Junghwan, Design of a solar-powered portable ECG device with optimal power consumption and high accuracy measurement, Appl. Sci. 9 (10) (2019).

[12] Duc Tri Phan, Thi Tuong Vy Phan, Ngoc Thang Bui, Sumin Park, Jaeyeop Choi, Junghwan Oh, A portable device with low-power consumption for monitoring mouse vital signs during in vivo photoacoustic imaging and photothermal therapy, Physiol. Meas. (2020).

[13] Tsung-Han Tsai, Wei-Ting Kuo, An efficient ECG lossless compression system for embedded platforms with telemedicine applications, IEEE Access 6 (July) (2018) 42207–42215.

[14] Wu Hao, Xiaoyan Sun, Jingyu Yang, Wenjun Zeng, Wu Feng, Lossless compression of JPEG coded photo collections, Ieee Trans. Image Process. 25 (June 6) (2016) 2684–2696.

[15] Wen-Yan, Zhao Zhong-Hua, Wang, A lossless compression method of JPEG file based on shuffle algorithm, 2010 2nd International Conference on Advanced Computer Control, 2012.

[16] Daniel Yunge, Sangyoung Park, Philipp Kindt, Samarjit Chakraborty, Dynamic alternation of Huffman codebooks for sensor data compression, IEEE Embed. Syst. Lett. 9 (Sept 3) (2017) 81–84.

[17] Chandan Nayak, Suman Kumar Saha, Rajib Kar, Durbadal Mandal, An efficient and robust digital fractional order differentiator based ECG pre-processor design for QRS detection, IEEE Trans. Biomed. Circuits Syst. 13 (Aug 4) (2019) 682–696.

[18] Shubhankar Saxena, Rohan Jais, Malaya Kumar Hota, Removal of powerline interference from ECG signal using FIR, IIR, DWT and NLMS adaptive filter, 2019 International Conference on Communication and Signal Processing (ICCSP), 4-6 April, 2019.

[19] 2019 Colorado State University-WALTER SCOTT, JR. COLLEGE OF ENGINEERING, REAL-TIME DSP LABORATORY5: Infinite Impulse Response (IIR) Filters on the C6713 DSK.

[20] Levkov, Mihov, Ivanov, Daskalov, Christov, Dotsinsky, Removal of power-line interference from the ECG: a review of the subtraction procedure, Biomed. Eng. Online (Aug) (2005).

[21] Chunyu Tan, Liming Zhang, Wu Hau-tieng, A novel blaschke unwinding adaptive-Fourier-decomposition-based signal compression algorithm with application on ECG signals, IEEE J. Biomed. Health Inform. 23 (2) (2019) 72–682.

[22] Leonardo Vidal Batista, Elmar Uwe Kurt Melcher, Luis Carlos Carvalho,

[23] Nikolay N. Ponomarenko, Karen O. Egiazarian, Vladimir V. Lukin, Jaakko T. Astola, High-quality DCT-Based image compression using partition schemes, IEEE Signal Process. Lett. 14 (2) (2007) 105–108.
[24] K. Ranjeet, A. Kumar, R.K. Pandey, ECG Signal Compression Using Different Techniques, Indian Institute of Information Technology Design and Manufacturing, Jabalpur-482005, MP (India), 2011, pp. 231–241.
[25] Ferda Ernawan, Siti Hadiati Nugraini, The optimal quantization matrices for jpeg image compression from psychovisual threshold, J. Theor. Appl. Inf. Technol. 70 (3) (2014).
[26] Rahul Kher, Signal processing techniques for removing noise from ECG signals, J. Biomed. Eng. Res. (March) (2019).
[27] Chia-Hsu Kuo, Mu-King Tsay, Lu Cheng-Chang, An efficient repetition finder for improving dynamic Huffman coding, IEEE Trans. Commun. 45 (Nov 11) (1997) 1363–1366.
[28] Texas Instruments, ADS1293 24-bit, 3-ch, Low-Power Analog Front END (AFE) for ECG Applications 17 Dec 2014.
[29] Y. Wiseman, The still image lossy compression standard - JPEG, Comput. Sci. 1 (2015) 295–305.
[30] Muzhir Shaban AL-Ani, Fouad Hammadi Awad, The JPEG image compression algorithm, Int. J. Adv. Eng. Technol. (May) (2013).
[31] M. Fink, G. Montaldo, M. Tanter, Time reversal acoustics, IEEE Ultrasonics Symposium, 2004.
[32] Krzysztof Sozanski, A linear-phase IIR filter for audio signal interpolator, 2013 Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2013.
[33] S. Cuomo, G. De Pietro, R. Farin, A. Galletti, G. Sannino, A revised scheme for real time ECG Signal denoising based on recursive filtering., Biomed. Signal Process. Control 27 (May) (2016) 134–144.
[34] Microchip, PIC32MZ2048EFM100, 2019.
[35] A. Wong, Kong-Pang Pun, Yuan-Ting Zhang, Chiu-Sing Choy, An ECG measurement IC using driven-right-leg circuit, IEEE International Symposium on Circuits and Systems, 2006.
[36] G.B. Moody, R.G. Mark, The impact of the MIT-BIH arrhythmia database, IEEE Eng. Med. Biol. Mag. 20 (3) (2001) 45–50.