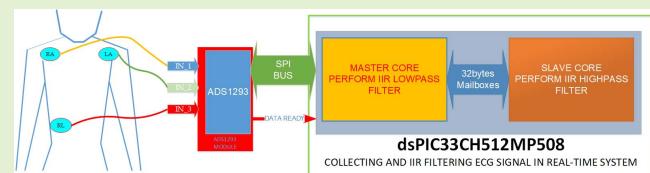


Real-Time Filtering and ECG Signal Processing Based on Dual-Core Digital Signal Controller System

Ngoc Thang Bui^{ID}, Student Member, IEEE, Duc Tri Phan^{ID}, Thanh Phuoc Nguyen^{ID}, Giang Hoang^{ID}, Jaeyeop Choi^{ID}, Quoc Cuong Bui^{ID}, and Junghwan Oh^{ID}, Member, IEEE

Abstract—ECG devices are increasingly used in the monitoring and timely detection of cardiovascular abnormalities. One of the biggest challenges for ECG device designers is to eliminate noise from the surrounding environment to obtain high-quality and real-time ECG signals. A microcontroller usually has low processing speed, so it is not appropriate for real-time processing of ECG signals with high sampling frequency. Processing continuous signals requires a high processing speed or processor that can be used with many partial processing hierarchies. In 2018, Microchip Technology Inc. launched a new family of 16-bit digital signal controllers (DSCs) with two cores: the master core can be operated at 180 MHz and the slave core can be operated at 200 MHz. The new DSC family is suitable for mobile application systems, such as ECG devices, which can collect and process data in real time with a process hierarchy. In this paper, we present an application of the two cores of DSCs in collecting and processing ECG signals in real time. In addition, we implement an infinite impulse response (IIR) filter to eliminate noise from the surrounding environment. Specifically, the master core collects and implements a high-pass IIR filter with five poles, and the slave core implements the low-pass IIR filter with seven poles and sampling frequency of ECG signals up to 400 Hz.

Index Terms—ECG, real-time embedded system, multi-core system, wearable device.



I. INTRODUCTION

IN RECENT years, cardiovascular diseases (CVDs) have become the leading cause of death worldwide. Approximately 17.9 million people died from CVDs in 2016, accounting for 31% of the total global deaths. 85% of these deaths are caused by heart attack and stroke. The majority of CVD deaths occur in low-income and middle-income countries, accounting for more than three-quarters [1].

The use of ECG device is one of the most common and effective methods to monitor heartbeats [2]. However, ECG signals often have relatively low amplitudes of 0.1 mV to

2.5 mV and are easily affected by environmental noise [3], [4]. Many types of noises can affect ECG signals [3], [5]–[7]. These noises come as power frequency interference, baseline drifts, muscle movement interference, and contact interference [3], [8]. Among them, power frequency interference and baseline drifts are the two major types of noises and directly affect ECG signals [3], [9], [10]. Therefore, eliminating these types of noises is important and necessary for processing ECG signals [11], [12].

The baseline drift noise is usually generated by breathing and muscle movement and is mainly at 0.05–2 Hz [3], [8]. The other one is at the 50/60 Hz frequency noise with sine forms created by the effect of the grid on biomedical signals [13]. Baseline deviation usually has a low frequency signal [6]. The remaining power frequency interference has a defined frequency of 50 or 60 Hz depending on the grid in each country [11], [14]. To eliminate these noises, we designed a low-pass filter to eliminate power frequency interference and a high-pass filter to eliminate baseline drifts [9], [11].

The filters can be implemented on a hardware [15] or software [16], [17]. In this study, we focused on designing filters on the digital signal controller system, so the filter

Manuscript received November 1, 2019; revised February 6, 2020; accepted February 12, 2020. Date of publication February 19, 2020; date of current version May 15, 2020. This work was supported by a Research Grant of Pukyong National University (2019). The associate editor coordinating the review of this article and approving it for publication was Prof. Diego Barretino. (*Corresponding author: Junghwan Oh*)

The authors are with the Interdisciplinary Program of Biomedical Mechanical and Electrical Engineering, Center for Marine-Integrated Biomedical Technology, Department of Biomedical Engineering, Pukyong National University, Busan 48513, South Korea (e-mail: jungoh@pknu.ac.kr).

This article has supplementary downloadable material available at <http://ieeexplore.ieee.org>, provided by the authors.

Digital Object Identifier 10.1109/JSEN.2020.2975006

implementation on hardware is not presented [3]. The software filters create flexibility in design that easily changes the parameters of filter [8]–[10], [13], [18]. It is suitable for mobile devices that require compact size and low energy consumption [19]–[21].

In recent years, the development of microprocessors clearly shows the trend of many cores in the CPU [22]. The performance of the system is increased by using modern operating systems on the basis of multi-core CPU architecture and distributing tasks between processing cores [23]. In addition, due to the limited time of device operation, some real-time applications are required to operate in systems that operate with a limited power supply, which also raises the need for optimization in equipment [12], [24]. Therefore, minimizing energy consumption while achieving the desired performance is necessary.

Nowadays, multi-core systems are widely used in real-time embedded systems [22]. Some of the most potential areas where multi-core systems can be applied are image processing, neural network, and signal processing algorithms [8], [11]. Multi-core processors are integrated circuits attached with two or more processors to increase performance, reduce power consumption, and efficiently handle multiple tasks [23].

A dual-core processor consists of two separate processing cores attached in the same chip [25]. Therefore, the time to connect and exchange data between the two cores is reduced. Moreover, the data processing time of dual-core chips is faster than that of single-core chips [22]. Ideally, a dual-core processor has nearly as twice as the power of a single-core processor. In fact, dual-core processors are approximately one and a half times more powerful than single-core processors [22]. Multi-core processing is a growing industry trend as single-core processors clearly present the physical limits in complexity and speed and are difficult to be applied in modern applications. Companies have produced or are working on multi-core products, including NVIDIA, AMD, ARM, Broadcom, Intel, and Microchip.

The design of the dual-core processor allows it to be able to split information for processing in multiple units, so it is more efficient than a single-core processor. The performance increases are significant for applications that need to run more than one process at the same time. Data processing requires switching between different threads, which reduces processor performance. For dual-core processors, due to the ability to process two data streams simultaneously, it reduces the time to switch between threads thus increasing performance compared to single-core processors.

The dsPIC33CH dual-core family system is designed with advanced embedded control applications. Because of the new DSC family, the integration of software becomes easier. Writing programs for each core is an independent task, but these programs can be integrated into only software. The dsPIC33CH has a master core, whereas the other core is a slave [25]. The slave core executes a dedicated control code, which has a critical time, whereas the main core runs the user interface and monitors the system and communication functions.

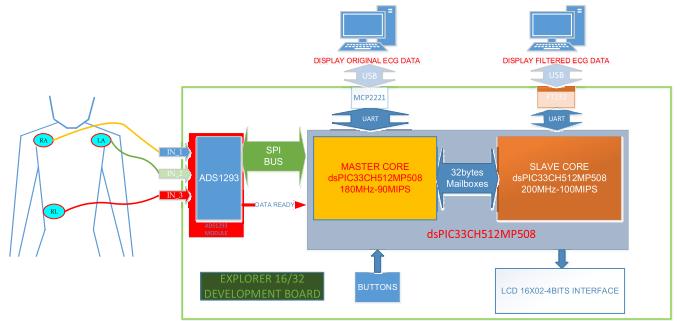


Fig. 1. Real-Time ECG system.

TABLE I
COMPARED PARAMETERS BETWEEN THE IIR AND FIR FILTERS

Type of filter	IIR	FIR
Computation al Speed	Fast-Low Order	Slow-High Order
Phase/ Delay Stability	Not constant Sometimes	Constant Always

In this work, we newly designed a novel ECG device by implementing the two cores of DSC for real-time collecting and processing ECG signals using ADS1293 [26] from Texas Instruments and dsPIC33CH512MP508 [25]. Digital filters are used for filtering ECG signals in real time. The master core implements a high-pass filter to eliminate baseline drift [3], [6], whereas the slave core implements a low-pass filter to eliminate power frequency interference [5], [11], [27]. The ECG signals before and after applying filters are transmitted to a personal computer (PC) via USB to COM (USB2COM). The structure of this system is described in Fig. 1 [28].

The remainder of this paper is organized as follows: Section II presents an IIR filter algorithm and the filter structure selected for the embedded system. Section III presents the data processing method on firmware. Section IV presents the overall system implementation. Section V details the performance evaluation. Finally, Section VI draws the conclusions.

II. EXPERIMENTAL SECTION

A. Selected Type of Filter for the System

The most common digital signal processing systems include finite impulse response (FIR) and IIR filters. FIR and IIR are also used to eliminate noises [11]. FIR filters have the advantages of simple structure and no time delay, but they involve a large number of calculations. The IIR filter is a recursion from the FIR filter and can use the small-calculation FIR filter to achieve the same effect. The FIR filter is always stable, but the IIR filter needs to be designed to achieve the required stability. The IIR filter has an unstable time delay so it may cause signal distortion. Table I compares the characteristics between these types of filters.

IIR filters have high computational efficiency and short latency. IIR filters include zeros and poles and require less memory than FIR filters. It should be suitable for DSC systems with low computational speed. Therefore, we selected the IIR



Fig. 2. Filter structure for processing ECG signals.

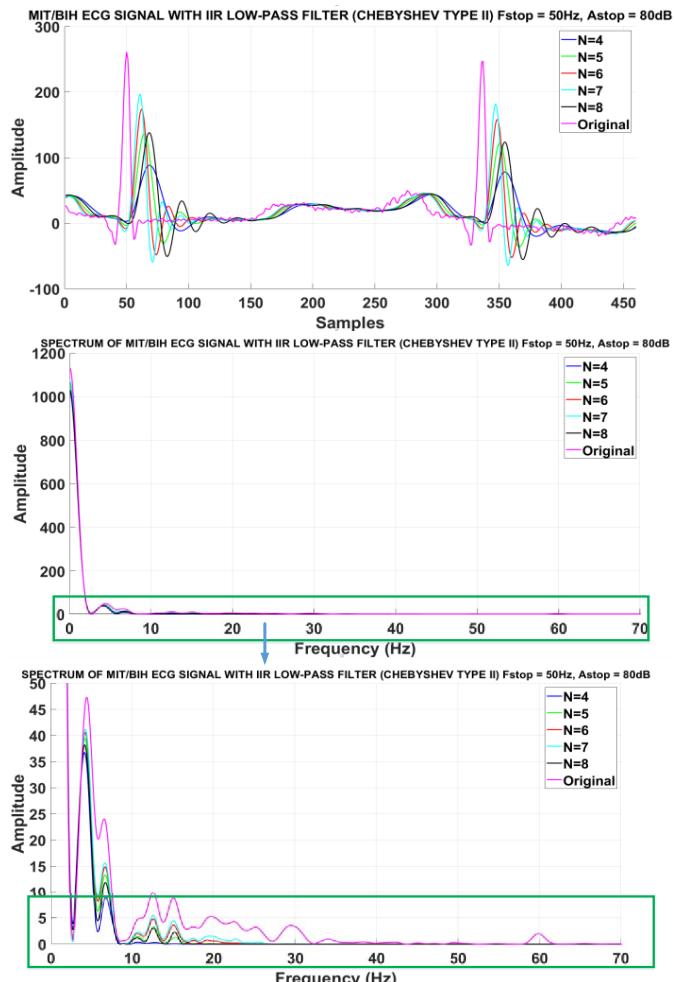


Fig. 3. Response of ECG signals with different orders.

filter for the ECG signal filter design in this study. The filter is designed to include two types of IIR filters. The IIR high-pass filter is used to remove baseline drift interference frequencies, which are less than 0.05 Hz [3], and the IIR low-pass filter is used to remove the power frequency interference [11].

B. Selected Filter Methods

The characteristics of the low-pass filter and a comparison of the performance of the methods of the IIR filter is described in Table II.

The attributes of the different IIR filter methods are as follows:

- Butterworth: flat response in the pass and stop band
- Inverse Chebyshev: flat in the pass band, with a narrower transition width than the Butterworth filter
- Chebyshev: has ripple in the pass band

On the basis of the characteristics presented in Table II, we chose the design of the high-pass filter IIR filter with cutoff

TABLE II
COMPARISON METHODS OF THE IIR FILTER

Method	Pass Band	Transition Width	Stop Band
Butterworth	Flat	Wide	Monotonic
Inv.Chebyshev	Flat	Narrow	Ripple
Chebyshev	Ripple	Narrow	Monotonic

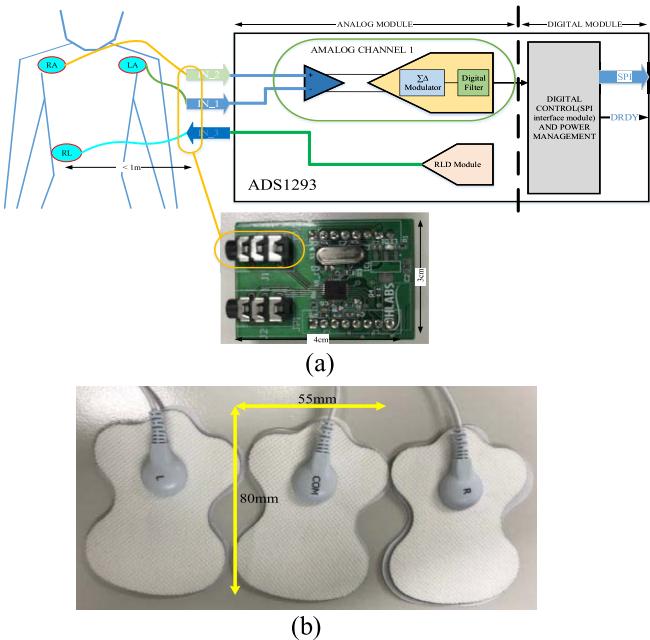


Fig. 4. (a) ECG front-end module, (b) Electrodes.

at a frequency of 0.05 Hz and Butterworth method to eliminate noise baseline drift. We also selected the low set cutoff filter pass at a frequency of 40 Hz with Inv.Chebyshev method to eliminate 60 Hz noise power interference. The structure of the filters is described in Fig. 2.

C. Design Coefficients for the High-Pass Filter

As mentioned in the previous section, the IIR high-pass filter with the Butterworth method with a cutoff frequency of 0.05 Hz was selected as the filter type to eliminate baseline drift noise. In this section, we discuss the selection of filter number and filter response corresponding to that number. The IIR filter structure is a type of cascade second-order section, so the number of filters of the filter is even.

In Fig. 3, an ECG signal from the MIT/BIH database is shown after applying the low-pass IIR filter with different orders. The order of filter is 7 ($N = 7$) that the signal response is good and it is appropriate for this system.

III. DATA PROCESSING METHOD

A. Designing ECG Circuit

In this paper, we use ADS1293 IC to collect ECG signals from the human body. The detailed configuration of this IC is described in Fig. 4. With 3 ADC modules, the ADS1293 IC

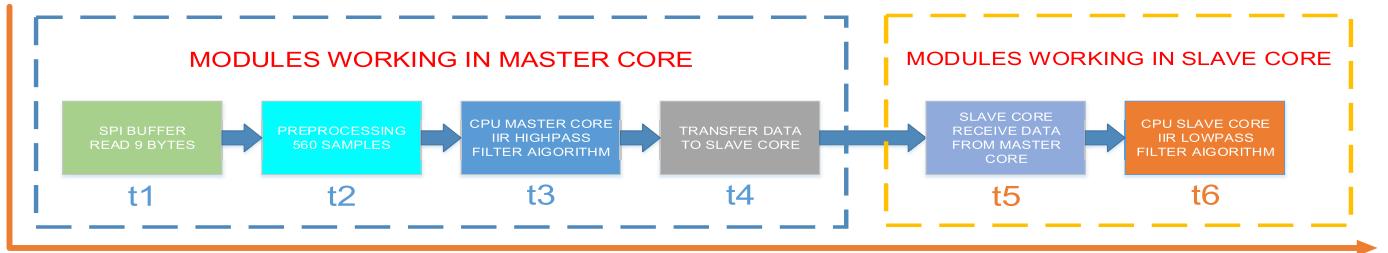


Fig. 5. Sequence of implementation of modules in the ECG system.

can simultaneously convert 3 analog signals, thus enabling the development of an ecg signal acquisition application with 5 leads. To collect ECG signals from the human body we use electrodes that are affixed directly onto the human body (the electrodes described in Fig. 4) [29]. The connection between electrodes and ADS1293 using wires about 1m long. Both Analog and digital modules in IC ADS1293 are rated at 3.3V, compatible with DSC voltage. Analog module to convert analog signal to digital and put it into the buffer of the digital module (SPI slave), the digital module transmits this data to the master chip via the SPI protocol.

Besides, general mode rejection, or CMR, is one of the most important performance parameters for ECG systems where we have used the Rigth-Led Driver (RLD) circuit built into the ic. ADS1293 to eliminate noise and improve chip performance. Enabling RLD mode is done by configuring register $0 \times 0C$ in ADS1293 chip.

The ADC signal conversion in the ADS1293 chip is done automatically with the sampling frequency configurable by the registers R1 (0×25), R2 (0×21) and R3 ($0 \times 22 - > 0 \times 24$). The conversion process starts the DRDYB pin to high level after the ADC conversion ends the data in the buffer of the SPI module DRDYB pin is lowered to create an interrupt signal that excites the master chip the data is ready to read.

B. Preprocessing Data

Before the dsPIC33CH512MP508 [25] processed data using IIR filter, the data had been collected from ECG ADS1293 [26] chip via the Serial Peripheral Interface (SPI)[30]. As the ADS1293 [26] supports streaming mode, it can continuously read 9-bit data from three channels inside the ADS1293 [26]. In addition, we configure the data-ready bar (DRDYB) signal in ADS1293 [26] as an external interrupt signal for dsPIC33CH512MP508 [25] to accurately determine when to read data after the data in ADS chip is ready.

As the effect of the right-leg drive (RLD) signal received is offset, after receiving a frame of this data, correcting this offset value is necessary. In this design, we adjust the offset value by subtracting the first value of each received data frame. This process is performed through the following steps.

Step 1: Configurable ADS1293 [26] with three analog input channels

Streaming mode (9 bytes), DRDYB (connect to IN1), and RLD (connect to IN3)

Step 2: Signal DRDYB [31] form ADS1293 [26] is a trigger for dsPIC33CH512MP508 [25]. If this trigger is activated,

then dsPIC33CH512MP508 [25] begins to read 9 bytes from ADS1293 [25] through SPI and store these data into a buffer (buffer with 2560 bytes).

Step 3: Offset data from the buffer.

C. IIR Filter Algorithm Based on Firmware

The raw data after the preprocessor module will be included in the IIR filter with the cascade IIR filter structure. Each section of the cascade structure has an order equal to 1 or 2. The calculation process for each section is performed through the following steps:

Step 1: Initialize the filter coefficients.

$$b = [b_1, b_2, b_3];$$

$$a = [a_1, a_2, a_3].$$

Step 2: Perform the loop to calculate section output values.

$$d = \text{zeros}(3, 1); \text{ \% clear delay line}$$

$$y = \text{zeros}(N, 1); \text{ \% allocate output array}$$

for n = 1:N

$$d = [0; d(1:end-1)]; \text{ \% update delay line}$$

% compute feedback

$$d(1) = x(n) - a(2)*d(2) - a(3)*d(3);$$

% compute feedforward

$$y(n) = b(1)*d(1) + b(2)*d(2) + b(3)*d(3);$$

end.

IV. OVERALL SYSTEM IMPLEMENTATION

This system was also developed on Microchip 16-bit DSC with two cores. The master core operates at a frequency of 180 MHz and is responsible for relaying SPI with ECG chip ADS1293 [26] to read data. Then, the data are transmitted to a PC via USB2COM transmission with a baud rate of 230400 bps via DMA [32] technique to save CPU processing time. At the same time, the CPU of the chip also performs a high-pass filter to filter high-frequency noises from this signal. Then, the data are transmitted to the slave core to continue processing. Each master core transmission transmits 32 bytes of data to the slave core.

After receiving all data from the master core, the slave core continues to implement the low-pass filter to filter out 60 Hz noise signal. The data are then transferred to the PC via USB2COM at the same baud rate of 230400 bps for display and comparison with the signal before processing.

A. Hardware Implementation

The whole wearable monitoring system constructed for this study is illustrated in Fig. 1. The embedded platform core

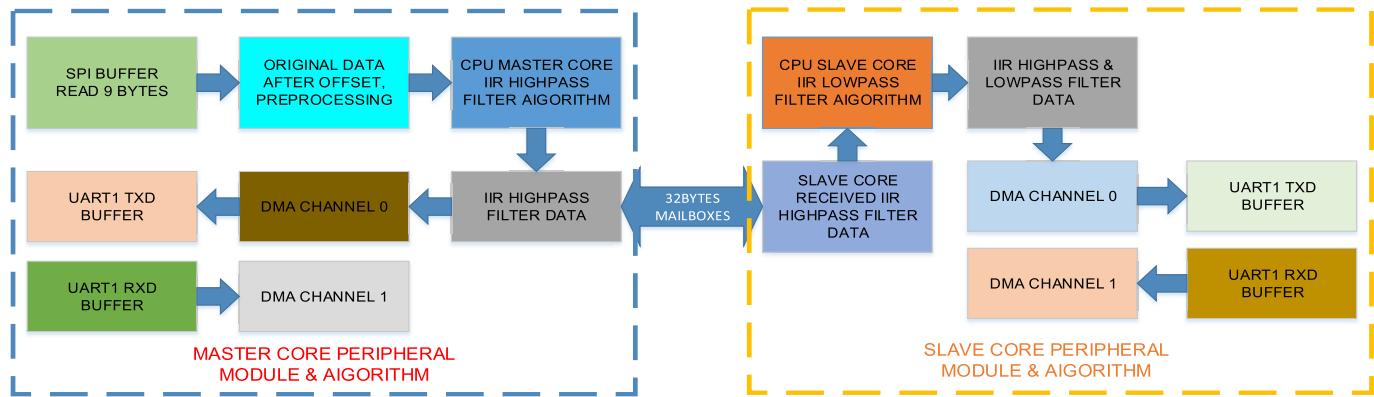


Fig. 6. Peripheral and algorithm inside dsPIC33CH512MP508.

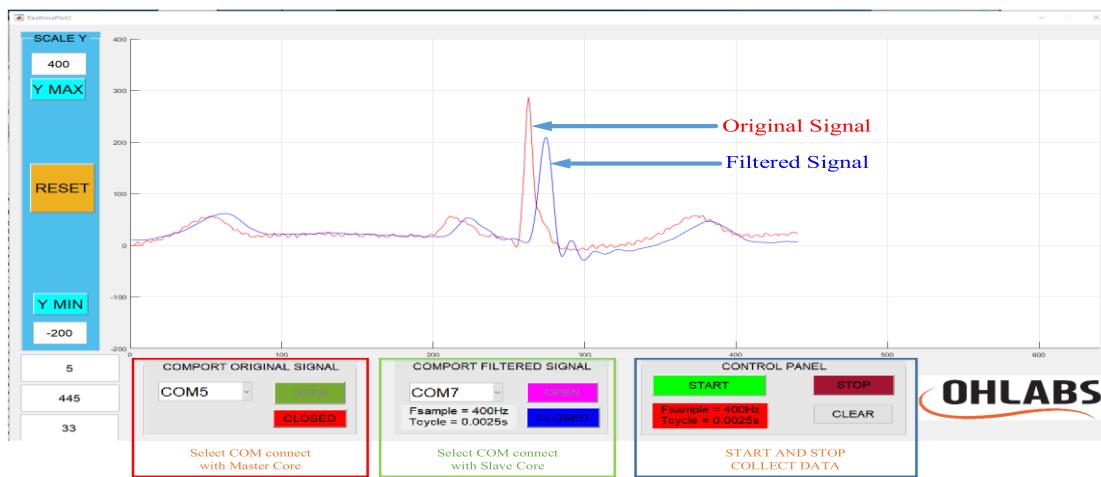


Fig. 7. Software display ECG before and after applied IIR filter.

is dsPIC33CH512MP508 [25], which is a 16-bit DSC from Microchip. This DSC collects data from ADS1293 [30] chip via the SPI transmission line and then transmits unprocessed data to the PC via the UART1 [33] module of the master core. The UART1 [33] module of the master core is directly connected with USB2COM IC (MCP2221). The UART1 [33] module incorporates DMA0 and DMA1 modules to optimize data transmission and reception. The data are then further processed by the master core with the fifth-order IIR high-pass filter. After processing, the transmission data are forwarded to the slave core via the Master Slave Interface (MSI)[30] module. After the slave core receives the data from the master core transmission, it will continue to perform the seventh-order low-pass filter. The data are then transmitted to the PC via the UART1 [33] module of the slave core. The UART1 [33] module is connected to the USB2COM FT232 conversion chip, and the data transmission process of this module also uses the DMA [32] module to optimize the operation of the whole system. The operation details of this firmware are described by the algorithm flowchart in Fig. 6.

B. Monitor Display Software

To display the received data, we developed a software to receive and display data from the USB2COM port.

The software can be directly connected to two COM ports to display the original and filtered data after the filter is implemented. This software is written based on MATLAB 2018. Fig. 7 shows the main monitor of this software.

The data-receiving software includes two parts: Part 1 receives data from the serial port and saves the data to the workspace. Part 2 receives data from the serial port stored in the workspace and draws real-time signals on the graph. The second part is limited by the updated data capability of MATLAB (MATLAB is capable of updating data with the highest frequency of 1 kHz). We have tested and found that the software stably works with 200 Hz frequency, and the sample number makes 320 samples corresponding to the time of 1.6 s.

V. PERFORMANCE EVALUATION

After completing the hardware, firmware, and software design, we conducted tests to evaluate the effectiveness of this system. The test was conducted in three parts: Part 1 aims to evaluate the performance of the firmware with the measurement of processing time of modules. Part 2 implements the filter with the data from the MIT/BIH database [34], [35], which evaluated the filter performance and efficiency. Part 3 measures the ECG signals on volunteers to assess the

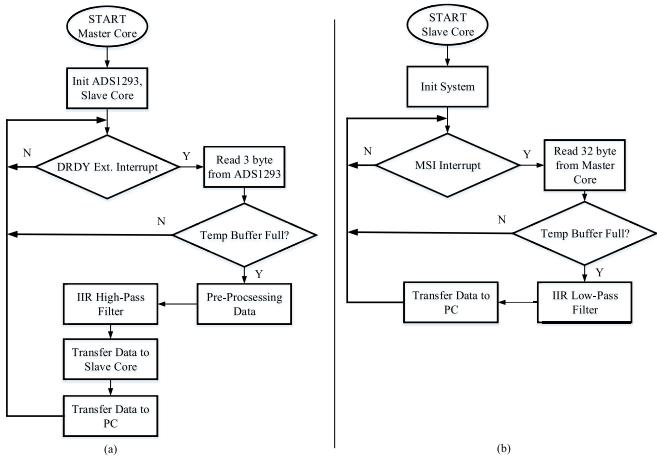


Fig. 8. Flowchart of the firmware operation inside. (a) Master core. (b) Slave core.

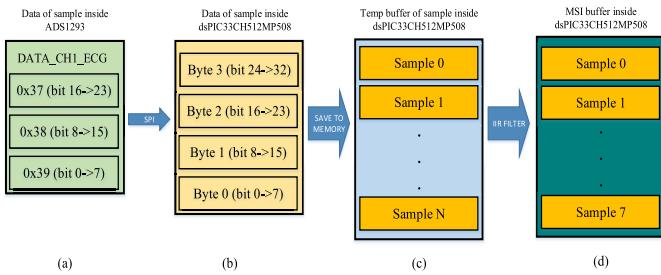


Fig. 9. Format data of: (a) Original data from ADS1293. (b) One sample after read via SPI interface (4 byte). (c) Temporary buffer with N samples. (d) Temporary MSI interfaces buffer with 8 samples per time.

accuracy and response of the filter in real time. The hardware settings for these tests are described in Fig 13. Firmware was written by the software suit MAPB X v5.2 with the compiler MPLAB XC16 v1.36 and the embedded tool MCC v3.

In this study, we focus on evaluating the performance of dual-core chips so that the core master and slave core are configured at the maximum clock frequency for the maximum core master frequency of 180MHz and slave core is 200 MHz.

Fig. 8 describes the process of reading and processing data in master core and slave core. In this description, we only present data reading from 1 channel of ADS1293 chip, with 3 channels of similar process with each data reading through SPI of 9 bytes.

The IIR high-pass filter and Low-Pass filter in firmware are implemented according to the Cascaded Second-Order Systems (Biquads) structure [36] with a coefficient table detailed in various tests.

Fig. 9 depicts the data structure in dsPIC33CH512MP508, the initial data in ADS1293 buffer consists of 3 bytes (24-bits) after being put into the DSC buffer which were converted into 4 bytes format with MSB bytes (byte 3 (bit 24 -> 32) has a value of 0x00). The subsequent reading of data is stored in the temp buffer in this 4-byte format. The process of transferring data from master core to slave core uses temp buffer different from the 8 samples format corresponding to each 32 byte data transmission.

TABLE III
CPU SPEED AND DATA READ TIME

Data memory (byte)	CPU clocks	Operating time (us)
10	50 MHz	215
10	100 MHz	215
10	180 MHz	215

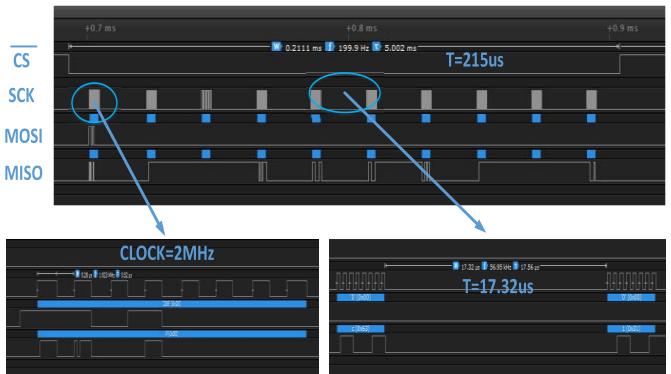


Fig. 10. SPI signal and processing time.

TABLE IV
NUMBER OF SAMPLES AND PROCESSING TIME AT CPU CLOCK OF 180 MHZ IN THE PROCESSING DATA MODULE

Number of Samples	Data memory (byte)	Operating time (ms)
320	1,290	2.256
640	2,570	4.497
1,000	4,010	7.15
1,600	6,410	11.58

A. Performance Evaluation of the Firmware

In this test, we separately evaluated the operation of the modules. The operation sequence of the modules in the firmware is described in Fig. 13.

1) **SPI Module Evaluation:** We configured the clock of the SPI module to 2 MHz and operated in mode 0. The CPU of DSC starts to read data from ADS1293 chip after receiving the DRDYB signal. CPU sends 0×27 data to ADS1293 chip to read 9 bytes in the ADS1293 buffer. The test was conducted with the CPU operating at different frequencies. The results are shown in Table III and Fig. 10.

Table III shows that the data reading time mainly depends on the SPI module clock. However, due to hardware limitations in our test, increasing the module clock rate higher is impossible because of data loss status.

2) **Preprocessing Module Evaluation:** The module preprocessing signal used to offset the data in each block includes the number of samples, with each sample having 32-bit length (4 bytes). The test was also carried out with many different samples, and the results are shown in Table IV.

TABLE V

NUMBER OF SAMPLES AND PROCESSING TIME AT CPU CLOCK OF 180 MHz IN THE HIGH-PASS FILTER MODULE

Number of Samples	Data memory (byte)	Operating time (ms)
320	2,586	718.4
640	5,146	1,424
1,000	8,026	2,203
1,600	12,826	3,412.4

TABLE VIII

NUMBER OF SAMPLES AND PROCESSING TIME AT A CPU CLOCK OF 180 MHz IN THE LOW-PASS FILTER MODULE

Number of Samples	Data memory (byte)	Operating time (ms)
320	2,592	771.8
640	5,152	154.6
1,000	8,032	2,411.8
1,600	12,832	3,860

TABLE VI

NUMBER OF SAMPLES AND PROCESSING TIME AT A CPU CLOCK OF 180 MHz IN THE TRANSFER DATA MODULE

Number of Samples	Data memory (byte)	Operating time (ms)
320	2,580	17.64
640	5,140	24.33
1,000	8,020	34
1,600	12,820	42.3

TABLE VII

NUMBER OF SAMPLES AND PROCESSING TIME AT A CPU CLOCK OF 180 MHz IN THE RECEIVER DATA MODULE

Number of Samples	Data memory (byte)	Operating time (ms)
320	2,580	17.64
640	5,140	24.33
1,000	8,020	34
1,600	12,820	42.3

3) High-Pass Filter Module Evaluation: In this module, the CPU implements a high-pass filter with the filter parameters shown in Table XI, and the results are presented in Table V. The number of bytes used in this module is as follows: $N_{sample} * 4 * 2 + 26$, where 26 is the number of bytes for the temporary variable in program C.

4) Transfer Data Module Evaluation: The main purpose of this module is for the master core to transmit data to the slave core by using the interrupt mechanism of the MSI module inside dsPIC33CH512MP508. Using mode mailboxes, each data transmission consists of one block with 32 bytes. The transmission time and memory usage are described in Table VI.

The number of bytes used in this module is as follows: $N_{sample} * 4 * 2 + 20$, where 20 is the number of bytes for the temporary variable in program C.

5) Receiver Data Module Evaluation: This module is similar to the transfer data module presented above, but it works on the slave core. The test results are presented in Table VII.

The number of bytes used in this module is as follows: $N_{sample} * 4 * 2 + 20$, where 20 is the number of bytes for the temporary variable in program C

TABLE IX

OPERATING TIME WITH SINGLE-CORE AND DUAL-CORE SYSTEMS

Number of Samples	Single Core (ms)	Dual-Core (ms)
320	1,768	754.8
640	3,486	1,506
1,000	5,560	2,358

6) Low-Pass Filter Module Evaluation: In this module, the CPU in the slave core implements a low-pass filter. The filter parameters are shown in Table XI, and the results are presented in Table VIII.

The number of bytes used in this module is as follows: $N_{sample} * 4 * 2 + 32$, where 32 is the number of bytes for the temporary variable in program C.

7) Comparison of Operating Time Between the Single-Core and Dual-Core Systems: In this test, we compared the execution time of the modules with the firmware written on the single-core and dual-core systems. The firmware was made on dsPIC33CH512MP508, and the master and slave cores operate at 180 MHz.

Table IX shows that the dual-core chip has outstanding speed compared with the single-core chip.

B. Test System With ECG Data From the MIT/BIH Database

In this test step, we used data taken from an online data source. The patient's heart rate data was recorded in 3600 sample format with a recording time of 10 seconds. This database records the data of ECG signal of 45 patients with 17 different classes but we chose the class normal sinus rhythm (NSR) for testing in this paper [37]. Due to the SRAM memory limit in DSC, we selected a data frame with 640 samples taken from the MIT / BIH database with a 360 Hz sampling frequency equivalent to the 1.78 second ECG signal recording time (those data is taken directly from the firmware). The timer with a period time of 1.78 seconds is used to describe the time of data collection. Every 1.78 seconds, the master core transmits the data before processing it to the PC and then executes the high-pass filter. Then, it transmits the data to the secondary core to perform the low-pass and slave-core filters for later data transmission. The software on the PC will collect the two data at the same time. The data will be used to redraw the signal with MATLAB. The coefficients of the

TABLE X

COEFFICIENTS OF THE IIR HIGH-PASS FILTER (BUTTERWORTH) AT A SAMPLING FREQUENCY OF 360 Hz, CUTOFF FREQUENCY OF 0.05 Hz, AND ORDER OF 5

Sections	Section 1	Section 2	Section 3
b[0]	9,997	9,993	9,996
b[1]	-19,994	-19,986	-9,996
b[2]	9,997	9,993	0
a[0]	10,000	10,000	10,000
a[1]	-19,995	-19,986	-9,991
a[2]	9,995	9,986	0

TABLE XI

COEFFICIENTS OF THE IIR LOW-PASS FILTER (INV.CHEBYSHEV) AT A SAMPLING FREQUENCY OF 360 Hz, CUTOFF FREQUENCY OF 50 Hz, ORDER OF 7, AND ASTOP OF 80 dB

Sections	Section 1	Section 2	Section 3	Section 4
b[0]	2,196	1,453	702	1,941
b[1]	-2,756	-1,381	101	1,941
b[2]	2,196	1,453	702	0
a[0]	10,000	10,000	10,000	10,000
a[1]	-16,854	-14,527	-12,868	-6,118
a[2]	8,489	6,052	4,372	0

TABLE XII

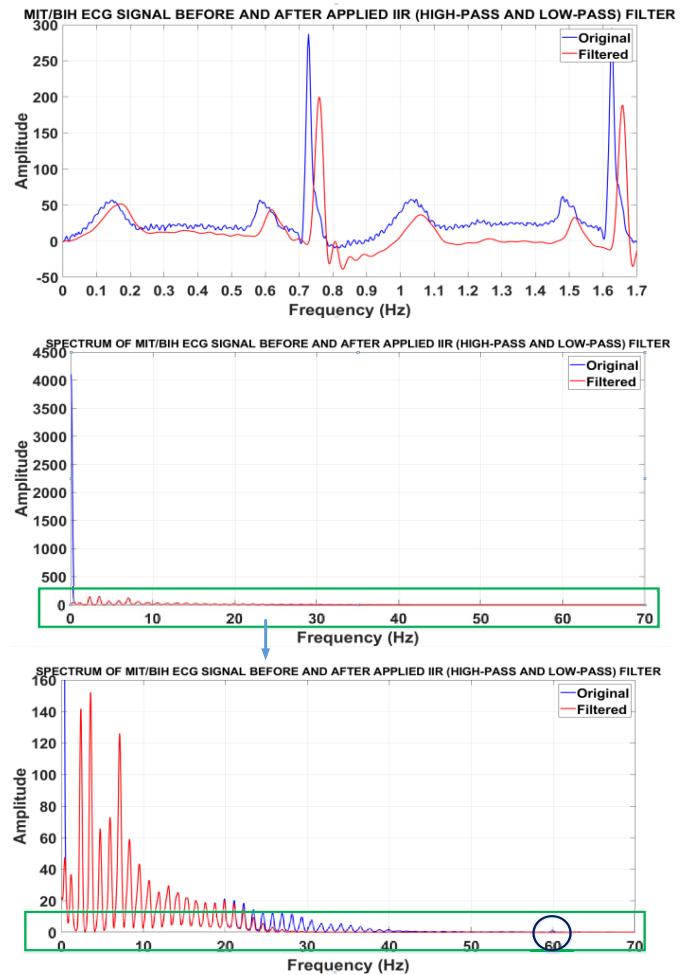
SNR AFTER APPLIED HIGH-PASS AND LOW-PASS FILTER

Type of Data	SNR Original	SNR Filtered
NSR	1.3678	59.6523
APB	13.6244	77.6878
AFL	13.6607	182.12
AFIB	29.4667	1175.1
SVTA	12.3659	26.33

filter is described in [Table X](#) (coefficients of high-pass filter) and [Table XI](#) (coefficients of low-pass filter). The results are shown in [Fig. 11](#) (signals after applying the high-pass and low-pass filters).

Through the FFT analysis described in [Fig. 11](#), we see that there are two types of noises NSR data of MIT/ BIH: baseline drift and power frequency interference (60 Hz) and the high-pass and low-pass filters have been used. like eliminating these two types of noise completely.

In addition, we conducted additional tests with other data such as normal sinus rhythm (NSR), atrial premature beats (APB), atrial flutter (AFL), atrial fibrillation (AFIB), Supraventricular Tachycardia (SVTA), to evaluate the signal to noise ratio (SNR) of ECG signals after applying these filters [38]. 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 0.05Hz to 40Hz are considered ECG signals. Signal to Noise Ratio (SNR) is calculated according to these parameters. The result is described in the [Table XII](#).



[Fig. 11.](#) MIT/BIH ECG signals after applying high-pass IIR filter with a cutoff frequency 0.05 Hz and low-pass filter with a cutoff frequency 40 Hz.

According to the results presented in [Table XII](#), the energy part of the signal is concentrated in the lower frequency range less than 0.05Hz. The high-pass filter eliminates the majority of the signal in this frequency range thereby raising the SNR in the frequency range of the ECG signal.

The second part of the test is similar to the first part, however the number of samples is less than 320, corresponding to a time of 1.6 second. The signal includes only one signal after applying the high-pass and low-pass filters and real-time display on the software. The results are shown in [Fig.14](#).

Based on the two test steps, the real-time data collection operation system is performed. The high-pass and low-pass filters are quite effective after removing the two main types of noises, which are baseline drift and power interference in ECG signals.

C. Test System With Volunteers

In this study, we tested the system with a number of users. The real system was applied to some volunteers' body at certain positions, as shown in [Fig. 13](#). In the first step, we configured the sampling frequency for the device as 400 Hz and sampling time as 1.6 s. The steps for collecting and

TABLE XIII

COEFFICIENTS OF THE IIR HIGH-PASS FILTER (BUTTERWORTH) AT A SAMPLING FREQUENCY OF 400 Hz, CUTOFF FREQUENCY OF 0.05 Hz, AND ORDER OF 5

Sections	Section 1	Section 2	Section 3
b[0]	9,998	9,994	9996
b[1]	-19,996	-19,988	-9996
b[2]	9,998	9,994	0
a[0]	10,000	10,000	10000
a[1]	-19,995	-19,987	-9992
a[2]	9,995	9,987	0

TABLE XIV

COEFFICIENTS OF THE IIR Low-Pass Filter (Inv.Chebyshev) AT A SAMPLING FREQUENCY OF 400 Hz, CUTOFF FREQUENCY OF 40 Hz, ORDER OF 7, AND ASTOP OF 80 dB

Sections	Section 1	Section 2	Section 3	Section 4
b[0]	2,077	1,365	568	1,437
b[1]	-3,323	-1,925	-318	1,437
b[2]	2,077	1,365	568	0
a[0]	10,000	10,000	10,000	10,000
a[1]	-18,068	-16,207	-14,804	-7,126
a[2]	8,899	7,012	5,620	0

TABLE XV

COEFFICIENTS OF THE IIR HIGH-PASS FILTER (BUTTERWORTH) AT A SAMPLING FREQUENCY OF 200 Hz, CUTOFF FREQUENCY OF 0.05 Hz, AND ORDER OF 5

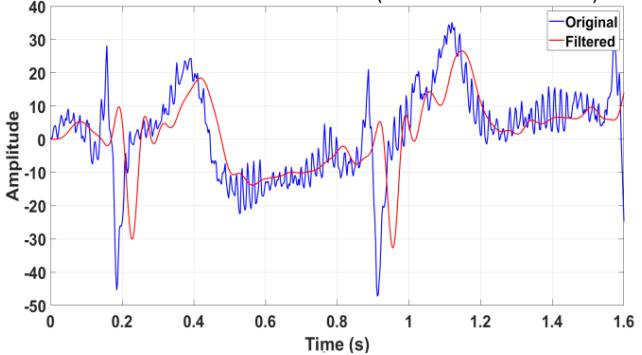
Sections	Section 1	Section 2	Section 3
b[0]	9,995	9,987	9,992
b[1]	-19,990	-19,974	-9,992
b[2]	9,995	9,987	0
a[0]	10,000	10,000	10,000
a[1]	-19,990	-19,975	-9,984
a[2]	9,990	9,975	0

processing data after receiving data are similar to those with the MIT/BIH data. The coefficients of the high-pass and low-pass filters are shown in Tables XIII and XIV, respectively. Fig. 12 show the signals after applying the high-pass filter, low-pass filter, and both filters.

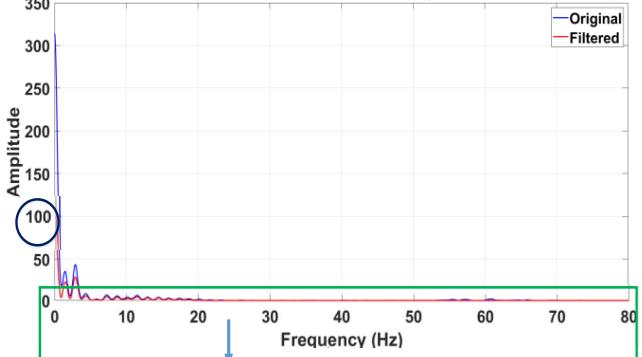
The second test step is to reduce the sampling frequency of the system to 200 Hz. The number of samples is also reduced to 320 samples for 1.6 seconds. The device is connected to the software and displays real-time data. The filter is now redesigned with the high-pass filter coefficients shown in Table XV and the low-pass filter coefficients shown in Table XVI. The results are shown in Fig. 14.

In this test, we re-evaluate the system's ability to process ECG signals in real time. We evaluated the performance of the CPU with the implementation of filters that need high computational capability. The strengths of the dual-core chip of the CPU are promoted to obtain the superior processing

ECG SIGNAL BEFORE AND AFTER APPLING IIR (HIGH-PASS AND LOW-PASS) FILTER



SPECTRUM OF ECG SIGNAL BEFORE AND AFTER APPLING IIR (HIGH-PASS AND LOW-PASS) FILTER



SPECTRUM OF ECG SIGNAL BEFORE AND AFTER APPLING IIR (HIGH-PASS AND LOW-PASS) FILTER

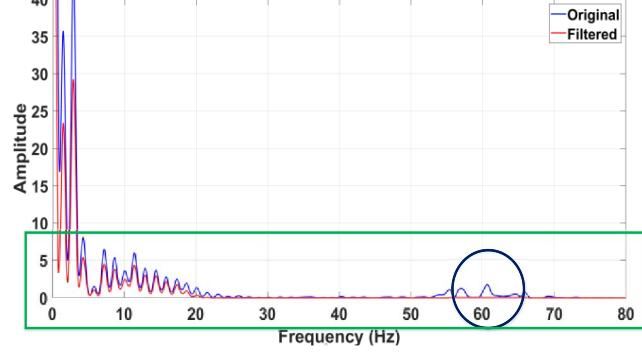


Fig. 12. Real ECG signals after applying the high-pass IIR filter with a cutoff frequency of 0.05 Hz and the low-pass filter with a cutoff frequency of 40 Hz.

TABLE XVI

COEFFICIENTS OF THE IIR Low-Pass Filter (Inv.Chebyshev) AT A SAMPLING FREQUENCY OF 200 Hz, CUTOFF FREQUENCY OF 30 Hz, ORDER OF 7, AND ASTOP OF 80 dB

Sections	Section 1	Section 2	Section 3	Section 4
b[0]	2,242	1,489	750	2,083
b[1]	-2,560	-1,203	239	2,083
b[2]	2,242	1,489	750	0
a[0]	10,000	10,000	10,000	10,000
a[1]	-16,450	-14,019	-12,311	-5,834
a[2]	8,373	5,794	4,049	0

capability with the single-core CPU. The dual-core system is capable of real-time ECG signal processing. It is also suitable for applications for mobile systems.

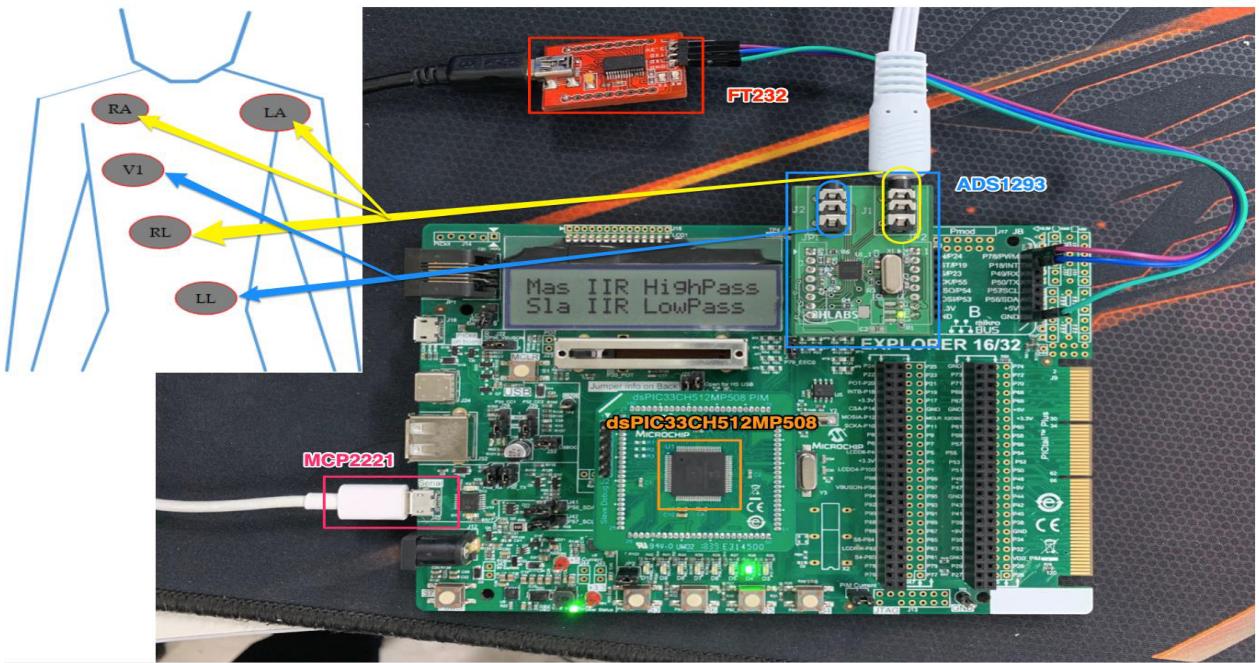


Fig. 13. Real system for testing the ECG signals.



Fig. 14. Real-time processing and display ECG signal.

VI. CONCLUSION

In this paper, we have completed the design of an ECG device that can collect signals in real time. The results achieved with this design include: filtering ECG signal using IIR filter, demonstrating the power of dual-core processor compared to single-core with the following details:

With the results presented in the section above, this system can perform ECG data filtering in real time. The baseline drift

noise and noise power frequency interference at 60 Hz are eliminated from the data after passing the IIR filter.

The use of the dual-core processor has increased processing capacity by nearly 2 times than the single-core processor, which increases the efficiency and responsiveness of the system in real time. The response time of the system depends on the number of samples processed in 1 unit of time.

In the other hand, we focus on assessing data processing capacity of dual-core chips compared with single-core chips. The data processing, algorithms, and specialized calculation modules have not been applied in the firmware. Therefore, the performance of the firmware has not been optimized.

Besides, as the memory limit of the slave core is relatively small, handling real-time data in a relatively short time of 1.6 seconds is only possible. The DMA module is used in the firmware to optimize its operation during the data transfer process with the PC.

The display software was written on MATLAB has a relatively slow data update cycle, which affects real-time ECG signaling. The current software only draws the signal in real time with the sampling frequency of the ECG signal at 200 Hz.

REFERENCES

- [1] *Cardiovascular Diseases (CVDs)*, World Health Org., Geneva, Switzerland, May 2017, p. 17.
- [2] J. Mateo, A. M. Torres, A. Aparicio, and J. L. Santos, "An efficient method for ECG beat classification and correction of ectopic beats," *Comput. Electr. Eng.*, vol. 53, pp. 219–229, Jul. 2016.
- [3] J. Li, G. Deng, W. Wei, H. Wang, and Z. Ming, "Design of a real-time ECG filter for portable mobile medical systems," *IEEE Access*, vol. 5, pp. 696–704, Oct. 2017.
- [4] E. Spano, S. Di Pascoli, and G. Iannaccone, "Low-power wearable ECG monitoring system for multiple-patient remote monitoring," *IEEE Sensors J.*, vol. 16, no. 13, pp. 5452–5462, Jul. 2016.

- [5] M. Levkov, G. Mihov, R. Ivanov, I. Daskalov, I. Christov, and I. Dotsinsky, "Removal of power-line interference from the ECG: A review of the subtraction procedure," *Biomed. Eng. OnLine*, vol. 4, Aug. 2005, Art. no. 23.
- [6] S.-W. Shin, K.-S. Kim, C.-G. Song, J.-W. Lee, J.-H. Kim, and G.-W. Jeung, "Removal of baseline wandering in ECG signal by improved detrending method," *Bio-Medical Mater. Eng.*, vol. 26, no. s1, pp. S1087–S1093, Aug. 2015.
- [7] S. K. Bashar, E. Ding, A. J. Walkey, D. D. McManus, and K. H. Chon, "Noise detection in electrocardiogram signals for intensive care unit patients," *IEEE Access*, vol. 7, pp. 88357–88368, 2019.
- [8] N. Jain and D. K. Shakya, "Denoising baseline wander noise from electrocardiogram signal using fast ICA with multiple adjustments," *Int. J. Comput. Appl.*, vol. 99, no. 2, pp. 34–39, 2014.
- [9] A. Milchevski and M. Gusev, "Improved pipelined wavelet implementation for filtering ECG signals," *Pattern Recognit. Lett.*, vol. 95, pp. 85–90, Aug. 2017.
- [10] R. Sameni, M. B. Shamsollahi, C. Jutten, and M. Babaie-Zade, "Filtering noisy ECG signals using the extended Kalman filter based on a modified dynamic ECG model," in *Proc. Comput. Cardiol.*, Sep. 2005, pp. 1017–1020.
- [11] S. Saxena, R. Jais, and M. K. Hota, "Removal of powerline interference from ECG signal using FIR, IIR, DWT and NLMS adaptive filter," in *Proc. Int. Conf. Commun. Signal Process. (ICCP)*, Apr. 2019, pp. 4–6.
- [12] M. B. Tayel, A. S. Eltrass, and A. I. Ammar, "A new multi-stage combined kernel filtering approach for ECG noise removal," *J. Electrocardiol.*, vol. 51, no. 2, pp. 265–275, Mar./Apr. 2018.
- [13] A. S. Barhatte, R. Ghongade, and S. V. Tekale, "Noise analysis of ECG signal using fast ICA," in *Proc. Conf. Adv. Signal Process. (CASPA)*, Jun. 2016, pp. 9–11.
- [14] J. Mateo, E. M. Sánchez-Morla, and J. L. Santos, "A new method for removal of powerline interference in ECG and EEG recordings," *Comput. Electr. Eng.*, vol. 45, pp. 235–248, Jul. 2015.
- [15] B. Baker, *Delta-Sigma ADCs in a Nutshell*. Newton, MA, USA: EDN, 2007.
- [16] B. Pisani, "Digital filter types in delta-sigma ADCs," Texas Instrum., Dallas, TX, USA, Appl. Rep. SBAA230, May 2017.
- [17] S. Cuomo, G. De Pietro, R. Farina, A. Galletti, and G. Sammino, "A revised scheme for real time ECG signal denoising based on recursive filtering," *Biomed. Signal Process. Control*, vol. 27, pp. 134–144, May 2016.
- [18] M. Chen, X. Zhang, X. Chen, M. Zhu, G. Li, and P. Zhou, "FastICA peel-off for ECG interference removal from surface EMG," *Biomed. Eng. OnLine*, vol. 15, no. 1, p. 65, Dec. 2016.
- [19] C. Beach *et al.*, "An ultra low power personalizable wrist worn ECG monitor integrated with IoT infrastructure," *IEEE Access*, vol. 6, pp. 44010–44021, 2018.
- [20] Y. Liang and Z. Chen, "Intelligent and real-time data acquisition for medical monitoring in smart campus," *IEEE Access*, vol. 6, pp. 74836–74846, 2018.
- [21] J.-Y. Kim and C.-H. Chu, "Analysis and modeling of selected energy consumption factors for embedded ECG devices," *IEEE Sensors J.*, vol. 16, no. 6, pp. 1795–1805, Mar. 2016.
- [22] Z. Li and S. He, "Run-time timing prediction for system reconfiguration on many-core embedded systems," *J. Syst. Archit.*, vol. 95, pp. 47–54, May 2019.
- [23] I. Assayad, "A platform-based design framework for joint SW/HW multiprocessor systems design," *J. Syst. Archit.*, vol. 55, nos. 7–9, pp. 409–420, Jul./Sep. 2009.
- [24] H. Liu, D. Fujita, L. Zhang, and A. Suzuki, "Real-time pulse waveform profiling algorithm for wearable applications," *IEEE Access*, vol. 6, pp. 59296–59306, 2018.
- [25] *dsPIC33CH512MP508*, Microchip, Chandler, AZ, USA, May 2019.
- [26] *ADS1293 24-Bit, 3-Ch, Low-Power Analog Front END (AFE) for ECG Applications*, Texas Instrum., Dallas, TX, USA, Dec. 2014.
- [27] R. Kher, "Signal processing techniques for removing noise from ECG signals," *J. Biomed. Eng. Res.*, vol. 1, pp. 1–9, Mar. 2019.
- [28] *Explorer 16/32 Development Board*, Microchip, Chandler, AZ, USA, Sep. 2014.
- [29] C.-H. Luo *et al.*, "An ECG acquisition system prototype design with flexible PDMS dry electrodes and variable transform length DCT-IV based compression algorithm," *IEEE Sensors J.*, vol. 16, no. 23, pp. 8244–8254, Dec. 2016.
- [30] *Serial Peripheral Interface (SPI) With Audio Codec Support*, Microchip, Chandler, AZ, USA, Jul. 2014.
- [31] *I/O Ports With Edge Detect*, Microchip, Chandler, AZ, USA, Jun. 2018.
- [32] *Direct Memory Access Controller (DMA)*, Microchip, Chandler, AZ, USA, Jan. 2019.
- [33] *Multiprotocol Universal Asynchronous Receiver Transmitter (UART) Module*, Microchip, Chandler, AZ, USA, Jun. 2018.
- [34] X. Dong, C. Wang, and W. Si, "ECG beat classification via deterministic learning," *Neurocomputing*, vol. 240, no. 21, pp. 1–12, May 2017.
- [35] S. Chen, W. Hua, Z. Li, J. Li, and X. Gao, "Heartbeat classification using projected and dynamic features of ECG signal," *Biomed. Signal Process. Control*, vol. 31, pp. 165–173, Jan. 2017.
- [36] *REAL-TIME DSP LABORATORY5: Infinite Impulse Response (IIR) Filters on the C6713 DSK*, Colorado State Univ.-Walter Scott, Jr. College Eng., Fort Collins, CO, USA, 2019.
- [37] P. Plawiak, "ECG signals (1000 fragments)," Nov. 2017.
- [38] U. Satija, B. Ramkumar, and M. S. Manikandan, "A new automated signal quality-aware ECG beat classification method for unsupervised ECG diagnosis environments," *IEEE Sensors J.*, vol. 19, no. 1, pp. 277–286, Jan. 2019.

Ngoc Thang Bui (Student Member, IEEE) received the M.S. degree in automatic control from the Ho Chi Minh City University of Technology, Vietnam, in 2012. He is currently pursuing the Ph.D. degree in biomedical engineering with Pukyong National University. His research interests include high-speed A/D converter and biomedical data analysis.



Duc Tri Phan received the B.S. degree in mechanical engineering from the Ho Chi Minh City University of Technology, Vietnam, in 2018. He is currently pursuing the M.S. degree in biomedical engineering with Pukyong National University. His research interest includes biomedical signal analysis.



Thanh Phuoc Nguyen received the B.S. degree in mechatronics and the M.S. degree in automatic control from the Ho Chi Minh City University of Technology, Vietnam, in 2002 and 2009, respectively, and the Ph.D. degree in biomedical engineering from Pukyong National University, South Korea, in 2019. His current research interests include biomedical data analysis and the design, modeling, and fabrication of high-frequency ultrasonic transducers for medical imaging applications.



Giang Hoang received the M.S. and Ph.D. degrees in mechatronics from Pukyong National University Busan, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in biomedical engineering.

His research interest includes scanning acoustic microscopy.





Jaeyeop Choi received the B.S. degree in biomedical engineering from Pukyong National University, South Korea, in 2018, where he is currently pursuing the M.S. degree in biomedical engineering. His research interest includes scanning acoustic microscopy.



Junghwan Oh (Member, IEEE) received the B.S. degree in mechanical engineering from Pukyong National University in 1992, and the M.S. and Ph.D. degrees in biomedical engineering from The University of Texas at Austin, USA, in 2003 and 2007, respectively. His research interest is in the development of novel light and ultrasonic-based diagnostic imaging modalities for biomedical engineering applications.



Quoc Cuong Bui received the B.S. degree in control engineering and automation from the Hanoi University of Science and Technology, Vietnam, in 2019. He is currently pursuing the M.S. degree in biomedical engineering with Pukyong National University. His research interest includes biomedical signal analysis.