

O. D. ÇAĞATAY

IZMIR KATIP CELEBI UNIVERSITY

2019

**IZMIR KATIP CELEBI UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**DESIGN OF ARDUINO SHIELD TO RECORD REAL-TIME
ELECTROCARDIOGRAPHY SIGNALS**



M.Sc. THESIS

Onur Doğanay ÇAĞATAY

Department of Biomedical Technologies

MAY 2019

**IZMIR KATIP CELEBI UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES**

**DESIGN OF ARDUINO SHIELD TO RECORD REAL-TIME
ELECTROCARDIOGRAPHY SIGNALS**

M.Sc. THESIS

**Onur Dođanay AĐATAY
(Y150101022)**

Department of Biomedical Technologies

Thesis Advisor: Asst. Prof. Dr. Yalın İŐLER

MAY 2019

İZMİR KATİP ÇELEBİ ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ

ELEKTROKARDİYOĞRAFI SİNYALLERİNİN GERÇEK ZAMANLI KAYDI
İÇİN ELEKTRONİK FİLTRELİ ARDUNİO EKLENTİSİ TASARIMI

YÜKSEK LİSANS TEZİ

Onur Doğanay ÇAĞATAY
(Y150101022)

Biyomedikal Teknolojiler Ana Bilim Dalı

Tez Danışmanı: Dr. Öğr. Üyesi Yalçın İŞLER

MAYIS 2019

Onur Dođanay AĐATAY, a M.Sc. student of **IKCU Graduate School Of Natural And Applied Sciences**, successfully defended the thesis entitled “**DESIGN OF ARDUNIO SHIELD TO RECORD REAL-TIME ELECTROCARDIOGRAPHY SIGNALS**”, which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :

Asst. Prof. Dr. Yalm İŐLER
İzmir Katip elebi University



Jury Members :

Assoc. Prof. Dr. SavaŐ ŐAHİN
İzmir Katip elebi University



Asst. Prof. Dr. Rukiye UZUN
Zonguldak Blent Ecevit University



Date of Submission : 22.04.2019

Date of Defense : 20.05.2019





To my family,



FOREWORD

I would like to thank my advisor, Assistant Professor Yalçın İŞLER for guiding and supporting me over the period of my thesis study.

I would like to thank my family for love, support, and encouragement against the difficulties I faced over the years. In particular, I would like to thank my father Mansur ÇAĞATAY, my mother Ayşe ÇAĞATAY and my sisters Didem and Derya ÇAĞATAY.

May 2019

Onur Doğanay ÇAĞATAY



TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	ix
TABLE OF CONTENTS	xi
ABBREVIATIONS	xiii
LIST OF TABLES	xv
LIST OF FIGURES	xvii
ABSTRACT	xix
ÖZET	xxi
1. INTRODUCTION	1
1.1 Definition of the ECG	1
1.2 Background of the ECG	1
1.3 Purpose of the Thesis	2
2. METHODS	3
2.1 Schematic Design	3
2.1.1 Power section	3
2.1.2 Input selection section.....	6
2.1.3 Gain selection section	7
2.1.4 Low-pass filter section	9
2.1.5 High-pass filter section	11
2.1.6 Notch filter section.....	13
2.1.7 Output selection section	13
2.1.8 Led indicators section	15
2.2 Printed Circuit Board Design	15
2.3 Embedded Software Design	16
2.4 PC Software Design	16
2.4.1 Main window	17
2.4.2 Open button.....	17
2.4.3 Port tab	18
2.4.4 Amp tab.....	18
2.4.5 LPF&HPF tab	19
2.4.6 Channel tab	19
2.4.7 The graphic plot area.....	20
3. RESULTS	21
3.1 Input Selection.....	21
3.2 Gain Selection	21
3.3 Low-Pass Filter Selection.....	22
3.4 High-Pass Filter Selection	22
3.5 Output Selection	23
3.6 Whole Proposed System.....	23
3.7 Printed Circuit Board of the Proposed System.....	24
3.8 Acquiring the ECG Signal via the Proposed System	25

4. DISCUSSION 29
REFERENCES 31
APPENDIX A: The Arduino Program..... 33
APPENDIX B: The PLOT Function of the Computer Program..... 38
APPENDIX C: The Pin Description of the 74HC4052 39
CURRICULUM VITAE 41



ABBREVIATIONS

ADC	: Analog-to-Digital Converter
char	: Character Type Variable
ECG	: Electrocardiography (or Electrocardiogram)
EKG	: Elektrokardiyografi (veya Elektrokardiyogram)
f_c	: Cut-Off Frequency
float	: Floating-Point Type Variable
GUI	: Graphical User Interface
H	: High Voltage Level (Logic 1)
HPF	: High-Pass Filter
Hz	: Hertz
IC	: Integrated Circuit
IDE	: Integrated Development Environment
int	: Integer Type Variable
KHz	: Kilohertz
KΩ	: Kiloohm
L	: Low Voltage Level (Logic 0)
LPF	: Low-Pass Filter
MΩ	: Megaohm
ms	: Millisecond
mV	: Millivolts
NC	: Not Connected
Ω	: Ohm
op-amp	: Operational Amplifier
PC	: Personal Computer
PCB	: Printed Circuit Board
SMD	: Surface-Mount Device
V	: Volts
X	: Don't Care Level (Logic X)

LIST OF TABLES

	<u>Page</u>
Table 2.1 : The function table of the 74HC4051.	7
Table 2.2 : Required gain resistor (R_INA_GAIN) values for desired gain values of both common instrumentation amplifiers of INA128 and INA129.	8
Table 2.3 : Required resistor values for low-pass filter.	11
Table 2.4 : Required resistor values for the high-pass filter.	13
Table 2.5 : Function table of the 74HC4052.	14





LIST OF FIGURES

	<u>Page</u>
Figure 2.1 : The schematic of the whole electronic design.....	3
Figure 2.2 : Power design.	4
Figure 2.3 : Switching power regulator for negative voltages.....	4
Figure 2.4 : Noisy switching power voltage.	5
Figure 2.5 : Negative linear voltage circuit design.	5
Figure 2.6 : Input selection circuit design.....	6
Figure 2.7 : Gain selection circuit design.	8
Figure 2.8 : Offset circuit design.	9
Figure 2.9 : The Sallen Key Low-Pass Filter.....	9
Figure 2.10 : Bode diagram of a Sallen Key Low-Pass Filter.	10
Figure 2.11 : The Sallen Key High-Pass Filter.	11
Figure 2.12 : Bode diagram of a Sallen Key High-Pass Filter.....	12
Figure 2.13 : Twin T-Notch filter.	13
Figure 2.14 : The selection of the output signal.....	14
Figure 2.15 : LED indicators to show the status of the proposed design.....	15
Figure 2.16 : The PCB of the whole electronic design.	15
Figure 2.17 : Arduino IDE.	16
Figure 2.18 : The interface of the PC program.	17
Figure 2.19 : The Open button.	17
Figure 2.20 : The PORTS tab to determine which channel will be connected to the shield.	18
Figure 2.21 : The AMP tab to determine which amplification ratio will be applied to the ECG signal.	18
Figure 2.22 : The LPF&HPF tab to determine cut-off frequencies of both low-pass and high-pass filters.	19
Figure 2.23 : The Channel tab to choose which ECG signal will be transferred to the actual output (or to the ADC).....	20
Figure 2.24 : The graphic output of the ECG signal on the computer program.	20
Figure 3.1 : Input combinations.	21
Figure 3.2 : Gain selection.	22
Figure 3.3 : Cut-off frequency selection for the LPF.....	22
Figure 3.4 : Cut-off frequency selection for the HPF.	23
Figure 3.5 : Output selection.....	23
Figure 3.6 : The whole block diagram of the proposed system.	24
Figure 3.7 : The implemented system.....	25
Figure 3.8 : A real application of the proposed system.	25
Figure 3.9 : The oscilloscope screen of an amplified but filtered ECG signal.	26
Figure 3.10 : The oscilloscope screen of an amplified and filtered ECG signal.....	26
Figure 3.11 : The real-time acquisition of an ECG signal.	27



DESIGN OF ARDUINO SHIELD TO RECORD REAL-TIME ELECTROCARDIOGRAPHY SIGNALS

ABSTRACT

In this thesis, an Arduino extension board (shield) is designed to take the electrocardiography (ECG) signals. Firstly, the positive and negative terminals of an ECG signal are selected to determine which input pins are connected to these terminals. User can select one of eight possible inputs for each terminal, which means the user can apply one of the possible 64 channel combinations to inputs of the design. Then a dedicated instrumental amplifier amplifies the signal. The voltage-gain of the amplifier can be selectable among available options. The amplified ECG signal is filtered in an order by a low-pass filter, a high-pass filter, and a notch filter. The user selects the cut-off frequencies of the low-pass and the high-pass filters in addition to the center frequency of the notch filter from available predetermined options. Finally, the user can prefer to transfer one of these outputs from these stages to either the Arduino board or the developed computer program. As a result, a fully flexible ECG shield is designed.



ELEKTROKARDİYOĞRAFI SİNYALLERİNİN GERÇEK ZAMANLI KAYDI İÇİN ELEKTRONİK FİLTRELİ ARDUINO EKLENTİSİ TASARIMI

ÖZET

Bu tezde, elektrokardiyografi (EKG) sinyallerinin alınabilmesi için bir Arduino genişleme kartı tasarlanmıştır. Öncelikle, EKG sinyalinin pozitif ve negatif uçlarının kartın hangi bağlantı noktalarına bağlandığı seçilir. Kullanıcı her bağlantı noktası için sekiz farklı olası girişten birisini seçebilir. Böylece kullanıcı 64 farklı giriş kombinasyonunun birisini seçerek kartın girişlerine uygulayabilir. Sonra sinyal özel bir enstrümantasyon yükselteci tarafından kuvvetlendirilir. Yükseltecin gerilim kazancı sunulan seçenekler arasından seçilebilir. Kuvvetlendirilen EKG sinyali bir alçak geçiren, bir yüksek geçiren ve bir çentik filtre tarafından süzülür. Alçak geçiren ve yüksek geçiren filtrelerin kesim frekansları ile çentik filtrenin merkez frekansı önceden tanımlı olarak sunulan mevcut seçenekler arasından kullanıcı tarafından seçilir. Son olarak, kullanıcı bu çıkışlardan birini Arduino kartına veya geliştirilen bilgisayar programına aktarılmasını tercih edebilir. Sonuç olarak, tamamen esnek bir EKG genişletme kartı tasarlanmıştır.



1. INTRODUCTION

Electrocardiography (ECG) shows the electrical activity of a heart. Augustus D. Waller measured the first ECG signal in 1887. In 1901, Einthoven made changes on string galvanometry to produce ECG as a graph [1]. In 1902, Einthoven published the first recorded electrocardiography with a string galvanometer. In 1906, Einthoven was recorded with string galvanometry and published the first organized presentation of abnormal ECGs [2;3]. In 1912, Einthoven interviewed at Chelsea Clinical Society in London. This is the first place wherever the ECG was used [3].

1.1 Definition of the ECG

ECG is the recording of electrical activity in the heart to examine the operation of the heart muscle and the neural transmission system. This chart is called electrocardiograph (ECG). The instrument to produce this graph is also called electrocardiogram (ECG). Potential changes that occur with the electrical activity of the heart are spread through the whole body simultaneously with the help of tissues around the heart. In order to obtain such a signal, two electrodes are placed onto two different body parts and the voltage difference between these electrodes are recorded and visualized [4].

1.2 Background of the ECG

The ECG signal is a biological signal that can be detected by using electrodes attached to the body, resulting from the electrical activity of the heart muscles that provide the heart to pump blood. The ECG is used for the clinical diagnosis of various diseases associated with the heart. The ECG signal consists of fiducial points symbolized by the letters of P, Q, R, S, T, and U [5]. Each of these letters symbolizes each phase of the movement of the heart muscle to pump blood. The voltage and frequency levels of the ECG signal determine the abnormalities of the heart. The ECG is limited by the frequency band of 0.1 - 150 Hz and the voltage amplitudes of

10-2 - 10-4 V. That is the reason why an ECG must be amplified to the working voltage level of today's microcontrollers and must be filtered from unwanted noisy components [5;6].

1.3 Purpose of the Thesis

All ECG related experiments are conducted using professional ECG devices for clinical purposes. There are many commercial devices available in the market. On the other hand, there are only a few options when a researcher wants to develop a new device by self. In a new design, the researcher should use a microprocessor- or a microcontroller-based system. In addition, a novice student who wants to experiment related to basic ECG experiments should buy commercial devices or try to fit their studies into the limitations given by the available electronic boards.

In this thesis, it is aimed to design a full-flexible ECG data acquisition board

- that is pin compatible to an Arduino board (which is a famous microcontroller board),
- whose all parameters including cut-off frequencies of the filters and voltage-gain can be chosen among available well-suited options by using the Arduino sketch of the computer program,
- whose signals of each stage can be viewed on the computer screen, and
- whose signals of each stage can be transferred directly to the Arduino board.

2. METHODS

2.1 Schematic Design

There are ten distinct sections in the electronic design (Figure 2.1). These sections are Power, Inputs, Gain of The Signal, Active Low Pass Filter, Active High Pass Filter, Notch Filter Selection Signal and Decoupling. These parts are described in detail in the following sections [7;8].

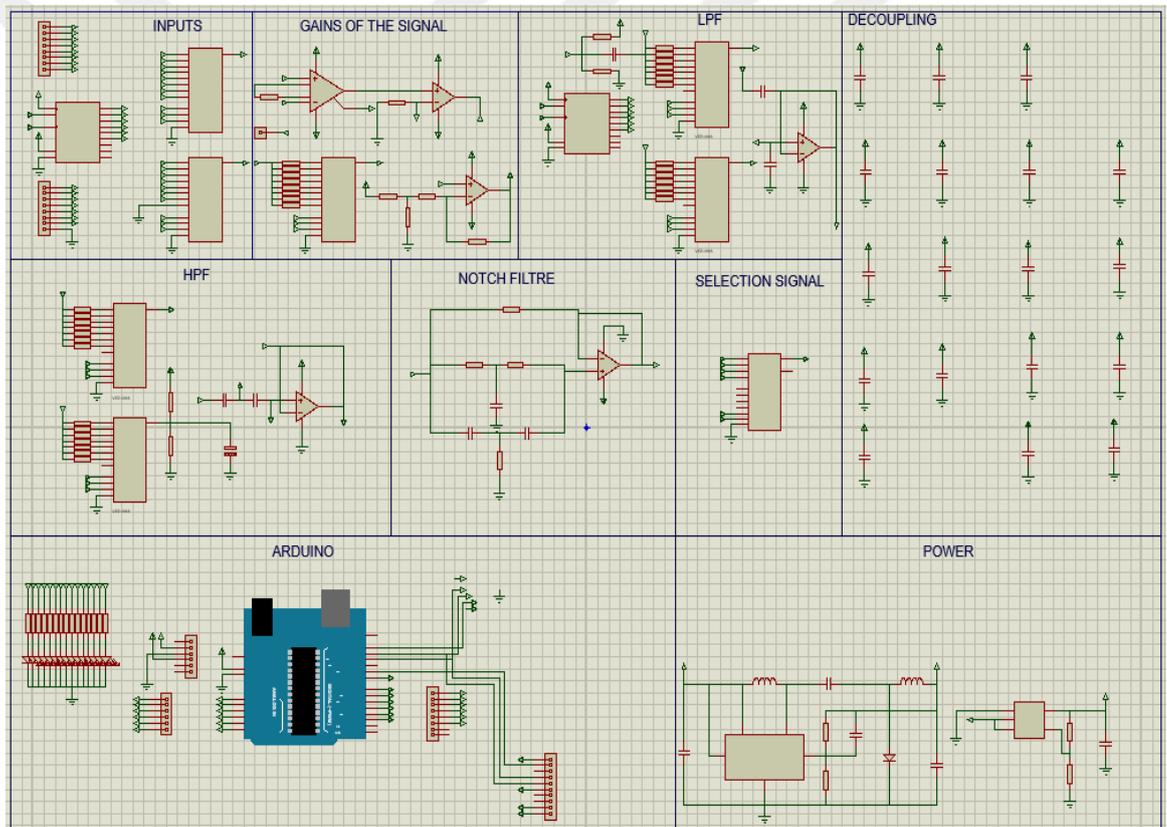


Figure 2.1 : The schematic of the whole electronic design.

2.1.1 Power section

There are two power supply voltage levels (Figure 2.2) that are required by amplifier and filter circuit ICs. The first power supply is the Positive Power Supply. The other one is the Negative Power Supply.

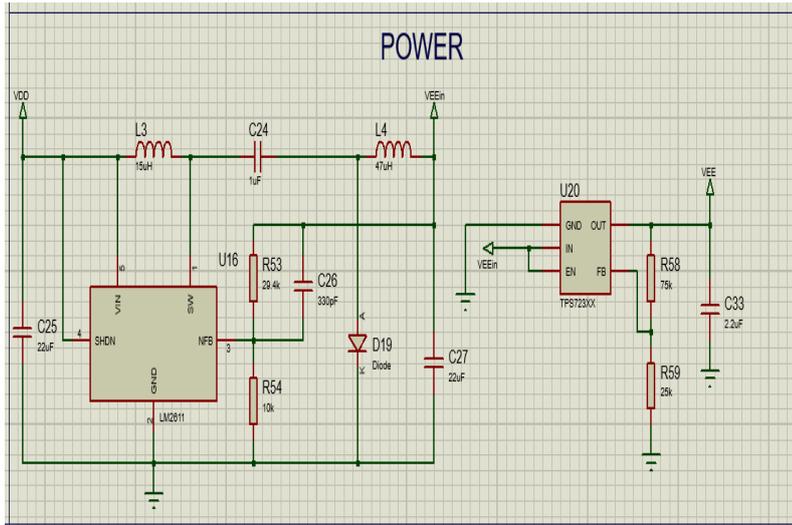


Figure 2.2 : Power design.

The ECG shield takes its positive power voltage from the Arduino Uno Board. There is a linear regulator on the Arduino Uno board which produces +5V from +12V. If the Arduino Uno board is powered by a PC via USB port, +5V is supplied to the ECG Shield directly.

The Negative Power Supply has two parts. In the first part, a switching regulator is used (Figure 2.3) [8].

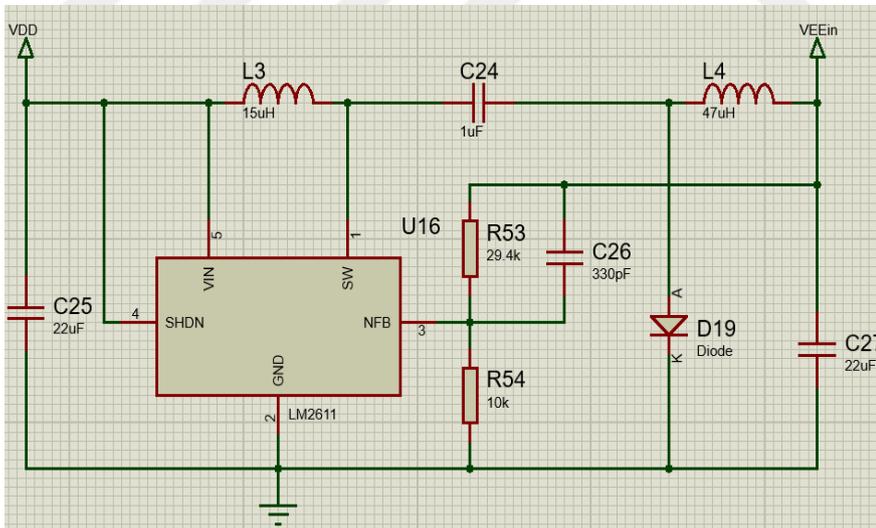


Figure 2.3 : Switching power regulator for negative voltages.

In the circuit design, the input power voltage (VDD) is +5V. The negative voltage is obtained by a switching regulator. The resistors in the design provide to adjust the negative voltage value. The negative voltage output value (V_{EEin}) is given;

$$V_{EEin} = -1.25x \frac{(R_{53} + R_{54})}{R_{54}} \quad (2.1)$$

where R_{53} and R_{54} are adjunct resistors for the negative voltage regulator circuit.

The problem with the switching regulator design is noise. The circuit produces noise on the output voltage [9]. The noisy voltage can be seen in Figure 2.4.

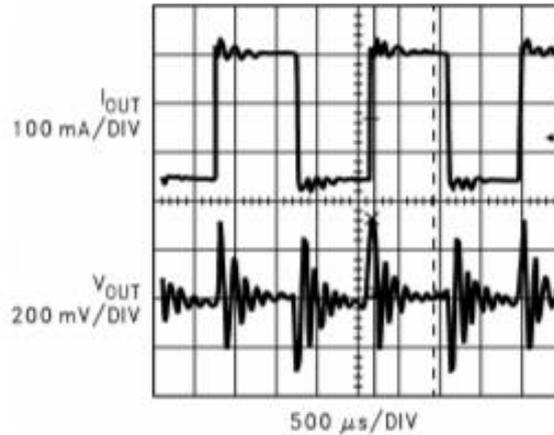


Figure 2.4 : Noisy switching power voltage.

Because of the noise, a linear regulator is used after the switching regulator circuit in general. In the second part, a linear regulator is used to eliminate such a noise on the negative voltage. After the linear regulator, the noise on the negative supply voltage is eliminated. In addition, a decoupling capacitor should be used for an unexpected noise. The linear negative supply voltage circuit is given in Figure 2.5.

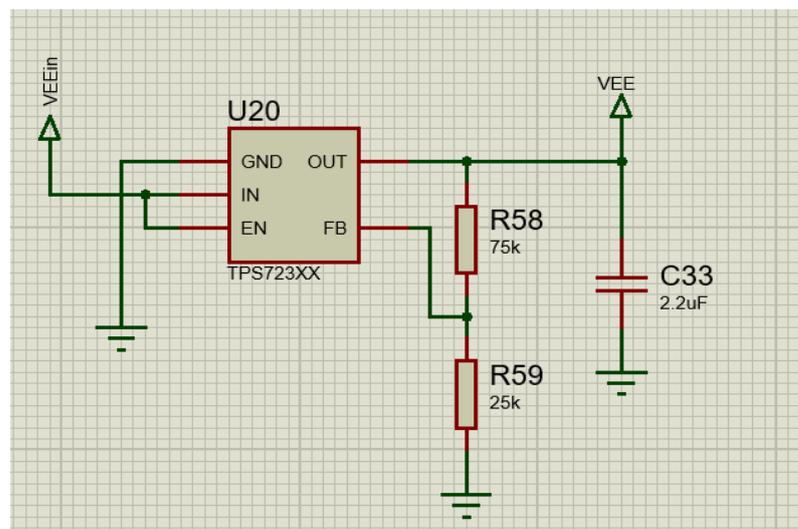


Figure 2.5 : Negative linear voltage circuit design.

The negative voltage output value (VEE) is given;

$$VEE = -1.25x \frac{(R_{58} + R_{59})}{R_{59}} \quad (2.2)$$

where R_{58} and R_{59} are adjunct resistors for the negative voltage regulator circuit. In this case, the maximum value of VEE will be less than the maximum of its input voltage of VEE_{in} .

2.1.2 Input selection section

The input selection has two distinct parts (Figure 2.6). These parts are responsible to determine which input combination will be applied to the other signal processing stages and acquired at the output of the proposed design.

In the first part, the IC of 74HC595, which is a shift register, is used to generate selection signals for both the positive and the negative terminals of the signal under interest. There are three selection pins for each terminal, which means eight possible input channels for each. The Arduino board transfers these selection signals by using data (DS) and the clock (ST_CP) pins. Since there are eight possible channels for each terminal, there 64 possible signal terminal combinations. The positive terminal is shown in In1 pin and the negative terminal is shown in In2 pin in the figure.

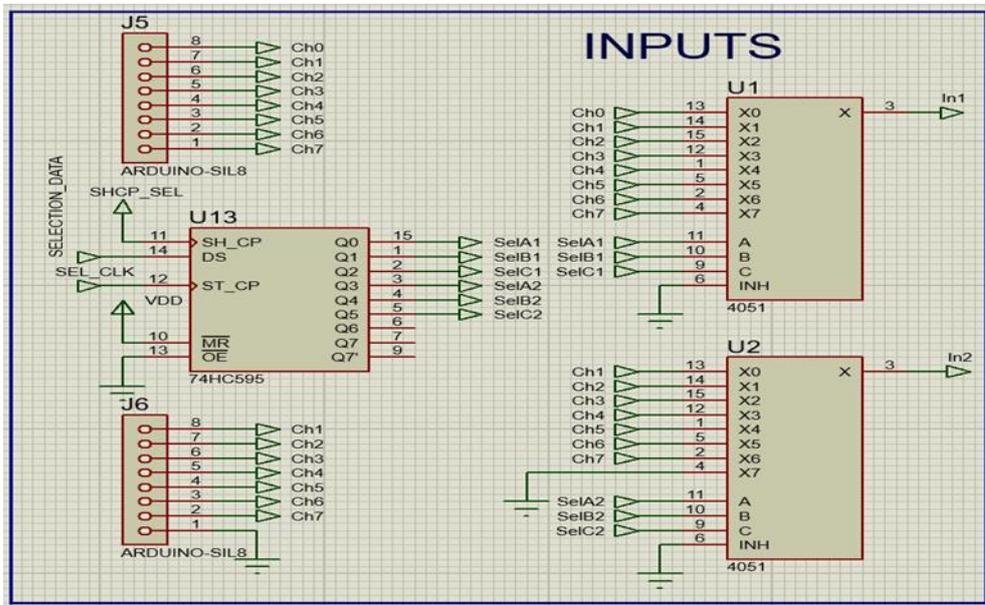


Figure 2.6 : Input selection circuit design.

The second part of the input selection circuit is based on the 74HC4051 ICs that are 8-channel analog multiplexers and demultiplexers. The IC provides the select the

input pins. It can do it with using three pins. In the PC program, the pins are determined logic high or low. U1 and U2 ICs are 74HC4051 and the select input pins of them are connected to the Arduino Uno Board input-output pins. So the output of this IC can be selected. The function table is shown below (Table 2.1) [10].

Table 2.1 : The function table of the 74HC4051.

Inputs				Output
E	S2	S1	S0	
L	L	L	L	Y0 to Z
L	L	L	H	Y1 to Z
L	L	H	L	Y2 to Z
L	L	H	H	Y3 to Z
L	H	L	L	Y4 to Z
L	H	L	H	Y5 to Z
L	H	H	L	Y6 to Z
L	H	H	H	Y7 to Z
H	X	X	X	Switch off

The select input pins of 74HC4051 are S2, S1, and S0. As a result, the output of this IC depends on the values of these select pins. For example, the output (Z) of 74HC4051 will be the value of 3rd input (Y3) if S2 is logic low, S1 is logic high, and S0 is logic high. Similarly, the output (Z) of 74HC4051 will be the value of 5th input (Y5) if S2 is logic high, S1 is logic low, and S0 is logic high. For this run mode, the enable pin (E) must be connected to a logic low level.

On the other hand, if the enable pin (E) is logic high, the IC does not give an output signal. In this case, it generates a high impedance.

2.1.3 Gain selection section

This section is designed to amplify the signal amplitude (Figure 2.7). It has four distinct parts. In the first part, INA128 (INA129 may also be preferred instead) is used to amplify for the first time. The gain values are defined in the datasheet as tables (Table 2.2) [11].

The second part of this schematic is a non-inverting amplifier using a special operational amplifier (op-amp). The voltage gain of this part can be calculated by

$$A_v = 1 + \frac{R_{GAIN,i}}{R_{GAIN,o}} \quad (2.3)$$

where i is the index to selected resistor value from 1 to 8. When the voltage gain is selected in the computer program, the corresponding gain resistor is connected to the op-amp.

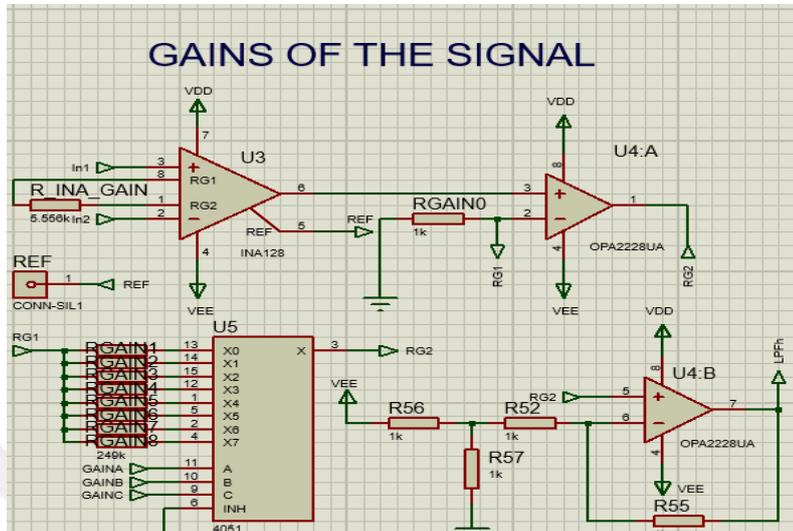


Figure 2.7 : Gain selection circuit design.

Table 2.2 : Required gain resistor (R_{INA_GAIN}) values for desired gain values of both common instrumentation amplifiers of INA128 and INA129.

DESIRED GAIN (V/V)	INA128	INA129
1	NC	NC
2	50K	49.4K
5	12.5K	12.35K
10	5.556K	5489
20	2.632K	2600
50	1.02K	1008
100	505.1	499
200	251.3	248
500	100.2	99
1000	50.05	49.5
2000	25.01	24.7
5000	10.00	9.88
10000	5.001	4.94

The third part provides the selection of the voltage gain by using a 74HC4051 IC. From the related literature, we obtained the required gains as 100, 200, 250, 500, and 1000. Although eight possible voltage gains are available in the design, we used five of them.

The final part is also a non-inverting op-amp amplifier circuit with a fixed gain of 2. In order to eliminate zero drift and additional noises derived from larger amplification in a single step, this multi-stage amplification is preferred [12].

2.1.4 Low-pass filter section

This section is designed to eliminate the high-frequency components of the signal. It has two adjacent parts. In the first part, a DC offset voltage of 2.5V is added to the ECG signal. Since there are no negative voltages in the ADC peripheral of the Arduino boards, the applied analog signal should be between 0 V and 5 V. The offset design is given in Figure 2.8 [13].

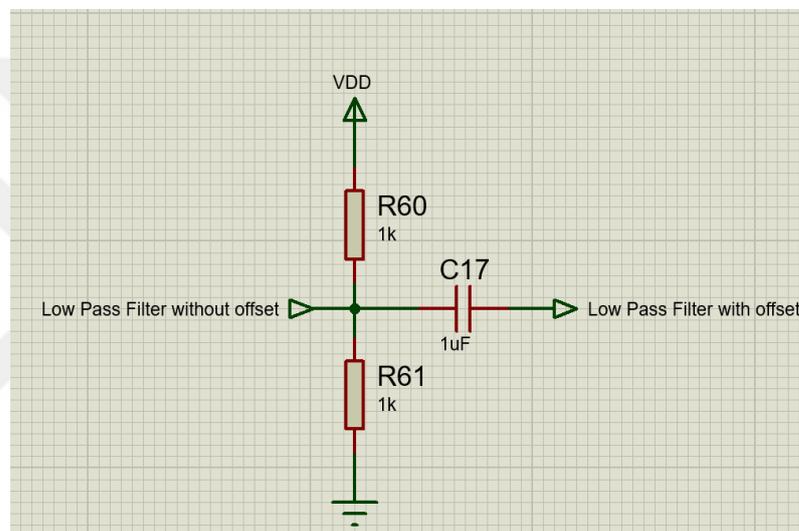


Figure 2.8 : Offset circuit design.

In the second part of this section, a Sallen-Key Low-Pass Filter circuit is used (Figure 2.9) [14]. This filter has a second-order design.

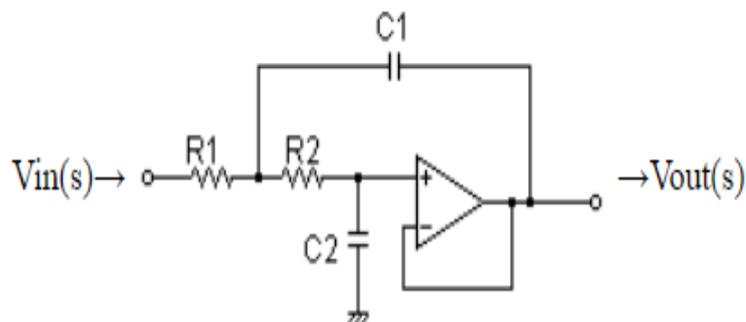


Figure 2.9 : The Sallen Key Low-Pass Filter.

The transfer function of the Sallen Key low-pass filter is

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{R_1 C_1 R_2 C_2} \frac{1}{s^2 + s \left(\frac{1}{R_2 C_1} + \frac{1}{R_1 C_1} \right) + \frac{1}{R_1 C_1 R_2 C_2}} \quad (2.4)$$

The cut-off frequency of this filter depends on capacitors and resistors given in the design. The formula of the cut-off frequency can be calculated by [15]

$$f_c = \frac{1}{2\pi\sqrt{R_1 C_1 R_2 C_2}} \quad (2.5)$$

For example, when we used that $R_1 = R_2 = 1.6k\Omega$ and $C_1 = C_2 = 0.1\mu F$, then the cut-off frequency is [14;16]

$$f_c = 994.7184Hz \quad (2.6)$$

The Bode diagram for this example gives the magnitude and phase changes with respect to frequency (Figure 2.10). The magnitude reduces drastically after the calculated cut-off frequency. Therefore, if the frequency of the input signal of the Sallen Key low-pass filter circuit is lower than 1 KHz, the output signal has the same amplitude. On the other hand, if the frequency of the input signal is higher than 1 KHz, the output signal becomes attenuated.

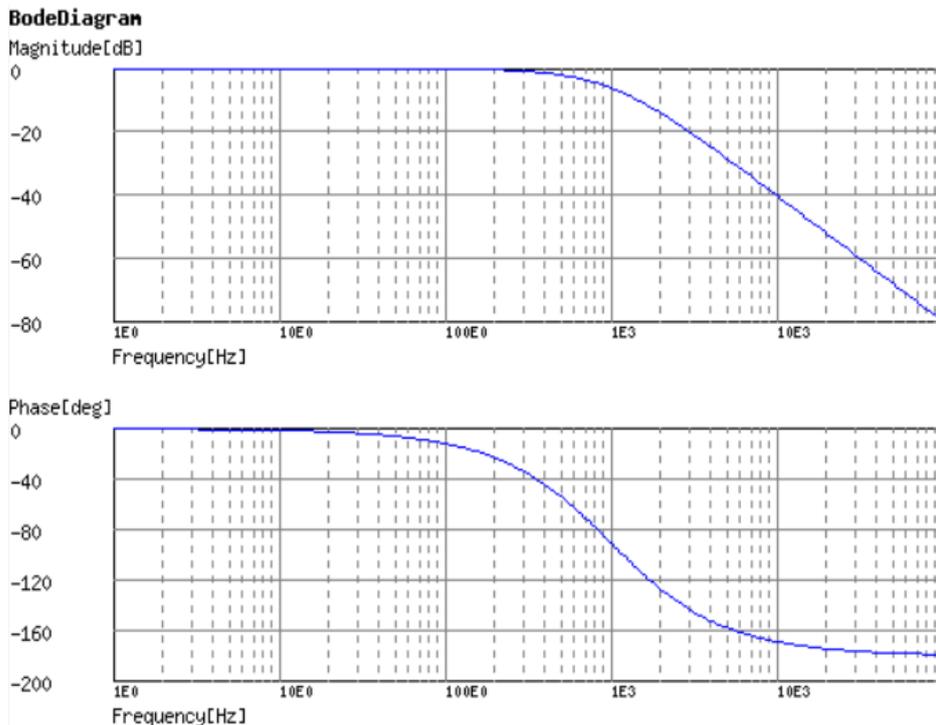


Figure 2.10 : Bode diagram of a Sallen Key Low-Pass Filter.

Since there are two capacitors and two resistors in the circuit, the cut-off frequency is adjusted by changing these four values. In the design, capacitors are fixed to 100 nF. Therefore, the cut-off frequency of the filter can be adjusted by changing the values of R_1 and R_2 . To change the values of these resistors, two distinct 74HC4051 ICs are integrated into the design. The list of resistor pairs and the corresponding cut-off frequency is given in Table 2.3 [14].

Table 2.3 : Required resistor values for low-pass filter.

Cut-off frequency	R_1	R_2
50 Hz	33 K Ω	33 K Ω
100 Hz	16 K Ω	16 K Ω
200 Hz	8.2 K Ω	8.2 K Ω
500 Hz	3.3 K Ω	3.3 K Ω
1 KHz	1.6 K Ω	1.6 K Ω
2 KHz	820 Ω	820 Ω

2.1.5 High-pass filter section

This section is designed to eliminate the low-frequency components from the ECG signal. It has a similar circuit with the LPF design but the offset part. The other change in the circuit by comparing to the LPF is the location changes between resistors and capacitors (Figure 2.11). This filter has a second-order design.

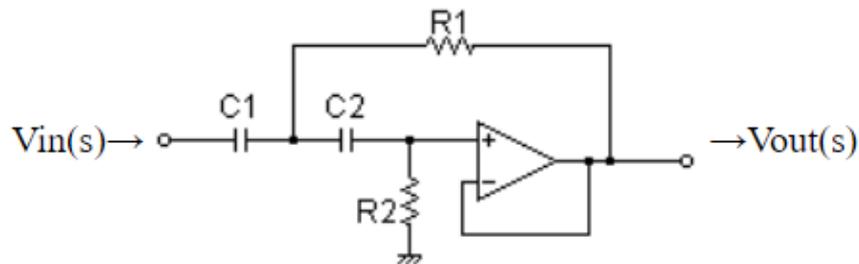


Figure 2.11 : The Sallen Key High-Pass Filter.

The transfer function of the Sallen Key high-pass filter is

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{s^2}{s^2 + s \left(\frac{1}{R_2 C_1} + \frac{1}{R_2 C_2} \right) + \frac{1}{R_1 C_1 R_2 C_2}} \quad (2.7)$$

The cut-off frequency of this filter depends on capacitors and resistors given in the design. The formula of the cut-off frequency can be calculated by [17;18]

$$f_c = \frac{1}{2\pi\sqrt{R_1 C_1 R_2 C_2}} \quad (2.8)$$

For example, when we used that $R_1 = R_2 = 15 \text{ k}\Omega$ and $C_1 = C_2 = 1 \mu\text{F}$, then the cut-off frequency is

$$f_c = 10.6103 \text{ Hz} \quad (2.9)$$

The Bode diagram for this example gives the magnitude and phase changes with respect to frequency (Figure 2.12). The magnitude increases drastically before the calculated cut-off frequency. Therefore, if the frequency of the input signal of the Sallen Key high-pass filter circuit is higher than 10 Hz, the output signal has the same amplitude. On the other hand, if the frequency of the input signal is lower than 10 Hz, the output signal becomes attenuated.

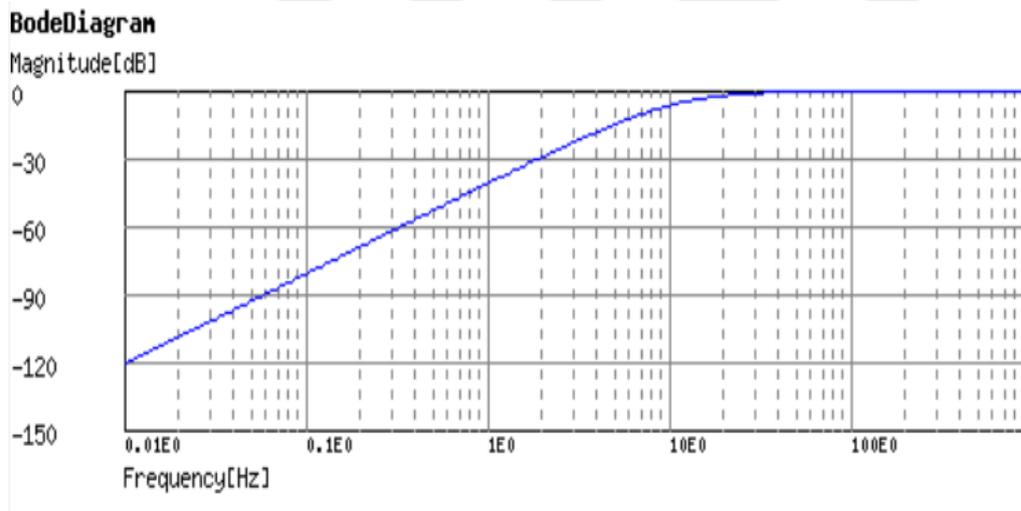


Figure 2.12 : Bode diagram of a Sallen Key High-Pass Filter.

Since there are two capacitors and two resistors in the circuit, the cut-off frequency is adjusted by changing these four values. In the design, capacitors are fixed to $1 \mu\text{F}$. Therefore, the cut-off frequency of the filter can be adjusted by changing the values of R_1 and R_2 . To change the values of these resistors, two distinct 74HC4051 ICs are integrated into the design. The list of resistor pairs and the corresponding cut-off frequency is given in Table 2.4 [14].

Table 2.4 : Required resistor values for the high-pass filter.

Cut-off frequency	R ₁	R ₂
0.01 Hz	16 MΩ	16 MΩ
0.05 Hz	3.3 MΩ	3.3 MΩ
0.1 Hz	1.6 MΩ	1.6 MΩ
0.5 Hz	330 KΩ	330 KΩ
1 Hz	160 KΩ	160 KΩ
10 Hz	16 KΩ	16 KΩ

2.1.6 Notch filter section

This section is designed to reduce the interference from the power line frequency that is 50 Hz in some countries and 60 Hz in other countries [19]. In this thesis, the Twin T-Notch Filter is used to isolate 50 Hz from the ECH signal (Figure 2.13).

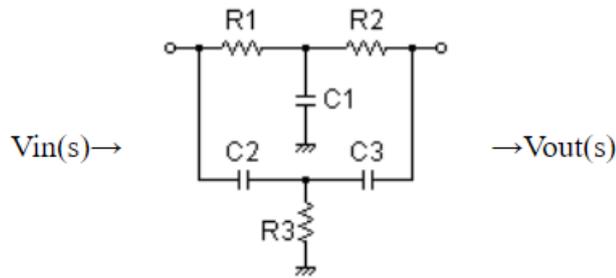


Figure 2.13 : Twin T-Notch filter.

The center frequency of the Twin T-Notch Filter can be calculated by [20]

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{1}{C_2 C_3 R_3 (R_1 + R_2)}} \quad (2.10)$$

$$f_0 = \frac{1}{2\pi} \sqrt{\frac{\frac{1}{C_3} + \frac{1}{C_2}}{C_1 R_1 R_2}} \quad (2.11)$$

For example, 50 Hz component of the ECG signal can be eliminated by determining $R_1 = R_2 = 47k\Omega$, $R_3 = 12k\Omega$, and $C_1 = C_2 = C_3 = 0.1\mu F$ [19; 20].

2.1.7 Output selection section

This section is designed to transfer one of the processed ECG signals to the output (V_{IN++}). For this purpose, a 74HC4052, which is a 4-channel analog multiplexer and

demultiplexer IC, is used to select this signal to the ADC pin of the Arduino Uno board. The pin description is given in Appendix C [21;22].

The possible output signals are determined as amplified ECG signal (V_{IN+}), amplified and low-pass filtered ECG signal (LPF), amplified and low-pass filtered and high-pass filtered ECG signal (HPF), amplified and low-pass filtered and high-pass filtered and notch-filtered ECG signal (NOTCH) by using two selection signals (Figure 2.14). The functional behavior of this section is summarized in Table 2.5.

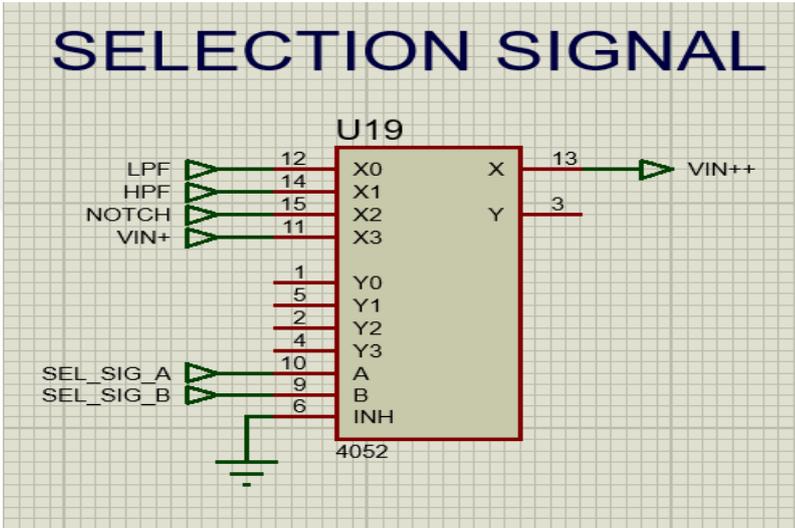


Figure 2.14 : The selection of the output signal.

Table 2.5 : Function table of the 74HC4052.

Inputs		Output
SEL_SIG_B	SEL_SIG_A	V_{IN++}
L	L	LPF
L	H	HPF
H	L	NOTCH
H	H	V_{IN+}

There is an inhibitory (INH) pin in the IC. If this pin is logic low, then the IC obeys the function table as given. On the other hand, if this pin is logic high, then the IC does not give an output signal and it adjusts its outputs as high impedances [22].

2.1.8 Led indicators section

This section is designed to show the status of the extension board. There are 15 LEDs that show the status of the corresponding selection signal (Figure 2.15). In fact, these are selection pins of all 74HC4051 and 74HC4052 multiplexer ICs.

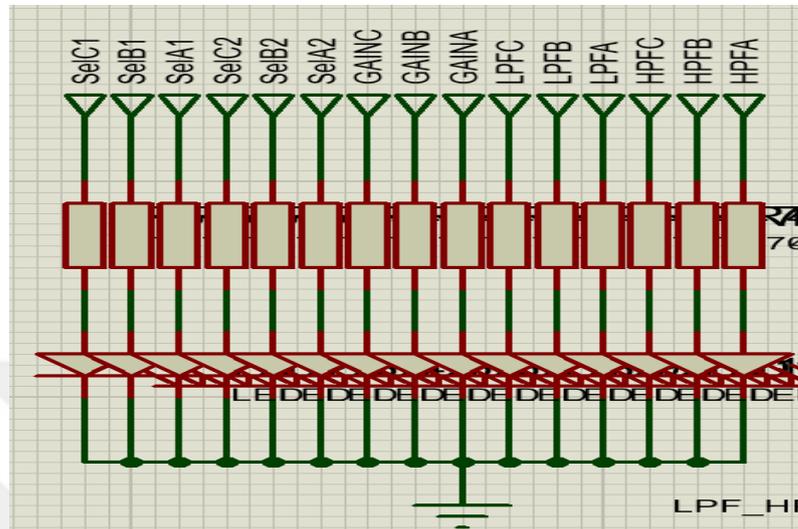


Figure 2.15 : LED indicators to show the status of the proposed design.

2.2 Printed Circuit Board Design

A two-sided printed circuit board (PCB) is designed to place all components together (Figure 2.16). Extra GND places are placed on both sides of the PCB. In addition, extra vias and pads are located around the PCB to generate a weak Faraday cage. These two design tricks are applied to reduce noises [23].

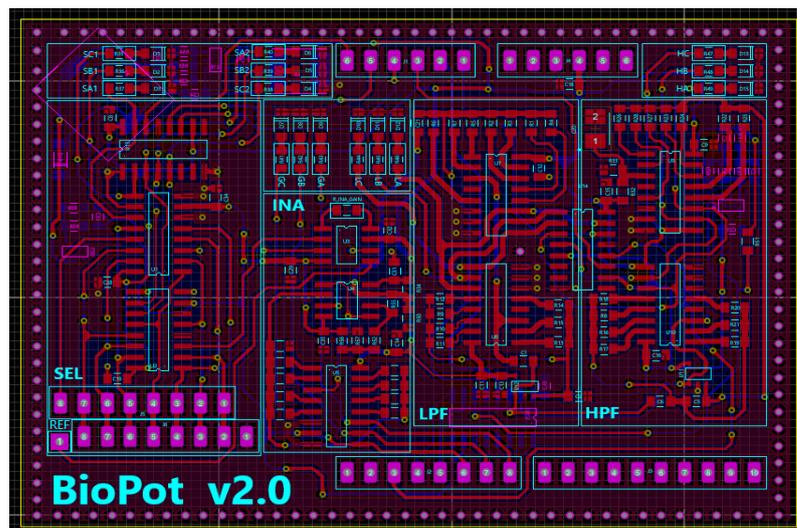


Figure 2.16 : The PCB of the whole electronic design.

2.3 Embedded Software Design

Arduino IDE is used to develop embedded software [24;25]. Since it is easy to use and open source software, it has very common use in programming Arduino boards. It is structured on two basic functions: setup() and loop(). The main window of the IDE is shown in Figure 2.17.

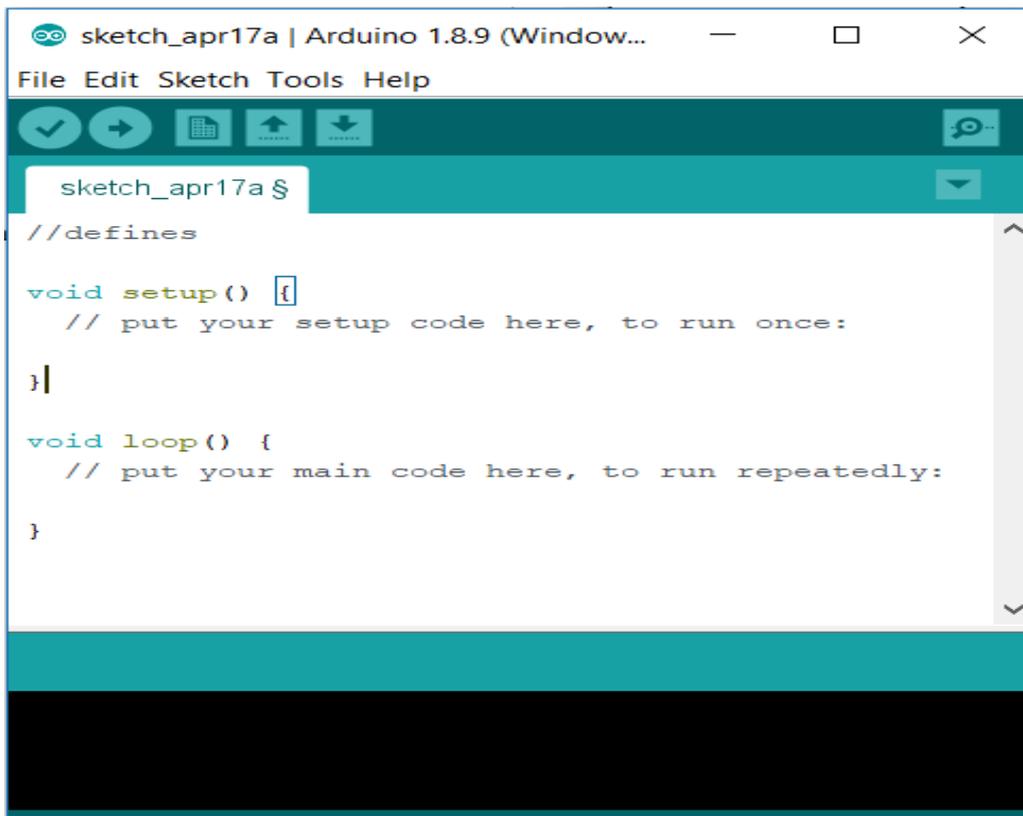


Figure 2.17 : Arduino IDE.

The setup() function runs only once at the beginning. That is the reason why it contains initializing the variables, defining the digital pins' directions, the baud rate of the serial port, etc. On the contrary, the loop() function runs until the board switched off from the power supply. It contains the main duty of the board [25;26]. The whole program developed for the Arduino Uno board can be found in Appendix A.

2.4 PC Software Design

Visual Studio 2017 Community Edition is used to develop a computer program. Since this IDE is freely available and easy to use, it is preferred in many studies in the literature. It has a powerful exception handling routines including try-catch block

[27]. The software is adjusted to read serial data from the Arduino board by using the timer routine. The serial port is read by every 10 ms. Although it can be adjusted to other values, we preferred this value after some trials. It has also a graphical user interface (GUI) design tool to make the developed programs user-friendly. The whole code for the PC program can be found in Appendix B.

2.4.1 Main window

The prepared GUI for the PC program is the main window of the program (Figure 2.18). All the settings can be adjusted and the output plot can be seen on this window at the same time.

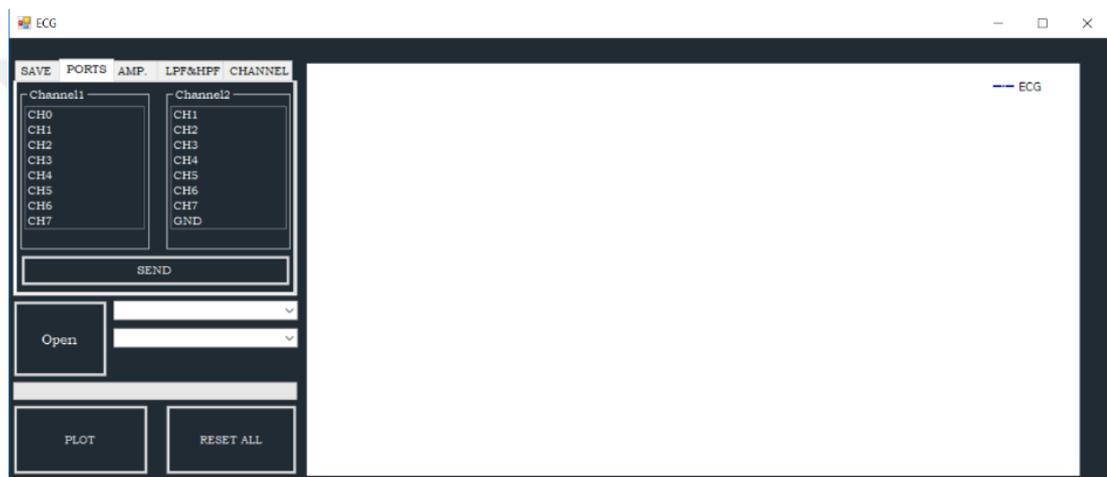


Figure 2.18 : The interface of the PC program.

The SAVE tab is used to save all settings to the hard disc. The RESET ALL button is used clear all settings and the plot screen. The PLOT button is used to start drawing the data on the plot area in real time.

2.4.2 Open button

The button is used to open serial port communication. There are two combo boxes next to it (Figure 2.19). The upper one is used to choose the serial port number. The other one is used to choose the baud rate. After these selections, the open button is clicked to initialize the serial communication with the Arduino board.



Figure 2.19 : The Open button.

2.4.3 Port tab

This tab is used to select input channels. There are eight channels (CH0 to CH7 and GND) for both, which means 64 combinations are available for inputs (Figure 2.20). Channel1 will be the positive terminal on the circuit and Channel2 will be the negative terminal on the circuit.

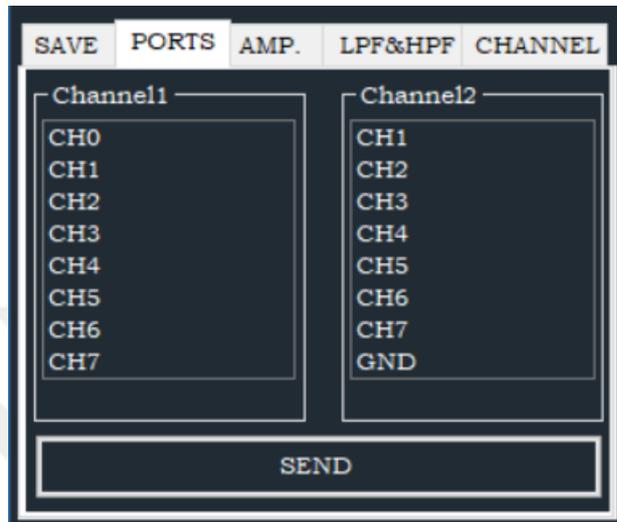


Figure 2.20 : The PORTS tab to determine which channel will be connected to the shield.

2.4.4 Amp tab

This tab is used to adjust the voltage gain. There are eight values are defined in the software for future studies, but the first five of amplification values are used during the thesis since upper values cause to exceed the upper limit of ADC inputs (Figure 2.21).

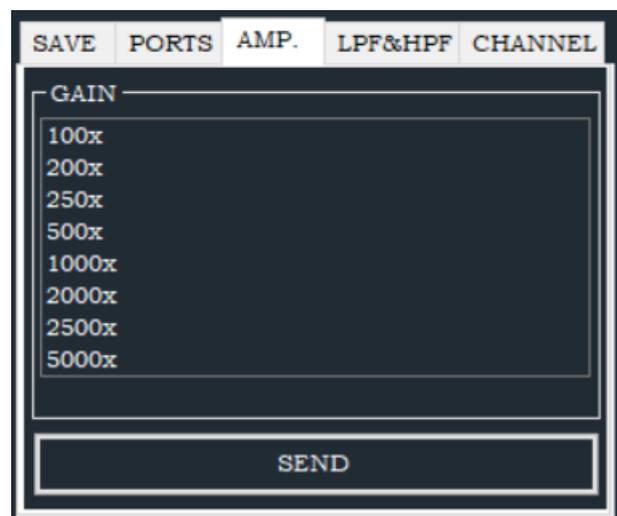


Figure 2.21 : The AMP tab to determine which amplification ratio will be applied to the ECG signal.

2.4.5 LPF&HPF tab

This tab is used to adjust the cut-off frequencies for both low-pass filter (LPF) and high-pass filter (HPF). Five values are available for LPF as 50 Hz, 100 Hz, 200 Hz, 500 Hz, 1 KHz, and 2 KHz. Similarly, five values are also available for HPF as 0.01 Hz, 0.05 Hz, 0.1 Hz, 0.5 Hz, 1 Hz, and 10 Hz (Figure 2.22). When the user selects these values, the PC program transmits these parameters to the Arduino board and the Arduino gives the required selection commands to the ICs on the proposed shield.

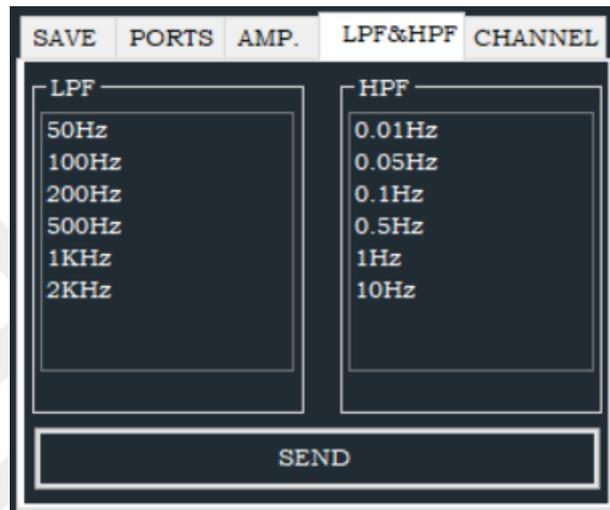


Figure 2.22 : The LPF&HPF tab to determine cut-off frequencies of both low-pass and high-pass filters.

2.4.6 Channel tab

This tab is used to choose which processed signal will be transferred to the actual output. Thus, this output is printed to the plot screen on the PC program via the serial port. There are four processed ECG signals as (1) the actual signal (amplified but not filtered), (2) amplified and low-pass filtered, (3) amplified, low-pass filtered, high-pass filtered, and (4) amplified, low-pass filtered, high-pass filtered, notch-filtered (Figure 2.23).

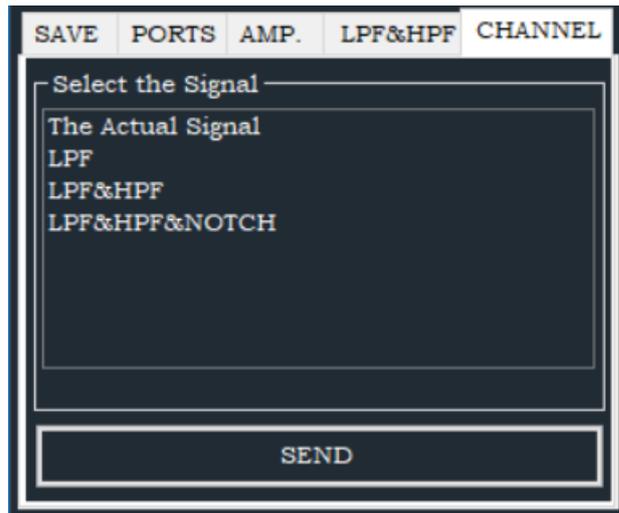


Figure 2.23 : The Channel tab to choose which ECG signal will be transferred to the actual output (or to the ADC).

2.4.7 The graphic plot area

This plot area is used to visualize the ECG signal coming from the Arduino board via the serial port. The signal plotting is also in real-time (Figure 2.24).

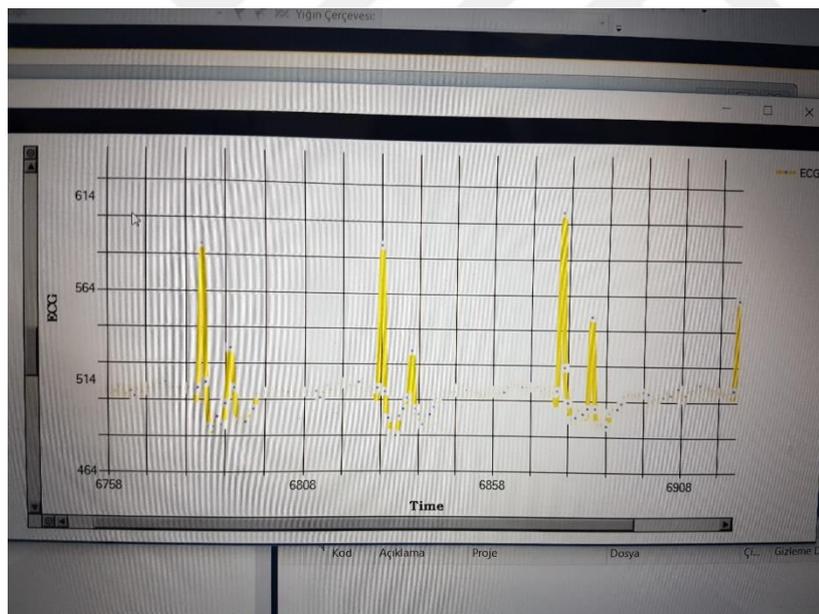


Figure 2.24 : The graphic output of the ECG signal on the computer program.

3. RESULTS

3.1 Input Selection

There are two inputs ICs and eight different input pins for each IC. The ECG device can accept 64 different input combinations to the input ICs (Figure 3.1).

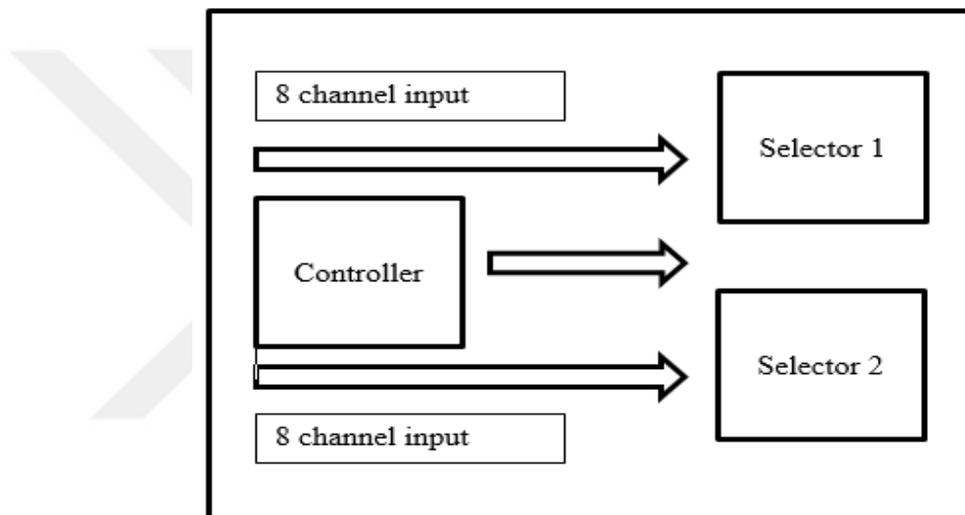


Figure 3.1 : Input combinations.

3.2 Gain Selection

After the input signal ports are determined, the signal is applied to the Gain Unit. The gain is can be selected one of six different gain options for the ECG signal (Figure 3.2).

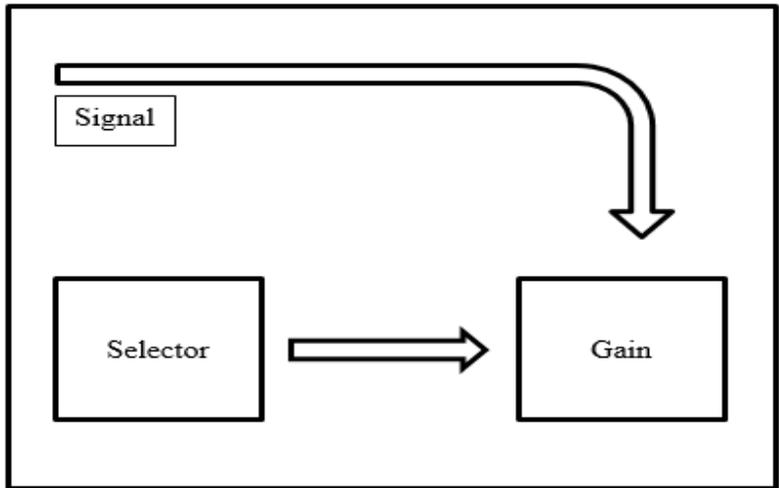


Figure 3.2 : Gain selection.

3.3 Low-Pass Filter Selection

The amplified ECG signal is applied to the Low-Pass Filter (LPF) Unit. The cut-off frequency of the LPF can be selected among the six different options that are very common in ECG and other biomedical signals (Figure 3.3).

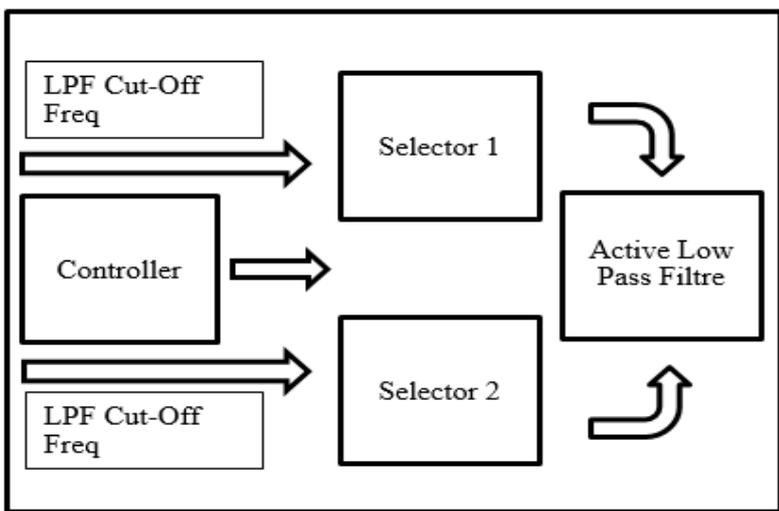


Figure 3.3 : Cut-off frequency selection for the LPF.

3.4 High-Pass Filter Selection

The amplified ECG signal is applied to the High-Pass Filter (HPF) Unit. The cut-off frequency of the HPF can be selected among the six different options that are very common in ECG and other biomedical signals (Figure 3.4).

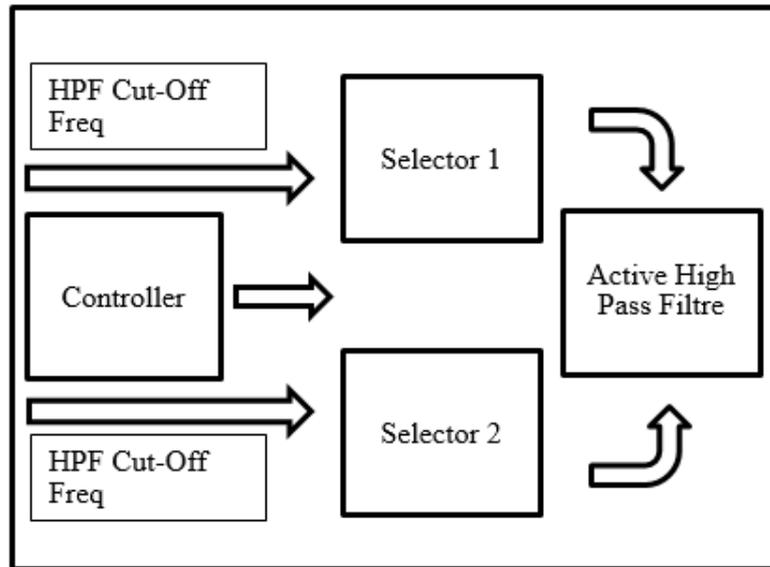


Figure 3.4 : Cut-off frequency selection for the HPF.

3.5 Output Selection

After the ECG passed all the stages of the proposed system in addition to the notch filter unit, the user is free to transfer one output of the four different ECG stages to the actual output of the system. There are four options: Amplified signal, Low-pass filtered signal, High-pass filtered signal, and Notch filtered signal (Figure 3.5).

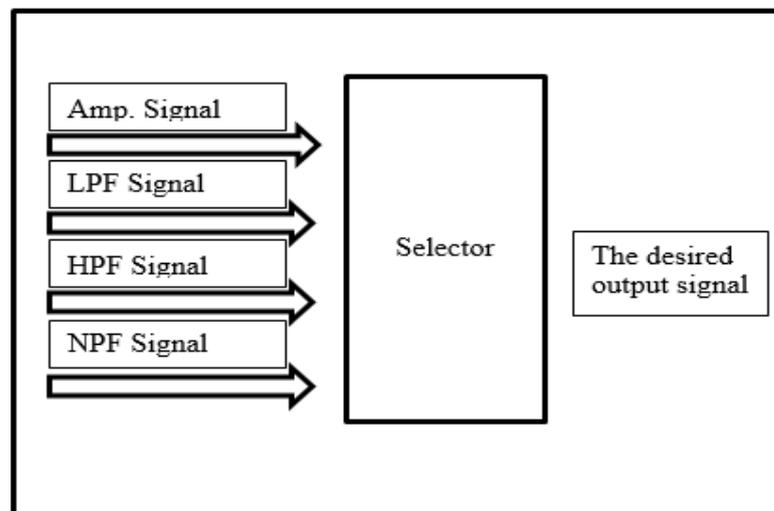


Figure 3.5 : Output selection.

3.6 Whole Proposed System

The whole system has five functional stages: Input selection, Gain selection, Low-pass filter, High-pass filter, Notch-filter, and Output selection. The user is free to

transfer one of these outputs to the actual output of the proposed board. The actual output of the SCG signal is connected to an ADC pin of the Arduino board. Arduino board converts the analog ECG signal to the digital ECG signal. By being transmitted to the serial port, this digital ECG signal can be viewed in the PC program of the proposed system in real-time (Figure 3.6).

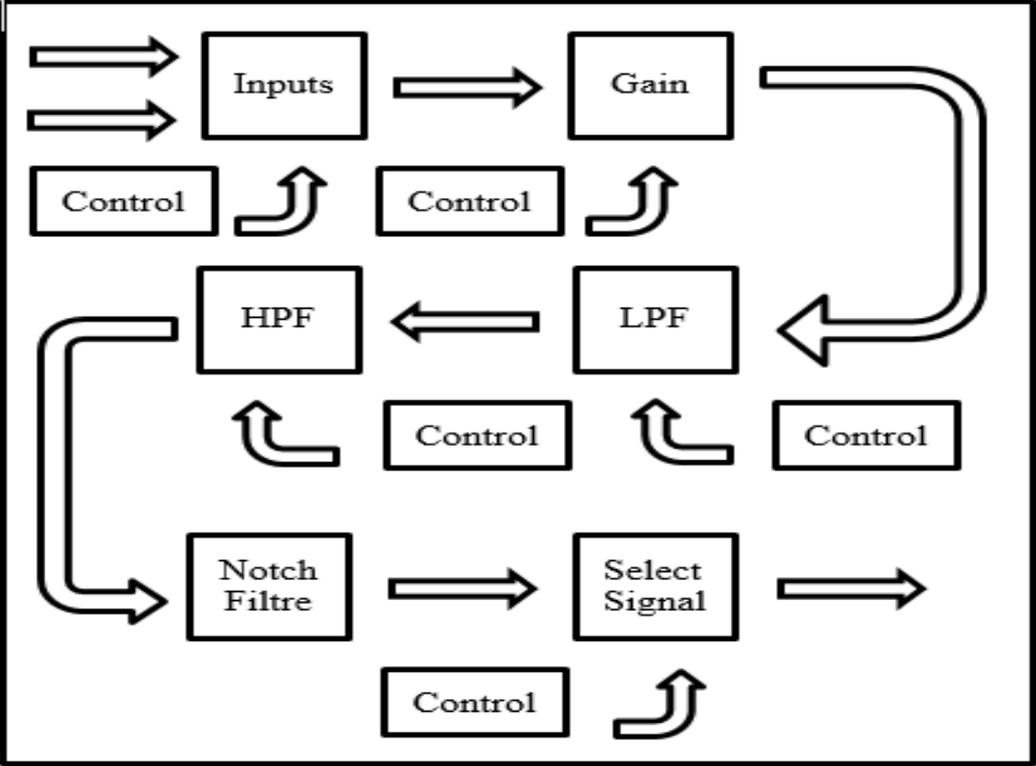


Figure 3.6 : The whole block diagram of the proposed system.

3.7 Printed Circuit Board of the Proposed System

The whole system was implemented on a printed circuit board (PCB) by using SMD devices (Figure 3.7).

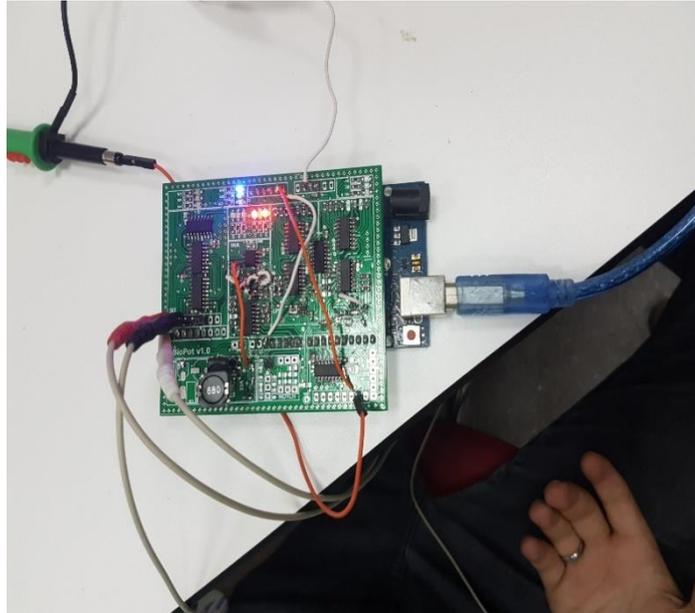


Figure 3.7 : The implemented system.

3.8 Acquiring the ECG Signal via the Proposed System

After the components are soldered to the prepared PCB, the proposed shield is ready to acquire an ECG signal. This section of the thesis shows some photographs from a real application. First, we connected to the proposed system to a volunteer by using Ag/AgCl ECG electrodes (Figure 3.8).

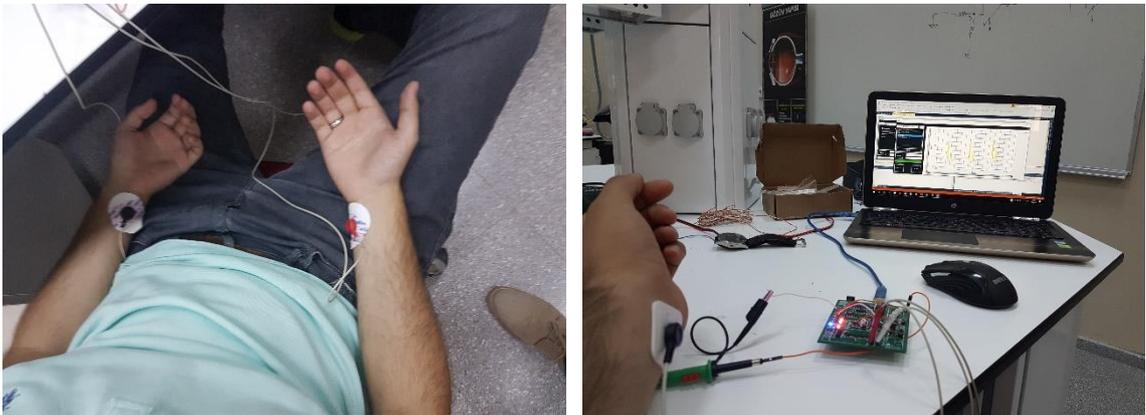


Figure 3.8 : A real application of the proposed system.

After connecting the proposed system's inputs to a volunteer, we choose to transfer amplified signal (before the LPF stage) to the actual output. We prefer to acquire this ECG signal to an oscilloscope input channel (Figure 3.9).

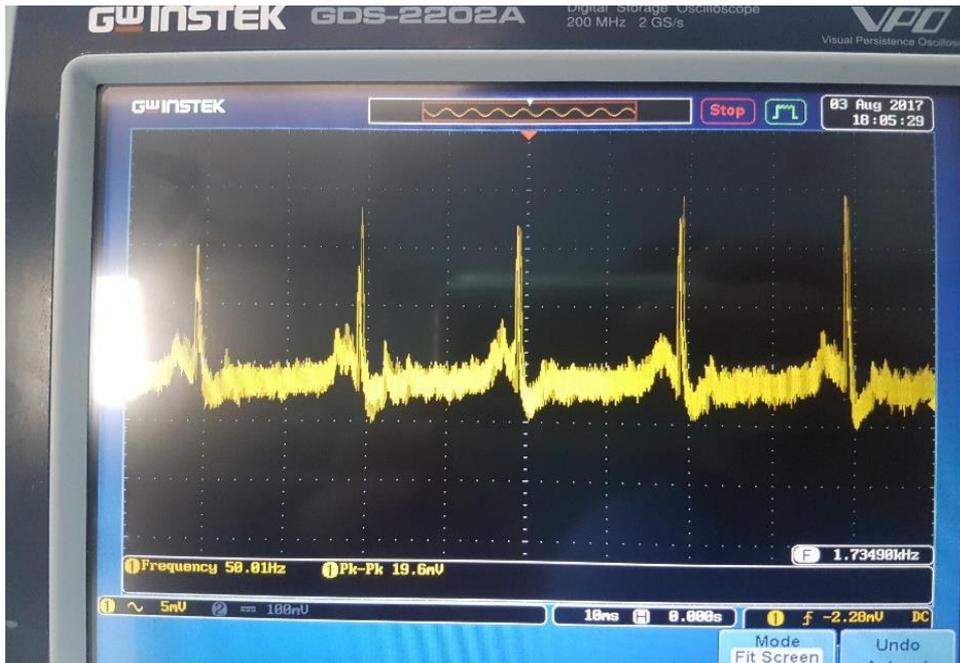


Figure 3.9 : The oscilloscope screen of an amplified but filtered ECG signal.

Next, we choose to transfer amplified, low-pass filtered, high-pass filtered, and notch-filtered ECG signal (after all stages) to the actual output. We prefer to acquire this ECG signal to an oscilloscope input channel again (Figure 3.10).

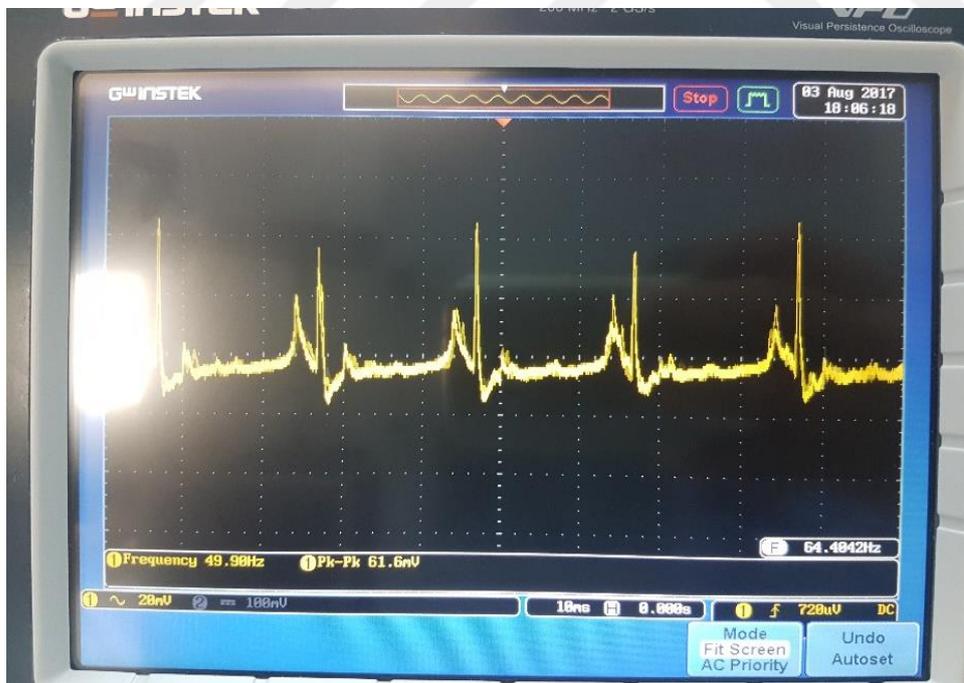


Figure 3.10 : The oscilloscope screen of an amplified and filtered ECG signal.

Finally, we choose to transfer amplified, low-pass filtered, high-pass filtered, and notch-filtered ECG signal (after all stages) to the actual output again. We prefer to acquire this ECG signal to the PC program (Figure 3.11).

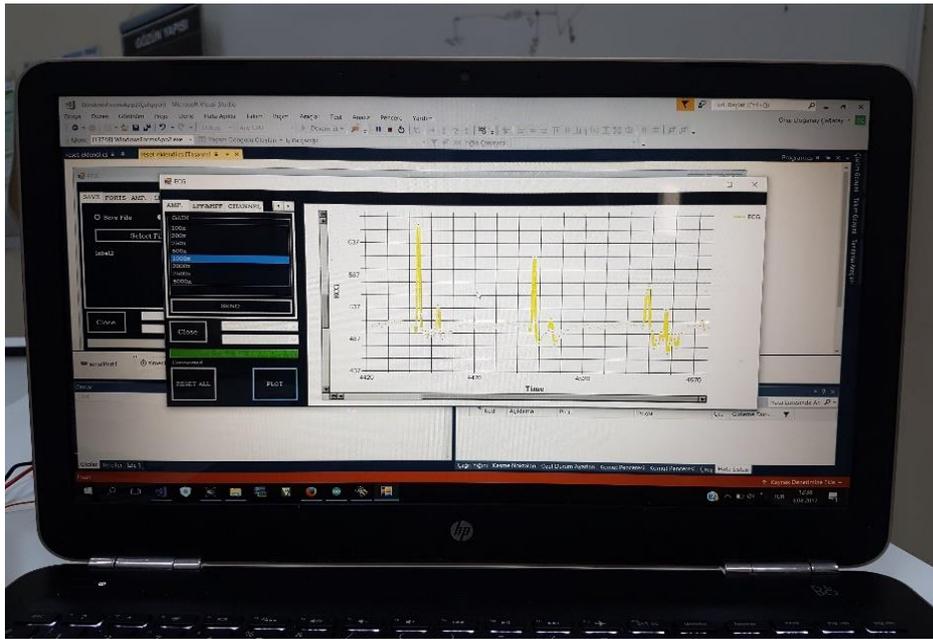


Figure 3.11 : The real-time acquisition of an ECG signal via the computer program.

4. DISCUSSION

Single-use electrodes were used to detect the electrical activity of the heart from the body skin. In this study, ECG signals were recorded where their amplitude is less than 1mV. This signal is amplified where the common integrated circuits can easily handle it. Noises were eliminated from the ECG signal by using filters. The whole proposed circuit was implemented on a two-sided PCB with surface-mount devices. The final design is completely compatible to an Arduino board and it can connect to the Arduino via the extension slot.

The components used in this thesis were selected among low-noisy, easy-to-operate, and easily available electronic components from the market. Resulting ECG signal from the human subjects proves that the proposed system gives the same ECG graph by comparing to commercial ECG devices.

As a result, since this system is designed as an Arduino extension board (shield), it can easily be adapted to other biological signals. In addition, it can be used in student experiments related to biomedical instrumentation, biomedical signal processing, microcontrollers, medical device design, graduation projects, etc. in the research laboratories.

REFERENCES

- [1] Schweitzer, P., & Keller, S. (2002). Willem Einthoven-inventor of electrocardiography. *Vnitřní Lekarství*, 48 (Suppl 1), 20-23.
- [2] Baltazar, R. F. (2009). Basic and bedside electrocardiography. 1st ed. Philadelphia, PA.
- [3] Rivera-Ruiz, M., Cajavilca, C., & Varon, J. (2008). Eithoven's string galvanometer-the first electrocardiograph. *Texas Heart Institute Journal*, 35(2), 174-178.
- [4] Goldberger, E. (1942). A simple, indifferent, electrocardiographic electrode of zero potential and a technique of obtaining augmented, unipolar, extremity leads. *American Heart Journal*, 23(4), 483-492.
- [5] Riseman, J. E. F. (1952). P-Q-R-S-T. *Journal of Medical Education*, 27(6), 420.
- [6] Rautaharju, P. M. (1987). A hundred years of progress in electrocardiography 1: Early contributions from Waller to Wilson. *Canadian Journal of Cardiology*, 3, 362-374.
- [7] www.labcenter.com/tutorials/
- [8] www.ele.uva.es/~jesman/BigSeti/ftp/Cajon_Desastre/Software-Manuales/EBook%20-%20Proteus%20Manual.pdf
- [9] <http://www.ti.com/lit/ds/symlink/lm2611.pdf>
- [10] https://assets.nexperia.com/documents/data-sheet/74HC_HCT4051.pdf
- [11] <http://www.ti.com/lit/ds/symlink/ina128.pdf>
- [12] https://www.analog.com/media/en/training-seminars/design-handbooks/Op-Amp_Applications/Section1.pdf
- [13] <https://www.khanacademy.org/science/electrical-engineering/ee-circuit-analysis-topic/ee-resistor-circuits/a/ee-voltage-divider>
- [14] <http://sim.okawa-denshi.jp/en/OPseikiLowkeisan.htm>
- [15] <https://www.electronics-tutorials.ws/filter/second-order-filters.html>
- [16] <http://www.ti.com/lit/an/sloa024b/sloa024b.pdf>
- [17] <http://sim.okawa-denshi.jp/en/Fkeisan.htm>
- [18] <https://www.sciencedirect.com/topics/engineering/sallen-key>
- [19] <https://pdfs.semanticscholar.org/98f1/e5f1759a10e74e8bd1a56c4dce5eb4e3f234.pdf>
- [20] <http://sim.okawa-denshi.jp/en/TwinTCRkeisan.htm>
- [21] http://www.komponenten.es.aau.dk/fileadmin/komponenten/Data_Sheet/MOS-TTL/hc/74HC4052.pdf
- [22] Rao, J. V., & Praroopa, Y. (2011). Logic Circuits and Gates in Pre A*-Algebra. *Asian Journal of Applied Sciences*, 4(1), 89-96.
- [23] <http://kisi.deu.edu.tr/ozlem.karaca/D.pdf>
- [24] <https://www.arduino.cc/>
- [25] Studie zur Bedeutung des Sektors Embedded-Systeme in Deutschland (Embedded Systems Study in Germany). (2008).

www.bitkom.org/files/documents/Studie_BITKOM_Embedded_Systeme_11_11_2008.pdf.

- [26] Berger, A. S. (2001). *Embedded systems design: An introduction to processes tools and techniques*: CMP Books.
- [27] Hejlsberg, A., Torgersen, M., Wiltamuth, S., & Golde, P. (2010). *The C# programming language*: 4th edition.
- [28] Scott Wiltamuth, Peter Golde & Anders Hejlsberg. (2003). *C# Language Specification*.



APPENDIX A: The Arduino Program

```
//The initial function makes all variables to default values.
void initial(){
char init[10] = "1111111111";
for(int a=0; a < 10; a++) {
SEL_DATA_WRITE((init[a]-48), HIGH);
LPF_HPF_DATA_WRITE((init[a]-48), HIGH);
}
digitalWrite(GAINA, LOW);
delay(10);
digitalWrite(GAINB, LOW);
delay(10);
digitalWrite(GAINC, LOW);
delay(10);
digitalWrite(SEL_SIG_A, LOW);
delay(10);
digitalWrite(SEL_SIG_B, LOW);
delay(10);
}
//The printSerialString function is used to communicate with the PCB.
void printSerialString() {
if( serInIndx > 0) {
if(serInString[0] == '0' ) {
for(serOutIndx=0; serOutIndx < serInIndx; serOutIndx++) {
LPF_HPF_DATA_WRITE((serInString[serOutIndx]-48), HIGH);
}
}else if(serInString[0] == '1' ){
for(serOutIndx=0; serOutIndx < serInIndx; serOutIndx++) {
SEL_DATA_WRITE((serInString[serOutIndx]-48), HIGH);
}
}else if(serInString[0] == '2' ){
digitalWrite(GAINA, LOW);
delay(10);
digitalWrite(GAINB, LOW);
delay(10);
digitalWrite(GAINC, LOW);
}else if(serInString[0] == '3' ){
digitalWrite(GAINA, HIGH);
delay(10);
digitalWrite(GAINB, LOW);
delay(10);
digitalWrite(GAINC, LOW);
}else if(serInString[0] == '4' ){
```

```

digitalWrite(GAINA, LOW);
delay(10);
digitalWrite(GAINB, HIGH);
delay(10);
digitalWrite(GAINC, LOW);
}else if(serInString[0] == '5'){
digitalWrite(GAINA, HIGH);
delay(10);
digitalWrite(GAINB, HIGH);
delay(10);
digitalWrite(GAINC, LOW);
}else if(serInString[0] == '6'){
digitalWrite(GAINA, LOW);
delay(10);
digitalWrite(GAINB, LOW);
delay(10);
digitalWrite(GAINC, HIGH);
}else if(serInString[0] == '7'){
digitalWrite(GAINA, HIGH);
delay(10);
digitalWrite(GAINB, LOW);
delay(10);
digitalWrite(GAINC, HIGH);
}else if(serInString[0] == '8'){
digitalWrite(GAINA, LOW);
delay(10);
digitalWrite(GAINB, HIGH);
delay(10);
digitalWrite(GAINC, HIGH);
}else if(serInString[0] == '9'){
digitalWrite(GAINA, HIGH);
delay(10);
digitalWrite(GAINB, HIGH);
delay(10);
digitalWrite(GAINC, HIGH);
}else if(serInString[0] == 'a'){
digitalWrite(SEL_SIG_A, LOW);
delay(10);
digitalWrite(SEL_SIG_B, LOW);
delay(10);
}else if(serInString[0] == 'b'){
digitalWrite(SEL_SIG_A, HIGH);
delay(10);
digitalWrite(SEL_SIG_B, LOW);
delay(10);
}else if(serInString[0] == 'c'){
digitalWrite(SEL_SIG_A, LOW);
delay(10);
digitalWrite(SEL_SIG_B, HIGH);
delay(10);

```

```

}else if(serInString[0] == 'd' ){
digitalWrite(SEL_SIG_A, HIGH);
delay(10);
digitalWrite(SEL_SIG_B, HIGH);
delay(10);
}else if(serInString[0] == 'r' ){
Serial.println("reset");
Serial.println(serInString[0]);
initial();
delay(10);
}
serOutIndx = 0;
serInIndx = 0;
}
}
//If the first bit of serInString[] variable is zero, the system understands the incoming
data is about to cut-off frequencies of low-pass and high-pass filters. The if loop is
given below.
if(serInString[0] == '0' ) {
for(serOutIndx=0; serOutIndx < serInIndx; serOutIndx++) {
LPF_HPF_DATA_WRITE((serInString[serOutIndx]-48), HIGH);
}
}
//If the first bit of serInString[] variable is '1', the incoming data is about adjusting
gain of ECG signal.
else if(serInString[0] == '1' ){
for(serOutIndx=0; serOutIndx < serInIndx; serOutIndx++) {
SEL_DATA_WRITE((serInString[serOutIndx]-48), HIGH);
}
}
//If the first bit of serInString[] variable is '2', the gain of ECG signal are 100 times.
else if(serInString[0] == '2' ){
digitalWrite(GAINA, LOW);
delay(10);
digitalWrite(GAINB, LOW);
delay(10);
digitalWrite(GAINC, LOW);
}
//If the first bit of serInString[] variable is '3', the gain of ECG signal are 200 times.
else if(serInString[0] == '3' ){
digitalWrite(GAINA, HIGH);
delay(10);
digitalWrite(GAINB, LOW);
delay(10);
digitalWrite(GAINC, LOW);
}
//If the first bit of serInString[] variable is '4', the gain of ECG signal are 250 times.
else if(serInString[0] == '4' ){
digitalWrite(GAINA, LOW);
delay(10);

```

```

digitalWrite(GAINB, HIGH);
delay(10);
digitalWrite(GAINC, LOW);
}
//If the first bit of serInString[] variable is '5', the gain of ECG signal are 500 times.
else if(serInString[0] == '5'){
digitalWrite(GAINA, HIGH);
delay(10);
digitalWrite(GAINB, HIGH);
delay(10);
digitalWrite(GAINC, LOW);
}
//If the first bit of serInString[] variable is '6', the gain of ECG signal are 1000 times.
else if(serInString[0] == '6'){
digitalWrite(GAINA, LOW);
delay(10);
digitalWrite(GAINB, LOW);
delay(10);
digitalWrite(GAINC, HIGH);
}
//If the first bit of serInString[] variable is '7', the gain of ECG signal are 2000 times.
else if(serInString[0] == '7'){
digitalWrite(GAINA, HIGH);
delay(10);
digitalWrite(GAINB, LOW);
delay(10);
digitalWrite(GAINC, HIGH);
}
//If the first bit of serInString[] variable is '8', the gain of ECG signal are 2500 times.
else if(serInString[0] == '8'){
digitalWrite(GAINA, LOW);
delay(10);
digitalWrite(GAINB, HIGH);
delay(10);
digitalWrite(GAINC, HIGH);
}
//If the first bit of serInString[] variable is '9', the gain of ECG signal are 5000 times
else if(serInString[0] == '9'){
digitalWrite(GAINA, HIGH);
delay(10);
digitalWrite(GAINB, HIGH);
delay(10);
digitalWrite(GAINC, HIGH);
}
//If the first bit of serInString[] variable is 'a', the amlitued signal is going to go to
the ADC pin of Arduino Uno Board.
else if(serInString[0] == 'a'){
digitalWrite(SEL_SIG_A, LOW);
delay(10);
digitalWrite(SEL_SIG_B, LOW);
}

```

```

delay(10);
}
//If the first bit of serInString[] variable is 'b', the low-pass filtered signal is going to
go to the ADC pin of Arduino Uno Board.
else if(serInString[0] == 'b' ){
digitalWrite(SEL_SIG_A, HIGH);
delay(10);
digitalWrite(SEL_SIG_B, LOW);
delay(10);
}
//If the first bit of serInString[] variable is 'c', the high-pass filtered signal is going to
go to the ADC pin of Arduino Uno Board.
else if(serInString[0] == 'c' ){
digitalWrite(SEL_SIG_A, LOW);
delay(10);
digitalWrite(SEL_SIG_B, HIGH);
delay(10);
}
//If the first bit of serInString[] variable is 'd', the notch filtered signal is going to go
to the ADC pin of Arduino Uno Board.
else if(serInString[0] == 'd' ){
digitalWrite(SEL_SIG_A, HIGH);
delay(10);
digitalWrite(SEL_SIG_B, HIGH);
delay(10);
}
//And if the first bit of serInString[] variable is 'r', all system is reset.
else if(serInString[0] == 'r' ){
Serial.println("reset");
Serial.println(serInString[0]);
initial();
delay(10);
}
}

```

APPENDIX B: The PLOT Function of the Computer Program

The plot_Click function for PC software is plot the ECG signal in real time in the screen. The function is given below.

```
private void plot_Click(object sender, EventArgs e){
chart1.ChartAreas[0].CursorY.IsUserEnabled = true;chart1.Visible = true;
i++;
if (i > 1) i = 0;
if (i == 0) timer1.Enabled = false;
else if (i == 1) timer1.Enabled = true;
chart1.Series[0].ChartType =
System.Windows.Forms.DataVisualization.Charting.SeriesChartType.Spline;
chart1.BackColor = System.Drawing.Color.Black;
chart1.ChartAreas[0].CursorX.IsUserEnabled = true;
chart1.ChartAreas[0].CursorX.IsUserSelectionEnabled = true;
chart1.ChartAreas[0].AxisX.ScaleView.Zoomable = true;
chart1.ChartAreas[0].CursorY.IsUserSelectionEnabled = true;
chart1.ChartAreas[0].AxisY.ScaleView.Zoomable = true;
chart1.ChartAreas[0].AxisX.ScrollBar.Size = 15;
chart1.ChartAreas[0].AxisX.ScrollBar.IsPositionedInside = false;
chart1.ChartAreas[0].AxisX.ScrollBar.Enabled = true;
chart1.ChartAreas[0].AxisY.ScrollBar.Size = 15;
chart1.ChartAreas[0].AxisY.ScrollBar.IsPositionedInside = false;
chart1.ChartAreas[0].AxisY.ScrollBar.Enabled = true;
chart1.ChartAreas[0].AxisX.MajorTickMark.Interval = 10;
chart1.ChartAreas[0].AxisY.MajorTickMark.Interval = 10;
chart1.ChartAreas[0].AxisX.MajorGrid.Interval = 2;
chart1.ChartAreas[0].AxisY.MajorGrid.Interval = 0.5;
chart1.ChartAreas[0].AxisX.LineColor = System.Drawing.Color.Green;
chart1.ChartAreas[0].AxisX.LineWidth = 2;
chart1.ChartAreas[0].AxisX.MajorGrid.LineColor = System.Drawing.Color.Green;
chart1.ChartAreas[0].AxisY.LineColor = System.Drawing.Color.Green;
chart1.ChartAreas[0].AxisY.LineWidth = 2;
chart1.ChartAreas[0].AxisY.MajorGrid.LineColor = System.Drawing.Color.Green;
}
```

APPENDIX C: The Pin Description of the 74HC4052

PIN	SYMBOL	DESCRIPTION
1	2Y0	Independent input or output
2	2Y2	Independent input or output
3	2Z	Common input or output
4	2Y3	Independent input or output
5	2Y1	Independent input or output
6	E	Enable input (active LOW)
7	V_{EE}	Negative supply voltage
8	GND	Ground (0V)
9	S1	Select logic input
10	S0	Select logic input
11	1Y3	Independent input or output
12	1Y0	Independent input or output
13	1Z	Common input or output
14	1Y1	Independent input or output
15	1Y2	Independent input or output
16	V_{CC}	Positive supply voltage

CURRICULUM VITAE

Name Surname: Onur Dođanay AĐATAY

Place and Date of Birth: İzmir / 04.02.1992

Address: 222/41 Sokak No 28 Daire:1 Buca/İZMİR

E-Mail: onurdoganaycagatay@hotmail.com

Education:

- 2014, Mersin University, Faculty of Engineering, Department of Electrical and Electronics Engineering
- 2013, University of Gavle SWEDEN, Faculty of Engineering, Department of Electrical and Electronics Engineering

List of Publications:

- Y. İşler, O. D. ađatay, [Elektrokardiyografi Sinyallerinin Gerek Zamanlı Kaydı İin Arduinio Eklentisi Tasarımı \(Design of Arduino Shield to Record Real-Time Electrocardiography Signals\)](#), 6th International Symposium on Innovative Technologies in Engineering Science (ISITES-2018), 413-418, Alanya / Antalya / Turkey, 9-11 November 2018.