People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research

University M'Hamed BOUGARA – Boumerdes

Institute of Electrical and Electronic Engineering

Department of Electronics

Final Year Project Report Presented in Partial Fulfilment of the Requirements for the Degree of

'Master'

In Electrical and Electronic Engineering

Option: **Computer Engineering**

Title:

Implementation of a Biometric Identification System using Electrocardiogram ECG/EKG

Presented By:

Badr Eddine ZEBBICHE

Supervisors:

Dr. Dalila CHERIFI, IGEE

Mr. Farid HARIZI, CDTA

Registration Number:...../2015

Acknowledgment

I have taken humble efforts in this graduation project and it would not have been possible without the kind support and help of many individuals and we would like to extend all our sincere thanks to all of them.

I am highly indebted to ***Mr Farid Harizi*** *from CDTA Research Center for proposing the project, for his guidance and constant supervision as well as to* ***Dr Dalila Cherifi*** *for providing necessary advices and support regarding the project & also for her encouragements in completing the project. Without her recommendation to HackNYAD, I could not make it to the top1 team Hackee.me in New York University Abu Dhabi.*

I would like to express my gratitude towards my parents primarily & family for their kind co-operation and encouragement which helped in completion of this project.

I would like to express my special gratitude and thanks to all our professors and instructors all over the years spent in IGEE EX. INELEC.

My thanks and appreciations also go to all our colleagues in the institute (especially my roommates: Sofiane Tazekrit and Sadek Ramzi Kraouchi) and all the people surrounding me whom they willingly helped me out with their abilities (Rahman Ali from Georgia Tech in Atlanta, Mm. Amel AmirAli for bringing some electronic components from UK, Mr Yacine Bourras for the graphical design, and all volunteers to construct ECG database…etc.).

Abstract

Nowadays, Biometrics is extensively being used for the purpose of authentication in security related aspects. Biometrics deals with individuals' identification through their physiological characteristics such as fingerprint, Deoxyribonucleic Acid DNA, Electrocardiogram ECG, Face, Voice…etc. Many of these models have limitations and contains: difficultly of extraction (DNA), unique utilization of the hardware (IRIS)…etc. Hence, ECG is chosen for its accuracy, hardware utilization in tele-monitoring, and high security level.

In this presented treatise, an ECG based biometric system has been developed. The project is divided into two main parts: Hardware and Software. The hardware part, performs the signal acquisition, amplification, and digitalization. The hardware is provided with the alternative of wired via UART or wireless transmission via XBee RF modules. In software part, the biometric algorithm uses the concepts of machine learning and pattern recognition. Basically, the algorithm is divided into two main part: training (enrollment) phase and testing phase. In training phase, the system gets indoctrinated with a set of ECG data recordings of different people. Then, it extracts their features after pre-processing in a form of spectral information. Features are taken into higher dimensionality space of 2^8 (256D) and scattered separately to form labeled classes (Principal Components Analysis – PCA). Once the testing data arrives, classification process affects them into their corresponding class, which is identification decision, using the Euclidean distance.

The algorithm was developed based on the MIT-BIH Arrhythmia Database and then tested on our customized database acquired using the developed hardware. The two main aspects that have been under focus are the execution time and accuracy or identification rate. An accuracy of 98% has been achieved with the developed system.

Keywords:

Electrocardiogram, ECG/EKG, Biometrics, Data Acquisition, Signal Processing, Wireless, Machine Learning, Pattern Recognition, Principal Components Analysis PCA.

ملخّص مختصر

في عصرنا الحاليّ، تُستعمل التطبيقات البيومتريّة بكثرة لأهداف متعلّقة بأمور أمنيّة. تقوم التطبيقات البيومتريّة بتحديد هويّة الأشخاص من خلال خصائصهم الفيزيولوجيّة كالبصمة، الحمض النّووي، الإشارة الكهربائيّة للقلب أو عن طريق صفاتهم السلوكيّة كالصّوت مثلاً. بعض هذه الصفات أو الخصائص يمكن نسخها أو تقليدها كالبصمة، أو يصعب استخلاصها كالحمض النووي، أو حتّى معدّاتها غير متوفرة بكثرة و باهظة الثمن (كتجهيزات القزحيّة و شبكيّة العين). تمّ اختيار مخطّط الإشارة الكهربائيّة للقلب (أو تخطيط كهربيّة القلب حسب تسمية ويكيبيديا) لعدّة أسباب أهمّها: الاستعمالات الإضافيّة للتجهيزات الخاصة به في مجال الطبّ عن بعد (كالمراقبة عن بعد)، تخطيط القلب لديه درجة عالية من الدقة كما يعكس حالة الفرد الحيويّة (عكس البصمة...)، و علاوة على ذلك إمكانيّة توسيع المشروع في المستقبل ليشمل التطبيقات البيوطبيّة كالكشف عن حالات المرض القلبيّة أو الحالات الشّاذّة.

في مذكّرة التخرج هذه، تمّ تطوير نظام بيومتري للتعرّف على الأشخاص من خلال النشاط الكهربائيّ للقلب الخاصّ بكلّ شخص. يشمل المشروع العتاد/التجهيز و كذلك البرمجيّة المتمثلة في الخوارزميّة البيومتريّة. في التجهيز الإلكترونيّ، يقوم الجهاز المصمّم او النموذج المخصّص بالتقاط الإشارات الكهربائيّة من الجسم و تضخيمها ثمّ رقمنتها على شكل إشارات رقميّة و ذلك من أجل ارسالها إلى وحدة حوسبة أو معالجة رقميّة سواء سلكيّاً عبر المنفذ التسلسلي COM باستخدام الـ USB أو لاسلكيّا باستخدام وحدات راديو XBee ذات بروتوكول زيجبي.

في الجزء المتعلّق بالبرمجيّة، تستعمل الخوارزميّة أساسّيات "التعلّم الآلي" في أنظمة الذّكاء الاصطناعيّ و التعرف على الأنماط. لهذا الغرض، تمّ استعمال مصنّف من نوع "محلّل المكوّنات الأساسيّة" لتصنيف الإشارات القلبيّة للفئات المقابلة لها (بمعنى الأشخاص). في الأساس، تنقسم الخوارزمية إلى قسمين رئيسيّين: مرحلة التدريب (الانتساب) ومرحلة الاختبار. في مرحلة التدريب، يتمّ تلقين النظام مع مجموعة من تسجيلات البيانات الرقميّة لمخطط الإشارة الكهربائيّة للقلب من أشخاص مختلفين. تقوم الخوازميّة باستخلاص الخواصّ و الميزات على شكل معلومات طيفيّة في مجال التردّدات بعد عمل معالجة رقميّة أوّليّة. تؤخذ الميزات إلى فضاء ذو بعد عالي 2^8 (البعد 256) ثمّ تنشر بطريقة متباينة و متباعدة لتمكين المصنّف (PCA) في التمايز و التصنيف. عند قدوم عيّنات الإختبار، تتمّ عملية التصنيف لتحديد هويّة الشخص و أخذ قرار نهائيّ على أساس معايير احصائيّة.

وقد تم تطوير الخوارزمية على أساس قاعدة بيانات MIT-BIH ذات المصداقية ومن ثم اختبارها على قاعدة بيانات مخصصة تمّ الحصول عليها باستخدام الجهاز الذّي تمّ وصفه سابقاً تمّ تجميعها من طرف طلبة مختلفين. تمثّل دقّة الخوارزميّة و الوقت المنقضي لأدائها أهمّ الجوانب المغطّات وقد حققت دقة تحديد الهويات بنسبة 98٪ باستعمال النّظام المُطوَّر.

كلمات مفتاحيّة:

مخطّط الإشارة الكهربائيّة للقلب، تخطيط كهربيّة القلب، تطبيقات بيومتريّة، جهاز تجهيز إلكترونيّ، برمجيّة، خوارزميّة، التقاط الإشارات الرّقميّة، تحليل الإشارات الرقميّة، اتصالات لاسلكيّة، التعلّم الآلي، ذكاء اصطناعيّ، التعرف على الأنماط، نظريّة التصنيف، محلّل المكوّنات الأساسيّة.

Outlines

Acknowledgment i
Abstract ii
Outlines iv
List of Figures vii
List of Tables viii
List of Acronyms ix

General Introduction

Introduction 1
Chapters' Overview/Highlights 2

Chapter I: Biometrics & Electrocardiogram ECG/EKG

I.1. Introduction to Biometrics 3
I.1. Biometrics definition and applications 3
I.2. Biometric Models 3
I.2. Electrocardiogram generalities 5
I.2.1. Cardiovascular system 5
I.2.2. Electrical Activity of the heart 6
I.3. Basics of ECG 7
I.3.1. ECG/EKG Waves and Complexes 8
I.3.2. ECG Derivations/Leads 9
I.3.2.1. Limb Leads 9
I.3.2.2. Augmented limb leads 10
I.3.2.3. Precordial leads 11
I.3.3. Measurements System 11
I.4. ECG Interpretation 12
I.4.1. Features Extraction 12
I.4.2. Interval Analysis 12
I.5. ECG Applications 13
I.5.1. Biomedical (Detection of Cardio anomalies) 13
I.5.2. Biometric (Identification Systems) 14
I.6. Reasons for choosing ECG 14

Chapter II: ECG/EKG Processing for Biometrics

II.1. Process Overview 15
II.2. Signal Pre-processing 16
II.2.1. Noises and Artifacts on ECG 16
II.2.2. Digital Filtering Generalities 17
II.2.2.1. IIR and FIR Filters 17
II.2.2.2. Case used filters (Butterworth, Median) 18
II.3. Features Extraction 19
II.3.1. Peaks Detection 19
II.3.3. Framing/Windowing 19
II.3.4. Spectral Information 20
II.4. Classification of ECG Signals 21
II.4.1. Introduction to ECG Classification 21
II.4.2. Principal Component Analysis PCA 22
II.4.2.1. PCA Theory 22
II.4.2.2 Steps of Implementation 23
II.4.3. Proper Space Formation 23
II.4.4. Training and Testing Phases 24
II.4.5. Classification/Decision 24

Chapter III: ECG/EKG Acquisition Hardware Implementation

III.1. Introduction 25
III.2. Sensors 26
III.2.1. Electrodes and Patches 26
III.2.2. Einthoven's Triangle 26
III.3 ECG/EKG Shield 27
III.3.1. Overview 27
III.3.2. Functionality (Schematics explanation/Description) 28
III.3.3. Shield Setting-up 29
III.4. Arduino 30
III.4.1. Introduction 30
III.4.2. "Duemilanove" Board | Characteristics 30
III.4.3. Programming 31
III.5. X-Bee/ZigBee Protocol 31
III.5.1. Overview 31
III.5.2. RF Communication 31

III.5.3. ZigBee Protocol 32
III.5.4. X-Bee Configuration 32

III.6. ECG Data Acquisition 33
III.6.1. Hardware Programming (flowchart) 33
III.6.2. Serial communication (UART, X-Bee) 35
III.6.2.1. Basics of serial communication 35
III.6.2.2. USB & X-Bee Data Transfer 35
III.6.3. ECG Reading 36
III.6.3.1. Packets reconstruction 36
III.6.3.2. Monitoring Alternatives 37
III.6.3.2.1. DAQ on Matlab 37
III.6.3.2.2. LabVIEW Application 40
III.6.3.3. ECG Saving for Database 40

Chapter IV: Identification Realization and Results

IV.1. Methodology and Application Overview 42
IV.2. ECG Pre-Processing 44
IV.3. Features Extraction implementation 45
IV.3.1. Peaks Detection 45
IV.3.2. Framing 46
IV.3.3. Spectral Information 48
IV.4. Proper Space Creation 49
IV.5. Training Phase 49
IV.5.1. Classes Labeling 49
IV.5.2. Features Projection 50
IV.6. Testing Phase/Identification 50
IV.6.1. Test Data into Proper Space 50
IV.6.2. Dimensionality Reduction 51
IV.6.3. Classification/Matching Processing 52
IV.6.3.1. Classification using Euclidean Distance 52
IV.6.3.2. Demonstration 53
IV.7. Results and Discussions 54

Conclusion x

References xi
Appendices xii

List of Figures

	Figure Name	Page N°
Figure I.1:	Anatomy of the heart, which consists of four chambers: the right and left atrium and the right and left ventricle.	5
Figure I.2:	Isolated conduction system of the heart.	6
Figure I.3:	A sequence animation of an ordinary ECG waves with their respective cardio blood pumping.	7
Figure I.4:	Intervals, complexes and segments of a normal heartbeat.	8
Figure I.5:	Einthoven triangle electrodes placements for limb derivations	10
Figure I.6:	Goldberger electrodes placements for augmented limb leads	10
Figure I.7:	Electrodes placement for precordial leads	11
Figure I.8:	Schematic representation of typical biomedical measurement system	11
Figure I.9:	A typical EKG waveform and its corresponding feature boundaries [4].	12
Figure I.10:	Diagram of a normal ECG Signal	13
Figure II.1:	Flowchart of training and test phases of biometric identification.	15
Figure II.2:	Impulse responses of FIR and IIR Filters.	17
Figure II.3:	(a) PCA from 2D to 1D | (b) Reduction from 3D to two principal components in (c).	22
Figure III.1:	(a) Overall ECG DAQ hardware system (b) Wireless data transfer using X-Bee (via ZigBee protocol) or (c) via USB COM port.	25
Figure III.2:	ECG electrodes (case-used ones and thumb electrodes).	26
Figure III.3:	Electrodes gel.	26
Figure III.4:	ECG/EKG Arduino shield by Olimex.	27
Figure III.5:	Pins and connections of ECG/EKG Shield.	28
Figure III.6:	Jack connections.	28
Figure III.7:	The Arduino Duemilanove board.	30
Figure III.8:	X-bee module and its USB Adapter.	31
Figure III.9:	Putty Terminal for X-Bee modules configuration.	32
Figure III.10:	Hardware Configuration/Programming flowchart.	34
Figure III.11:	Serial Communication over USB between Arduino and PC.	35
Figure III.12:	Data packet of 6 ECG channels.	36
Figure III.13:	Offline Acquisition results from Raw ECG/EKG using Matlab.	38
Figure III.14:	ECG monitored on stripchart on Matlab.	39
Figure III.15:	ECG acquisition Graphical User Interface using LabVIEW.	40
Figure III.16:	Block diagram of the LabVIEW monitoring application.	41

Figure IV.1:	MyDB snapshot from the workspace and directory.	42
Figure IV.2:	Methodology process of individuals' identification.	43
Figure IV.3:	Filtering ECG signal of MIT BIH 100.m M1 ECG signal.	44
Figure IV.4:	Pan Tompkins' peak detection algorithm applied to 100.m MIT BIH.	46
Figure IV.5:	3D mesh of concatenated ECG frames.	47
Figure IV.6:	MIT BIH 100.mat magnitude spectrum.	48
Figure IV.7:	Magnitude Spectrums of different ECGs on MIT-BIH DB.	48
Figure IV.8:	Identification testing phase flowchart.	51
Figure IV.9:	Demonstration of recognized individual using our identification method.	53
Figure IV.10:	Identification rate results of MIT BIH Database (a) and our customized database (b) in terms of principal components number.	54
Figure IV.11:	Identification rate (%) in terms of frames/QRSs number with 10 PCs.	55
Figure IV.12:	Linear incremental of elapsed execution time with number of frames/QRSs.	55

List of Tables

	Table Name	Page N°
Table III.1	Jumpers Connections of ECG shield.	29
Table III.2:	X-Bee modules configuration commands using Putty Terminal.	33
Table III.3:	Data transfer alternatives and readings testing.	37
Table III.4:	Buffer size testing on Matlab to store coming data.	37
Table IV.1:	Inputs and Outputs for direct use of Pan Tompkins Algorithm.	45
Table IV.2:	Accuracy and execution time in terms of testing heart-beats number at 10th dimension.	56

List of Acronyms

ECG/EKG	Electrocardiogram
QRS	QRS Complex
NSR	Normal Sinus Rhythm
PIN ID	Personal Identification Number / Identification/Identity
CMOS	Complementary Metal Oxide Semiconductor
IT	Information Technology
DNA	Deoxyribonucleic Acid
PCA	Principal Component Analysis
UART	Universal Asynchronous Receiver/Transmitter
DSP	Digital Signal Processing
VLSI	Very Large Scale of Integration
Fs	Frequency of Sampling
IIR	Infinite Impulse Response
FIR	Finite Impulse Response
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
PL	Power Line
EMG	Electromyography
FSM	Finite State Machine
USB	Universal Serial Bus
COM	Communication Port
VCP	Virtual Com Port
FTDI	Future Technology Devices International
IDE	Integrated Development Environment
KB	Kilobyte
RF	Radio Frequency
CMD	Commend Terminal
ADC	Analog to Digital Converter
MSB	Most Significant Byte/Bit
LSB	Most Significant Byte/Bit
ABS	Acquired Bits per Second
RAM	Random Access Memory
DAQ	Data Acquisition
LabVIEW	Laboratory Virtual Instrument Engineering Workbench
MIT	Massachusetts Institute of Technology
BIH	Beth Israel Hospital
HPF/LPF/ BPF	High Pass Filter/ Low Pass Filter/ Band Pass Filter
IEEE	Institute of Electrical and Electronics Engineering

General Introduction

Introduction

Biometrics is the science and technology of measuring and statistically analyzing biological data. It is widely used for authentication purposes. Nowadays, Biometry and its applications plays a crucial role in security and user authentication applications as well as many other applications like: Commercially (ATMs, Physical access control, Phone and Personal Computers…), crime investigation, time and attendance applications, boarders control in Airports, and for other biomedical purposes.

Several biometric parameters has been used for long time such as fingerprints, DNA, Iris, signature and voice. Some of these models can be copied easily (fingerprint from glass or signature), while others are hard to extract like DNA, meanwhile others' hardware is not widely available. Hence, we intend to develop a system using electrocardiogram that has high recognition rate while its hardware could have further uses in other aspects. The electrocardiogram is the electrical activity of the heart yielded from the depolarization and repolarization on the heart muscles during the process of blood pumping. ECG is a bio-signal that characterizes the individual and it is even different between identical twins. ECG acquisition hardware has other purposes such as monitoring in hospitals for telemedicine purposes and it could be used further than biometrics in biomedical applications.

In this work, we aim to develop a hardware/software ECG-based identification system to identify individuals from their heart's electrical activity (ECG). The hardware is an embedded system composed of sensors, ECG acquisition board (Olimex), and a micro-controlling unit between computer and the hardware. It is desirable to configure the hardware to hold up to six channels of ECG recordings using up to six cascaded ECG acquisition cards (6 people measurement simultaneously). We intend to make two alternatives of data transmission: via Serial COM port (through USB) or wireless transfer using XBee Radio Frequency modules of Zigbee protocol (for the sake of wireless monitoring). At the level of the software part, we use an identification process using Matlab to identify different people from their ECGs. This could be done by applying the basics of machine learning and pattern recognition. Hence, our highlighted objective is to write Matlab script in optimized way to reach high accuracy in shortest period of time.

We strongly believe that our project is tremendously beneficial for both biometric high security applications as well as telemedicine applications. This project has a socio-economic impact and could be expanded for further work to cover biomedical cardio anomalies detection.

Chapters' Overview/Highlight:

This dissertation consists of four chapters. The **first chapter** initiates the basic biometric aspects and state of the art. It introduces ECG/EKG and its characteristics and what can be extracted as features for interpretation (biometric and biomedical purposes). The chapter concludes with the "Why ECG?" as a biometric model. What are the advantages and disadvantages and why specifically ECG and not other biometric models.

The **second chapter** explain all the steps of the identification process using ECG starting from the pre-processing, feature extraction, and classification. This chapter recalls theoretical background of digital signal processing that will be used for pre-processing (filtering, windowing…etc.). Additionally, an initiation to machine learning and pattern recognition (classification) has been evoked in order to introduce PCA, Principal Component Analysis. The chapter ends up by demonstrating how the matching process is made in order end-up with the final identification decision.

In order to acquire the ECG/EKG Signal, an acquisition hardware design has been implemented to construct a customized database in order to test the performance of our algorithm. Hence, **chapter three** provides details of the used acquisition hardware starting from sensors and prototyping platform boards (Arduino, ECG Olimex Shield, XBee wireless module...etc.) till data reached the computer. Data acquisition consists of three main parts: Sensor readings, data transfer (Wireless and via Serial Port) and finally reading data from serial port on PC.

Ultimately, **chapter four** exhibits the Matlab implementation of ECG classification using PCA following the same scenario explained in chapter two. ECG goes through pre-processing, features extraction of spectral information, features database building and ends up with classification decision for class affectation for identification purposes. The identification (biometric) was based on the MIT-BIH database for the sake of validation. Then, we collected our database from ten volunteering students at IGEE institute to be used in our software. Results are discussed in comparison with MIT-BIH Arrhythmia database in terms of efficiency and accuracy of identification as well as execution time.

Chapter I:

Biometrics and ECG/EKG

Electrocardiogram

I.1. Introduction to Biometrics:

I.1.1. Biometrics definition and applications

Literally, Biometrics is extracted from two Greek words: Bio (life) and Metry (Measurement). Thus, the term implies to measurement of bio characteristics and physical behaviors of the human being for identification purposes. This term was used until 1980s. Nowadays, Biometry and its applications plays a radical role in security and user authentication applications.

Biometrics is the science and technology of measuring and statistically analyzing biological data. In IT, biometrics usually refers to technologies for measuring and analyzing bio signals such as Iris, ECG, Fingerprint…etc. for security and authentication purposes.

Any biometric system consists of four main modules: Acquisition, features extraction, classification, and finally decision-making. Depends on the used model of identification, the acquisition differs from one model to another. The following points exposes the most known biometric models [1].

Biometric Applications: Biometric technologies has diverse applicable solutions in different industries and businesses. Following are some examples:

- Physical Access Control: Protection in banks, governmental top secret locations, or secure military bases uses biometric identification instead of ordinary keys, PINs or keycards.
- Time and Attendance: Biometrics brings efficiency to a workplace by ensuring timing commitment of employees.
- Border Control/Airports: Most of international airports' security checkpoints uses biometric solutions in order to optimize time due to the huge number of daily travelers.
- Justice/Law Enforcement: Identity management is very important in crime scenes, investigations and homicides. Thankfully, biometrics enforces tracking criminals for justice/law purposes.

I.1.2. Some Biometric Models [2]

a) **DNA**

DNA is a large double-helix shaped molecule consisting of nucleotides (sugar molecule, phosphate molecule, and a nitrogenous base). Adenine (A), Cytosine (C), Guanine (G), and Thymine (T) are the four bases that biometric identification focuses on their sequencing for features' extraction.

DNA goes through Electrophoresis process in order to separate it into two stands and then placed into a coloring gel. An assembler machine labels each color with a binary representation to be used for matching. The Matching is no more than a simple comparison of sequencing of the four bases, i.e. bits. Generally, performance of small samples takes like 10 seconds which means no real-time aspect. Due to its individuality and pure uniqueness, DNA provides the most effective and reliable biometric identification model. However, the main hurdle is that it required sophisticated technics of extraction and handling.

b) **Iris**

The formation of the iris begins at early stage during the third month of gestation. It has very complex unique pattern, observable from distance, and it's stable. Its pattern provide very distinctive features like: ligaments, ridges, zigzag collarets… etc. Due to visibility of the iris, scanning can be done even with simple CMOS cameras which makes it user-friendly.

c) Face Recognition

The most the common recognition method is the face. It several advantages like mass identification, which means the possibility of detecting several people in a single captured image like in security cameras or airport surveillance cameras. However, many conditions can make the face recognition ineffective. Lighting, resolutions, or even facial expressions like big smile can reduce the effectivity of the identification system. Facial Recognition remains the most reasonable cost biometric system. Thus, it is widely used.

d) Fingerprint

Ridges and valleys of the fingerprint are the main features that could be extracted for identification process. A simple scanner perform image acquisition and sent it to a host computing unit. Matching scanned fingerprints with already existing features on a database is mainly based on minutiae and the much scrutinized details.

This model has several advantages like the reliability of authentication with very low false rejection rate and false acceptance rate provided that these rates are relative depending on the scanner and the detection algorithm. Nevertheless, there is always a possibility of fooling the identification system by faking fingerprints or extracting them from touched objects. Conditions like dry skin, low resolutions, or scars of the finger can lead to rejection by the host computer.

Identification through Fingerprint is widely used in biometric passports, IDs, as well as business/enterprises attendance especially with its reasonable costs.

e) Electrocardiogram ECG/EKG

ECG is completely unique and differs from one human being to another, even between identical twins. Our focus will be on biometric identification through ECG. Coming chapters will discuss this in more detailed way.

I.2. Electrocardiogram Generalities

I.2.1. Cardiovascular system

The cardiovascular system, called also the circulatory system, is an organ system that permits blood to circulate and transport nutrients (electrolytes, amino acids, etc.), carbon dioxide, oxygen, hormones, and blood cells from and to the cells inside the human body. This complex system consists of the heart (the anatomical pump) with its intricate conduits (arteries, veins, and capillaries). These last traverse the whole human body carrying blood [3].

A heart consists of four main components (shown in Figure I.1):

- Heart valves: They regulate the blood flow direction.
- Heart muscle: gives structure to the entire heart and divides the heart into four chambers
- Electrical system: synchronizes and controls the heart muscles' movements.
- Coronary arteries: carries blood, nutrient and O_2 to tissues

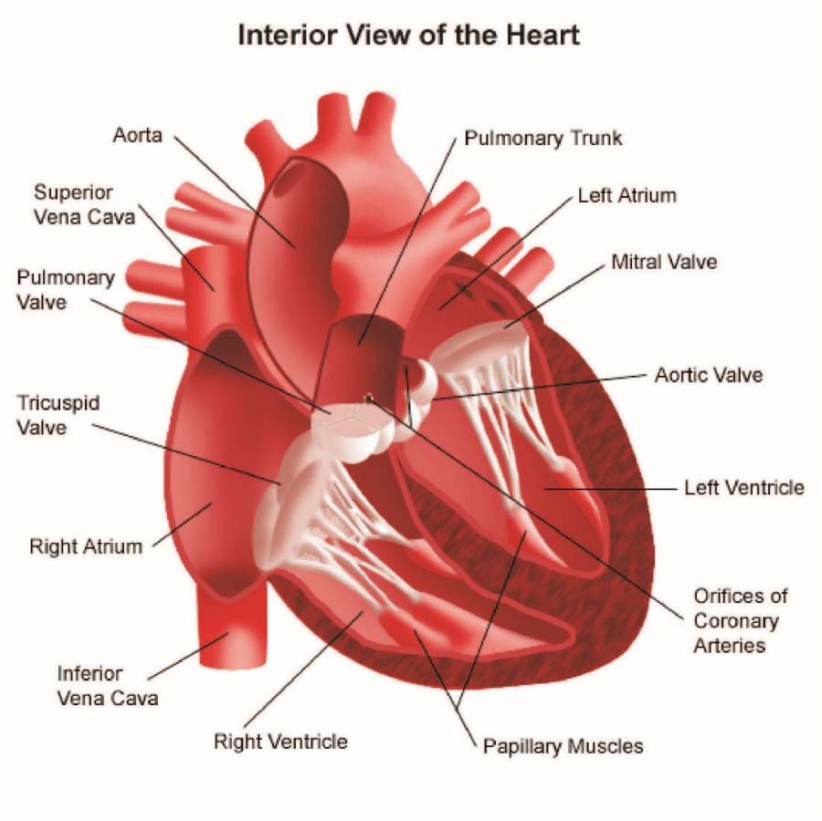


Figure I.1: *Anatomy of the heart, which consists of four chambers: the right and left atrium and the right and left ventricle [4].*

I.2.2. Electrical Activity of the heart

The electric currents of the heart have been measured for more than hundred years. A normal heart beat is initiated by a small pulse of electric current. This small electric "pulse" is rapidly spreading inside the heart to make the heart muscle contract.

When all heart muscles contracted at the same time, there would be no pumping. Therefore, the electric activity starts at the top and spreads down the heart, and then it goes up again, causing the heart muscle to contract optimally to pump blood.

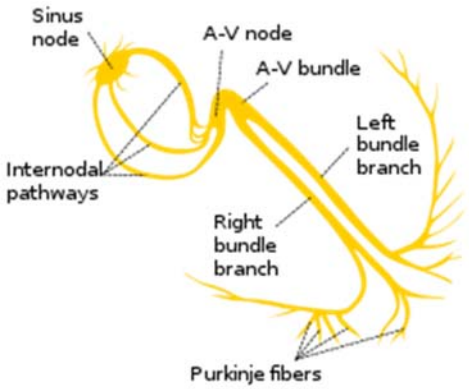


Figure I.2: *Isolated conduction system of the heart (Image extracted from Wikipedia).*

a) Electrical charges generation

Inside the heart, there are cells that are specialized in producing electricity. These are the “pacemaker cells”. These cells produce electricity by quickly changing their electrical charge from positive to negative and vice versa. The initial electric wave in a heartbeat is initiated on the top of the heart because of the spreading ability between adjacent cells. This initial wave is enough to keep a chain reaction going.

b) Repolarization and depolarization:

The cells change their electrical potential (electric charge) by what is so called depolarization and repolarization. Depolarization occurs when negatively charged ions inside the cell travel out from the cell through the cell membrane and positively charged ions travel in (repolarization).

c) Electrochemical mechanism:

Cardiac muscle has many similarities like the neurons and skeletal muscle, as well as some important unique properties. Like a neuron, a given myocardial cell has a negative membrane potential when at rest. Stimulation above a threshold value induces the opening of voltage-gated ion channels and a flood of cations into the cell. The positively charged ions entering the cell cause the depolarization characteristic of an action potential.

Like skeletal muscle, depolarization causes the opening of voltage-gated calcium channels and release of Ca_2^+ from the t-tubules. This influx of calcium causes calcium-induced calcium release from the sarcoplasmic reticulum, and free Ca_2^+ causes muscle contraction. After a delay, Potassium channels reopen and the resulting flow of K^+ out of the cell causes repolarization to the resting state [5][6].

I.3. Basics of ECG

In the beginning of the 20th century, he Dutch scientist Willem Einthoven developed what is nowadays known as the ECG signal fundamentals. In 1924, Willem Einthoven was awarded "Nobel Prize in Physiology/Medicine" for his discovery of the electrocardiogram mechanism.

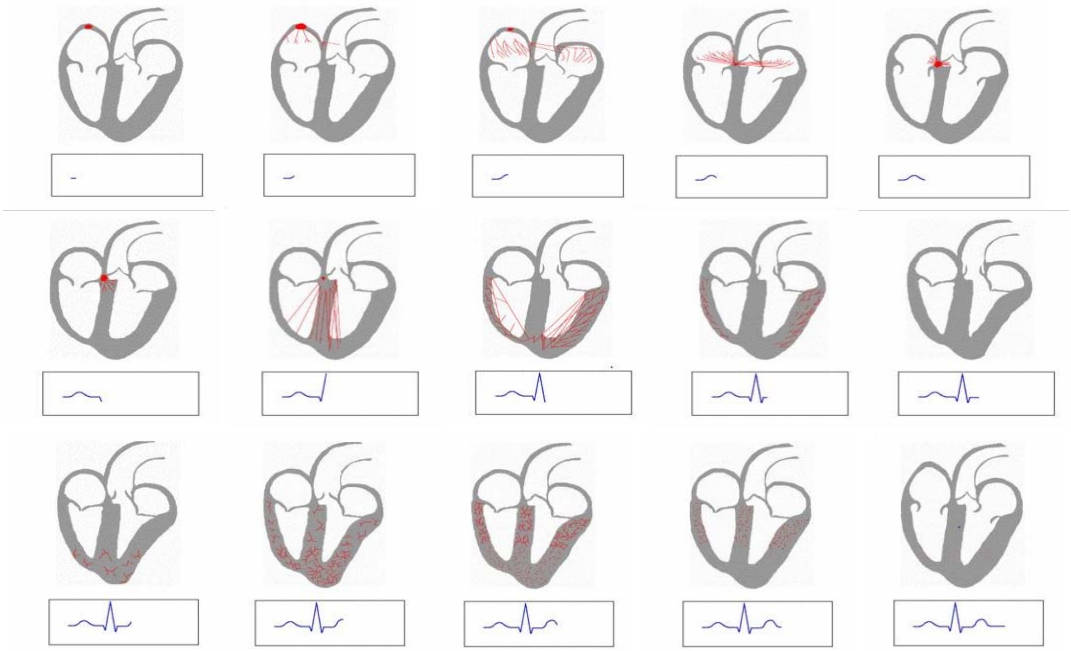


Figure I.3: *A sequence animation of an ordinary ECG waves with their respective cardio blood pumping (Animation from Wikipedia).*

Electrocardiography (ECG) is a data recording the electrical activity of the heart, which can be detected using attached electrodes to the skin. ECG is an objective measurement for the activation, transportation and recovery of heart activities. Since ECG demonstrates the activity of heart, it is a very important reference to diagnose cardiovascular diseases. In addition to that, the uniqueness of ECG for specific person makes it an important asset in biometric identification.

Figure I.3 shows consecutive series of snapshots at each stage of the electrocardiogram with their respective blood pumping action inside the heart's vessels.

I.3.1. ECG/EKG Waves and Complexes:

The ECG signal consists of waves, intervals, segments and one complex called QRS shown in figureI.4 below.

- **Wave**: A positive or negative deflection from the baseline indicated a specific electrical even. The wave of an ECG signal includes the P wave, Q wave, R wave, S wave, T and U Waves.
- **Interval**: The time between two specific ECG events. The intervals that are commonly measured on an ECG signal include the PR interval (PQ interval), the QT and RR interval.
- **Segment**: The length between two specific points on the ECG signal which are supposed to be at the baseline amplitude (not negative or positive). The segment ST segmented the TP segment.
- **Complex**: The combination of multiple waves grouped together, the only main complex on the ECG is the QRS complex.
- **Point**: There is only one single point on the ECG known as J point which is the junction between the end of the QRS and the beginning of ST segment.

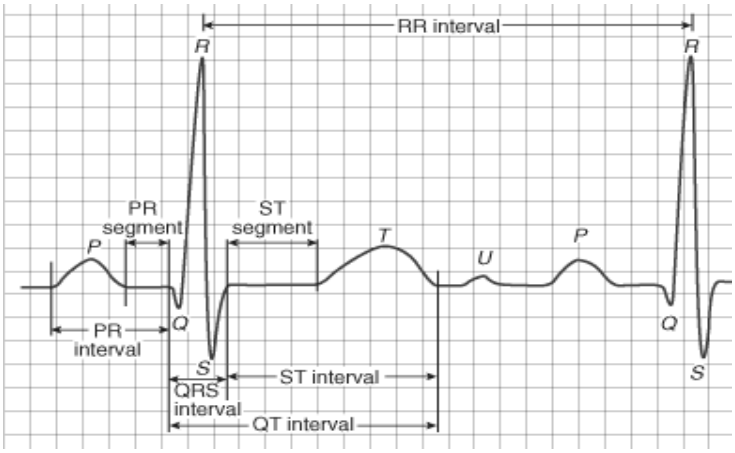


Figure I.4: *Intervals, complexes and segments of a normal heartbeat [7].*

P Wave:

The P wave occurs when the SA node (sinus node or sinoatrial node) creates action potential that depolarizes the atrial. The P wave should be upright in the lead II and biphasic in V1 if the action potential is originating from SA node. In this setting, the ECG is said to demonstrate a “normal sinus rhythm” aka NSR. As long as the atrial depolarization is able to spread through the AV node to the ventricles, each P wave should be followed by a QRS complex.

QRS Complex:

The QRS Complex is the combination of the Q wave, R wave, and S wave and it represents a ventricular depolarization. This term can be confusing since not all ECG leads contain all three of these waves, however a QRS complex is said to be present anyways. For example, the normal QRS in lead V1 does not contain a Q wave, but only R and S waves. This combination is still referred as the QRS complex for this lead. This normal time interval of this complex is generally 0.08 – 0.1 seconds (80 to 100ms). When the duration is between 0.1 - 0.12 seconds, then is it intermediate or slightly prolonged. If QRS duration is greater than 0.12, then is it considered as abnormal [8].

The QRS duration will lengthen when electrical activity takes a long time to travel throughout the ventricular myocardium. The normal conduction system in the ventricle is called the His-Purkinje system and consists of cells that can conduct electricity quite rapidly. Thus, normal conduction of an electrical impulse through the AV node then to the ventricles via the His-Purkinje system is fast causing a normal QRS duration. When electrical activity does not conduct through this system, a longer time is required which means widening the QRS duration [8].

T Wave:

After the QRS Complex, The T wave occurs and it is a result of ventricular repolarization. T wave should be upright in most of the derivations of ECG (except aVR and V1). T Wave should be asymmetric naturally. The second portion of this wave should have a steeper refuse when compared to the inclination of the first part of the T wave. If the T wave appears symmetric, cardiac pathology may be present such as ischemia [8].

I.3.2. ECG Derivations/Leads

ECG derivations, known as leads, are the potential difference between two specific points on the human body where measurements are being performed. In general, electrocardiography hardware can measure simultaneously several ECG leads. Each derivation corresponds to a heart contraction in order to capture specific cardio electric field. There exist three types of derivations: limb leads (I, II, III), augmented limb leads (aVI, aVR, aVF), and precordial leads (V1, V2, V3, V4, V5, V6) [9].

I.3.2.1. Limb leads:

These leads are bipolar which means potential difference between two points on the human body with respect to the ground potential. Three electrodes are placed triangularly in order to form what's known as Einthoven Triangle (shown in Figure I.5) [9].

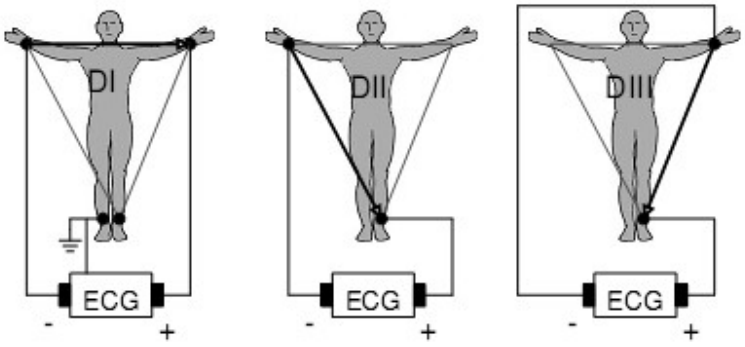


Figure I.5: *Einthoven triangle electrodes placements for limb derivations [9].*

The three bipolar limb derivations are:

DI (Derivation I): $DI = V_L - V_R$
DII (Derivation II): $DII = V_F - V_R$
DIII (Derivation III): $DIII = V_F - V_L$

Where: *V_L Left Arm Potential, V_R Right Arm potential, V_F Left Leg potential*

I.3.2.2. Augmented Leads:

Augmented leads were measured the first time by Wilson where he placed the electrodes on the central terminal of Einthoven's triangle. Wilson named these unipolar derivations (VL, VR, and VF). Later on, Goldberger introduced new adapted derivations called: aVL, aVR, and aVF (illustrated in Figure I.6.) where the letter "a" is "Augmented". These leads are extracted from the ordinary bipolar leads with an amplification factor of 1.5 [9].

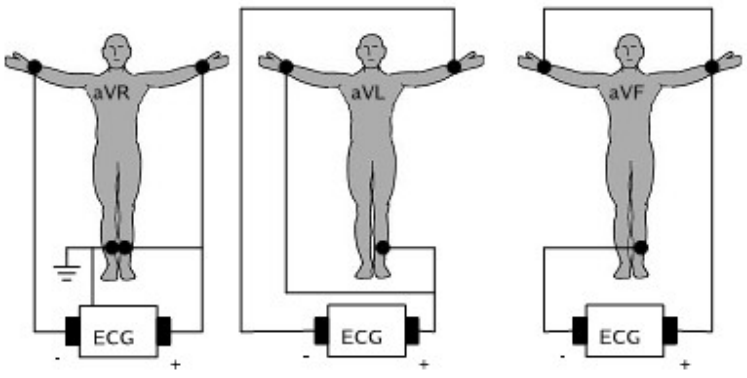


Figure I.6: *Goldberger electrodes placements for augmented limb leads [9].*

I.3.2.3. Precordial leads:

Precordial leads do lie on the horizontal (transverse) plane, in perpendicular matter to the other leads. Six electrodes are act as the positive poles while Wilson central terminal is used as the negative pole in order to result the six precordial leads. Following are the electrodes placement of the 6 leads (shown in Figure I.7) that are [9]:

V1: Between ribs 4 and 5, slightly to the left of sternum.

V2: Between ribs 4 and 5, slightly to the left of sternum.

V3: Between leads V2 and V4.

V4: Between Ribs 5 and 6.

V5: Horizontally even with V4.

V6: Horizontally even with V4/V5.

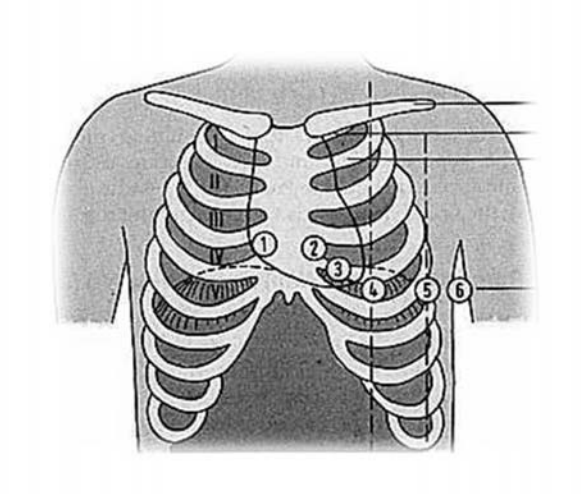


Figure I.7: *Electrodes placement for precordial leads [9].*

I.3.3. Measurement Systems:

From electrode to paper: Similarly to seismograph, an electrocardiograph records the heart electric waves in millivolts from the electrodes, or transducers, that are placed in the previously mentioned manners for a specific lead out of the 12 leads. These electrodes are responsible of controlling ink needles that write on a grid paper. The higher the intensity of the electric pulse, the upper the needle will move upper on the paper. The paper rotates at a specific speed beneath the needle, which results an ink curve [10].

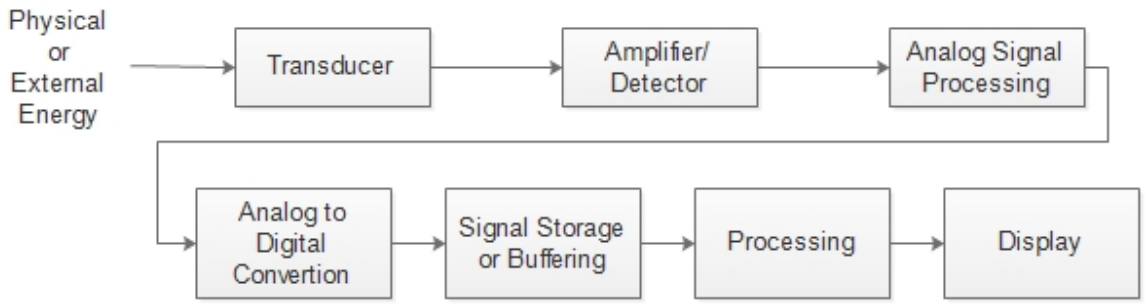


Figure I.8: *Schematic representation of typical biomedical measurement system [10].*

Modern systems use sophisticated electronics to acquire the ECG hardware (see figure I.8 above). ECG, like any other signals, is captured through transducers in touch with the skin using an isolation matter. The captured millivolts are analogally amplified and then digitalized in order to transmit them to a computing unit. The conversion from analog to digital using DACs occurs at certain speed known as sampling rate.

Transducers: A transducer is a device that converts energy from one form to another. ECG Transducers captures the ionic energy on top of the body and provides it in form of tiny electric pulses/voltages. Hence, transducers are called Electrodes.

I.4. ECG Interpretation

I.4.1. Features Extraction

In signal processing, there are several ways to extract desirable features from an ECG signal whether on time-domain or frequency-domain. Vector analysis enables efficiency of computations, accuracy, and robust extraction. Traditional time-domain analysis approaches are mainly based on interval analysis, which is discussed in the next point. Intervals are the main features used for interpretation. In frequency-domain, spectral analysis is introduced to analyze the repeated phenomena of specific ECG signal. In terms of vector space in the frequency domain, the spectrum consists of set of exponentials with different frequencies with their corresponding magnitudes, which appears as a resulting changes in our signal [11].

I.4.2. Interval Analysis

The timing between the onset and offset of particular features of the ECG, which is referred to as an interval, is of great importance since it provides a measure of the state of the heart and can indicate the presence of certain cardio-logical conditions.

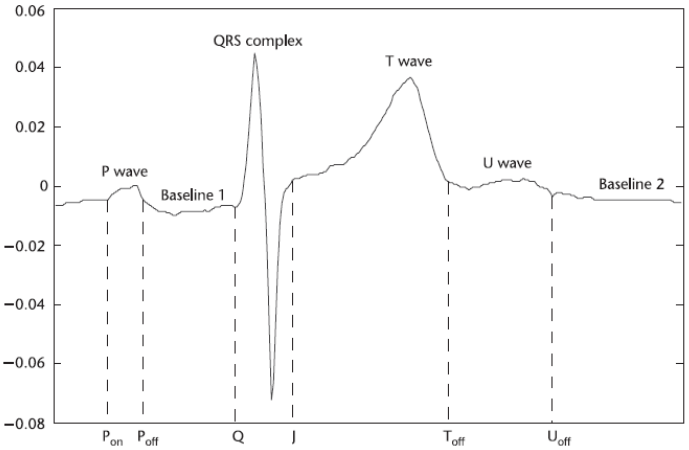


Figure I.9: *A typical EKG waveform and its corresponding feature boundaries [11].*

The two most important intervals in the ECG waveform are the QT interval and the PR interval. The QT interval is defined as the time from the start of the QRS complex to the end of the T wave (i.e., Toff−Q) and corresponds to the total duration of electrical activity (both depolarization and repolarization) in the ventricles. Similarly, the PR interval is defined as the time from the start of the P wave to the start of the QRS complex (i.e., Q−Pon) and corresponds to the time from the onset of atrial depolarization to the onset of ventricular depolarization. Changes in the QT interval are currently the gold standard for evaluating the effects of drugs on ventricular repolarization. In addition, changes in the PR interval can indicate the presence of specific cardiological conditions [11].

I.6. ECG Applications

I.5.1. Biomedical (Detection of Cardio anomalies)

A comprehensive EKG test is performed in a medical facility, which lasts for 5 to 10 minutes. Electrodes are attached to the subject's four extremities and to six locations on the front of the chest. The electrodes are attached by small suction cups, adhesive patches or Velcro strips. An ECG Machine records the data and monitors it as well as it saves it in a computer. Another type of testing called "Stress EKG test" which keeps the patient exercising until reaching a specific heart hate. This type of test provides higher probability of figuring out many details about the heart [12].

The following diagram is a normal ECG signal:

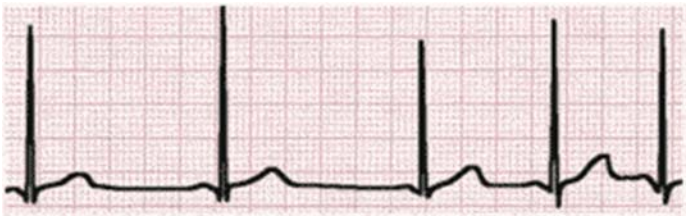


Figure I.10: *Diagram of a normal ECG Signal [12].*

Analyzing the pattern and frequency of the ECG recorded, many information can be extracted as follows:

1. Heart rate
2. Rhythm of the Heart
3. Conduction abnormalities
4. Evidence of increased thickness of heart muscle (hypertrophy)
5. Identification of damaged muscles
6. Heart orientation in the chest cavity

7. Acutely impaired blood flow to heart muscle
8. Warning signs of abnormal cardiac rhythm disturbances [12].

Based on the previous observations, ECG will help to define and diagnose for the following health conditions:

1. Fast or irregular heart rhythms
2. Slow Abnormal rhythms
3. Abnormal conduction of electrical impulses (These are symptoms for cardiac disorders).
4. Prior heart attacks Prediction (myocardial infarction)
5. Reduced blood flow (unstable angina)
6. Enlarged heart chambers (cardiac dilatation)
7. Proof of abnormal blood electrolytes (such as calcium, potassium, magnesium)
8. Heart Inflammation [12].

I.5.2. Biometric (Identification Application)

Different studies in the recent past have shown the feasibility of ECG as a new candidate of biometric for individual authentication.

Our main topic of this Master dissertation provides greater details in the coming chapters about the biometric identification aspect of the ECG.

I.6. Reasons for choosing ECG

ECG/EKG is one of the most interesting bio signals for biometric/biomedical utilization more than the other models like Face, fingerprints… etc. Reasons behind choosing ECG as modality for our thesis is as follows:

- Performance degree and high accuracy of identification
- Impossible to be faked (like Fingerprint for example) as it's an indication of human being's life.
- Wireless monitoring of the heart activity.
- Further studies could be done with ECG/EKG like cardio anomalies detection with the same analogy of machine learning and pattern recognition.

I.7. Summary:

This chapter introduced the state of the art of this project and exposed generalities about the Electrocardiogram and it uses. The chapter concludes by "why" ECG as a biometric model. The coming chapter explains the procedure of how ECG can be useful for biometric identification.

Chapter II:
ECG/EKG Processing For Biometrics

II.1. Process Overview

This chapter introduces the theoretical background of all the steps performed in the biometric processing of the electrocardiogram ECG. It is delineated to follow the biometric algorithm's scenario starting from the signal pre-processing till reaching a final decision of identification.

There are mainly two phases of the algorithm: Training and Testing. In training, machines are being indoctrinated and exposed to certain pattern of data (ECG training samples). Features get extracted from ECGs of the database and taken to higher dimensionality proper space for more detailed discrimination. In the test phase, the data passes through the same training scenario and ends up by being affected into specific set or class in the proper space using what is known as classifiers. The following diagram in Figure II.1 elucidates more the two phases.

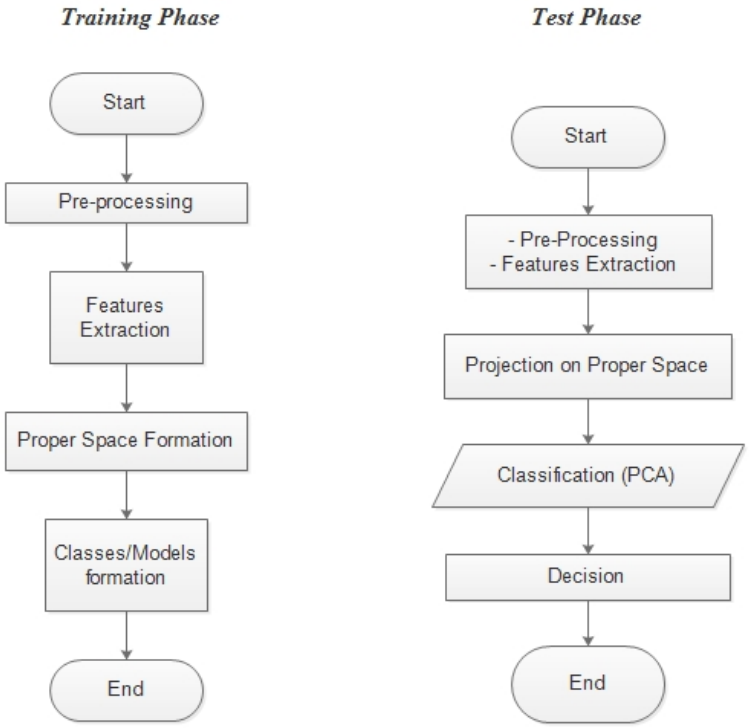


Figure II.1: *Flowchart of training and test phases of biometric identification.*

This chapter exposes all the signal processing techniques, classification, and algebraic theories applied that have been used in our coding implementation on the fourth chapter.

II.2. ECG Signal Pre-Processing

II.2.1. ECG Noises/Artifacts

The ECG recordings might be corrupted by different sources of noises/artifacts, and the following six categories can be successfully treated or filtered with the previously mentioned pre-processing techniques [13]:

- Power-line interference (PL): The Power Line often contaminates the ECG recordings, due to differences in the electrodes' impedances and to stray currents through the patient and the electrodes' cables, or in measurement instruments themselves.

- Electromyography - EMG noise: It is common in subjects with uncontrollable tremor, disabled persons, and kids, people fearing the ECG procedure, and tense muscles during stress testing exercises.

- Baseline drift: This is caused by changes in electrode to skin polarization potentials/voltages, by electrode movement, by respiration movement or by body motion in general to cause noise on the ECG.

- Flat line/missing lead: Bad electrode-to-skin contact causes this type of noise/artifacts. Other factors such as, electrodes' disconnection, producing a flat lead recording at a zero or at a saturation level, might be the cause as well.

- Low amplitude signal: A persistent low amplitude signal in some leads/derivations may be produced either by a projection almost perpendicular to the cardiac vector; or by an increased electrode-to-skin impedance. It might appear as sudden changes from normal to low amplitudes and vice-versa.

- Steep slope/spike noise: High-frequency noise caused by abrupt movement of the electrodes can produce distorted QRS, known erroneous/spurious beats. The pre-processing techniques or the QRS detection algorithms are able to detect and remove the effect of such spurious spikes.

II.2.2. Digital Filtering Generalities

In order to eliminate all types of ECG contaminations in terms of noise and artifacts are introduced to de-noise it as well as smoothing it. There are two types of digital filters: Infinite Impulse Response and Finite Impulse Response. Butterworth and Median filters are invoked as they will be used on the experimental part on chapter four.

II.2.2.1. IIR and FIR Filters:

Discrete-time Linear Time-Invariant systems are classified into FIR or IIR systems, that is, having finite or infinite impulse response *h(n)* as shown in figure II.2.

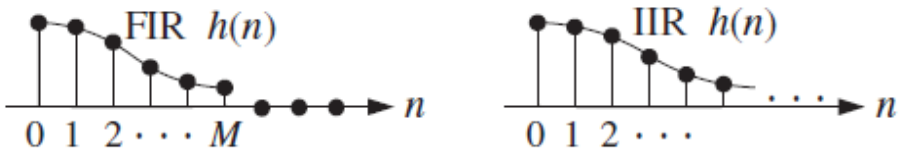


Figure II.2: *Impulse responses of FIR and IIR Filters.*

As inferred from its name, Finite Impulse Response has impulses only on finite time interval [0,M] where M is the filter order.

$$\{h(0), h(1), h(2), \ldots, h(M)\}$$

The length of the $h(i)$ coefficients (weights or taps) is: $L = M + 1$

FIR Filters are characterized by the following input/output equation (II.1):

$$y(n) = \sum_{m=0}^{M} h(m)\,x(n-m) \qquad \text{(II.1)}$$

Explicitly expressed as a weighted sum of the present input sample *x(n)* and the past M samples:

$$y(n) = h(0)x(n) + h(1)x(n-1) + h(2)x(n-2) + \cdots + h(M)\,x(n-M) \qquad \text{(II.2)}$$

Similarly, IIR filter has infinite pulses defined on the interval $0 \leq n < \infty$ and expressed by the following equation II.3:

$$y(n) = \sum_{m=0}^{\infty} h(m)\,x(n-m) \qquad \text{(II.3)}$$

Filters, whether analog or digital, remove selected frequencies and are named according to their range of allowed frequencies (*lowpass*, *highpass*, *bandpass*, *bandstop*). Any filter is characterized by its type, order, bandwidth and its attenuation characteristics.

II.2.2.2. Case used Filters (Butterworth and Median)

a) Butterworth

In 1930, an English scientist Stephan Butterworth described this filter. Butterworth is a flat response in the passband and its transfer function is given by:

$$H(z) = \frac{B(z)}{A(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_N z^{-N}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_M z^{-M}} \qquad \text{(II.4)}$$

The frequency response of Butterworth: $|H(jw)|^2 = \frac{H_0}{1+(\omega/\omega_0)^{2n}}$ (II.5)

b) Media Filter

Median filter is a non-linear digital filter to perform noise reduction on any signal or image. In pre-processing, Median Filter is used mainly for smoothing signals and images, and removal of baseline wander noise. It is widely used because it preserves edges while removing noise.

Following is an example of 3-dimentional window size median filter on a simple 1D vector:

$X =$	3	3	9	4	52	3	8	6	2	2	9

y[1] = median[3 3 9] = 3
y[2] = median[3 4 9] = 4
…
y[9] = median[2 2 9] = 2
y[10] = median[2 9 9] = 9

$Y =$	3	4	9	4	8	6	6	2	2	9

In our ECG case, Median filter is applied on the hardware (Described on the third chapter) to de-noise and smooth the acquired ECG signal before transmission in order to avoid accumulated noise of the wireless transmission.

II.3. Features Extraction

II.3.1. Peaks Detection

In order to detect the peaks in the QRS complex of any ECG signal, Pan J. and Tompkins WJ. Developed an algorithm that was published in IEEE Tran. Biomed. Eng. In 1985 [14]. The algorithm follows a scrutinized signal preprocessing and decision rules as follows:

- Pre-Detection: The read ECG signal goes through cascaded types of filters. First, the sampling rate needs to be up-scaled/down-scaled in case if our signal is not sampled at 200 Hz in order to be compatible with algorithm's sampling rate. After that, the signal will be pass through LPF, HPF, BPF and derivative filter for noise cancelation as well as smoothing our signal (filtering high and low frequencies). Squaring nonlinearly will enhance the dominant peaks and then the signal is averaged with a moving window to get rid of noise.

- Decision Rule: Firstly, the Maxima points are located with the first step, called Fiducial Mark, with their weighting w[k]. Each time the algorithm uses two thresholds, THR_SIG and THR_NOISE, and adapts them continuously. After locating the QRS complexes, the algorithms checks back if there are any missing complexes again. It makes sure to avoid any multiple detection of QRS. T-Wave Discrimination, which is one-step before outputting the signal, ensures the non-overlapping between a T-Wave and a QRS complex by assuming that a potential QRS cannot occur after 200ms refractory period.

Finally, the detection is finalized by double checking the detection with the output a band pass filter and finding the original indices on the raw ECG signal.

II.3.2. Framing/Windowing

In the emerging field of medical image processing, computer vision, pattern recognition and other digital signal processing applications, window technique is vastly used. A window function is a mathematical function that is zero-valued outside of some chosen interval. When another function is multiplied by a window function, the product is also zero-valued outside the interval [15].

A major effect of windowing is that the discontinuities of the frequency response are converted into transition bands between values on either side of the discontinuity.

There are many window techniques available for designing the FIR filter and they are: Hanning window, Hamming window, Blackman window, Rectangular window, Bartlett window, Kaiser Window.

The rectangular window: As inferred from its name, has value of one along its length. The following equation defines the rectangular window.

$$w(n) = 1.0 \text{ for } n = 0, 1, 2, \ldots, N-1 \qquad \text{(II.6)}$$

Some of used windows in our ECG case are characterized by their following equations:

Hamming window: $$w(n) = \alpha - \beta \cos\left(\frac{2\pi n}{N-1}\right) \; for \; -\frac{N-1}{2} \leq n \leq \frac{N-1}{2} \qquad \text{(II.7)}$$

With $\alpha = 0.54$ and $\beta = (1-\alpha) = 0.46$

The width of the main lobe is approximately 8π/N and the peak of the first side lobe is at -43dB. The side roll off is 20 dB/decade.

Hanning window: $$w_{Hann}(n) = \begin{cases} 0.5 - 0.5\cos\frac{2\pi n}{N-1}, 0 \leq n \leq N-1 \\ 0, otherwise \end{cases} \qquad \text{(II.8)}$$

The width of the main lobe is approximately 8π/N and the peak of the first side lobe is at -32dB.

Blackman window: $w_{Black}(n) = a_0 + a_1 + a_2 \cos\frac{4\pi n}{N-1} \; for \; -\frac{N-1}{2} \leq n \leq \frac{N-1}{2}$ (II.9)

With $a_0 = 0.42$; $a_1 = 0.5$; $a_2 = 0.08$

II.3.3. Spectral Information Extraction

Spectral information reflects the different frequency components of the ECG signal. As every single individual is characterized with its own ECG in terms of amplitude, frequencies, rhythm, it is interesting to extract the features in form of amplitude spectrum rather than using the ordinary techniques of interval analysis in time domain. Hence, applying Fast Fourier Transform to ECGs yields very differentiable patterns that could be as features in classification in later steps.

Discrete and Fast Fourier Transform (DFT and FFT):

The discrete Fourier transform (DFT) and its fast implementation, the fast Fourier transform (FFT), have three major uses in DSP: (a) the numerical computation of the frequency spectrum of a signal; (b) the efficient implementation of convolution by the FFT; and (c) the coding of waveforms, such as speech or pictures, for efficient transmission and storage [16].

The DFT may be easily developed from the discrete Fourier series representation for periodic sequences. DFT is defined by the following equation:

$$x(n) \overset{DFT}{\Longleftrightarrow} X(k)$$

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi nk/N} \qquad 0 \leq k < N\text{-}1 \qquad \text{(II.10)}$$

Fast Fourier Transform is an optimized technique to compute to the DFT of a signal. Applying the Discrete Fourier Transform on a discrete signal is the process of mapping it from the time domain to the frequency domain. In terms of vector space in the frequency domain, the base consists of set of exponentials with different frequencies applied to a specific signal and appears as a resulting change of response in our signal.

There are huge differences between DFT and FFT in terms of complexity and sampling number. FFT has lower complexity which computes $(N/2)\ log_2\ (N)$ compared to DFT. This is very important aspect for execution time on computer algorithms.

II.4. Classification of ECG Signals

II.4.1. Introduction to ECG Classification

Classification is a decision making task to assign a specific object to a group/class based on the number of observed attributes related to that object [17]. Is it frequently encountered in machine learning and pattern recognition for different purposes: Robotics, Image Processing, Signal Processing, Biomedical…etc.

ECG Classification differs according to purpose whether biometric or biomedical. It is very important to classify whether an ECG is normal or abnormal and in case if abnormal, the ECG itself can be classified into several cardio anomaly classes. Likewise, classes can be individuals which provides the biometric aspect for identification.

Classification algorithms are mainly composed of two main pillars: Training and decision making (classification/prediction). In training phase, machines are indoctrinated with a set of training samples in order to form models/classes. Classes can be any kind of interesting features from the desired signal. A testing sample goes through the features extraction. Then, a classifier assign it to its respective class in the database. This can test the feasibility and accuracy of the used classifier [17].

With the same analogy, unknown samples can be classified into different groups in the database of models/classes. There are two types of classifiers: Supervised and unsupervised classifiers. Some of them are in either time domain or frequency domain. Following point exposes several types of classifiers.

There exist several types of classification techniques (known as classifiers) like Support Vector Machine, K-Nearest Neighbor, Linear Discriminant Analysis, Neural Network and …etc. The one that is used in our case is the Principal Component Analysis.

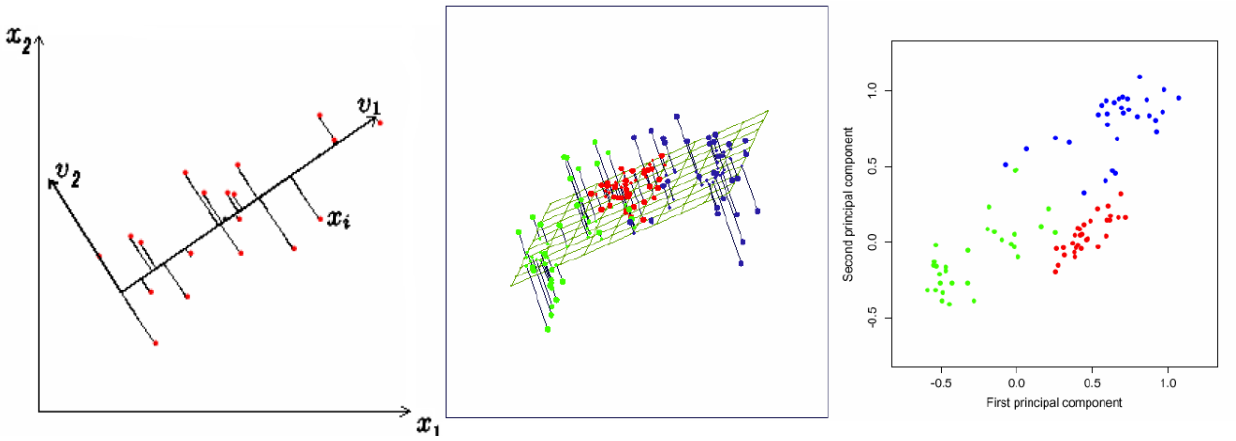
II.4.2. Principal Component Analysis (PCA)

II.4.2.1. PCA Theory

Principal Component Analysis PCA statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components [18]. PCA is performed to increase the dimensionality. Due to the heavier the computations in higher dimensionality spaces, PCA is introduced in order to reduce the number of components for the sake computations reduction. This means smaller execution time of classification process.

In machine learning and statistics, dimensionality reduction or dimension reduction is the process of reducing the number of components under consideration and can be divided into feature selection and feature extraction [19].

Feature extraction transforms the data in the high-dimensional space to a space of fewer dimensions. The data transformation can be linear, as in principal component analysis (PCA), as well as nonlinear.



(a) *(b)* *(c)*

Figure II.3: *(a) PCA from 2D to 1D | (b) Reduction from 3D to two principal components in (c) [20].*

The eigenvectors that correspond to the largest eigenvalues (the principal components) can now be used to reconstruct a large fraction of the variance of the original data. Moreover, the first few eigenvectors can often be interpreted in terms of the large-scale physical behavior of the system. The original space (with dimension of the number of points, i.e. d = *256*) has been reduced (with data loss, but hopefully retaining the most important variance) to the space spanned by a few eigenvectors (n < *d*). Figure II.3 illustrates visually dimensionality reduction. Scattered points in (a) are projected on principal component base vector that provides the maximum variance. 3D data in Figure II.3 (b) are rotated in such a way that can be discriminated linearly. Hence, only two components are taken and one is dropped with minor details loss. In machine learning this process is also called low-dimensional embedding.

II.4.2.2. Identification Implementation Steps

PCA approach could be summarized in the following points as general steps:

- Take the whole features' datasets (both training and testing data) consisting of d-dimensional samples.
- Compute the d-dimensional mean vector (i.e., this means finding the center of gravity in order to perform translation).
- Compute the scatter matrix (alternatively, the covariance matrix) of the whole data set.
- Get eigenvectors and corresponding eigenvalues from the covariance matrix.
- Perform dimensionality reduction: Span a subspace from the proper space. This point will be discussed in the point to come next.
- Perform distance measurements between classes (Euclidian Distance).
- Assign labels to the matched vectors (Labels assignment).
- Decision making and identification rate calculation (Labels Comparison).

II.4.3. Proper Space Formation

II.4.3.1. Overview

All classification of extracted features are going to be performed in higher dimensionality space known as features space or proper space. Basically, points in 1-D (dimension) are vectors of single component and one base vector *i*. Points in 2-D are vectors of two components and two base vectors *i* and *j*. In 3-D, points are vectors of three components and three base vectors *i*, *j*, and *k*.

II.4.3.2. Data Centralization and Rotation

Conceptually, the proper space should be formed in such a way to give data distribution with maximum variance. In order to get the variance, all training data needs to be centralized to a gravity center and rotated to give the maximum discrimination or distribution among the features in the proper space.

Mathematically, the translation of data into the gravity center is no more than subtracting the mean of all training data, known as scatter matrix. Rotation is the computation of the covariance matrix which is the results of multiplication of the scattered data matrix to its inverse. Covariance matrix provides tremendous benefit to the computation of eigenvalues and eigenvectors which will be used in the next space to form the space. The variance of each variable is the average squared deviation of its n values around the mean of that variable. Variance is described with the following equation.

$$V = \frac{1}{n}\sum_{i=1}^{n}\left(X_{i\,Training} - \overline{X}\right)^2 \qquad \text{(II.11)}$$

where n is the total number of training data vectors and $\overline{X}$ is the mean of all training data.

II.4.3.3. Axes and Power of discrimination

The covariance matrix computed from the scattered data matrix enables finding the eigenvalues and their corresponding eigenvectors. The following equation is helpful to verify the correctness of computations.

$$\sum \boldsymbol{v} = \boldsymbol{\lambda_i}\, \boldsymbol{v_i} \tag{II.12}$$

where $\boldsymbol{v_i}$ = Eigenvector, $\boldsymbol{\lambda_i}$ = Eigenvalue, and $\sum$ = Covariance Matrix.

In PCA, the eigenvectors are the axes of the features/proper space. The greater the eigenvector, the greater its power of discrimination among all projected data to this proper space. Note that dimensionality reduction on later steps is based on power of discrimination and their eigenvectors.

II.4.4. Training and Testing Phases

Training data and testing data needs to follow the previously described steps in order to be enable the matcher to classify them respectively. Features and testing portions are projected into the higher dimensionality. Assume that original matrix contains d dimensions and n observations and it is required to reduce the dimensionality into a k dimensional subspace then its transformation can be given by

$$Y = E_{ProperSpace}\, X_{Data} \tag{II.13}$$

Here E is the projection matrix which contains k eigenvectors corresponding to k highest eigenvalues, and where X is mean centered data matrix. Once features and testing features are on proper space, a matching process could be performed as explained in the coming point.

II.4.5. Classification Decision

Testing data are labeled to the corresponding person. In proper space, vectors are compared using Euclidean distance and assigned to the label of the found class. Many distances use the norm and the Euclidean distance. It is calculated using this formula:

$$\boldsymbol{D} = \sqrt{\sum (\boldsymbol{X}_{\boldsymbol{TestingVector}} - \boldsymbol{X}_{\boldsymbol{TrainigVector}})^2} \tag{II.14}$$

Comparison between assigned label and found label makes the decision deterministic and enables calculating the identification rate which reflects the algorithm's efficiency.

II.5. Summary:

This chapter consist of all basic steps for biometric identification process which will be used in the last chapter. The next one is about the acquisition hardware that was used to collect a database.

Chapter III:
ECG Acquisition Hardware Implementation

III.2. Introduction

The proposed acquisition hardware used in this project is a basic embedded system using the ECG electrodes (sensors), EKG shield and Arduino and connected to a computer via serial communication whether wirelessly via X-Bee or USB port. Up to 6 ECG cards can be cascaded as shields to Arduino.

The diagram below (Figure III.1) shows the overall interconnected hardware acquisition system:

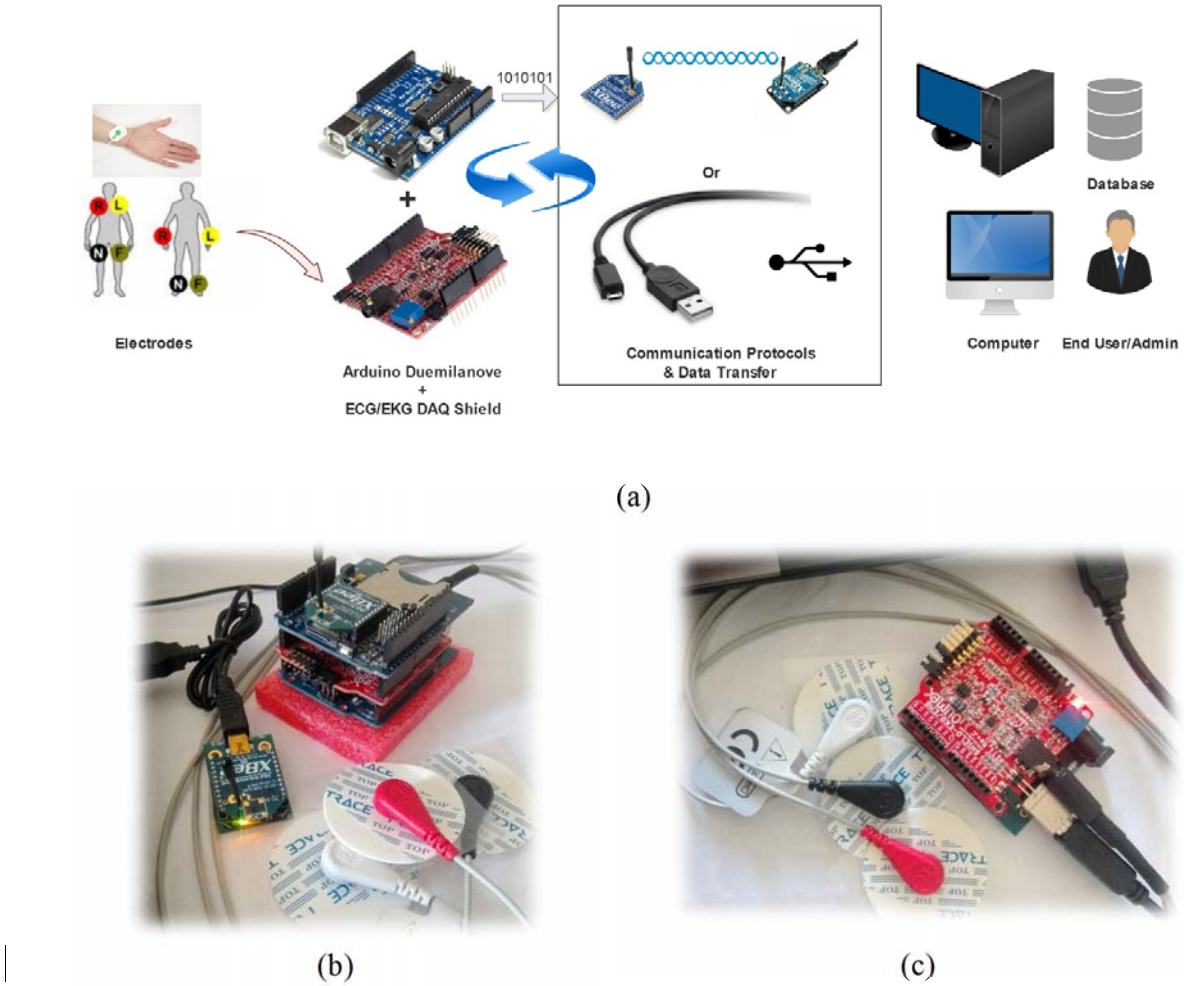


Figure III.1: *(a) Overall ECG DAQ hardware system (b) Wireless data transfer using X-Bee (via ZigBee protocol) or (c) via USB COM port.*

This chapter is divided into two main portions: Basics of all used separate parts and the implementation of the hardware and its configuration. The hardware parts follows the scenario from the skin, passing by the all components of the system, until reaching the USB port. The second portion exhibits the hardware configuration/programming of the Arduino in form of flowchart, and how data packets are gathered from COM port and reconstructed into ECG samples for monitoring and database construction as well.

III.2. Sensors

III.2.1. Electrodes and Patches

Electrodes used to measure ECG voltages are in direct contact with the human skin. It contains small piece of metal that is characterized by high ability to catch low amplitudes (0,05mV to 10mV), very high input impedance, and low current input at level of 1mA.

The quality of ECG recordings depend on the factors: High electrodes quality, good skin preparation, and maintained electrical pathway between skin and ECG machine. It happens so often that conditions are kind of strenuous and electrodes have to perform acquisition. This is relative to the finest quality materials and the components used for production.

There are different types (as shown in Figure III.2): the ones used in our case are three terminal electrodes. For such biometric application, it would be more relevant to use the thumb ones (similar to fingerprint scanning, but for ECG). Electrodes are marked L for left arm, R for right arm, and D for DRL.

Figure III.2: *ECG electrodes [21].*

- Gels: Conductive mediums (see Figure III.3) are put between the patches and the skin to maintain contact and remove electrostatic charges. The aqueous nature of the gel provides a skin impedance reduction which produces quality of recording with more stable baselines.

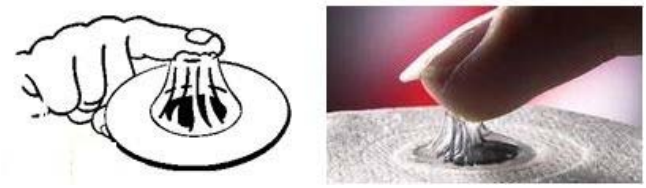


Figure III.3: *Electrodes gel [22].*

III.2.2. Einthoven's Triangle

Electrodes position to capture ECG in our case is Einthoven's bipolar lead-II. The electrodes are put on the hands to form an equilateral triangle. The leg is considered the neutral terminal.

III.3. ECG/EKG Shield

III.3.1. Overview:

The EKG-EMG acquisition hardware used in our project is a shield (an extension module) for Arduino platform boards manufactured by the Bulgarian company Olimex. The module is suitable for both Electrocardiography and Electromyography for the sake of data collection (for monitoring, processing…etc.). The shield is also compatible with other boards like Olimexino, Penguino, Maple, and PIC 32…etc.

The open hardware Olimex ECG/EMG shield is characterized by features for a flexible user-experience. It enables the developer to use up to six channels mounted on top of each other in form of stackable shield. Following are some of the features:

4. Digital outputs for generating a calibration signal (D4/D9)
5. Calibration potentiation with high precision
6. Voltage compatibility with both 3.3V and 5V Arduinos.
7. Possibility to connect passive or active electrodes.

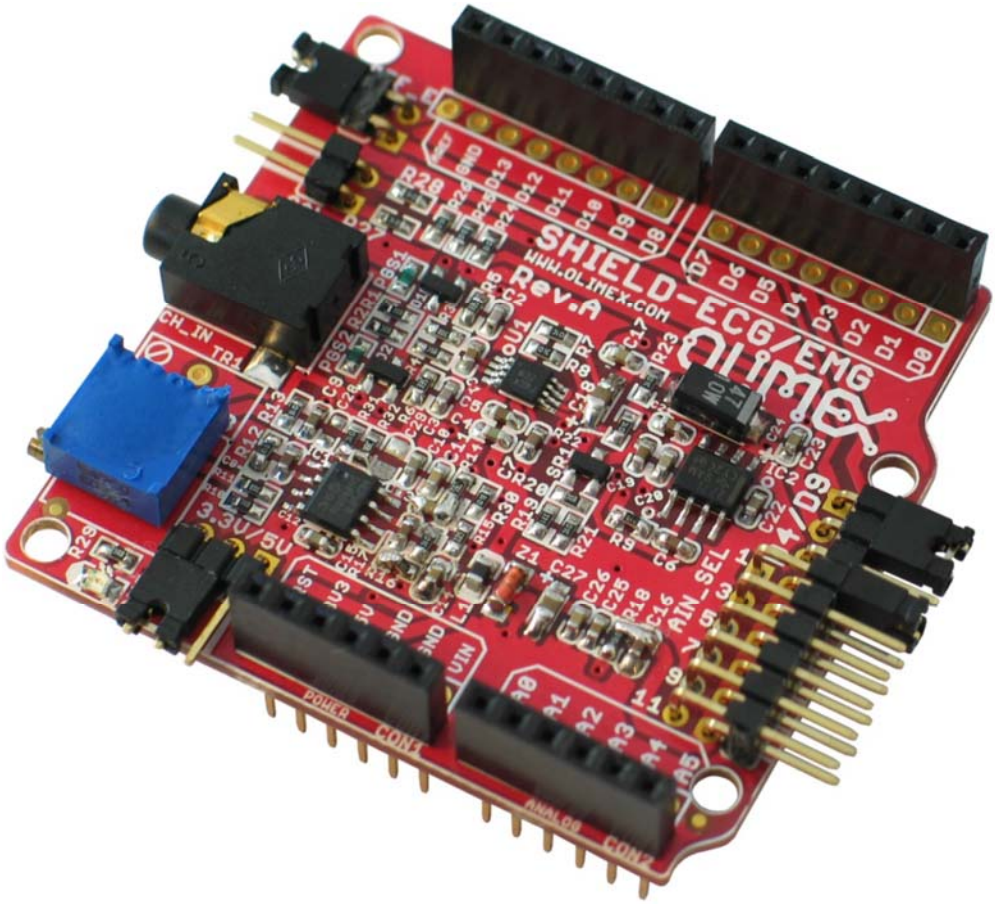


Figure III.4: *ECG/EKG Arduino shield by Olimex [23].*

III.3.2. Functionality (Schematics explanation/Description)

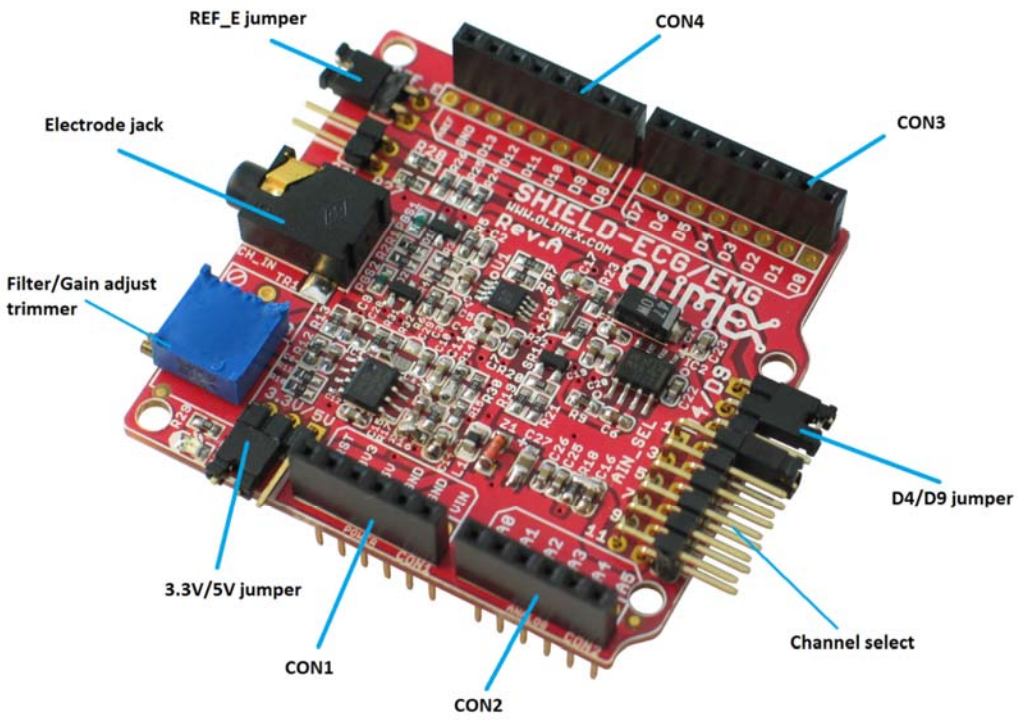


Figure III.5: *Pins and connections of ECG/EKG Shield [23].*

Electrode Jack: The Arduino shield ECG/EKG uses basic passive/active electrodes to be plugged similarly as the Audio Jack. The way of connecting is show below:

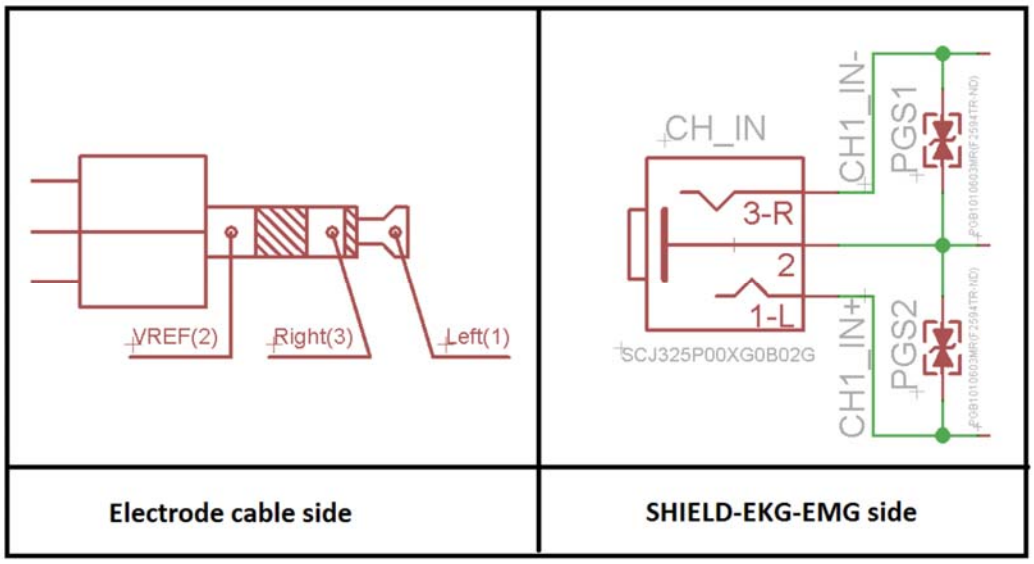


Figure III.6: *Jack connections [23].*

- **CON1-4:** These connectors are the same of Arduino pins for another compatibility use for other shields or utilization of MCU on Arduino.

- **REF_E Jumper:** If the host board provides a reference voltage to the shield, then it should be open. Otherwise, it needs to be closed to ensure that the board might not have any damage due to voltage conflicts.

- **Filter/Gain Adjust Trimmer:** This is a potentiometer is used for factory settings calibration. Generally, it is already calibrated.

- **3.3V/5V:** These pins controls select power level of the used board.

- **Channel Select:** This selects the channel the current shield would utilize. There is up to 6 channels that could be mounted. In our case, we need only a single channel for the ECG signal.

- **D4/D9:** Controls pin D4/D9 for interrupts signal generating. This is selected according to the used board.

III.3.3. Shield Setting-up:

1. Jumpers on the EMG/EKG shield needs to be in the following way:

Jumper	Position
REF_E	Closed
3.3V/5V	According to used board (5V with Arduino Mega and Duemilanove)
D4/D9	D9 position
ANI_SEL	Channel 1

Table III.1: *Jumpers Connections of ECG shield.*

2. Plug-in the shield on the Arduino board and connecting it to PC.
3. Installing VCP FTDI for proper Port COM selection.
4. Uploading the sketch to Arduino board: Tools -> Board -> Arduino Duemilanove w/ ATmega328 and Tools -> Serial port -> COMx then clicking on Upload.

5. A serial communication will start between the acquisition hardware and the COM port. The generic ElectricGuru monitor can be used to display the acquired raw ECG signal.

Note: Both TimerOne and FlexiTimer2 libraries should be included within your code in order to enable the code properly compile on the Arduino IDE.

III.4. Arduino

III.4.1. Introduction

For the sake of interfacing the ECG/EKG shield with the computer, Arduino board has been used for data transfer as well as sampling the analog signal from the Olimex shield. During ECG recording, the ATmega microcontroller is put to sleep mode. Analog values (up to six channels) are read from analog pins and then digitized using the Analog to Digital Converter on the board. For data transfer, the mini USB port of Arduino is used.

Arduino is an open source physical computing and prototyping platform board based on a simple input/output (I/O) board and a development environment that implements the Processing language. Arduino can be used to develop standalone interactive objects or can be connected to software on your computer (such as Flash, Processing, VVVV, or Max/MSP). The boards can be assembled by hand or purchased pre-assembled; the open source IDE (Integrated Development Environment) can be downloaded for free from Arduino's official website [24].

III.4.2. "Duemilanove" Board Characteristics

Duemilanove ("2009" in Italian) is an Arduino board based on ATmega328/ATmega168 microcontroller. It has 6 analog inputs, 14 digital I/O pins (6 can be PWM outputs). Its clock speed is 16MHz and it has USB connection, ISCP header, rst-button and power jack [24].

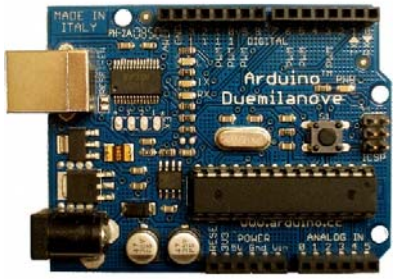


Figure III.7: *The Arduino Duemilanove board.*

Microcontroller	ATmega168
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limits)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	16 KB (ATmega168) or 32 KB
SRAM	1 KB (ATmega168) or 2 KB
EEPROM	512 bytes (ATmega168) or 1 KB
Clock Speed	16 MHz

III.4.3. Programming

Arduino is composed of two major parts: the Arduino board, which is the piece of hardware you work on when you build your objects; and the Arduino IDE, the piece of software you run on your computer. You use the IDE to create a sketch (a little computer program) that you upload to the Arduino board. The sketch tells the board what to do [9].

After installing both *FTDI Driver* (for port detection) and Arduino IDE, Arduino programs - called sketches - can be uploaded to the board. The IDE translate sketches into *C language* and is passed to the *avr-gcc* compiler in order to be understood by the microcontroller [9].

III.5. X-Bee/ZigBee Protocol

III.5.1. Overview

X-Bee modems are one of the easiest ways to create a wireless point-to-point or mesh network. XBee RF modules are embedded solutions providing wireless end-point connectivity to devices. These modules use the IEEE 802.15.4 networking protocol of ZigBee for fast point-to-multipoint or peer-to-peer networking. XBee modules are ideal for low-power, low-cost applications.

XBees has a relatively fast data transfer rate up to 250 kbps to the end node with 2.4 GHz for worldwide deployment. A good feature is that it has sleep mode for energy saving.

Figure III.8: *X-bee module and its USB Adapter.*

III.5.2. RF Communication

Radio Frequency (RF) communications is based on laws of physics that describe the behavior of electromagnetic energy waves. RF communication works by creating electromagnetic waves at a source and being able to pick up those electromagnetic waves at a particular destination. These electromagnetic waves travel through the air at near the speed of light. The wavelength of an electromagnetic signal is inversely proportional to the frequency; the higher the frequency, the shorter the wavelength.

The RF communication system then utilizes this phenomenon by wiggling electrons in a specific pattern to represent information. The receiver can make this same information available at a remote location; communicating with no wires.

III.5.3. ZigBee Protocol:

Zigbee is an adaptation of the IEEE 80.15 low-data rate WPAN standard. The technology came about as an alternative to Bluetooth and WiFi networking. Unlike Bluetooth and Wifi, Zigbee requires low data rate (from 250 kbps at 2.4 GHZ to 20 kbps at 868 Mhz).

Zigbee uses low energy consumption. In addition, these devices are low cost. Zigbee defines three difference device types: coordinator, router and end device.

Communication with X-Bee modules is done either via Arduino or via a USB dongle which is connected to a computer. The XBee module interfaces to a host device through a logic-level asynchronous serial port.

III.5.4. X-Bee Configuration

In order to configure two X-Bee modules to communicate with each, X-CTU software or Putty terminal can be used. XCTU is a multi-platform software designed to interact with XBee RF Modules through a graphical user interface.

For the sake of simplicity, we used Putty (shown in Figure III.9) to configure the three main parameters: Network ID, Destination ID, and MY Address for the two X-bees. Command lines on the CMD terminal are on Table III.2.

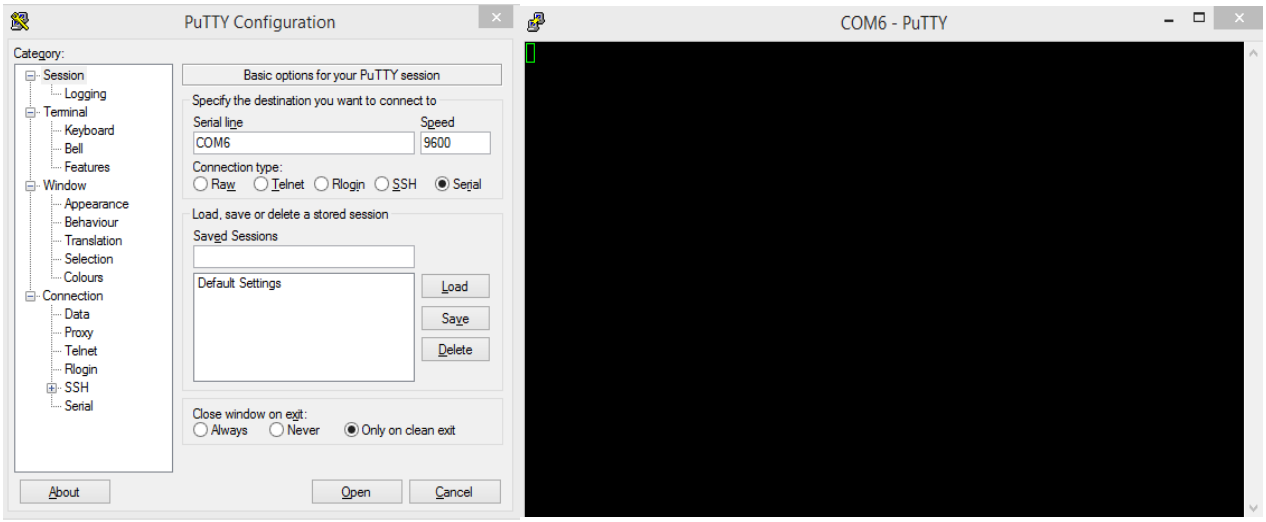


Figure III.9: *Putty Terminal for X-Bee modules configuration.*

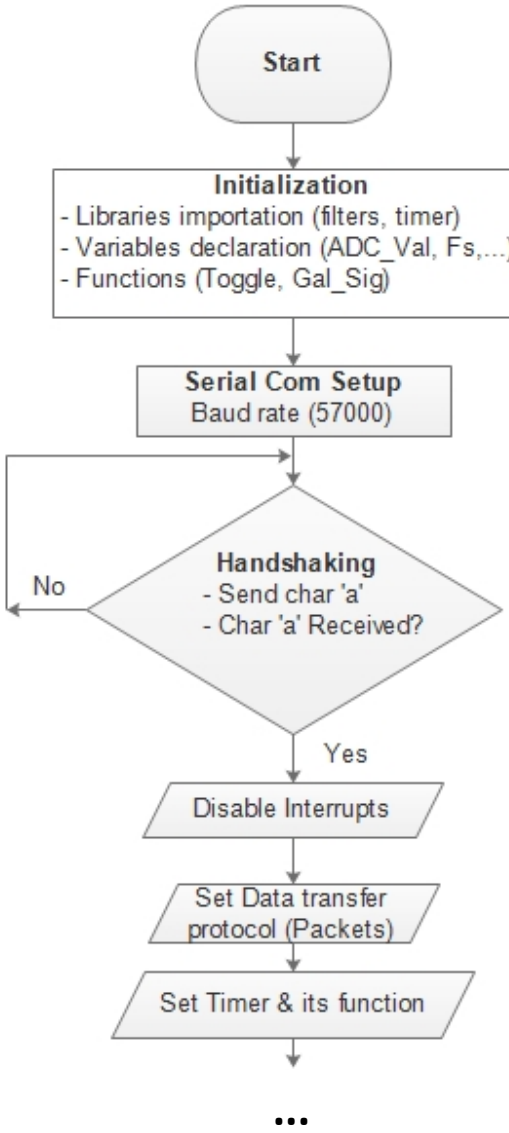
Transmitter X-Bee	*Receiver X-Bee*	*Comments*
ATID 1111, OK	ATID 1111, OK	Define network ID
ATMY 1000, OK	ATMY 1001, OK	XBee ID
ATDL 1001, OK	ATDL 1000, OK	Destination low address
WR	WR	Write configuration

Table III.2: *X-Bee modules configuration commands using Putty Terminal.*

III.6. ECG Data Acquisition

III.6.1. Hardware Programming (flowchart)

Electrodes conduct very tiny analog voltages that are fed into Olimex ECG/EKG card. The latter performs several operations of amplification, sampling, and filtering. The shield is connected to Arduino for pins compatibility. This combination is configured according to the below flowchart in Figure III.10.



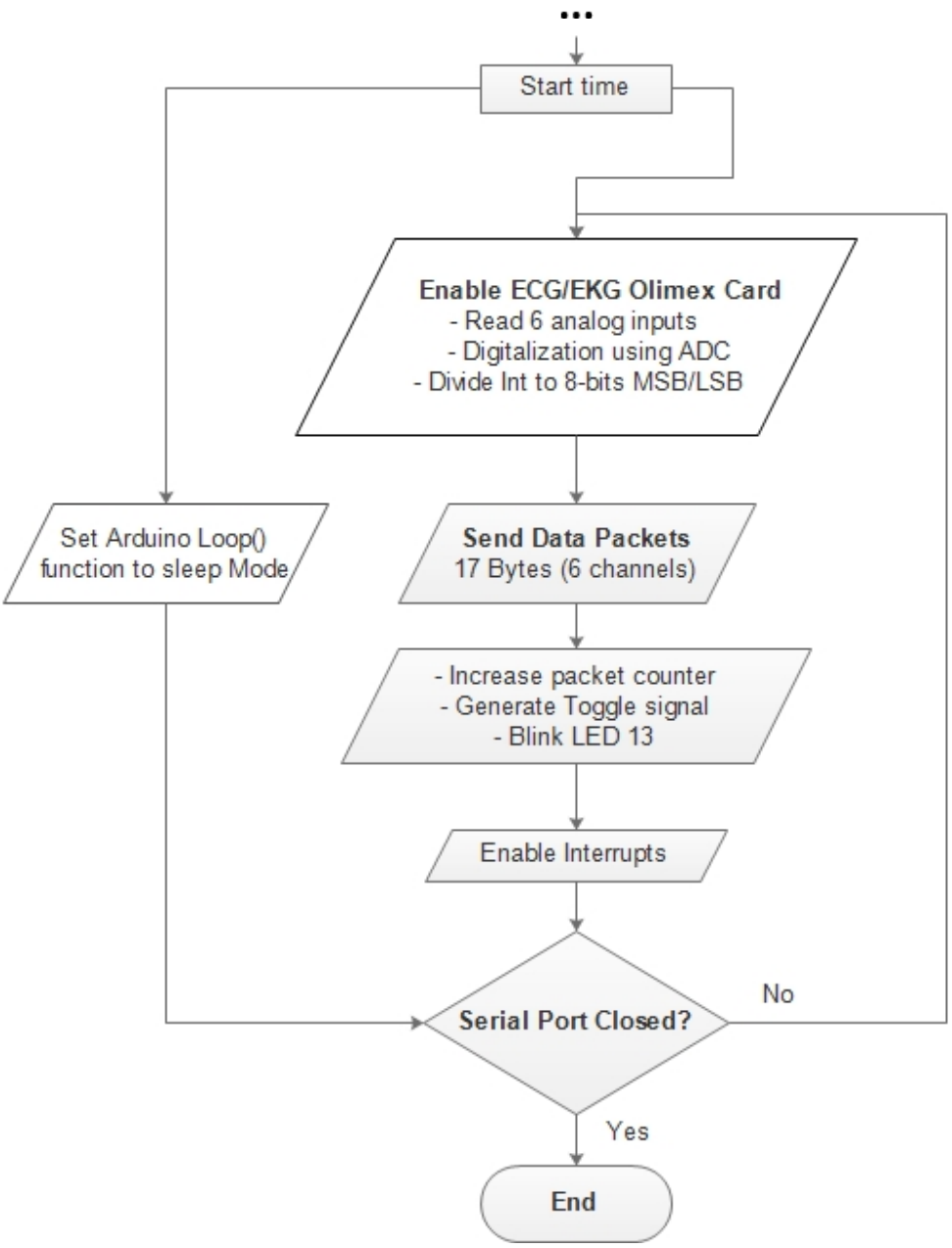


Figure III.10: *Hardware Configuration/Programming flowchart.*

At first, the program is initialized by calling libraries, functions declaration, and variables initialization to default values. A serial communication is setup at baud rate of 57600 bits per second. The serial port of a PC and the port of the Arduino needs to instantiate a handshaking to ensure the first data packet is received which is the header of the transfer protocol. The data acquisition happens according to time set for 256 samples per second. Each second, the ADC samples the six analog inputs and divide them into higher byte and lower byte (known by Most Significant Byte MSB and Least Significant Byte LSB. All bytes are stored in array of 17 bytes which is going to be transferred through serial port. Simultaneously, the Arduino loop() function is being disabled to sleep mode since all operations are being managed by the “Flexi Timer” library.

III.6.2. Serial communication (UART, X-Bee)

III.6.2.1. Basics of serial communication

Serial data transfer is when we transfer data one bit at a time, one right after the other. Sending data to the PC involves writing data to the outgoing stream buffer. Information is passed back & forth between the computer and Arduino by, essentially, setting a pin high or low.

The ATmega328 on the Arduino provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data.

To open the serial connection to the PC a call must be made to Serial.begin(speed). The parameter speed specifies the bits per second or baud rate. This must match the baud rate set in the Serial Monitor or in the terminal emulator program you are using. Similarly a call to Serial.end() will close the connection.

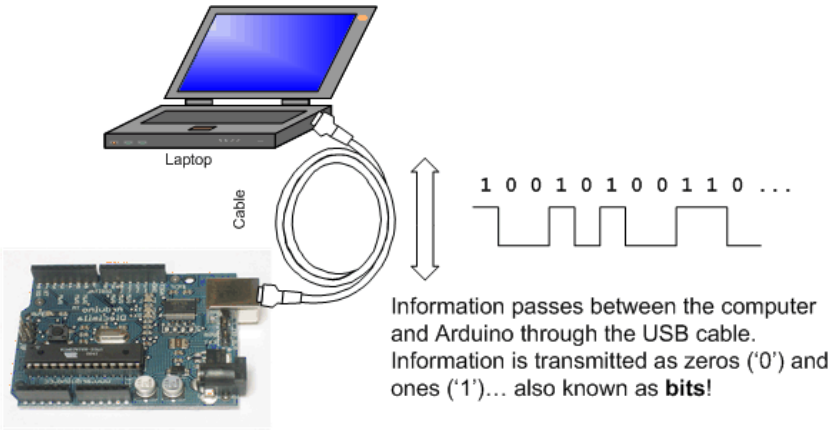


Figure III.11: *Serial Communication over USB between Arduino and PC.*

Baud Rate: Is the rate at which bits are being transmitted in terms of 0s and 1s over the serial communication protocol. Some of standard baud rates are: 9600, 14400, 19200, 28800, 38400, 57600, or 115200.

III.6.2.2. USB & X-Bee Data Transfer

The Arduino Duemilanove has 6 analog inputs (A0 … A5) which can enable cascading up to six EKG/EKG Olimex shields. Data are being sent in packet format in order to hold all the 6 ECGs being measured. A header and tail are added for synchronization that enables you identify the location of specific channel.

The collected ECG reading at certain instant is converted from analog to digital using the 10bits ADC and gets mapped between [0, 1024] range. As data bits of serial communication is limited to 8bits (1Octet/Byte), a single channel is split into two portions: high and low. Data packet used in our protocol is shown below in Figure III.12.

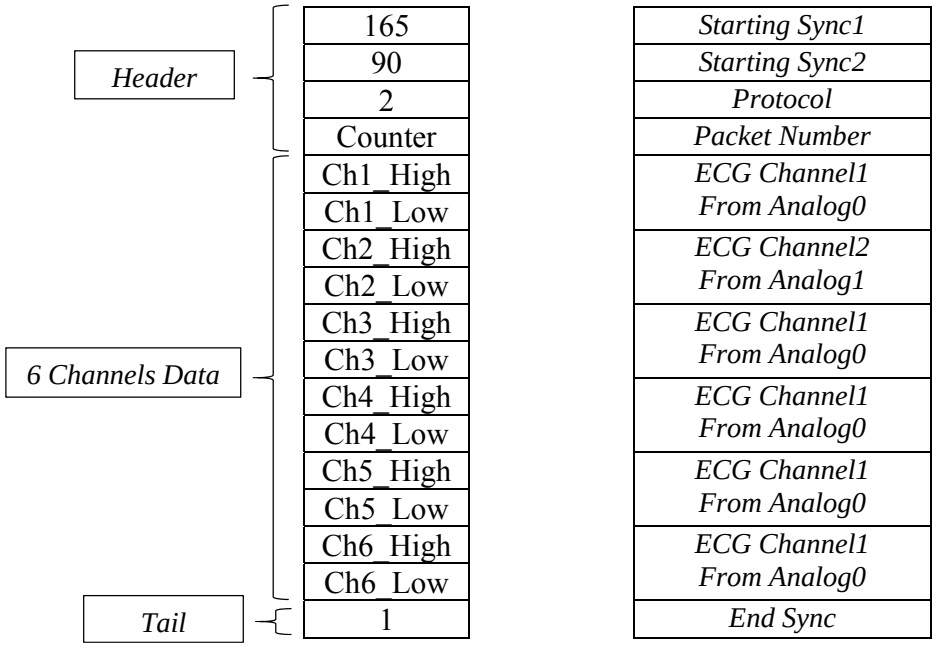


Figure III.12: *Data packet of 6 ECG channels.*

Baud rate calculation:

Acquired Bits/Second (ABS) = *Sampling Rate* X *(N° of Bytes per Packet)* X *(8 bits)*

ABS = 256 (samples per second) X 17(bytes per packet) X 8(bits per byte) = **34816 bits/s**

The baud rate set on the hardware should be greater than the acquired bits per second in order to be able to transfer all data without delay. Hence, the selected baud rate which is bigger than ABS is **56700 bps**.

III.6.3. ECG Reading

III.6.3.1. Packets reconstruction

The program on the hardware continues to execute synchronously with timer at sampling rate of 256Hz. This means having 256 packets each second, and each packet holds 6 channels data. Once the serial port on the computer is open, a handshake is established and data stream begins to flow into the computer's allocated memory or what is known as the buffer.

Several tested commands are presented in the Table III.3 below in order to perform correct readings of data bytes. We tested several commands of sending the data on the hardware and in parallel we used different function on Matlab to read them. Our observations are stated in this table:

C command on Hardware	***Data Type***	***Reading on Matlab | Comment***	***Desirable?***
Serial.write (*ADCVal*)	int16	Received ASCII symbols	No
Serial.println(*ADCVal*)	int16	Good for Matlab	OK
Serial.println(*ADCVal*,HEX)	int16	Numbers stops at 9	No
Serial.println(*ADCVal*,HEX)	int16	fscanf(s,'%x') on Matlab	Ok
Serial.println(*ADCVal*)	Char	Unread format problem	No
Serial.write(*ADCVal*)	Char	fread(s,64,'uchar') fread(s,512,'float32')	No No
Serial.write(*ADCVal*)	Byte	fread(s,'%d')	Yes

Table III.3: *Data transfer alternatives and readings testing.*

The buffer size which is the number of memory locations assigned for the data stream on the RAM is crucial. The table below presents the issues encountered to set it:

Buffer Size	***Observation***
s.InputBufferSize not set (default 512)	Counter does not continue
s.InputBufferSize =1024	Same issue
s.InputBufferSize =17000 (i,e: 1k Packets)	ECG getting cut at 1650th sample
s.InputBufferSize =170000 (i,e: 10k Packets)	OK Up to 2560 samples of raw ECG
Conclusion: The maximum allocated buffer size, the more ECG length can be acquired	

Table III.4: *Buffer size testing on Matlab to store coming data.*

Data packets needs to be read and reconstructed from binary received format (or bytes). The fread function on Matlab enables reading the value in decimal (int8) where the maximum value is 255. Hence, an easy way to combine high and low is to multiply the MSB by 256 and add the LSB.

```
while(i<=LoopNum)

        %% Get the data from the serial object
        %tic
Packet = fread(s,17);
DecimalDataCh(i) = (Packet(1+4+(Channel*2-2)))*256 + (Packet((1+5+(Channel*2-2))));

        %Increment the counter
        i=i+1;
        %toc
        %% Plot Data
end
```

Alternatively, data can be read as long vector, then converted to *17 X Length* Matrix. The matrix is converted into binary, concatenates the corresponding functions to the desired ECG Channel and then converted back to decimal between 0 and 1024.

```
function [DecimalData] = GetChannelDataDec(Matrix,Channel)

%% Conversion of Data & Concatenation
%StartingMatlab = 1;

High = decimalToBinaryVector((Matrix(1+4+(Channel*2-2),:)), 8, 'MSBFirst');
Low = decimalToBinaryVector((Matrix(1+4+1+(Channel*2-2),:)), 8, 'MSBFirst');

        BinaryData = [High, Low];
        DecimalData = binaryVectorToDecimal(BinaryData);
end
```

The resulting ECG is stored as a vector in *.mat format.

III.6.3.2. Monitoring Alternatives

III.6.3.2.1. DAQ on Matlab

Matlab reads data from the buffer of specifically allocated size in the Random Access Memory. Readings from buffer happen iteratively using standard loops at finite number of samples to be read.

Basically, ECG can be acquired **offline** without monitoring by direct reading from the buffer and plotting the full vector at the end (as show in Figure III.13).

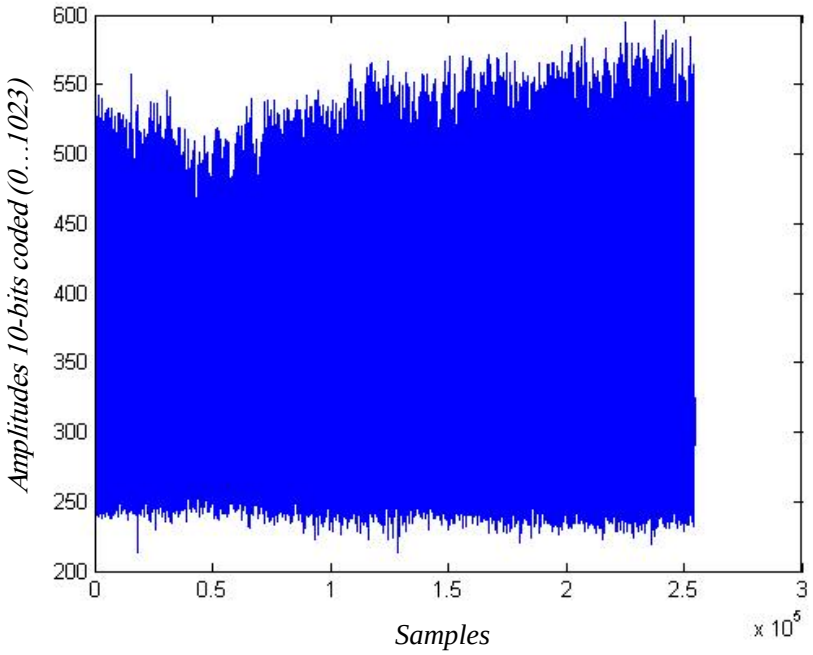


Figure III.13: *Offline Acquisition results from Raw ECG/EKG using Matlab.*

ECG can be plotted on a stripchart (see Figure III.14) but not in real time. All variables are initiated and serial port opened. A timer is set to perform data readings from the buffer at a compatible speed of the baud rate. However, any additional command in the loop causes an accumulated delay which will make the ECG delayed from the one acquired at certain instant.

The Timed stripchart code is as follows:

```
%% Initialize a stripchart
        Fs = 256;
        clf
        AxesWidth = 6;          % Axes Width (s)
        stripchart(Fs,AxesWidth);
    %% Setup a timer
% The timer has a callback that reads the serial port and updates
% the stripchart

t = timer('TimerFcn', 'getData(s,Fs)', 'Period', 1/256);
set(t,'ExecutionMode','fixedRate');
start(t)
```

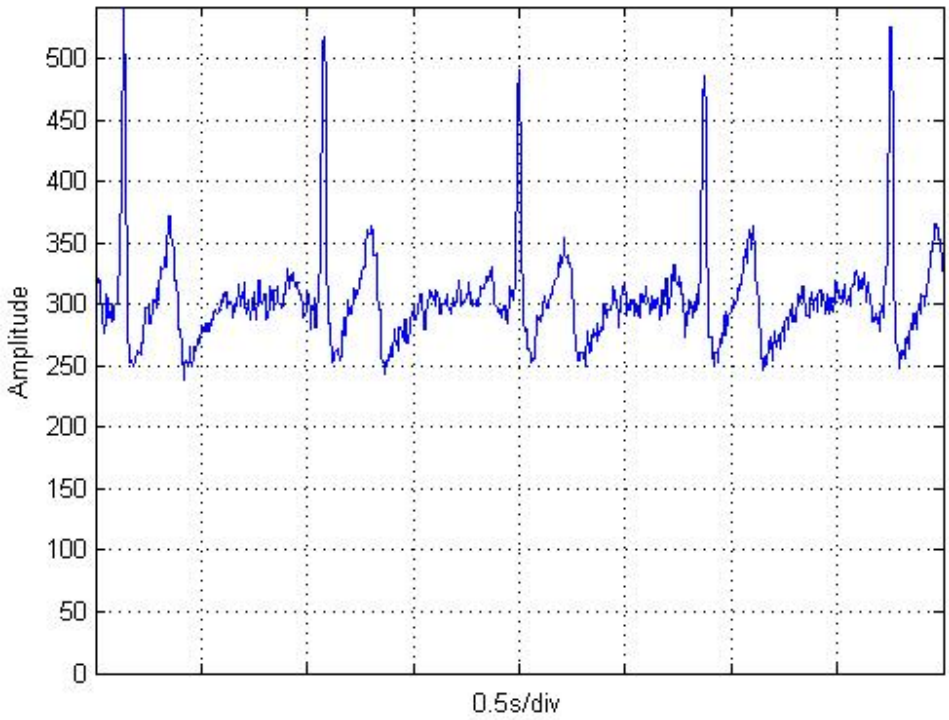


Figure III.14: *ECG monitored on stripchart on Matlab.*

Matlab delays the ECG when plotting live due to its time to perform readings from buffer, instantiate figure, and plot the acquired sample and all previous samples. This causes cumulative delay which is not desirable. Hence, the best way to form a database is to acquire the ECG offline and store it as vectors on Matlab. Another developed application just for the real-time monitoring is through LabVIEW in the following point.

III.6.3.2.2. LabVIEW Application

For the sake of monitoring, an alternative solution is provided since Matlab is not dedicated for real-time hardware interactions (for our case).

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a programming environment in which you create programs using a graphical notation. Programs that take weeks or months to write using conventional programming languages can be completed in hours using LabVIEW because it is specifically designed to take measurements, analyze data, and present results to the user.

Programs in LabVIEW have *.VI format. It is composed of two main parts: Front Panel which is the user interface and Block Diagram which is the graphical notation design. The figure below shows the customized designed interface used for ECG monitoring in our case, where its block diagram is provided on Figure III.16 in the next page.

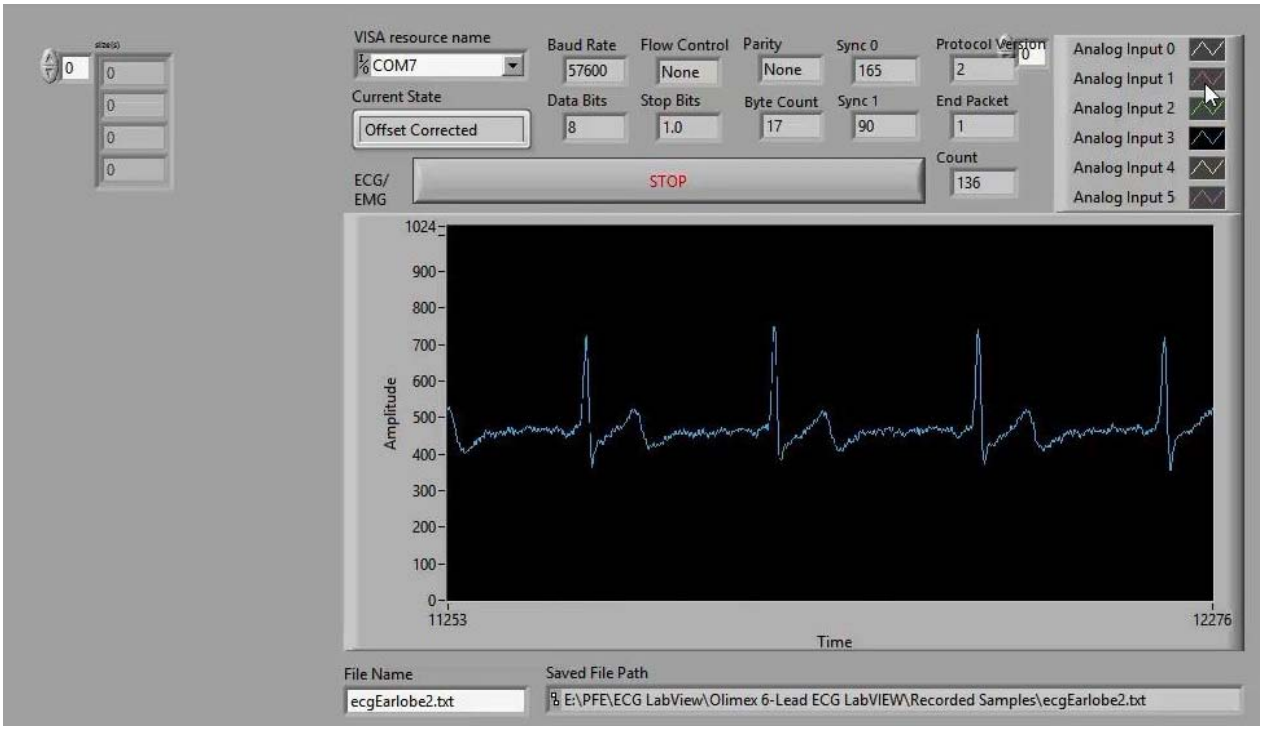


Figure III.15: *ECG acquisition Graphical User Interface using LabVIEW.*

III.6.3.3. ECG Saving for Database

At the end of acquisition process, several recordings for different people were performed in order to form a customized database. Ten ECG recordings were taken from 10 people for a duration of 12 minutes. ECG recordings can be saved in two different formats: *.CVS (Excel) or *.Mat (Matlab). Out of twelve minutes, 10 are used for the training (machine indoctrination/learning), while 2 minutes are used to test the biometric algorithms which we will be introducing in the coming chapter.

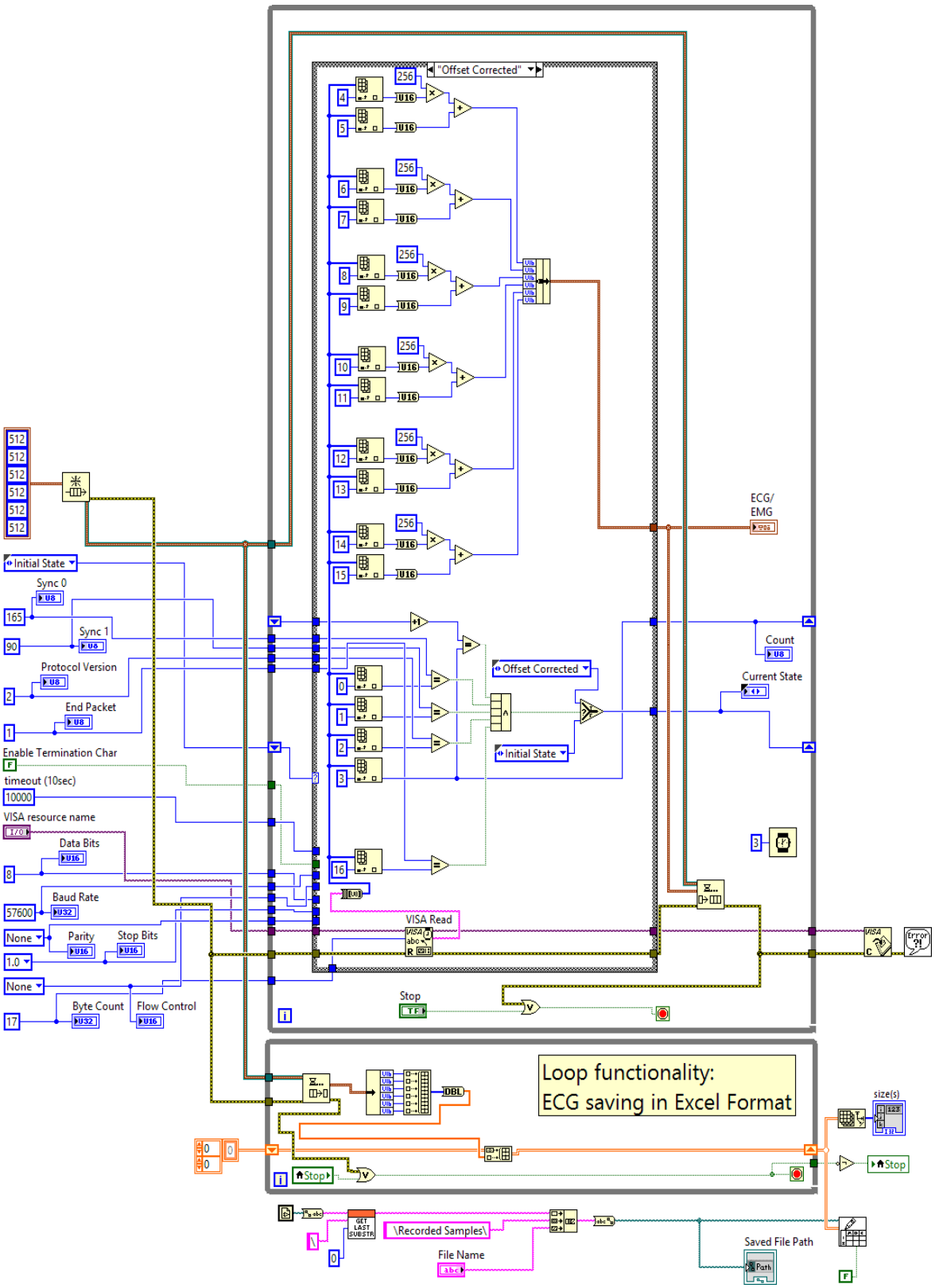


Figure III.16: *Block diagram of the LabVIEW monitoring application.*

Chapter IV:
Identification Realization and Results

IV.1. Methodology and Implementation Overview

For the sake of credibility, our algorithm is implemented using MIT-BIH Arrhythmia database and then to our customized acquired database using the hardware described in the previous chapter.

IV.1.1. MIT-BIH Arrhythmia Database:

Massachusetts Institute of Technology created a database of ECG records called MIT-BIH Arrhythmia Database in 1980 in Beth Israel Hospital. The MIT-BIH Arrhythmia DB contains 48 ECG recordings where each is half-hour excerpts of two-channel ECG recordings. These DB Materials are obtained from 47 subjects studied by the Arrhythmia Laboratory at BIH Hospital in Massachusetts. Twenty-three recordings were chosen randomly from a set that consists of 4000 EKG recordings 24 hours each. This set was collected from a hybrid population of inpatients and outpatients at BIH, about 60% and 40% respectively. The twenty-five remaining records were selected from the same previous set to include significant arrhythmias that would not be very clear in any random sample.

Each patient record in the MIT-BIH database, labeled 100 to 124 and 200 to 234, the sampling rate of the recordings is 360 samples/second with a resolution of 11-bits.

IV.1.2. Local Database (MyDB):

Our database consist of 10 ECG recordings of each 12 minutes (10 for training and 2 for Testing). These recordings were collected from 10 different volunteering students at IGEE institute. Each recording is collected from the hardware of 10 bits resolution under the name of "RVec_StudentName.mat". This raw data is reformulated manually to "id.mat" file which consists of name, image_name, two portions of divided recording (training ECG.mat and ECGTest.mat). The sampling frequency of our database is 256Hz. For further information about sampling refer to **Appendix III** at the end of this report. Below Figure IV.1. shows our database:

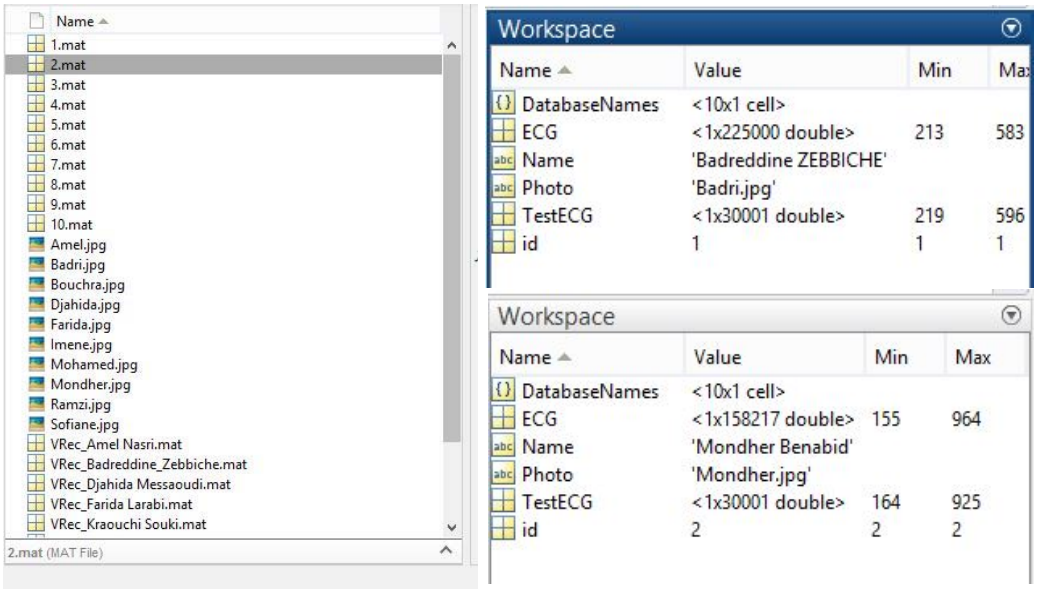


Figure IV.1: *MyDB snapshot from the workspace and directory.*

Once the algorithm is properly working with high accuracy as we explained the full process of identification in chapter two. The time to our database to be used for identification. The following organigram explains the methodology of our used process to identify individuals.

The full process in detailed way is explained in the coming points.

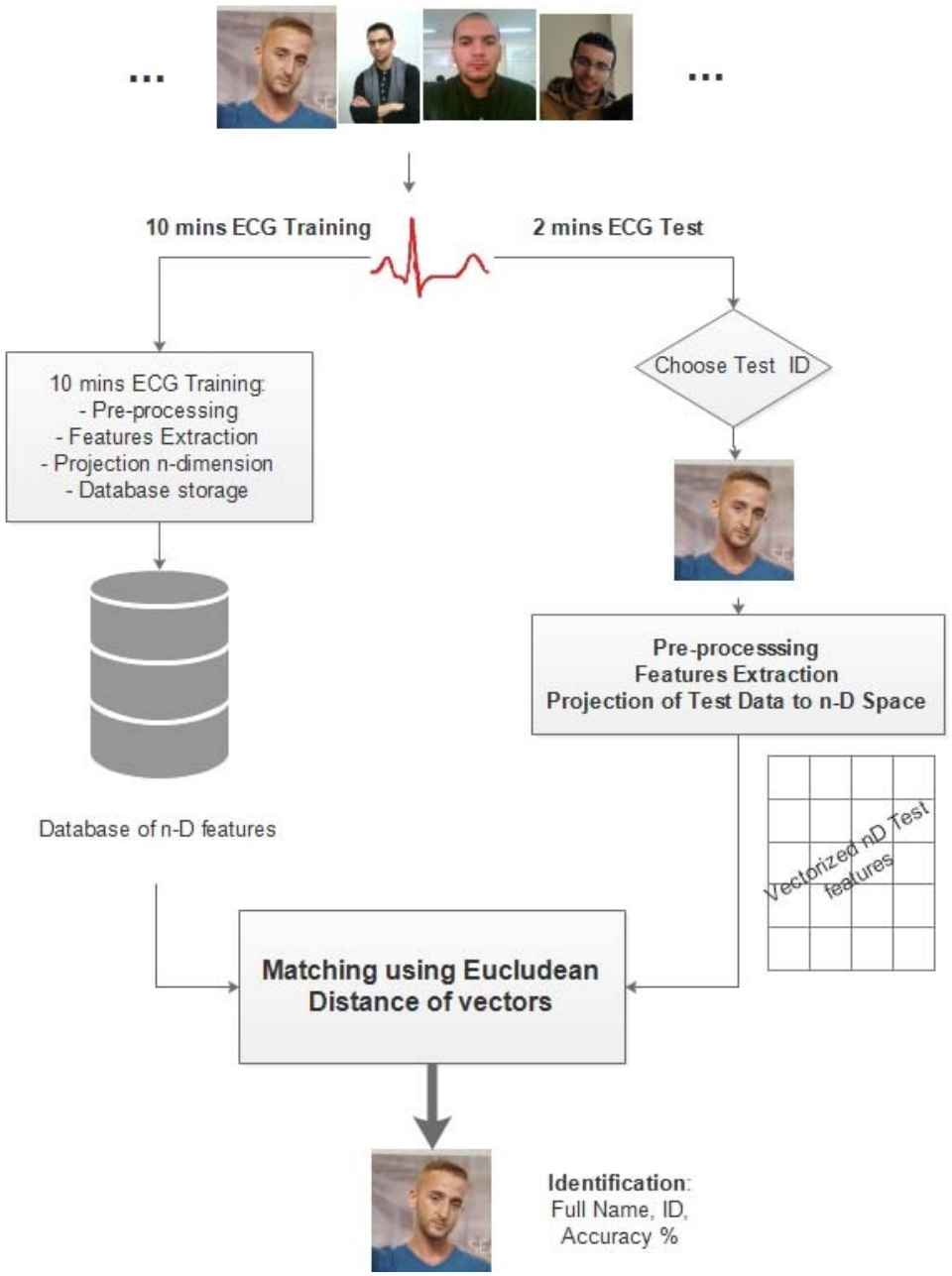


Figure IV.2: *Methodology process of individuals' identification.*

Note: *The algorithm was implemented using Matlab R2013a (See Appendix) on i3/4GB Ram PC. Hence, all results related to execution timing of the algorithm are relative to this PC's performance.*

IV.2. ECG Pre-Processing

ECG Pre-processing consist of de-noising and smoothing. The acquired ECG vector consist of noises distorting the general form of the ECG. The power line interference is narrow-band noise centered at 50 Hz (or 60 Hz) with a bandwidth of less than 1 Hz. The software scheme is more powerful and feasible for offline ECG signal processing. In terms of frequency, noises are categorized into high and low frequency noises. The modulation is low frequencies while baseline wanders and artifacts are of higher frequencies. We apply one of the following methods to de-noise and smooth the ECG signals where results are shown in figure below:

a. Applying HPF and LPF in order to smooth the signal and remove the modulation. Alternatively, it would be better to use a BPF (Butterworth):

```
ftype = 'band';
[b,a]=butter(2,[.008334,.99166]);
NonModulated = filter(b,a,signal);
corrected = NonModulated;
```

b. Apply FFT, equalize the high and low frequencies to zero, and apply iFFT:

```
ecg = signal;
samplingrate = 256;
fresult=fft(ecg);
size(fresult)
fresult(1 : round(length(fresult)*3/samplingrate))=0;
fresult(end - round(length(fresult)*3/samplingrate) : end)=0;
corrected=real(ifft(fresult));
```

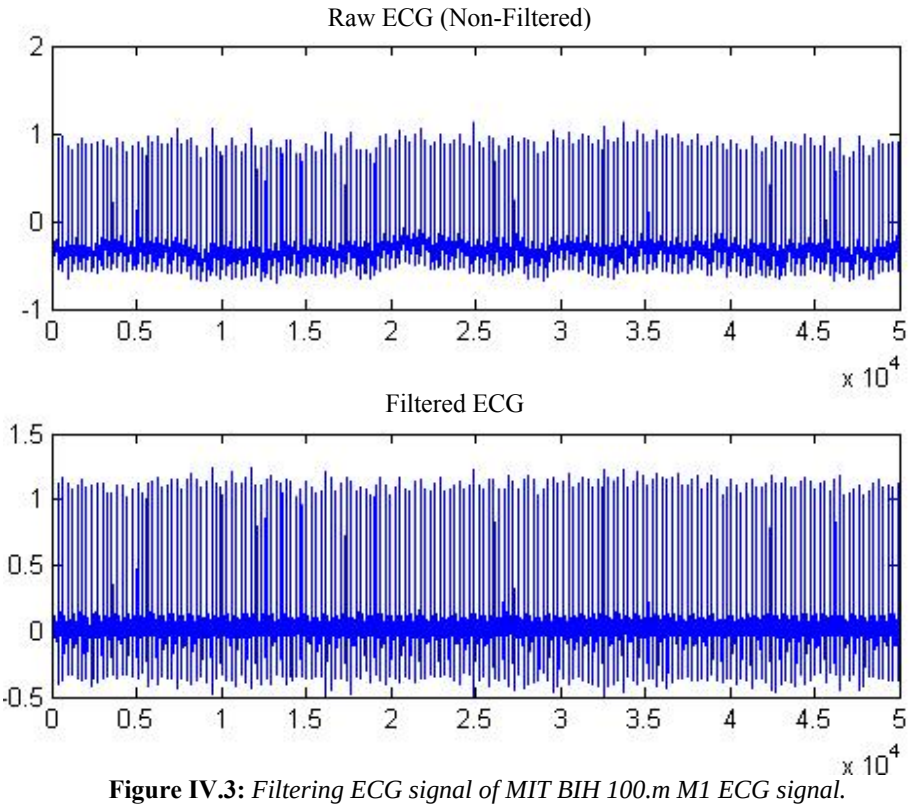


Figure IV.3: *Filtering ECG signal of MIT BIH 100.m M1 ECG signal.*

IV.3. Features Extraction Implementation

IV.3.1. Peaks Detection

Peak localization is an important step before proceeding further identification steps. As discussed previously about Pan Tompkins Algorithm on Chapter II, which will be applied directly to our signal, detects the peaks as well as providing the indices of each peak.

On MATLAB, the "*pan_tompkin.m*" file consists of the function declaration for direct use:

function [qrs_amp_raw,qrs_i_raw,delay]=pan_tompkin(ecg,fs,gr)

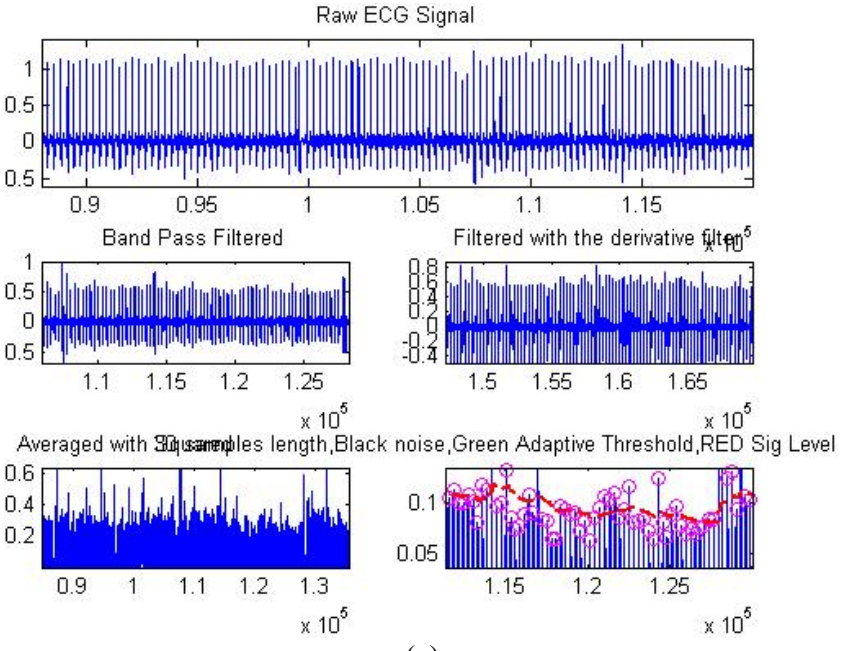
The function accepts three arguments as inputs and output three parameters depending on the need of use:

[PeaksDetected,Indexes,Delays] = pan_tompkin(FilteredECG,360,1);

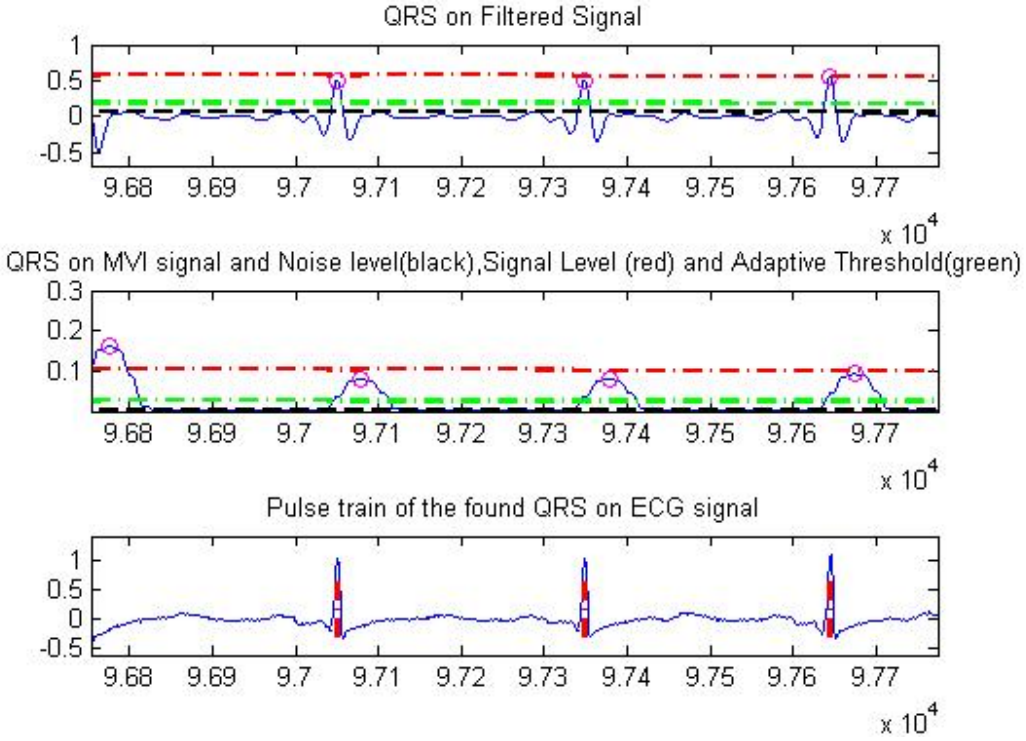
Inputs	Outputs
• ecg : raw ecg vector | 1D Signal • fs : sampling frequency • gr : flag to plot or not plot (Set/Reset to 1 or 0)	• qrs_amp_raw : amplitude of R waves amplitudes • qrs_i_raw : index of R waves • Delay : number of samples which the signal is delayed due to the filtering

Table IV.1: *Inputs and Outputs for direct use of Pan Tompkins Algorithm.*

Figures IV.4 (a) and IV.4 (b) illustrates visually the steps of R peak detection using Pan Tompkins Algorithm.



(a)



(b)

Figure IV.4: *Pan Tompkins' peak detection algorithm applied to 100.m MIT BIH.*

IV.3. Framing

A frame is a QRS complex of certain length centered and the previously detected peaks using Pan Tompkins Algorithm. The idea of framing is to create QRS complexes by applying a window of 256 (in order to cover QRS complex) width and concatenate all the constructed frames.

After the localization of all peaks' indices, we take 127 samples before the peaks and 128 samples after and store them in "frame" variable. In fact, this frame is a one-dimensional vector (a column or row). Iteratively, looping the same process over all the peaks and concatenating the created columns will construct a two-dimensional matrix of size (Window Size) x (number of detected peaks or Number of Frames). Using mesh plot on MATLAB, it yields the 3D plot shown in the Figure IV.5 below.

MATLAB Code:

```
%% Frames Creation

Matrix = [];
    for j = 2:length(Indexes)-1
        frame = FilteredECG(Indexes(j)-127:Indexes(j)+128)';
        %frame = frame.*window(@hann,256);
```

```
        switch jWin
            case 1
           frame = frame.*window(@hann,256);
            case 2
           frame = frame.*window(@hamming,256);
            case 3
           frame = frame.*window(@blackman,256);
        end
       Matrix = [Matrix frame];
    end

figure
mesh(Matrix')
title('Framed ECG Signal in 3D using Mesh in MatLab')
```

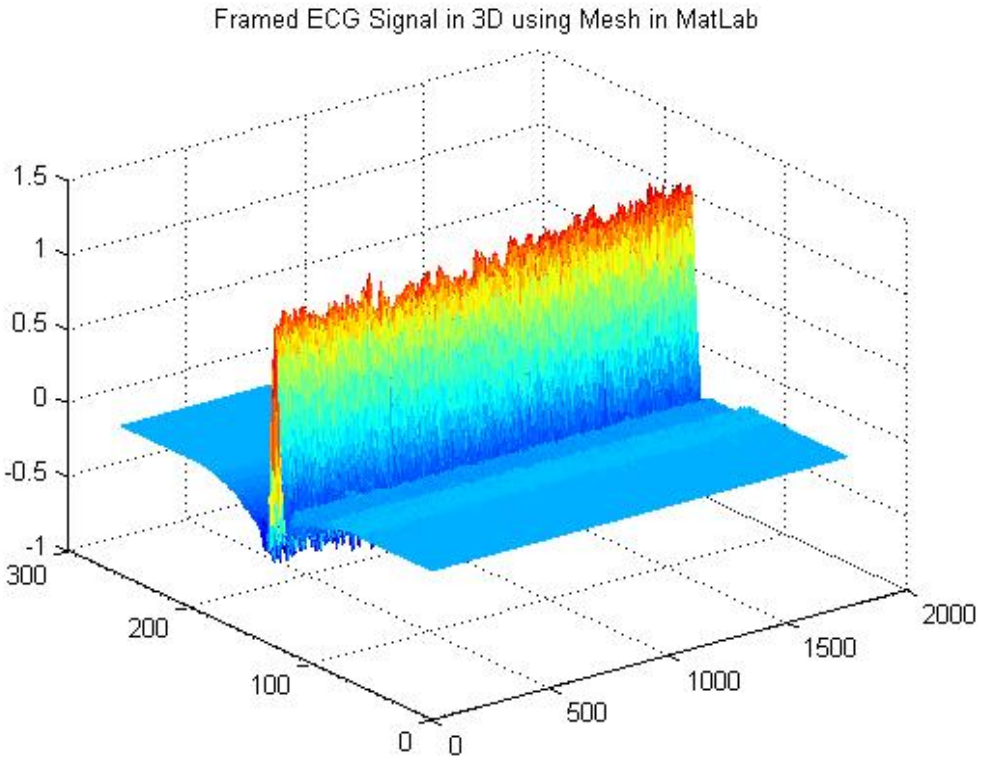


Figure IV.5: *3D mesh of concatenated ECG frames.*

The reason behind applying windows (Hann, Hamming, or Blackman) is to provide a statistical stability to the mesh matrix as well as the concentration of energy in the main lobe for FFT application.

Statistical parameters, such as mean and variance, needs to be more stable for proper space formation.

IV.3.3. Spectral Information

Concatenated frames from previous point are a two-dimensional matrix on Matlab. Spectral features are extracted in this case due to the complexity of differentiation in the time domain. FFT is applied to the matrix where operations are applied to each column. A magnitude spectrum is calculated using the "abs" function and raised to the power of 2. The visualization of the converted 3D mesh matrix is shown in figure IV.6. Discrimination is more visible between spectra.

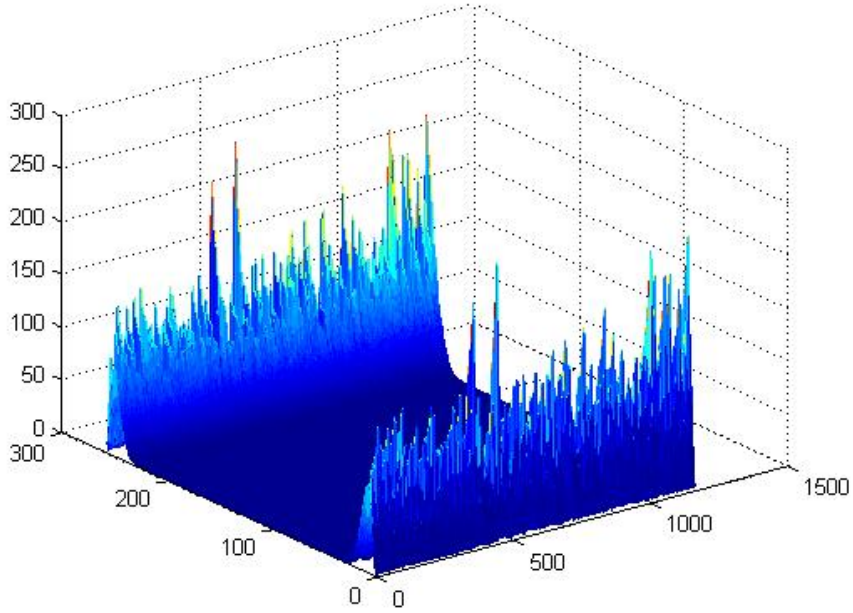


Figure IV.6: *MIT BIH 100.mat magnitude spectrum.*

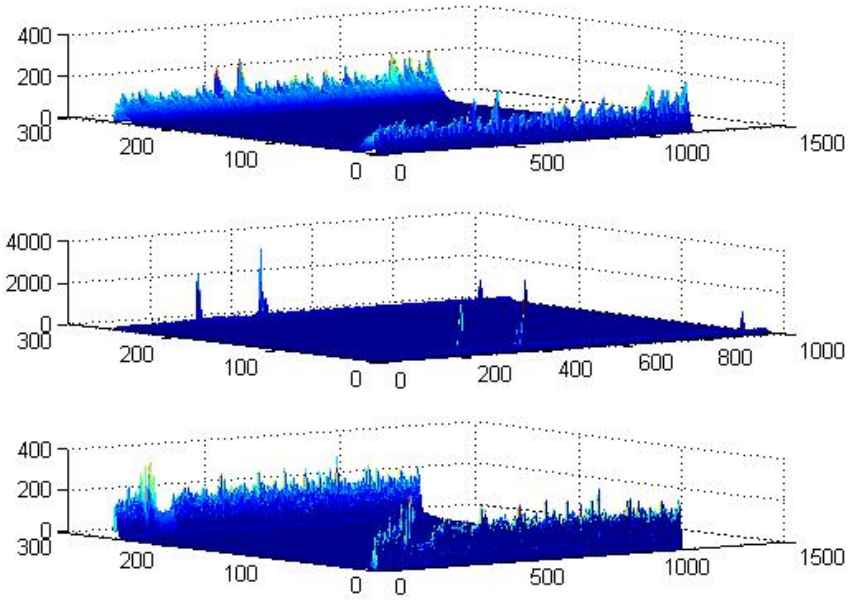


Figure IV.7: *Magnitude Spectra of different ECGs on MIT BIH DB.*

IV.4. Proper Space Formation

We intend to increase dimensionality to 2^8 which corresponds to the frame size. That is to say that a single frame will be considered as a point in the new proper space. This point is represented by vector of *256* components on *256* axes. Matching is performed on these vectors in the proper/features spaces using Euclidian Distance.

As inferred from the theoretical part on chapter two. The following steps are performed to form a higher dimensionality space:

- Concatenate all training data to form big matrix of data for projection.
- **Label** all the training data to their IDs.
- Compute the mean of all data and subtract it for data translation
- Compute the covariance matrix of scattered training data matrix.
- Compute eigenvalues and their corresponding eigenvectors.

On Matlab, we implemented the proper space form as follows:

```
%% All Data Concatenation
A = [];
TrainingLabels = [];
%for i = 1:10 %Just testing with 5,10
for i = 1:length(MatrixTraining)
    id = i;
    A = [A MatrixTraining{i}];
    %Identity = id*ones(1,length(MatrixTraining{1,i}));
    TrainingLabels = [TrainingLabels id*ones(1,length(MatrixTraining{1,i}))];
end

%% Reference translation to gravity center (mean)
VectorMeans = mean(A,2);
ReplicatedVectorMeans = repmat (VectorMeans,1,size(A,2));
CenteredMatrix = A - ReplicatedVectorMeans;

%% Principal Components Selection
CovACent1 = CenteredMatrix * CenteredMatrix';
%[EigVects1,EigVals1] = eig(CovACent1);
[EigVects,EigVals] = eig(CovACent1);
```

IV.5. Training Phase

IV.5.1. Classes Labeling

Each set of specific features belonging to specific person is grouped into class or model which is marked with "*Labels*". These labels help for comparison between training and the testing samples coming from the testing phase explained in the following points.

As explained in the code above, labels are put during the step of proper space formation.

IV.5.2. Features Projection

Before taking data to the new high dimensionality space, base vectors are computed by multiplying the scatter matrix transpose by the eigenvectors. These base vectors needs to be normalized before projection.

The data are then projected by multiplication to the normed base vectors. The following implementation demonstrate the normalization and projection:

```
%% Normalize all Base vectors of the newly proper space
NormedBaseVectors = [];
for i = 1:length(EigVects1)
    BaseVector = EigVects1(:,i);
    NormVi = norm(BaseVector);
    V1New = BaseVector/NormVi;
    NormedBaseVectors = [NormedBaseVectors V1New];
end
disp('Proper Space Formed');

%% Components calculation on the proper space
for i=1:length(CenteredMatrix)
    for j=1:256

ProperECGTraining(j,i)=product(CenteredMatrix,i,NormedBaseVectors,j,256);
    end
end
disp('Training data projected on the Proper Space');
```

The process of data projection means that frames (QRS complexes) are going to be converted into points in the new proper space where the basis is of length 256.

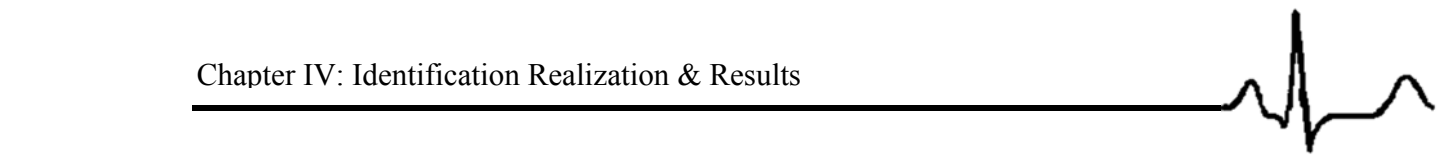
IV.6. Testing Phase/Identification

IV.6.1. Test Data into Proper Space

In testing phase, comparison is done in the same framework. Hence, all testing ECG portions go under the same scenario:

- Pre-processed: De-noising and smoothing.
- Features Extraction: Peaks detection, framing, and spectral information extraction.
- Projection on Proper space: multiplication by normalized base vectors.

Once testing features are on the proper space in form of vectors of 256 dimension, PCA classier affects them to its corresponding nearest Euclidean distance class.



On Matlab, our testing phase of is implemented according to the flowchart below (Figure IV.8):

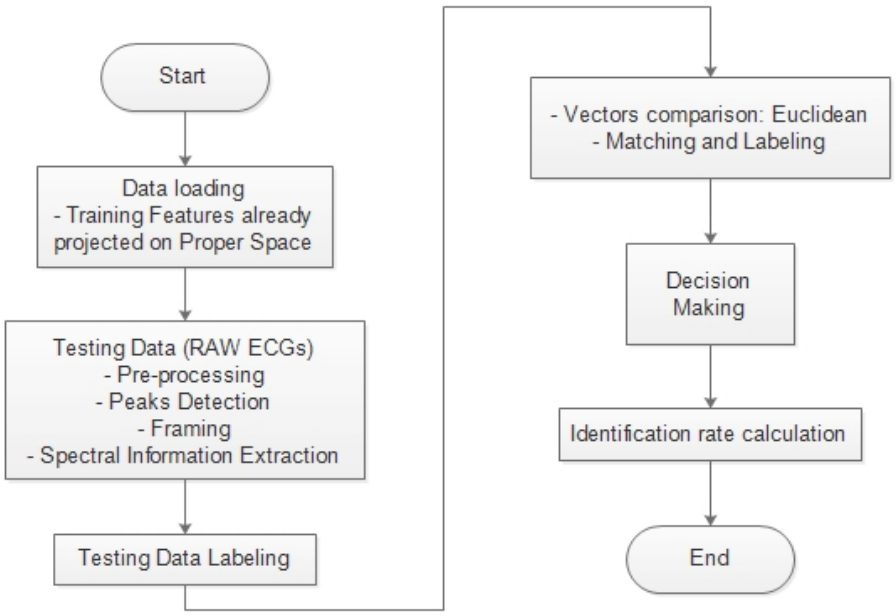


Figure IV.8: *Identification testing phase flowchart.*

IV.6.2. Dimensionality Reduction

Normed base vectors that have been extracted from eigenvectors with low eigenvalues does not have high power of discrimination and can be ignored with minor effects of data loss. The covariance matrix, which is diagonal has increasing order of eigenvalues.

In our algorithm, dimensionality reduction is left to the user to selected the number of principal components and it is implemented as follows:

```
%% Principal Components Selection
CovACent1 = CenteredMatrix * CenteredMatrix';
%[EigVects1,EigVals1] = eig(CovACent1);
[EigVects,EigVals] = eig(CovACent1);
EntryPCA = input('Enter the PCA Eigenvectors number (Max 256)\n');
EigVects1 = EigVects(:,(length(EigVects)-EntryPCA):end);
```

The number of chosen principal components affects the identification rate due to details loss. This is discussed in “Results & Discussion” section.

IV.6.3. Classification/Matching Process

IV.6.3.1. Decision making using Euclidean Distance

In our matching process, we used the simplest method of distance measurement using Euclidian distance. Training features are loaded from the database in form of vectors. Once of the testing frames arrives to the reduced-dimension proper space, each frame is compared with all training features on the database and then allocated to the nearest class label. Iteratively, the same process is applied to all testing features. The following code is the implementation of the matching process:

```
%% Distance Measurement Test & Finding the minimum Distance and Performing
Identification
disp('Distance measurements');

Distance = [];
tic

    for j=1:length(ProperECGTest(1,:))
        j
        Diff = bsxfun(@minus, ProperECGTraining, ProperECGTest(:,j));
        EucDist = sqrt(sum(Diff.*Diff));
        [mindist, FoundIndex] = min(EucDist);
        IdentifiedPerson(j)= TrainingLabels(FoundIndex);
    end

%% Identification Rate
disp('Calculating the Identification Rate');
j=1;
ind=0;
IdentificationRate=0;
for i=1:length(ProperECGTest(1,:))
    if(IdentifiedPerson(i)==LabelsTest(i))
        IdentificationRate=IdentificationRate+1;
    end
end
toc

disp('Identification Rate is found to be:');
res =(IdentificationRate/length(ProperECGTest(1,:)))*100;
```

Identification rate enhancement is one of the objectives of the project besides the minimization of execution timing. In order to compute the rate of identification (recognition), we compare between the found label of the nearest vector and the label of the testing vector. If they do match, a counter is increased. Otherwise, the matching is false of that heartbeat. The overall efficiency is given as a percentage out of all the testing frames. The above code portions shows the implementation of identification rate calculation and its resolutions.

IV.6.3.2. Demonstration

Once the final matching is done, our code shows the ID and photo of the identified individual as shown in the figure below. Once all found labels are assigned the spectral frame which we find closest using the Euclidian distance, all testing frames labels are stored in vector. A function display person is called to compute the statistical Mode of all identified heartbeats and displays photo stored on the database.

The following code is a portion of the function that returns the picture of the identified person:

```
IDPerson = mode(IdentifiedPerson);
% Printing Name or Displaying Image
X = ['E:\PFE\student_Badreddine_zebbiche\Final PFE\IGEE DB\' Photo];
%figure;
imdata = imread(X);
imtool(imdata)
```

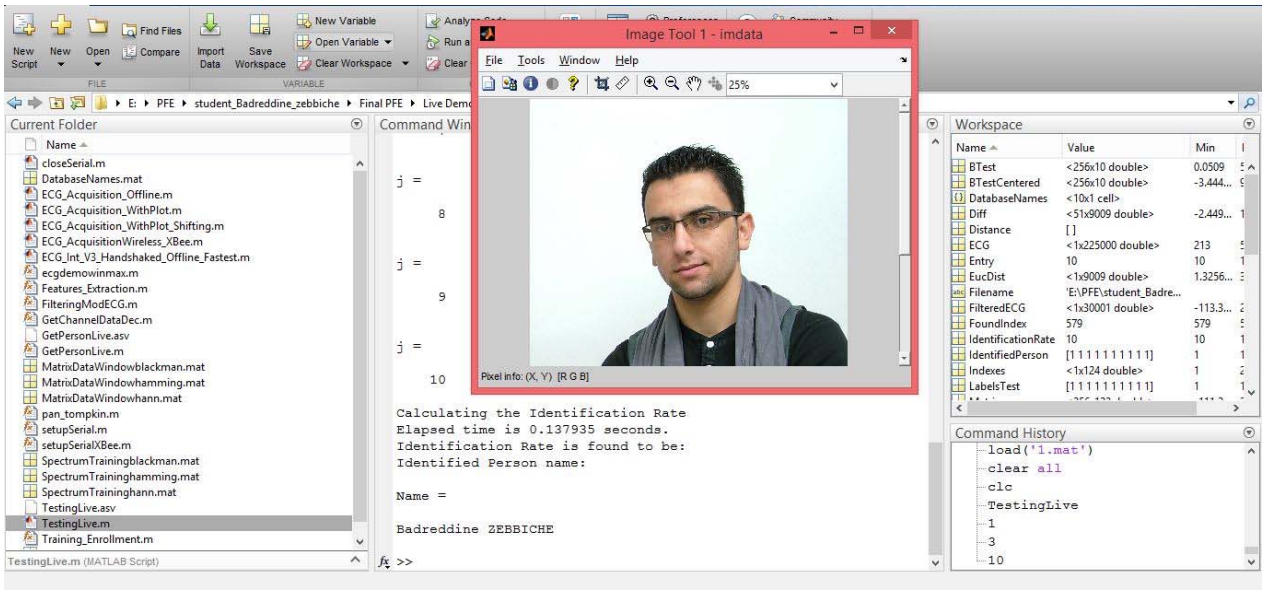


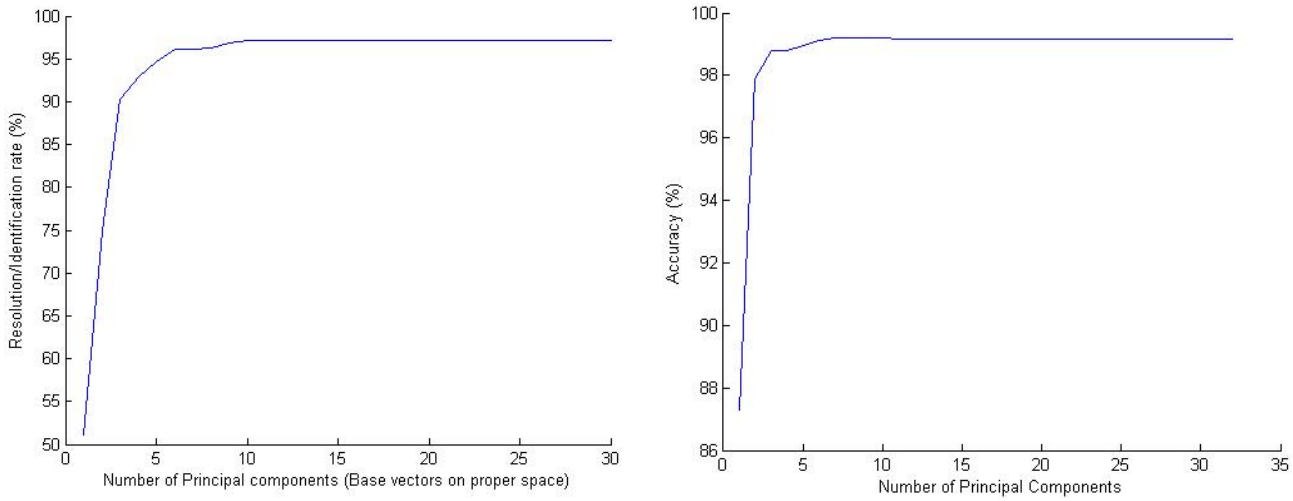
Figure IV.9: *Demonstration of recognized individual using our identification method.*

IV.7. Results and Discussions

In MIT-BIH, training portion is 2/3 of the overall length (30 minutes) while Testing was 1/3. We are interested in the **overall identification rate** of all the 48 people in the database. Similarly, our interest to test our database with the minimum dimension (Principal Components) as well as the number of heartbeats.

In order to study the effectivity of our algorithm, iterative execution of testing with variation of number of principal components is invoked to reduce the computations at certain stable accuracy of identification. Iterative process of the identification yields the overall percentage of identification (resolution).

Without dimensionality reduction (i.e.: full dimensionality of 256) and taking the full 10 minutes of testing portion MIT BIH, the algorithm takes hours of execution (refer to **Appendix I** to see results before optimization). Hence, it is desirable to study the dimension that gives you reasonable execution time at a stable accuracy with minor losses.



(a) (b)

Figure IV.10: *Identification rate results of MIT BIH Database (a) and our customized database (b) in terms of principal components number.*

The identification rate has transient and steady state as observed from Figure IV.10. The more principal components, the more efficiency of identification and stability. At the 10^{th} dimension, the algorithm yields stable identification rate. That is to say that 246 (256 -10) components are reduced. This is enormous calculations reduction on Matlab which means less elapsed timing of execution.

Fixing the number of principal components at 10 and varying the number of heart beats given to the algorithm in the testing phase, the results of the timing and identification are visualized in the following two figures:

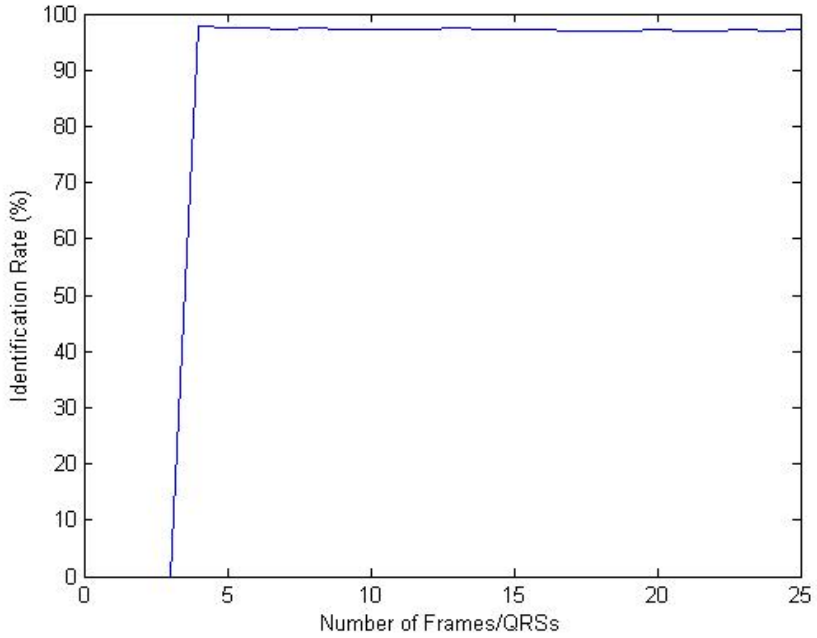


Figure IV.11: *Identification rate (%) in terms of frames/QRSs number with 10 PCs.*

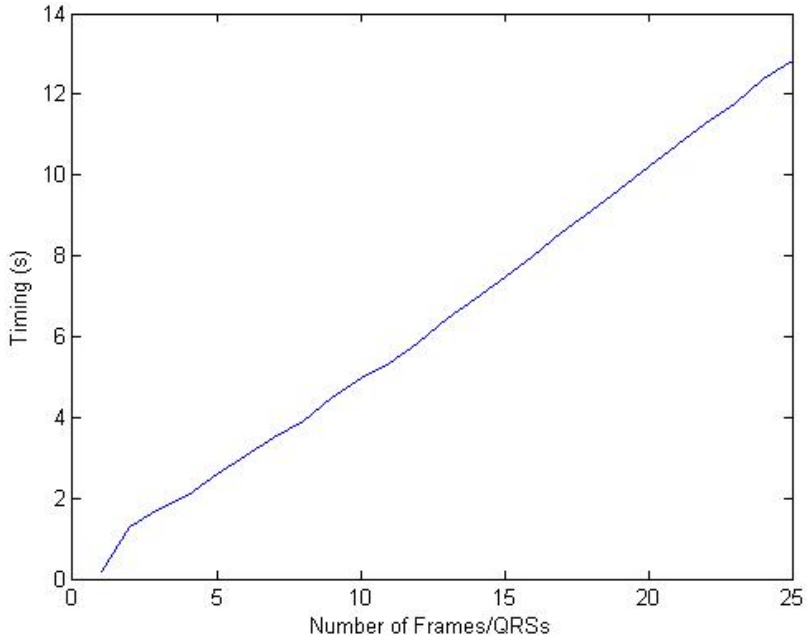


Figure IV.12: *Linear incremental of elapsed execution time with number of frames/QRSs.*

Clearly seen that starting from the 5th heartbeat, the accuracy (identification rate, or resolutions, or recognition percentage) is very significant.

Below is the tabulated data between both MIT-BIH Database and our customized database in terms of accuracy as well as the elapsed time of execution provided that 246 reduced components.

	MIT BIH Arrhythmia DB (48 ECGs)			**My Customized DB (10 ECGs)**	
QRS/Frames	**Accuracy (%)**	**Time (s)**		**Accuracy (%)**	**Time (s)**
1	N/A	N/A		N/A	N/A
2	N/A	N/A		N/A	N/A
3	N/A	N/A		N/A	N/A
4	96.9167	2.0865		87.3000	1.3122
5	97.5000	2.6061		97.9000	1.4363
6	97.5694	3.0504		98.8000	1.6894
7	97.3214	3.4900		98.8000	1.7997
8	97.3958	3.9194		98.9500	1.7820
9	97.2222	4.4934		99.1000	2.0916
10	97.0833	4.9562		99.2000	2.1279
11	97.3485	5.3601		99.2000	2.3192
12	97.2222	5.8488		99.2000	2.3136
13	97.4359	6.4409		99.2000	2.4378
14	97.3214	6.9186		99.1500	2.5155
15	97.3611	7.4688		99.1500	2.6327
16	97.1354	8.0092		99.1500	2.7485
17	96.8137	8.5767		99.1500	2.8741
18	96.8750	9.1098		99.1500	2.9876
19	96.9298	9.6339		99.1500	3.2740
20	97.0833	10.2144		99.1500	3.3453
21	97.0238	10.7627		99.1500	3.4746
22	97.0644	11.2928		99.1500	3.5269
23	97.1014	11.7827		99.1500	3.9175
24	96.9618	12.3936		99.1500	6.0401
25	97.0833	12.8436		99.1500	6.3122
Avg. accuracy	**97.2168**			**98.5205**	

Note: Average accuracy is calculated starting from the fourth frame/QRS complex

Table IV.2: *Accuracy and execution time in terms of testing heart-beats number at 10th dimension.*

Explanations and Deductions:

- Dimensionality: The more dimensionality, the greater details for comparison. Once greater details are provided, classification yields higher precision.

- Dimensionality reduction reduces timing of execution with minor data loss which affects accuracy.

- Sampling rate: Resolutions of acquired ECG signals and hardware used (sampling rate between MIT BIH and My DB) provides a better quality of signal. The more sampling rate, the more smoothed signal. Hence, further smoothing and filtering is required at the level of pre-processing.

- Size of Database: The size of the database is the number of people to differentiate between. The more classes, the harder for a recognition system to differentiate. This can be explained by the results on the above Table IV.2. Our database provided slightly high accuracy more than MIT-BIH because 10 classes are easier to classify compared to 48 classes.

- Satisfactory performance starts 5 QRS complexes and more (frames or ordinarily heartbeats) at the 10^{th} dimension.

- Timing of execution is purely relative to the computer's performance but the same pattern remain the same.

- The inconvenience of ECG as biometric model is that training needs to be conducted for long time, in different rhythms. However, the accuracy of ECG is pretty high and very significant.

- ECG can be used in high security applications: It cannot be faked like fingerprint.

- Identification can be performed in fractions of seconds to perform all calculations and vectors computations on Matlab.

Conclusion

Our work discussed here, implementation of a biometric identification system through ECG/EKG, aim was to develop a biometric system including software and hardware by applying all theoretical acquired background during the master's curriculum: Hardware (Courses: Embedded Systems, Digital Systems design), Algebraic concepts (Advanced Mathematics course), Digital Signal Processing, and Probability and statistics..

At the level of hardware, we performed tasks such as hardware programming using C, serial communication setup, wireless RF XBee module configuration. Moreover, our hardware can record up to six people simultaneously. For software part, hundreds of lines on to perform our following contributions:

- Data acquisition via Serial communication (XBee or Serial COM using USB).
- Monitoring on Matlab with shifting axes (with delay), stripchart (slight delay).
- LabVIEW graphical notation design for monitoring-oriented application.
- Signal Pre-processing (Filtering, Peaks detection).
- Classification using PCA (applying Linear Algebra concepts, Probability and statistics utilization for space formation and classification decision making).

For the classification task, we adopted the state-of-the art of one linear classifier which is the Principal Component Analysis. PCA is chosen for ease of implementation and using applied theory in terms of Algebraic analysis and statistical analysis. This methods can deliver high performance even with small number of training samples. We use the training data to make the machine learn and then we perform pattern recognition using testing portion. The matching was done using Euclidian distance. Identification rate and timing of execution are the highly under focus of the project. Results are compared between both MIT-BIH database and our database.

Ultimately, results were pretty satisfactory reaching 97%-98% using MIT BIH Arrhythmia database and MyDB respectively using the different spectral information extracted from different individuals.

Future scope:

For future perspectives, the application could be enhanced in real-time processing using one of the development languages (C#, Python…) instead of academic Matlab. Future development includes making the system autonomous instead of offline. Further work potentially takes into consideration different cardiac rhythms (running, stress…etc.) to increase the difficulty of classification.

As an expansion of the project with the same analogy of classification, detection of cardio anomalies for biomedical purposes is potentially will be the upcoming checkpoint.

References

[1] http://searchsecurity.techtarget.com/definition/biometrics

[2] Ankit Sharma, “ECG Based Biometrics Verification System Using LabVIEW”, Thapar University – India, Pages from 6 to 9, year 2009.

[3] en.wikipedia.org/?title=Circulatory_system (Definition extraction)

[4] http://www.edoctoronline.com/medical-atlas.asp?c=4&id=22168 (photo extracted)

[5] "Cardiac Muscle and Electrical Activity". OpenStax CNX: Anatomy & Physiology. OpenStax CNX. November 7, 2014. Retrieved January 2, 2015.

[6] "Cardiac Muscle Fibers". ZY 560 Mammalian Physiology. Auburn University Journal. Retrieved January 2, 2015.

[7] http://lifeinthefastlane.com/ecg-library/basics/pr-interval/ (photo extracted)

[8] BENKREIRA Amel, TALENZAR Ayacha, DELLACHE Amel, “ECG Classification Using Wavelets and Support Vector Machines”, Pages 5-9, year 2013.

[9] Dr. BENALI Radhwane, “Analyse du signal ECG par réseau adaptif d’ondelettes en vue de la reconnaissance de pathologies cardiaques“, Pages 27-33, year 2013.

[10] John Semmlow, “Biosignal and Biomedical Image Processing MATLAB based Applications – Editor Marcel Dekker, Inc.”, Page 20, 2004.

[11] Gari D. Clifford, Francisco Azuaje, Patrick McSha, “Advanced Tools and Methods for ECG Data Analysis – Editor Artech House”, P261 - P380, United Kingdom 2006.

[12] http://www.amperordirect.com/pc/help-ecg-monitor/z-ekg-purpose.html

[13] Ankit Sharma, “ECG Based Biometrics Verification System Using LabVIEW”, Thapar University – India, Page 18, year 2009.

[14] PAN.J, TOMPKINS. W.J,"A Real-Time QRS Detection Algorithm" IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, VOL. BME-32, NO. 3, MARCH 1985.

[15] P. Podder, T.Z. Khan, M.H. Khan, “Comparative Performance Analysis of Hamming, Hanning, Blackman window”, International Journal of Computer Applications (0975 – 8887) – Volume 96 – No.18, June 2014.

[16] Sophocles J. Orfanidis, “Introduction to Signal Processing”, Rutgers University, page 464, 2005.

[17] Guiqiang Pater Zhang, “Neural Networks for Classification: A Survey”, Georgia State University, Atlanta, 2000.

[18] https://en.wikipedia.org/wiki/Principal_component_analysis (Definition extraction of Principal Component Analysis).

[19] Roweis, S. T.; Saul, L. K., "Nonlinear Dimensionality Reduction by Locally Linear Embedding". Science 290, year 2000.

[20] Ali Ghodsi, "Data Visualization STAT 442 / 890, CM 462”, University lecture, pages 2 and 3, Sept. 18 and Sept.20-2006.

[21] SHIELD EKG-EMG PRO: https://www.olimex.com/Products/Duino/Shields/SHIELD-EKG-EMG-PRO/

[22] http://www.skintact.com/46.0.html

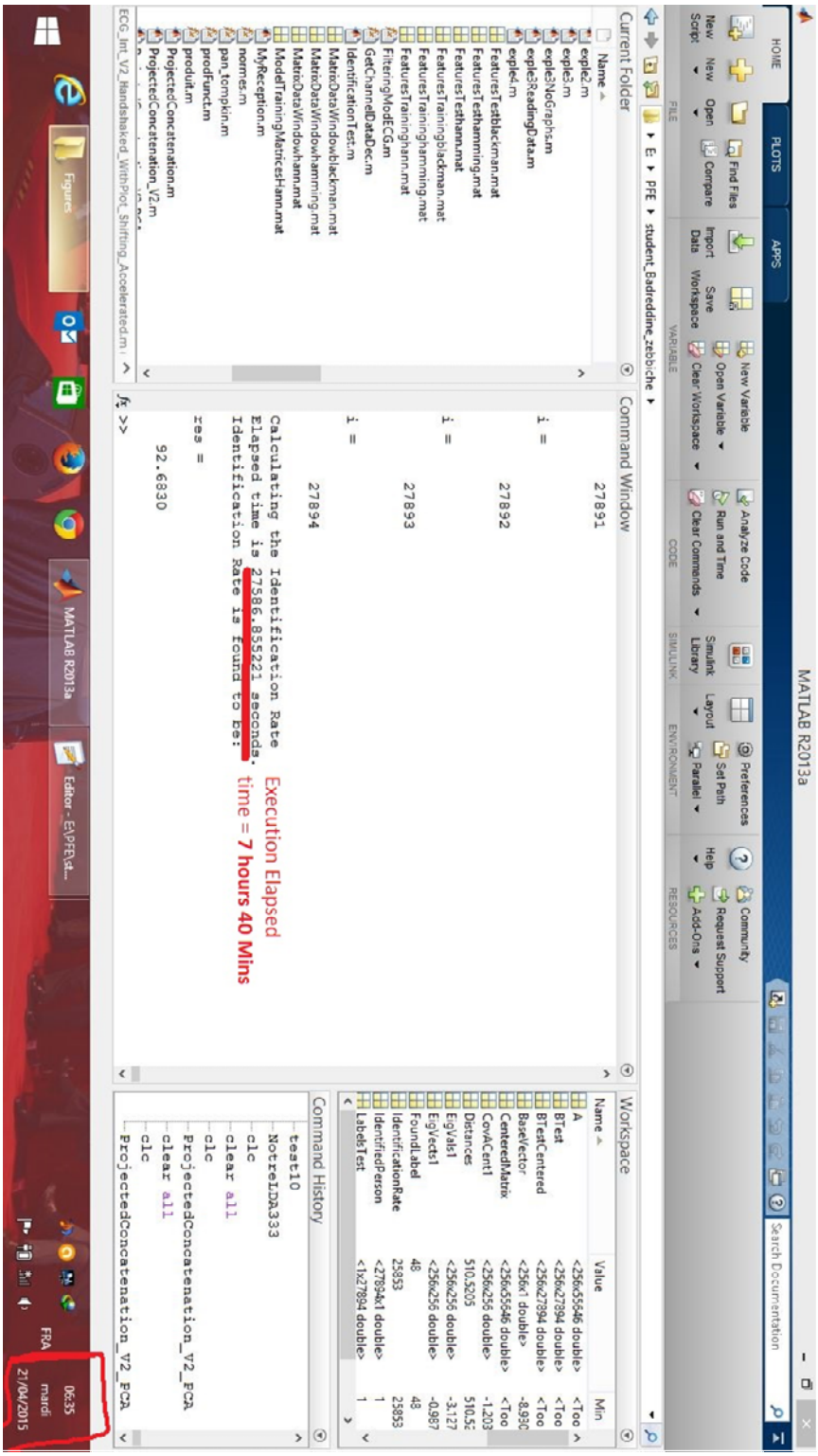
[23] www.Olimex.com

[24] Massimo Banzi – Co-founder of Arduino, “Getting Stated with Arduino” – Second Edition by O’Reilly, Page 1-17, 2011.

[25] www.Arduino.cc the official website of Arduino (photo extraction)

Appendices

Appendix I: Execution snapshot without dimensionality reduction



Appendix II: Sampling

In analog domain, signals are captured in terms of amplitude or intensity at any given time. Any analog signal is described as a function of time $x(t) = f(t)$. For electronic signals, it is not possible to find $f(t)$ that encompasses its complexity. Sampling is introduced in order to digitize the continuous analog signal (see Figure 2.1).

$$x[k] = x_1, x_2, x_3 \ldots x_N$$

Basically, sampling is slicing the signal into discrete point in time. Sampling cuts a continuous waveform $x(t)$ into equally spaced time intervals T_s, which is known as sampling time. The relationship between time and sampling internal is:

$$t = n\,Ts = n / Fs \text{ and } Fs = 1/Ts$$

Where: n is the position of the number in the sequence $X[k]$, $n = 0, 1, 2, \ldots$

Fs is the sampling frequency

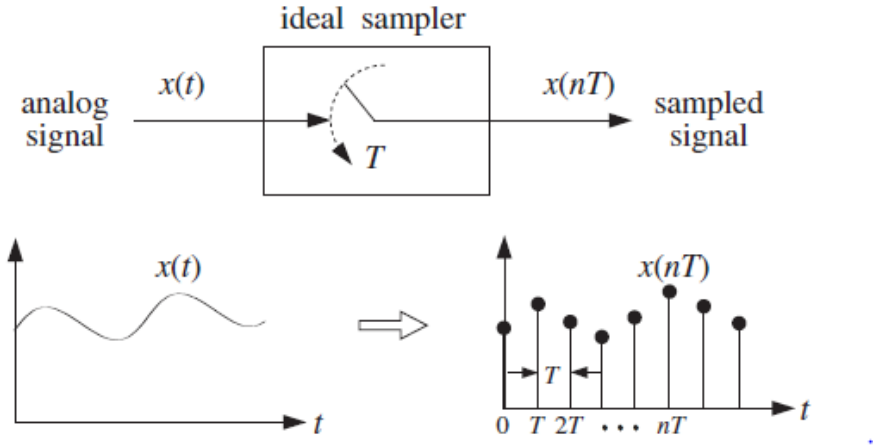


Figure Appendix V: An analog signal sampled into discrete one using ideal sampler

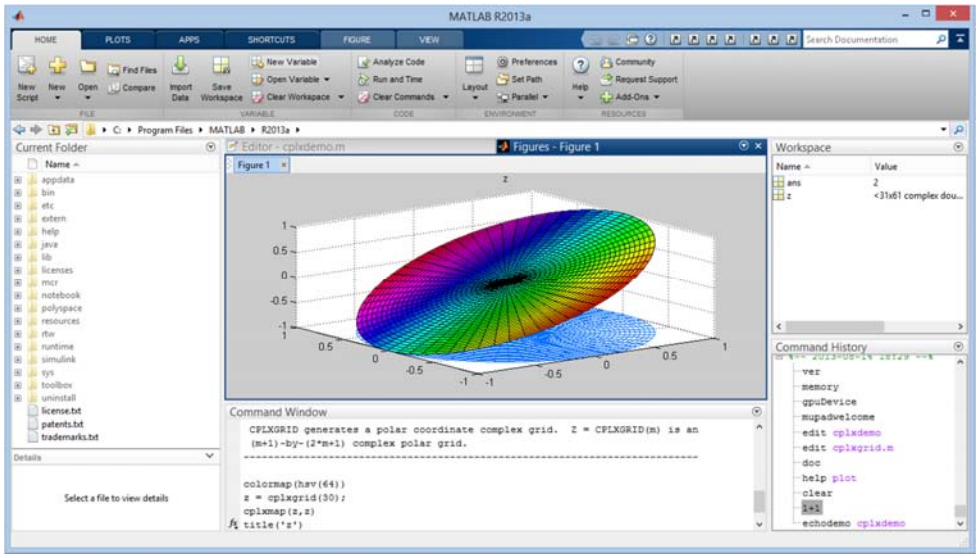
Two conditions should be met to perform sampling, referred as digitization as well, are bandwidth and the sampling frequency.

1. The continuous analog signal should be *bandlimited*. That is to say it should have a maximum threshold frequency, say *fmax*.
2. Sampling rate should be at least twice the maximum frequency.

The relationship between F*s* and *fmax* is defined by: $Fs \geq 2fmax$

Appendix III: About Matlab

MATLAB (matrix laboratory) is a multi-paradigm numerical computing environment and fourth-generation programming language. Developed by MathWorks, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, creation of user interfaces, and interfacing with programs written in other languages, including C, C++, Java, Fortran and Python.



Although MATLAB is intended primarily for numerical computing, an optional toolbox uses the MuPAD symbolic engine, allowing access to symbolic computing capabilities. An additional package, Simulink, adds graphical multi-domain simulation and Model-Based Design for dynamic and embedded systems.

The MATLAB application is built around the MATLAB scripting language. Common usage of the MATLAB application involves using the Command Window as an interactive mathematical shell or executing text files containing MATLAB code.

When creating a MATLAB function, the name of the file should match the name of the first function in the file. Valid function names begin with an alphabetic character, and can contain letters, numbers, or underscores. MATLAB supports developing applications with graphical user interface features. MATLAB includes GUIDE (GUI development environment) for graphically designing GUIs. It also has tightly integrated graph-plotting features.

Appendix IV: Some Matlab Codes for Hardware Acquisition

Note: For full codes and inquiries contact me via e-mail: Badreddinez [at] Hotmail [dot] fr.

serialSetup.m

```
function [s,flag] = setupSerial(comPort)
flag = 1;
s = serial(comPort);
set(s,'DataBits',8);
set(s,'StopBits',1);
set(s,'BaudRate',57600);
set(s,'Parity','none');
s.InputBufferSize = 5200000; %170000; %17000;%1024;512;
% Equivalent to be transfered in 2:30 mins
fopen(s);
a = 'b';
while(a~='a')
    a = fread(s,1,'uchar'); %% remain here until acknoledgment
end;
if (a == 'a')
    disp('serial Read');
end

fprintf(s,'%c','a');
mbox = msgbox('Serial Communication Setup: Connected to Arduino'); uiwait(mbox);
fscanf(s,'%u');
end
```

seriolClose.m

```
clc
clear all
if ~isempty(instrfind)
    fclose(instrfind);
    delete(instrfind);
end
close all
clc
disp('Serial Port Closed');
```

ECG_int_V3_Handshaked_Offline_Fastest.m

```
     clear all; clc; close all;

     MyReadDataGlobal = [];
     DecimalDataCh = [];
     ChannelData = [];
     Vmat = [];
     ChSel = input('Select your channel: ');
     Time = input('Enter The time of recording: ');
     Channel = ChSel; % select your channel that you want to read

    % Try-catch is to prevent Matlab from crashing when the program is finished
    try

    % Initialize serial port
Mode = input('Selection your mode: Wireless/USB/WirelessTimed [1/2/3] ');
    switch Mode
        case 2
            [s,Flag] = setupSerial('COM7');
        case 1
            [s,Flag] = setupSerialXBee('COM5');
        case 3
            Timing = input('How many seconds you want to record');
            [s,Flag] = setupSerialXBee('COM5',Timing);
        otherwise
            disp('Choose one of the two modes only');
    end

    pause(Time); % allow the buffer to start filling before readings
    tic
    MyReadDataGlobal =    fread(s,(s.BytesAvailable));
    fclose(s);
    toc

Step = 0;
for j=1:floor(length(MyReadDataGlobal)/17)
    Step = 17*(j-1);
    DecimalDataCh(j) = (MyReadDataGlobal(Step+1+4+(Channel*2-2)))*256 +
(MyReadDataGlobal((Step+1+5+(Channel*2-2)))); end

toc
    plot(DecimalDataCh);

    Saving = input('Do you want to save? Y/N\n','s');
    if Saving == 'y'
        ECG = DecimalDataCh;
        Acquisition = MyReadDataGlobal;
        FileName = input('Enter the name of the person \n','s');
        save(FileName,'ECG','Acquisition');
    end

    catch ME
    fprintf(1, 'Sorry, you"re going to have to close out of Matlab to close the
serial port\n');

    end
```