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 **Computer**

Option: **Computer**

Title:

Wireless Transmission for Vital Signals

Presented by:

- **Hachemi Latifa**
- **Meghough Thenhinane**

Supervisor:

Pr. BOUZID Merouane.

Dr. MEZIANE Nacéra .

Dr. ZITOUNI Abdelkader.

Registration Number:……/2018

ملخص:

في هذا المشروع المتواضع قمنا بتصميم نظام كامل يتم من خلاله الحصول على نبض القلب الذي ينتج خلال قيام القلب بضخ الدم. كماعملنا أيضا على إرسال نبض القلب إلى السحابة المعلوماتية باستخدام تطبيق أندرويد حتى يتمكن الطبيب من تحليل النتائج عن بعد.

Résumé :

Dans ce projet, nous avons conçu un système complet à travers lequel on obtient le battement cardiaque produit par le cœur lors du pompage du sang. On s'est intéressé aussi à l'envoi du battement cardiaque vers le cloud en utilisant une application android, et ce jusqu'à ce que le médecin peut analyser les résultats à distance.

Abstract :

In this modest project, we have designed a complete system through which we obtain the heartbeat produced by the heart when pumping blood. Then we worked to send these heartbeat to the cloud using an android application until the doctor can analyse the results remotely.

Dedication

This Master graduation thesis is dedicated to:

First and foremost , We have to thank Our parents for their love and support throughout our life , we would like to thank them for giving us strength to reach for the stars ,chase our dreams and own the best education so far . Who never stop giving of themselves in countless ways,

This thesis is also dedicated to someone special who helped us , encouraged us and supported us unconditionally , Krizou Lokmane

Who has always been a constant source of support and encouragement during the challenges of our master journey and especially our final year project .

It is dedicated also to our sister and brother , Nabila and Ililas who were ready all the time to give us all the help we needed .

Latifa Hachemi and Meghough Thenhinene

.

Acknowledgement

First of all we thank God for giving us courage to finish our studies.

We would like to express our special thanks of gratitude to our advisors , Dr.N.Meziane , Mr.A.Zitouni, and Mr.M. Bouzid who gave us the golden opportunity to do this wonderful project on the development of wireless acquisition system for ECG signal and sending it to the cloud.

We would also like to extend our very profound gratitude to Krizou Lokmane for all his efforts during the last months helping us , encouraging us and providing us with all the needed matrial to accomplish our project on time ,Thank you Lokmane.

We must also thank our parents and friends for their unfailing support and continuous encouragement throughout our years of study and through the process of researching and writing this thesis. This accomplishment would not have been possible without them.

We would also thank the jury for their kindness to evaluate our work

Thank you.

Liste of Contents

Chapter IV : Experimental tests and Results.

List of Tables

Chapter II

Liste of Figures

Chapter I

Chapter II

Chapter III

Chapter IV

List of Abbreviations

APK: Android Package

bpm : beats per minute.

CVD : Cardiovascular Disease.

ECG: Electrocardiogram.

EKG: Electrocardiogram.

GND : Ground.

HF: Hight Frequency.

HTTP: Hypertext Transfer Protocol.

IDE: integrated development environment.

iOS: iphone Operating System.

IoT: Internet of Things.

IT: Information Technology

JSON: JavaScript Object Nation.

LF: Low Frequency.

MF: Medium Frequency.

NDK: Native Developmet Kit.

PANs: Personal Area Networks.

SDN : ShutDown.

UML: Unified Modeling Language.

VHF: Very High Frequency.

VLF: Very Low Frequency.

XML: Extensible Markup Language

General introduction

Our society is growing more and more day after day in various fields, but this development cannot be without problems. In fact, the acceleration of the technological events is creating an atmosphere of unstable conditions for humans' health, which explains the spread of some chronic diseases in this current time. One of the most common diseases is the cardiovascular disease (CVD) which is the principal cause of death globally. More people die annually from CVDs than from any other cause [1].

Heart disease, which is the most prevalent cause of early death in most developed countries, can be caused by genetic factors, a poor lifestyle, a simple consequence of aging, or a combination of all the cited three factors [2]. Consequently, over 610000 people die of heart disease in the United States of America every year [3].

In order to diagnose cardiovascular disease and check its development, if there is any, measuring of biosignals from patients is a very important factor when it comes to diagnosis of heart disease. Among the various devices for measuring and detecting heart disease, electrocardiogram (ECG) is preferred due to its precision, convenience, and low cost [4, 5]. The ECG signal is used to observe the electrical activity of the heart. It can detect abnormal heart rhythms, insufficient blood and oxygen supply to the heart and an excessive thickening of heart muscle [6].

Nowadays, home health care is the fastest-growing expense in the Medicare program. Patients and families are choosing the option of home care more frequently, that’s why many wireless devices have been created [7].

In previous research, a portable ECG device using wireless LAN 802.11b protocols has been successfully designed. One problem that has not been resolved yet is the relatively big dimension of those devices which makes it difficult to the users to bring and live with them freely[8].

Looking at the needs and problems that still exist in previous research, this project aims to build an ECG device with wireless transmission based on smart-phone Android application. The designed wireless ECG device system consists of a mini-sized hardware that can be used without interfering with user's activities. The ECG signal

acquired will be sent via Bluetooth protocol to the cloud in case of heartbeat issue is presented. With this system, heart conditions of practicing athletes could remotely be monitored by a doctor far from the athlete place.

This manuscript consists of 4 chapters that are organized as follow:

Chapter I: Modern Electrocardiogram Systems.

Chapter II: Cloud-based electrocardiogram hardware design.

Chapter III: Cloud-based android application software design.

Chapter IV: Experimental tests and Results.

Chapter I

Modern Electrocardiogram Systems

I.1. Introduction:

In modern society, technology is playing a main role in most of the fields, including the medical field.

Nowadays, the patient doesn't have to visit the doctor each time so he can check his health status, but he can remotely check it with the help of the cloud computing based android technology system.

In this chapter, we will describe some generalities about the cardiovascular system, the Cloud computing and the Wireless technology, new technical tendencies adopted in modern ECG system.

I.2. ECG basics:

Cardiovascular disease (CVD) is a general term for health conditions affecting the heart or blood vessels states.

It is usually associated with a build-up of fatty deposits inside the arteries which is known as atherosclerosis and causes an increased risk of blood clots. It can also be associated with damage to arteries in organs such as the brain, heart, kidneys and eyes [9]. The exact cause of CVD isn't clear, but there are lots of things that can increase the risk of getting it [10].

I.2.1. Symtoms:

There are many different types of cardiovascular disease. Symptoms will vary, depending on the specific type of disease a patient has. However, typical symptoms of an underlying cardiovascular issue include:

- Pains or pressure in the chest which may indicate angina.
- Pain or discomfort in the arms, the left shoulder, elbows, jaw, or back.
- Shortness of breath, also known as dyspnea.
- Nausea and fatigue.
- Light-headed or faint.
- Cold sweat.

Overall, symptoms vary and are specific to the physiological condition of the individual, but these are most common [11].

I.2.2. Electrocardiogram:

The electrocardiogram is the process of recording the electrical activity of the heart over a period of time using electrodes placed on the skin. These electrodes detect the tiny electrical changes on the skin that arise from the heart muscle's, electrophysiological patterns of depolarizing and repolarizing during each heartbeat. It is a usually performed as cardiology test [12].

I.2.3. Heart anatomy:

The heart is one of the most important organs in an individual's body. It acts like a dual-chambered pump that circulates blood and provides the body with oxygen and nutrients it needs to survive [13].

The heart has four hollow chambers (figure I.1), two atria (right and left) and two ventricles (right and left). The Atria are smaller than the ventricles and its walls are much thinner and less muscular. These chambers are located in the upper part of the heart. Its function is to receive the blood and send it to the ventricles.

On the other hand, the ventricles are larger and muscular, because they have to fulfill functions that require the use of greater force [14].

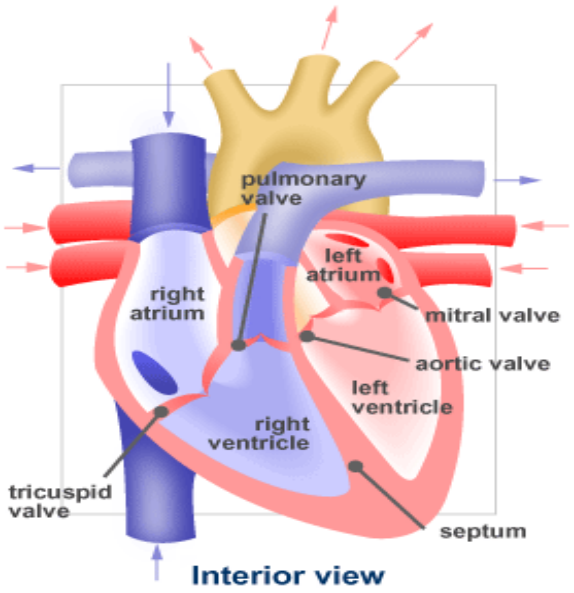


Figure I.1: A diagrammatic structure of the heart [15].

I.2.4. Electrical activity of the heart [16]:

The parts of the heart normally beat in orderly sequence: Contraction of the atria (atrial systole) is followed by contraction of the ventricles (ventricular systole), and during diastole all four chambers are relaxed. The heartbeat originates in a specialized cardiac conduction system and spreads via this system to all parts of the myocardium. The structures that make up the conduction system are the sinoatrial node (SA node), the internodal atrial pathways, the atrioventricular node (AV node), the bundle of His and its branches, and the Purkinje system**.** The various parts of the conduction system and, under abnormal conditions, parts of the myocardium, are capable of spontaneous discharge. However, the SA node normally discharges most rapidly, with depolarization spreading from it to the other regions before they discharge spontaneously. The SA node is therefore the natural cardiac pacemaker**,** with its rate of discharge determining the rate at which the heart beats. Impulses generated in the SA node pass through the atrial pathways to the AV node, through this node to the bundle of His, and through the branches of the bundle of His via the Purkinje system to the ventricular muscle. Each of the cell types in the heart contains a unique electrical discharge pattern; the sum of these electrical discharges can be recorded, on the skin surface, as the electrocardiogram (ECG).

I.2.5. ECG tracing:

The baseline of an ECG tracing is called the isoelectric line and denotes resting membrane potentials. Deflections from this point are lettered in alphabetical order, and following each, the tracing normally returns to the isoelectric point [17].

The first deflection is the P wave. Wave of atrial repolarization is invisible because of low amplitude. Normal P wave is no more than 2.5 mm of amplitude and less than 120 ms of duration (three 1-mm-divisions) in width in any lead [18].

The QRS complex represents depolarization of ventricular muscle cells. The Q portion is the initial downward deflection, the R portion is the initial upward deflection, and the S portion is the return to the baseline, or the so-called isoelectric point. Often, the Q portion is not evident, and the depolarization presents as only an “RS” complex. In any case, the complex does not represent ventricular contraction. One assumes that

contraction will commence at the peak of the R portion of the complex. Unlike contraction of the atria, ventricular contraction can be confirmed clinically by palpating a pulse or by monitoring a pulse oximeter wave form. A patient in cardiac arrest may have normal QRS complexes on his or her ECG. Ventricular muscle cells are depolarizing, but there is no contraction. This phenomenon is called pulseless electrical activity. Following depolarization, ventricular muscle repolarizes, and this event is great enough in amplitude to generate the T wave on the ECG tracing [17].

QRS duration is the width of that complex from the beginning to the end, irrespective of the number of present deflections. Normally it lasts no more than 120 ms [18].

The PR interval is measured from the beginning of the P wave to the beginning of the R portion of the QRS complex. This is conventional because the Q portion of the complex is so frequently indiscernible. Because the PR interval commences with atrial muscle depolarization and ends with the start of ventricular depolarization, one can assume that the electrical impulse passes through the AV node into the ventricle during this interval. If the PR interval is prolonged, one may deduce that AV block is present [18]. Normal range is 120-200ms and no longer[18].

The QT interval is measured from the beginning of the QRS to the end of the T wave. It represents the time in which the ventricles depolarize and repolarize and is a measure of ventricular action potential (AP) duration [18]. A typical ECG recording from a normal person is shown in figure I.2.

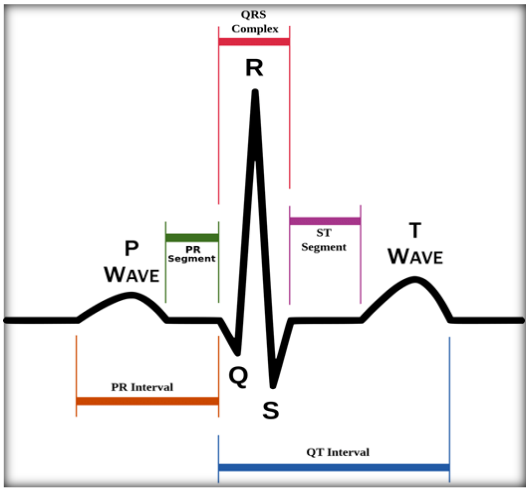


Figure I.2: Typical ECG record from normal person [18].

I.2.6. ECG leads placement:

A lead is a view of the electrical activity of the heart from a particular angle across the body (figure I.3). The ECG leads are grouped into two electrical planes. The limb leads (Lead I-III, aVR-aVl-aVF) view the heart from a vertical plane, while the chest leads (V1-V6) view the heart from a horizontal plane [19].

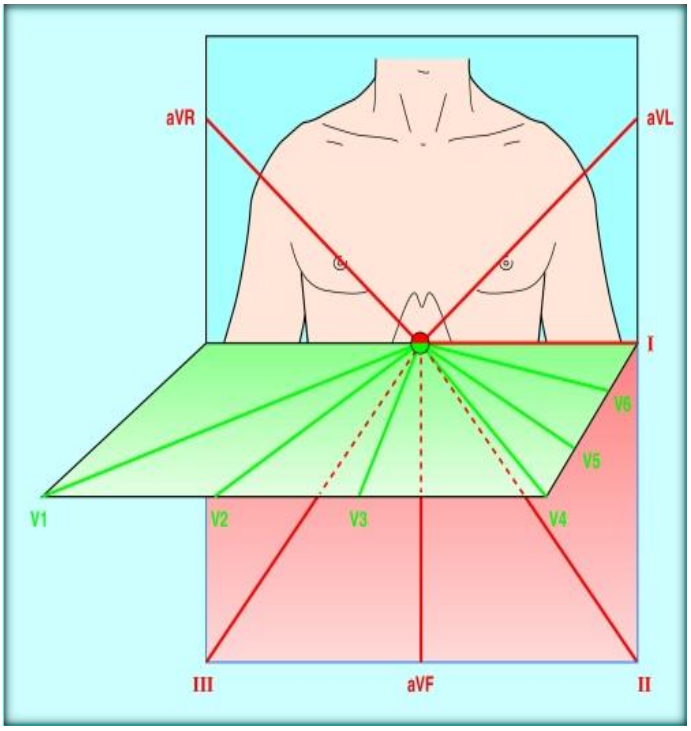


Figure I.3: ECG leads placement [20].

I.2.6.1. Limb Leads [21]:

- Lead I: The positive lead is above the left breast or on the left arm and the negative lead is on the right arm. It records the potential difference between the Left arm and Right arm.
- Lead II: The positive lead is on the left abdomen or left thigh and the negative lead is also on the right arm. It records the potential difference between the left leg and the right arm.

- Lead III: The positive lead is also on the left abdomen or left lower lateral leg but the negative lead is on the left arm. It records the potential difference between the left leg and the right arm.

- Lead aVR faces the heart from the right shoulder and is oriented to the cavity of the heart.
- Lead aVL faces the heart from the left shoulder and is oriented to the left ventricle.
- Lead aVF faces the heart from the left hip and is oriented to the inferior surface of the left ventricle. The figure I.4 below shows the limb leads placement.

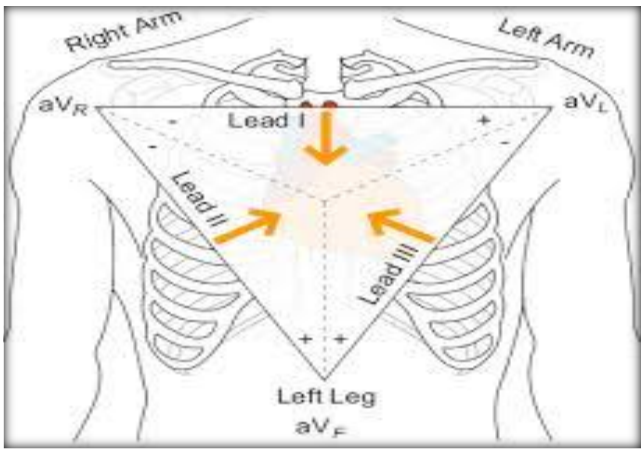


Figure I.4: Limb leads placement [21].

I.2.6.2. Chest leads [22]:

The chest leads Records the potentials in the horizontal plane. Each lead is positive**.**

The major forces of depolarization move from right to left.

- V1 - Fourth intercostal, right sternal border.
- V2 - Fourth intercostal, left sternal border.
- V3 - Equal distance between V2 and V4.
- V4 - Fifth intercostal, left mid clavicular line.
- V5 - Anterior axillary line, same level with V4.
- V6 - Mid axillary line, same level with V4 and V5.
- V1 and V2 are negative deflections.
- V3, V4, V5 and V6 become more positive (peak positive is V3 or V4). The figure I.5 shows the chest leads placement.

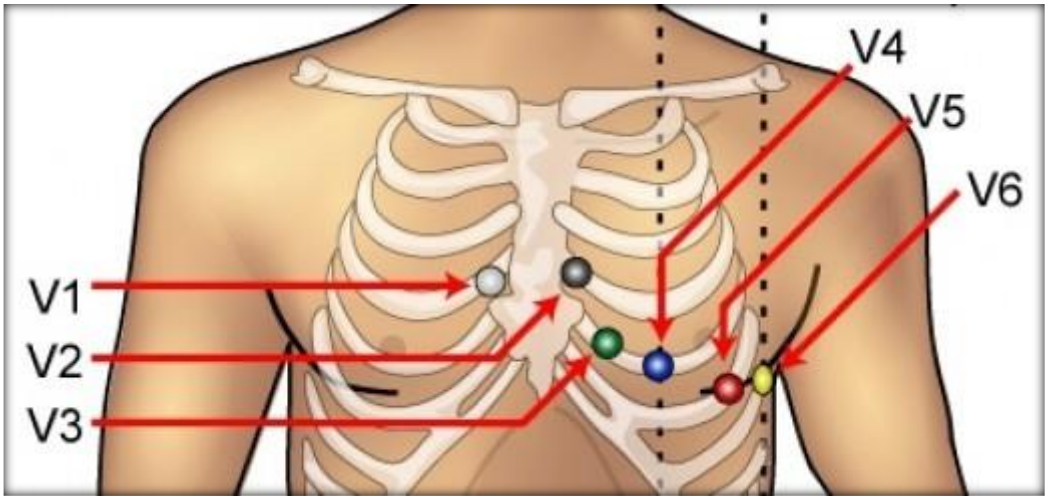


Figure I.5: Chest leads placement [23].

I.2.7. Heart rate:

Heart rate, also known as pulse, is the number of times a person's heart beats per minute. Normal heart rate varies from person to person, but a normal range for adults is 60 to 100 beats per minute [24].

Many studies, as well as expert consensus, indicate that the normal resting adult human heart rate is probably a range between 50 and 90 bpm (beats per minute). Though the American Heart Association states that the normal resting adult human heart rate is 60–100 bpm. Tachycardia is a fast heart rate, defined as above 100 bpm at rest. Bradycardia is a slow heart rate, defined as below 60 bpm at rest. During sleep a slow heartbeat with rates around 40–50 bpm is common and is considered normal. When the heart is not beating in a regular pattern, this is referred to as an arrhythmia. Moreover, abnormalities of heart rate sometimes indicate disease [26].

A normal heart rate depends on the individual, age, body size, heart conditions, whether the person is sitting or moving, medication use and even air temperature. Emotions can affect heart rate; for example, getting excited or scared can increase the heart rate [25].

I.3. Introduction to Cloud computing:

Cloud computing is an information technology paradigm that enables ubiquitous access to shared pools of configurable system resources and higher-level services that can be rapidly provisioned with minimal management effort, often over the Internet.

Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a public utility (figure I.6).

Third-party clouds enable organizations to focus on their core businesses instead of expending resources on computer infrastructure and maintenance. Advocates note that cloud computing allows companies to avoid or minimize up-front IT infrastructure costs. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved manageability and less maintenance, and that it enables IT teams to more rapidly adjust resources to meet fluctuating and unpredictable demand. Cloud providers typically use a "pay-as-you-go" model, which can lead to unexpected operating expenses if administrators are not familiarized with cloud-pricing models [27].

There are many clouds computing that can be used to receive the data like Amazon web services, Ubidots, Firebase and Backendless. All of these clouds can receive and send data, but we found that Firebase is the most suitable cloud for our project. The first reason is that firebase is provided by Google which has a large community in software development and a good repetition besides it’s for free and has too many resources and tutorials that can help the developpers understanding how clouds work properly. Other clouds could be used also like Ubidots which is an IoT (Internet of Things) cloud platform, it provides the user the ability to display graphs and signals, but this can be done if we send directly the results from Arduino to the cloud and it doesn’t have supports to the mobile application which made us choose firebase for our project.

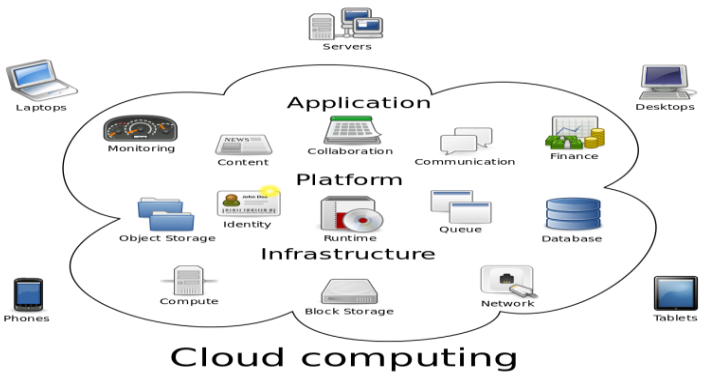


Figure I.6: Cloud computing [27].

I.4. Wireless Transmission:

Wireless transmission is a form of unguided media. Wireless communication involves no physical link established between two or more devices, communicating wirelessly. Wireless signals are spread over in the air and are received and interpreted by appropriate antennas.

When an antenna is attached to electrical circuit of a computer or wireless device, it converts the digital data into wireless signals and spread them all over within its frequency range. The receptor on the other end receives these signals and converts them back to digital data.

Radio frequency (figure I.7) is one of the most used ways to transmit data wirelessly. It is easier to generate and because of its large wavelength it can penetrate through walls and structures alike. Radio waves can have wavelength from 1 mm – 100,000 km and have frequency ranging from 3 Hz (Extremely Low Frequency) to 300 GHz (Extremely High Frequency). Radio frequencies are sub-divided into six bands.

Radio waves at lower frequencies can travel through walls whereas higher RF can travel in straight line and bounce back. The power of low frequency waves decreases sharply as they cover long distance. High frequency radio waves have more power.

Lower frequencies such as VLF(Very Low Frequency, LF(Low Frequency), MF (Medium Frequency) bands can travel on the ground up to 1000 kilometers, over the earth's surface.

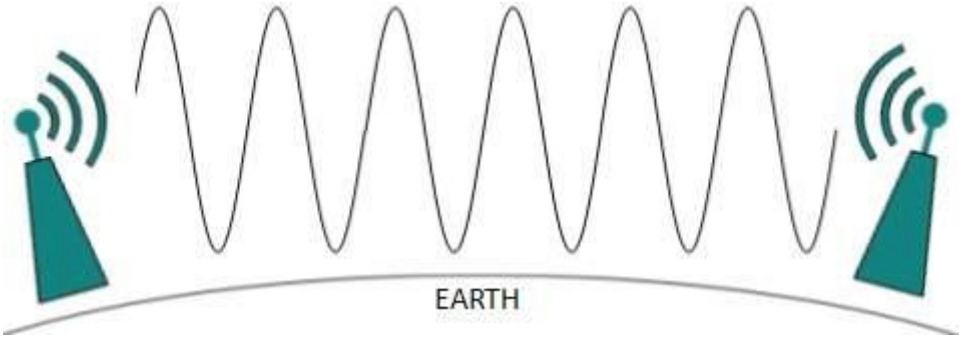


Figure I.7: Radio Transmission[28].

Radio waves of high frequencies are prone to be absorbed by rain and other obstacles. They use Ionosphere of earth atmosphere. High frequency radio waves such as HF (High Frequency) and VHF (Very High Frequency) bands are spread upwards. When they reach Ionosphere, they are refracted back to the earth.

Bluetooth is a wireless technology standard for exchanging data over short distances from fixed and mobile devices, and building personal area networks (PANs). Invented by Dutch electrical engineer Jaap Haartsen, working for telecom vendor Ericsson in 1994.

Bluetooth is a standard wire-replacement communications protocol primarily designed for low-power consumption, with a short range based on low-cost transceiver microchips in each device. Because the devices use a radio (broadcast) communications system, they do not have to be in visual line of sight of each other; however, a quasi optical wireless path must be viable. Range is power-class-dependent, but effective ranges vary in practice [28].

In this project we will use the HC-06 Bluetooth module which uses Radio transmission way to send the data from our system.

I.5. Conclusion:

In this chapter, we have defined the ECG and how it records the heart signal. We gave some descriptions that may help understanding the problem needed to solve. Also, we have introduced the basic concept of the cloud and wireless technology that makes it easy to understand how our cloud will behave. In the next chapter, we will define the components needed to realize our project and the implementation of the hardware design of the system.

Chapter II

Cloud-based electrocardiogram hardware design

II.1.Introduction:

Wireless technology has become an essential part of life in many parts of the world reaching even some of the more remote regions of the planet.

In this chapter, we will present our developed hardware system. First, we used the AD8232 ECG amplifier board to capture the ECG data in real time. Then, the data will be sent to the android application using an Arduino UNO card and a Bluetooth module (HC-06). Finally, the data will be sent to the cloud (Firebase).

II.2.ECG Measurement System:

In this section, we will enumerate all the specifications of the tools we've needed to build the ECG measurement system. The following diagram explains how we are going to connect them together (figure II.1).

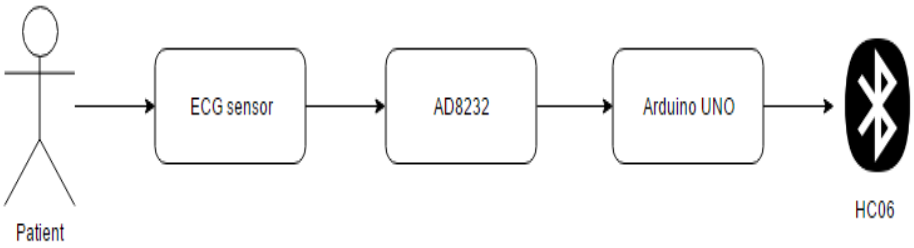


Figure II.1: Block diagram of the ECG measurement system.

II.2.1. Electrodes:

Electrodes are used for sensing bio-electric potentials as caused by muscle and nerve cells. ECG or EKG (Electrocardiogram) electrodes, usually in disposable form, work as transducers converting the ionic flow from the body through an electrolyte into an electron current and consequentially an electric potential measurable by the front-end of the EKG system (figure II.2). These transducers, known as bare-metal or recessed electrodes, generally consist of a metal such as silver or stainless steel, with a jelly electrolyte that contains chloride and other ions [29].

Figure II.2: Disposable Electrodes and its cables [29].

II.2.2. AD8232 ECG amplifier board:

II.2.2.1. Description:

The AD8232 is an integrated signal-conditioning block for the ECG signal and other biopotential measurement applications. It is designed to convert the low-level, noisy signal from the electrodes into a large, filtered signal that can be easily read by a low-resolution analog to digital converter (ADC).

The AD8232 can implement a two-pole high pass filter for eliminating motion artifacts and the electrode half-cell potential. This filter is tightly coupled with the instrumentation amplifier's architecture to allow both large gain and high pass filtering in a single stage, saving space and cost.

Additionally, an operational amplifier enables the AD8232 to create a three-pole low pass filter to remove line noise and other interference signals. The user can select the cutoff frequency of all filters to suit different types of applications.

The AD8232 includes a fast restore function that reduces the duration of otherwise long settling tails of the high-pass filters. After an abrupt change that rails the amplifier (such as a leadoff condition), the AD8232 automatically adjusts to higher cutoff filter.

This functionality allows the AD8232 to recover quickly, and therefore take valid measurements soon after the leads are connected to the subject [30].

II.2.2.2. Features: (Figure II.3)

- Resistor programmable gain range: 10^1 to 1000.
- Supply voltage range: ±4 V to ±8 V.
- Rail-to-rail input and output.
- Maintains performance over −40°C to +125°C.
- Excellent ac and dc performance.
- 110 dB minimum CMR @ 60 Hz, G = 10 to 1000.
- 10 µV maximum offset voltage (RTI, ±5 V operation).
- 50 nV/°C maximum offset drift.
- 20 ppm maximum gain nonlinearity.

Figure II.3: The AD8232 ECG amplifier board [30].

II.2.3. Arduino UNO:

II.2.3.1. Description:

Arduino is an open-source electronics platform based on easy-to-use hardware and software. Arduino boards are able to read inputs: light on a sensor, a finger on a button, or a Twitter message, and turn it into an output: activating a motor, turning on a LED, publishing something online. Over the years, Arduino has been the brain of thousands of projects, from everyday objects to complex scientific instruments. A worldwide community of makers: students, hobbyists, artists, programmers, and professionals, has

gathered around this open-source platform, their contributions have added up to an incredible amount of accessible knowledge that can be of great help to novices and experts alike [31]. Figure II.4 shows the Arduino board.

Figure II.4: Arduino board [31].

II.2.3.2. Main factures of Arduino:

Arduino has the following main features [32]:

- Microcontroller: ATmega168 or 328.
- Operating voltage: 5V.
- Input voltage (recommended): 7-12V.
- Input voltage limits: 6-20V.
- Digital I/O Pins: 14.
- 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 (ATMEGA328).

II.2.3.3. Arduino Software:

The Arduino software consists of two parts the Arduino bootloader and the Arduino IDE, as shown in the figure II.5:

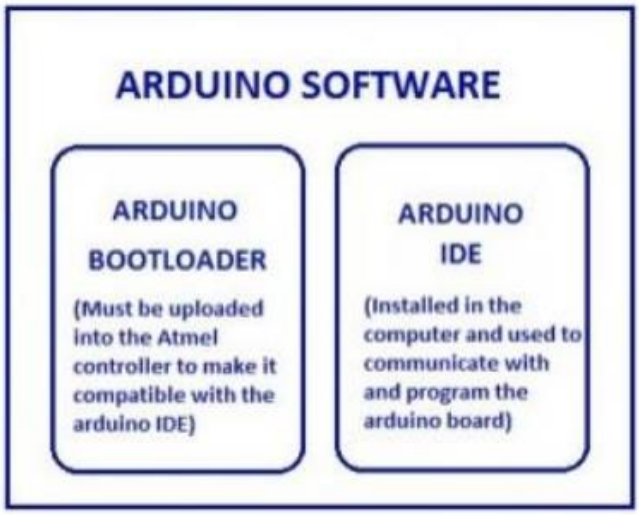


Figure II.5: Arduino two software parts[32].

II.2.3.3.1. IDE:

The Arduino integrated development environment (IDE) is a cross-platform application written in Java and derives from the IDE for the Processing programming language and the wiring projects. The Arduino IDE is: an editor, a compiler, an uploader, and a serial terminal interface (the Serial Plotter). all rolled into one app [33].

II.2.3.3.2. Serial Plotter:

The Serial Plotter, mentioned previously, monitors the exchanged data between the Arduino and the host computer system over the connected USB cable. The Arduino's Serial Plotter is meant to be useful to display graphs based on the transferred data from the sensor. The figure II.6 shows the IDE serial Plotter.

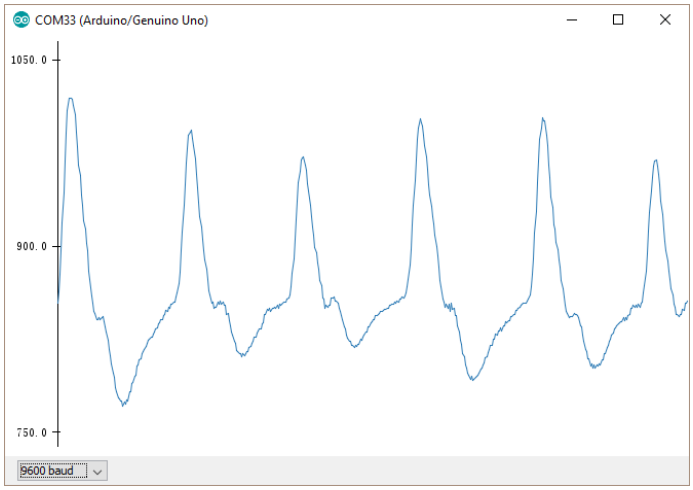


Figure II.6: IDE serial Plotter[33].

II.2.3.4. Sketch

A sketch is the name that Arduino uses for a program. It's the unit of code that is uploaded to and run on an Arduino board. Arduino programs are written in C or C++. The Arduino IDE (figure II.7) comes with a software library called "Wiring" from the original Wiring project, which makes many common input/output operations much easier. The users need only to define two functions to make an executable cyclic executive program:

- Setup (): a function run once at the start of a program that can initialize settings.
- Loop (): a function repeatedly called until the board powers off.

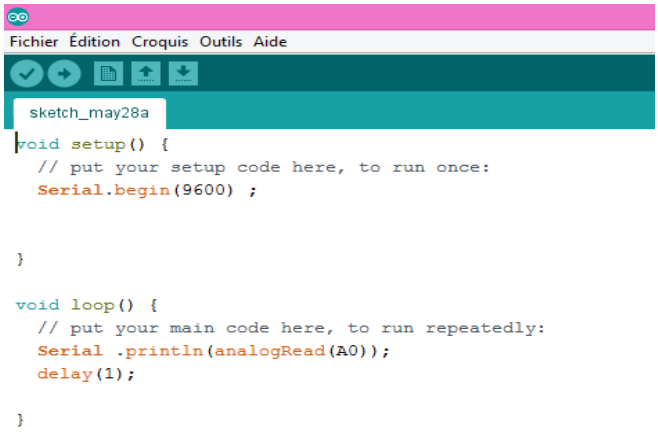


Figure II.7: The IDE software[33].

II.2.4. HC-06 Bluetooth module:

II.2.4.1. Description:

The HC-06 is a class 2 slave Bluetooth module designed for transparent wireless serial communication. Once it is paired to a master Bluetooth device such as PC, smart phones and tablet, its operation becomes transparent to the user. All data received through the serial input is immediately transmitted over the air. When the module receives wireless data, it is sent out through the serial interface exactly at it is received. No specific user code to the Bluetooth module is needed at all in the user microcontroller program.

The HC-06 module works with a supply voltage of 3.6VDC to 6VDC. However, the logic level of RXD pin is 3.3V and is not 5V tolerant. A Logic Level Converter is recommended to protect the sensor if connected to a 5V device (e.g Arduino Uno and Mega) [34].

II.2.4.2. Features:

Compatible with the Bluetooth V2.0 protocol [35]: (Figure II.8 shows the HC-06 module).

- Operating Voltage: 3.3V.
- Adjustable baud rate: 1200, 2400, 4800, 9600,19200, 38400, 57600, 115200.
- Size: 28mm x 15mm x 2.35mm.
- Operating Current: 40mA.
- Sleep Current: < 1mA.

Figure II.8: Bluetooth HC06 Module[35].

II.2.4.3. Specification:

In the following, are some specifications for the HC 06 Bluetooth module [36]:

- Bluetooth protocol: Bluetooth 2.0+ EDR standard.
- USB protocol: USB v1.1/2.0.
- Operating frequency: 2.4GHz ISM frequency band.
- Modulation mode: Gauss frequency Shift Keying.
- Transmit power: ≤ 4dBm, second stage.
- Sensitivity: ≤-84dBm at 0.1% Bit Error Rate.
- Transmission speed: 2.1Mbps (Max)/160 kbps(Asynchronous).
- 1Mbps/1Mbps (Synchronous).
- Supported configuration: Bluetooth serial port (major and minor).
- Supply Voltage: +3.3 VDC/ 50mA.
- Size: 36.5*16mm.

II.3.Hardware implementation:

In this section, we have developed and implemented two interfaces. Firstly, an interface to display the ECG signals in the Arduino IDE and synchronously sends the data to the android application. Secondly, an interface to calculate the heartbeat rate.

II.3.1. Interfacing Heart Rate Monitor AD8232 with Arduino:

The AD8232 ECG amplifier board measures the electrical activity of the heart through the electrode pads placed on the patient's skin. By interfacing the AD8232 board with Arduino UNO (figure II.9), we can get the ECG signal through processing Arduino IDE window [37].

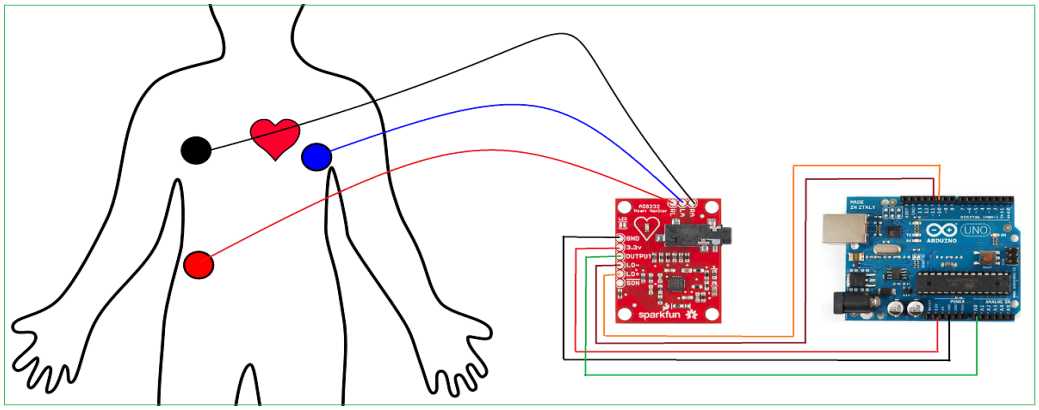


Figure II.9: Circuit diagram of AD8232 module with Arduino[37].

Based on figure II.9, we have the following pin configurations:

- Electrode pads pin configuration (table II.1):

Cable color	**Signal**
Red	RA(Right arm)
Green	LA(Left arm)
Yellow	RL(Right leg)

Table II.1: Electrode pads

- AD8232 to Arduino Pin configuration:

Board Label	Pin Function	Arduino Connection
GND	Ground	**GND**
3.3v	3.3v Power Supply	**3.3v**
OUTPUT	Output Signal	**A0**
LO-	Leads-off Detect –	**11**
LO+	Leads-off Detect +	**10**
SDN	Shutdown	**Not used**

Table II.2: AD8232 to Arduino

After defining the circuit diagram and the pins configurations, we have used the Arduino IDE to upload our code in the Atmega micro controller memory in order to receive and display the data into the serial Plotter. The code consists of two main parts: the setup function which configures the input/output pins and initializes the Serial baud rate, while the loop function to keep IDE read the data through the pins and display them in the serial Plotter, the following flowchart (figure II.10) illustrates the functions of the code.

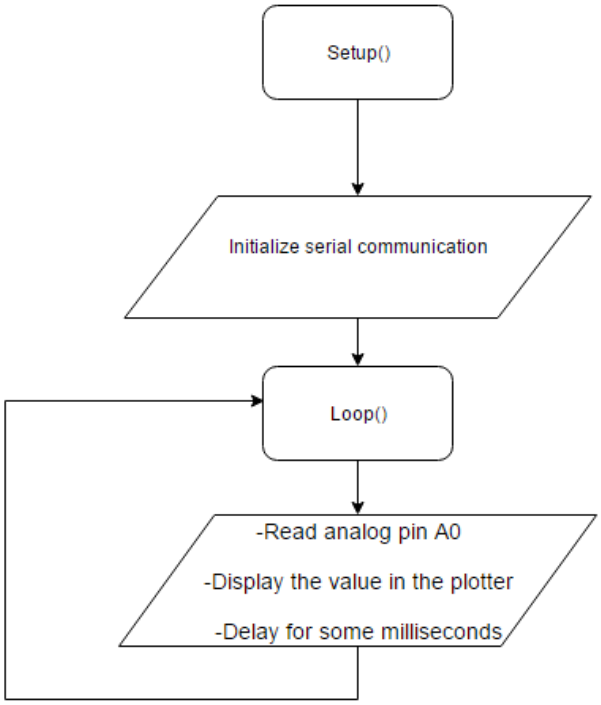


Figure II.10: flowchart representing the Interfacing of AD8232 with Arduino.

II.3.2. Interfacing Arduino with HC-06 Bluetooth module:

The Hc-06 module acts as a virtual serial port through which we can send and receive data. By using a serial terminal or a Bluetooth customized application on computer or phone, we can control and monitor our project. The table II.3 shows the pin connections of Arduino with HC-06 Bluetooth module.

HC-06	Arduino
GND	GND
VCC	5V
TX/TXD	Pin 0
RX/RXD	Pin 1

Table II.3: HC 06 module and Arduino Pin configuration

Combining the circuit diagrams (figure II.11 and figure II.9), we've got a system that can send continuously the data come from the AD8232 board which corresponds to the ECG signal through the HC06 Bluetooth module using the Arduino UNO to any device request connection.

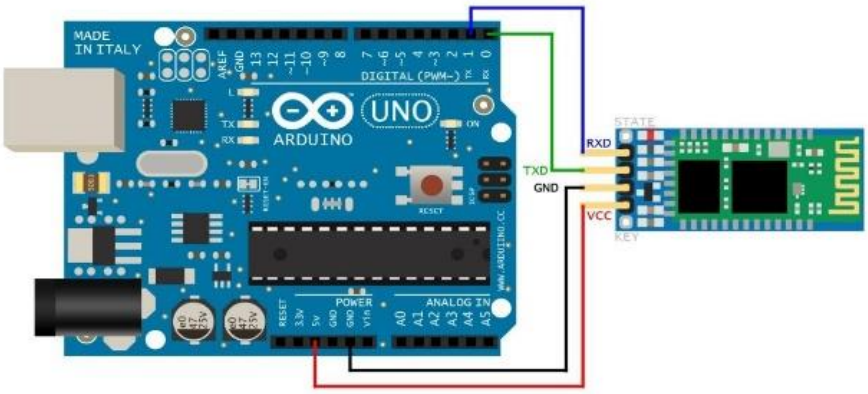


Figure II.11: Circuit diagram of the HC06 module with the Arduino.

The following flowchart (figure II.12) will show the way we send the data from the AD8232 board to the Android application using the HC06 module and the Arduino Uno card.

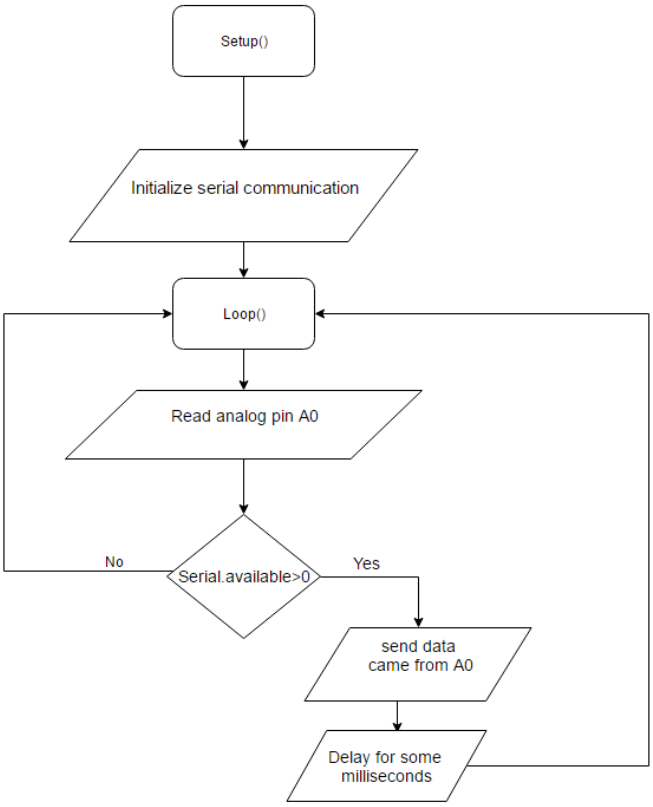


Figure II.12 flowchart representing the interfacing of the Arduino Uno card with the HC06 module to the AD8232 board.

II.3.3. Heartbeat rate monitor:

In this part, we have used an algorithm that detects the heartbeat and calculates the interval of its appearance. A simple explanation is the algorithm initializes all the variables that we need (oldtime, newtime, beat_time, flag_detek and check_beat_time).

The flag_detek checks the possibility of appearance of a heartbeat by calculating the difference between the old input data from A0 and the new one to a standard value. If it is above that value, the flag_detek will set as true which allows the algorithm to start calculating the time interval and the level of the heartbeat. To calculate the time interval, we set the newtime to the time when flag_detek set to true, for the beat_time we first check if the flag_detek is true, if yes, then the beat_time is equal to the current time minus the newtime. In order to get the heartbeat rate, we have used this equation:

$$\mathbf{HR} = \frac{\mathbf{60}}{\frac{\mathbf{beat\_time}}{\mathbf{1000}}} = \frac{\mathbf{6{\times}10^4}}{\mathbf{beat\_time}} \quad \text{..... (1)}$$

The result will be added to the old result if it is greater than 0 as following:

$$\boldsymbol{HR} = \mathbf{HR} \times \mathbf{60\%} + \mathbf{HR_{OLD}} \times \mathbf{40\%} \text{ ..... (2)}$$

If the absolute value of the difference of this new HR with HR_OLD is greater than 10, we will have to calculate it again using the same way but with different percentages as following:

$$\boldsymbol{HR} = \mathbf{HR} \times \mathbf{10\%} + \mathbf{HR_{OLD}} \times \mathbf{90\%}. \text{ ..... (3)}$$

Now, we got the heartbeat value, we set the flag_detek to false and repeat the algorithm again.

We have implemented this equations inside a function called: hitung_HR which we will call it in the loop function. The following flowchart (figure II.13) will descibe the code more clearly.

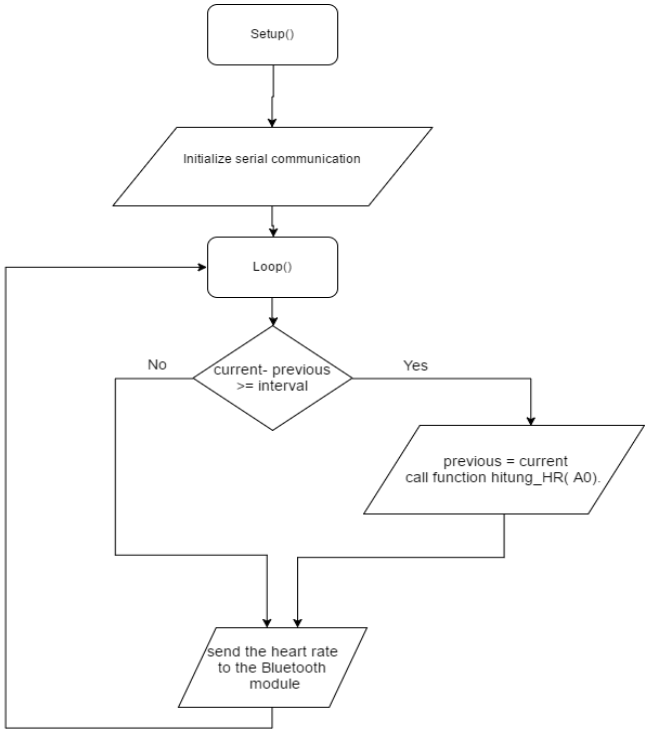


Figure II.13: flowchart representing the calculation of the Heartbeat and sending the result to the HC06 .

II.4.Firebase Cloud computing :

Firebase is one of the best cloud computing nowadays, its developed by Google groups. Firebase gives the tools to develop high-quality apps and grow the user base. Firebase Cloud Functions are a powerful tool. There are many use cases for Cloud Functions which are enumerated in the following list (figure II.14)[38]:

- Send out user notifications in case of events.
- Send a welcome email when a user signs up.
- Send confirmation emails when users are subscribing / unsubscribing.
- Perform automated Realtime Database Tasks to keep the database up to date and clean.
- If a user account is deleted, delete user's content from database as well.
- Track the number of elements in a Realtime Database list.
- Execute intensive tasks in the cloud instead of in your app.
- Automatically generate thumbnails for images which are uploaded into Cloud Storage.
- Send bulk emails to users.
- Integrating with third-party services.
- Translate text inserted into the Realtime Database by using Google Translate service.
- Process the payments.

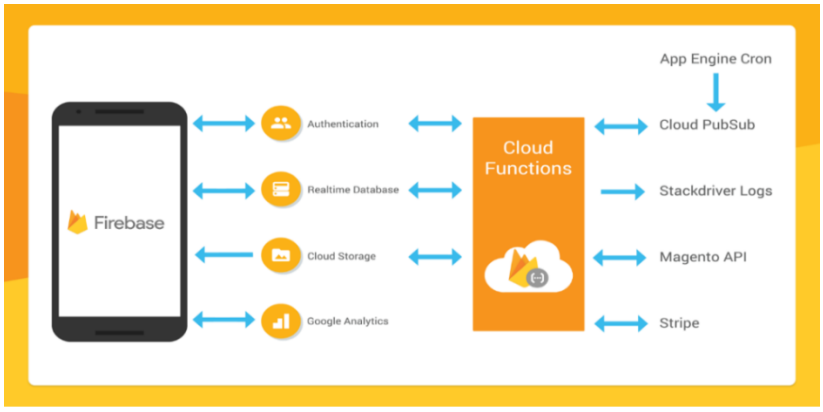


Figure II.14: Firebase functionalities[38].

II.4.1. Firebase Realtime database:

The Firebase Realtime Database is a cloud-hosted database. Data is stored as JSON (JavaScript Object Nation) and synchronized in Realtime to every connected client. When a user builds cross-platform apps with iOS (iPhone Operating System), Android, and JavaScript SDKs, all the clients share one Realtime Database instance and automatically receive updates with the newest data.

Instead of typical HTTP (Hypertext Transfer Protocol) requests, the Firebase Realtime Database uses data synchronization: every time data change, any connected device receives that update within milliseconds. Provide collaborative and immersive experiences without thinking about networking code.

The Firebase Realtime Database can be accessed directly from a mobile device or web browser; there's no need for an application server. Security and data validation are available through the Firebase Realtime Database Security Rules, expression-based rules that are executed when data is read or written.

With Firebase Realtime Database on the Blaze pricing plan, the user can support his app's data needs at scale by splitting his data across multiple database instances in the same Firebase project. Streamline authentication with Firebase Authentication on the user project and authenticate users across his database instances. Control access to the data in each database with custom Firebase Realtime Database Rules for each database instance [39].

For reliability performance, Firebase Realtime Database is a mature product:

- Stability you'd expect from a battle-tested, tried-and-true product.
- Very low latency, so it's a great option for frequent state-syncing.
- Databases are limited to zonal availability in a single region.

While for scalability, it scales to around 100000 concurrent connections and 1000 writes/second in a single database. Scaling beyond that requires sharing your data across multiple databases.

Firebase Realtime database can send simultaneously response around 100000 response/second for one single database, while can trigger more than 1000 function per one write while the data transfer into the database can reach 10MB/sec which provides a huge size to write bytes around 1.2MB/sec. That allows the user to stream his data in fastest way. All of those calculations and specifications and more can be found in the official website of Firebase Realtime database [40].

II.4.2. What You Should Know About Realtime Database?

Realtime Database is a cloud-hosted NoSQL database with SDK support for iOS, Android, and the web. It easily integrates with Firebase's other tools for authentication, file storage, analytics and others. This tool stores data in JSON documents, so everything is either a key or a value. Data synchronization uses web sockets, allowing for very snappy transactions. Realtime Database also handles updates for the user when a device is offline, syncing changes for the user when the network reconnects.

Developers love how quickly they can set up a Realtime Database backend and not have to worry about things like deployments, hardware, uptime, and scalability. The user can have a pretty robust backend server running in just a few minutes, letting him focus on the fun and unique parts of his app.

Realtime Database requires the user to write most of his system's application code on the client. This approach could be a positive or negative aspect depending on how the user looks at it. The user has to handle most logic in his client applications, which means duplicating code across Android, iOS, and web unless the user builds a Firebase Cloud Function to handle requests. If the user does that, though, he loses out on a lot of the built-in Realtime SDK features.

This client logic also requires the user to write his own data validation. Realtime Database has no concepts of data types. It will let him save values as strings or

numbers, or nested objects and arrays of strings and numbers to any field he wants. The arrays, however, are really just objects with indices used as keys. It's completely up to the user to manage these data types himself.

Queries fire and respond in near real time using Realtime Database, and it works beautifully. That is, as long as the user has thoroughly planned out his data structure in a way to support all the queries he wants to make, and will ever want to make, in his apps. That's no small feat. Even though it's fast, Realtime Database only supports queries on one field. Many users were disappointed to find that querying for data does not support many of the features developers are accustomed to using in more traditional relational databases.

Realtime Database's suggested solution to this which is flattening and denormalizing data. This is easier said than done when the user needs to track all of the locations a data element was copied into in case that data element ever gets updated. It's also difficult when supporting brand new features down the road without having to massively refactor his existing data schema[40].

II.4.3. Why Choose Realtime Database?

- You want a system that can easily track multiple live “streams” of data in Realtime. Examples include live chat, location-based events, Realtime games or collaboration.
- You expect to be making a lot of small writes/reads to the database.
- You're fine with handling JSON data objects, while validating type yourself.
- You do not foresee any need for complex queries on your dataset.
- You're ready to manage a flat and heavily denormalized database.

II.4.4. Add Firebase to the user app:

When using an Android Studio version 2.2 or later, the Firebase Assistant is the simplest way to connect the app to Firebase. The Assistant can connect the existing project or create a new one for the user with all the necessary Gradle dependencies. If the user is using an older version of Android Studio or have a more complex project configuration, he can still manually add Firebase to his app.

II.5. Conclusion:

In this chapter, we have described the ECG measurement system and the components used to achieve our goal including the cloud computing (firebase cloud).

In the next chapter, we will present the implementation of our software system (the android application and the Fire base cloud).

Chapter III

Cloud-based android application software design

III.1 Introduction

We propose in this section the implementation of our developed wireless transmission system, we will describe in detail each function and step from reading the patient's heartbeat signal till the configuration of the cloud to receive the data from the android application. The software system design will describe the functionality of the android application and the cloud.

III.2 Software System

In this section, we have implemented the Android application that can communicate with our hardware system and send notifications to the cloud whenever there is any issue with the patient's heart. The upcoming sections will focus on the steps which have been taken to build this system.

III.2.1 Introduction to Android studio:

Android Studio is the official Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA .On top of IntelliJ's powerful code editor and developer tools, Android Studio offers even more features that enhance the developer productivity when building Android apps, such as:

- A flexible Gradle-based build system.
- A fast and feature-rich emulator.
- A unified environment where you can develop for all Android devices.
- Instant Run to push changes to your running app without building a new APK (Android Package).
- Code templates and GitHub integration to help you build common app features and import sample code.
- Extensive testing tools and frameworks.
- Lint tools to catch performance, usability, version compatibility, and other problems
- C++ and NDK (Native Development Kit) support.
- Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

Each project in Android Studio contains one or more modules with source code files and resource files. Types of modules include:

- Android app modules.
- Library modules.
- Google App Engine modules.

By default, Android Studio displays your project files in the Android project view. This view is organized by modules to provide quick access to your project's key source files. All the build files are visible at the top level under Gradle Scripts and each app module contains the following folders:

- Manifests: Contains the AndroidManifest.xml file.
- Java: Contains the Java source code files, including JUnit test code.
- Res: Contains all non-code resources, such as XML (Extensible Markup Language) layouts, UI strings, and bitmap images.

The Android project structure on disk differs from this flattened representation. To see the actual file structure of the project, select Project from the Project dropdown [41].

You can also customize the view of the project files to focus on specific aspects of your app development. For example, selecting the Problems view of your project displays links to the source files containing any recognized coding and syntax errors, such as a missing XML element closing tag in a layout file. We have used the latest version of android studio (figure III.1) to build our application [41].

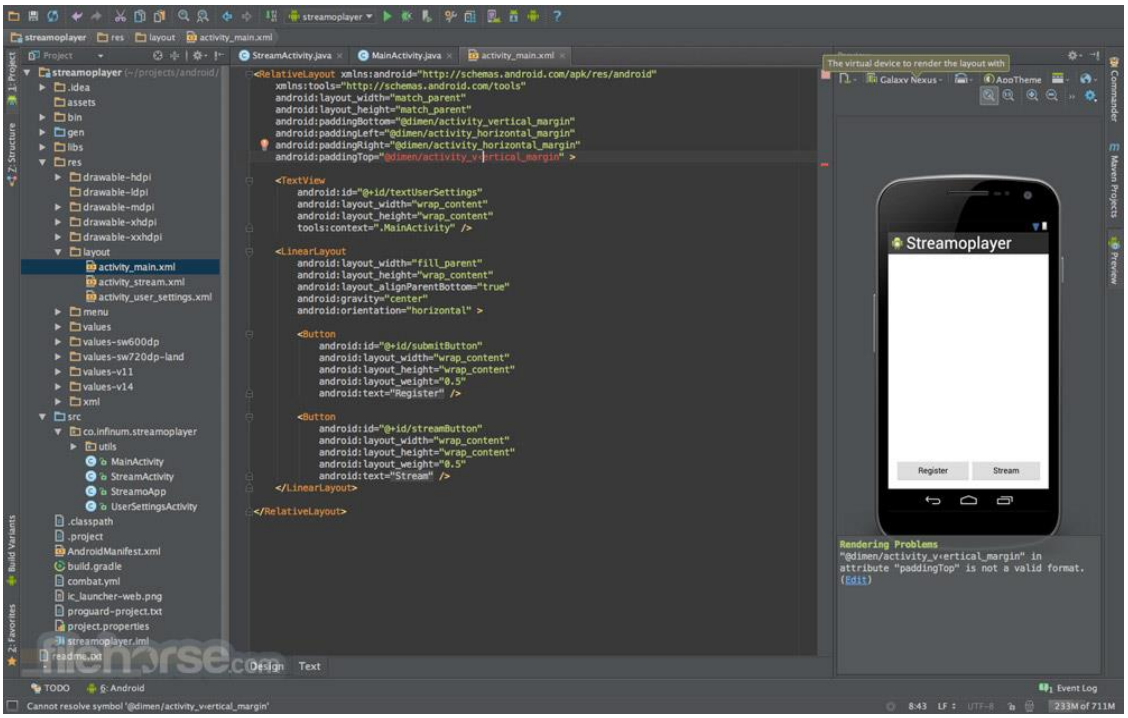


Figure III.1: Android SDK Emulator[41].

III.2.2 Android Application (*Healthy*).

The idea of our Android application, named *Healthy*, is that each patient will use our system, should first install the app in his/her phone to keep track of his heart rate status. The app will send a message to the doctor whenever there is a heart problem and make a sound with notification to the patient to let him know and people around him that he is in danger. *Healthy* will be connected to our hardware device via Bluetooth integrated in the mobile phone and will receive the data comes from the HC06 Bluetooth module. To have a good understanding of how this application works, we will discuss each function a side which will be divided into two parts: the Back-end and the Front-end of the app.

III.2.2.1 Permissions and Requests:

The app *Healthy* has only one permission which is Location permission, and this is for discovering Nearby Bluetooth devices. The application will ask this permission first when we install the app. For the requests, the application will ask the user to turn on his Bluetooth whenever it is off to make sure the connection is working perfectly.

III.2.2.2 "Healthy" Backend functionality:

The backend of our application is implemented using Java programming language. Mainly, there are two classes to handle the connection with the Bluetooth device and t other devices to track the sent data via the Hardware system using a thread to stream the upcoming data.

The first class is AddDevice, as any Android class it has the onCreate method to instantiate the layout and assign the views and initialize the objects, while onResume method handles the check permission and whether the Bluetooth is enabled or not by calling two functions checkPermission() and checkBluetoothEnabled(). Also, there is onStart method which provides the list of the paired and scanned Bluetooth devices. For onDestroy method, the android app will kill the Receiver which is responsible to check the Bluetooth status.

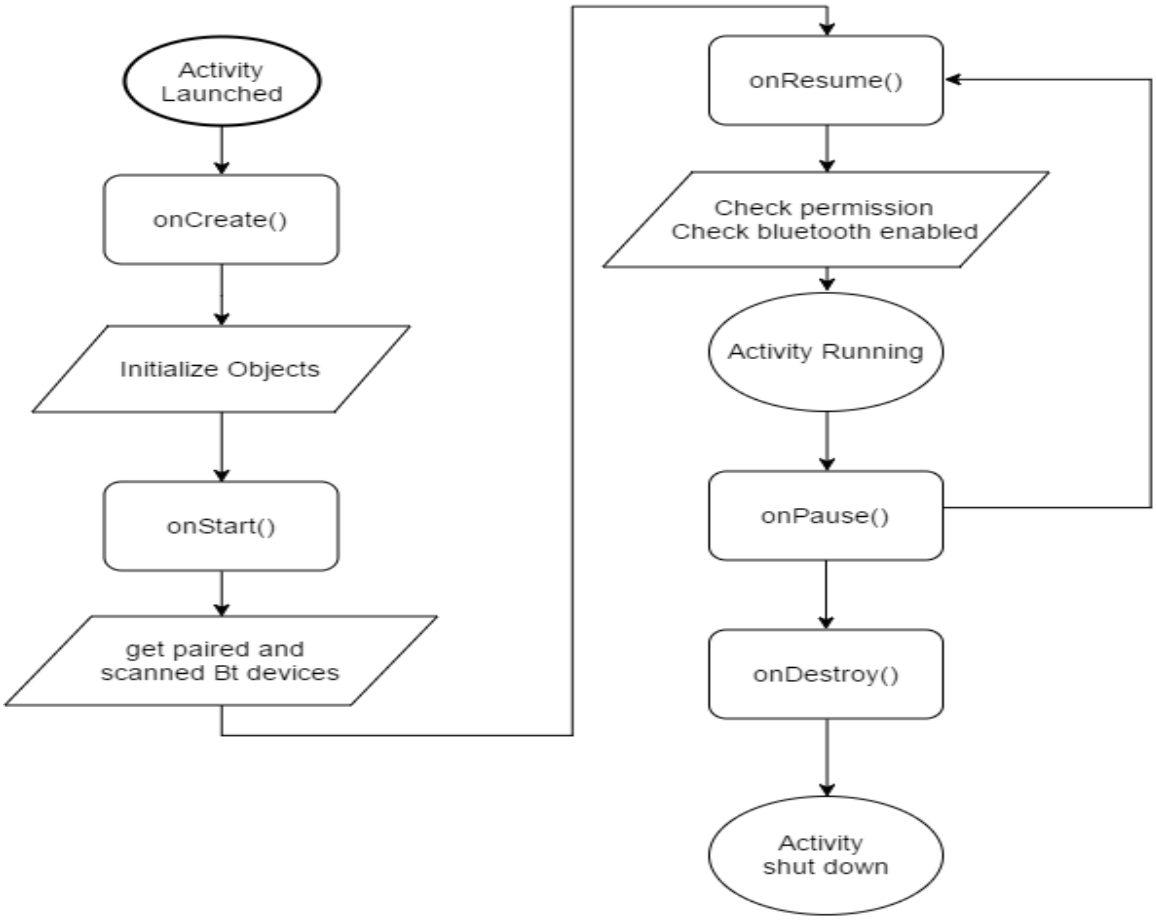


Figure III.2: flowchart representing AddDevice lifecycle.

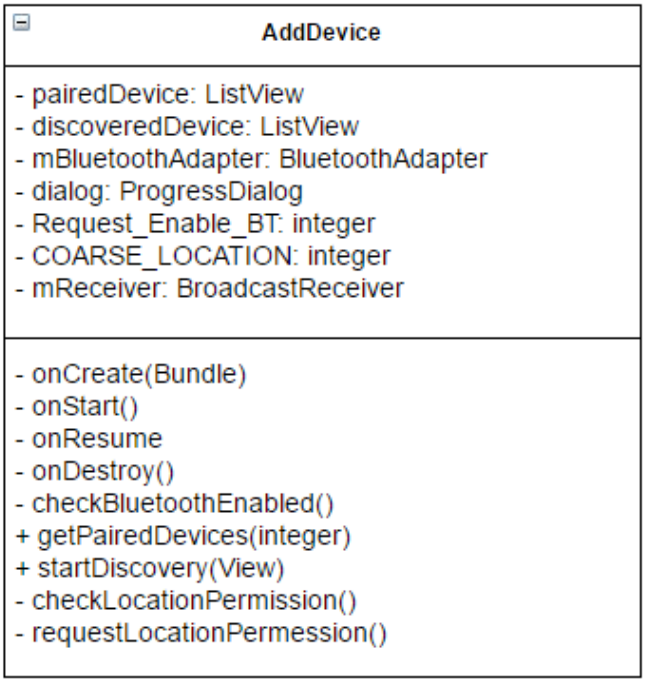


Figure III.3: UML of AddDevice.

From the lifecycle chart (figure III.2) and UML of the class (figure III.3), we can understand that the main function of this class is search and get the list of all possible nearby Bluetooth devices from the user and get the Mac Address of the selected device and pass it to the showData class.

For the second class, which is showData, it handles the data comes from the hardware system and display it to the patient through showData layout and check whenever there is problem in the heart beat rate. This class has two inner classes the first one is connectingThread class and the second is connectedThread class.

ConnectingThread class establishes the connection between the HC06 Bluetooth module and our application and will keep track of the connection whenever something happened. It will notify the user such as Connected, or Connection lost and requests the application to kill the thread and the whole activity. To establish the connection, it will need the Mac Address of the selected device from the AddDevice class. We have used shared preference to save the Mac address and the reason of that is shared preference save the selected device even if we closed the application. In other words, it is a small database with one entry. While connectedThread keeps listening to the data comes from the HC06 module and save it in a byte buffer. Also, showData gets the info of the user

and the current heartbeat and if there is any problem (the hearbeat < 60bpm or the hearbeat > 100bpm), it will send it to the cloud which is in our case Firebase cloud. The following UML diagram (figure III.4) explains how these classes are connected together.

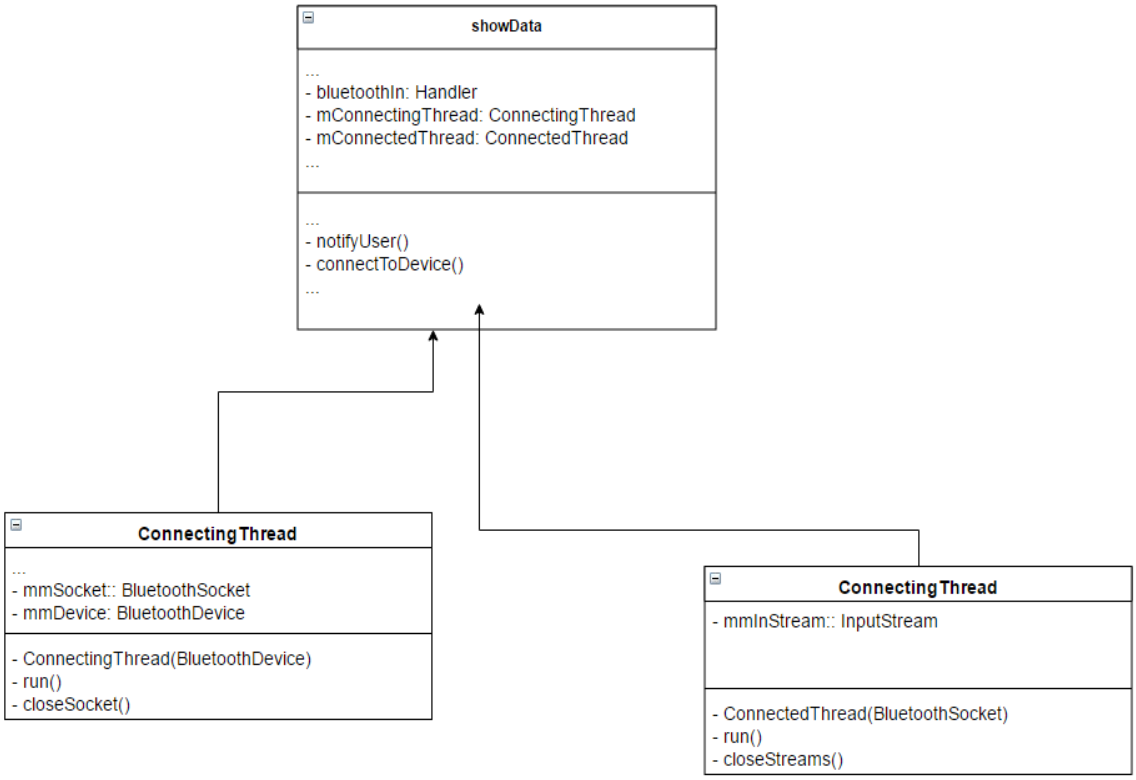


Figure III.4: ShowData UML diagram.

To send a data to Firebase, we have implemented, first, a message handler to extract the input data and check if it goes above the range level, fixed previously between 100 bpm and 60 bpm. The heartbeat rate, the user info and the time of the happened event will be sent to the firebase in one tree where the doctor can check his patients status and respond with the right action.

Finally, we have made a general overview of the important functionalities of each class and summarize them in one flowchart (figure III.5) .

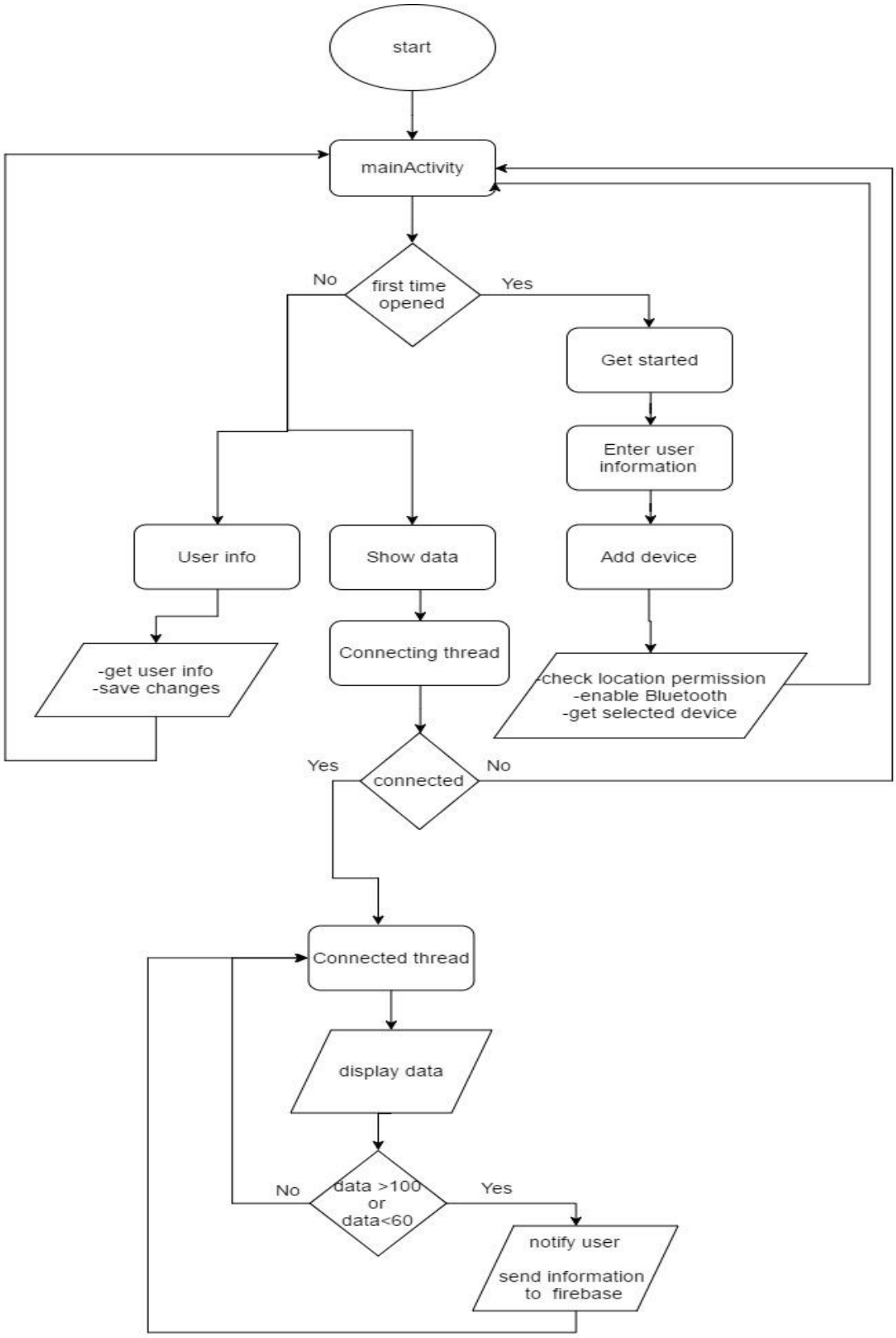


Figure III.5: flowchart representing the backend system of ***Healthy*** app.

III.2.2.3 "Healthy" Frontend functionality:

For the Frontend, we have used XML scripting language to design our interface. We have divided the application into three main layouts in addition to three onboarding layouts. The next figures will show different layouts and all the functionalities that they contain.

- **Onboarding layouts:**

There are three layouts (figure III.6) which will be shown when you install our *Healthy* application and they contain information and the user guide.

Figure III.6: Onboarding layouts.

Another layout is the user info or welcome layout (figure III.7); it will ask the user to insert his information such as full name, address, phone number and description of his situation: the medicines he takes, the date of the surgery he made and any other information that may help the doctor.

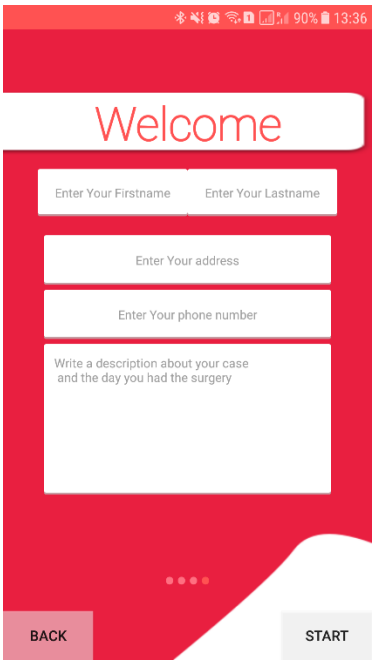


Figure III.7: User Info Welcome layout.

- **Main layout :**

The main layout (figure III.8) contains three buttons that will lead to the sub layouts in the applications and they are : Add device, User info and show data layouts.

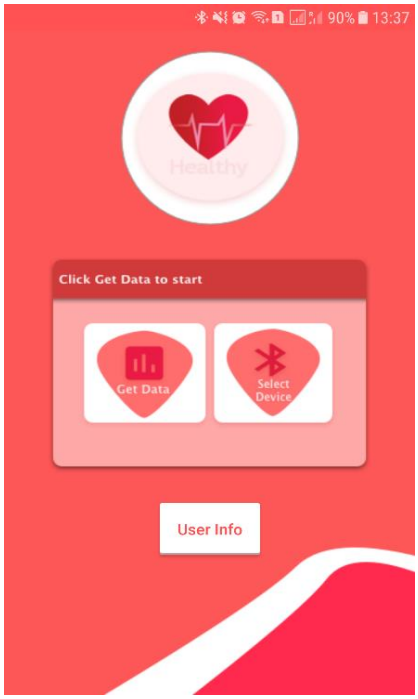


Figure III.8: *Healthy* main layout.

- **Add device layout** (figure III.9)**:**

This layout displays the result of paired and scanned devices we have found, it will show a progress dialog while it scans the nearby devices. After finishing scanning, it will dismiss the dialog and show a text which informs the user that the scan function ended.

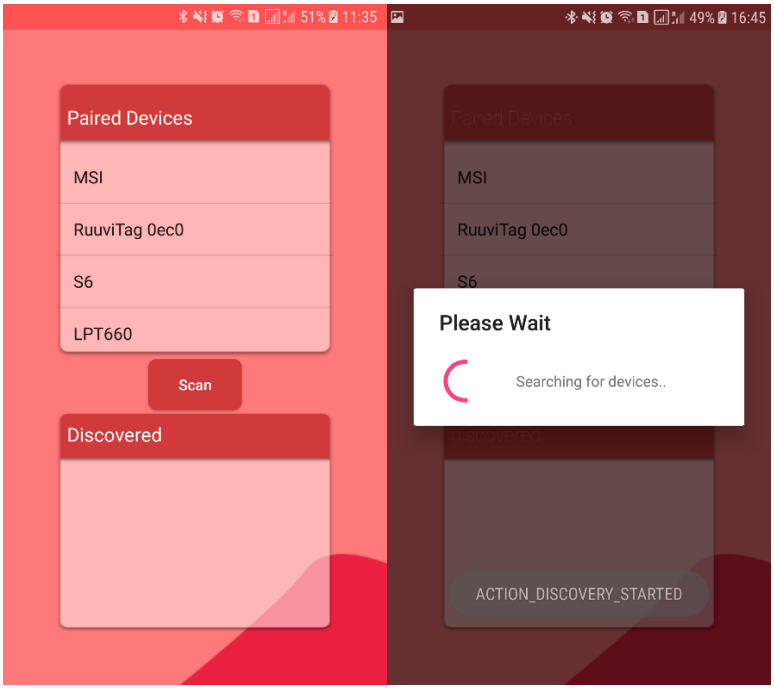


Figure III.9: Add device layout and its functions.

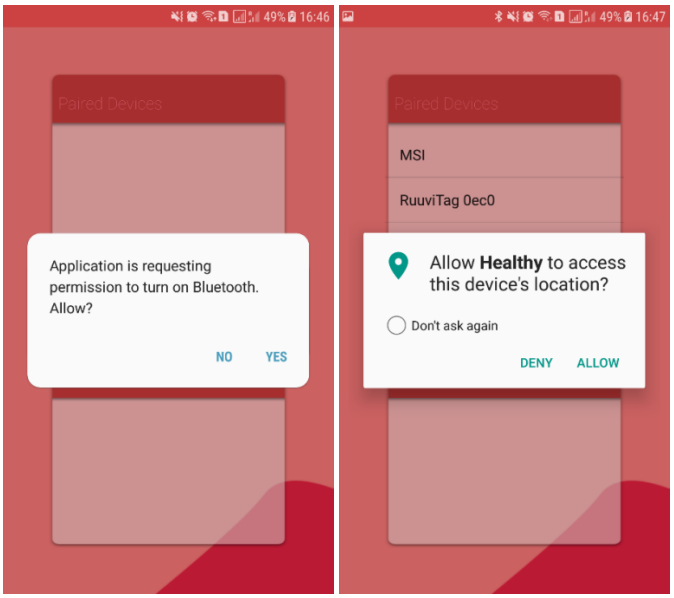


Figure III.10: Permissions and requests handled by Add device layout.

- **Show Data layout** (figure III.11)**:**

This layout is responsible for displaying the data and check the notification of the user whenever there is a problem in his heartbeat rate, and the connection status.

Figure III.11: Show data Layout.

Figure III.12: Alert Notification.

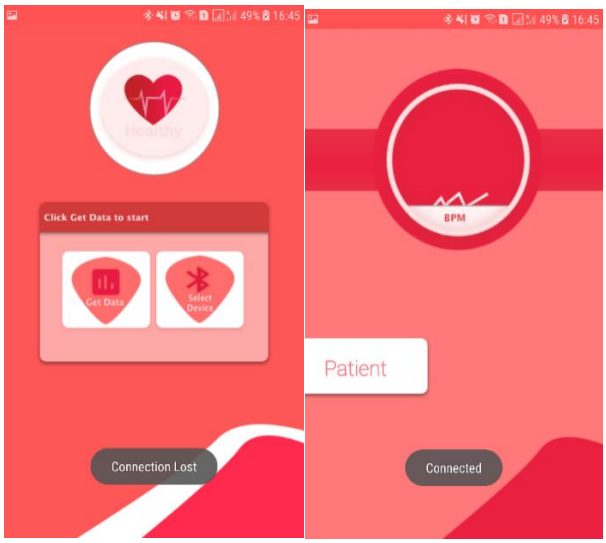


Figure III.13: Connection status.

- **User Info layout** (figure III.14)**:**

This layout is responsible for displaying the user information and providing the user the possibility of changing his/her information by modifying the text and press the save button.

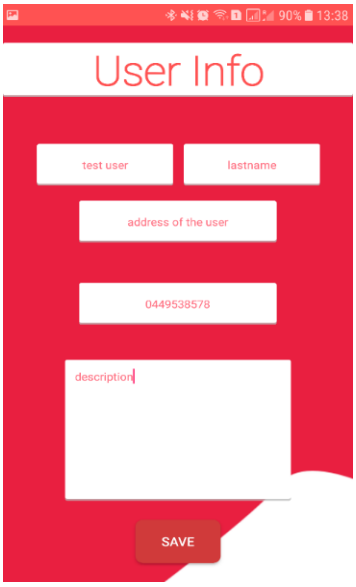


Figure III.14: User Info layout.

III.2.3 Firebase Realtime Database:

The last step in our software implementation phase is setting up the Firebase Realtime Database (figure III.15). The first thing to do is to add the dependencies of the

firebase and the Google services file to the project file. Next, we have initialized the Realtime database from the firebase website and declared a tree in the showData class with a child equals to the username to display all the info of the user in the database. It is important to notice that the user info will be sent to the firebase only if his current heartbeat rate goes above or below the stable range. The following figures (figures III.15 and III.16) show an example of user info provided by firebase cloud system.

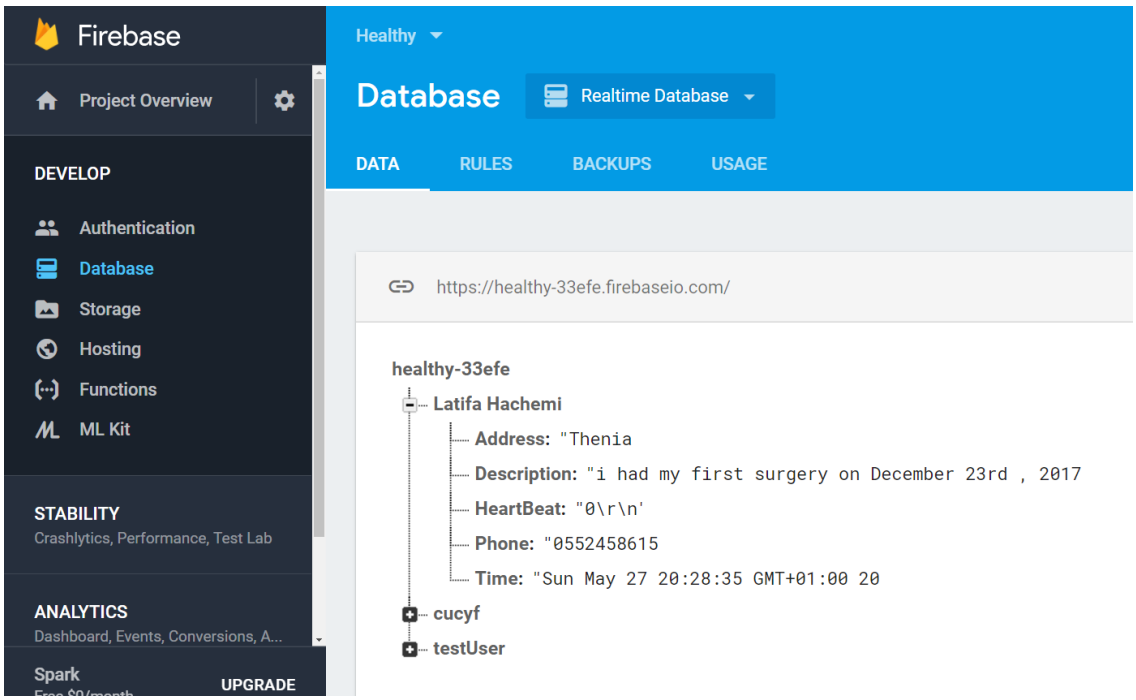


Figure III.15: Realtime database info.

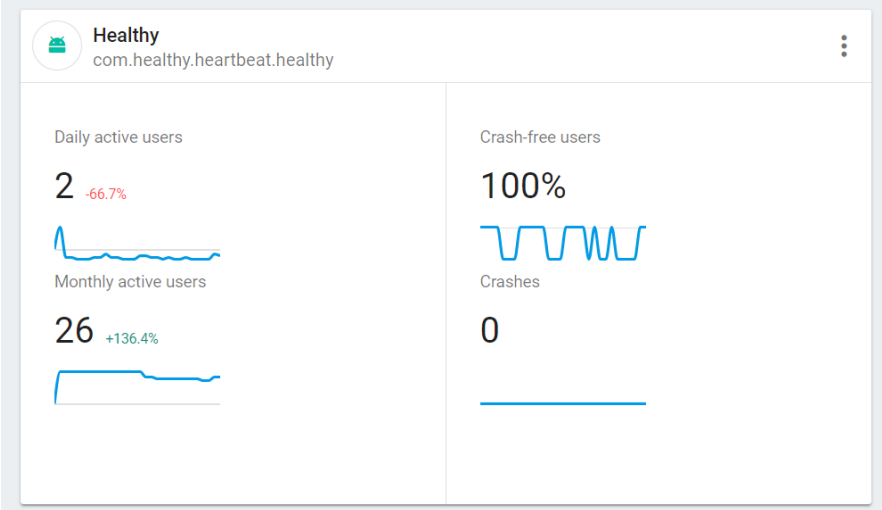


Figure III.16: Application usage info.

III.3 Conclusion:

Finally, after going through all the steps of the software development and the management of our Cloud-based android application design, we got our project to work in real hardware tests. The next Chapter IV will present the different real tests made with all the hardware and the software designs together. In addition, the experimental results will be enumerated and validate the performances of the whole designed system.

Chapter IV

Experimental tests and results

IV.1 Introduction:

In this section, we will present the experiments tests and results of our designed system. We have selected some participants as tests subjects whom were asked to give their feedbacks about our project. In addition, we will discuss all the enumerated results and some notes we have concluded from the system in use.

IV.2 Preliminary tests on the Arduino:

The first step we have made before starting implementing our project was testing our Arduino UNO board. For that, we have used the Serial Plotter in the Arduino IDE to display a sinusoidal signal generated from the function generator and acquired by the Arduino UNO board. The sinusoidal signal has an amplitude of 2.3 V and a frequency of 245 Hz**.** The results have shown that only the positive values were displayed because the Arduino is an ADC converter which has no determination of negative values. For that, we had to use a function called *map* () which allows the user to change the range of the input signal from the original range (0 to 1023) to any specified range. By using *map* () function, the result was great where the plotter displayed the exact signal amplitude (figure IV.1). The software solution has allowed avoiding the use of a summator circuit which shifts the negative values to the positive range, and minimizes the circuit size.

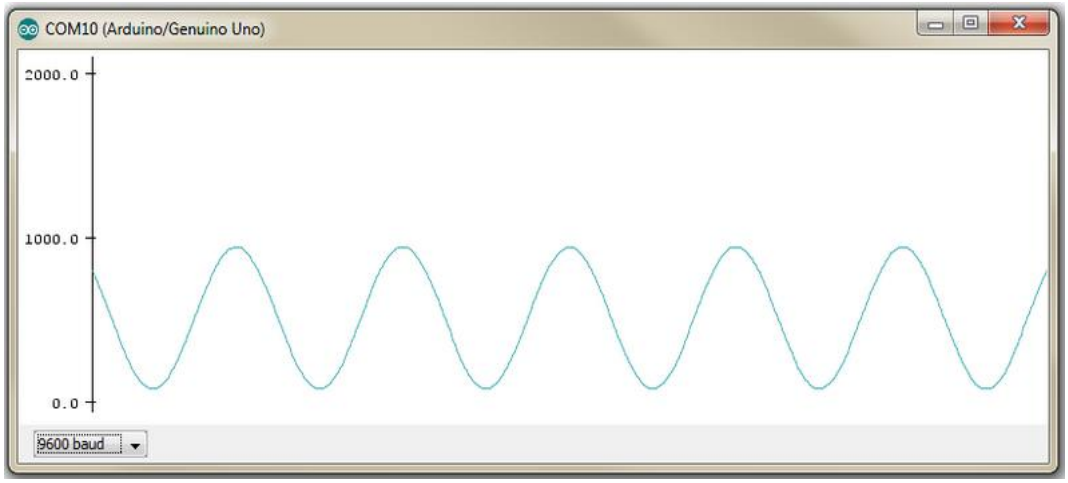


Figure IV.1: Acquired sinusoidal signal on the Serial plotter of the Arduino IDE.

IV.3 Testing and results of the system:

We have passed our system through two different tests to see whether it works perfectly or needs some extra work adjustment Figure IV.2. The first test was displaying the signal in our laptop to check the functionality of the system and its accuracy. The second test was calculating the heartbeat rate and displaying the results in the mobile phone and the cloud.

Figure IV.2 : Test set up and User testing the signal before sending data to the app.

The ages of the tests subjects are 24 and 21 repectively. They were asked to try our system and to make both tests that we have mentioned before.

IV.3.1 First test (Displaying the signal):

In this test, the participants were asked to put on the disposable electrodes in their both right and left arms as measuring electrodes and their right leg as a ground electrode. We have asked them to breathe naturally. Figures IV.3 and IV.4 show the acquired ECG signals displayed on the Serial plotter of the Arduino IDE.

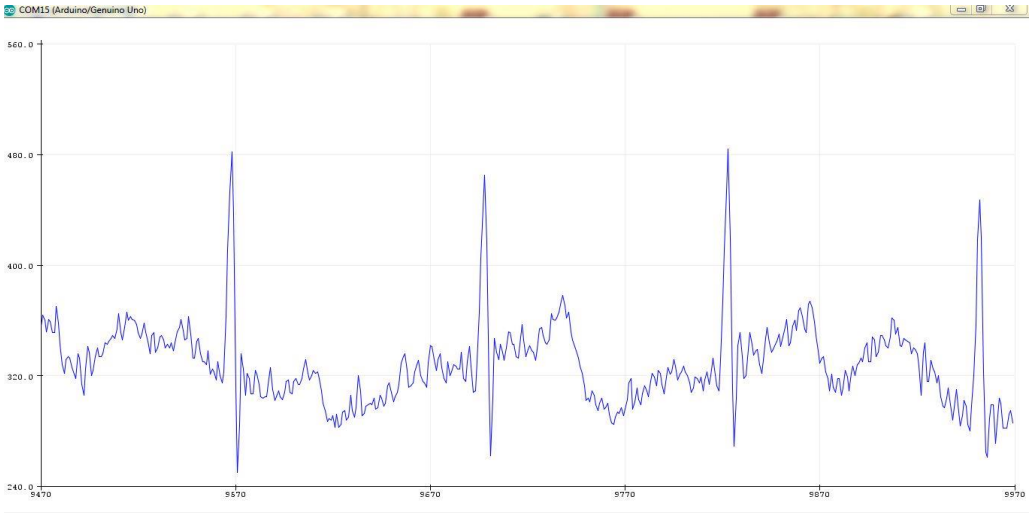


Figure IV.3: Displayed ECG signal of participant 1.

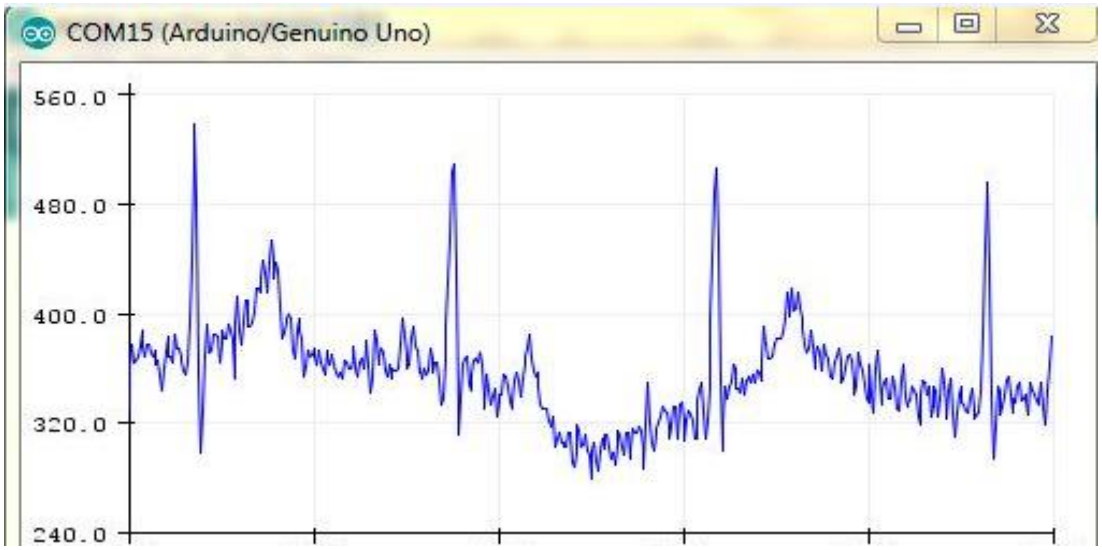


Figure IV.4: Displayed ECG signal of participant 2.

IV.3.2 Second test (Sending the data):

In this test, we have asked the participants to install our *Healthy* application in their mobile phones. The reason of that is to send their heartbeat in Realtime from our system and display it into their mobile phone. We gave them the following tasks of the tests protocol :

- Put the electrodes in right leg and both arms.
- Turn on the Bluetooth in the mobile device.
- Open *Healthy* mobile application.
- Enter user information.
- Press scan button in the select device layout to discover nearby Bluetooth devices.
- Select HC 06 from the list if there is any.
- Start receiving data by pressing "Get Data" button.
- Check Firebase website if there is any notification of a heartbeat problem.

✓ **Subject N°1:** Subject has heart rate less than 60 bpm:

The people we've tested on as tests subjects are healthy people with no previous cardiac issus history. Our application sends the heartbeat to the Cloud only in case that it is out of the normal range specified before. For that, we intentially disconnected one electrode so the heartbeat could be less than 60bpm. In this case , the data will be send to the Cloud otherwise the app doesn't send anything.

The following figures. IV.5 to IV.11 illustrate: the test set up, the heartbeat displayed on the Serial plotter of the IDE, the heartbeat displayed in *Healthy* app, the heartbeat values displayed in Firebase web version and the heartbeat values displayed in firebase android version respectively for Subject N°1.

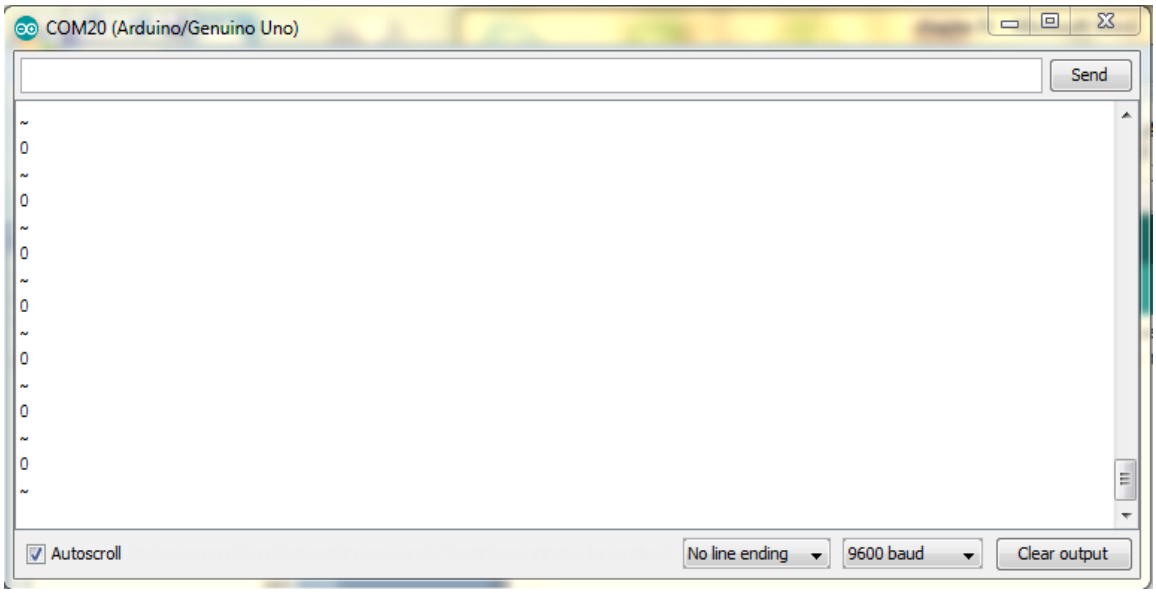


Figure IV.5: Heartbeats displayed in the serial monitor.

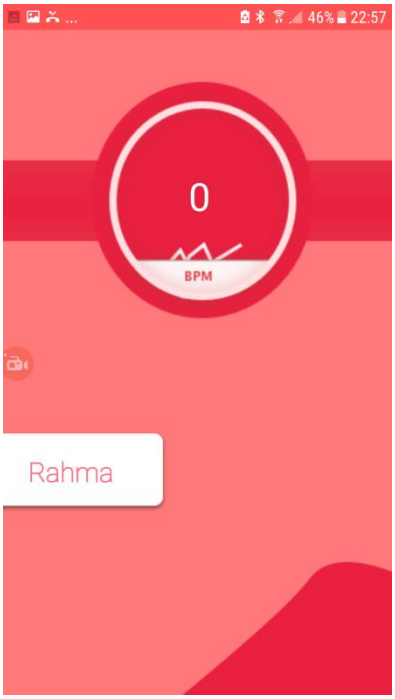


Figure IV.6: Heartbeat displayed in *Healthy* app.

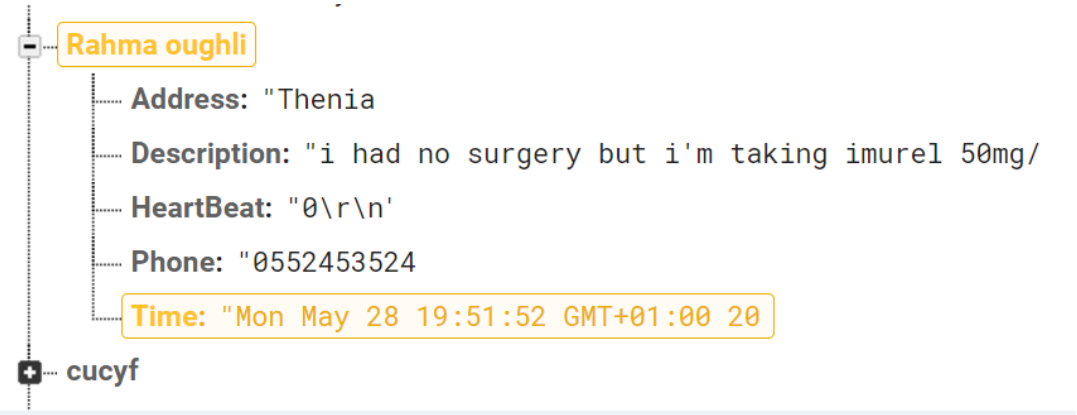


Figure IV.7: Heartbeat values displayed in Firebase web version.

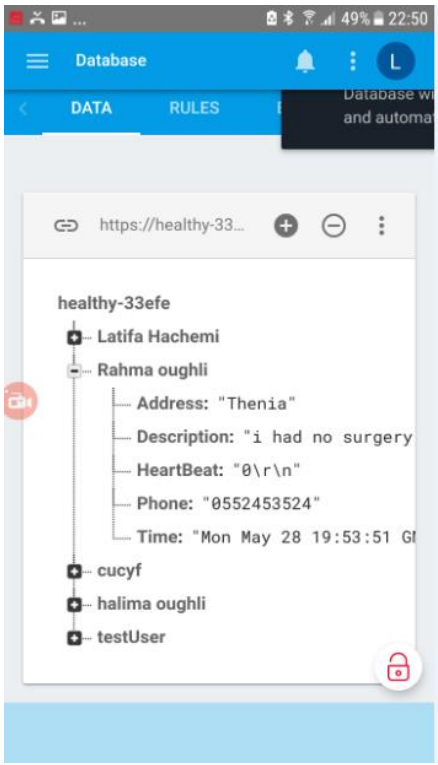


Figure IV.8: Heartbeat values displayed in firebase android version.

✓ **Subject N°2 :** Subject has normal heart rate (60<heartbeat<100) .

The figures IV.9 and IV.10 show the heartbeats displayed in the serial monitor of the IDE and the heartbeat displayed in *Healthy* app respectively. The data didn't be send to the Cloud because there is no heartbeat issu with that test subject.

Figure IV.9: Heartbeats displayed in the serial monitor.

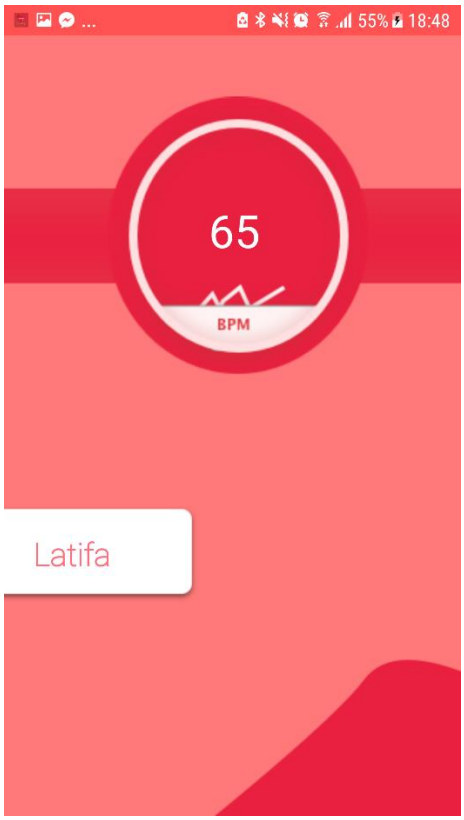


Figure IV.10: Heartbeat displayed in *Healthy* app.

IV.4 Discussions:

After going through the conducted experiments, we have made the following analysis:

IV.4.1 First test subject:

This test subject doesn't have any heartbeat problems, so we have disconnected one of the electrodes in order to stimulate the Cloud and send a zero result. The experimental results have displayed the user data (name and heartbeat) in our android application *Healthy* and the cloud as well. Also, the test subject (the user) has got an alert notification in her phone.

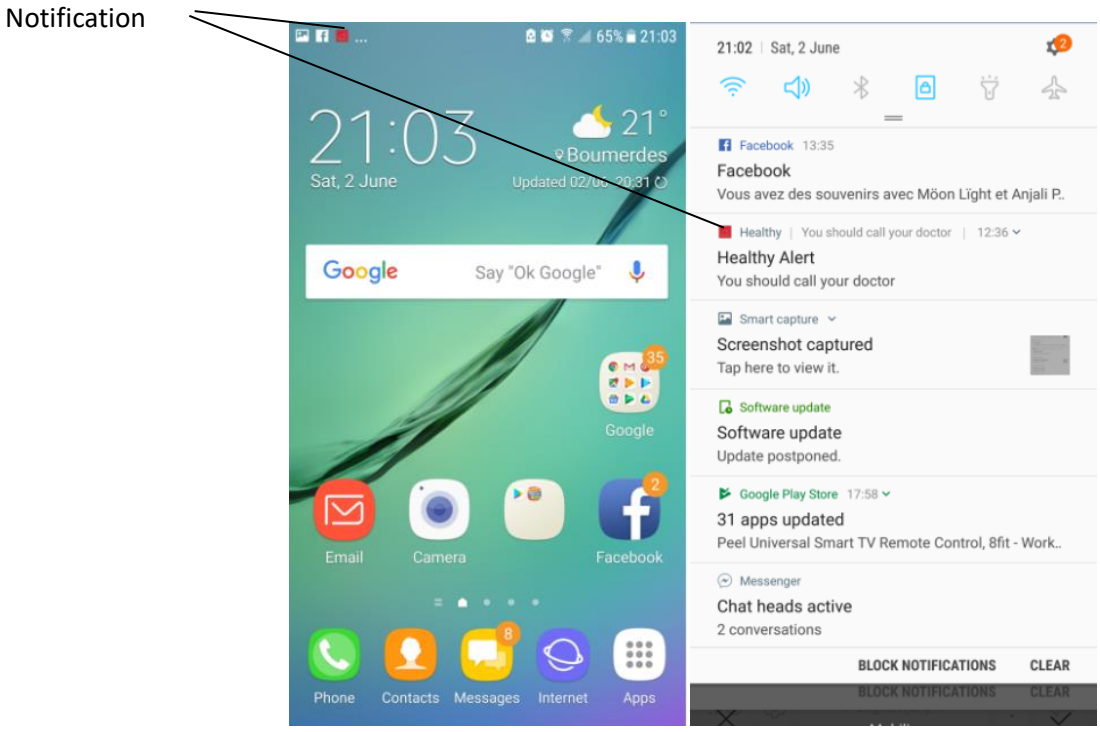


Figure IV.11 : mobile notification.

IV.4.2 Second test subject:

This subject has a healthy heart rate; so our system didn't respond by sending any notification to the cloud. It just displayed the value on the *Healthy* android application.

After testing our system with the two subject tests, we have got satisfied results that confirm the functionality of our system. The participants have a good impression about our system and were satisfied with the results we got. Moreover, the participants felt safe and secure because the *Healthy* app receives the data continuously from the hardware (the electronic part: electrodes, AD8232 ECG amplifier board and Arduino). The participants appreciate the fact that the *Healthy* app sends the heartbeat to the Cloud and they got notified by an alarm whenever it detects a problem with their heartbeat.

The most notes we can get from these experiments are:

- The system has succeeded in reading the signal from different users.
- The system has succeeded in sending the data to the *Healthy* application in the fastest way.
- The *Healthy* Application didn't crash at all while the testing phase.
- The *Healthy* application succeeded in connecting with the system easily.
- All the users agreed about the fact that the application is simple and easy to use by everyone.

IV.5 Financial aspect:

We wanted to evaluate the financial aspect of the designed system. For that, our project is divided into the hardware and the software parts.

In the hardware part, we give the different components and modules we have used with their unit price. The components and their unit price are in the following:

- Arduino Uno module: unit price 4500 DZD.
- AD8282 Heart rate monitor: unit price 6000 DZD.
- HC-06 Bluetooth module: unit price 2500 DZD.
- Electrodes: unit price 650 DZD (40 electrodes) .
- Jumpers: unit price 400 DZD (10 jumpers) .

In the software part, we precise the number of hours we spent in programming the Arduino and the android application.

According to researches, a java developer gets about 46.87 $ per hour, but in Algeria he gets about 412.5 DZD/hour. So, for a total work of 120 hours we spent in programming our system, we get: 49500 DZD when considering the Algerian cost.

Finally, the total cost of the project design (hardware and software) is about: 63550 DZD.

The total cost of the project design is the cost of one prototype which is usually expensive because it is the developed product. If we consider commercializing this system, the series

manufacture of this system would be much cheaper than the initial developed prototype. Usually, those products are made customized to the patients and allow minimizing the hospitalization duration and cost, even the discomfort relied to it.

IV.6 Conclusion:

The experimental results we have got from the different tests and evaluation phase confirms that our ECG wireless transmission system is simple and easy to use for everyone. The system has proved that we can remotely track easily the patient heartbeat and stay updated and ready to any emergency that can occur in anytime without moving to any hospitals or clinics. This fact saves us money, time and can save patients' lives in the right moment.

Conclusion

Heart disease is the most prevalent cause of early death in most developed countries. It may be predicted but it’s hard to know when exactly. Furthermore, it is difficult to track all the patients at the same time. In this project, we have designed a Cloud-based electrocardiogram device prototype that can help the doctors to monitor the patients' heart and make sure that they have a good heartbeat rate and take actions whenever there is any problem occurred.

The objective of this project is to implement a system that reads the current subject heartbeats, send them in Realtime and display them in an Android application *Healthy* which keeps track of the level of the heartbeat and notify the doctor in the firebase-cloud whenever there is a heartbeat problem.

Regarding the experimental results from the evaluation phase, we have got satisfied results that confirm the functionality of our system. Moreover, the participants felt safe and secure because the *Healthy* app receives the data continuously from the hardware (the electronic part: electrodes, AD8232 ECG amplifier board and Arduino). The participants appreciate the fact that the *Healthy* app sends the heartbeat to the Cloud and they got notified by an alarm whenever it detects a problem with their heartbeat.

During this project we've encountered some difficulties like:

- Sending the heart beats values from our *Healthy* Android app to the cloud because of the lower internet flow connection in Algeria even all mobile and phone operators are implementing the 3G or 4G.
- We couldn’t plot the ECG signal sent to the cloud because of the limited memory of Arduino that needs an extra memory (microSD memory card).

Even, we had few problems with the implementation of the system, but we've learned many things during this journey like better dealing with the Arduino Uno and the Android studio.

Future Work:

In order to make our system more reliable and productive, we propose some modifications and implementing other functions that provide both the doctors and the patients the ability to monitor the patient's heart status. Such as plotting a Realtime graph in the cloud or send the user's data for the past three days by adding a microSD to the Arduino to store the data and send it whenever there is any problem to the cloud. Another suggestion is to add an email notification that notifies the doctor if there are any changes in the database. The last suggestion is to add a GSM module that keep track of the user's place and send his/her coordinates to the hospital whenever he had a heart attack in order to make sure that the patient will have a quick treatment and avoid any problem that may lead to his/her death. Finally, the aim of this entire project is to contribute in improving the quality of life for people having heart issues.

Bibliography

[1]. Cardiovasculardiseases (CVDs). (n.d.). Retrieved May 4, 2018,fromhttp://www.who.int/mediacentre/factsheets/fs317/en/

[2]. blog-overview-heart-disease-modern-societys-common-killer. (n.d.). Retrieved May 10, 2018,from http://www.sciencecare.com/blog-overview-heart-disease-modern-societys-common-killer/

[3]. HeartDiseaseFacts&Statistics | cdc.gov.(n.d). RetrievedMay 10, 2018,from https://www.cdc.gov/heartdisease/facts.htm

[4]. Dr.MezianeNacira.''Contribution à la faisabilité d'une bioinstrumentation embarquée à base d'électrodes sèches dédiée à l'électrocardiographie ambulatoire.N°d'ordre : 02/2015-D/EL.Thèse de doctorat , USTHB University.

[5]. Dennis Joe HarmahKathirveluD.''Anubiquitousminiaturizedandroidbased ECG monitoring systems '',2013 IEEE International Conference on Emerging Trends in Computing , Communication and Nanotechnology (ICECCN 2013)

[6]. KavitaS.Patil and Rohita.HJagdal.''Remote experts help in case emergency : A cloud computing solution for heartdisease patient'' . International journal of computer applications(0975-8887).Vol. 77-No, 17.2014

[7]. Home Health Care - - American Family Physician.(n.d). RetrievedMay 10, 2018,from https://www.aafp.org/afp/1998/1101/p1608.html

[8]. Mini Wireless ECG for Monitoring Athletes' ECG Signal. (n.d). RetrievedMay 10, 2018,from http://www.iosrjen.org/Papers/vol4_issue6%20(part-1)/C04611318.pdf

[9]. Cardiovasculardisease.(n.d). RetrievedMay 14, 2018 ,from https://www.nhs.uk/conditions/cardiovascular-disease/

[10]. (n.d.). Retrieved May 25, 2018, fromhttps://www.nhs.uk/conditions/Cardiovascular-disease/

Bibliography

[11]. Cardiovascular disease - Medical News Today.(n.d). Retrieved May 10, 2018,from https://www.medicalnewstoday.com/articles/257484.php

[12]. Electrocardiography.(n.d). RetrievedMay 19, 2018,from https://en.wikipedia.org/wiki/Electrocardiography

[13]. The heart is one of the most important organs in an individual's body. It acts like a dual-chambered pump that circulates. (n.d.). Retrieved May 26, from https://www.yourgreenpal.com/blog/your-engine-s-lifeblood-what-type-of-oil-should-you-use

[14]. The Parts of the Heart and TheirFunctions | Life Persona.(n.d). RetrievedMay 9, 2018,from https://www.lifepersona.com/the-parts-of-the-heart-and-their-functions

[15]. .(n.d). Retrieved May 9, 2018,from https://encryptedtbn0.gstatic.com/images?q=tbn:ANd9GcRxqrRRkj0gDiLA2EJ2x7pM-a2pPUFcTY27rUk7pFW33GQXDfjY4w

[16]. Chapter29: Origin of the Heartbeat& the Electrical Activity of the Heart.(n.d). Retrieved May 16, 2018,from http://accessmedicine.mhmedical.com/content.aspx?bookid=1587&sectionid=105778150

[17]. Fundamentals of Electrocardiography Interpretation.(n.d). Retrieved May 10, 2018,from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1614214/

[18]. Waves and Intervals on the ECG - ECG .(n.d). RetrievedMay 20, 2018,from http://www.medicine-on-line.com/html/ecg/e0001en_files/05.htm

[19]. 12-Lead ECG Placement .(n.d). RetrievedMay 14, 2018,from http://www.emtresource.com/resources/ecg/12-lead-ecg-placement/

[20]. ABC of clinicalelectrocardiography: Introduction.(n.d). RetrievedMay 11, 2018,from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1122339/

[21]. Field 12 Lead ECG Diagnosis.(n.d). RetrievedMay 1, 2018,from http://www.publicsafety.net/12lead_dx.htm

[22]. .(n.d). RetrievedMay 17, 2018,from https://www.nottingham.ac.uk/nursing/practice/resources/cardiology/images/bipolar_triangle02.gif

[23]. .(n.d). RetrievedMay 1, 2018,from http://nursingcrib.com/wp-content/uploads/2014/03/ecg-lead-placement-520x245.jpg

[24]. What Is a Normal Heart Rate? .(n.d). RetrievedMay 5, 2018,from https://www.livescience.com/42081-normal-heart-rate.html

[25]. https://en.wikipedia.org/wiki/Cloud_computing

[26]. Tutorials Point. (2018, January 08). DCN Wireless Transmission. Retrieved May 10, 2018, from https://www.tutorialspoint.com/data_communication_computer_network/wireless_transmission.htm

[27]. Heart rate.(n.d). RetrievedMay 10, 2018,from https://en.wikipedia.org/wiki/Heart_rate

[28]. .(n.d). RetrievedMay 29, 2018,from https://ay12-14.moodle.wisc.edu/prod/pluginfile.php/189091/mod_resource/content/1/AD8232.pdf

[29]. Bobbie, P. O., Arif, C. Z., Chaudhari, H., & Pujari, S. (2004). Electrocardiogram (EKG) data acquisition and wireless transmission. WSEAS Transactions on Systems, 3(8), 2665-2672.

[30]. Arduino - Introduction. (n.d.). Retrieved May 10, 2018,from https://www.arduino.cc/en/Guide/Introduction

[31]. Arduino Microcontroller Feature Comparison. (2013, May 07). Retrieved from https://www.robotshop.com/blog/en/arduino-microcontroller-feature-comparison-2-3631

[32]. Arduino - Environment. (n.d.). RetrievedMay 8, 2018, fromhttps://www.arduino.cc/en/Guide/Environment

[33]. HC-06 Bluetooth Module. (n.d.). RetrievedMay 10, 2018, from https://www.sgbotic.com/index.php?dispatch=products.view&product_id=2471

[34]. Serial Port Bluetooth Module (Slave): HC-06. (n.d.). Retrieved May 10, 2018,from https://www.itead.cc/wiki/Serial_Port_Bluetooth_Module_(Slave):_HC-06

[35]. Bluetooth Transceiver Module HC-06. (n.d.). Retrieved May 10, 2018,from http://wiki.sunfounder.cc/index.php?title=Bluetooth_Transceiver_Module_HC-06

[36]. Heart Rate Monitor AD8232 Interface Arduino. (2017, December 01). Retrieved May 10, 2018,from http://www.theorycircuit.com/heart-rate-monitor-ad8232-interface-arduino/

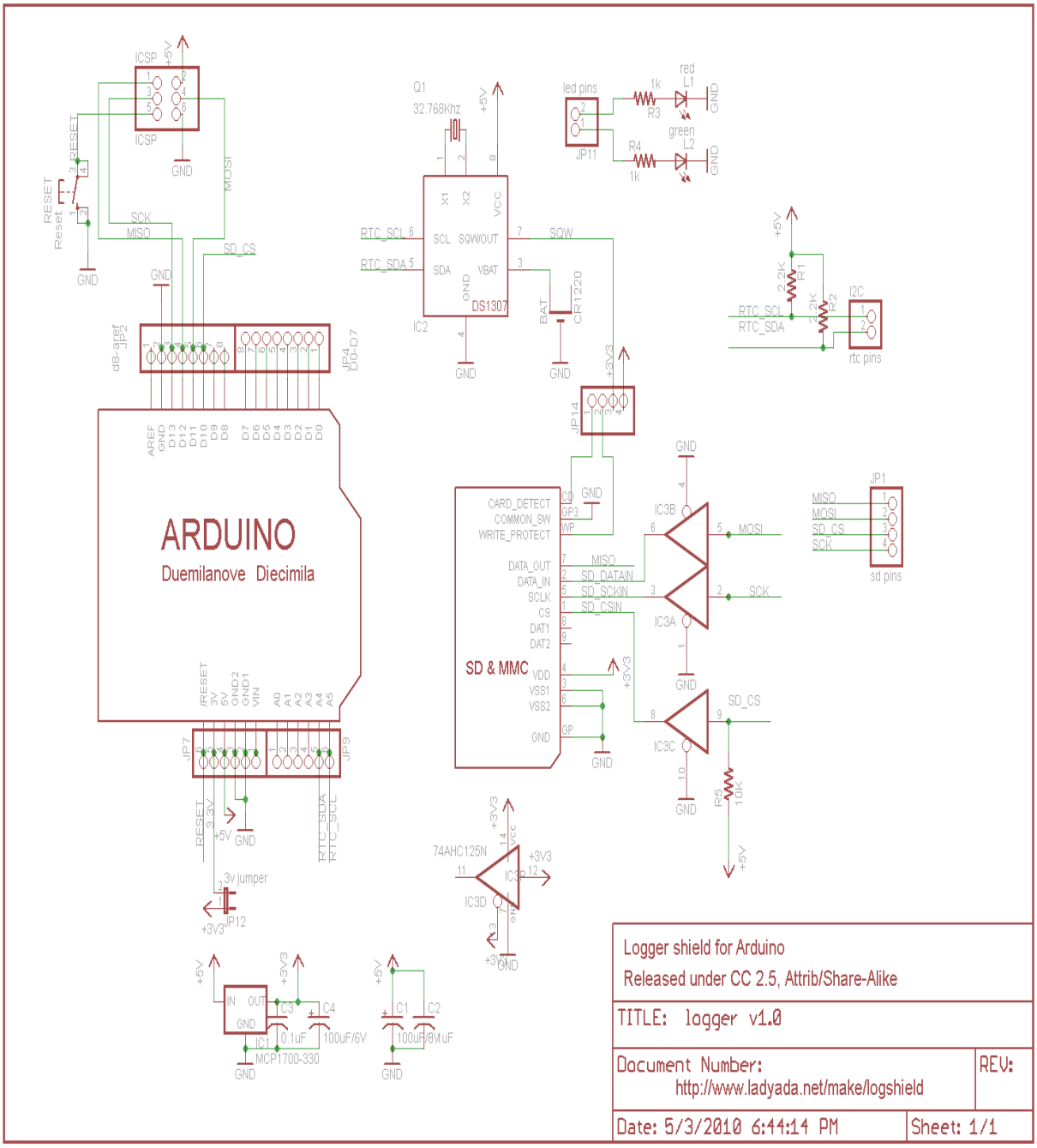
[37]. Cloud Functions for Firebase |Firebase. (n.d.). RetrievedMay 23, 2018,fromhttps://firebase.google.com/docs/functions/

[38]. Firebase Realtime Database | Firebase Realtime Database | Firebase. (n.d.). RetrievedMay 10, 2018, from https://firebase.google.com/docs/database/

[39]. Realtime Database Limits | Firebase Realtime Database | Firebase. (n.d.). Retrieved May 10, 2018,from https://firebase.google.com/docs/database/usage/limits

[40]. Meet Android Studio | Android Developers. (n.d.). RetrievedMay 10, 2018,fromhttps://developer.android.com/studio/intro/

Arduino UNO Reference design :



AD8232 Reference design :