BACHELOR'S THESIS / BIOMEDICAL ENGINEERING 2018

# Development of an Educational Tool for Building Biomedical Devices and Analysis the Collected Data

Joana Gelabert Xirinachs

Universitat
Pompeu Fabra
Barcelona

# Development of an educational tool for building biomedical devices and analysing the collected data

Joana Gelabert Xirinachs

BACHELOR THESIS UPF / 2017-2018

THESIS SUPERVISOR

Tenured professor Javier Macia PhD

Universitat Pompeu Fabra
Barcelona

To my family. Thank you for being an immeasurable and indispensable source of care.

# Acknowledgments

Javier Macia has been an unconditional support during the whole project. His dedication to find a clear definition of the initial idea and his help during its development has been crucial. He has provided all the programming assistance needed and honest advice on how to shape the multiple ideas and motivations into the final result of the thesis.

The kind help from Veronica Moreno, who is part of the *Unitat de Suport a la Qualitat i la Innovació Docent* from the *Escola Superior Politècnica* and the *Departament de Tecnologies de la Informació i les Comunicacions*. Her recommendations were essential to develop the educational content of the thesis.

# Abstract

Biomedical Engineering is nowadays an emergent and rapidly growing knowledge field. Combining medicine and biology with technology can lead to the creation of life-changing applications. The Do It Yourself (DIY) community is also taking part in this area of expertise and several biotechnological open-source projects can be found on the Internet. This thesis aims to provide a tool to allow even non experimented individuals to understand what a biomedical device is and how can it be built and has been developed in accordance with the open-source and communal mentality.

This project performs a thorough development of the educational application of the tool. This tool is a program that facilitates High School teachers to introduce biomedical engineering and, specifically, biomedical devices to young learners. The project includes an interactive guide that explains how to connect different biosensors to an Arduino Uno board, the automatic creation of the code needed for the resulting device to function and an on-line platform to analyse the data recorded from the sensors stored in a database.

In order for the project to become a fully functional educational tool, all the interfaces created are collected into a teaching unit, which is a lesson plan where all the activities to be performed in the classroom are explained. Its main objective is to provide the young students an appealing and interactive project proposal.

# Keywords

DIY, High school, sensors, biomedical device, data analysing, teaching unit.

# Prologue

The Internet offers almost unlimited communication possibilities. The World Wide Web allows a ridiculously huge amount of information to be stored and available for almost anybody. It is interesting, not to say necessary, to create communal knowledge sites and share relevant information about any field of expertise.

Technology is in constant growth. The open-source model of creation decentralizes software development projects and encourages collaborative building and upgrading of the previous available content. It could be enriching to extend this methodology to fields closer to science. There are obvious material limitations for bringing scientific development and research to interested individuals or small collectives, but there are also plenty of projects that can be carried out in an open and cooperative way.

Not only reconsidering the way knowledge is created can be beneficial but also rethinking the way this knowledge is transmitted. New educational methodologies are starting to use project-based lessons to attract the students’ attention and motivation. Combining these techniques with academic curriculums that include the latest progress of science and technology could lead to a significant and favorable scientific and social impact.

# Index

Page.

# List of figures

# 1. INTRODUCTION

Do It Yourself (DIY) methodology is defining a growing biotechnological social movement where open-source projects allow individuals or small organizations to create powerful and accessible tools. The growing communal knowledge available on the Internet has reached Biomedical Engineering, which is nowadays in the spotlight of science and technology. This thesis adopts this working philosophy and aims to contribute by developing a program that allows creating biomedical devices combining multiple sensors.

## 1.1 State of the Art

There are several open source projects already developed that use a microcontroller board and some sensors to collect biometric signals. Internet forums are full of information about how to build and program, for example, a sensor of the heartbeat on the finger using an LED and a photo-transistor [1] or a lie detector [2].

In terms of educative DIY technology projects, several tools to allow introducing creativity and technology in the young learners' teaching plan already exist. One of the most popular and interesting is Scratch [3]. Scratch is a visual programming language and online community targeted primarily at children where the users can create their own interactive stories, games and animations. It has been developed by the Lifelong Kindergarten group at the MIT Media Lab. In other words, it allows children to have a first experience in programming in a friendly and attractive environment that might capture their attention and might motivate them to explore further the possibilities of developing programs.

Scratch is a reference for this project, as it wants to awake in young learners the curiosity for biomedical devices providing them a first contact where they can actually create something by themselves.

The novelty, innovation and powerfulness provided by this project come from combining the open-source projects that collect biometric signals and the development of a user-friendly teaching tool. There are currently no projects that include such a versatile sensor choosing facilities, an elaborate data transfer and interpretation platforms. Thus, this project initiates a new path for bringing biotechnological DIY activities to the open-source community and the High School classrooms simultaneously.

## 1.2 Objectives

The main objective of this project is to create a program that allows designing biomedical devices combining different sensors connected to a single Arduino Uno board. Moreover, the program aims to be part of the collaborative projects of the biotechnological DIY community. Thus, it has to permit the upgrading of its content.

One of the main applications considered for this project is the educational. This program has been developed in order to help High School teachers to introduce biomedical engineering to their teenager students. More precisely, the programs developed in this project must allow students to understand what a biomedical device is, what can it be useful for and how can it be built. In order to achieve this purpose, the project includes the development of an intuitive graphical user interface that creates the code necessary to run the device designed by the students and a clear and useful platform to interpret the data obtained. The intuitiveness of the interfaces to be handled by the user is a crucial objective as it is addressed to young adults with no background in biomedical engineering or programming.

Furthermore, one last purpose of the project is to create the teaching unit that will allow bringing the materials developed to the classroom, which includes defining the activities that will lead the student to understand and acquire the knowledge of the topic. They will consist in doing some research about illnesses or medical conditions that can be diagnosed, treated or avoided combining the sensors available in the project, which are a heart activity sensor, a sound sensor, an accelerometer and a temperature sensor, or adding new ones to the platform. Thus, the tool designed, has the final goal of teaching students how to build a biomedical device, allowing them to program it, use it and interpret the results.

The methodology of the project will be presented using the perspective of the educational application but the utility of the program created, as presented in the main objective, is not necessarily limited to it.

## 1.3 Fundamentals

A medical device, following the definition from the World Health Organization [4], is an instrument, machine or software designed to be used alone or combined with other appliances to satisfy medical purposes such as diagnosis, prevention, monitoring, treatment or alleviation of a disease or injury; replacement or modification of physiological processes or the anatomy; maintenance of other medical devices; or a tool useful for providing information through examinations of the subject. Pharmacological, immunological or metabolic means may be a complementary tool to use but are not considered to be primary intended actions of a biomedical device.

The devices created through this project are an educational resource that allow young learners to understand what a medical device is but are not intended to be used as one. The process of validation is out of the scope of this project as it is not designed to be actually used in a medical environment but in a simulation of one in the classroom.

# 2. METHODOLOGY

This project includes the development of two interactive interfaces and four other programs that allow transferring data when necessary. In addition, a teaching unit that explains how the project can be introduced into the High School program is created as a resource for educators. The first part of the project consists of the designing of the biomedical device, its assembly and the programming needed to make it start collecting data. The first interactive interface is used here and it works as a guide to help the user connect in a correct way the sensors to the Arduino Uno board. The second part of the project includes the mobile application that allows storing the data from the sensors to a database and the web pages needed to interpret and represent it.

A flux diagram that helps visualizing the data transfer process can be found in Figure 1. As seen, each operation shows which programming languages have been used to develop it and whether the data is transmitted to the proceeding step via Bluetooth or Internet connection. Each procedure will be further explained in each regarding section of the Methodology chapter.

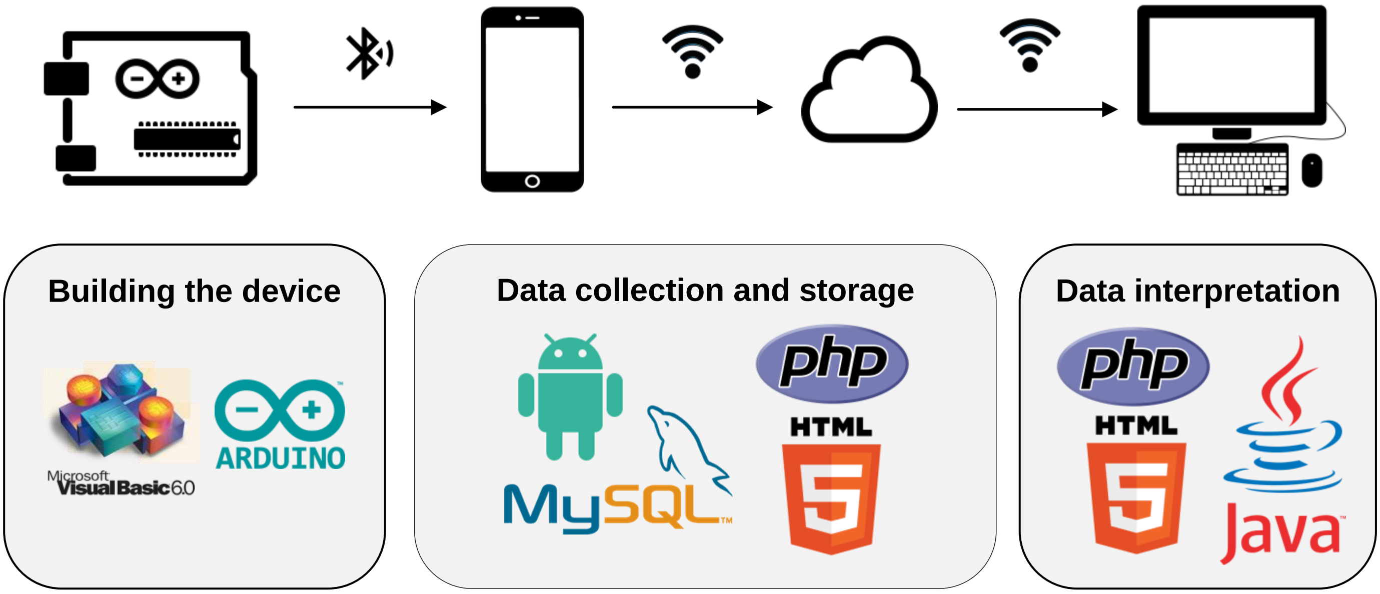


Figure 1. Diagram of the project

## 2.1 Hardware

The first task for the student carrying out the academic activity is the habituation with the hardware. Understanding what an Arduino board is, basic notions on how the sensors collect data and familiarizing with the cables and the pins. In order to build the biomedical device, an Arduino Uno board with a Bluetooth module, several cables and four sensors are at the user's disposal. These are a temperature sensor, a heart activity sensor, an accelerometer and a sound sensor. Furthermore, the first interface, which creates the code needed to generate data from the sensors and the Arduino, allows including as many new sensors as desired to the final assembled device and the whole data analysing system. Combining different connected sensors to the microcontroller board, then, the final biomedical device will be built. Depending on the theoretical background introduced in the session and depending on the illness that the student decides to diagnose, prevent, monitor or treat, the choice or addition of sensors will be made.

The four sensors mentioned are the following: An ADXL345 accelerometer[5] (figure 2), an analog heart rate monitor sensor[6] (figure 3) an analog sound sensor[7] (figure 4) and an LM35 linear temperature sensor[8] (figure 5). The microcontroller board used is the Arduino Uno and the Bluetooth module is the HC-05.

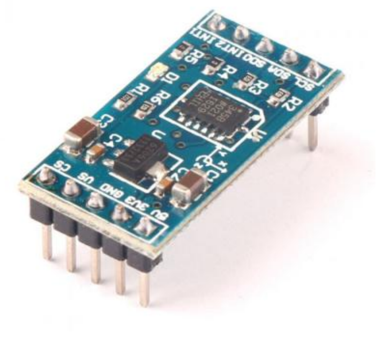


Figure 2. Accelerometer

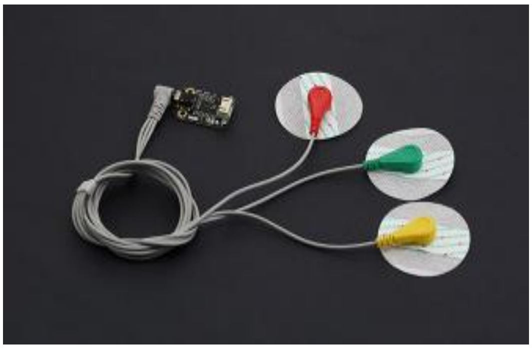


Figure 3. Heart activity sensor

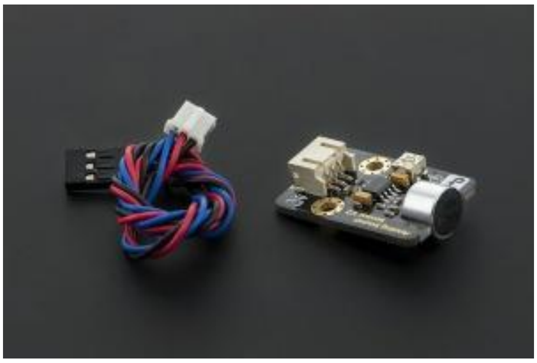


Figure 4. Sound sensor

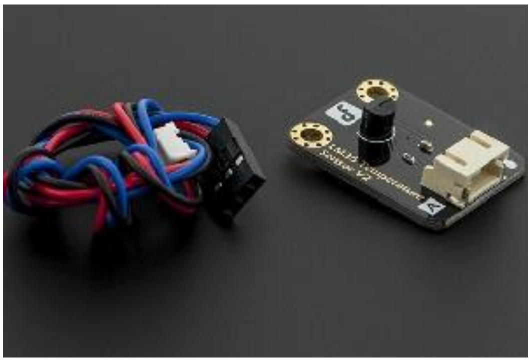


Figure 5. Temperature sensor

## 2.2 Building the device

Once the student is familiarized with the hardware, the step to follow is the building process of the biomedical device. Having received a theoretical session, described in the teaching unit, about how to do some research to decide which disease to work with, the user will proceed to establish the connections between the sensors chosen and the Arduino board.

The interface designed to help the student assembly the biomedical device and prepare it for the data collection has been developed using Visual Basic 6.0 [9], which is a version of the BASIC programming language from Microsoft specialized for developing Windows applications. This first platform can be described using two different approaches, which are what the student will see or will interact with and what the program will perform meanwhile and will go unnoticed by the student. Following this distinction and focussing on the first approach, how the interface works and the possibilities that it offers can be defined as following. When executing the program, a first widow is displayed (figure 6). Initially, it will only appear the image of the Arduino Uno board at the center and a drop-down list of the sensors available on the upper-left side, as indicated with the text "*Choose sensor to use*". This list includes already the option of uploading a new sensor. When chosen from the list, the image of the sensor will pop just above the button that allows starting its configuration. The user is allowed to choose the same sensor as many times as desired. This way, if the medical device designed by the student works using two or more sensors of the same kind at the same time, this program can support its creation as well. Once a sensor is selected and used, this first window is displayed again in order to facilitate the choosing of an added one. Thus, it works as an intuitive map that exposes which sensors have been used and combined.

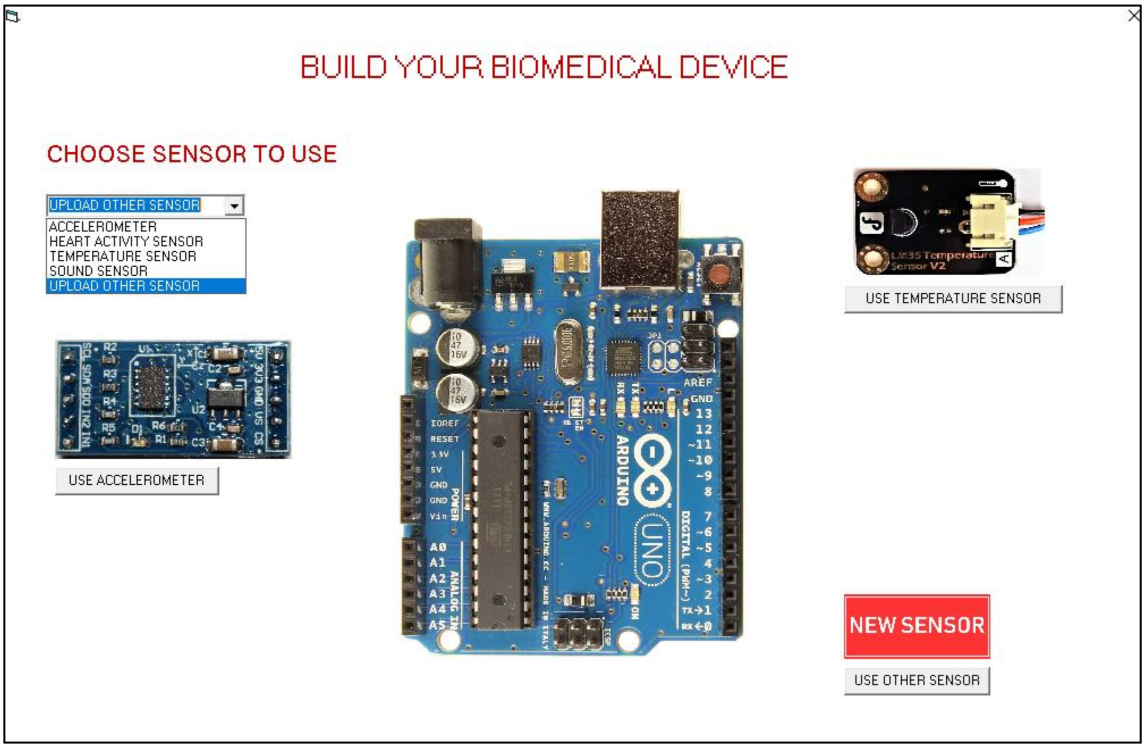
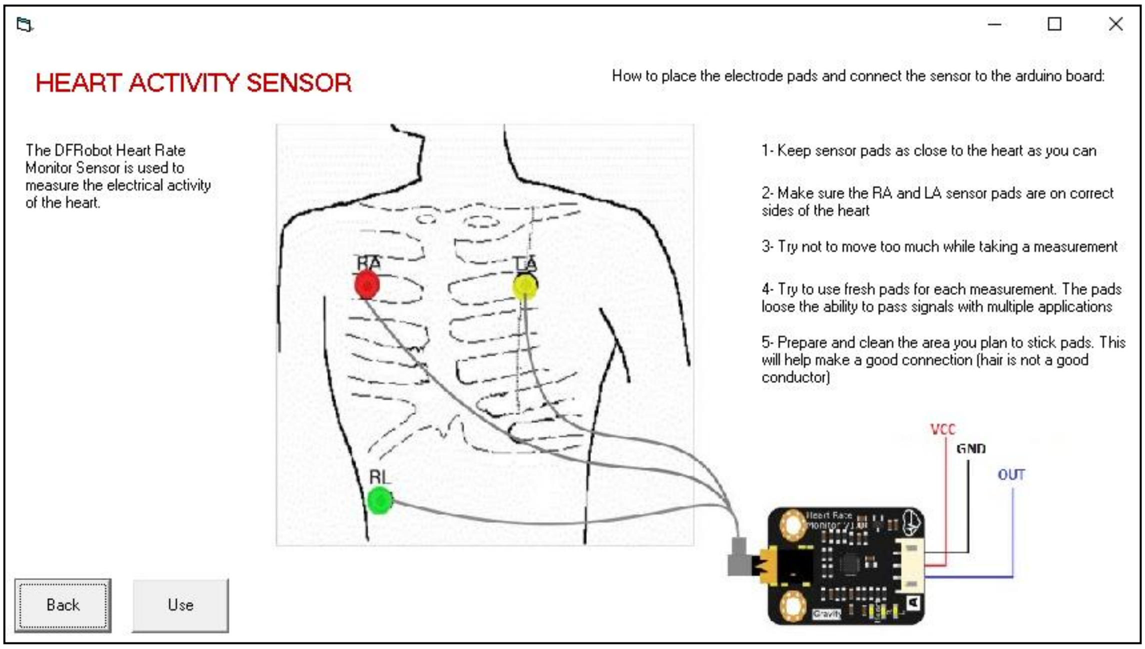


Figure 6. First window of the Visual Basic interface

Once a sensor has been selected form the drop-down list, not just its image appears but a button underneath it as well. When clicked, this button closes the main window and opens a new one the purpose of which is to provide information about the sensor picked. If the sensor chosen is one of the four available, as seen in Figure 7, the content in the window explains what type of data the sensor registers, how it has to be connected and the way it has to be placed, if necessary.



HEART ACTIVITY SENSOR

The DFRobot Heart Rate Monitor Sensor is used to measure the electrical activity of the heart.

How to place the electrode pads and connect the sensor to the arduino board:

1- Keep sensor pads as close to the heart as you can

2- Make sure the RA and LA sensor pads are on correct sides of the heart

3- Try not to move too much while taking a measurement

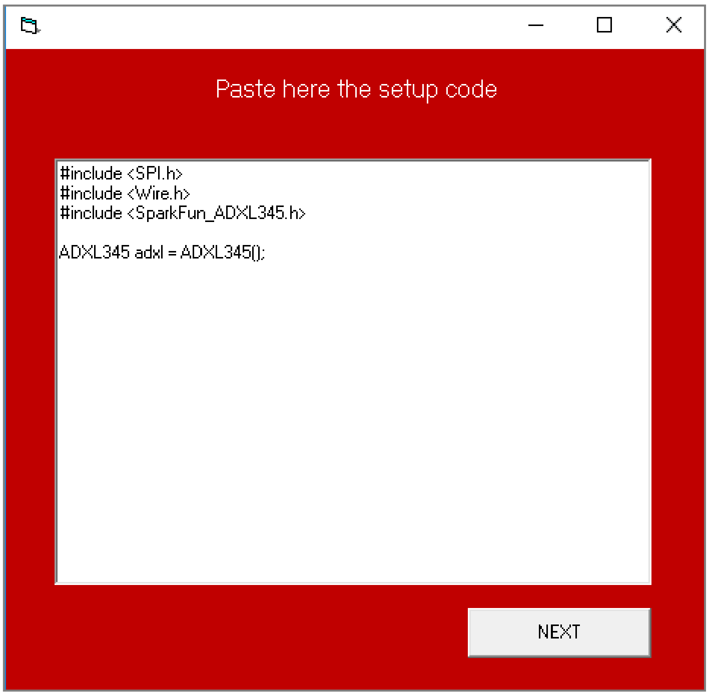
4- Try to use fresh pads for each measurement. The pads loose the ability to pass signals with multiple applications

5- Prepare and clean the area you plan to stick pads. This will help make a good connection (hair is not a good conductor)

Figure 7. Information window of the heart activity sensor

On the other hand, if the student decides to incorporate a new sensor to the project and, therefore, clicks on the “Use other sensor” button, the window that pops up is not intended to provide any information but to ask for it. Pasting the Arduino code needed for the new sensor to function in the two available blank text boxes is the only step that the user has to follow, along with clicking the “Next” button located underneath both of them. Two text boxes are needed as one only asks for the setup code, in case it is needed (Figure 8), and the second one asks for the code that will be included in the data recollecting loop function. Figure 8 shows the example of introducing a setup Arduino code of a new sensor, in this case being another kind of accelerometer that requires the use of several libraries.

These entire set of windows that provide specific information about each new sensor added to assembly the final biomedical device can be visited unlimited times. Thus, the user is allowed to recapitulate how the connections have to be established.



```
#include <SPI.h>
#include <Wire.h>
#include <SparkFun_ADXL345.h>

ADXL345 adxl = ADXL345();
```

Figure 8. Setup code of the new sensor window

The process of guidance for each sensor is brought to an end at the window where the young learner will connect virtually and, then, physically each corresponding pin (Figure 9). The setup of the window displays an image of an Arduino Uno board and the sensor selected to be used. The student, then, has to draw the cables that connect the pins of the sensor with the ground, voltage source and analog pin of the microcontroller board. This window is not only meant to be directive but interactive, as the user here has to decide where to connect the outputs, draw the lines and actually establish the connections between the gadgets. The student will understand that performing the connections on the screen at the same time than with the tangible materials is the way to tell the Arduino board what it will have to understand from the data received through the different channels.

Having established all the needed connections, the user has the option to add another sensor by clicking the button "Use another sensor". Once an analog pin from the Arduino board is chosen, it remains blocked. This way, the error of connecting two different sensors at the same place is avoided. The sensors selected for the project will appear listed with their images on the right side of the window also for the student to recall what has been already used.

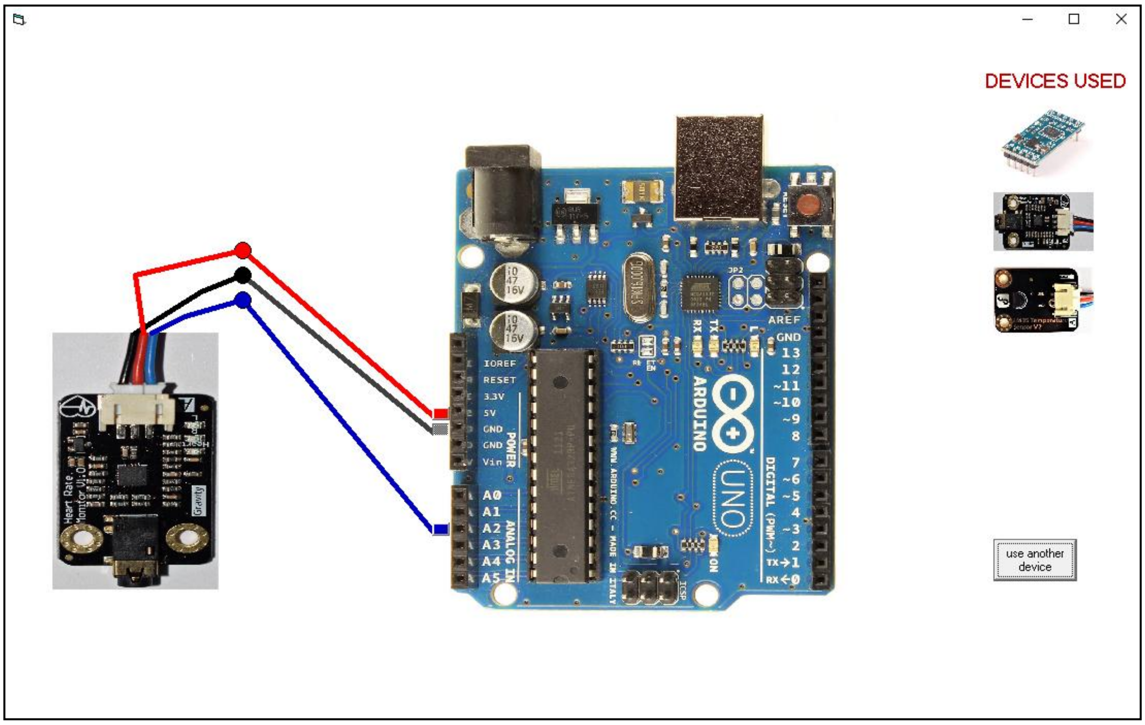


Figure 9. Connections window

The exit cross of this window ends the processes of this program. When the student has already connected all of the sensors that have to be used for their device and has drawn all the cables on the interactive window, the Arduino board has to be connected to the computer. Having closed the Visual Basic program the young learner will have to upload the Arduino code to the microcontroller board. The Arduino IDE [10] will be opened and the only task that the student will have to carry through is to decide the frequency of sampling and upload the code. The student will be able to examine the code automatically created from their interaction with the former platform and will have to type in this frequency where indicated by comments. Uploading the Arduino code is the last step that has to be executed by the student in order to have the device ready to start gathering data.

Focussing on the second approach, what the program carries out while the student is interacting with its windows can be described as following. All of the windows coded using Visual Basic have the objective to write the piece of Arduino code needed for the sensors to work and extra information that will be used when storing the data in the database. The first interactive window that the user works with registers every sensor that the student decides to use. A document form the notepad, the extention of which is .txt, is created and every time a sensor is selected its reference is automatically written on this document. The usefulness of this file will be further discussed when explaining how the webpage where the results are studied has been created.

The connections established by the student are also recorded in a different file. The information that is taken into account is which analog pin has been chosen and, in the case of uploading a new sensor, whether the data needs any kind of processing or not. All the Arduino code that calls these sensors' functions is written and uploaded to the Arduino IDE.

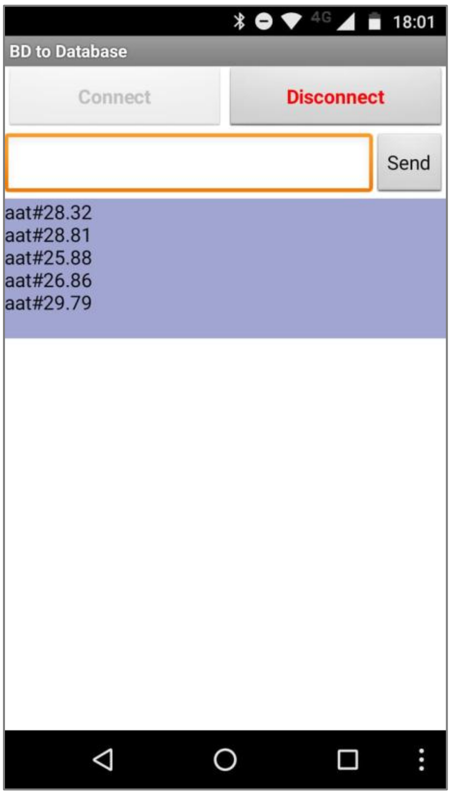
Thus, the resulting information provided by the Visual Basic interface is collected in these text files and is uploaded to the Arduino IDE as if they were libraries. The final Arduino code ready to be uploaded to the microcontroller board, ultimately, consists of the combination of the orders received from the first interface, the sampling frequency added by the user as explained previously and the lines of code that activate the Bluetooth module.

## 2.3 Data collection

Uploading the Arduino file starts the process of collecting data from all the sensors connected to the board and sending it through the Bluetooth module. This data has to be processed in order to be understood and classified due to its format. The Visual Basic interface is designed to mark the information received from the Arduino Uno board depending on the sensor where it comes from. What is sent via Bluetooth is a number, which concords with what has been sensed, along with a three-letter long marker and a hash symbol (#). The final form of a temperature, sound, heart rate or any other information rececived, then, is *xxx#num* where *xxx* is the marker specific to determine what kind of information the *num* is providing and the # allows it to be read in further processes.

The third program that has to be used and that takes part in the data collection is an Android application. There are multiple tutorials on the internet that explain how to build an app the function of which is only to receive data via Bluetooth and send it to a webpage. The one used in this project has been developed using MIT App Inventor

[11], which is a visual and blocks language for building Android applications. Its intuitiveness allows non-experienced people to build complex and powerful programs or projects up. Specifically, the application used in this project has only the option of being connected or disconnected and allows the content received to be displayed on the same screen (Figure 10).



aat#28.32
aat#28.81
aat#25.88
aat#26.86
aat#29.79

Figure 10. Android app

The only purpose of this program, then, is to receive the text string from the microcontroller board and send it to the next platform, where the data sorting and storing will take place. Figure 10 shows the example of this application receiving the first information strings from a biomedical device that contains only a temperature sensor. This data displayed, as said, is automatically sent to the data storing platform.

## 2.4 Data storage

The data storing process begins with the webpage receiving the data from the mobile application. The web pages of this project have been coded using PHP [12], HTML [13] and JavaScript [14]. Both PHP and HTML are languages wide known for web development. They are usually combined as the first one is used to create dynamic web content while the second one allows this content management, such as interpreting and composing text, images, and other visual material. JavaScript is another language of the World Wide Web that also permits creating interactive pages. In this part of the project, HTML operates with the database communication and JavaScript is used to create the dynamic plots. In order to being able to post the websites created onto the Internet, an account at Hostinger, which is an Internet hosting service, and a website domain was created. Moreover, a MySQL database is used to store all the information that will be provided by the biomedical device and the instructions given by the user on how to manage the results.

The first webpage needed to deposit the data into the database has been coded using only HTML as it does not need to be seen by the user or to be interactive. Therefore, it will remain unnoticed by the student. The text string *xxx#num* is sent from the android application and is read in this webpage. The hash symbol allows differentiating the left part of the string, which is the marker that indicates which sensor is providing such information, and the right part, which is the actual value sensed. Once the string has been interpreted, the value sensed is inserted to a table in the database. In order to do so, the connection with the MySQL database has to be done in the first place along with the table creation. This table will have as many columns as sensors connected to the Arduino Uno board and the number of rows will depend on how many times the sensors have displayed a value. Furthermore, time and date columns are added. They record in which moment each value is inserted to the table and, this way, the results interpretation can take into account information from different times or days and the conclusions extracted from the biomedical device by the student developing a project can be richer.

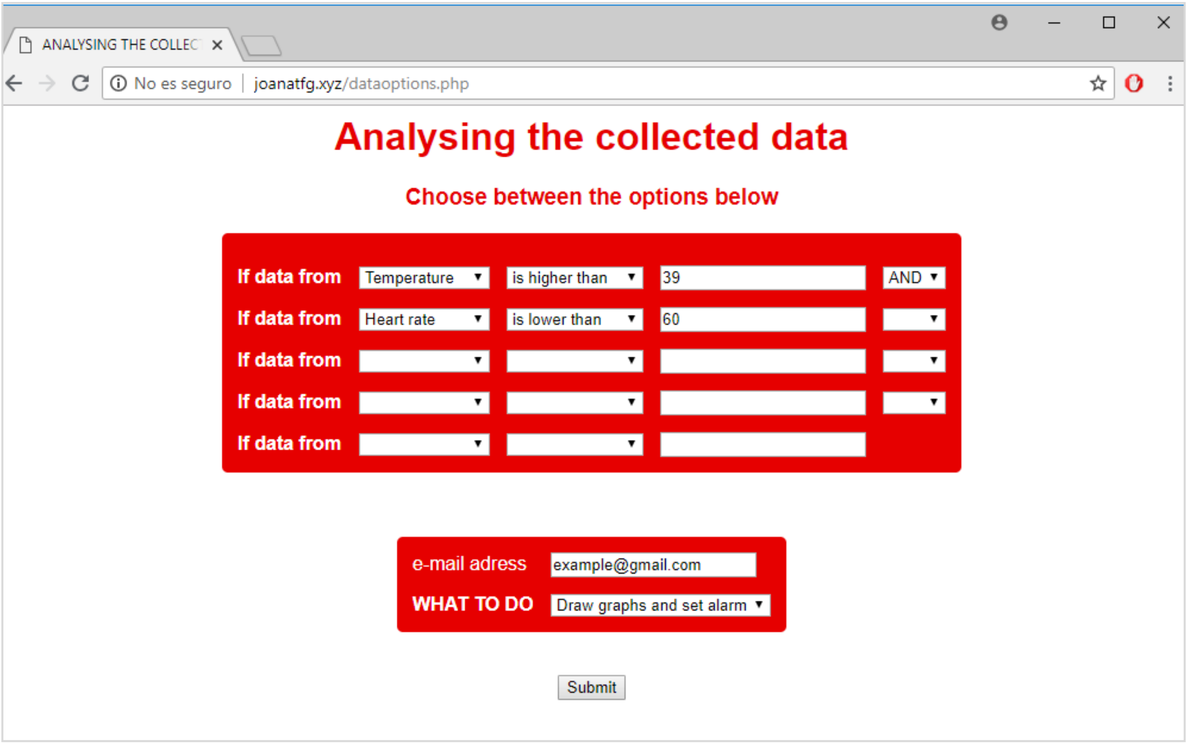
Simultaneously, another table at the MySQL database is created. This will list the sensors connected to the microcontroller board and will be useful when interpreting the data recorded.

The content from MySQL or the coding of any webpage developed in this thesis is not available for the student to play with. They are inaccessible as they perform their functions in an automatic way and are meant to facilitate the data study providing a public site where to see the results.

## 2.5 Data interpretation

The main objective for the young learner is to create a gadget able to prevent, treat or diagnose a disease. Thus, selecting what information is relevant from all the data that will be recorded is what will define the powerfulness of the device.

To facilitate the student the options for interpreting the data, a webpage that provides different operations and establishes relations between the results obtained from different sensors has been developed. This site's objective is to ask the student how they want to treat the data. A table has been created and it contains four interactive columns per row, as seen in Figure 11. Every row of the table aims to be a sentence that contains an order in the form of a condition. Below this first table there is another interactive block where the user is asked what to do in case the condition is met. Thus, the order will be defined half in the first table, where the student will define the condition of what must happen in order to act; and half in the second block, where the action will be chosen. Initially, all the fields are found blank. Figure 11 shows an example that will be developed further.



Analysing the collected data

Choose between the options below

| | | | | |
|---|---|---|---|---|
| If data from | Temperature ▾ | is higher than ▾ | 39 | AND ▾ |
| If data from | Heart rate ▾ | is lower than ▾ | 60 | ▾ |
| If data from | ▾ | ▾ | | ▾ |
| If data from | ▾ | ▾ | | ▾ |
| If data from | ▾ | ▾ | | |

e-mail adress example@gmail.com

WHAT TO DO Draw graphs and set alarm ▾

Figure 11. Data analysing website

In the condition-defining table, the first drop-down list allows the student choose to which sensor's data the command will go to, as it has a list of all the sensors that are connected to the Arduino Uno board. Recalling the data storing process, the table from the MySQL database that lists the sensors is used here to create the options from the drop-down list. The webpage is made to establish connection with the database and use the texts strings from the table to write the multiple options. The second drop-down list asks the user what mathematical relation is contained in the order that wants to be given. The options are *"higher than"*, *"lower than"*, *"equal to"* and *"different from"*. The third interactive column is a text box where the student will be asked to introduce a value. Thus, the condition will be established. One clear example of a possible result from interacting with these three columns could be: *If temperature is higher than 39ºC*. As explained before, the action to do in case this condition is met will be chosen in the last part of this website. Finally, the last column contains, in another drop-down list, the possibility to add more conditions to the final command. The list includes an *"and"* and an *"or"*. If this column is used, the program expects the next row to be filled following the same dynamics. Thus, an example of writing two conditions to be fulfilled could be: *If temperature is higher than 39ºC and if heart rate is lower than 60 beats per minute*. This particular example can be seen in Figure 11.

Once the condition is defined, the action has to be determined. The second block contains two interactive items. The first one is a text box where the user is asked to type in their e-mail address. The second one is another drop-down list with the actions that can be carried out. The options are *"set an alarm"*, *"draw the graph"* and *"set an alarm and draw the graph"*. The alarm, in this case, is an automatic e-mail sent to the direction indicated. The drawing of the plots is performed, in case the condition is met and the action chosen includes drawing graphs, in a different webpage.

The *"submit"* button found below the second information block has to be clicked after having filled all the desired fields. When clicked, the webpage sends the command defined by the user to a secondary webpage that will upload it to the database and will start scanning and analysing the data waiting for the condition to be met to start the action. The data from the database, received from the mobile application, is scanned every brief amount of time to ensure that the student will be aware any time if the condition defined in the table is met.

Thus, this data analysing tool defines an intuitive user-friendly way of deciding how to use the information provided from the sensors. It allows concluding the development of the biomedical device, as the e-mail sent or a webpage that will automatically start plotting the values desired at real time are the last part needed for the device to be fully serviceable.

## 2.6 Teaching unit

A teaching unit is a detailed description of the activities held to guide the teacher through a lesson. It describes the time and material needed to complete the activities, the dynamics of each part of the session and how to evaluate the work done. A lesson plan must reflect the interests of the students and must take into account their needs. Thus, in order to be inclusive, it has to take into consideration the different learning rates that can coexist.

The project includes a development of a teaching unit (see SI). This lesson plan completes the thesis as it allows the program to reach the educational system and facilitates and promotes the inclusion of Biomedical Engineering to the High School academic curriculum.

# 3. RESULTS

The resulting gadget from the first interface is the biomedical device. As commented in the posterior sections, the possibilities are almost unlimited as the platform allows including as many new sensors as desired. The tangible result is made up from the Arduino Uno board, the Bluetooth module, the sensors connected, the cables needed to establish those connections and the battery of the board that permits the device to be portable.

From the android application, the results are the display of the biometric sensed data on the screen. As this interface is not meant to be interactive, there is no need for it to provide more information.

Finally, the obtained results from the web pages where the data is analyzed depend on the options chosen by the user in the data interpretation table. The conditions applied will determine whether the data will be plotted and when the user will be alerted. Thus, the final results of the program are available through any device with internet connection.

The data collected and stored throughout the different platforms is analysed in real time. The sensors gathering information is the first stage of the process and the plotting is the final. During this data transferring levels a delay of time is accumulated. It can be interesting to study this lapse of time found between the body generates the biometric signal and the displaying of it at the web page. The processes where time difference is added are the Bluetooth transfer of the sensed information, the reception of it by the mobile application, the time needed for the app to send it to the webpage, the storing in the database operation, the collection from the database done by the webpage and, lastly, its plotting. Examined using a chronometer and inspecting the time of storing recorded by the MySQL database, this interval of time is not higher than 2 or 3 seconds approximately.

# 4. DISCUSSION

The problematic addressed in this thesis was the lack of Biomedical Engineering lessons in the High School curriculum. Moreover, the programs developed to find a solution to this situation were designed to be part of the biotechnological DIY community. Thus, the main progress provided by this project is the creation of the programs able to guide the user to build a biomedical device and its sharing.

Compared to other tools developed so far, this project offers more versatility. It allows the integration of as many sensors as desired and the posterior on-line analysis of their data collection. The data analysing web page permits the user to monitor any biometric signal desired almost anytime and anywhere.

The hardware used can be easily acquired and their prices are low, which makes the tool developed in this thesis excellent for bringing science and technology closer to every interested individual. In order for it to be fully accessible, it could be uploaded to a web-based hosting service such as GitHub. Thus, the powerfulness of the thesis would exponentially increase as it could be improved and uploaded again by anybody capable and willing to add features or new functions to it.

The sensors used have not been validated as the biomedical devices resulting from the project are not meant to be medical or healthcare services but a tool for researching, further developing, learning and experimenting.

The development of the lesson plan completes the project as it fully develops one of its multiple applications. Introducing this innovative project to the educational system could be positive for the students. Working in a project-based dynamic might increment their interest for the subject and might benefit, ultimately, all the Biomedical Engineering community.

As future work, adding more options to the data analysing interfaces could be considered. Having more alternatives to interpret de information gathered could lead the user to achieve more useful conclusions.

# BIBLIOGRAPHY

[1] Hackster.io, 2016-12-14, Heart Rate Monitoring System
https://www.hackster.io/hrms/heart-rate-monitoring-system-8da2fa

[2] Hackster.io, 2016-08-14, Arduino Lie Detector
https://www.hackster.io/BuildItDR/arduino-lie-detector-a0b914

[3] Majed Marji, *Learn How to Program with Scratch*: *A Visual Introduction to Programming with Games, Art, Science, and Math*, No Starch Press, 2009

[4] World Health Orgainsation, Medical Device – Full Definition
http://www.who.int/medical_devices/full_deffinition/en/

[5] *Digital Accelerometer ADXL345 Data Sheet*, Analog Devices, Inc., 2009

[6] *Single-Lead, Heart Rate Monitor Front End Data Sheet*, Analog Devices, Inc., 2012

[7] *Electret Condenser Microphone*, DFRobot, 2017

[8] *LM35 Precision Centigrade Temperature Sensors*, National Semiconductor Corporation, 2000

[9] Noel Jerke, *Visual Basic 6: The Complete Reference*, Paperback, 1999

[10] Massimo Banzi and Michael Shiloh, *Getting Started with Arduino: The Open Source Electronics Prototyping Platform*, Maker Media, Inc., 2014

[11] David Wolber, Hal Abelson, Ellen Spertus and Liz Looney, *App Inventor 2: Create Your Own Android Apps*, O'reilly Media, Inc., 2011

[12] Robin Nixon, *Learning PHP, MySQL & JavaScript: With jQuery, CSS & HTML5*, O'reilly Media, Inc., 2018

[13] Jessica Niederst Robbins, *Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript, and Web Graphics*, O'reilly Media, Inc., 2012

[14] Edgar D'Andrea, *Java 2. Curso de Iniciación*, INFORBOOK'S S.L., 2006

# SUPPORTING INFORMATION