Smart Bracelet - Armisael

By Team FUREKA

Wu Dan

Yang Yibin

Xing Jiajia

Yang Fan

Raffael Candido Rabelo

Faculty Advisor: Dr. N. Sertac Artan

Full project technical report submitted in partial fulfillment of

the requirement of the Degree of Bachelors of Science

in the Department of Electrical and Computer Engineering

in the School of Engineering and Computing Sciences

of New York Institute of Technology

Fall 2015

Table of Contents

Introduction

This report, written by Team Fureka, explains how the project is designed and all the technical functionalities of components. The report includes the motivation for its development, components setup & tests, methods used, and criteria to evaluate its functionality, discussions, and conclusions based on the observations.

Our project is a three piece combination: the bracelet, the exoskeleton and the mobile application. The Smart Bracelet: Armisael is an innovative design that differs from all the other wearable devices in a very unique way. Our product is focused towards people that want to have a healthier life, including those who have heart conditions. The main difference of Armisael to the other wearable devices is that our system focuses on using the user's heart rate to guide his/hers exercise.

In this report, you will find sections that describe: components & sensor details, circuit schematic, flow chart and mobile application information. This is a full report to help the understanding of our project: how to build another prototype and the source code interpretation. Note that in the Discussion section we will give our comments in regards the enhancement of the project. We believe this is a great project that deserves further implementation.

Motivation

People who concern about their health are constantly increasing in population. Every year more and more people realize the importance of doing exercises or practicing some kind of sport. We can notice the rising number of people who join to a morning run and gyms are crowded than before.

Even though the wearable market is increasing towards smartwatches sales, there is still a lot of room for smart bracelets (wristband) wearables. Smart bracelets are projected to increase its unit's sale by 2 million in comparison to this year's projection (drop of 3 million units from 2014 to 2015) [4]. We are confident that our work will innovate the wristband's market and how people use smart bracelets. Armisael is a revolution in the wearable technology: the system's combination allows the user to choose their goals and based on those goals Armisael will help the user to adjust its physical condition; and it will give real time feedback to the user, allowing him/her to take a rest or to speed up depending on the case. This is a next step that was not found in any other smart bracelet system.

Our product is not focusing on counting calories, but it rather informs the person's performance by its heart rate condition. Therefore we also include and help people with heart rate problems by letting them know how much their heart is being affected by any activity. There were many new releases in 2015 on smart bracelet's market and most of new releases only one has similar functionality as our Armisael: the Jawbone UP3 [5]. However, our approach has one main advantage: provides real time feedback to customers.

Components Details

This section gives the full detail information of all the components used in this project. In order accurate record heart data and transfer that data to our cell phone application, we need the following components: microcontroller (Pro Trinket 3V – 12MHz), Bluetooth (CC2541), Heart Rate sensor (SON1205), Temperature & Humidity sensor (HDC1000), Battery (Lithium Ion Polymer Battery), LED light (Red) and a Smartphone (Android version 4.4). Below there is a full description of each component we are going to use.

1. Adafruit pro trinket:

The Adafruit pro trinket is a small size board, which is great to be used into a bracelet. The 3V version has a BAT+ pin when connected to the USB, provides a 3V output for the battery, which could help us when we want to recharge for the battery. The pro trinket can be coded using the Arduino IDE.

2. CC2541

CC2541 is a 4.0 Bluetooth transmission component, which will transmit data from both bracelet and the smart phone. In this way, the measurement data can be coded and passed to the smart phone. On the smart phone, the user can check out the data, accordingly to the measurement, and it also sends a command back to the bracelet. This Bluetooth module consumes roughly 0.06 mW (on active mode, which is when is transmitting the data). This module also is compatible with I2C communication or via USB.

3. HDC1000

   HDC1000 is a temperature & humidity sensor and it does not consume a lot of power (around 3.6 µW). In addition, its measurement accuracy is ±0.2 °C and ±3%. We will use I2C communication with the microcontroller in order to transmit its data to the microcontroller.

4. SON1205

   SON1205 is what we used to measure the heart beat rate. Since according to the data sheet of SON1205, this component could output two different kinds of heart beat wave forms, the normal heart beat wave and the square wave. The low pass filter is already assembled in this chip, so, we just need to connect it to the IO port and calculate the times of heart beat for every one second.

5. Lithium Ion Polymer Battery

This battery is very handy for the application of our project. It will be used on the bracelet part providing 3.7V and 150 mAh (same current output as the microcontroller, providing less risk in burning the equipment). When fully charged, this battery can provide up to 4.2 V which is pretty good in order to keep our components turned on. Note that the increase of the voltage does not affect negatively our components. The battery itself fits perfectly for the bracelet. Its tiny dimensions are just enough to not consume too much space and its weight is also irrelevant comparting to other components. Its physical dimensions are: 4.65g, size of 19.75mm x 26.02mm x 3.8mm / 0.77" x 1.02" x 0.14" and power lead length: 127.5mm / 5".

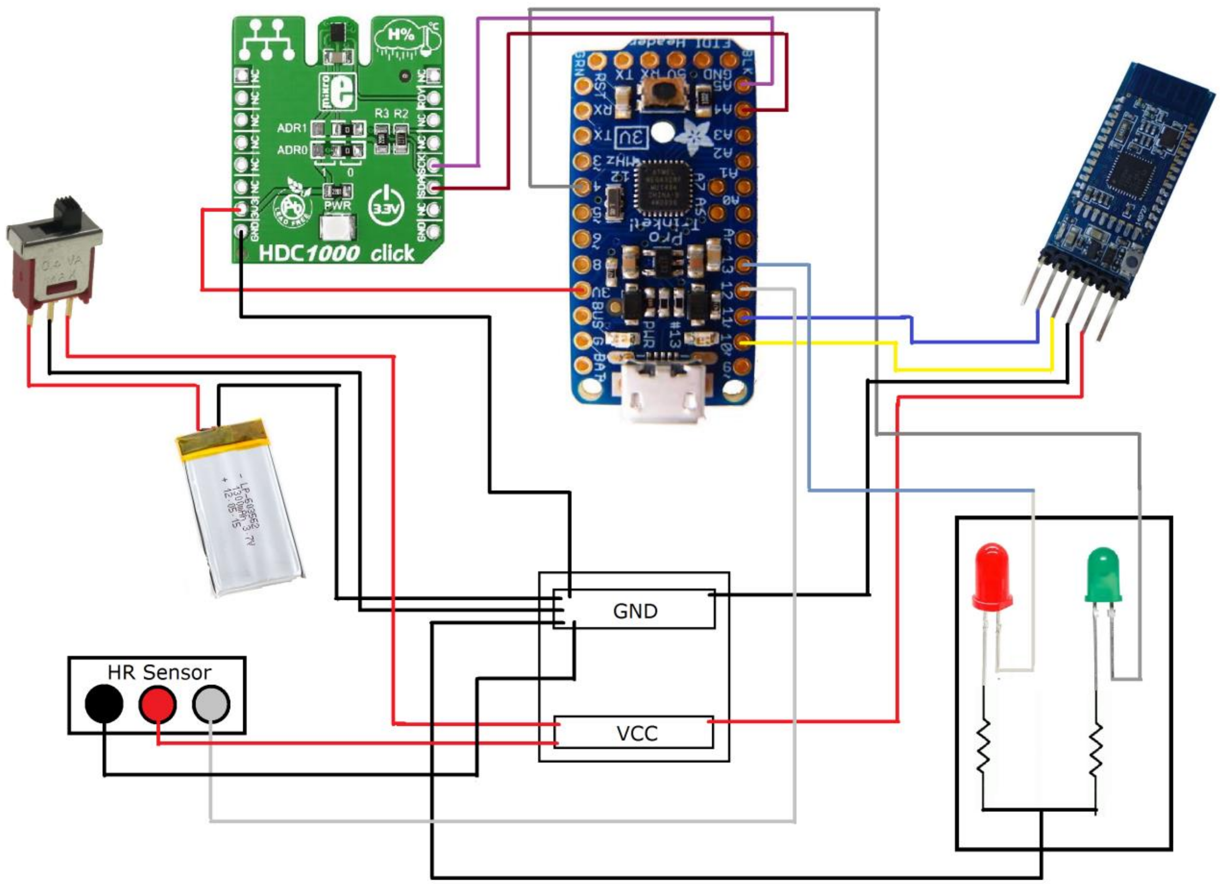
6. Metal Gear Servo Motor

   The servo motors we have 15 kg.cm / 208.3 oz.in (at 6V) and 13 kg.cm / 180 oz.in (at 4.8V) torque. This motor provides the necessary torque in order to successfully compress the spring.

Project's Breakdown

The Smart Bracelet

Figure 1 - Armisael Smart Bracelet Circuit Diagram



This figure illustrates well how the schematic looks like. All the components are on the same level besides the Heart Rate Sensor on the low right corner of the figure. The Heart Rate sensor is located under the BLE 4.0 module and the HDC1000, facing the skin of the user. Besides the HDC1000 all the other components are powered from the battery when the switch is pushed to the right. Since HDC1000 is the only component powered from the Pro Trinket, the microcontroller does not lose power and can operate very well.

The microcontroller's pins are divided in three: Digital GPIO, Analog pins and Communication pins. The communication pins are 10 and 11 (RX and TX respectively) and they connect to the BLE module. Note that the microcontroller 's pin 10 connects to the TX of the BLE and pin 11 to RX.

The digital GPIO pins of the microcontroller is connected to the Heart Rate signal output (12), the green LED (14) and the red LED (13). The analog pins are connected to the HDC1000 SDA (A4) and SCL (A5) pins.

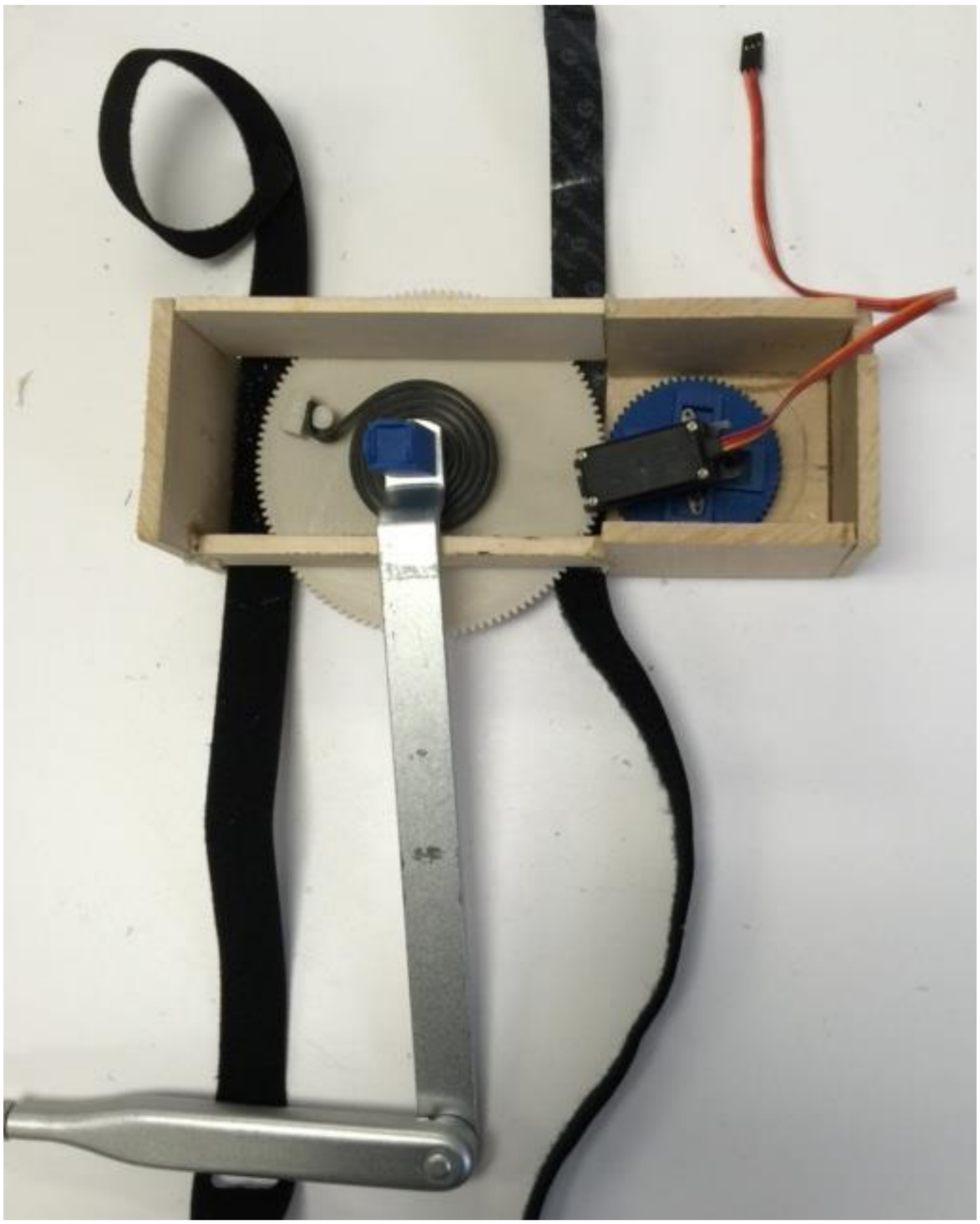
The battery is connected to a ground PCB and to the switch. The switch has two links on ground and one link to the battery's power. The PCB located to the right of the battery has a GND (ground) and power (Vcc) locations. The BLE, Heart Rate, and the Pro Trinket are connected to both GND and Vcc locations of the PCB. The HDC1000 is powered from the Trinket, 3.3V pin of the HDC to 3.3V of the Trinket and GND to GND.

The Exoskeleton

Figure 2 - The Exoskeleton

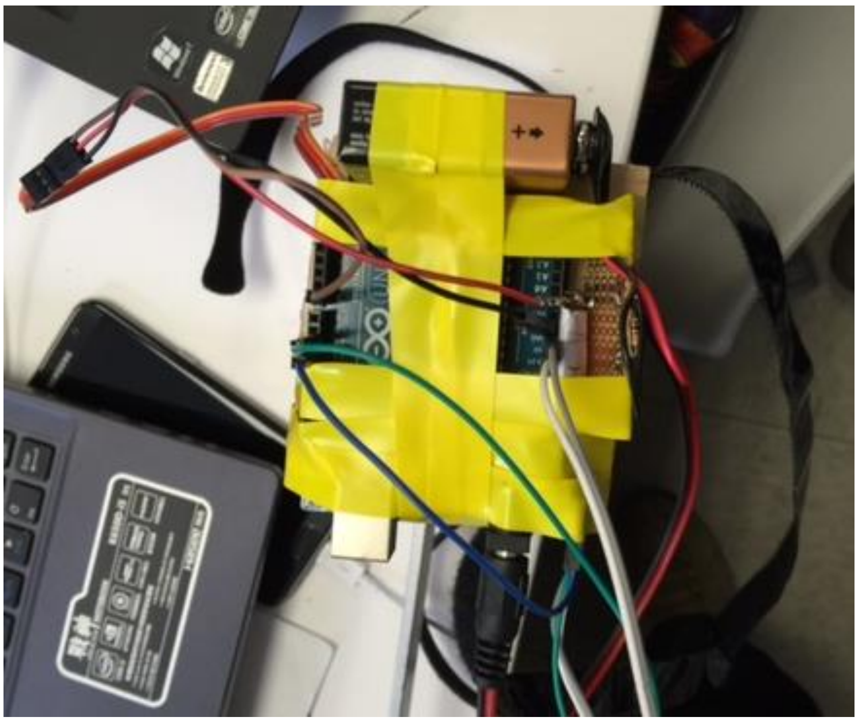
The Exoskeleton is designed to relieve or increase tension on the user's leg accordingly with the user's heart rate data. In the next set of figures we will analyze the components that are used to build the exoskeleton. The outside part of the exoskeleton is made of footbrakes, Velcro straps and wood. The wood gives a solid object and the Velcro straps are used to keep it tightened to the user's legs. On the foot we have a platform used to attach the footbrake, and the Velcro strap is also used to tightened the foot to the platform.

Figure 3 - Inside the Exoskeleton



The inside is composed of two 3D printed gears, one servo motor, one spiral spring and a few 3D printed components (on top of the spring and the footbrake). The 3D printed gears have two sizes: the smaller one is the one driven by the servo motor and its small size is to make our exoskeleton not too big; the larger gear is used to make it easier for the servo motor to be able to compress the spring when it rotates. This is the main reason why we made the second gear to big: previous servo motors were not able to compress the spring. The 3D printed components are small and larger pieces that fits through the footbrake, the spring and the gear. This way these components will not detach while on use. The gear used as mentioned before is 15 Kg.cm at regular voltage. The reason we kept it is because even with a larger gear, regular servos were not able to successfully rotate the spring's gear.

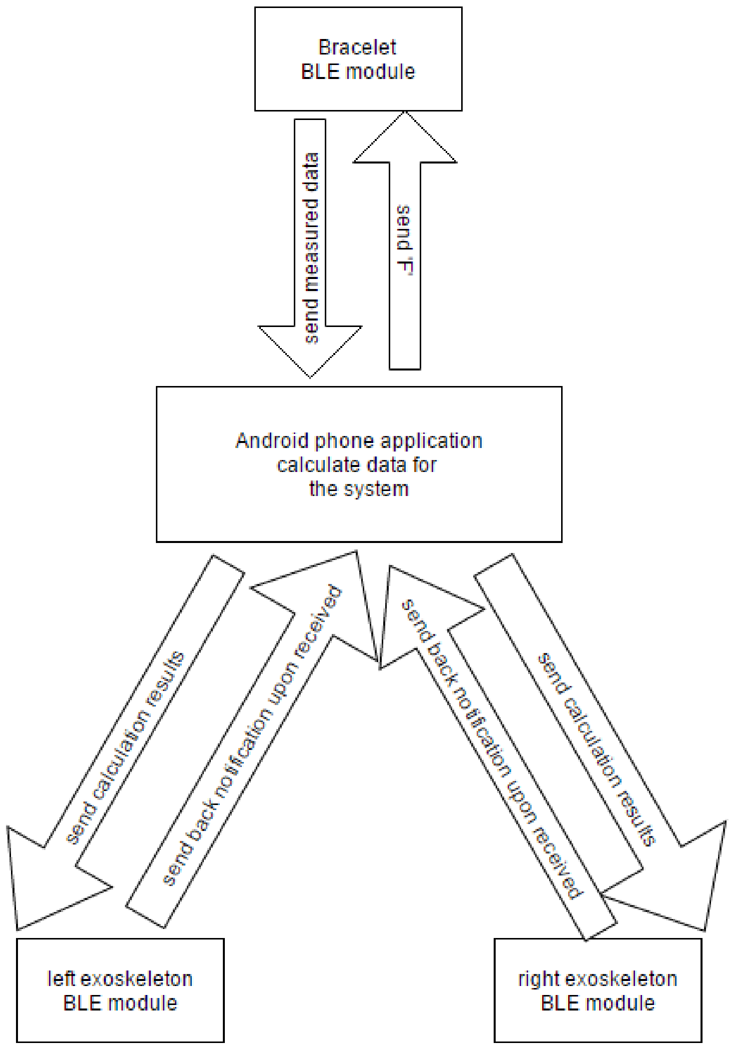
Figure 4 – Top of the Exoskeleton



The top of the exoskeleton we have the battery connected to a voltage divider and then to the Arduino Uno, in each leg. The voltage divider is used because we use the 9V battery while the Arduino needs 6V. The communication pins for the Uno is the same for the Pro Trinket previously used. We attach pins TX and RX of the BLE to pins 10 and 11 on the Arduino Uno respectively. The servo motor's signal is connected to pin 7. The connection is very simple and is only to process the data received from the application and to send back an “Ok” message.

The System’s Software Analysis

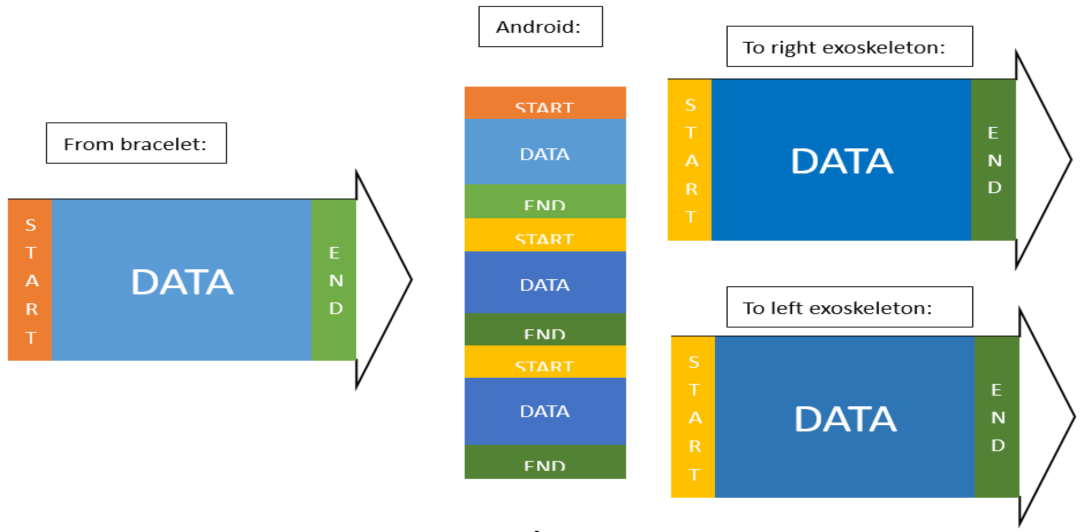
Figure 5 - System’s Detailed Flow Chart



So, in this part of article, a brief explanation of how the entire system worked would be demonstrated. According to the different functions and the place where the code uses, we can divide the entire system code into four categories, which can be the code for Android application, the code for bracelet, the code for the exoskeleton on the left leg and the code for the exoskeleton on the right leg. Since the function on the right leg and the function on the right leg is almost the same, in this article, the explanation of the code for right leg would not be included. The following flow chart shows the idea that how the entire Bluetooth network.

For the Android application, the entire Bluetooth communication system is depending on the cell phone. As the Bluetooth Low Energy system (shorten as BLE below, where specifically, we can call it HM10) is a very low level system, where a lot of bug might occur when we are trying to make communications between all the four systems, the Android cell phone application is set as the major server for the entire network. As all the message transferring is based upon the cell phone, all the other BLE components are easily getting crashed when we are using it, we tried a lot of methods in order to make the design out come true. First of all, he can define he cell phone as the server of the network, and the bracelet as the clients, as well as the exoskeletons. As in a network, clients communicate with the server in three different methods, including using frequency to distinguish, using time slots to distinguish and using coding system to distinguish. In the BLE network, since basic frequency of the Bluetooth would be at 2.4GHz, a frequency defined system would not be allowed in our project. Also, since the coding system is already added to the BLE, we would not add another coding system to that, making the communication much leggier. So, instead, in our project, we will use a time slot division system, where every client only occupies a very small time slots, and the server would sent different response to different clients depending to current connection. Also, another detail we noticed is that when we are trying to use the microcontroller to send data to the cell phone, or use cell phone to send data to the microcontroller, a very small latency would happen during the transferring, while that small amount of time is not very easy to measure. Since we are not going to send a large amount of data, and the time duration that the data would be sent would not be very long, we could use a flexible time division method. As we defined in the bracelet, bracelet would send a very short message to the cell phone every ten seconds, so that a large percentage of that ten seconds would totally not be in use at all in this system. When we are designing the project, we thought that why we do not make use of that ten seconds. In the Android application, a handler is defined so that the following three steps would be made in a loop. It will try to connect to the bracelet first. Once the cell phone receives the data from the bracelet and the command message that the transferring is already finished, the cell phone would switch to the next microcontroller that in charge of turning the servo motor. According to the heartrate that sent by the bracelet, the cell phone would calculate the exercising state that the user is at, and also calculate the degree that the servo motor should rotate to. In that way, a command message would be sent to the exoskeleton. As we set the exoskeleton as always sending back the data no matter what it received, we can infer that when the cell phone receive the data from the exoskeleton, the exoskeleton has already set the angle for the servo system. So, up receiving the data, the handler would be triggered to switch to the next BLE module. This loop would keep running all the time in the cell phone. The following chart shows how we designed the flexible time division system.

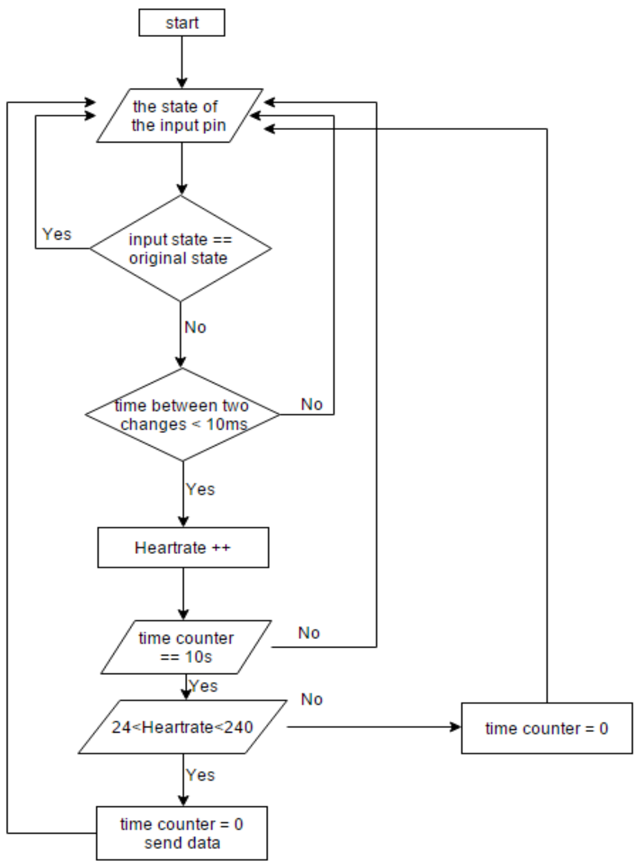
Figure 6 - System's Objective Flow Chart



For the code executed on the bracelet, we would divide it with three different functions, which is measuring heartrate, measuring temperature and humidity and communicating with BLE system. The heart rate is measured by the pulse on the skin. As is known, during the beat of the heart, the skin on the wrist, which covers the artery, would have the similar pulse. So, we use a LED light to shoot onto that part of skin. The light would be reflexed toward the light sensor next to the LED source. In that way, by measuring the change of the light through the light sensor, a similar data of heartrate would be calculated. According to our experiment, the light sensor would have a lot of noise due to the out environment, such as a sudden change of the light coming to the wrist, or a shake or move of the skin compare to the light sensor. In order to get rid of the noise, making the user's heartrate shown in the cell phone not strange in any mean, we added some methods to get rid of the noise. First of all, as the output of the heartrate sensor is a pulse signal, where pulse indicates the notable change in the light, we would calculate the rising edges as one pulse of the heart. An integer value is in the system and keeps adding when a pulse comes in. If two pulses are too close in the time, where specifically we define that short time as 10ms, that pulse would not be calculated in the system. Also, after getting rid of those pulses coming too close, we still have to remove some strange beats. In our project, we define "strange" as being less than 4 times per ten seconds, or being more than 40 times per ten seconds, which means, users could not get their heartrate below 24/m or over 240/m. In this way, when the sensor measurement gets out of that range, it would directly drop that period of data. As we measure the heartrate with the time unit of ten seconds, we made an array of integers, which containing six elements. When the time counter get to the 10 seconds, it will push the heartrate data into that array, while every element would be pushed front and the first, also the oldest data, would be removed from the array. So in this design, the heart rate is sent to the cell phone every ten seconds. The HDC1000 is the component is the sensor we use to measure the temperature and the humidity. It uses the I2C communication system, it can be listed as the simplest part in our entire project, which is also the very early part done in the project. As the library of the

HDC1000 is already designed for the Arduino boards on the Internet, after importing that, we can easily measure the data by calling the method defined by that library. The other part designed in our project is the BLE component. The data sent from Tx pin is already coded by ASCII codes. We can easily send characters through the BLE. And another detail we discovered is that if we use an integer value to define the data we receive, a number referring to the code table in the ASCII would eventually came out. As BLE component should be kept in awaken during the time BLE systems are communicating with each other, the cell phone is actually kept sending character "F" to the bracelet. And once the bracelet receiving data is not null, it would send the heartrate message and the temperature message to the cell phone.

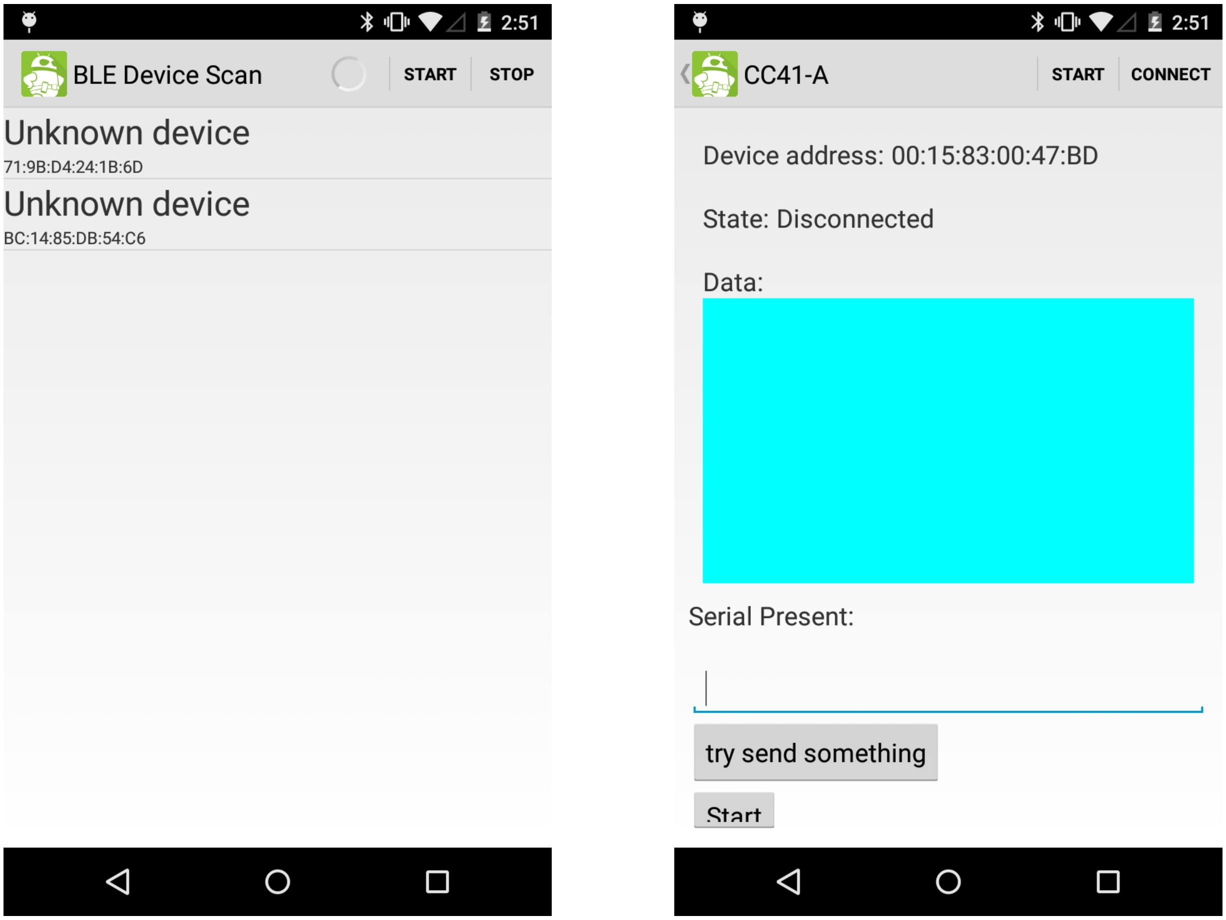
Figure 7 - Bracelet's Flow chart



The code executed on the exoskeleton is also very simple. During the time when no data is received, it would not set angle to the servo molto. Once receiving the data from the cell phone containing the angle where servo motor should be set to, it would send the data back toward the cell phone, and rotate to that degree.

The following two pictures illustrate how the mobile application looks like.

Figure 8 - Mobile Application



On the left we have the first page of the application which searches the BLE device: if the device does not have any name, Unknown Device will be displayed and the device's address. On the right we have the app when the user selects, or in our case the app selects automatically, the BLE device that will be used. It also has the following: start/connect to the BLE device chosen, displays device's address, the state of the connection, display screen and the area where the user could send data to the bracelet (or the exoskeleton).

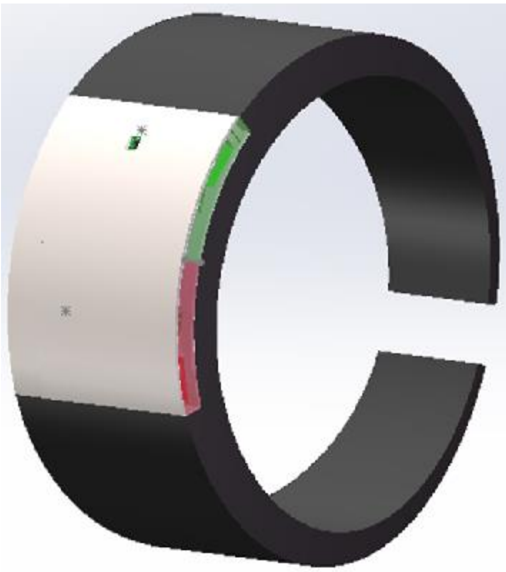
Discussions

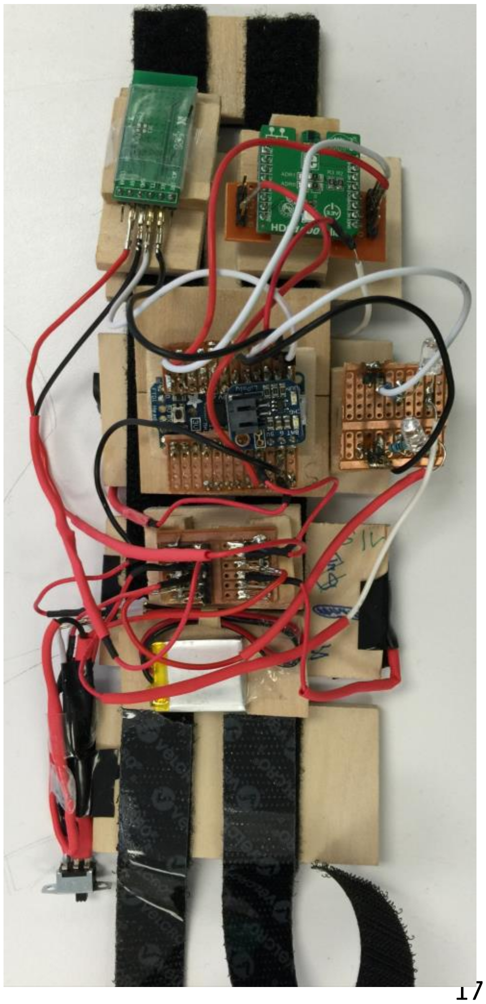
Accomplishments & Comments

Our first and most important achievement was to fully build and develop the smart bracelet prototype. We made a futuristic design when we first designed the virtual version of the bracelet. Our change was due to we condensed all the necessary components together, plus we designed using a good but not accessible material. We changed to the material we had access to and we left our components more spread. The reason is because since we did not have many components to spare, we could not focus on making as small as possible. In the end we did have a very comfortable and functional prototype

Figure 9 - Futuristic Bracelet vs. Realistic Design

.





Another huge accomplishment was the development of the exoskeleton. We were able to come up with a structure that was feasible to build and would not bother the user. As part of this achievement is the fact that we were able to find spiral springs and servo motors with high torque since most on the market has very little torque (4 lb.in or 4 kg.cm). To be able to find these components really helped to produce it instead of only propose.

One of the most important achievements we had was the development of the mobile application. This application was pretty much the hidden heart and mind of the project. Without the app we would not be able to produce as much as we did unless we changed components to previous versions (which would consume more energy). The app became the bridge between the data transmission of the system: bracelet to exoskeleton and because of that we decided to use its processing power to make the computation necessary for our needs. Letting the bracelet to focus only on recording and sending data and the exoskeleton to receive data and activate the servo motor.

Another important part of this project was to stablish a multiple way communication system between each piece. The CC2541 BLE module is based on the HM10 module which accepts AT+ commands. The reason we were unable to disconnect and connect with other Bluetooth modules using Pro Trinket microcontroller is because the HM10 modules were not able to update their firmware. Without updating the firmware, the HM10 did not understand more than a feel commands for its configuration, hence why we decided to use all the three modules as slaves and the Android phone as the master module in order to switch connections between each module. With the current firmware, the module was only able to understand the basic commands such as change password, name, get address, get/set power of transmission, inquire for other slave and connect to other HM modules, change to slave & master types, check version, reset and sleep.

Further Development

We proposed that we were going to add two sets of vibration motors, so the user could feel on their skin about their change in each type of exercise. Since we did not implement this feature, we are adding this feature as a next and very good improvement on top of what we accomplished.

Another good idea for improvement would be to find a way to update the firmware of all the BLE components. This would make a huge impact, because would change the data transfer to the bracelet and would allow the mobile app only to focus on profile designing and displaying data. Updating the BLE firmware will make the bracelet itself much more powerful than now. It's computing process will increase as well as the ability to connect to different BLE devices, but it will boost the real time feedback by the system. This

The profile app is also a very important improvement. If it can be improved to combine the BLE searching and the data display into a one app that also have profile setup, then the app will also be more powerful and will boost the system's speed as well. At the moment we use a range of 60-80% out of 200 (maximum) of heart beat, so creating a profile and letting the user to choose a different set of profile for different performances is a huge next step that we would love to see being implemented.

On the hardware side, if the new group can double the processing power on the bracelet (2 microcontrollers), than will be easier for the bracelet itself to communicate with the Android app and the exoskeleton at the same time without any interference on each signal. Another option would be research in another Bluetooth module that can connect to at least two other Bluetooth devices.

We know that any future groups that want to improve our project are more than welcome to use their own ideas. These are some set of ideas that we tried to implement ourselves, but we were forced to drop some of the ideas due to the building and developing process of a functional bracelet.

Conclusions

We are proud of working on the Smart Bracelet: Armisael project. This was a very challenging project in terms of putting in practice what we proposed. We realized the difficulty of working with prototypes and combining the entire algorithm and making the code work as we head in our minds.

Another very interesting part of the project that we went over on this report was the communication side of it. One of the most challenging aspects of our project was the data transmission: setting and understanding BLE modules and its work took more than enough time, also another factor was to send the right data through Bluetooth and through the microcontrollers. We changed our “data center” many times in order to fully be able to get the data sent for our project needs. Originally we were sending the data through the microcontroller to the exoskeleton, and then we had to change and use the mobile app to do this job.

We are able to guide our user by engaging in a very active exercise, hence why we use 60-80% out of a 200 maximum of heart beat. If the user’s heart rate goes beyond the 80%, then the system will decrease the tension on the spring and the user will get a message to relax for a while. And if the user’s heart rate goes below 60%, then the system will increase the tension on the spring, forcing the user to run harder, and the user will get a message to speed up.

Resources and Budget

The following table will give an amplified idea of resources we will intend to use and have to make this project low-cost, very affordable, with a good life spam, low power consumption and comfortable.

Table 1 - Budget Resources and Description

Component	***Description***	***Price***
Microcontroller Pro-Trinket 3V & 16 MHz 1x	The most important component of our bracelet. This microcontroller is the bridge between all the action happening on the bracelet: connects with the Bluetooth module; transfer data from sensor to the application; and reads the commands from the application to activate the LED lights	Adafruit.com - $9.95
Microcontroller Arduino Uno 5V 2x	Used to activate the exoskeleton's servo motor and send data back via BLE	Microcenter $24.95 – We already had 2
Temperature & Humidity sensor – HDC1000 1x	This is a very low-power, low-cost and very accurate digital sensor. It measures temperature and humidity.	Mouser Elecronics $19.00 -
Bluetooth Module – TI CC2541 3x	This is a 4.0 Bluetooth module. It operates at 2.4 GHz and is also very low-power consumption.	Ebay - $5.00 Sears - $40.00**
Android Cell Phone: Version 4.4 1x	This is the Android platform we will make our development with. Mainly because it supports Bluetooth 4.0 module.	Free – We managed to get a phone sample with one of our colleagues from Senior Design class.

Heart Rate Sensor – SON1205 1x	This is also very low-power, low-cost and light sensor.	aliexpress.com - $30 unit price Without Shipping fee
10mm Red & Green LED 1x Red (Pack of 2) 1x Green (Pack of 2)	High visibility, low-power consumption, small and very low-cost. Blue LED comes alone and Red comes in a pack of two LEDs.	Radioshack.com: Red - $2.49 Green - $2.49 Without Shipping fee
Lithium Ion Polymer Battery – 3.7 V 1x	Rechargeable battery and very small.	Adafruit.com: $5.95
Strips of Wood 2x	This will be used to the creation of the exoskeleton part.	NYIT Bookstore $2.00 each
Spiral Torsion Spring (Clock Spring) 1x	Spring to be used (on the skeleton)	California Tarps $11.99 without shipping or tax
Servo Motor 2x	Servos used to on the spring	Thinker Sphere.com – NY $13.99 each
Footbrake 2x	Used to be the suspension on the exoskeleton	Item.taobao.com $15.00 each
Total:	Here is the total cost of the project.	$158.09 (Containing only the ship from TI store).

References

[1] ICT Geelong, "ICT INDUSTRY - R&D AND COMMERCIALISATION STRATEGY" https://www.rdv.vic.gov.au

[2] Relation between heart rate and pulse transit time during paced respiration, Michael J. Drinnan1, John Allen and Alan Murray, Physiological Measurement, Volume 22, Number 3

[3] "*Aerobic Exercises Intensity and Target Heart Rate,*" HPMC Occupational Medic Services Aug 27, 2015 - http://www.hanford.gov/health/?page=112

[4] Gartner Says in 2015, 50 Percent of People Considering Buying a Smart Wristband Will Choose a Smartwatch Instead, STAMFORD, Conn., Nov 18, 2014. http://www.gartner.com/newsroom/id/2913318

[5] Gonzales, Zuriñe Dopacio, (Feb 10, 2015). "*The New Smart Wristbands for 2015*,"Wearables Technology - https://www.wearable-technologies.com/2015/02/the-news-smart-wristbands-for-2015/

[6] Danova, Tony, (Jan 25, 2014) "Smart Watches And Fitness Bands – Here's Why The Wrist Will Be The Next Frontier For Computers," Business Insider, http://www.businessinsider.com/smart-watch-and-fitness-band-sales-2014-1

[7] Resting Heart Rate Table, Topend Sports - http://www.topendsports.com/testing/heart-rate-resting-chart.htm

[8] Solarbotics Motor Specification VPM2 https://solarbotics.com/download.php?file=159

[9] Aerobic Exercise, Medicine Net (Nov 9, 2015), http://www.medicinenet.com/aerobic_exercise/article.htm

[10] Stevens, Eric C. "*Go Anaerobic: What It Is and Why to Do It,"* Breaking Muscle - http://breakingmuscle.com/strength-conditioning/go-anaerobic-what-it-is-and-why-to-do-it

Appendix A

The Relation between Aerobic Exercise Intensity and Target Heart Rate

When starting an exercise program, calculating a target heart rate zone can be very beneficial to ensure that you are exercising safely and effectively. Heart rates are referred to in “beats per minute” or bpm. The “target” heart rate zone is between 60-80% of your maximum predicted heart rate. In order to calculate Heart Rate Training Zones, you first need to calculate your Maximum Heart Rate. This is determined by your age:

220 – Age = Maximum Heart Rate

For Example: A 55-year-old would have the following calculation for Max Heart Rate:

220 – 55 years = 165 beats per minute, or bpm

To calculate their target heart rate zone (60-80%):

Max heart rate x target % = Target Heart Rate

165 x 60% (or .60) = 99 bpm

165 x 80% (or .80) = 132 bpm

This person’s target heart rate zone is 99 bpm to 132 bpm.

The Target Heart Rate Zone (60-80% of Maximum Heart Rate) is an area of moderate intensity activity that leads to improvements in your aerobic capacity and burns fat. This zone provides many benefits for all fitness levels, including those who want to lose weight, those who are training for an athletic event, or those who are looking to have more energy and get fit.

Exercising below this zone (50-60% of Maximum Heart Rate) is the Fat Burning Zone, because at this intensity, fat is metabolized for energy use at a higher rate. This intensity is often recommended to individuals who are extremely de-conditioned or new to exercising. While the name of this zone leads you believe that you will burn more fat at this zone, there is less cardiovascular (heart) benefit at this zone and less overall caloric burn. Again, if you are de-conditioned, have a heart or respiratory disease, or are new to exercising this would be an appropriate zone for you.

The High Intensity/Anaerobic Zone (80-100% of Max Heart Rate) is recommended for highly fit individuals, such as athletes. This zone places a high demand on the cardiovascular system and does not burn much fat. Individuals may use this zone as part of "interval training", where your heart rate reaches the High Intensity Zone for a short period (less than 60 seconds) and is allowed to recover to the Target Heart Rate Zone (65-85% of Max Heart Rate) for a period of time (60 seconds to 4 minutes).

Figure 1) Target Heart Rate to Guide Exercise Intensity

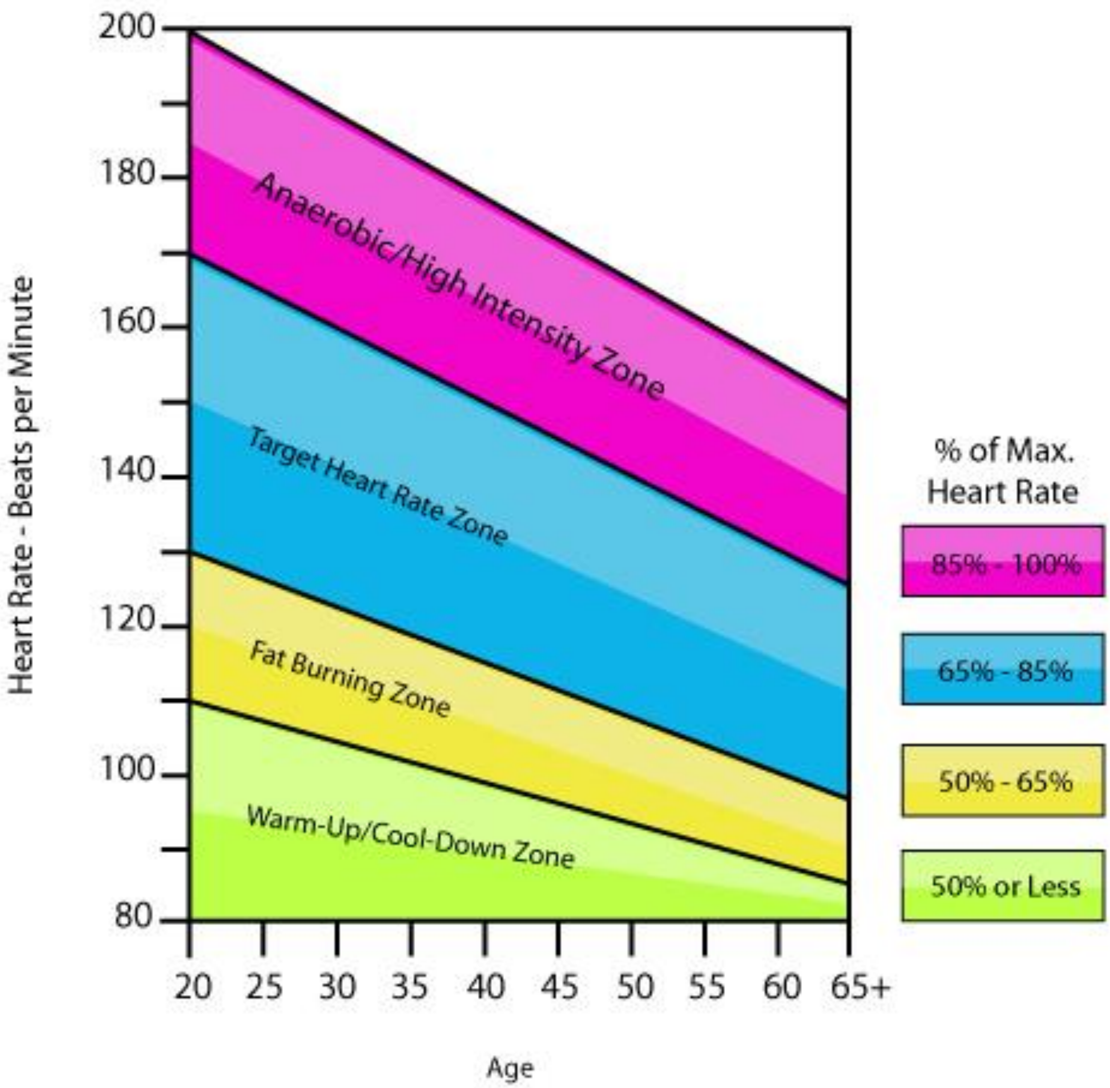


Figure 2) Heart Rate in BPM for Each Intensity Zone Chart

Use this chart to estimate your heart rate in bpm for each intensity zone

	50%	**55%**	**65%**	**75%**	**85%**	**95%**
Age	*(220-Age) x .5*	*(220-Age) x .55*	*(220-Age) x .65*	*(220-Age) x .75*	*(220-Age) x .85*	*(220-Age) x .95*
20	100	110	130	150	170	190
25	97.5	107.25	126.75	146.25	165.75	185.25
30	95	104.5	123.5	142.5	161.5	180.5
35	92.5	101.75	120.25	138.75	157.25	175.75
40	90	99	117	135	153	171
45	87.5	96.25	113.75	131.25	148.75	166.25
50	85	93.5	110.5	127.5	144.5	161.5
55	82.5	90.75	107.25	123.75	140.25	156.75
60	80	88	104	120	136	152
65	77.5	85.25	100.75	116.25	131.75	147.25

Appendix B

- Aerobic exercise is sometimes known as "cardio"- exercise that requires pumping of oxygenated blood by the heart to deliver oxygen to working muscles.
- Aerobic exercise stimulates the heart rate and breathing rate to increase in a way that can be sustained for the exercise session. In contrast, anaerobic ("without oxygen") exercise is activity that causes you to be quickly out of breath, like sprinting or lifting a heavy weight.
- Examples of aerobic exercises include cardio machines, spinning,running, swimming, walking, hiking, aerobics classes, dancing, cross country skiing, and kickboxing. There are many other types.
- Aerobic exercises can become anaerobic exercises if performed at a level of intensity that is too high.
- Aerobic exercise not only improves fitness; it also has known benefits for both physical and emotional health.
- Aerobic exercise can help prevent or reduce the chance of developing some cancers, diabetes, depression, cardiovascular disease, andosteoporosis.
- An aerobic exercise plan should be simple, practical, and realistic. Specific equipment (such as cardio machines) may be used but is not necessary for successful aerobic exercise.

1.1.1 What is aerobic exercise?

Imagine that you're exercising. You're working up a sweat, you're breathing hard, your heart is thumping, blood is coursing through your vessels to deliver oxygen to the muscles to keep you moving, and you sustain the activity for more than just a few minutes. That's aerobic exercise (also known as "cardio" in gym lingo), which is any activity that you can sustain for more than just a few minutes while your heart, lungs, and muscles work overtime. In this article, I'll discuss the mechanisms of aerobic exercise: oxygen transport and consumption, the role of the heart and the muscles, the proven benefits of aerobic exercise, how much you need to do to reap the benefits, and more.

1.1.1.1 The beginning

It all starts with breathing. The average healthy adult inhales and exhales about 7 to 8 liters of air per minute. Once you fill your lungs, the oxygen in the air (air contains approximately 20% oxygen) is filtered through small branches of tubes (called bronchioles) until it reaches the alveoli. The alveoli are microscopic sacs where oxygen diffuses (enters) into the blood. From there, it's a beeline direct to the heart.

1.1.1.2 Getting to the heart of it

The heart has four chambers that fill with blood and pump blood (twoatria and two ventricles) and some very active coronary arteries. Because of all this action, the heart needs a fresh supply of oxygen, and as you just learned, the lungs provide it. Once the heart uses what it needs, it pumps the blood, the oxygen, and other nutrients out through the large left ventricle and through the circulatory system to all the organs, muscles, and tissues that need it.

1.1.1.3 A whole lot of pumping going on

Your heart beats approximately 60-80 times per minute at rest, 100,000 times a day, more than 30 million times per year, and about 2.5 billion times in a 70-year lifetime! Every beat of your heart sends a volume of blood (called stroke volume -- more about that later), along with oxygen and many other life-sustaining nutrients, circulating through your body. The average healthy adult heart pumps about 5 liters of blood per minute.

1.1.1.4 Oxygen consumption and muscles

All that oxygen being pumped by the blood is important. You may be familiar with the term "oxygen consumption." In science, it's labeled VO_2, or volume of oxygen consumed. It's the amount of oxygen the muscles extract, or consume from the blood, and it's expressed as ml/kg/minute (milliliters per kilogram of body weight). Muscles are like engines that run on fuel (just like an automobile that runs on fuel); only our muscles use fat andcarbohydrates instead of gasoline. Oxygen is a key player because, once inside the muscle, it's used to burn fat and carbohydrate for fuel to keep our engines running. The more efficient our muscles are at consuming oxygen, the more fuel we can burn, the more fit we are, and the longer we can exercise.

Appendix C

<u>What Is Anaerobic Exercise?</u>

When I am coaching novices on getting their sprint on, I ask them to imagine a blood thirsty Rottweiler (not to pick on Rotties) trying to take a gash out of their hamstring - run like that**. Or as a sign I recently read said, "Run like you stole something." That is what it means to run anaerobically.** You can't do it for long. Why? Because training anaerobically <u>means training without oxygen</u>. Anaerobic exercise is defined as short duration, high intensity exercise lasting anywhere from merely seconds up to around two minutes. After two minutes, the body's aerobic system kicks in. Examples of anaerobic exercise are ones that use fast twitch muscle fibers such as jumping and sprinting. By using and developing those fibers we enhance that musculature.

<u>What Anaerobic Training Does</u>

The anaerobic effect happens in the body when we exert ourselves at 84% of our max heart rate and above. **When we train in this level of intensity for short bursts of energy, we create what is called EPOC, or** <u>excess post-exercise oxygen consumption</u>**.** In essence, EPOC is an after burn effect of calories burning at rest for up to 38 hours post exercise.[1] This type of training can be incorporated into both our cardiovascular exercise as well as our strength routines.