Social Fabrics

PROJECTO DE MESTRADO

Cátia Sofia Ferreira de Sousa

MESTRADO EM ENGENHARIA INFORMÁTICA

UNIVERSIDADE da MADEIRA

A Nossa Universidade

www.uma.pt

Novembro | 2010

a

c

SOCIAL FABRICS

By

Cátia Sofia Ferreira de Sousa

csfs.ferreira@gmail.com

n.º 2012504

Adviser: Valentina Nisi
valentina.nisi@gmail.com

Co-Adviser: Ian Oakley
ian@uma.pt

A thesis for Master's degree in Computer Science at Madeira University

November 2010

Abstract

Existing wearable computing research and indeed commercial products, have explored how to control phones and music players in pockets. They have typically relied on interaction via simple flexible button sensors. This thesis proposes, design and develops new ways of interacting which explore the potential of clothes, such as pulling or stretching. Its aim to present and demonstrate the value of embodied and intuitive inputs based on standard clothing elements such as zips, fasteners, beads, Velcro and magnets. Individual interactions for each are described and discussed before a final combination application, the MusicHoodie, which is developed to control an MP3 player. A simple usability test on this system reveals a range of interesting and promising results about which were the most acceptable and understandable inputs. This thesis closes with a discussion of the implications and contributions of the work it presents.

KEYWORDS: Wearable computers; Input technologies; Wearable technologies; Interactive clothing.

Acknowledgements

Many thanks to my co-advisor Ian Oakley for his orientation and support. Ian was always there to listen to me and to give me advice. He was always available to answer my questions even by e-mail, being so quick and so assertive. He taught me how to write academic papers and how to achieve goals. His constant guidance made me find solutions and paths.

Many thanks to my advisor Valentina Nisi for being so supportive while I was choosing the master project.

Besides my advisors I am grateful to Eng. Filipe Santos for his availability and confidence in providing me material.

A big thank you to my very close friend Paulo for listening to my concerns and ideas. Thank you for confronting my work and making me see that another path would absolutely enrich my thesis.

Finally, special thanks to my family for believing in me. They gave me the opportunity to live and be where I am now.

Content

Figures

Introduction 1

"The thread-up and thread-down of the weaving process corresponds to the 0 and 1 binary logic of computer circuitry."

Sabine Seymour (2008)

1. Overview

Wearable computing is a new area of technology, explored for the last four decades, with some unique aspects such as more convenient access to monitoring data. For instance, health monitoring is intended to be used continuously, and it is much more practical to access information with a technology integrated in clothing than to have to go out to access this data.

It is also convenient having technologies integrated into clothes because in some situations it is not opportune to have things in one's hands. Wearables permit interaction through voice, body movement or body temperature, for instance, and in this way they make hands-free possible. Wearables also make access to data feasible wherever it is wanted, making access to it easy.

They are personal. For some of them, only the person wearing it has the data feedback with no one else noticing. For instance, when hugs are sent and received just the one wearing it feels the hug. It is a private communication. It strengthens relationships between friends, mainly when they are abroad, because the system is always available so that they can get in touch easily.

They give real time feedback to people, and in this way they have immediate access to data. This can make them change behaviors. For instance, health feedback can lead people to change their eating habits and become healthier.

2. Technologies

Humans have long wanted to integrate computers on their own body, and nowadays we are taking huge steps in that direction. Technology is much more advanced than it was ten years ago. We have moved away from desktop to laptop computers and currently we are at the stage where we have computers in our pockets. It is already possible to develop some of the wearable ideas of the science fiction movies. The glove that Tom Cruise wears in *Minority Report* has already been developed by MIT. With a special glove, Tom Cruise controls a huge screen in front of him (1). This is a new way of controlling interfaces by wearing a special glove and using hand movements instead of the mouse. However, wearables like the one that Robert Downey wears in *Iron Man* is not yet possible to replicate. Robert wears a powerful armored suit that gives him like super hero powers. Science fiction is and will ever be a fertile field for wearables.

Technology is undergoing huge developments. Nowadays there are a vast variety of materials that can be used on wearables, from solar cells to conductive fabrics. The technology used in each project depends on which interaction is desired and the degree of comfort and aesthetics wanted.

Wearable computers depend entirely on technology. They are made from components that can be embedded in clothes. As components evolve technologically, better and high-tech wearables can be developed, and new ones emerge. A wearable computer is a computer that can be worn on the body wherever you go. The main advantage is that they make it possible to have diverse types of interaction, such as body movement, voice, eyes or hands.

3. Main Applications

Wearable computers are valuable in the fields of medicine, military, sports and fashion.

The wearable computers for medicine are attached to the patients and used to for a more effective prevention and treatment of diseases. They are being used for data processing, health monitoring and guidance. Carnegie Mellon University developed a wireless device called DiMA (2), Diabetes Management Assistant, by which patients can monitor and regulate their glucose level. Since it is a wireless device, it sends the data to the patients' and their doctors' computers. Thus, patient and doctor are on the same page without obtrusiveness. Another example is the Drishti (3) wearable, which gives guidance to visually impaired and disabled people. It is a head-mounted wearable that gives guidance based on a GPS system, taking into consideration obstacles in their way.

The wearables for the military are developed to monitor the field around soldiers. They are designed for warfare conditions. The main concern in their design is the battery life. They must be reliable in terms of battery life, as it is a high-priority requirement that the wearable always be available whenever the soldier needs it. There has been some research on exoskeletons to help injured and life-long paralyzed people so they would be able to move their body again, but researchers are considering other applications to commercialize them, such as to help soldiers carry heavy weapons and aid them in hostile situations. The USA is currently doing research to create exoskeletons for its army. DARPA (Defense Advanced Research Projects Agency) robotics developed an exoskeleton to help soldiers carry supplies, by which it is possible to carry up to 75kg at 1.3m/s (4).

The wearables for sports are produced with the aim of encouraging people to do exercise and monitoring and guiding their exercise, as well. There is a prototype developed by MIT called

TIKL (5), which means Tactile Interaction for Kinesthetic Learning. This wearable is an extension of a teacher, giving feedback on the joints if the wearer makes an incorrect movement. In this way he can learn a new sport quicker by himself. This suit can make the learning of golfing tactics considerably easier. With this device we become independent, because we can train and learn sports by ourselves.

The wearables for fashion are clothes, accessories or jewelry that merge aesthetics and style with technology. They are used to transmit, receive emotions and experiences. They can be customized and personalized. Moreover, attaching technology to clothes makes them interactive. Columbia Sportswear Company launched a pair of boots on the market, called Bugathermo (6), which heat your feet for hours. It is a pair of boots for winter with a heating system. The wearer can control the desired temperature and charge them when the battery runs out.

As we can see, more and more our bodies are becoming extensions of technical advances. Products whose construction once seemed impossible are now a reality.

4. Goals and Motivation

The motivation to do this project comes from the possibility of creating products far beyond conventional ones. Although wearables have been developed for several domains and purposes, they still have common issues.

One is the reliance on buttons. For instance there are wearables that can be integrated into clothing, which are already being commercialized, and are only using this technology to control audio or phone devices. So, the aim of this thesis is to explore new ways of interaction such as stretching or pulling.

Second is the reliance on displays. Several wearables use screens to give feedback to users. However, they are not as flexible as clothing. So, the goal of this project is to use voice feedback instead of using visual feedback through screens. There is a research project that uses this kind of feedback, which is Earpod, and their findings hint that controlling menus through audio is reliable (7).

Finally, it is also an aim of this work to implement and test the input devices created, developing a prototype application in a piece of cloth.

The new era of computers is the wearables. Computers are getting smaller and smaller and they are becoming capable of integration into clothing. It is very enriching to be involved and be able to contribute to this field with new types of input devices.

5. Contributions

Reading this thesis, people will acquire the knowledge needed to build new types of input for wearables described above in this report. They are very manual and highly crafted input devices, all documented with photos and circuitry schematics with all steps presented.

Readers will learn the advantages and disadvantages of conductive materials, such as conductive thread, conductive fabric and stretch sensor and also good ways to isolate conductive circuitries. They will also be made aware of which materials are suitable and which ones are not suitable for use on wearables.

6. Thesis Outline

This thesis is made up of six chapters, which are briefly reviewed below.

Chapter 2 shows the state of the art of wearable technologies as well relevant projects. It begins with a general overview about wearables. Afterwards there is a presentation of the current trends of wearable technologies and also what wearables have been developed up to the present that use those technologies.

Chapter 3 is about the construction and testing of new inputs to wearables. All inputs were developed manually. Afterwards, a pullover was developed in order to test these inputs. This pullover uses these inputs to manipulate an mp3 music player. The wearer can play, pause, stop music and select and attach which albums he wishes.

Chapter 4 shows the construction of the pullover. All steps are documented with photos and circuitry schematics. This wearable was built through an incremental process. Several modules where developed. Each one was responsible for a different task on the mp3, and after each iteration a new module was attached until the pullover was totally built. During the development, software decisions were already made so that the implementation could be possible. All implementation code is in the appendices because it was not appropriate to put it in the chapter; however, it is not less important.

Chapter 5 deals with the evaluation of the pullover, based on tests of its usability, in which several people participated. This chapter describes the process used to conduct the users through the test. Difficulties encountered while interacting with the pullover were registered, and all the results are analyzed and presented.

Chapter 6 shows the conclusions of this work, setting forth which goals were achieved. It talks about the contribution of building these input devices for the field of wearable computers. It presents the main findings from the development of this thesis. Finally, it shows which further work can be done.

State of the Art 2

The first part of this chapter deals with an approach that talks about which fields compose a wearable. Talking about issues/ challenges, tradeoffs and desirable factors in regard to wearables, and giving some examples of prototype wearables. The second part it is a dictionary of the different components that can be used to create wearables.

1. Analysis Approach

Smailagic introduced an approach in which wearables are composed of three fields: human, computer and application (8). Each of them has different aspects that should be taken into consideration while building a wearable.

The human field involves the wearability; it is the interaction between the body and the wearable. In other words, it is how comfortable the wearable is. This not only involves comfort but also social acceptance, status and aesthetics. Social acceptance means that no wearable should embarrass anyone. The users' culture has to be analyzed, and the wearable must be uniform with it. Status is another important aspect. Some groups of people wear objects only because of class. For instance, a sign of status is a diamond ring. People use it because they feel a cut above, like belonging to a higher social class.

The computer field involves the software implementation, interface design and hardware development. Software implementation is related to architectural and logical decisions. Interface design is related to the structure of visual aspects, decisions of input and output technologies. Hardware development is related to the decisions concerning physical components like fabric, battery, and size.

The last field is the application. It is related to the functionalities of the wearable. Only essential tasks should be implemented on the wearable. They tend to have lots of wires, with several gadgets attached. If secondary tasks which are not essential for the wearable context are attached, more complexity arises at hardware, software and application levels, without necessity.

So, according to Smailagic, in order for a wearable be worn, it needs to satisfy those three fields.

2. Non-Technical Issues

There are challenges that wearables need to overcome as design, acceptability, wearability and privacy. Each one of them is described in the next sections.

2.1. Design

Designing interfaces for wearables is a challenge. They need to go beyond human computer interaction principles and guidelines. Human computer interaction principles are designed for activities on which the user has total visual focus. It is about providing efficient feedback to the user, structure visual components, providing help and documentation, reducing short-term memory so that the user does not need to remember steps to perform tasks. These are some of the human computer interaction principles that are followed to design desktop computer interfaces. However, wearables have to go beyond them. Contrary to computer interfaces, wearables are used while users are performing other activities, and their attention is divided between different tasks.

There are two requirements that wearable interfaces have to accomplish. Firstly, they should not demand full attention. Secondly, since wearable interfaces are composed of input and output technologies, they should be small and light. Small and light technologies attached to clothes are easily unnoticed by the wearer, which makes it comfortable. However, is not only making the hardware smaller but also making the hardware regard the human body as a context.

They need to be developed taking into consideration the form of the human body. It is not only a matter of making them smaller. Since our body form is convex, devices should have a concave shape. They are more stable and fit better than a plane device. If people are going to use wearables daily they need to be as comfortable as possible. They need to be recognized as an extension of our body. The desktop mouse is an example of this aspect. After being used several times, it is recognized as extension of our hand to interact with computer interfaces. (9)

Beyond the issue of regarding the human body as a context and not demanding full attention interfaces face another challenge, which is to visualize and access information in a suitable way. People can do different activities while interacting with wearables. Sometimes it can be more suitable to have hands-free access to data, at other times it may be more suitable through eyes-free access.

Most current wearables use small screens to display information, and buttons to control data. Screens are not flexible and washable, and in the situations where hands- and eyes-free are needed, interacting with screens is not the best solution. Also, when using screens in outdoor applications, the sun will be a problem because of the glare. In regard to buttons, they are not the best solution for hands-free, either. Buttons have another problem which is that they are being used today to control continuously linear data, for instance, controlling iPod sound volume. (10) Clearly, the sound volume is a range, which made it continuously linear data. Wearables are using buttons to control this kind of data, where sliders or levers should be used, for instance. (11) When controlling sound volume with a button, if the user wants to change the sound volume from loud to quieter, he needs to click it several times or hold it for some time, which is not user friendly.

There is wearable research to use sound output to control information, for instance the Ear-Pod. The EarPod has a touch-based circular menu, in which the user has selection feedback via audio. It is similar to an iPod, but it does not have any LCD display. It demonstrated that it is feasible to navigate through menus only with audio feedback. (7)

Others are using gestures as input to control information. GestureWrist is a prototype which can identify hand gestures and in this way the user can interact with applications, while performing different hand gestures. (12)

There are other prototypes that are using sensory associations to present information, such as the Hug Shirt (Figure 1). The Hug Shirt is a prototype which sends hugs over distance. The user has a shirt with sensors that feel touch pressure, skin warmth and heart-beat rate. Then hug is resent and the friend receives the hug through the cell phone and he feels the hug on his shirt. (13)

Figure 1 – Hug Shirt sending hug

These types of interaction have downsides. Gesture interfaces have drawbacks such as, performing gestures while being focused on other activities. Also, if there are a large number of gestures that the user can do, it would be difficult for the user to memorize them. The downside of sound feedback devices is that in noisy environments, the user will probably not hear the feedback.

Choosing which type of interaction to use on a wearable is a complex task, because it depends on the context of the application, which makes developing wearable interfaces more complex than desktop interfaces. An evaluation must be made as to how many visual, auditive and cognitive tasks can be combined.

2.2. Wearability

Despite design challenges, there are also wearability challenges, mainly essentials for products that will be commercialized. Wearability is how comfortable a wearable is. For a wearable to be comfortable, it needs to be accepted as an extension of our body and should be also unobtrusive.

In order for wearables to be accepted as an extension of our body, they need to be accepted as belonging to our bodies. Humans have a perception that what is between 0-13 cm around their body belongs to them. The thinner and nearer as possible the device is from the body, the better the human perception is that it belongs.

They also need to be comfortable. Considerations such as the growth of body muscles also need to be considered. Muscles can change their fat and consequently wearing the wearable can be uncomfortable. Adjustable strips, stretchable fabrics can be a solution. However, stretchable fabrics can become too tight if there is a big increase in the amount of fat, or too loose. Also fasteners or Velcro can be used. However, everything depends on the type of situation and the place where the wearable will be attached.

Devices should be attached at unobtrusive places such as the ones showed in Figure 2. Heavier devices should be placed on the stronger muscles, and the components should not interfere with the user flexibility. For instance, a weightier device in the upper arm will not cause fatigue because is a strong muscle. Also wearables should not block flexibility or the human body. Space must be created on the wearable to permit body movement. It is important notice that comfort is not only associated with fatigue but also with harm and anxiety. (9)

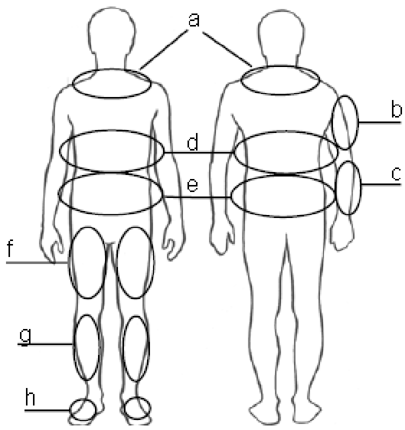

Figure 2 - The most unobtrusive areas: (a) collar area, (b) back of the upper arm, (c) forearm, (d) rear, side, and front ribcage, (e) waist and hips, (f) thigh, (g) shin, and (h) top of the foot

2.3. Acceptability

Beyond design and wearability issues, there are also acceptability issues. There are social rules for each society that should be observed. People usually do not tend to go against those rules. For instance, in some cultures no one will carry a head mounted wearable to a formal meeting. Beyond social acceptance there are gender issues. The Stress Outsourced wearable (14) is a social wearable where friends can send and ask for massages to relieve stress through social networking like Facebook or Twitter. This is an example of a jacket that would not be used for all types of relationships, for instance, between masculine friends.

So, acceptability is about being uniform with personal values and customs.

2.4. Privacy

Beyond design, wearability and acceptability issues, there are also privacy issues. Privacy issues are related to wearables that have embedded webcams which can record other people. People that are incidentally recorded while the wearer is using his device may react negatively. This is also true for monitoring systems, since personal information is being captured and sometimes sent through wireless. There is a need to guarantee that unauthorized people will not access the information, especially on health systems, where trust is a high priority. (15)

3. Technical Issues

Computers have become smaller over the years; however they still have to become mechanically flexible, water-resistant, with low production cost and battery efficiency.

In order to have a higher energy battery life a huge and heavier battery is needed. Smaller batteries are lighter; however the battery life is shorter. Since wearables should always be available, in some applications is not possible to reduce battery size. Reducing the battery size will reduce the wearable's performance.

Wearables are portable devices, and to carry a heavy battery is uncomfortable, while on the other hand, carrying a small battery has the outlet availability problem. For instance, in a warfare environment a soldier obviously will not find outlets to recharge his wearable.

Some research is being done in this field and there are several hypotheses to solve this issue. The first one is to use a secondary battery, and the wearer would only need to recharge it. This will add one more routine task to the wearer's life. Also, the wearer could forget to recharge the battery. A good way to avoid this would be after taking off the wearable and putting it on the closet hanger it charges automatically, like toothbrushes and wireless home phones.

The second possibility would be while the wearer is walking on the streets, the wearable battery is automatically being recharged through wireless. A disadvantage of this method is the same as the mobile phones: there are places where it is not possible to catch the antenna signal.

The third possibility would be to have a self-sufficient battery, which could be capable of generating its own energy. This could be achieved by using solar cells. However, in indoor environments solar cells do not work. So, there is another way, which is to generate energy through body movement. There is a pair of sneakers that actually can generate its own energy. The way that it works is when the wearer's heel strikes or bends while walking, energy is generated. This method is capable of generating energy for small circuits like a PIC microcontroller and one RFID. (10)

An issue that arises with battery recharging is the heat, which is another challenge that technology has to overcome. While batteries are charging they get hot and a heat dissipation system is needed, because the human body is very sensitive to heat. Besides batteries, microcontrollers also get hot and can be uncomfortable. A way to avoid this could be by reducing the microcontrollers performance.

According to Starner and Maguire (11) from MIT, our body is the most effective and complex thermoregulation system. So, what they suggest is to put wearables in contact with skin, because the skin can regulate our body temperature through sweat. However, devices should have a textile under them in order to absorb the water and avoid the rubber glove effect. Also lighter devices should be placed on the bottom part of the arms and legs because of the pendulum effect. Basically, the movement of our arms and legs while walking makes the airflow and in this way a wearable can be efficiently cooled.

4. Wearable Technologies

This section describes technologies which can be integrated into clothing to create wearable devices. This section intends to provide an overview of the technical possibilities for building a wearable.

It is divided into sections covering input, output, computation, energy and materials, which represent the key components required to build rich interactive systems. For each technology in the sections input, output and computation, an example of a wearable is presented.

4.1. Input

There is a wide range of input devices that are suitable for wearable scenarios, including those which require explicit activation (such as switches or toggles) and also those which passively (and typically continuously) sense aspects of the environment or a user's activity. In all cases, these items typically come in a slim form-factor and are either sewn on the top of the fabric, or sandwiched between two layers. A non-comprehensive list of input devices is described in the next subsections.

4.1.1. Button Sensor

A button sensor can be compared to a light switch. It closes when pressed, otherwise it is open. It is useful for interactions that only have two states, like turn on and off. Buechley developed a type of buttons special for wearables, which can be easily sewn on clothing and besides can be washed. Cooperatively with SparkFun a commercial version was developed, which is the one that is at Figure 3. (16) (17)

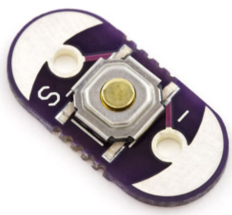

Figure 3 –Lilypad pack Button Sensor

A button sensor is very common in wearable computing and is heavily commercialized on the market. The leading manufacturer of wearables is one which is selling button sensors to control audio or phone devices just by plugging them (Figure 4). (18)

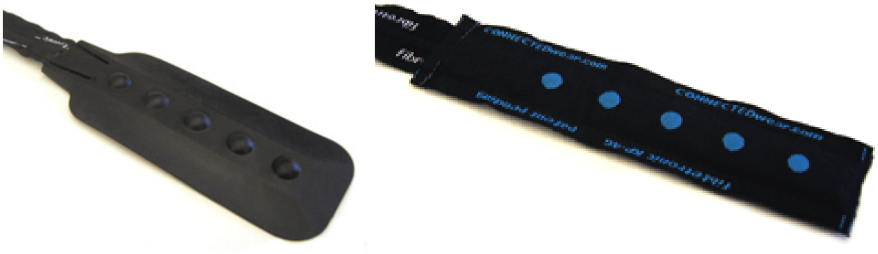

Figure 4 – Buttons sensors by Fibretronic, Linear Keypad at left and Lite Keypad at right

Woolrich, O'neil and Ecko are attaching these sensors to their products (Figure 5) to control the iPod, and currently are available to customers. (10)

Figure 5 – Hoodie by Ecko

DEW Company also developed a wearable integrating these keypad buttons, called The Quiver (19). The wearer can control an iPod, mp3 or mobile phone by attaching it to the wearable pocket (Figure 6).

Figure 6 – The Quiver by DEW

4.1.2. Capacitive Sensor

These kinds of sensors are used in situations where low to no force is desirable. They sense proximity and touch by detecting the changes in the capacitive value induced by the body. It is possible to make homemade capacitive sensors and attach them to clothes, for instance by connecting a foil to a microcontroller.

As can be seen at Figure 7, the person interacts with a piece of foil that acts as a capacitive sensor, because the microcontroller is programmed to act as a capacitive sensor. So, when the person brings his hand closer, the sensor is activated.

This sensor is a good replacement for those buttons that do not require human pressure.

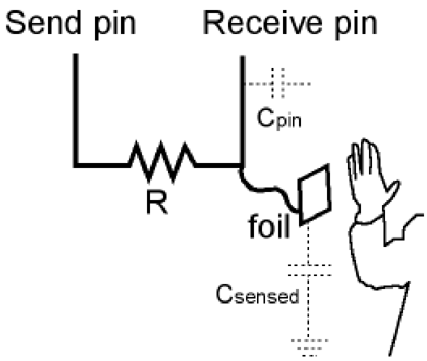

Figure 7 – Capacitive Sensor Illustrative Scheme (20)

GestureWrist and GesturePad are two prototypes in which the first one detects hand gesture and the second one works as a module sensor. GestureWrist is a bracelet that can detect grasping and pointing clearly through capacitive sensing. Since it is a prototype to detect hand gesture, it was also attached an accelerometer to measure the inclination of the forearm. GesturePad consists of a layer of capacitive sensors array, which can be attached to the inside of any clothing and measure the capacitive values independently. (12)

4.1.3. Stretch Sensor

Stretch sensor is an analog sensor that measures the stretch, and can be washed (Figure 8). When stretched, its resistor gradually increases. When relaxed, it has a nominal resistance of 1kΩ per 25cm. Its resistance will approximately double to 21KΩ per 25cm when stretched 50%.

Figure 8 – Flexible Stretch Sensor

This sensor can be used to measure force and stretch and can be put, for instance, around the chest to measure the respiration rate. Different values will be measure in the inhalation and exhalation and in this way it is possible to measure the respiration rate by it.

A flexible Stretch Sensor can be obtrusive and expensive compared with knitted sensors, so several prototypes are using knitted sensors to develop prototypes. At Philips research laboratories a jacket to detect body movement was developed by attaching several knitted stretches (Figure 9). When the user wears the jacket, he is asked to put his arms down, outstretched, forward and then up in order to calibrate the system. In this way, the maximum and minimum values of each stretch sensor is obtained, and are used in the algorithms for detecting the angles of the joints. Afterwards, motions and gestures are recognized. (21)

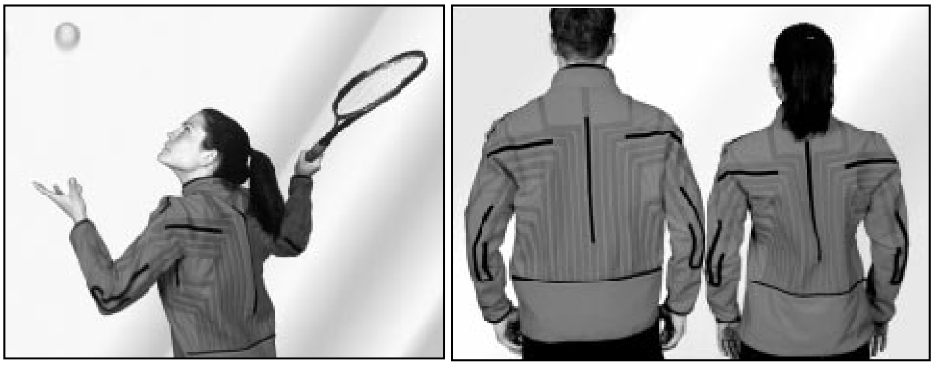

Figure 9 – Jacket, stretch sensors are shown in black

4.1.4. Bend Sensor

They sense bending. They are a kind of resistor that increases their resistance when bent from convex to concave shapes (Figure 10). It is the unique component with this ability. They are also pressure sensitive, and can be used as a force or pressure sensor.

Figure 10 – Flex Sensor

A good place in the human body to use it is in joints. In this way arm or leg movement can be detected without making the user uncomfortable while wearing it.

GaitShoe (22) is a wearable which was built to analyze gait movement. It has two bend sensors embedded to detect foot flexion. The bend sensor values change according to the angle of flexion. Beyond the bend sensor it has other sensors, making it fully possible to detect patterns while walking. This wearable revealed being accurate to detect heel-strike and toe-off and also found gait patterns for healthy gait and Parkinsonian gait, which is a disease that impairs motor skills.

4.1.5. Accelerometer

An accelerometer sensor measures acceleration and thus they can detect movement as well as inclination. An example of this sensor is the 3G Triple Axis Accelerometer. This sensor can be washed and is quite similar to the rigid button sensor. It was also developed by Buechley.

There is a wearable that uses an accelerometer sensor to detect if the wearer has the right posture (23). Firstly, the wearable learns the user's correct posture and then every time that he is not at the correct posture he is alerted through a vibration sensor that his posture is unhealthier.

4.1.6. Light Sensor

The light sensor measures the intensity of the light in a specific place. When exposed to the daylight, it measures 5V. Covering the sensor with your hand will measure OV. The light sensor can be washed. Like the button sensor in Figure 3, it can be sewn on clothes. It belongs also to the pack that Buechley developed to make clothes interactive.

A quite common use of this kind of sensors is to analyze the ambient surroundings. For example, Macintosh uses a light sensor to detect when it is too dark and then activates the keyboard light.

There is also a prototype developed by Carnegie Mellon University called eWatch (24). It is a wrist-mounted watch which is able to do online analysis while collecting data from the surrounding environment. As usual, the watches gives the time of day, but besides this, it is able to connect to the iCal (calendar of Macintosh) to upload the eWatch calendar. This wearable uses a light sensor to change the brightness of the display.

4.1.7. Pressure Sensor

A Pressure Sensor measures the force that is being applied. When pressure is applied on it, new values are read. It is possible to construct a pressure sensor only with fabrics. It looks like a sandwich. On one side a non conductive thread is fused with conductive thread. The other side is the same. In the middle it is only needed to put a velostat fabric and then sew around the edges. When the sensor is pushed the velostat resistor decreases. In this way it is possible to measure the force made on it. (25)

There is a wearable, SensorHogu (26), which is being used for sports and uses pressure sensors. It is a system that was developed to help Taekwondo judges and referees to give scores at a Taekwondo match. In Taekwondo, scores are given according to accurate and powerful kicks at legal scoring areas of the body. So, SensorHogu detects the amount of force delivered to the opponent and sends it wirelessly to the judges and referees. The amount of force is measured by pressure sensors. In this way, players and judges all benefit from this system.

4.1.8. Temperature Sensor

This sensor is a small thermistor type temperature sensor. It reads temperatures in Celsius degrees. It returns 0.5V for 0C degrees, and 0.75V for 25C degrees. For each degree it returns 10mV. A typical application of this sensor is to measure body temperature. It looks like the button sensor in Figure 3 and also can be washed and sewn on clothes. It is another sensor that belongs to the Buechley pack.

Dance performance has been gaining much attention from arts and science researchers. Park et. al, have developed a wearable which makes performances interactive. The actors are the ones that control the stage through their performance, for instance the lights. The wearable has inputs and outputs that make it possible to control the stage. In the future there will be embedded temperature sensors to measure actors' body temperature and reflect it on the dancers' stage clothes. (27)

4.2. Output

Like the input technologies there is a good diversity of output devices that are suitable for wearable scenarios. The output technologies are the ones which do something depending on the input value received. They do not analyze the surrounding environment. They only serve as actuators. They can be sewn on top of the clothes and also printed on the clothes. A non-comprehensive list of output devices will be presented in the next sections.

4.2.1. LEDs

They provide light and are quite often used in wearable computers. There is a huge variety of LEDs for wearables like the Tri-color, red, blue, yellow, white, green, ultraviolet color and infrared. The RGB LEDs do combinations between green, red and blue so that any color can be emitted. The red, blue, yellow, white and green LEDs emit only their color. Finally the infrared LED emits infrared light. They can be washed if well sealed. Also there are RGB LEDs that have been developed also by Buechley, can be washed and sewn easily on clothes (Figure 11).

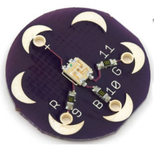

Figure 11 – LilyPad LED

There is also another type of LEDs that are quite common on wearables. It is the high-flux LEDs. They are square and can easily be bent and attached to the clothes. They are much brighter than the typically LEDs.

Another kind of LED that is quite interesting for wearables is the ultra violet LED. When combined with photochromic and phosphorescent ink they produce an interesting light emitting effect, causing a bright object to glow.

The application presented is the AirWear (Figure 12). It is a shirt that was developed at Carnegie Mellon University by Kim et. al. This shirt measures the surrounding air quality and expresses it. It is made with an arduino LilyPad, a VOC sensor and LEDs. The VOC sensor measures the air pollution and the LilyPad controls the LEDs. If the air pollution is low the LEDs blink from left to right slowly, otherwise they blink very quickly. (28)

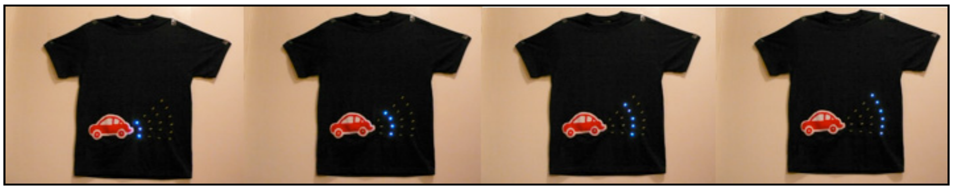

Figure 12 - WearAir, a lightened T-shirt, four arrays of LEDs lighten in a sequence from left to right

4.2.2. Vibration Sensor

It vibrates when it receives an input. It works great as a physical indicator without notifying anyone but the wearer. It is a small sensor also developed for the LilyPad arduino. It can be washed.

TapTap is a haptic system which can record and transmit human touch remotely, through a scarf (Figure 13). This scarf uses vibration sensors, solenoids, air bladders and Peltier junctions to represent the touch. The vibration sensors and solenoids replicate stroking. The air bladders are filled with air to simulate pressure, like someone pressing on you. The Peltier junctions are activated to simulate the heat of human contact. (29)

Figure 13 – TapTap scarf, example of some positions that the wearer can use it

4.2.3. Buzzer

Buzzer sensor uses two input/output pins on the microcontroller to create different noises based on different frequency of input toggling. The higher the input frequency pitch, the louder the noise it generates. It is loud enough to be heard inside a pocket but it is not obtrusively loud.

There is a hat (Figure 14) that detects when it is being pressed, by a pressure sensor, and then the buzzer is triggered and emits sound (30). It does the police siren sound.

Figure 14 – New York police department hat prototype

4.2.4. Electroluminescent Wire

It looks similar to neon but has the flexibility and versatility of wire. It glows when an AC current is applied. It is typically comprised of a dielectric layer between two conductive electrodes and a layer screen printed with phosphor powder.

The electroluminescent wires are flexible to bend into any shape. They create no heat and can be applied to nearly any surface. They are waterproof if the connections are completely sealed. However, this is not true if submerged in pressurized water, like in the ocean, unless the cover of the EI-wire is made of plastic. So, it can be washed by hand and it is weather proof.

The skirt in Figure 15 uses electroluminescent wire. When there is too much air pollution at the surrounding, it makes more electroluminescent wires glow. (31)

Figure 15 – Skirt with Electroluminescent Wires, air pollution awareness

4.2.5. OLED Display

The OLEDs Displays are used to present data. They are similar to an LCD screen. However, the OLED displays are really smaller, thinner and flexible (Figure 16). It is a really interesting device for use in wearable devices, because of their flexibility, size, durability and broader range of operating temperatures. However, they cannot be washed.

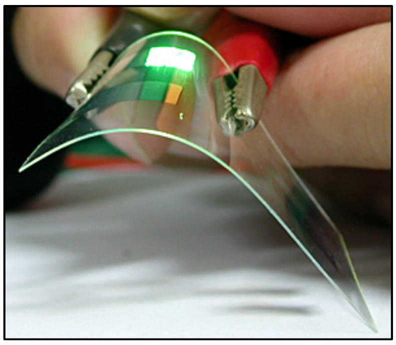

Figure 16 – OLED Display

Philips developed a screen which presents information through an LEDs screen. It is Phillips Lumalive Shirts (Figure 17). These LEDs are tri-color LEDs which can have any color wanted. In this way each LED works as an pixel in a photo. Phillips researchers discovered that by sealing them into a laminated plastic, they could make a flexible display. (32)

Figure 17 – Phillips Lumalive Shirts

This wearable is capable of making the clothes interactive, and also promotes socialization.

4.2.6. Inks

There are three types of inks for wearables; thermochromic, photochromic and phosphorescent dyes.

The thermochromic inks change their colors according to the temperature. The photochromic change colors when exposed to ultra-violet light, for instance sunlight or UV LEDs. The phosphorescent ink absorbs UV light from sunlight or from UV LEDs and then glows.

These inks are really easy to apply. It is only needed to paint the wearable with them as if they were another fabric ink.

Reach (33) is a project, the aim of which is to find new ways of communication and expression in order to be included dynamically in clothing. The first prototype developed used hats. These hats change their state when they are near other people wearing the garments. They were painted with thermochromic ink and when they came near another person with the garment, the circuitry voltage increased and since behind the thermochromic ink there is a layer of conductive thread the thermochromic ink is heated. When two people get in contact, the dotted hat one is wearing grows flowers, while the flowered hat on the other person receives a dot (Figure 18).

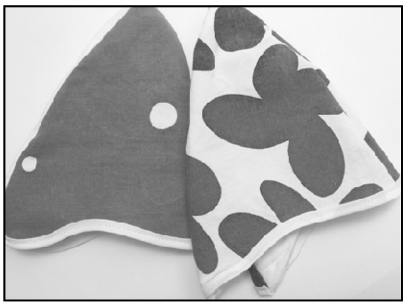

Figure 18 – Thermochromic hats

Another prototype involved in this project is a scarf which changes its state based on environmental temperature (Figure 19). When it gets cold outside the scarf heats up and the pattern changes. A temperature sensor triggers the heating of the conductive thread that is sandwiched between the two layers of the scarf. In this way thermochromic ink is heated creating different colors on the pattern.

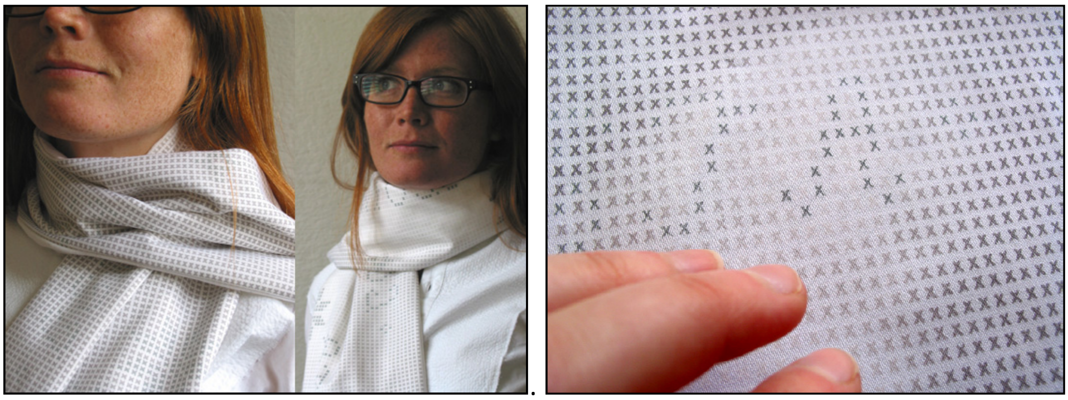

Figure 19 – Thermochromic Scarf creating patterns

Regarding phosphorescent ink there was a research project that was done with this ink. (34) It is not a wearable; however the ink was applied to a textile. It was a project made for an art museum where several text fragments were printed on fabrics using phosphorescent ink, on different kinds and qualities of fabric. Each fabric had its own shape. In order for people to read the text special devices were built. These devices were sticks with a UV-lamp mounted (Figure 20).

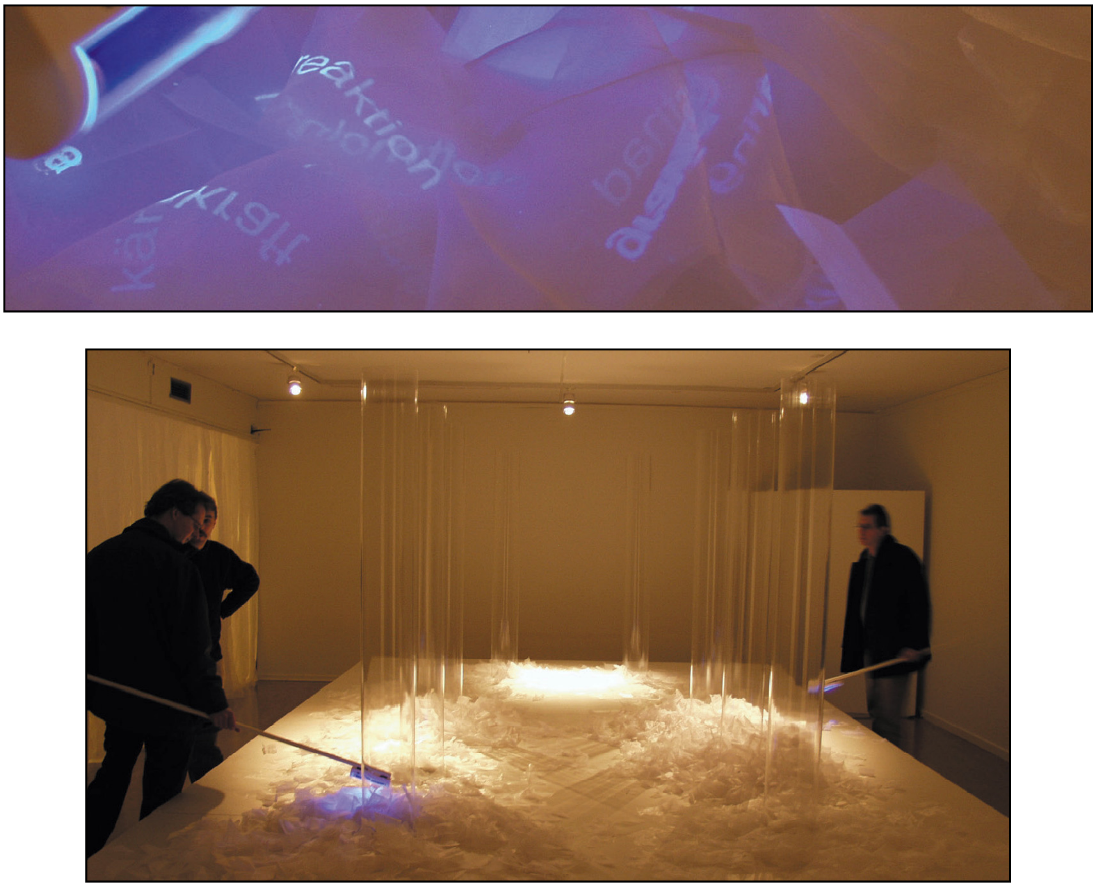

Figure 20 – Phosphorescent textile fragments

4.2.7. Shape Memory Alloy

This is a combination of two or more metallic elements composed of nickel and titanium. The most popular is Nitinol. They are the unique components that remember their shape. They can be bent into a particular shape by heating, and when cooled they will return to their original form. They have hardness and elasticity properties that change radically at distinct temperatures. They can be used to make fabrics shrink and curl with the application of a small current.

Kukkia and Vilkas are two animated dresses that change their shape (35). The Vilkas dress shrinks, revealing the knee and the lower thigh (Figure 21 at left). The Kukkia dress is composed of a 3D flower which opens and closes every 15 seconds (Figure 21 at right). Both of the dresses have the Nitinol shape memory alloy embedded, which when heated on the Vilkas dress shrinks the dress and on the Kukkia dress closes the flower.

Figure 21 – Shape Memory Dress (a) Vilkas (b) Kukkia

4.3. Computation

A device is needed to control all those sensors. This device should be very well selected otherwise it will be uncomfortable for the wearer.

There are several microcontrollers that can be used such as PIC, ATMEL, arduino and seeeduino. PIC and ATMEL cannot be washed and they are difficult to sew into the clothes, as well. Moreover, they are too big. Arduino surface mounted versions have the same drawbacks as PIC and ATMEL microcontrollers. The ones that are more suitable to sew on wearables are LilyPad arduino and Seeeduino film. They will be presented at the next sections.

4.3.1. LilyPad Arduino

So, a specific version of the arduino, the LilyPad arduino (Figure 22), is a good solution for a wearable microcontroller. It is cheap, easy to use, washable, and small and its software is open-source. LilyPad arduino is aboard that is 5 x 5 cm in size.

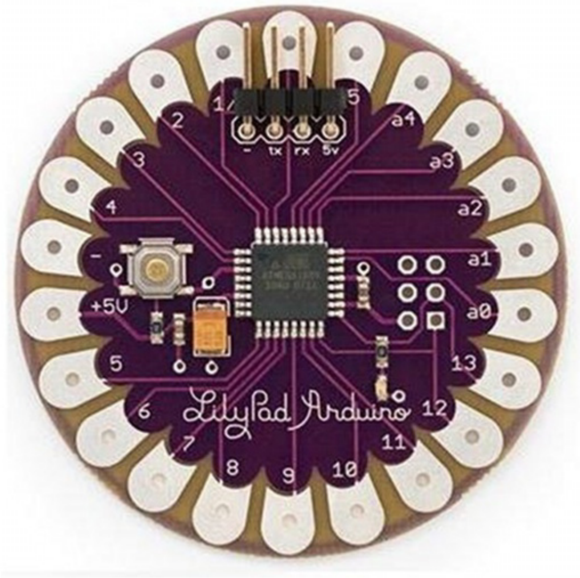

Figure 22 – LilyPad arduino main board

The LilyPad arduino is highly utilized in wearable prototypes. It is cheap and easy to use. WearAir, the prototype described at section 4.2.1 of this chapter, uses this microcontroller to control the sensors. Also the police hat that uses a buzzer uses this microcontroller to control the buzzer, at section 4.2.3.

There are two devices that can be attached to the LilyPad and used as a wireless communication between the arduino and other devices. The first one is the wireless Xbee communication and the other one is the Bluetooth Mate.

The Xbee shield is a radio transceiver that you can sew on to your clothing to create wireless wearables or attach to the arduino (Figure 23). There are different types of Xbee. They differ in their area of wireless coverage. Some cover up to 100 meters and other up to 1600 meters.

Figure 23 – Xbee Shield LilyPad with wire antenna

The bluetooth Mate works as a serial RX/TX pipe and any serial stream from 9600 to 115200bps can be passed seamlessly from the computer to the target (Figure 24). These units have been tested successfully over open air at 106m. It has the same pin out as the FTDI Basic, and is meant to plug directly into the LilyPad main board. Because the pin has been arranged to do this, the Bluetooth Mate cannot be directly plugged into an FTDI Basic board. The TX and RX need to be swapped.

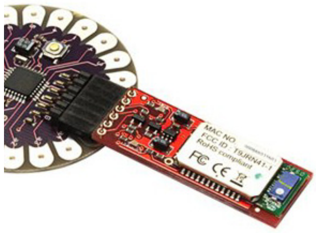

Figure 24 – Bluetooth Mate

Presently, the wearable developers are using the XBee because it is cheaper than Bluetooth devices and can interact with mobile phones via a short radio area.

Ubiquitous Drums is a wearable which allows the wearer to play drums by touching some areas of his body. There are four pressure sensors on the body where the wearer can press and play the drums. This prototype is connected with iPhone and in this way XBee is used to make this wireless connection. (37)

There is a shirt which reads how many unread emails are in the user's inbox (Figure 25), using a Bluetooth connected to a LilyPad arduino to communicate with an Android phone, and get the number of emails. Since there are only seven LEDs, if the user has three unread inbox emails it activates LED one and two. It is a good combination to show up to 126 emails (all LEDs will be turned on). However, it is not possible to show the user higher amounts of emails. (38)

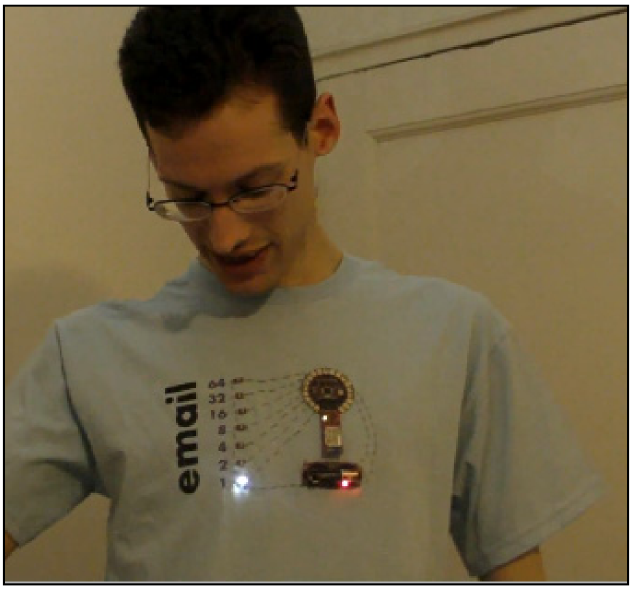

Figure 25 – Shirt using Bluetooth Mate

4.3.2. Seeeduino Film

Seeeduino Film (Figure 26) is another microcontroller suitable for wearables, which was launched on September 14th, 2010. It is suitable for use on prototype wearables due to being super slim and flexible. Each separated bar of Seeeduino Film is 2 x 2 cm. (39) Its pin map can be seen at Figure 27.

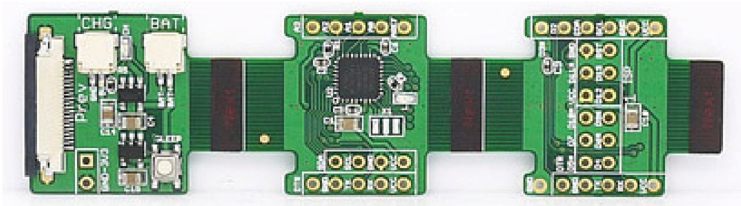

Figure 26 – Seeeduino Film

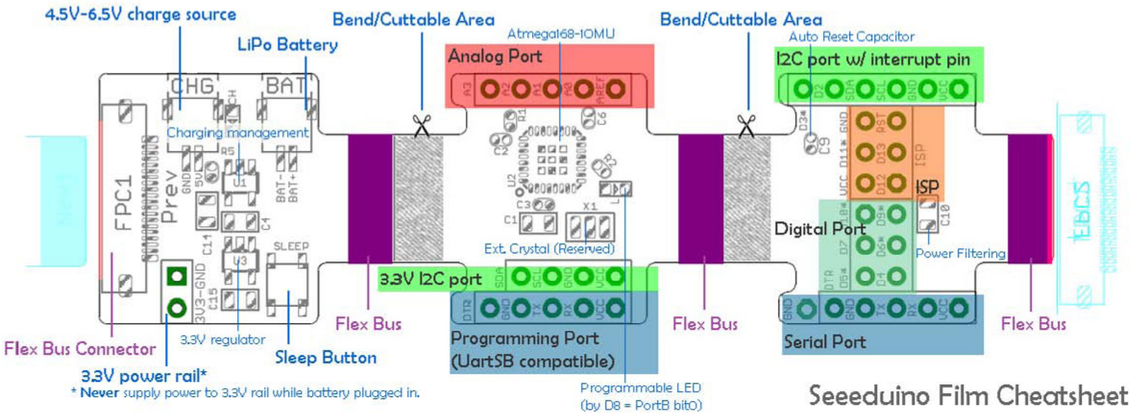

Figure 27 – Seeeduino film pin map and structure

Currently, there is only one sensor made for seeeduino film which is the seeeduino motion (Figure 28). It was launched on October 18th, 2010. Seeeduino motion frame is a motion board. It is composed of one barometer and one accelerometer, which provide sensing and logging of height, temperature, air pressure and motion.

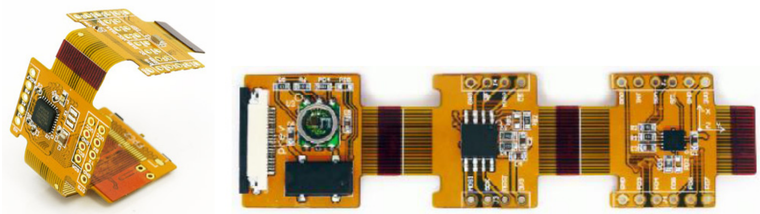

Figure 28 – Seeeduino Motion Frame

Attaching seeeduino motion to the microcontroller will be like building a chain (Figure 29). On wearables conductive thread can be used to connect the boards, instead of the FPC connector interface.

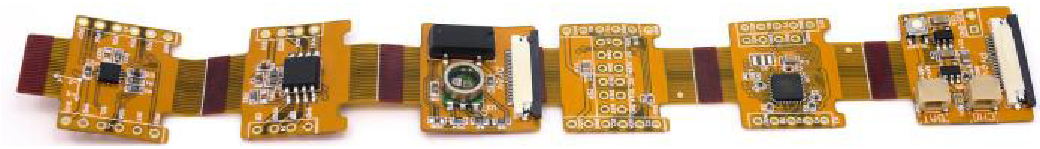

Figure 29 – Seeeduino Motion Frame attached with Seeeduino Film

This microcontroller is very useful to use at small apertures due to its size and slim board, for instance in a wallet. It cannot be washed and only have four analog ports.

4.4. Energy

When is not preferable to use a cable between the microcontrollers and the computer, a battery should be used. The batteries suitable for the previous microcontrollers are the following ones.

4.4.1. Solar Cells

Solar cells are made of a refined, highly purified form of silicon. They convert light energy into electrical energy. They are an excellent sustainable and renewable power source for projects. They offer the advantage of acting as light sensors, able to distinguish between light and dark and between different times of day.

4.4.2. LilyPad Power Supply 5V

It is a small power supply. It was designed to be as small and inconspicuous as possible. It is used with one AAA battery. It has the advantage of being able to be sewn into clothing (Figure 30).

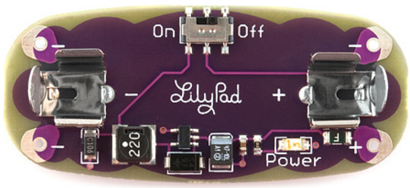

Figure 30 – LilyPad Battery 5V

4.4.3. LilyPad Power Supply 3V

This is a power supply (Figure 31) which can also be used with LilyPad arduino. It is suitable for projects which require a smaller battery, for instance because of aesthetic issues. However, this battery has a shorter duration than the one presented above.

Figure 31 – LilyPad Battery 3V

4.4.4. Seeeduino Power Supply

Seeeduino has a power management board, it is only necessary to connect a Li-po battery (Figure 32). Batteries between 4.5 and 6.5V can be attached and seeeduino regulates the voltage only for 3.3V. The power management board has a sleep button which by means of software can put the battery into power down or idle mode. To wake up the battery, it is only necessary to press the sleep button or trigger any external interrupt or timer.

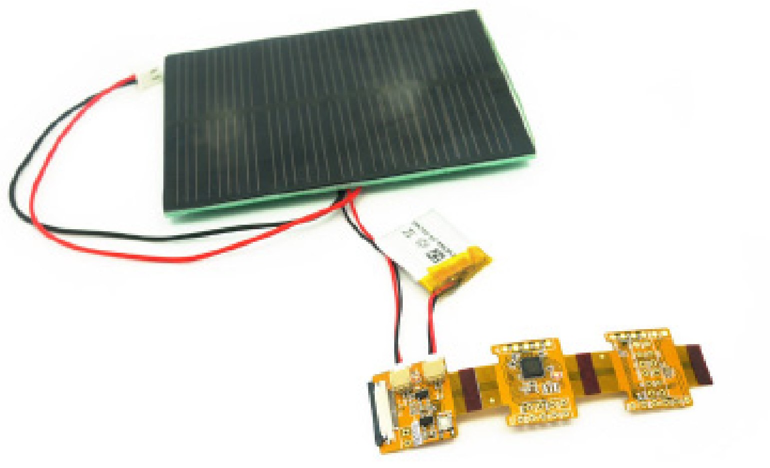

Figure 32 – Seeeduino Film, Power management board

4.5. Materials

There are conductive materials and non-conductive materials suitable for wearables. Conductive materials such as thread, fabrics, and inks allow the easy connection of input, output, communication, and energy devices on the textiles. There is a good variety of them, some more suitable for certain situations and others for other cases. A non-comprehensive list of them is presented in the next sections. Firstly, all conductive materials will be presented and then all non-conductive materials, heat shrinking tube and magnetic ink.

4.5.1. Conductive Thread

A conductive thread can be used to connect devices in wearables. It is used as any other thread. In order to make the circuit, you just need a needle. Most threads are metalized with an alloy of various metals, which can include silver, copper, tin and nickel. The core is normally cotton or polyester. There are different types; each one is suitable for different purposes:

Silver Plated Nylon 117/12 2ply: a light-weight thread, with each 20cm having a 100Ω resistance.

Silver Plated Nylon 234/34 4ply: a thick thread that frays easily. It has a low resistance, each 20cm having 17Ω resistance. Figure 33 shows this type of thread.

Conductive Thread from Lame Life Saver: a medium-weight thread with minimal fraying. It is a bit thick but is okay for sewing by hand or machine. Each 20cm has 20Ω resistance.

234/34 4 Ply HC Conductive Silver Thread: an excellent thread with low resistance and light fraying. The thread is thick like Silver Plated Nylon 234/34 4ply but it is much easier to work with because of its light fraying. The main drawback is the cost because it is only available in a large 1kg cone. However, a sample pack can be ordered for $30. Each 20cm has a 9Ω resistance.

235/34 HC Conductive Silver Thread: an excellent thread too, also with low resistance and light fraying. Moreover, it is quite thin. However, it is quite expensive but a sample pack can also be ordered for $5. Each 20cm also has a 9Ω resistance.

Resistive thread 66 Yarn 22 3ply: an excellent thread to use as a resistor. It is thin and good for hand and machine sewing. Each 20cm has a 2KΩ resistance.

Stretch conductive thread: The conductive thread is not stretchy by itself but when sewn on a stretch fiber, it works as stretch sensor. It is a nice thread but can be a little annoying to work with. Each 20cm has a resistance 800 - 1.8KΩ.

Gunze & Mitsufuji Thread: the conductive thread with the lowest resistance; however, it is hard to find and expensive. Each 20cm has a 6Ω resistance.

Bekinox VN12/1*275/100Z Thread: emits heat when conducting voltage. It is possible to change thermochromic textile paint with 5V 400-500mA. This will differ with kinds of thermochromic paints and environmental temperature.

Figure 33 – Conductive Thread

4.5.2. Conductive Yarn

They are the same as the conductive thread. The difference is that is possible to use wool knitting gadgets with them. The problem is that they are very expensive and also are sold in industrial quantities.

4.5.3. Conductive Epoxy

Conductive Epoxy (Figure 34) is an adhesive which allows the making of electrical connections between components and materials where the heat of a soldering iron could be damaging. It is an electrically conductive adhesive with copper or silver filaments. This component is usually used to repair traces on circuit boards. Also, it can be used in textiles, paper and other materials to build circuitry.

Figure 34 – Conductive Epoxy

4.5.4. Conductive Fabric

Conductive fabric (Figure 35) is highly conductive. It is composed of metallic elements such as silver, nickel, tin, copper and/or aluminum. They are lightweight, durable, and flexible and have the capability to conduct electricity with low resistivity. They can be sewn like traditional textiles. They are soft, washable and wearable.

They can be used to create flexible and soft circuit boards and sensors.

Figure 35 – Conductive Fabric

4.5.5. Conductive Hook

Conductive Hook (Figure 36) is a nylon hook coated with metallic elements, usually silver. It is like an ordinary hook but can conduct electricity with low resistance. They can be sewn like traditional textiles. They make excellent switches and connection points for electronics sewn on textiles.

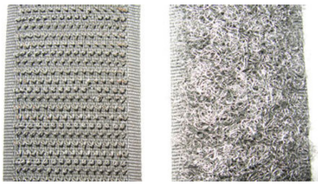

Figure 36 – Conductive Hook

4.5.6. Conductive Tape

It is made of copper pigments. It is useful for making connections between components and also to repair solar panels.

4.5.7. Conductive Pen

It is a pen with high purity silver pigments. It is really good for making repairs, for instance when the sensors are not in contact with the microcontroller pins but they are almost touching (Figure 37).

Figure 37 – Nickel Conductive Pen

4.5.8. Metal Fastener

They are an excellent way to make electrical pluggable connections. Moreover, most of the wearables come with them.

They are hooks, loops, hook clasps, magnetic clasps, screws, socket clasps, buttons and zippers. For instance, the beads are really good to do the tilt sensors.

4.5.9. Conductive Paint

They are useful to make textiles conductive. They have a high percentage of silver pigments (Figure 38). Their major problem is that they crack and lose conductivity due to the porous and uneven surface of the fabrics. Another way of having conductive paint is mixing graphite powder with ink. Since graphite is conductive, mixing both of them will create a conductive ink.

Figure 38 – Conductive Paint

4.5.10. Heat Shrink Tubing

Heat Shrink Tubing is composed of polyolefin. It is a flexible tube that shrinks when heated. It can be heated with a hair dryer. It comes in a variety of colors and diameters (Figure 39). It is used to isolate conductive thread and electronic components.

Figure 39 – Heat Shrink Tubing different sizes

4.5.11. Magnetic Ink

This is a water-based ink mixed with metal particles. It creates a magnetically receptive surface, turning any material into a surface where magnets are attracted. It can be used in any fabric to convert it into a magnetic surface.

5. Summary

This chapter focused on two main points, overview of wearables and a dictionary of technologies. The first part discussed an approach that wearables should follow. Technical and non-technical challenges were discussed, showing which desirable factors wearables should have. In the second part, technologies were discussed, such as input, output, computation and materials, listing different types of wearables that were built using them.

Knowledge about principles and guidelines for the development of wearables was acquired, as well as technologies that are available for building wearables. Guidelines for do not require full user attention and embed wearable technologies as naturally as possible into clothing should be followed. Examples of wearables were given, which help inspire the building of new input devices.

In the next chapter, prototypes that were developed in this thesis based on these technologies and principles discussed will be presented.

Prototypes 3

At the beginning of this thesis several prototypes were constructed. To acquire technical skills with wearable technologies, and provide rich and embodied interactions were the main goals. Prototypes with interactions such as pulling and stretching were built.

Each section of this chapter presents one of the prototypes, in which a primary prototype description is given as well as the procedure followed to build them. Afterwards, a brief conclusion about their performance and advantages and disadvantages are given.

The first two prototypes were developed following a tutorial on instructables website. They were developed with the aim of becoming familiarized with wearable technologies, since this was the first time to work with these kinds of materials. All the prototypes were developed using arduinos nano and duemilanove, because they are more suitable for prototyping, since surface-mounted components can be attached to the breadboard and to these microcontrollers in a straightforward way.

1. Button Sensor

This prototype is a three-button sensor which is totally handmade (Figure 40). It is simple to build, and can be personalized for different types of wearables. It is possible to create it in different shapes and patterns. It is made entirely with textile materials and is washable. It was built following a tutorial on instructables (40).

Buttons are excellent for the prototyping level, due to their simple circuitry, which makes it simple to learn and become familiarized with conductive materials and the basic notions of electronics. In this case, the microcontroller is only giving power, which was a good way to first became comfortable with circuitry, and afterwards move on to a solution using an arduino and processing language syntax.

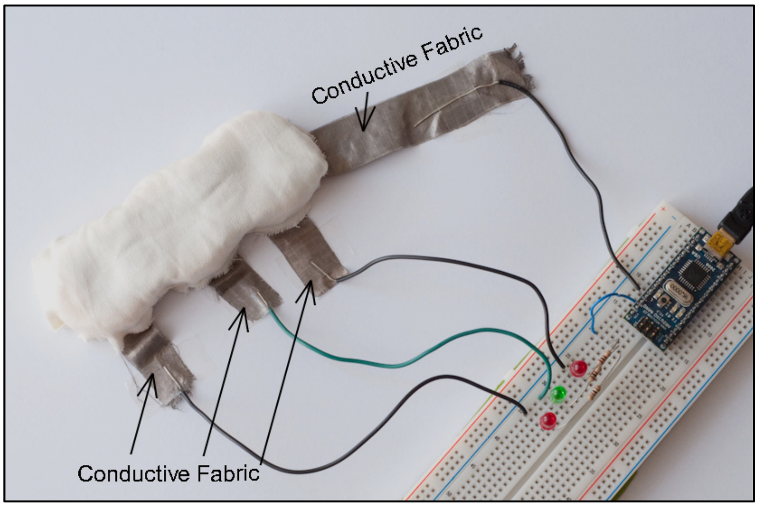

Figure 40 – Three fabric button sensors

1.1. Development

In order to be able to develop this prototype, all materials and tools needed were gathered, and then several steps were taken to build it.

First, a pattern was made on paper in the shape of the sensor. Two pieces of non-conductive fabric and one piece of foam were cut according to the sensor shape. On the top part of the fabric button sensor a strip of conductive fabric was glued with fusible fabric. Fusible fabric is a kind of fabric which has glue on one side and not on the other. While passing the iron on the non-glue side, the glue melts and it is glued to other fabrics. So, conductive fabric was attached to the non-conductive textile putting the fusible fabric on conductive fabric edges and passing the iron. In this way, the conductive fabric did not need to be sewn to the non-conductive fabric, using the fusible fabric instead. On the button part of sensor three different conductive fabrics were also attached (Figure 41 left).

Afterwards, three holes were made in the foam, and the foam was attached between the layers (Figure 41 right) in order to isolate both conductive fabrics, and all this sandwich was sewn around the edges. The three bottom pieces of conductive fabrics were connected to the plus pin of each LED. Consequently, the other pin of LED (negative) was connected to a 220Ω resistor, which was connected to ground. Finally, the top strip of conductive fabric was connected to 5V (Figure 42).

This sensor works as a normal button sensor. When the sensor is pressed, the top and bottom conductive fabrics of the sensor get in touch and current is conducted turning on the respective LED. In other words, the circuit gets closed and an LED shines.

For a better understanding of how the circuitry works, a schematic is presented at Figure 43.

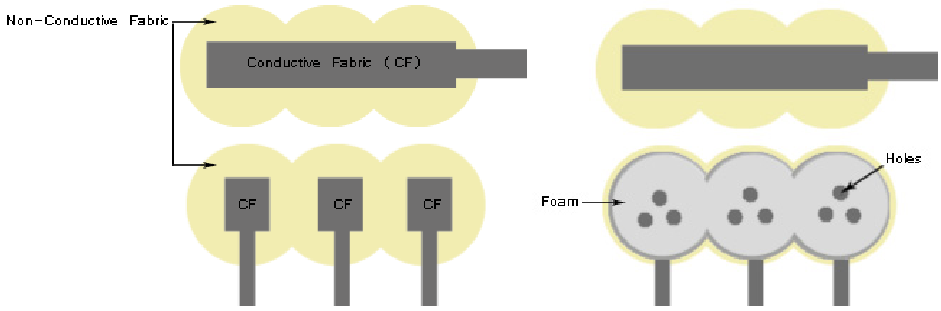

Figure 41 – Illustration of button sensor production

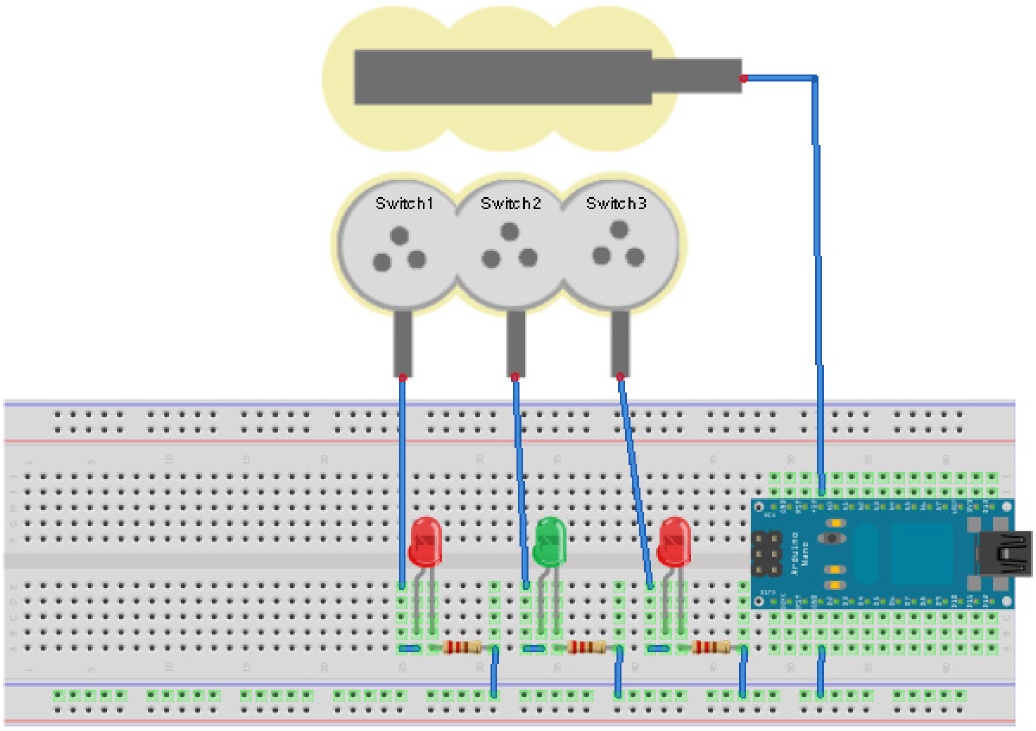

Figure 42 – Illustration of button sensor circuitry, made on Fritzing

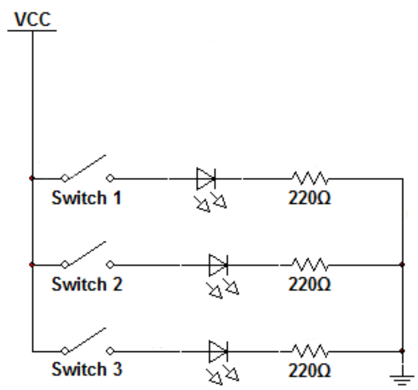

Figure 43 - Button sensor circuitry schematic

1.2. Materials and Tools

Conductive fabric, non-conductive fabric, 1.5 cm thick foam, fusible fabric, regular thread, black wire, three 220Ω resistors, and one green and two red LEDs were used to build the button sensor, as well as tools such as pen and paper, fabric scissors, iron, sewing needle and a multimeter.

1.3. Conclusion

Despite this sensor being a button sensor, it has interesting properties. It is flexible, which means while wearing it, the wearer does not feel a rigid square on his body and also, its size and pattern can be personalized for the best fit on clothes. It is also very accurate and it has the same performance as a rigid button.

2. Stroke Sensor

This prototype is a sensor which reacts to stroking. It is totally handmade (Figure 44). Building it required a lot of sewing. It can be personalized for each shape and color desired. Besides being handmade, it is built entirely from textile materials, which makes it washable.

Like the button sensor, the stroke sensor was built in order to acquire knowledge and dexterity with wearable technologies, more precisely with conductive thread and microcontroller arduino. In this prototype the arduino is also only giving power. This sensor was built following a tutorial on the instructables website (41).

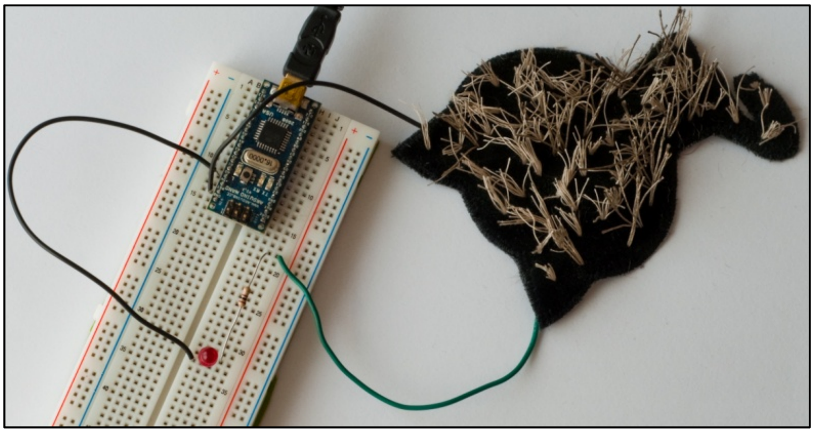

Figure 44 – Stroke Sensor

2.1. Development

A cat pattern was sketched on paper so that a fluffy cat fabric stroke sensor could be made. A smooth velvet fabric was cut according to this pattern. Smooth velvet fabric was used because it is smoother to the touch.

Non-conductive thread was sewn on the fabric cat pattern to imitate the cat fur. Conductive thread was also sewed mixed with non-conductive thread (Figure 45 right, blue thread). The way it works is that when the cat is being petted, the circuit closes and voltage passes through one side of the conductive thread to the other. One side of the conductive thread is connected to the ground and other side is connected to a 220Ω resistor, and then to the LED negative pin. Consequently, the other pin of the LED is connected to 5V (Figure 45 left). When, the cat stroke sensor is petted conductive thread from one side of the row touches conductive thread from other side, closing the circuitry, and the current flows, triggering the LED.

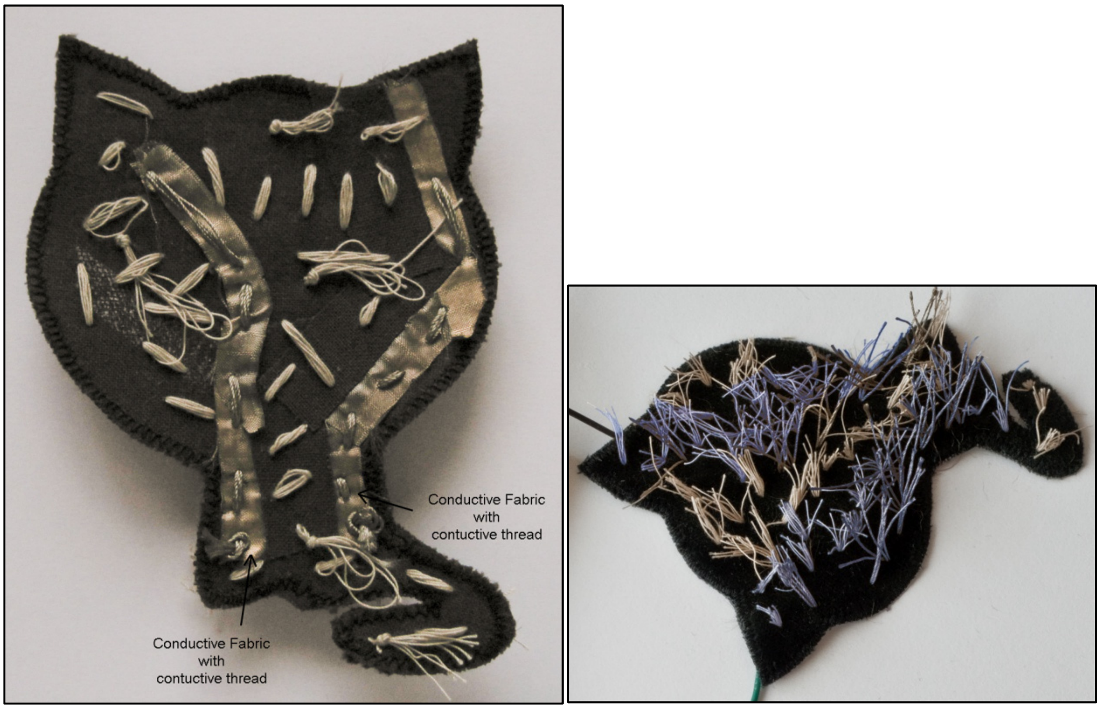

Figure 45 – Stroke Sensor, conductive thread

2.2. Material and Tools

Conductive fabric, non-conductive fabric, fusible interfacing, conductive thread 117/17 2ply, regular thread, black wire, one 220Ω resistor, and one red LED were used to build the stroke sensor, as well as tools such as pen and paper, fabric scissors, iron, sewing needle and sewing machine.

2.3. Conclusion

This sensor is very expansible. It can be personalized wherever wanted and used in a variety of applications, for instance, children's toys, books and clothes.

3. Zip Sensor

This prototype was an attempt to make an interactive zip. The idea behind the zip sensor was to use a gadget that is very common on clothing, like trousers, jackets, shirts, pullovers and shoes, and make it intelligent.

It passed through three upgrades. Firstly, it was developed as a digital sensor. Secondly, the zip was developed as an analog sensor, which is more suitable for this object. It passed through several upgrades. At this stage the arduino was already programmed to capture data, in order to be able to analyze the performance of the zip. The upgrades that the zip passed through are described in the following sections.

3.1. Digital Zip Sensor

This is the first version of the zip sensor (Figure 46). It works like a digital sensor, when it reaches the bottom an LED is turned on, otherwise it is off. It cannot be washed because of the wires.

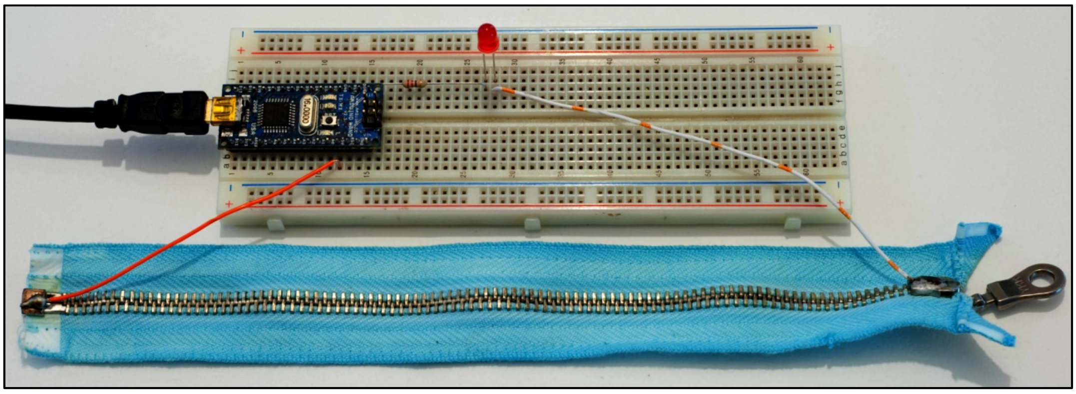

Figure 46 – Digital Zip Sensor

3.1.1. Development

Different wires were soldered to the bottom stop of the zip and on the zip slider. At the bottom stop one wire was soldered, which was connected to 5V. Another wire was soldered to the zip slider and connected to the positive pin of the LED. Then the negative pin of the LED was connected to a 220Ω resistor and connected to the ground (Figure 47). When the solder of bottom stop touches the solder of the zip slider, the LED is turned on.

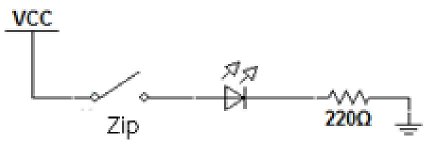

Figure 47 – Digital Zip Sensor circuit diagram

3.1.2. Material and Tools

A zip, one red LED, wire, solder, and a 220Ω resistor were used to build the digital zip sensor digital, as well as tools such as a soldering iron and a multimeter.

3.2. Analog Zip Sensor

The way that the previous zip (section 3.1.) was developed was not satisfactory for a zip. A zip works as a linear continuous sensor, like when we are opening or closing a zip we pass through different stages. However, the digital zip sensor considered that a zip has only two states, totally open and not totally open. When it was totally open, the LED was turned on, otherwise the LED was turned off. This approach of considering only two states is not true, because while opening and closing a zip, we pass through several states, which consist of continuously linear data.

Research was done, and a tutorial was found indicating how to build an analog zip sensor (42). So, this new version of the zip (Figure 48) was built based on this tutorial. It functions as a potentiometer, which is a variable resistor. At this step the arduino was already collecting data from an analog port.

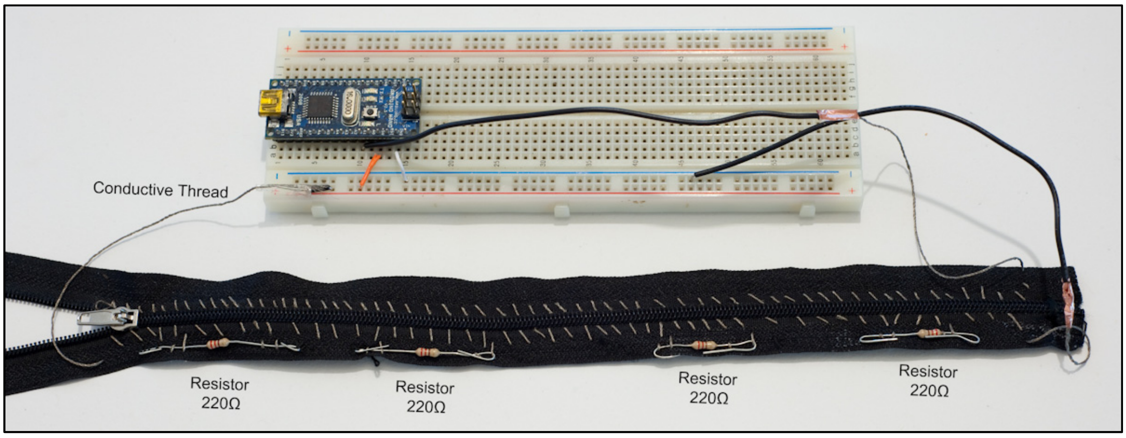

Figure 48 – Analog Zip with resistors

3.2.1. Development

The conductive thread and four 220Ω resistors were sewn by hand to one side of the zip tape (Figure 49). The conductive thread was sewn as close as possible to the elements of the zip. The top tip line was connected to 5V and the bottom tip line was connected to the ground.

On the other side of the zip tape another conductive thread was sewn in, where the bottom tip of the zip was connected to the analog port 1 of the arduino. This port was measuring the variation of voltage.

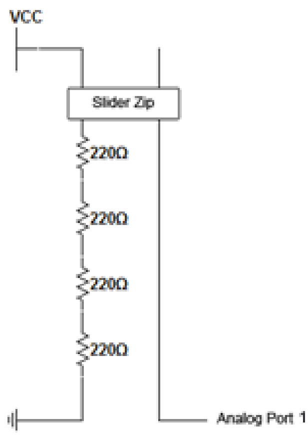

Figure 49 – Analog Zip Scheme

3.2.2. Material and Tools

A zip, four 220Ω resistors, wire, conductive tape and conductive thread 117/17 2ply were used to build the analog zip sensor, as well as tools such as a needle, fabric scissors and a multimeter.

3.3. Performance

After building this sensor, tests of its performance were done. The task performed was to open and close the zip, measuring the voltage at the analog ports every 50ms. Three data samples were collected, performing the same task, in order to obtain more accurate results. The following charts are the data samples. Each chart represents the voltage while opening and closing the zip, in a range between 0 to 1023, where 0 is 0V and 1023 is 5V, because this is the range over which the analog port reads the voltage. The other axis of the chart is the time in seconds.

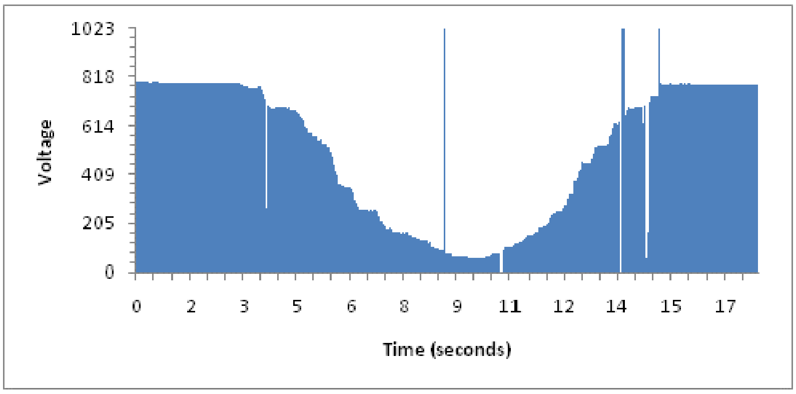

Figure 50 – Analog zip with resistors attached, while opening and closing the zip; first data sample

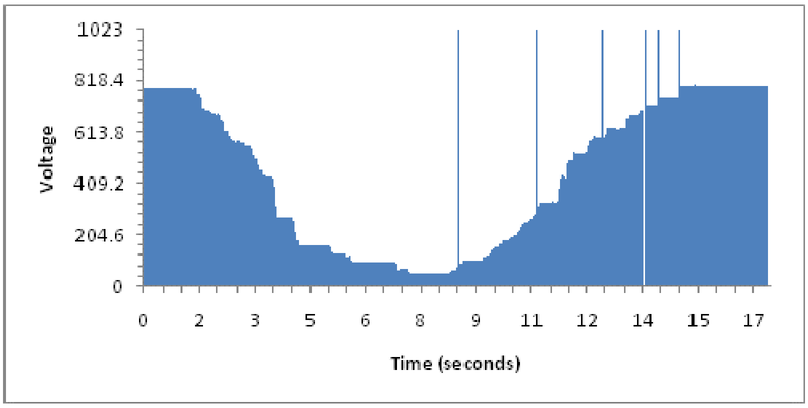

Figure 51 - Analog zip with resistors attached, while opening and closing the zip; second data sample

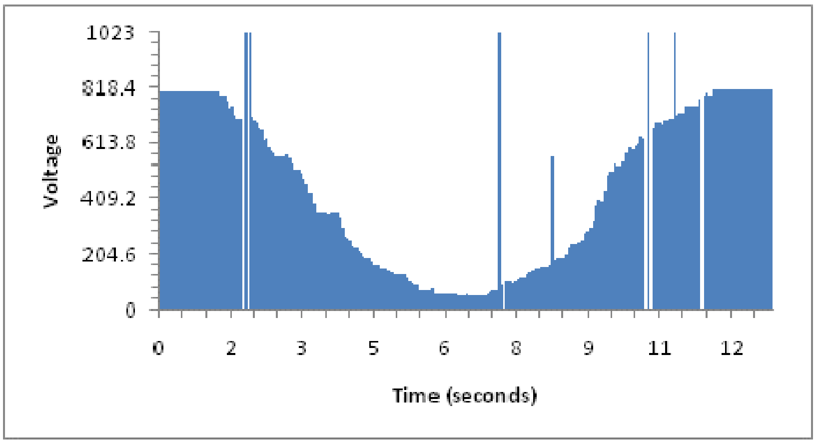

Figure 52 - Analog zip with resistors attached, while opening and closing the zip; third data sample

As can be seen from the data above, this sensor is inaccurate. There are values quickly changing from 0 to 1023 while opening and closing the zip, when it should not happen. If this sensor were used to control a sound volume of a song, while turning up the music, the music volume would be unstable. This instability happens because the slider loses contact with the conductive thread. This can be solved with a software filter; also, doing a rolling average can make voltage values smoother. Besides software stabilization, hardware stabilization can also be done. Using a resistor connected from the analog port to ground would make the sensor read the value 0 instead of making extreme changes between 0 and 1023. However, by using software, the 0 and the 1023 values can be filtered. So, a software filter and a rolling average were implemented. The next charts are the data collected using this software stabilization.

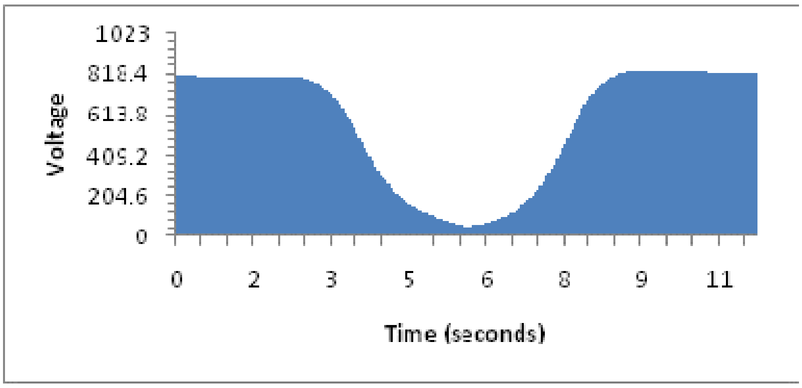

Figure 53 – Analog zip with resistors attached, with software filter and rolling average; first data sample

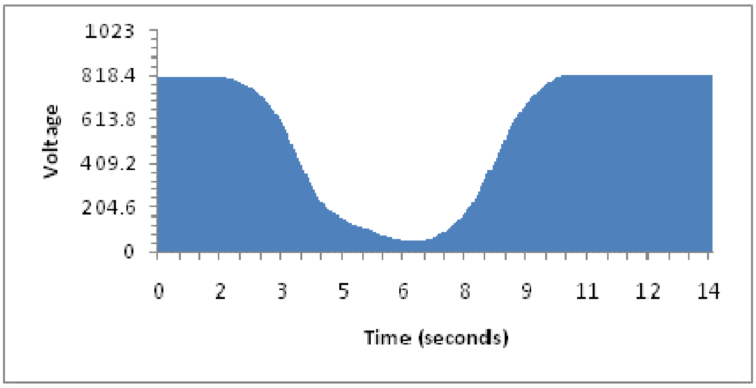

Figure 54 - Analog zip with resistors attached, with software filter and rolling average; second data sample

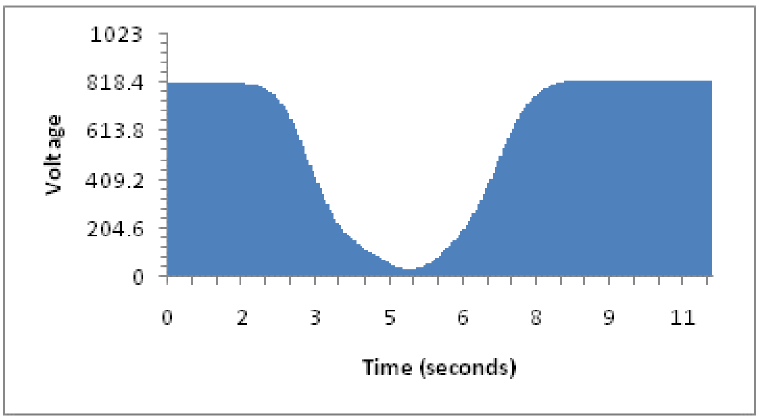

Figure 55 - Analog zip with resistors attached, with software filter and rolling average; third data sample

Using rolling average and a filter software, the sensor is stable. An average of 20 values had to be made to get this smoothness in the values.

3.4. Improved Analog Zip Sensor

This version was made by us, in order to build a more stable and simple zip (Figure 56). This zip was made with conductive thread only. The conductive thread works like a potentiometer. In order to limit the current in the circuit a 820Ω resistor was attached so that no one component was damaged because of high current.

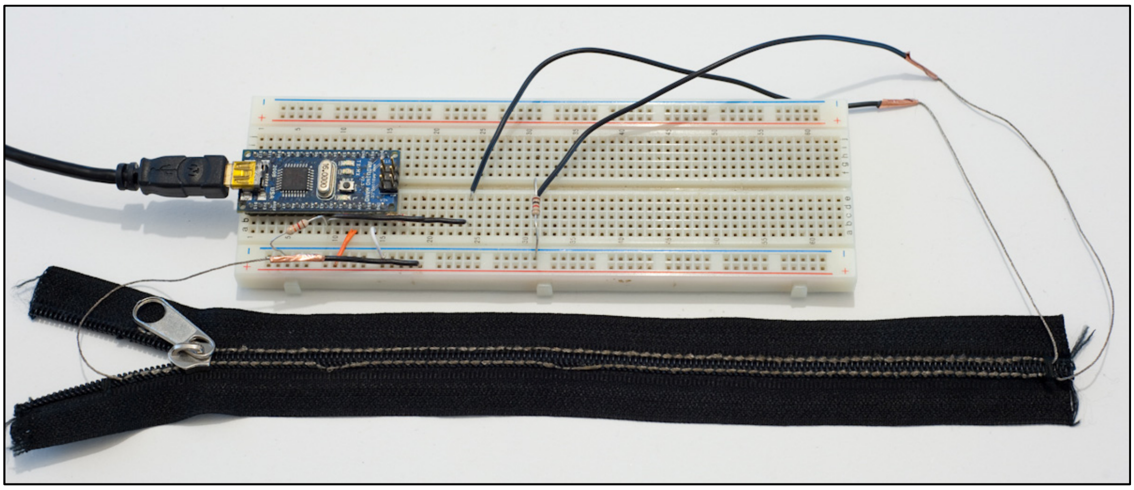

Figure 56 – Analog Zip with conductive thread

3.4.1. Development

On one side of the tape, conductive thread was sewed using a sewing machine. The top side was connected to 5V and the bottom side to a 820Ω resistor, and then connected to ground. On the other side of the tape, another conductive thread was sewed, and the bottom part was connected to the analog port A2 (Figure 57). A 12kΩ resistor was also connected in parallel at the analog port 2, in order to stabilize the circuitry, this resistor being optional. The 820Ω resistor was attached to the circuitry to reduce the current, since the conductive thread has a small resistance, if no resistor was included, the circuitry could be damaged.

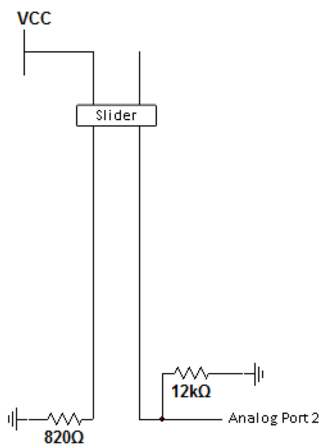

Figure 57 – Analog Zip Improved Scheme

3.4.2. Material and Tools

One 820Ω and one 12kΩ resistor, and conductive thread 117/17 2-ply were used to build the new version of the analog zip sensor, as well as tools such as sewing machine, fabric scissors and a multimeter.

3.5. Performance

After building this sensor, tests of its performance were done. The task performed was the same as that for the previous zip sensor. The zip was opened and closed, and the voltage was measured at the analog port every 50ms. In order to have the same degree of equality between this test and the one made with the zip sensor with resistors, the 12kΩ which stabilizes the sensor was removed. So, these tests were done without any hardware stabilization. Three samples were also collected, which are presented below.

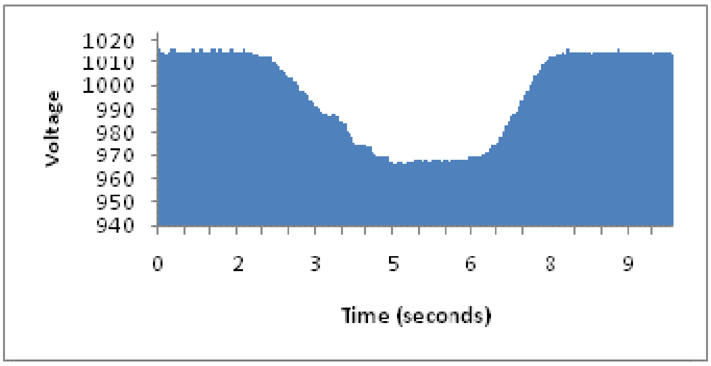

Figure 58 – Analog zip with conductive thread only, while opening and closing the zip; first data sample

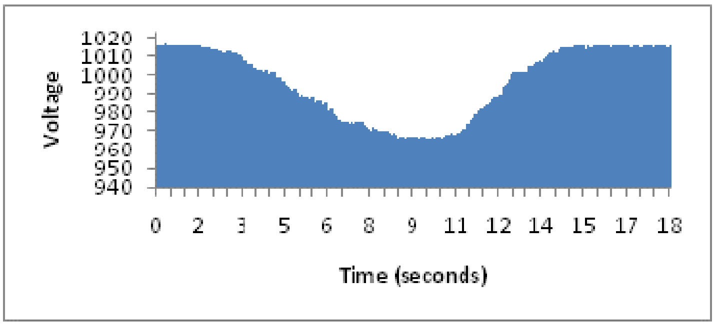

Figure 59 - Analog zip with conductive thread only, while opening and closing the zip; second data sample

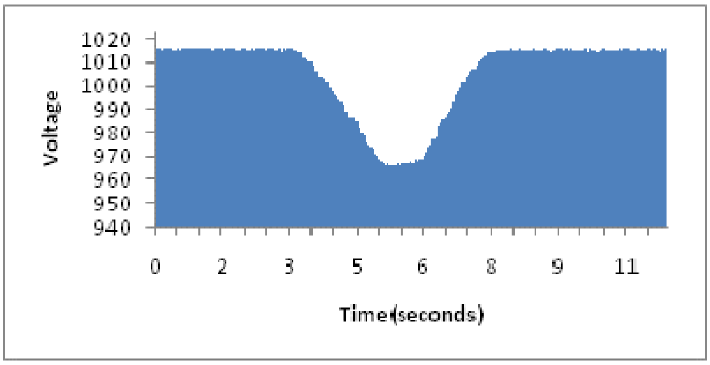

Figure 60 - Analog zip with conductive thread only, while opening and closing the zip; third data sample

This sensor is accurate. There are no extreme changes in the data. This sensor is at another scale, ranging only from 940 to 1023, because this is a small potentiometer, hence, the smaller range. Comparing these charts with the analog zip with resistors without software stabilization this sensor is more stable. However, in applying this sensor to control the sound volume there would be some disturbance. So, the next charts show the application of a rolling average, to get smoother values.

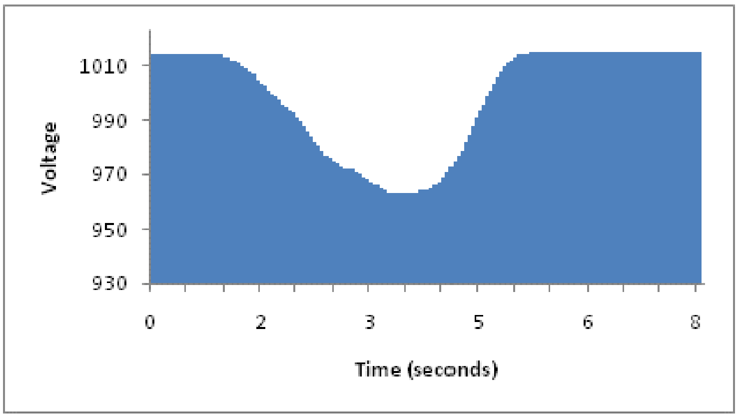

Figure 61 - Analog zip only with conductive thread only, with software rolling average; first data sample

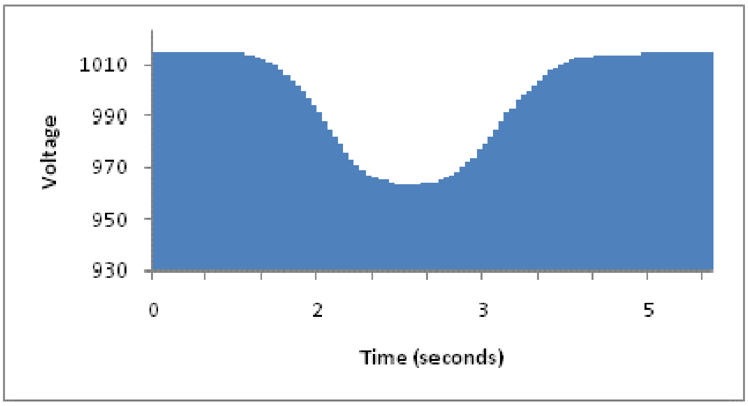

Figure 62 - Analog zip only with conductive thread only, with software rolling average; second data sample

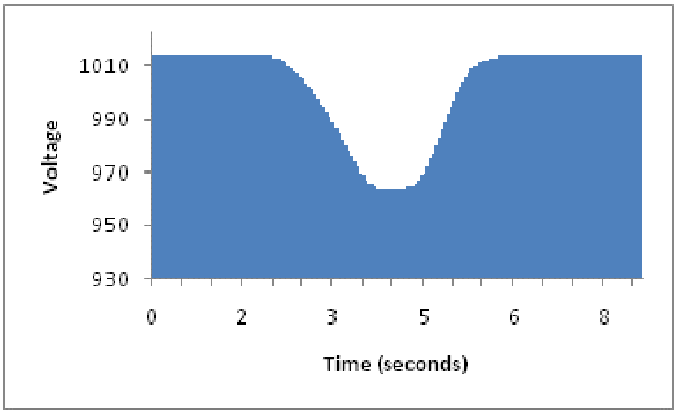

Figure 63 - Analog zip only with conductive thread only, with software rolling average; third data sample

Using rolling average data produced greater stability. A rolling average of 10 values was used, half the values used on the other zip. This sensor can be attached to clothes and is washable.

3.6. Conclusion

Both zips are accurate when a software stabilizer is applied. However, the first zip cannot be washed and attached to clothes because of the resistors. The analog zip with conductive thread can be attached to clothes and washed.

4. Stretch Sensor

The idea behind the Stretch Sensor was to use it as a multi-data sensor. It was thought that this could be an interesting way of identifying where the sensor is being stretched and grabbed, for instance, for selecting menus. Two versions of this sensor were developed, and are presented in the following sections. But, firstly the principle behind them will be presented.

4.1. Tension Voltage Divider

The voltage divider is an electronic principle, which says that measuring the voltage of a linear circuitry at different points will measure different voltages.

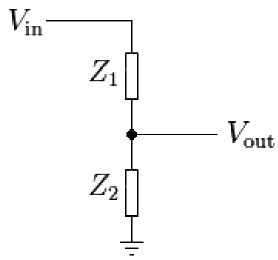

Figure 64 – Voltage Divider

In Figure 64, V_{in} is the input voltage, which in our case is 5V, and V_{out} is the position of the linear circuitry where the tension will be measured. Z_1 and Z_2 are the resistors. The equation to measure the voltage at V_{out} applies Ohm's Law, where V is voltage (tension), R is resistance and I is current:

$$V=R\times I$$

Then, applying Ohm's Law, we have:

$V_{in} = (Z_1 + Z_2) \times I$, (equation 1)

$V_{out} = Z_2 \times I$, (equation 2)

These equations 1 and 2 will be solved in order to the variable I:

$$I = \frac{V_{in}}{(Z_1 + Z_2)} \leftrightarrow$$

$$V_{out} = \frac{Z_2}{(Z_1 + Z_2)} \times V_{in}$$

, (equation 3)

So, in order to know what the V_{out} voltage is, just apply equation 3. For instance, if the resistances are the same, Z_1 and Z_2 both 1kΩ, and if the voltage applied at the circuitry is 5V, V_{out} will be 2.5V. So, a voltage divider is directly proportional to the resistor value. In order to use this principle to know where the sensor is being pulled, let us see the following example:

Considering that we have Z_1 and Z_2 which have the same resistance value, which is 1kΩ, supposing when they are stretched their resistance increases 1kΩ, being 2kΩ.

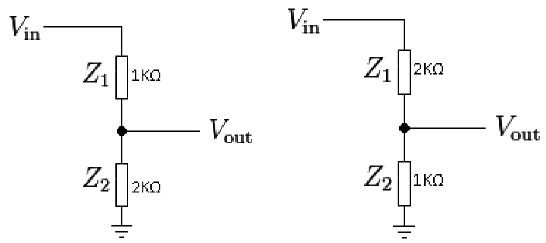

Figure 65 – Stretching resistors Z_2 and Z_1

In this case, in the situation where Z_2 is stretched (Figure 65 left), the output tension will be:

$$V_{out} = \frac{2}{(2+1)} \times 5 = 3{,}33V$$

On the other hand, when Z_1 is stretched (Figure 65 right), the output tension will be:

$$V_{out} = \frac{1}{(2+1)} \times 5 = 1{,}67V$$

So, when both resistors are relaxed the output voltage is 2,5V, when the one that is connected to ground is stretched the output voltage increases, and when the one that is connected to power is stretched the output voltage increases. In this way it is possible to know if it is the first or second position that is being stretched.

4.2. Single Stretch Sensor

Based on the tension voltage principle, a prototype was developed to identify the positions where it is being pulled at different parts of the stretch sensor. Three output points were added to read the voltage at three different points, V_{out1}, V_{out2}, V_{out3} (Figure 66).

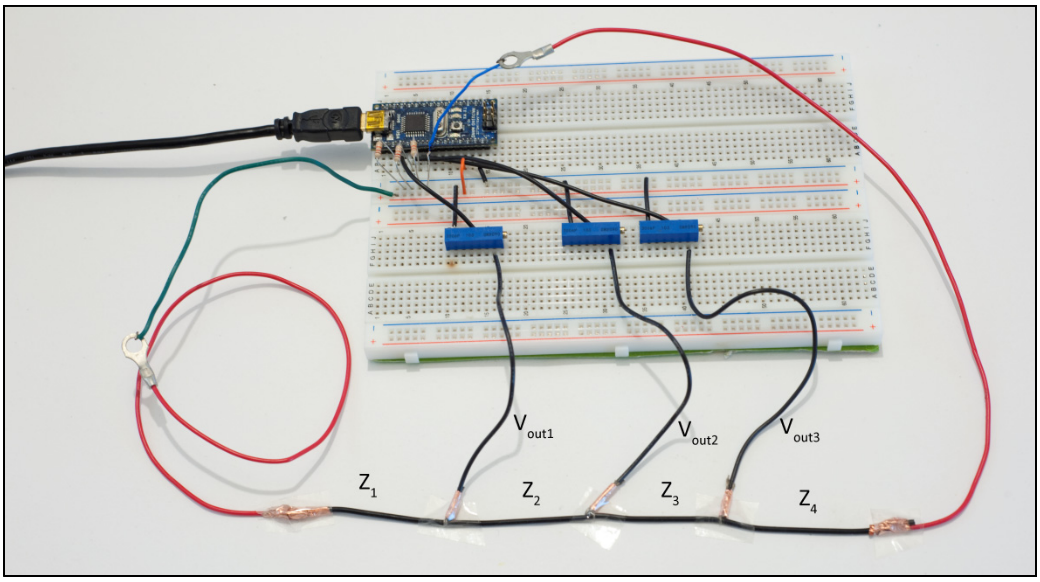

Figure 66 – Single Stretch Sensor

At this case we have the circuitry, which is shown at Figure 67.

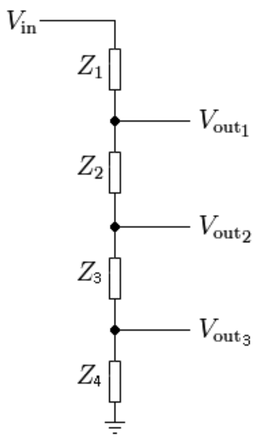

Figure 67 – Stretch Sensor Voltage Divider, resistors and output voltage

Where, Z_1 plus Z_2 and Z_3 and Z_4 is the total sensor resistance, 10kΩ. The voltage at each of the points is:

$$V_{out1} = \frac{Z_2 + Z_3 + Z_4}{(Z_1 + Z_2 + Z_3 + Z_4)} \times V_{in}$$

$$V_{out2} = \frac{Z_3 + Z_4}{(Z_1 + Z_2 + Z_3 + Z_4)} \times V_{in}$$

$$V_{out3} = \frac{Z_4}{(Z_1 + Z_2 + Z_3 + Z_4)} \times V_{in}$$

Now supposing that all four resistors are 2kΩ resistors, and when each one of them is grabbed by both hands and stretched, their resistances increase 1kΩ, being 3kΩ. So, the following table presents the results for each voltage output.

Table 1 – Result of voltage divider equations

	Z_1		Z_2		Z_3		Z_4	
	normal	stretched	normal	stretched	normal	stretched	normal	stretched
V_{out1}	3	2,73	3	3,18	3	3,18	3	3,18
V_{out2}	2	1,82	2	1,82	2	2,27	2	2,27
V_{out3}	1	0,45	1	0,45	1	0,45	1	1,36

Analyzing this table it is possible to make the following conclusions:

- Z_1 is stretched, all decrease their voltage;
- Z_2 is stretched, V_{out1} increases its voltage, and others decrease;
- Z_3 is stretched, V_{out1} and V_{out2} increases their voltage, and V_{out3} decreases;
- Z_4 is stretched, all increase their voltage.

4.2.1. Development

A stretch sensor of 24cm, with approximately 10kΩ resistance was used. It was divided into four equal parts and three conductive threads were connected to the analog ports to measure the output voltage (Figure 68). In this way there were three analog ports (A1, A2, and A3) reading the voltage at three different points of the stretch sensor. In order to stabilize the values read at the analog ports 12kΩ resistors were connected from the analog port to the ground. Also, three potentiometers were attached in order to stabilize the values at the analog ports.

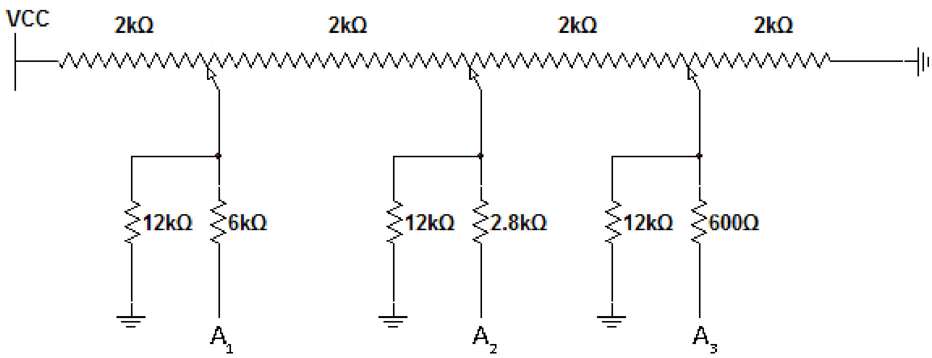

Figure 68 – Single Stretch Sensor: Voltage Divider Scheme

4.2.2. Material and Tools

A 24cm(10") stretch sensor, three 10kΩ potentiometers, three 12kΩ resistors, wire, conductive tape, conductive thread were used to build this sensor, as well as tools such as a multimeter.

4.2.3. Performance

In order to see how accurate this sensor is, data was collected. An analysis will be also be made to see if the data is consistent with the theory. Firstly, charts regarding the stretching of Z_1 will be presented.

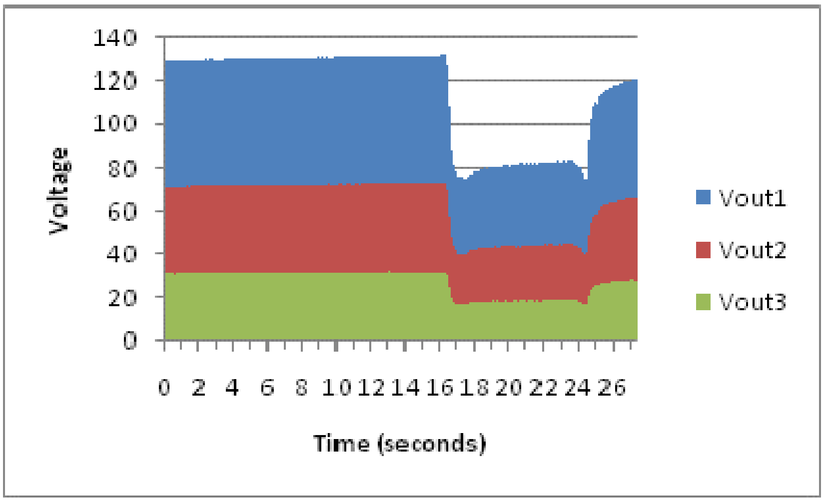

Figure 69 – Stretching and relaxing the first position; first data sample

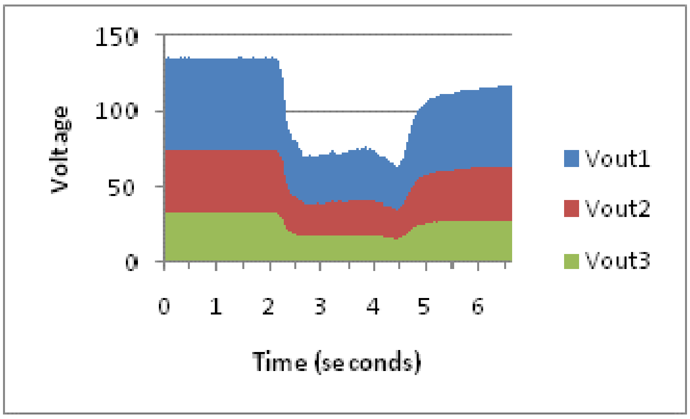

Figure 70 - Stretching and relaxing the first position; second data sample

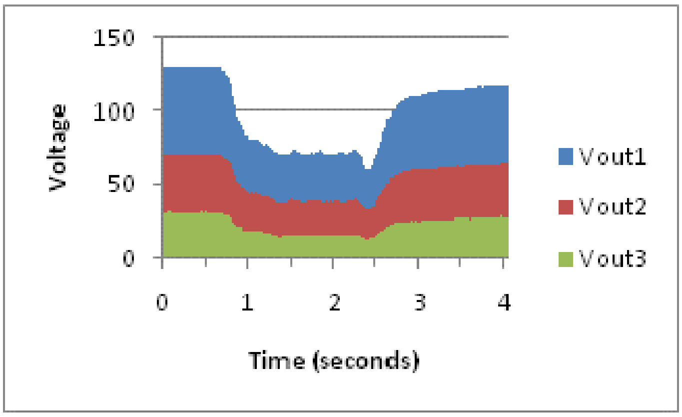

Figure 71 - Stretching and relaxing the first position; third data sample

It can be seen that there is a big change in the values when Z_1 is stretched. It is easy to observe that all output voltages decreased considerably, which is consistent with the theory. The more it is stretched, the greater the decrease. The next step is to present data regarding when Z_2 is stretched.

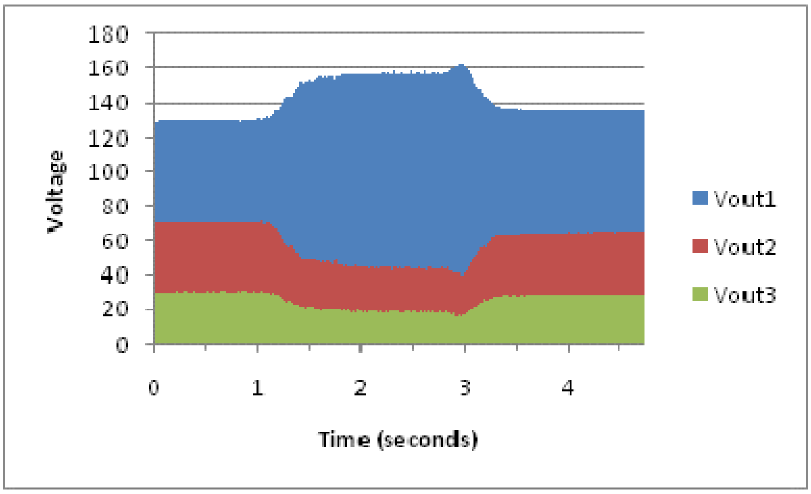

Figure 72 - Stretching and relaxing the second position; first data sample

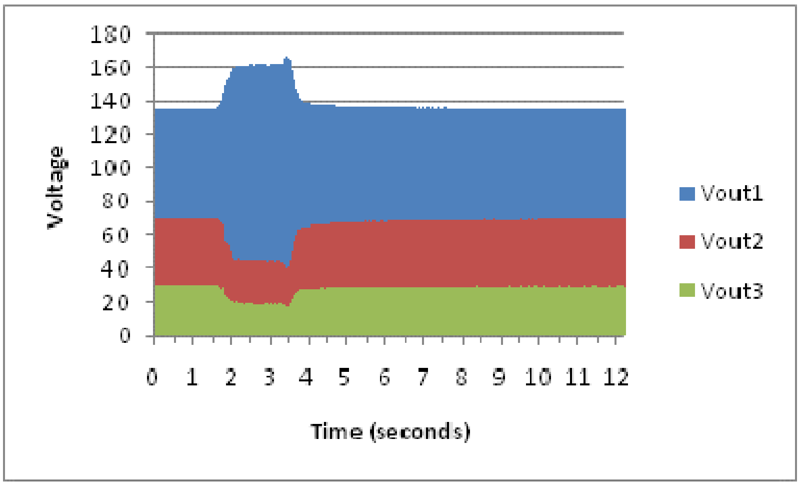

Figure 73 - Stretching and relaxing the second position; second data sample

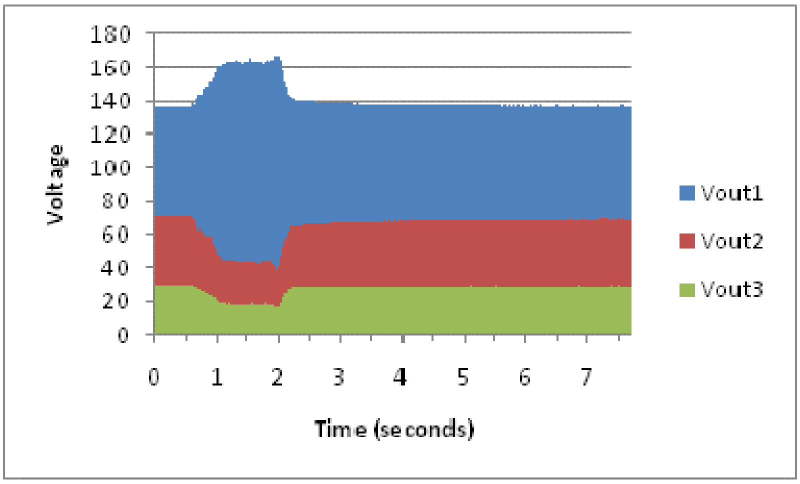

Figure 74 - Stretching and relaxing the second position; third data sample

It is noticeable when the second position was stretched, because of the voltage output values. One of the output voltages increased and the other two decreased, which is also consistent with the theory. In the following charts the data when the Z_3 is stretched will be presented.

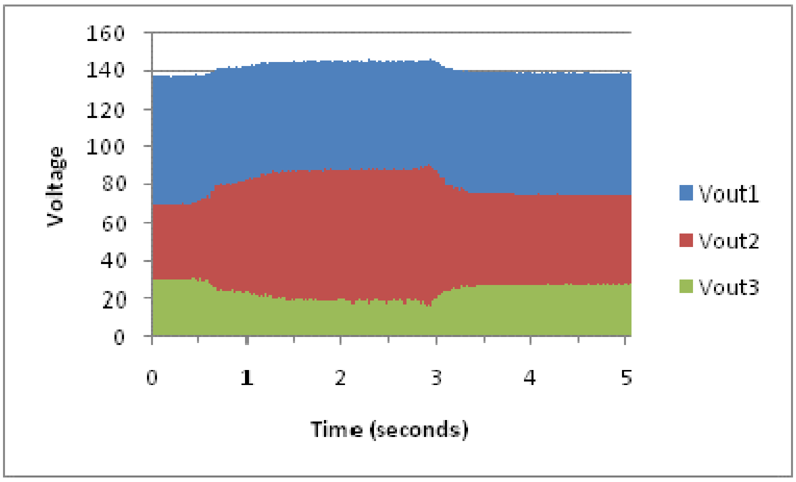

Figure 75 - Stretching and relaxing the third position; first data sample

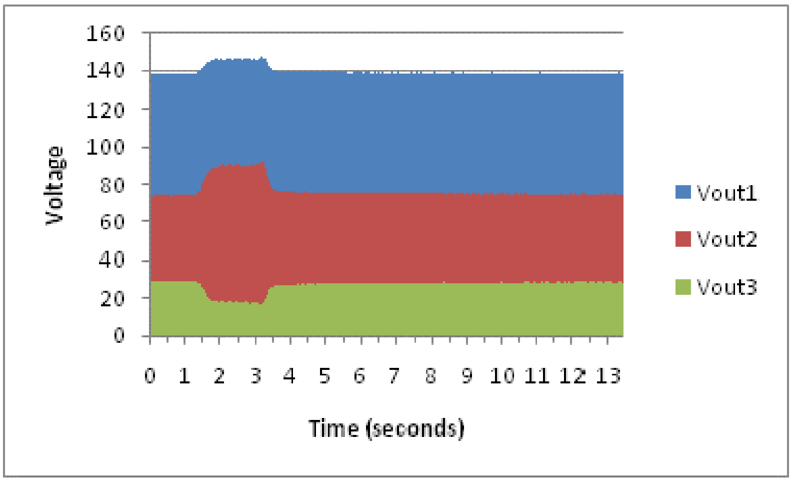

Figure 76 - Stretching and relaxing the third position; second data sample

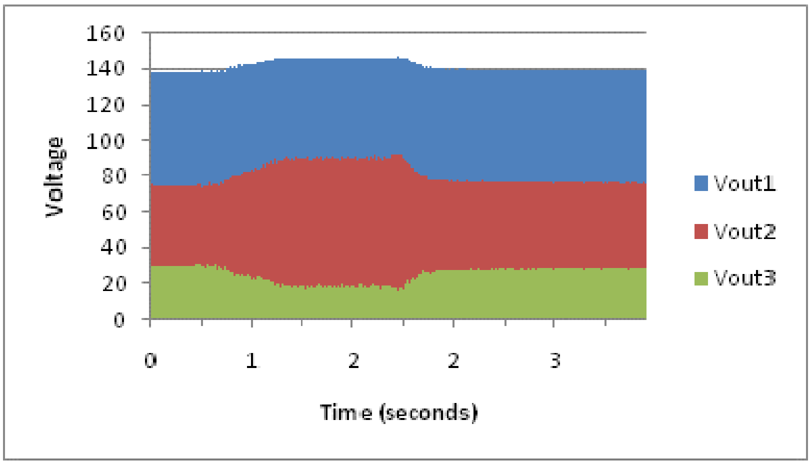

Figure 77 - Stretching and relaxing the third position; third data sample

At this position of stretching, we can see a different pattern in the output voltages. However, there is not a big difference in the output voltage range, when this position is stretched. Both V_{out1} and V_{out2} increase and V_{out3} decreases, which is consistent with the theory. The following charts present the data when Z_4 is stretched.

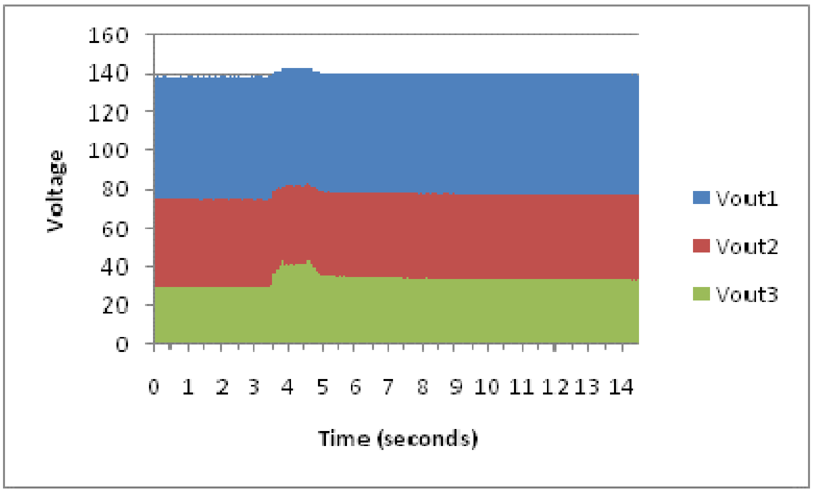

Figure 78 - Stretching and relaxing the fourth position; first data sample

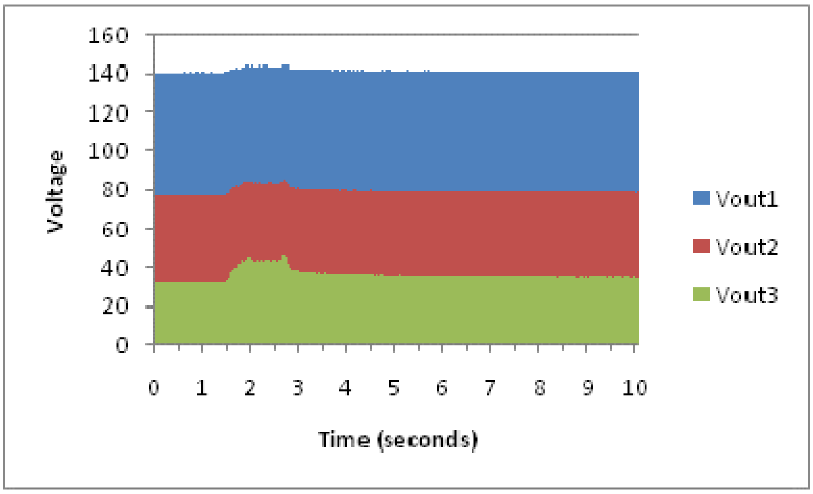

Figure 79 - Stretching and relaxing the fourth position; second data sample

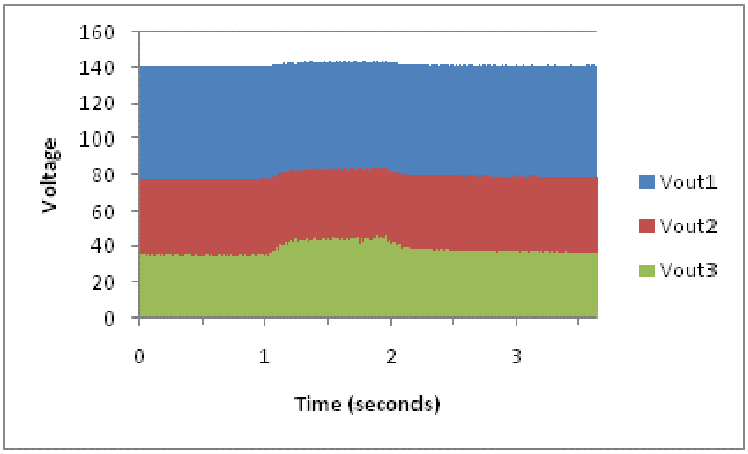

Figure 80 - Stretching and relaxing the fourth position; third data sample

When stretching the sensor at the fourth position, there are very few changes in the voltage output. We notice that all three increased, but the V_{out1} is a very smooth change. These are also consistent with the theory.

4.2.4. Conclusion

Using this sensor, when stretching at the first position all output voltages decreased. When stretching the second position, V_{out1} increased its voltage, and the others decreased. When stretching the third position, V_{out1} and V_{out2} increased their voltage, and V_{out3} decreased. When stretching the fourth position, all of them increased their voltage.

At the first two positions the range change was clear, while being stretched. On the other hand, at the third and fourth positions it was not so clear; besides they all were being stretched the same amount.

The performance of this sensor could be improved by applying a rolling average and making the values less noisy. However, this is not worthwhile, because by attaching more positions, a lot of comparisons must be made to identify which of the positions is being stretched. Also, the last positions have little change, which makes it difficult to identify.

4.3. Divided Stretch Sensor

The stretch sensor was cut into four pieces (Figure 81), which are Z_1 , Z_2 , Z_3 , and Z_4 and then a tension division was created only with the stretch sensor and a fixed resistor. Hence, it is not necessary to verify conditions to see which position is being grabbed and stretched, because each stretch sensor is connected to a different analog port, which measures the output voltage, meaning each stretch sensor is independent from the others. Moreover, it is just one resistance that changes, which is the stretch sensor.

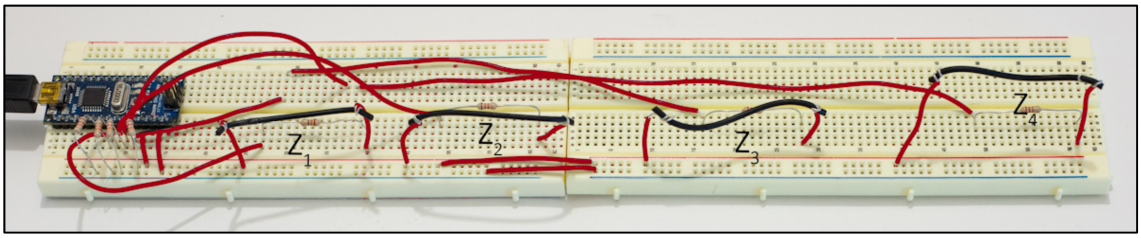

Figure 81 – Stretch Sensor Divided

4.3.1. Development

A 24cm stretch sensor was cut into four parts, and each part was connected to a 220Ω resistor and then connected to the analog arduino port. Also 12kΩ resistors were attached to the analog ports in order to stabilize them and create a division voltage. The scheme at Figure 82 is only for one piece of stretch sensor (Z1). The other pieces are exactly the same, the only difference being that they are connected to analog port A6, A5 and A4 instead of A7.

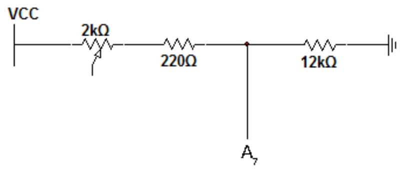

Figure 82 – Stretch Sensor Scheme

4.3.2. Material and Tools

A 24cm(10") stretch sensor, four 12kΩ and four 220Ω resistors, wire, were used to build this sensor, as well as tools such as a multimeter.

4.3.3. Performance

In order to see how accurate this sensor is, three samples of data were collected. The following charts are refer to the stretching of Z_1, Z_2, Z_3, and then Z_4, which are first, second, third and fourth positions, respectively.

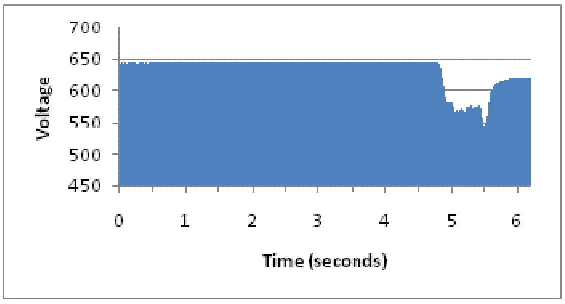

Figure 83 - Stretching the first position; first data sample

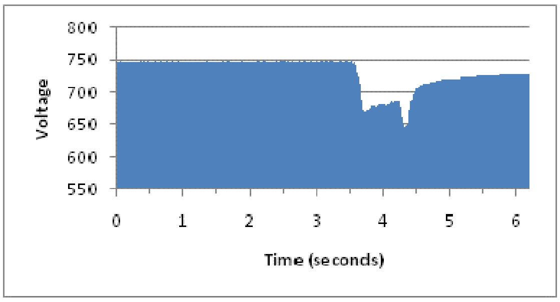

Figure 84 - Stretching the second position; first data sample

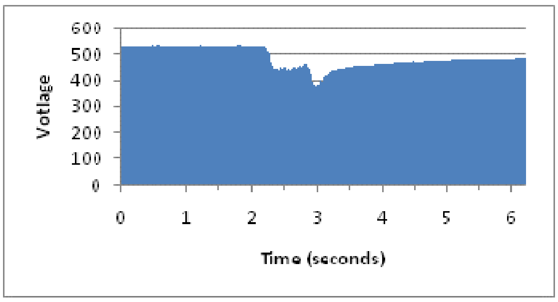

Figure 85 - Stretching the third position; first data sample

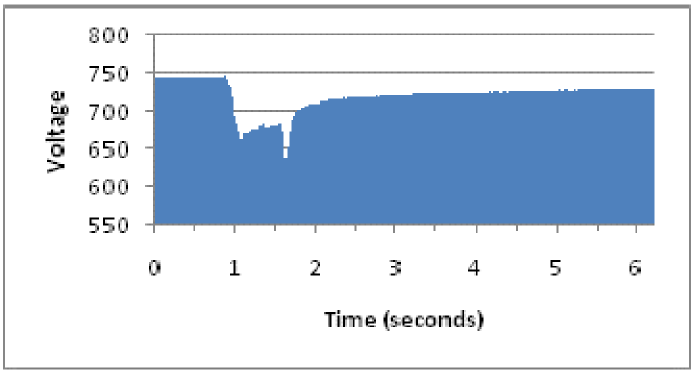

Figure 86 - Stretching the fourth position; first data sample

The following charts are the second data samples collected, while stretching first, second, third and fourth positions.

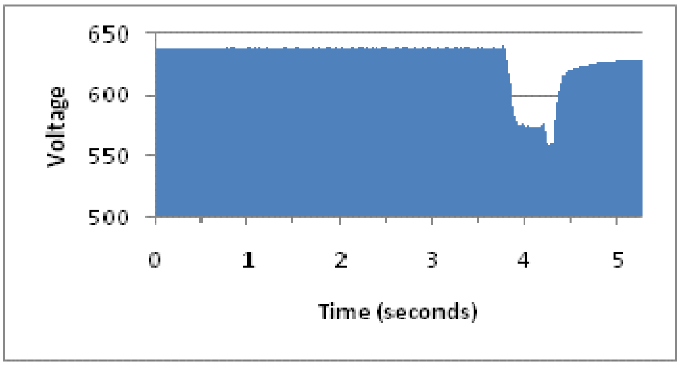

Figure 87 - Stretching the first position; second data sample

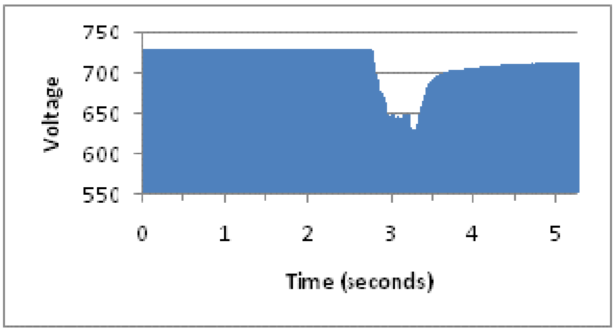

Figure 88 - Stretching the second position; second data sample

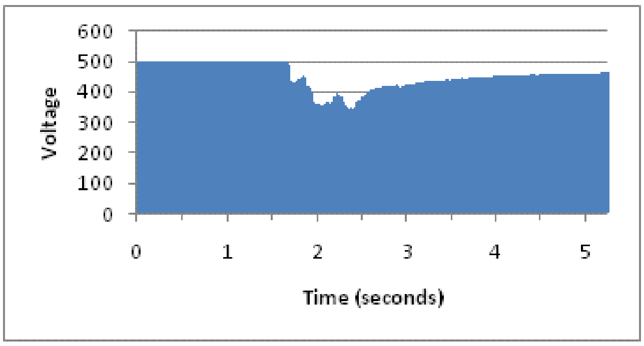

Figure 89 - Stretching the third position; second data sample

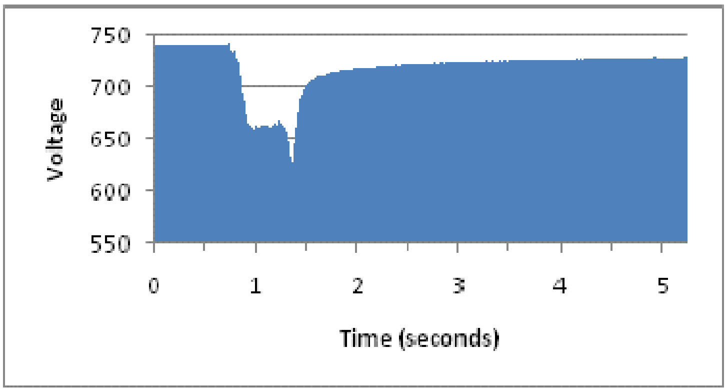

Figure 90 - Stretching the fourth position; second data sample

The following charts refer to the third and last data sample, while stretching first, second, third and fourth positions.

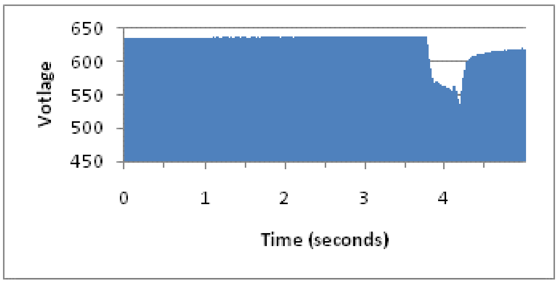

Figure 91 - Stretching the first position; third data sample

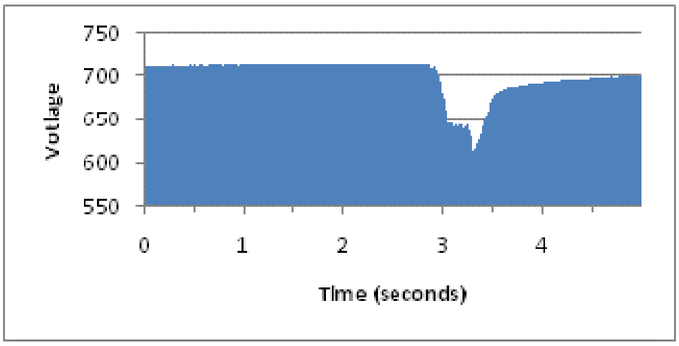

Figure 92 - Stretching the second position; third data sample

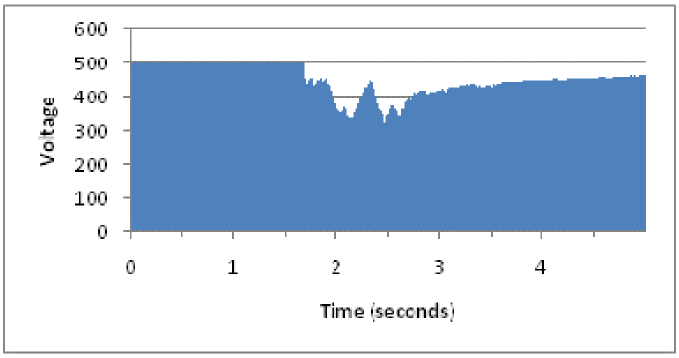

Figure 93 - Stretching the third position; third data sample

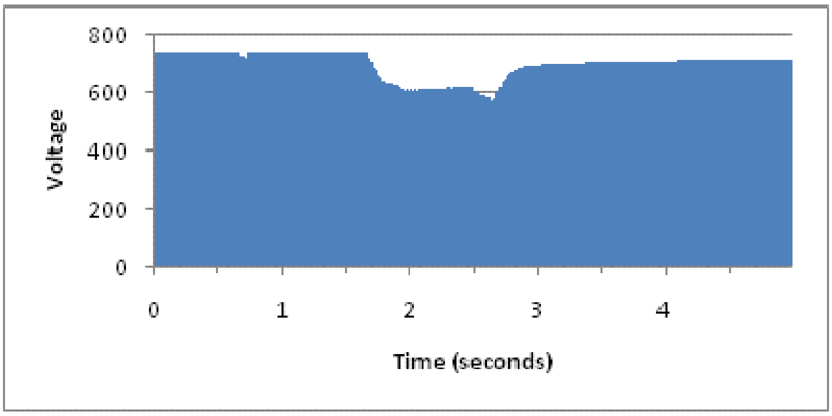

Figure 94 - Stretching the fourth position; third data sample

As can be seen below, it is easy to identify which position was stretched. There is a big change in the range of all positions. The third position is the one that has more disturbances because holes began appearing on the stretch sensor due to its being stretched several times (Figure 95).

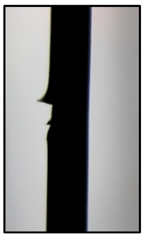

Figure 95 – Stretch sensor damaged, because of stretching

4.3.4. Conclusion

The second version of this sensor was more accurate. It was clear at all positions when it was stretched. Compared to the first version of the sensor, which was not cut, this one just read the values at the analog port and it is not necessary to see in which range it is in order to identify the position. However, using the cut stretch sensor, the use of more wires is required.

in general, the cut stretch sensor is more accurate and it is simpler to identify where it is being grabbed and stretched, and by applying a rolling average its values are smoother.

5. Toggle Sensor

The idea behind the construction of a toggle sensor originated in the same way as the zip. Lots of clothes have beads, and using them as an input in wearables would be an unobtrusive way for interaction. This sensor could be used to detect where the position of the bead was and according to that position have a different output. Several improvements to the prototype were made.

5.1. Version One

Initially the toggle was built connected to analog ports, since using five analog ports, the ranges that did not matter could be filtered. For instance, people usually play with the bead, and while twisting and touching the toggle, if the analog ports were activated that would not be very pleasant. So, analog ports were used in order to filter those values. Only the bead would be detected to activate the outputs. The prototype developed is the one in the following pictures. This sensor was tested with two kinds of fabrics in order to see if the quality of the fabric interferes with the results (Figure 96 and 97).

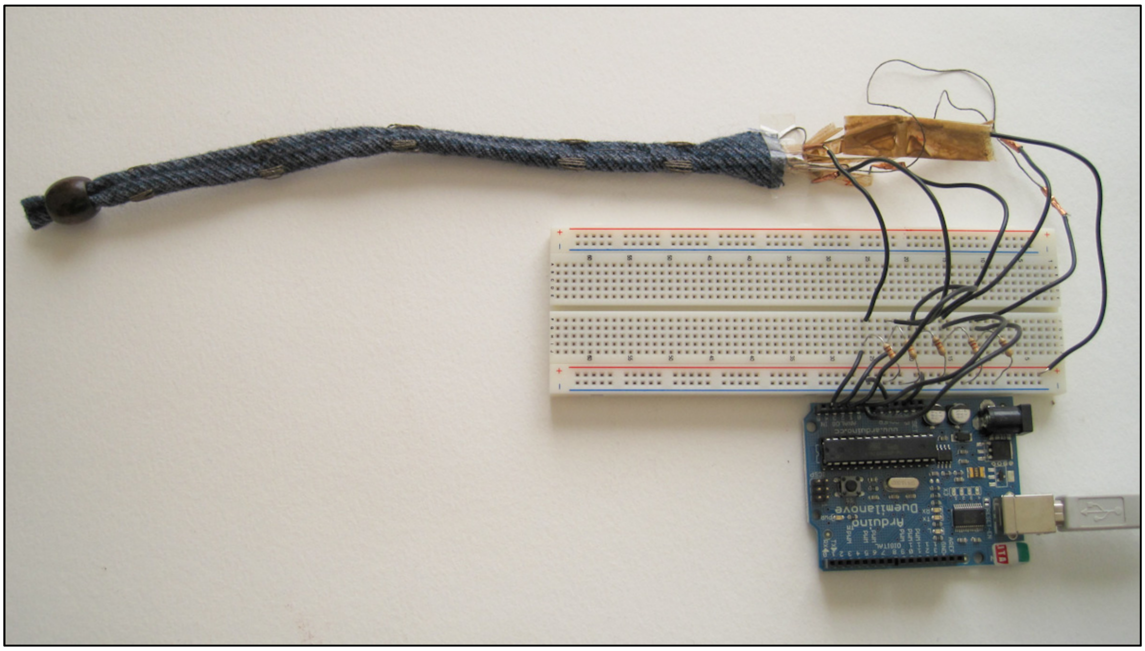

Figure 96 – Toggle sensor using five analog ports, high-quality fabric

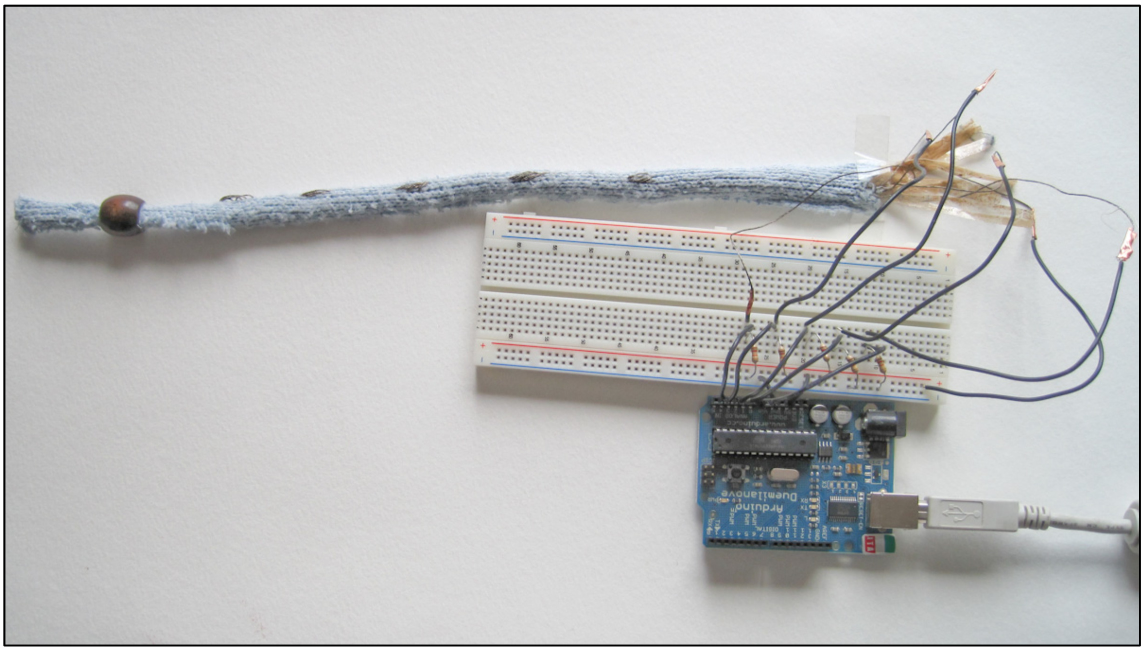

Figure 97 – Toggle sensor using five analog ports, low-quality fabric

5.2. Development

One side of the toggle a conductive thread was sewed, where one side was connected to 5V. Then, at the other side five pieces of conductive thread were sewed and connected to 5 analog ports, from 0 to 4. From the analog ports a 12kΩ resistor was attached, as can be seen at Figure 98.

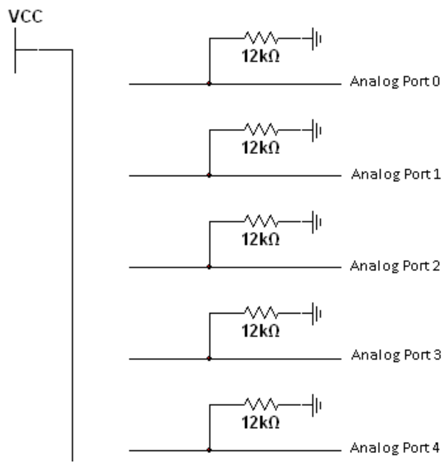

Figure 98 – Toggle sensor using five digital ports schematic

5.3. Material and Tools

Conductive thread 117/17 2ply, fabric, wire, bead and five 12kΩ resistors were used to build this version of the toggle sensor, as well as tools such sewing needle, sewing machine, fabric scissors, and a multimeter.

5.4. Performance

The degree of reliability of this prototype was tested. Three data samples were registered, and are presented bellow, for the high-quality fabric toggle and the low-quality fabric toggle. Firstly, the low-quality fabric, which frays, was tested. The following charts refer to it.

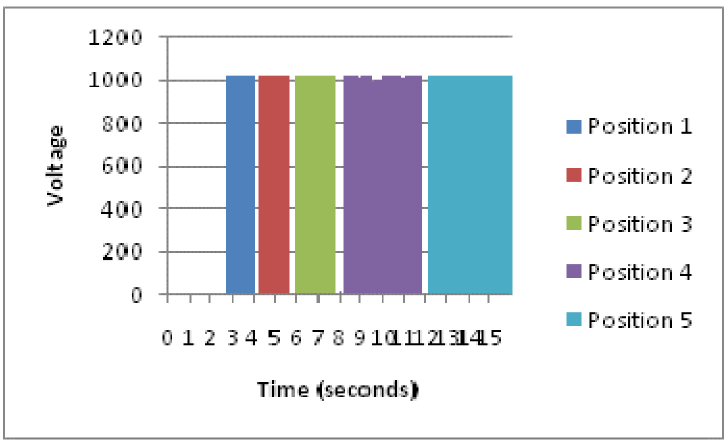

Figure 99 – Toggle sensor using five analog ports – low-quality fabric, measures 1022 when activated; first data sample

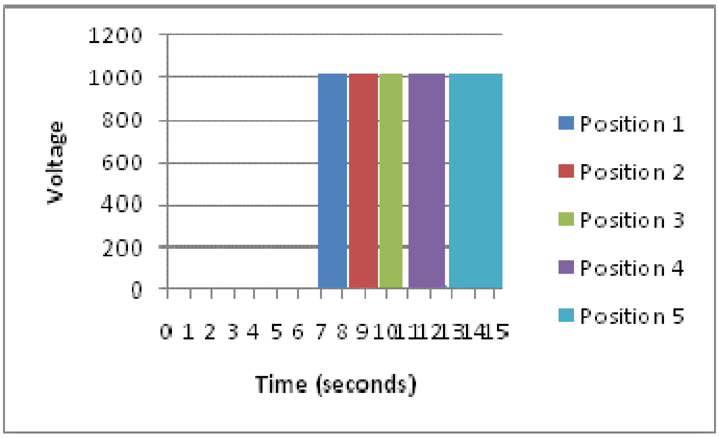

Figure 100 - Toggle sensor using five analog ports – low-quality fabric, measures 1022 when activated; second data sample

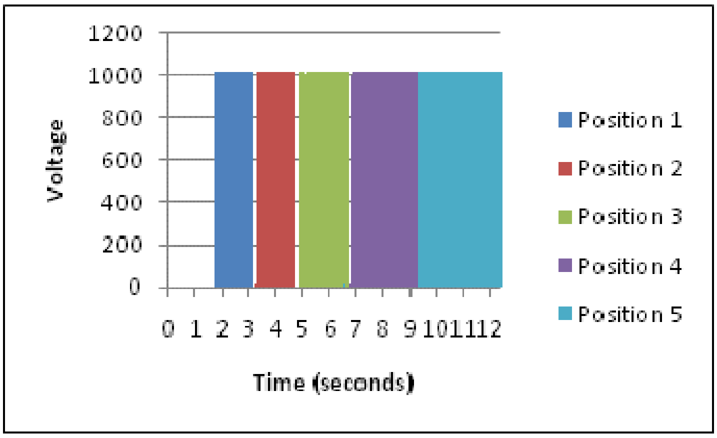

Figure 101 - Toggle sensor using five analog ports – low-quality fabric, measures 1022 when activated; third data sample

Secondly, the high-quality fabric, which does not fray, was tested. The following data is for high-quality fabric toggle.

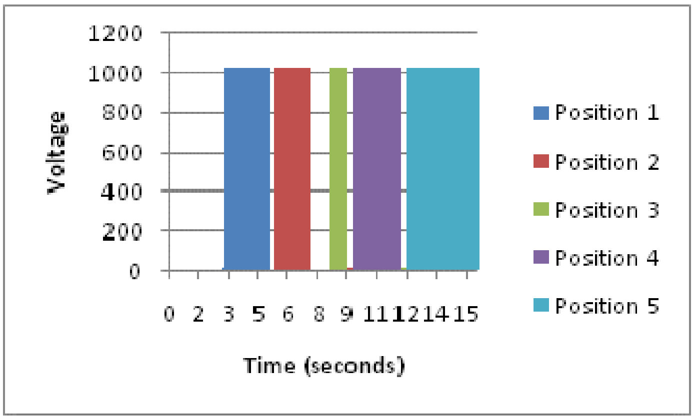

Figure 102 - Toggle sensor using five analog ports – high-quality fabric, measures 1022 when activated; first data sample

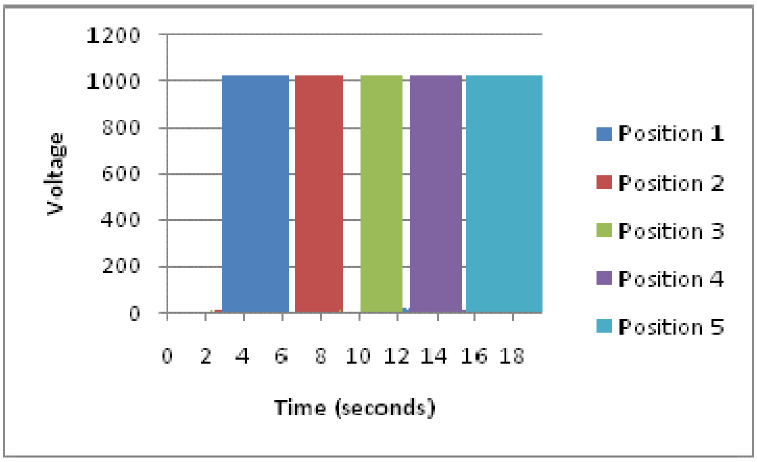

Figure 103 - Toggle sensor using five analog ports – high-quality fabric, measures 1022 when activated; second data sample

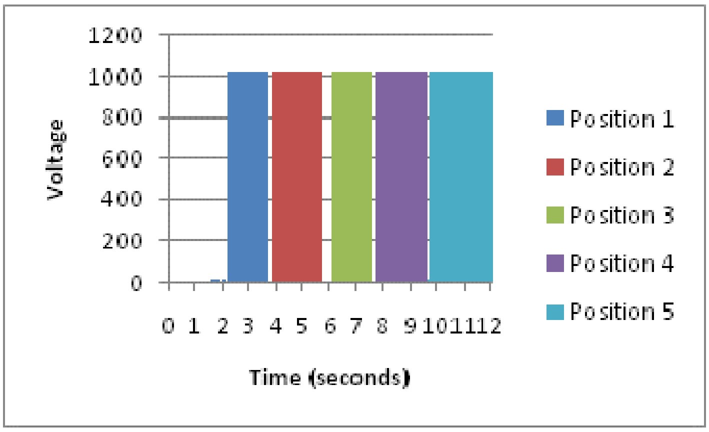

Figure 104 - Toggle sensor using five analog ports – high-quality fabric, measures 1022 when activated; third data sample

As can be seen, both toggles are accurate, using a low or high-quality fabric. Even though the low-quality frays, it does not make a big change in the values. It is reliable. Also, the positions were grabbed with the hands and did not interfere with the values; they can be filtered. However this prototype has a problem, which is using too many ports. Each position has to be connected to a port, which is not feasible. Thus a new version arose.

5.5. Version Two

In this version the same principle used for the zip sensor was used, which is having one analog port that measures the voltage output, changing when the resistance changes. In this way, it is possible to have just one analog port instead of five. Moreover, is possible to have as many outputs as desired with just one analog port measuring the output voltage. The following Figures 105 and 106 show the prototype.

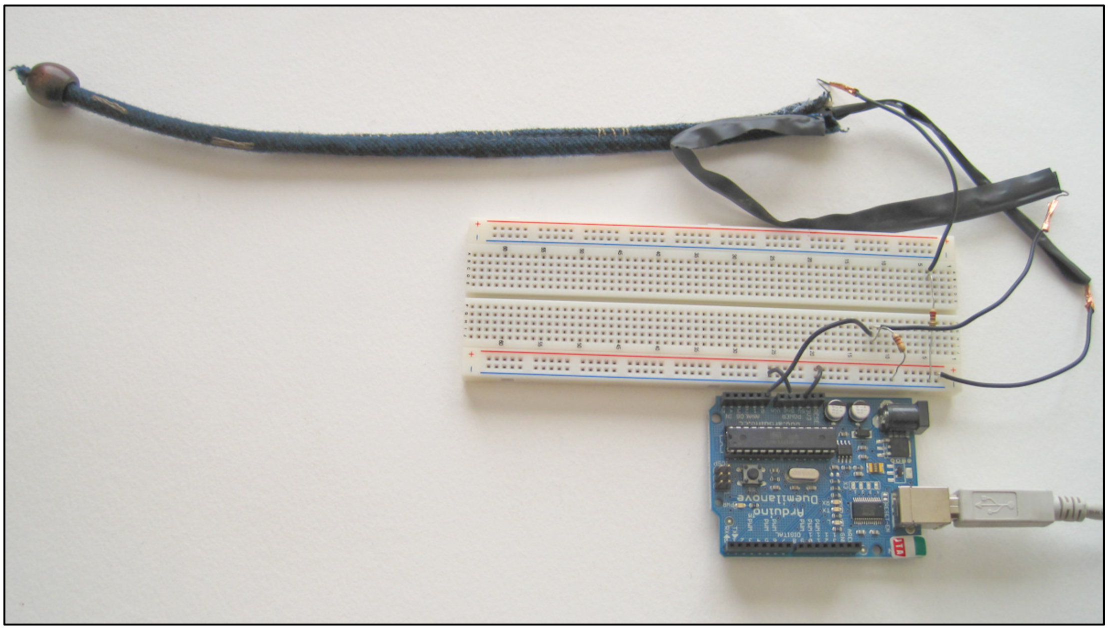

Figure 105 – Toggle sensor using one analog ports, high-quality fabric

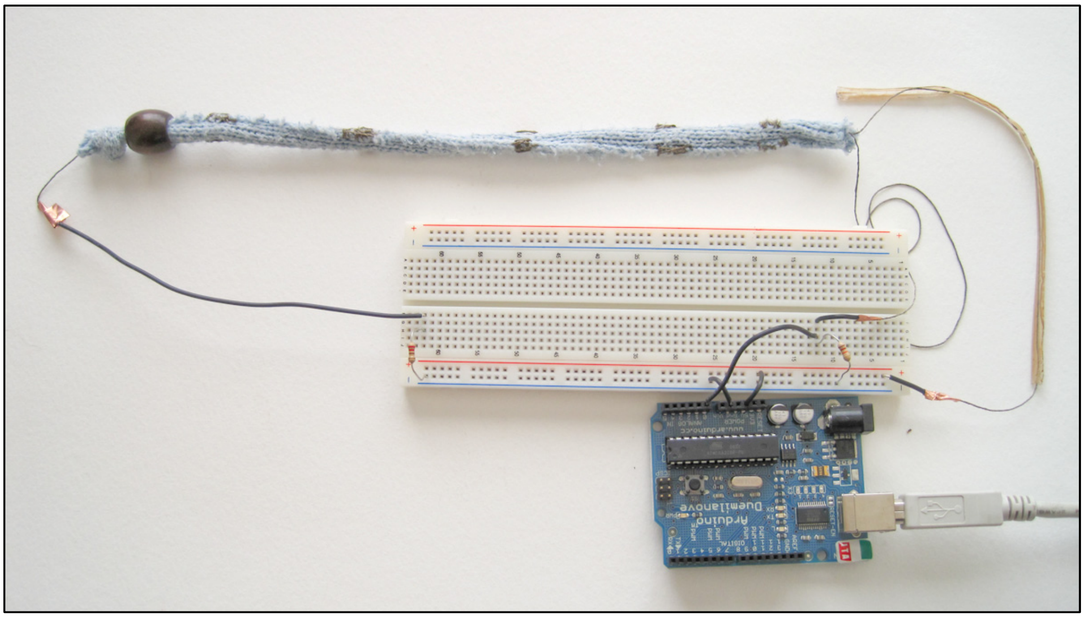

Figure 106 – Toggle sensor using one analog ports, low-quality fabric

5.6. Development

On one side of the toggle a conductive thread was sewed and its top part was connected to 5V and the bottom side a 820Ω resistor and then to ground. This resistor was used to limit the voltage in the circuitry since the conductive thread has a low resistance. Then on the other side a conductive thread was sewed at five locations and connected to only to one analog port, which was A0. From the analog port a 12kΩ resistor was attached to stabilize the values (Figure 107).

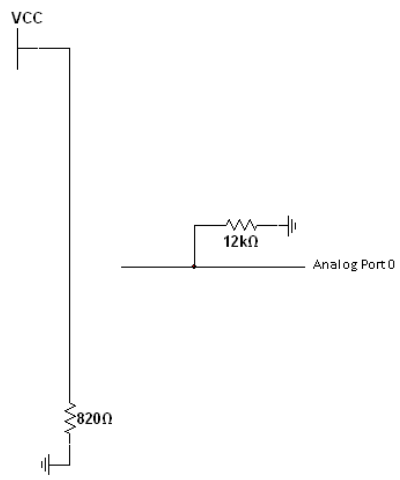

Figure 107 – Toggle sensor using one analog port schematic

5.7. Material and Tools

Conductive thread 117/17 2ply, fabric, wire, bead, and a 12kΩ and 820Ω resistors were used to build this version of the toggle sensor, as well as tools such as sewing needle, sewing machine, fabric scissors, and a multimeter.

5.8. Performance

The reliability of this prototype was tested to detect which position was being selected. Three data samples were registered and are presented bellow. The following charts refer to low-quality fabric.

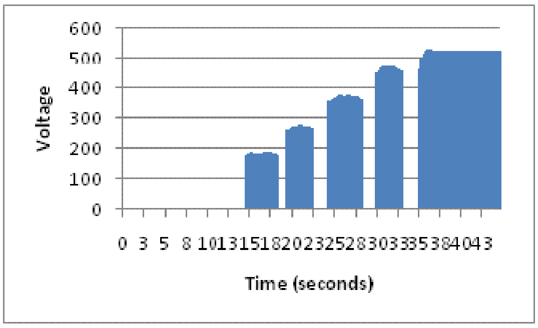

Figure 108 - Toggle sensor using one analog port, low-quality fabric; first data sample

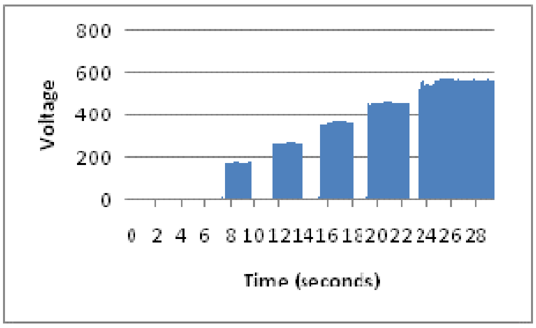

Figure 109 - Toggle sensor using one analog port, low-quality fabric; second data sample

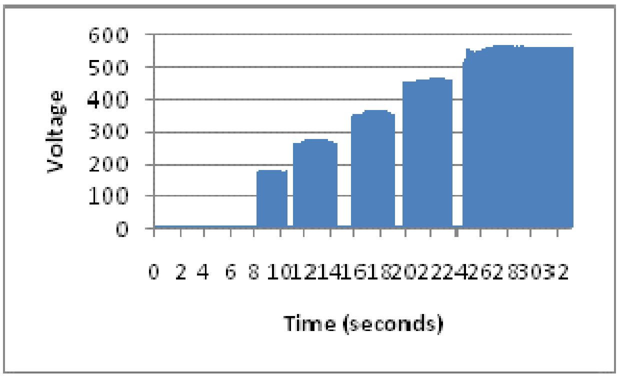

Figure 110 - Toggle sensor using one analog port, low-quality fabric; third data sample

The following charts refer to high-quality fabric.

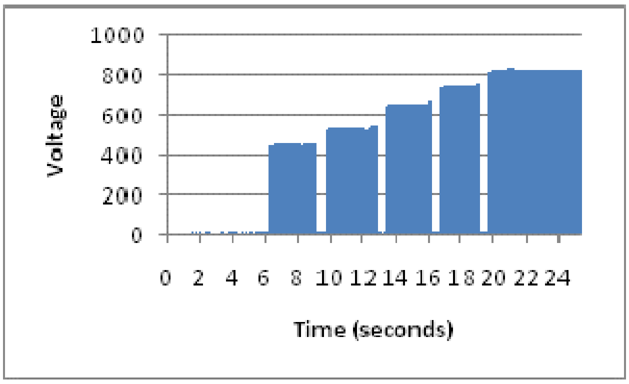

Figure 111 - Toggle sensor using one analog port, high-quality fabric; first data sample

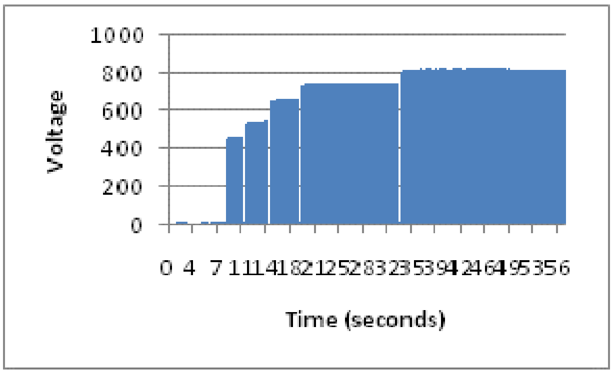

Figure 112 - Toggle sensor using one analog port, high-quality fabric; second data sample

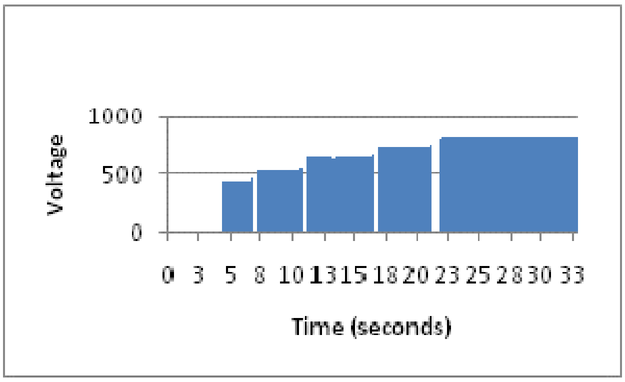

Figure 113 - Toggle sensor using one analog port, high-quality fabric; third data sample

As can be seen from the charts, it is possible to see where the selection is being made, without occupying several ports.

5.9. Toggle Sensor Conclusion

The second version of the toggle is better because it is efficient, as it uses only one analog port. It does not occupy as many ports as the first version. It was noticed that whether using a fabric that frays or another that does not fray (Figure 114), both are reliable.

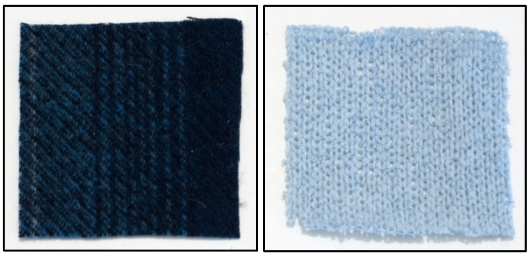

Figure 114 - Wool Fabric

In general, this sensor can be used on wearables to control different outputs. It has a precise feedback that occupies only one analog port.

6. Summary

In this chapter three button sensors, a stroke sensor, a zip sensor, a stretch sensor and a toggle sensor were built. Zip, stretch and toggle sensors were built by us. Both of them had good performance. Some of them are good for applications that require analog inputs and others for digital inputs. Button and stroke sensors are suitable for digital input applications and zip, stretch and toggle sensors are suitable for analog input applications.

During the development of these prototypes it was learnt how to use conductive thread, conductive fabric and flexible stretch sensor and be familiarized with arduino syntax. The stretch sensor proved to be very sensitive, and when it has been used several times, its performance can be changed.

The next step was to build applications with these prototypes, which is the next chapter. Therefore, the next chapter will show all the process of building a wearable using some of these sensors, and new ones will arise also.

MusicHoodie Application

This chapter presents the construction of a pullover that works as an MP3 player. It differs from the majority of wearables that were developed in the past which were based on buttons. The aim of constructing this pullover was to explore and create new ways of interacting with wearable computers.

This chapter is composed of four sections covering planning, design, hardware development and software implementation. The first section provides a description of the pullover features. Basically it includes all models. The second one includes the design of the wearable. The third one discusses the physical construction as well as the components and tools used. The final section describes the software implementation and architecture. Additional details and code are in appendices A to G.

1. Planning

The intended users of this system are people who use an MP3 player with some frequency whilst out and about. As well as regular users, this also encompasses groups such as joggers. Users are represented at Figure 115 by the Listener. Basically, the Listener represents everyone who will wear the pullover and interact with it.

The functionalities the system supports are those of a standard mp3 player. This includes playing, pausing and stopping music. Also, moving forward and backward in an album as well as operations for listening to music; random mode, repeat mode or normal mode. It is also possible to change the volume and select which genre or artist is being played.

In order to clarify how each mode works, a description of them will now be presented. In the random mode an album's music is played in a random way. On the other hand, in the repeat mode the album's music is played in sequential order. Both of them make a transition to a subsequent song when the current one is finished. The last mode that the Listener can interact with is the normal mode. In this mode the album's music is played sequentially but does not repeat.

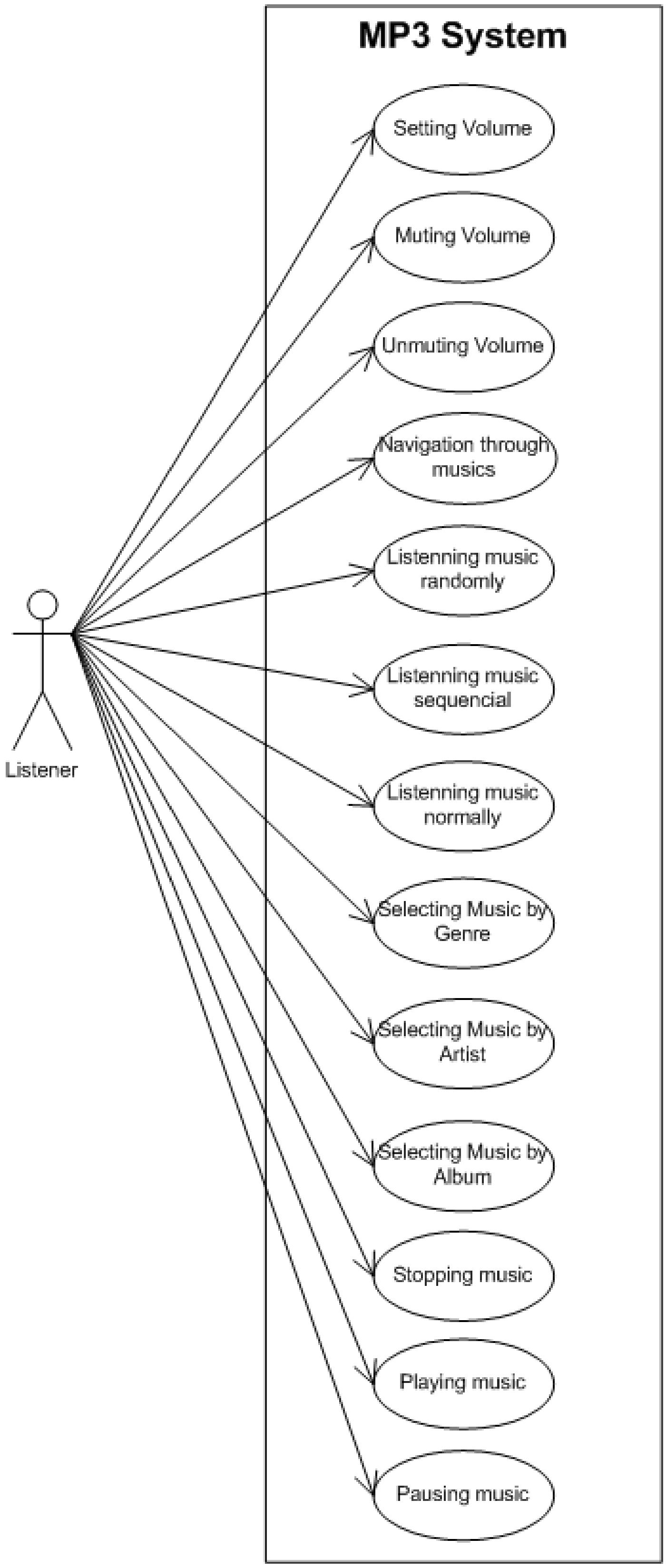

Figure 115 - MusicHoodie Use Case Diagram

2. Design

The design of the MusicHoodie was developed in Illustrator. The aim was to develop an intuitive and simple design for users to interact with. Since a hoodie is a casual style, the design was developed following urban lines (Figure 116).

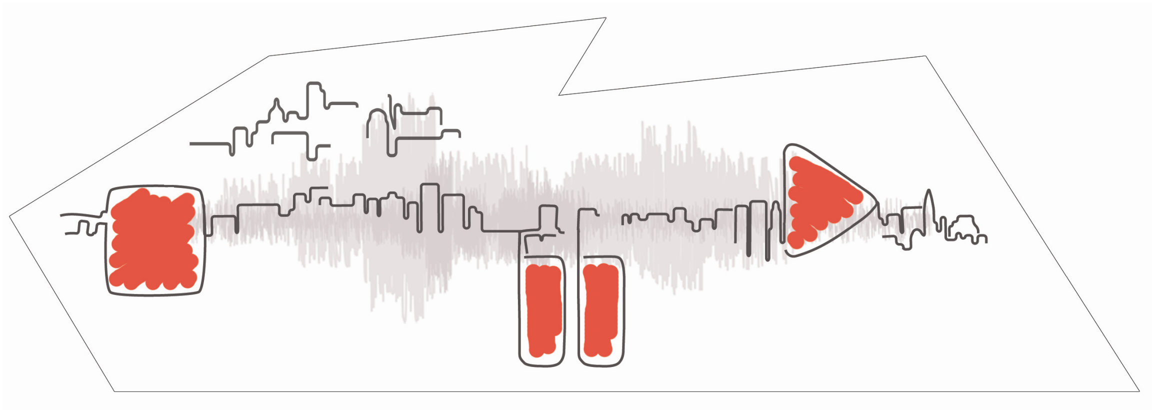

Figure 116 – Play, Pause, Stop music design

3. Hardware Development

Several materials and tools were needed to build the MusicHoodie. Some of them are washable and others not. For instance, the resistors, FTDI and LEDs cannot be washed. The resistors could be replaced by material that could be washed such as highly resistive conductive thread. However, this would add considerably more expense. The FTDI could be replaced by LilyPad battery and by this way could be washed. The LEDs could be replaced by the Buechley LEDs which are washable, making the jacket washable.

The list of material and tools will be presented below. The cost of the material list was around 170€.

3.1. Materials

- 1 pullover
- 1 zip
- 1.5cm thick foam
- 2 arduino LilyPad 168
- 2 FTDI
- 2 mini USB cable
- Three 100pF capacitor
- 32 rare earth magnets
- 4 LEDs
- Acrylic
- Conductive fabric
- Conductive thread 117/17 2ply
- Copper tape

- Fabric ink
- Flexible stretch sensor
- Fusible fabric
- Non-conductive fabric
- Non-conductive thread
- Paper
- Four 220Ω resistors
- Four 1kΩ resistors
- Four 2.2kΩ resistors
- Fourteen 12kΩ resistors
- Six 10MΩ resistors
- Shrink tube
- Silicone
- T-shirt transfer sheets
- Solder
- Wire

3.2. Tools

- Brush
- DeskJet printer
- Hair Dryer
- Iron
- Needle
- Fabric scissors
- Sewing Machine
- Soldering Iron
- Multimeter
- Pliers
- Stripper tool

4. Construction

The pullover was built in several parts, like modules. Each module was designed, created tested and refined individually. The final stage of the work was the integration of these components. The next sections are the descriptions of these modules.

4.1. Album Attachment and Selection

The first module developed, was the album sensor. The idea was that using this sensor people could choose which CD they would like to listen to. The way that they would select would be by the artist, genre or album. So, the idea that arose was that people could pick a CD and then attach to the pullover to listen to it, like what people usually do when they want to listen to something in the CD player. They just pick a CD and insert it into the machine in order to listen to it.

4.1.1. Album Tokens

Small wearable tokens representing the album's music were created in order to provide a experience similar to that of the CD player interaction, described above. They were created by printing the album CD cover on the fabric. Then, magnets were sown to acrylic, and put in the middle of the album fabric. Then this sandwich was sewed on the edge. At their back was done a circuitry with solder and copper tape, which makes arduino identify each album (Figure 117 at right). It was used solder for a better contact, because initially was being used only copper tape but was not doing a good contact. Fifteen were made in all. Four of them could be attached to the pullover at a time, and then the user only needed to select which one he wanted to hear. The albums used were the "Demons Days" Gorillaz (Figure 117 at left), the "Party Time Funky House Classics", the "Viva la Vida" Coldplay and the "Chill with Bach".

Figure 117 – "Demon Days" Gorillaz album

4.1.2. Album Tokens Attachment

The first version of the album tokens attachment was constructed with Velcro (Figure 118). It was immediately obvious that Velcro was not a good idea. Velcro has problems such as highly inconsistent contact adhesion. This is a problem if you want to use conductive materials in the Velcro layers. As the material ages, electric contact begins to degrade and will eventually be lost.

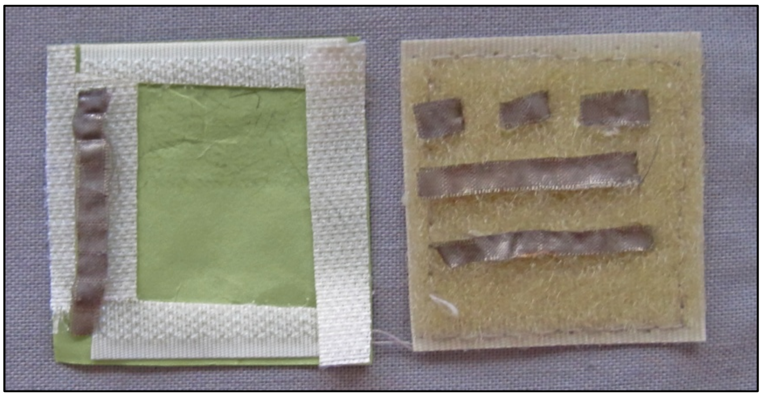

Figure 118 – Velcro Album Selection

The second idea for the selection sensor also had some problems. It was constructed with fasteners (Figure 119). It solved the problem of the poor contact in Velcro, but it was too much work to manually connect fasteners in order to have the CD plugged into the wearable.

Figure 119 – Fasteners Album Selection

A good solution was simply magnets (Figure 120). They easily attach and unattach to the wearable, and also they make good contact. Moreover, they are aesthetic.

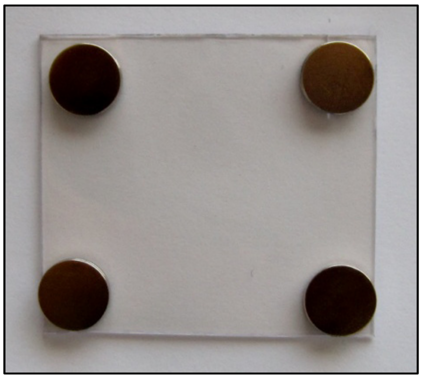

Figure 120 – Magnets Album Selection

However, several things must be considered when using magnets on clothing. Firstly, the thicker the surrounding material, the stronger the magnet that should be used. Secondly, in order to ensure that magnets stay on the fabric, they must be well fixed. Super glue does not work with rare earth magnets. The solution is to sew them onto the fabric. However, if sewing is not a good idea, hot glue can serve as an alternative. However, a large amount of glue will weaken the magnetic field. An aspect that improves the magnetic field is when attaching the magnets to the clothes you put them as carefully as possible. This is not a simple thing to do in wearables; you need to be very precise, and sometimes the magnets slip during gluing.

4.1.3. Identification Album Token

The original way that was thought to identify the albums that were attached was through RFID. However, RFID readers are typically too large for a wearable scenario and that would not be aesthetic in this wearable.

An attempt to use a 100mm pressure sensor to identify which album had been selected was made. However, this sensor is not appropriate for wearables. Firstly, the sensor is felt by the wearer as being somewhat rigid and unwieldy, which is uncomfortable. It feels it like a piece of plastic. Secondly, for wearables that bend it would not work. The idea was to identify which album was being pressed, but since the pressure sensor was located on the arm, when the user bends the elbow, it interferes with the reported data.

So a conductive fabric and conductive thread circuitry was developed to replace the RFID. The conductive thread was used to make the connections to the microcontroller, and the conductive fabric for the connections with the album.

This circuitry to identify the album tokens works as a matrix. There are three rows and sixteen columns (see figure 121) made of conductive fabric. When the album is attached the three rows get connected, due to the solder that is on the album's back. Since all columns and rows are conductive fabric the solder conducts voltage through the rows.

Each row has a different responsibility. The third one identifies which album was attached. The way that works is with a four-digit binary code. Each album has a different binary code. Since it is a four-digit code, it is possible to have fifteen albums. The code 0000 is not possible because it is the default code, which means when none of albums are attached, the arduino reads this code. Arduino is connected to four digital ports and these are the ones that create the four-digit binary code. Each digital port, by default, reads the value LOW. When an album is connected, the line on the back of the album token connects one of these ports to 5V and the arduino begins reading the value HIGH. In this way, the combination of these four digital ports generates the album code. The line on each album back creates a different code. The ones that were used in this wearable were 0001, 0010, 0100 and 1111. However, other codes could have been created as long as they were between 0001 and 1111. It is important to notice that every four digital ports that give the code are the same in each different spot where the users can attach the album. (See Figure 121 with special attention to the third row).

The second row identifies the position of the album on the pullover's sleeve. The way that works is that every spot is connected to a different arduino port. There are four spots, which mean that there are four different digital ports. When the album is inserted in the spot the respective port gets HIGH. In this way it is known where the album was placed.

The first row is only connected to 5V so its responsibility is only to give voltage to the other rows, through the line on the back of the album.

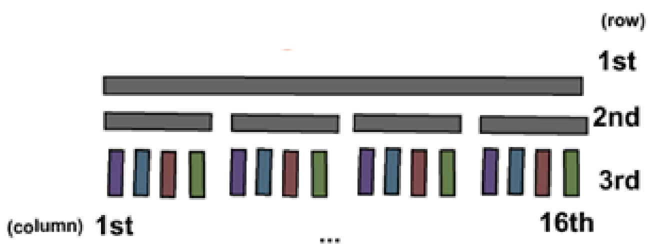

Figure 121 – Representation of the album sensor, all rows and columns are made of conductive fabric

With this system, what was missing was the ability to identify which album was being selected by the user. So, four button sensors were developed and put inside the pullover, behind the pullover's sleeve circuitry. These four button sensors were built as described in the prototypes chapter. Each one of those buttons is connected to a digital port. When one of them is pressed the arduino reads the value HIGH, on the respective port. In this way, it is known which position was pressed by associating this information with the system that reads the album code; it is possible to determine which album was pressed.

4.2. Change Volume

The next module that was integrated was the zip sensor. The construction process was the same as described in the prototypes chapter. The sewing of the zip on the sleeve was delicate due to the space available - there was little space to manipulate the sewing machine.

4.3. Play, Pause, Stop Music

After sewing and testing the zip sensor, the capacitive sensor was implemented. The capacitive sensor was used to issue commands to play, pause and stop music. Three capacitive sensors were built. They were built following the same process. The difference was that each one of them was connected to three different arduino analog ports so that each could be activated and sensed individually. The idea arose from the capacitive sensors that were built using foil described at State of the Art Chapter 2.

They were constructed by attaching a fusible fabric to a piece of conductive fabric (Figure 122), and then all the rest was software implementation. The arduino has a library that turns arduino ports into capacitive sensors, the Capacitive Sensing Library (20). To use this library there is a need for some circuitry considerations, such as placing a resistor between the arduino ports. What is recommended to use is a piece of foil in the port, which will sense the electrical capacitance. However, in this case conductive fabric was used instead of the foil because it is more suitable for wearables. It is also suggested to attach a capacitor connected to the ground in parallel with the conductive fabric, in order to stabilize the sensor values. Another important aspect is that since the conductive fabric is what is on the inside of the pullover, a fusible fabric is required in order to avoid the sensor sensing the electrical capacitance of wearer's chest. The wires that connect the capacitive sensor to the arduino sense capacitance. In order to avoid this, they have to be isolated with shrink tube, and then linked into a conductive material connected to the ground. One possible way to connect the conductive material to the ground is using conductive tape that is grounded. In this way it was guaranteed that the change in electrical capacitance is only detected when someone touches the sensor on the outside, like putting their finger over the sensor.

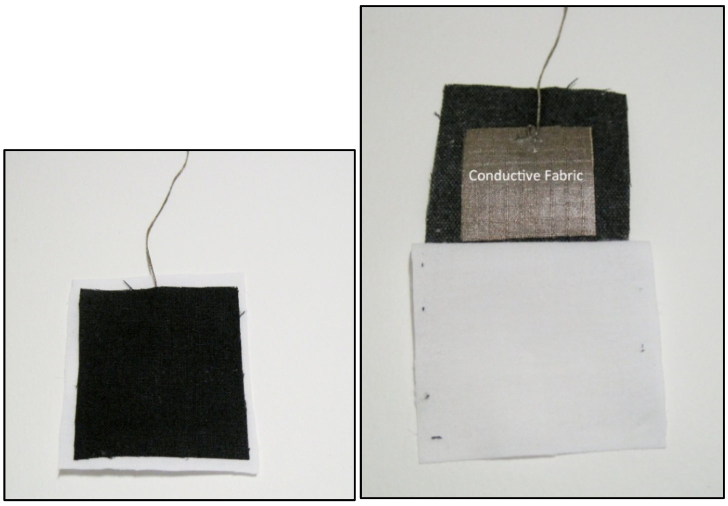

Figure 122 – Capacitive Sensor

Capacitive sensors are sensitive to the thickness of the surrounding material, also. Generally, the greater the thickness of the surrounding material, the higher the resistance to put in the circuit. This resistance makes it more sensitive. For instance, the greater the resistance, the greater the distance over which the capacitive sensor can detect changes. However, since the material is thicker, the distance at which it can detect changes to electrical capacitance will be really closer to the fabric.

4.4. Mode Selection

By this time, the volume and selection mode was already implemented and all circuitry tested and working perfectly. Afterwards, the mode Selection was integrated using the toggle sensor. It works the same as the zip. O one side, there is a conductive thread the top of which is connected to 5V and the bottom to the GND. On the other side there is another conductive thread that is connected to an analog port. They do not touch each other. There is a bead that the user uses to select the mode. That bead has conductive thread inside. So, in this way one conductive thread line is connected to another line. The circuit can only be completed through this bead. The way that this works is this: when the user selects the Repeat mode or Randomize mode, the analog port reads a specific value and in this way it is known which mode is being selected. The normal mode is detected if the repeat and randomize mode were not selected.

4.5. Navigation Album's Music

The last module is traversing back and forth through the music on an album. They work the same way, the only difference is that one is programmed to go forward on the album's music and the other to go backward. Both of them were developed using a stretch sensor placed in the bottom hem of the pullover. When the user stretches the hem, the music goes backward or forward depending of which part of the hem is being stretched. To prevent inadvertent activation when the user is simply wearing the pullover, two buttons were attached at the ends of each stretch sensor. In order to enable input from the stretch sensor, those buttons must be pressed in order to override the protection against unintended activation. The buttons were like the one in the prototypes chapter.

4.6. Integration of All Modules – Final Prototype

All sensors described bellow at this state were all already tested and working. Final adjustments were done and then the implementation was done. The next pages are images of the pullover, from Figure 123 – 126. Then, all the schematics of the wearable are showed, at Figure 127 and 128.

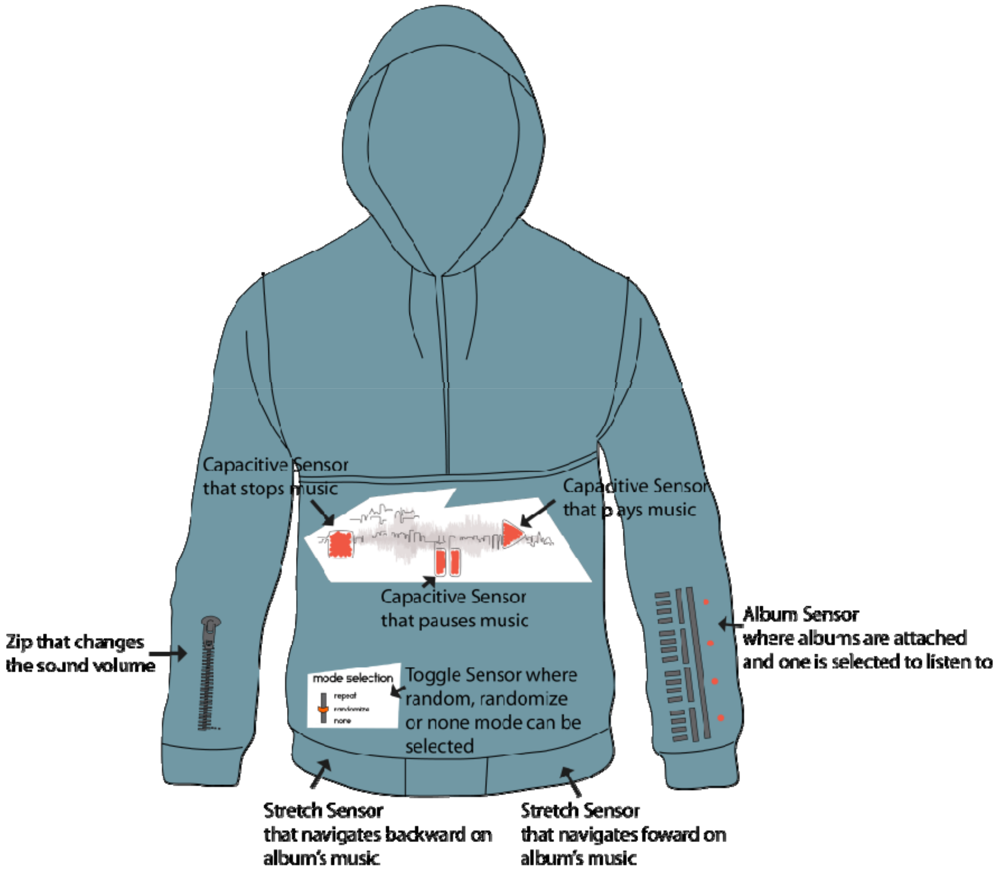

Figure 123 – MusicHoodie Final prototype Illustration

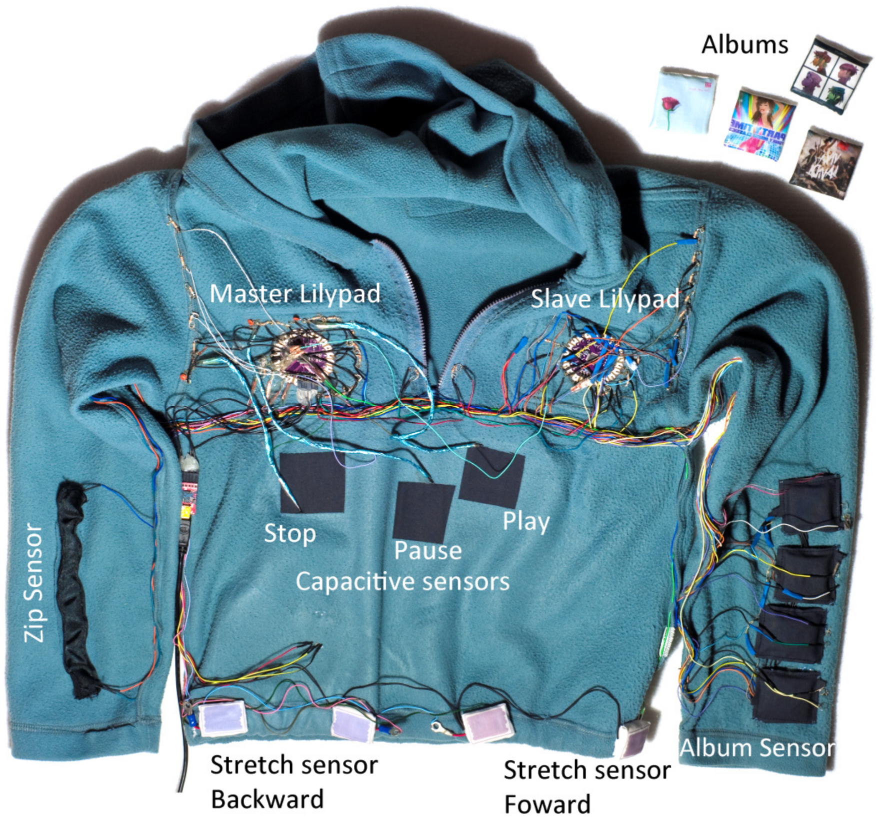

Figure 124 – Final Circuitry

Figure 125 – Final MusicHoodie without album

Figure 126 – Final MusicHoodie with album

4.7. Schematic

This section shows in detail the circuitry developed. At Figure 127, is presented a representation of the circuitry that was implemented at the pullover. This representation was done using the Fritzing software. At Figure 128 is showed the technical schematic.

It is important emphasize that the circuitry was all done at the pullover, without using breadboards as support. The illustration with the breadboard has the same circuitry. It was done for a better visualization.

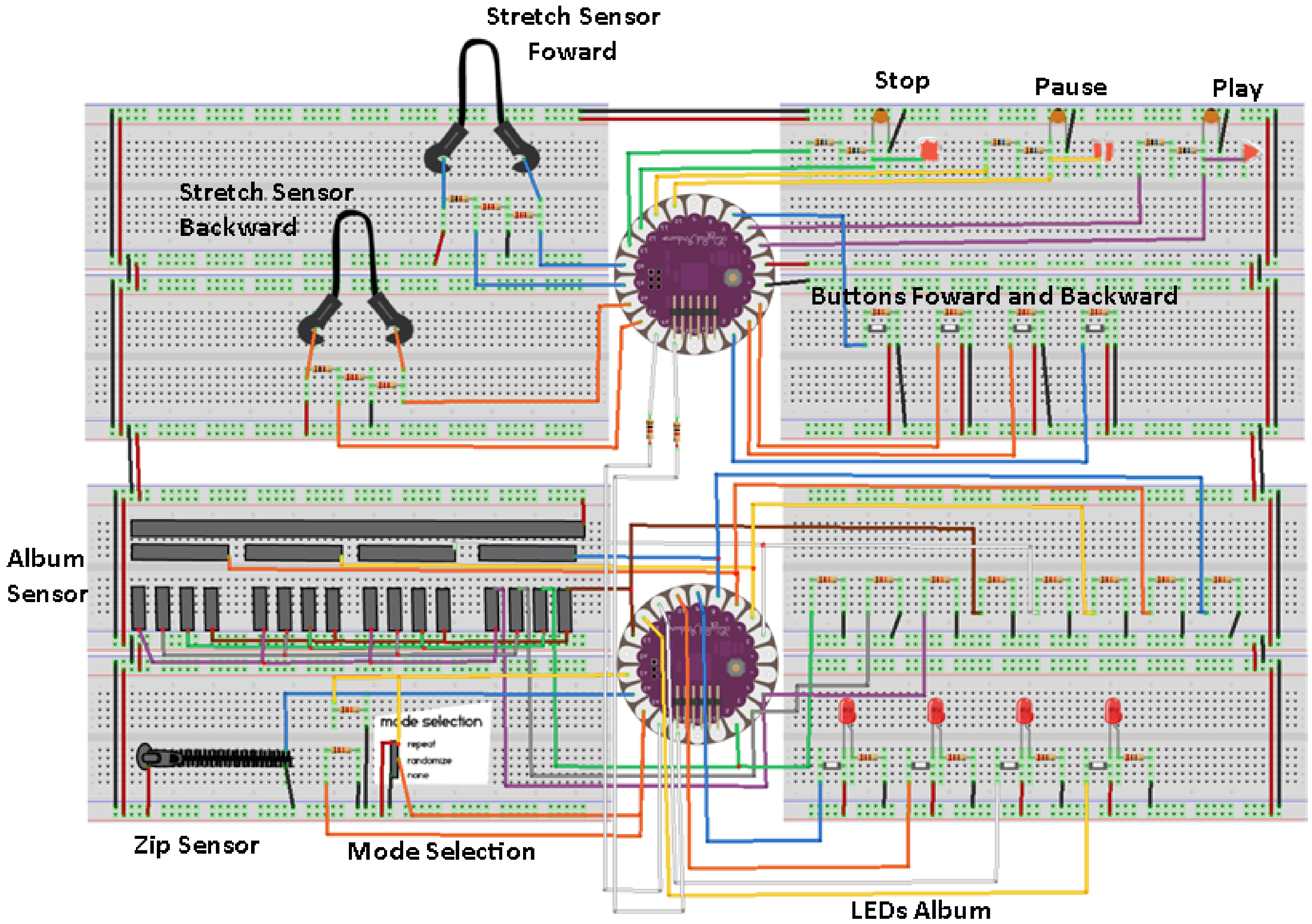

Figure 127 - Breadboard circuit scheme of the MusicHoodie made on Fritzing. For an enlarged view, consult the CD.

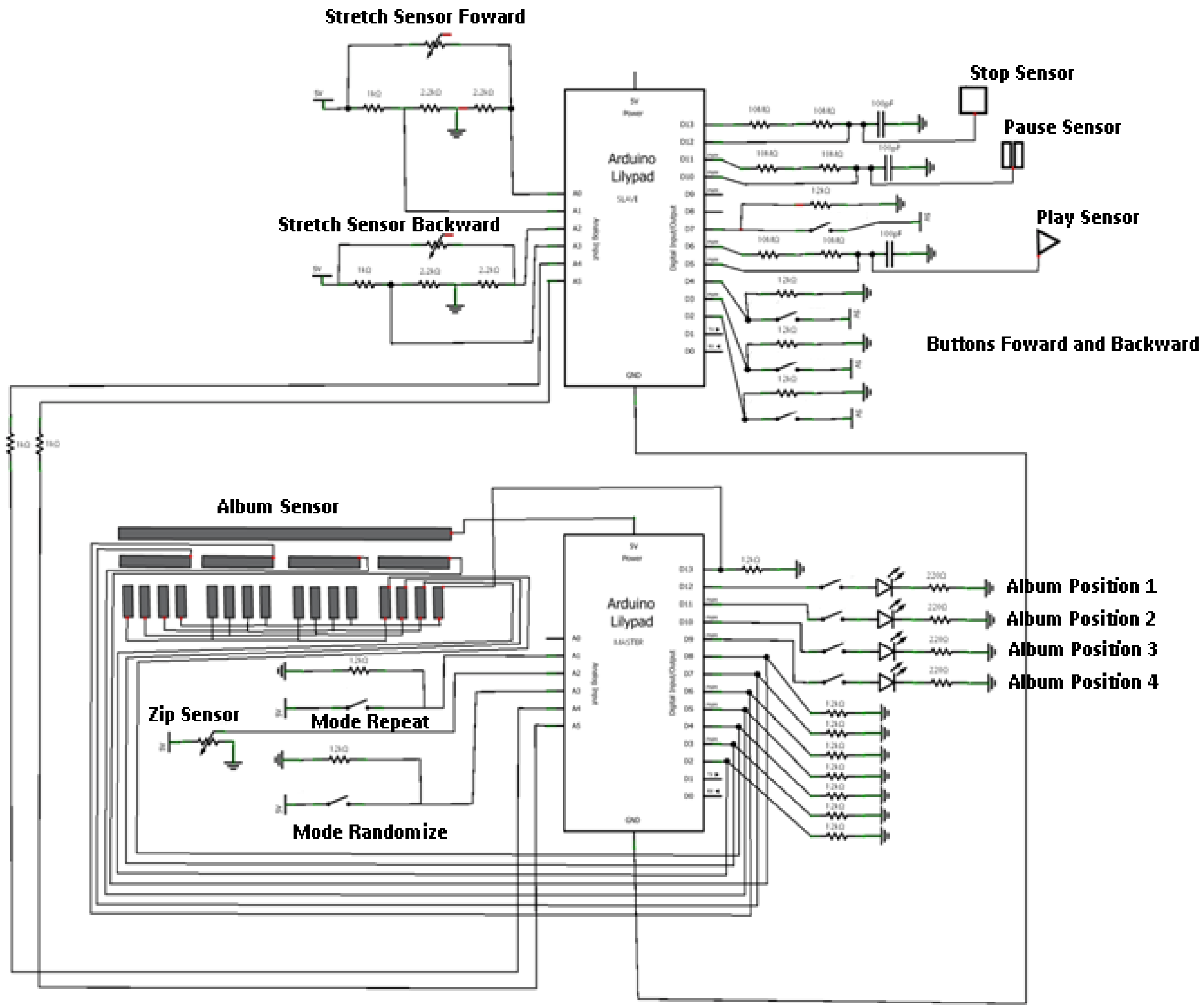

Figure 128 - Logical circuit scheme of the MusicHoodie made on Fritzing. For an enlarged view, consult the CD

5. Interface and Operation

The user wears the pullover and then attaches the CDs that he wants to listen to. He can upload only four CDs. The CDs are attached to the pullover with magnets. There are magnets on the inside of the pullover and on the inside of the album, so when the wearer is trying to attach the album, it is automatically magnetized to the pullover. Then the arduino reads which album is in each position of the pullover.

Afterwards, the wearer needs to select which album he wants to listen to. He only needs to press the album that he wants to hear and his choice is saved. An LED turns on, and he has the feedback that his album was selected. At this time, he is able to play any music from the album. He only has to touch on the play button. The arduino reads if there was a change in the capacitive sensor and if there was, it plays the music. Afterwards, the wearer is able to go forward or backward on the album. He can go forward through the music on the album by stretching the left hem of the pullover, and backward by stretching the right hem. Every time he goes forward or backward he hears the name of the music at the same time the music begins playing. This is only to give him a feedback of which music he is listening to, because the system does not have an LCD on the pullover to check the name of the music that begins playing.

Afterwards he can pause, stop or change the volume. The pause and stop are like the play. In order to get the volume changed, he only has to push the zipper slider up or down and the sound gets louder or softer. If the slider is pushed up the music gets louder. Otherwise, if the slider is pushed down the music gets softer.

This wearable also conceptually supports exchanging albums with friends. For instance, if the wearer finds a friend that is wearing the same pullover and if he wants to listen to his friend's albums, he actually can. They only need to swap albums and attach them to their pullovers. The wearer can also choose how he wants to listen to the album, if it is in a random or repeat mode or neither of them. He only has to slide a bead up or down according to his desire.

6. Software Implementation

The system required two LilyPad arduino microcontrollers to accommodate input from all the sensors. One of them was programmed as a slave and the other one as a master. In this way all values from the sensors could be retrieved. The master was programmed to read only the values from the zip, mode selection, and album sensors and send it to the slave (Figure 129). Then the slave LilyPad received all values that the master LilyPad reads and also reads the play, pause, stop, and stretch and button sensors. Then, all values are sent to a host PC running an executable in the processing programming language. This application reads and processes the data. It then asks the slave for more values. This architecture between slave LilyPad and processing was chosen, in order to prevent the communication buffer from getting full.

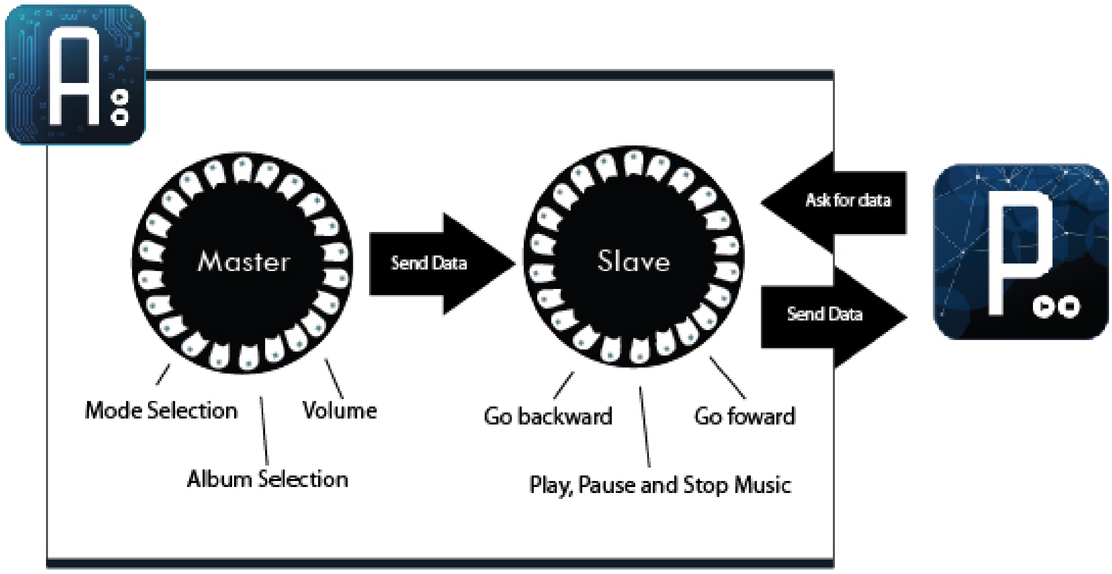

Figure 129 – Connection between Arduino and Processing

The code that was implemented is composed of several classes (Figure 130), each one with different responsibilities.

The class WearableMP3 is the main class. This class receives the sensor values from the slave and is the one that analyses whether any sensor was activated.

The Stabilizer class is the one that stabilizes the values from sensors. For instance, the stretch sensor is quite unstable so the values were normalized with this class. The spikes that were interfering with getting accurate values were normalized, as the rolling average between them was calculated and the values that were too far from each other were discarded. This class was only used for the stretch sensor and for the zip, for guarantee 99.9% stability.

The album class is responsible for all options that can be made with an album. In this class, music is created and associated to an album. This is the class responsible to play, pause, stop, go forward, backward, change volume, and repeat or randomize an album's music.

The music class is responsible for creating all music objects, and also the commands of play, pause, stop and changing the volume of music.

The voice class is the class that creates all objects with the names of the music for display to the user. It is responsible for playing and changing the volume of the music chosen.

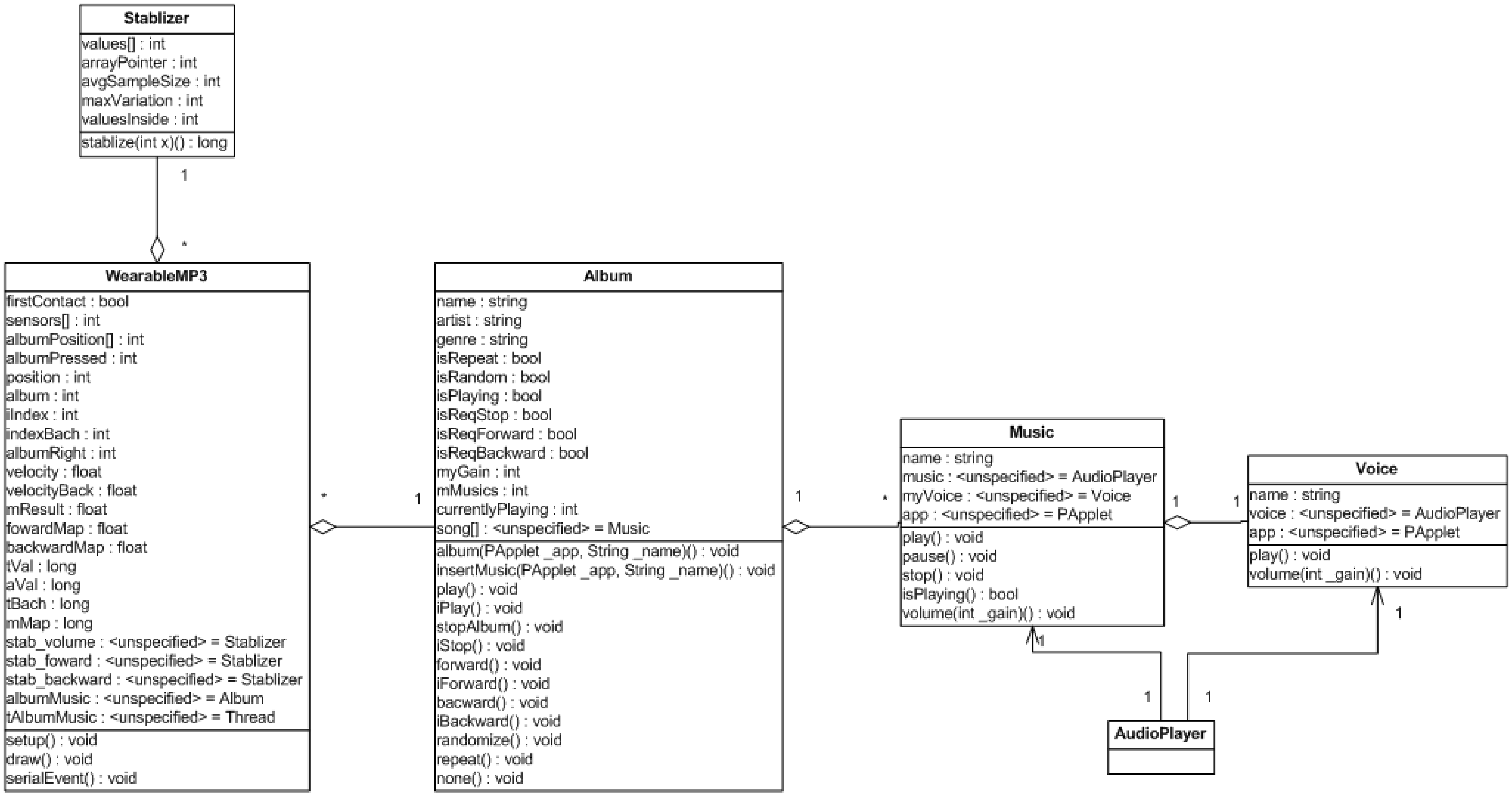

Figure 130 – MusicHoodie UML

7. Summary

In this chapter five different ways to interact with wearables were presented. After developing the MusicHoodie, conclusions arrived and are synthesized in the following paragraphs.

Some wearable materials are delicate and fragile. Conductive fabric unravels easily, and the edges need to be insulated so that it does not cause short circuits. A possible way to avoid this could be by doing zip zag sewing on the edges. On the stretch sensor, after stretching several times, holes may appear. This interferes with the output values that the stretch sensor sends. Care must be taken with this sensor, like not stretching it too much, or using it too much. Ultimately, current sensors may not be suitable for use in wearables.

From all the circuitry path insulation tested in this wearable, the most efficient one was painting the conductive thread with fabric ink. Gluing the conductive thread with UHU glue is not a good solution, because some parts of the path do not get insulated. The best choices tested are shrink tube and fabric ink.

In regard to the microcontroller used, it is quite slow. In this LilyPad Version (168) the reset button has to be pressed every time the code is uploaded. Depending on the place where the arduino is attached, sometimes it is not easy to press the reset button. For future work, the use of the 368 version is recommended.

In the next chapter a user study that was made to the pullover to see how suitable all these interactions and sensors are for wearables, will be presented.

User Study 5

This chapter will describe the process employed to do a usability test on the MusicHoodie wearable. Usability tests are carried out to obtain feedback about how real users would interact with a prototype, find issues and fine-tune and optimize them before building a deployment version.

Wizard of Oz, a qualitative method, was chosen in order to assess whether the interactions with all the components were usable and effective. Wizard of Oz is a usability test in which someone is controlling the interface without the user's knowledge, while he is interacting with it. During the process users were asked to verbalize what they were thinking, which was recorded without their being aware, so that the data could be analyzed later. Also, notes were taken during the process. This data was put together and was analyzed to create the results. After interacting with the wearable, the users were asked some questions, in order to obtain feedback about their impressions of the overall system.

1. Wizard of Oz

Wizard of Oz is a usability test which makes it possible to understand which visual interface aspects are not clear and also identify interactions that are not natural, which are required to perform specific tasks. There is a wizard which controls the system, while the user is interacting with it. Usually the wizard is in another room so that the user does not realize that the system is being controlled by someone else. Then, tasks are given to the user and the evaluation begins. While performing the tasks users are asked to think aloud, notes are taken and the evaluation is recorded. After performing all the tasks, user is asked to answer some questions, which can be consulted in Appendix H. Wizard of Oz is useful for expensive prototypes and in situations where it is hard to know beforehand how the user will naturally interact. (43)

In this case, Wizard of Oz was chosen because the wearable circuitry is very sensitive. Music albums have to be attached in a specific way; otherwise the pullover does not play music. This could make the user frustrated, losing too many time to put the albums in the right position, and could affect understanding of whether the new types of inputs are natural or not. This could be solved by building a more robust circuitry, but it would involve a much greater cost and would require more time. So, the Wizard of Oz was chosen for the testing.

The user test was carried out with 6 mp3 users. At the beginning of the evaluation, personal questions were asked such as age and gender, which can be consulted in appendix H. A general overview of the MusicHoodie was given with a short explanation about what they were going to do.

After the users understood what they had to do and had answered the personal questions, they were asked to perform the following tasks:

1. Choose "Viva La Vida" album
2. Play music
3. Lower the volume
4. Pause and Play the music again
5. Go to the next song
6. Stop and Play music
7. Go to the previous song
8. Increase the volume
9. Select randomize mode

While they were performing the tasks, they did not notice that they were not the ones who were controlling the prototype. They were asked to perform some of the tasks several times, such as play, increase or decrease the volume, in order to know if the interface was easy to learn. While they were performing the tasks they were interacting with a mannequin which was wearing the pullover (Figure 131).

Figure 131 – Mannequin wearing the MusicHoodie wearable

2. Results

The users were between 26 and 55 years old and were Portuguese. Three were male and other three were female. All of them use mp3 players; four use it daily, one weekly and one monthly.

2.1. Attachment and Selection of Albums

This input was well accepted by the users. They liked this interaction and liked having the albums on their sleeve because it is different and cool. A user told us that it was like seeing the time, simple and easy. Since the albums are attached on the sleeve, while wearing the Hoodie they would be showing what kind of music they are listening to. This could bring privacy issues, and one aim of this test was to evaluate that. Most users said that it would not bother them to show the music that they were listening to. It would be a good way to socialize. However, there was one user who said he would not to like to show which albums he was listening to. He would like to have his privacy.

2.2. Navigating the music on the album

In going backward and forward, people usually did not understand where it was. They saw the buttons, but it was not clear to them that they had to stretch to go forward or backward. However, after learning the procedure, they did not forget. Also, it was not clear which one was to go backward and which one to go forward. So, some labels were attached. Users said that this interaction was too far away from the other inputs. This is because, firstly, they interact with mode selection and zip, which are on their arms and only afterwards they interact with these inputs, which actually are in a different location on the pullover. Besides being far away from the zip and mode selection, users did not like to have to interact with both hands. Also, they had to look to see exactly where it was they had to perform the interaction.

2.3. Play, Pause, Stop music

The design of the capacitive sensors to play, pause and stop a song were not intuitive for users. They thought that the whole central part of the pullover, which has the three buttons, was only an input. It was not intuitive that there were three inputs. Afterwards, however, they discovered that there were three inputs, play, pause and stop. They were capable of interacting again with them without hesitating to think about which one plays, pauses or stops a song. Some users had to bend to see these inputs, which they said that is not comfortable. Also, after eating they said that they would not like to touch their belly to play, pause or stop albums, because the touch would be uncomfortable for them.

2.4. Change Volume

The reaction to this input was good. Users liked to interact with it. At first users did not understand that a zip could change the volume of a song. After discovering this, they found it a lot of fun and intuitive, because it is just pulling up and down to make the music louder or softer. Interesting and practical were the adjectives used to describe this input. It was one of the most well-accepted inputs. Users liked its simplicity and location. As is the case of the mode selection input, this one can be interacted with using only one hand and does not demand too much attention to control it.

2.5. Mode Selection

At first glance users did not understand how they could select a mode. Firstly, they thought it was by touching the words, and then they realized that it was by pushing the bead up and down. As was the case of the backward and forward input, they found it uncomfortable having to use both hands to pull the bead up and down. It required too much of their attention to put the bead at the right place. Also, the words are not the most appropriate, because a user thought that the “repeat mode” was to repeat the same song over and over, and what it actually does is repeat the songs of the album sequentially.

2.6. General Overview MusicHoodie

The most well-accepted inputs were the album selection and sound volume control. Both of them were useful, intuitive and interactive. People liked them because they were simple, easy to use, and they could use only one hand without it requiring too much of their attention. The place where they were located was also well accepted.

All inputs that where located on the belly where not well accepted, such as play, pause and stop music; going backward and forward through the songs on the album ; and mode selection. Besides the location, the last two required the use of both hands for interaction, which is not practical.

Regarding the whole interface, users would like to have all functionalities near each other, for instance, having all the controls on just one arm. The interface must have a quick learning process, to understand what does what. The words used for mode selection and play, pause and stop music inputs should be redesigned for a deployment version, making them more self-evident. Mode selection should be called Play Album Mode, and then only have repeat and randomize modes. Concerning the play, pause and stop inputs, they should be attached without any background. In this phase a background was used because the pullover is blue and a white background makes a better contrast for identifying them. However, this confuses users, as they think there is only one input.

Overall, users found it more enjoyable and ergonomic to interact with the MusicHoodie than with other mp3 players, because they do not need to carry a device in their pockets, and they have more fun. However, users would not like to wear a pullover with all these wires everyday.

2.7. Future Applications

Users would like to use this wearable for other applications also, like being a database of the music albums for all the players the user has, and also to control appliances.

This pullover could be used to control not just the mp3 player, but also the car and computer mp3 player. A user suggested this pullover be used as a small database that has the music albums and wherever you are or go, that music plays on the nearby device, because he sometimes listens to music on the computer and then when he goes to the car, he still wants to listen to the same music. In this way, by using this pullover he could select the album and then play it on the computer if he is at the computer; or if he is in the car, play it in the car.

Another user would like to use this jacket to control kitchen appliances, because she sometimes wants to watch television while making dinner and she needs to interrupt what she is watching. So, using the pullover to remotely control the microwave timer, for instance, or turn on/off the oven would be interesting. She would be able to do different tasks at the same time without diverting too much attention from the cooking task.

3. Summary

This chapter presented the user test performed on the MusicHoodie wearable. Firstly, the Wizard of Oz usability test was described, and the reason given for deciding to use it. Then the tasks that users had to do were explained and the users were described. Then conclusions were presented. Some inputs proved to be suitable and others not, for the application tested.

In the next chapter all the conclusions and analysis of this thesis will be presented.

Conclusion 6

The aim of this work was to extend the wearable technologies area. Since this is not a new field, the aim was to develop new inputs for fashion wearables. As was said before, many of the current wearables are based on buttons, and this could be transformed with a new step. The aim of this work was to use other types of inputs instead of buttons. It was also a goal of this work to provide other types of feedback instead of using screens, since screens are sometimes uncomfortable for the wearer. Hence, sound feedback was used.

The beginning of this thesis searched out which technologies were being used and which projects were already developed and were successful. Guidelines were learnt, such as not demanding full attention of the user and taking into consideration that their attention would be divided through different tasks. Regarding the technologies that are currently available for wearables, it was learnt that they should be used for different purposes. This gave the knowledge, guidelines and inspiration needed to begin exploring new types of inputs. Several prototypes were developed such as button, stroke, zip, stretch and toggle sensors.

The first two sensors were developed to obtain dexterity and familiarity with wearable technologies, such as conductive fabric and conductive thread. Through the construction of these two prototypes it was learnt that conductive fabric and thread fray easily and are highly conductive. The button sensor developed using conductive materials was cheaper than buying the one that can be washed, designed by SparkFun and Buechley. However, both of them are suitable for wearables. The fabric button sensor, like the stroke sensor, is good for building personalized shapes and pattern sensors.

On the other hand, the last three sensors were developed to explore new ways of inputs for wearables. The zip sensor began with the idea of exploring gadgets that already exist on clothes and make them interactive. Initially, the zip sensor was developed as a digital sensor, however, quickly proved to not be the best type of sensor for a zip. So, this sensor was transformed to act as an analog sensor. After it was developed, it was seen to be inaccurate, and a software filter and rolling average was applied. It was also noticed that it is not a good sensor to attach to clothes due to the resistors used in it. So, a last stage of the zip was developed. This one can be attached to any piece of clothing and is a very accurate sensor. During the construction of the zip, it was noticed that using conductive thread on the sewing machine was tricky. The conductive thread had to be put on the bobbin of the sewing machine in order to be able to sew it. This sensor can be used on clothes to increase and decrease brightness of LEDs and also to surf up and down on menus.

The stretch sensor was used as a multi-data sensor. The idea was that by using this sensor it would be possible to know where people were holding it and how much they were stretching it. The underlying principle used was the tension voltage divider. However, too many comparisons would have to be made to identify the positions with four or more positions. Also, the last two positions in a four-position stretch sensor were difficult to identify due to their slight change in values. The only way to obtain reliable values to identify the positions was by cutting the stretch sensor. By having different pieces of stretch sensor, where it was being held and stretched could efficiently noted. It was noticed that the stretch sensor is very fragile and in some situations it is not suitable for wearables.

The toggle sensor arose the same way as the zip sensor. Lots of clothes have toggles and an embedded sensor would make it interactive and unobtrusive. Firstly, one was developed using five analog ports, however there were too many ports occupied only to have five different outputs. Thus, a second version was built only using one analog port, in which it was possible to identify five positions or more with only one analog port. Both of the versions were accurate and could be easily attached to clothes, but the last version, which uses only one analog port, is more efficient.

At this stage, these sensors needed to be tested, so an application was developed to support this test. The application created was the MusicHoodie pullover which works as an mp3 player. In this application new sensors were added such as the capacitive sensor and the album sensor.

The capacitive sensor is an elegant and unobtrusive sensor that can measure electrical body capacitance. Precautions such as isolating the part of the sensor that is in contact with the body were needed. Otherwise, the electrical body capacitance was always being measured, instead of only being triggered when someone touches it.

The album sensor is a workable way of attaching objects to clothes, and provides the ability to select any one of the objects. It is a good solution to replace an RFID, and could identify up to fifteen objects.

Finally, a user test was done. Zip and album sensors were the most acceptable and understandable by users. Users liked to interact with this sensor. In their words, these were “fun, simple and intuitive”. They liked the fact they could interact with only one hand. Stretch, capacitive, and toggle sensors were the ones that people had more difficulties interacting with. Firstly, they did not like to use both hands to interact with the toggle sensor. This is a design aspect that could be fixed to only use one hand. Users felt that it was not normal to go forward and backward using the stretch sensor, and also they had to use both hands. The play, pause and stop sensor, which are capacitive sensors, were thought to be just a button. This is a design problem with can be improved. Because since the pullover was blue, and the controls were red, a white background was used to create a good contrast, which confused the users into thinking that it was just one button. An interesting aspect that was noticed was that the sensors that were not so well accepted were located on the front part of the body. Some users had to bend to see the controls. For the zip and album sensor they just had to bend the arm and control it. It was an interaction that they were used to, because they are used to looking at the time on a watch, and as for the girls, they are used to wearing bracelets.

After building this prototype and testing it, it was learnt that all circuitry should be done with conductive thread. Users cannot try a wearable prototype that has too many wires, because they can disconnect something. Also, a jacket is easier to wear, and since there are components attached it is the best solution. LilyPad arduino version 168 is not suitable for wearables, because it needs to be reset while uploading data or resetting the software and it is not practical to have to click a button every time that this is needed.

The most suitable and acceptable sensors for wearables were zip sensor and album sensor.

4. Future Work

Novel inputs were developed, but we only used them in one domain. We could apply it to other applications, such as cell phones, social networks and sports. Applying it as a global controller of mp3 players could be interesting, as well. People could listen to the same music on the computer, in the car, or while moving from one place to other.

Improvements to hardware should also be made, like using conductive thread instead of wire, also using the wireless shield Xbee and battery. At this conceptual level, these hardware improvements were not a high requirement. At this stage more important than these improvements was to test if the input devices where well understood and accepted. Also an improvement to hardware, such as integrating the pullover with the iPod was not taken into consideration. However, in the future the iPod breakout board should be used to connect the pullover with the iPod. Also all resistors should be replaced by conductive thread with high resistance. At a conceptual level using this kind of conductive thread is very expensive, but for further work they should be used, because resistors are not washable and their pins can broken in the course of long use.

Improvements to software should also be made. The only improvement to software should be to analyze the values that sensors are sending and ignore values that are the same as the previous ones read by the sensor. In this way the system will require less processor time.

In the future, privacy issues should be taken into consideration. People who would not like to share their albums with other people and would not like to show which music are listening to should have this possibility, for instance, using blank albums, without showing the cover.

I look forward to seeing future work in this area because wearables will be the future of computers and human clothing.

References

1. *MIT Glove Mouse.* [Online] http://web.mit.edu/kimt/www/6.111/final/index.html.

2. *DIMA: developing mobile computers for diabetic patients.* s.l. : Carnegie Mellon Engineering , 2006, iNews.

3. *Drishti: An Integrated Indoor/Outdoor Blind Navigation System and Service.* **Lisa Ran, Sumi Helal and Steve Moore.** Gainesville, FL, USA : IEEE, 2004. Proceedings of the Second IEEE Annual Conference on Pervasive Computing and Communications. 8188941.

4. *A.B. Zoss; H. Kazerooni; A. Chu.* **(BLEEX), Biomechanical design of the Berkeley lower extremity exoskeleton.** s.l. : IEEE Industrial Electronics Society ASME Dynamic Systems and Control Division IEEE Robotics and Automation Society , 2006. 8971830.

5. *TIKL: Development of a Wearable Vibrotactile Feedback Suit for Improved Human Motor Learning.* **Breazeal, Jeff Lieberman and Cynthia.** s.l. : IEEE Robotics and Automation Society, 2007.

6. Columbia Sportswear. [Online] http://www.columbia.com/Women%27s-Bugathermo™/BL1452,default,pd.html.

7. *In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 1395-1404. DOI= http://doi.acm.org/10.1145/1240624.1240836.* **Zhao, S., Dragicevic, P., Chignell, M., Balakrishnan, R., and Baudisch, P. 2007. Earpod: eyes-free menu selection using touch input and reactive audio feedback.**

8. *Wearable Computers: A New Paradigm in Computer Systems and Their Applications.* **Smailagic, A.** s.l. : IEEE, 2003.

9. *Design for Wearability.* **Francine Gemperle, Chris Kasabach, John Stivoric, Malcolm Bauer, Richard Martin.** Pittsburgh, PA : IEEE, 1998. International Symposium on Wearable Computers. 6097138.

10. **Ecko Unlimited, New York urban brand.** *Ecko Unlimited Release Audio Hoodie.* [Online] http://fibretronic.com/news/Ecko%20iPod%20Hoodie.

11. **D.Wickens, Christopher, et al.** *An introduction to Human Factors Engineering.* s.l. : Prentice Hall, 2003. 978-0131837362.

12. *GestureWrist and GesturePad: unobtrusive wearable interaction devices.* **Rekimoto, J.** Zurich , Switzerland : IEEE, 2001. International Symposium on Wearable Computers. 7114076.

13. Hug Shirt. [Online] http://www.cutecircuit.com/products/thehugshirt/.

14. *Stress outsourced: a haptic social network via crowdsourcing.* **Keywon Chung, Carnaven Chiu, Xiao Xiao, and Pei-Yu (Peggy).** New York, NY, USA : ACM, 2009. In Proceedings of the 27th international conference extended abstracts on Human factors in computing systems (CHI '09). 2439-2448. DOI=10.1145/1520340.1520346 http://doi.acm.org/10.1145/1520340.1520346.

15. *Privacy, Wearable Computers, and Recording Technology.* **Srub, H.B. Johnson, K.Allen, A. Belloni and Starner, V.** Pittsburgh, PA , USA : IEEE, 2002. Second International Symposium on Wearable Computers, 1998. Digest of Papers. 10.1109/ISWC.1998.729541 .

16. **Seymour, Sabine.** *Fashionable Technology: The Intersection of Design, Fashion, Science, and Technology.* s.l. : Springer-Verlag/Wien, 2008.

17. SparkFun Electronics. *LilyPad Button Board.* [Online] http://www.sparkfun.com/products/8776.

18. ConnectedWear by Fibretronic. [Online] http://www.fibretronic.com/connectedwear/keypads.

19. **DEW, Canadian brand.** *Introducing The Quiver by DEW.* [Online] http://fibretronic.com/news/DEW%20Quiver.

20. Arduino Capacitive Sensing Library. [Online] http://www.arduino.cc/playground/Main/CapSense.

21. *Wearable sensor badge and sensor jacket for context awareness.* **Farringdon, J., et al.** San Francisco, CA , USA : IEEE, 1999. International Symposium on Wearable Computers. 6416571.

22. *Gait Analysis Using a Shoe-Integrated Wireless.* **Bamberg, S., et al.** s.l. : IEEE Computer Society [Technical Co-Sponsor] , 2008. IEEE Transactions on Information Technology in Biomedicine. 18632321.

23. **Rubow, Erik.** *Posture Correction with Wearable Electronics.* December 4, 2008.

24. *eWatch: A Wearable Sensor and Notification Platform.* **Maurer, U., et al.** Cambridge, MA : IEEE, 2006. International Workshop on Wearable and Implantable Body Sensor Networks. 9017513.

25. How to Get What You Want. *Simple Fabric Pressure Sensors.* [Online] http://www.kobakant.at/DIY/?p=232.

26. *Introducing Wearable Force Sensors in Martial Arts.* **Chi, E.H.** s.l. : IEEE Computer Society, 2005. IEEE Pervasive Computing. 8570177.

27. *A wearable wireless sensor platform for interactive dance performances.* **Park, C., Chou, P.H. and Sun, Y.** s.l. : IEEE, 2006. Fourth Annual IEEE International Conference on Pervasive Computing and Communications. 9016907.

28. *WearAir: Expressive T-shirts for Air Quality Sensing.* **Sunyoung Kim, Eric Paulos, and Mark D. Gross.** New York, NY, USA : ACM, 2010. In Proceedings of the fourth international conference on Tangible, embedded, and embodied interaction (TEI '10). 295-296. DOI=10.1145/1709886.1709949 http://doi.acm.org/10.1145/1709886.1709949.

29. *In Proceedings of the 27th international Conference Extended Abstracts on Human Factors in Computing Systems (Boston, MA, USA, April 04 - 09, 2009). CHI '09. ACM, New York, NY, 2439-2448. DOI= http://doi.acm.org/10.1145/1520340.1520346.* **Chung, K., Chiu, C., Xiao, X., and Chi, P. 2009. Stress outsourced: a haptic social network via crowdsourcing.**

30. *The LilyPad Arduino: using computational textiles to investigate engagement, aesthetics, and diversity in computer science education.* **Leah Buechley, Mike Eisenberg, Jaime Catchen, and Ali Crockett.** New York, NY, USA : ACM, 2008. In Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems (CHI '08). 423-432. DOI=10.1145/1357054.1357123 http://doi.acm.org/10.1145/1357054.1357123.

31. *Iossifova, M. and Kim, Y. 2004. HearWear: the fashion of environmental noise display.* In ACM SIGGRAPH 2004 Emerging Technologies (Los Angeles, California, August 08 - 12, 2004). H. Elliott-Famularo, Ed. SIGGRAPH '04. ACM, New York, NY, 7. DOI= http://doi.acm.org/10.1145/1186155.1186163 : s.n.

32. *Creating a magic lightning experience with textiles.* **Harold, P.** 2006, Vol. Password: Philips research technology, pp. 6-11. Issue 28.

33. *Reach: dynamic textile patterns for communication and social expression.* **Worbin, Margot Jacobs and Linda.** New York, NY, USA : ACM, 2005. In CHI '05 extended abstracts on Human factors in computing systems (CHI '05). 1493-1496. DOI=10.1145/1056808.1056949 http://doi.acm.org/10.1145/1056808.1056949.

34. *Textile displays: using textiles to investigate computational technology as design material.* **Lars Hallnas, Linda Melin, and Johan Redstrom.** New York, NY, USA : ACM, 2002. In Proceedings of the second Nordic conference on Human-computer interaction (NordiCHI '02). 157-166. DOI=10.1145/572020.572039 http://doi.acm.org/10.1145/572020.572039.

35. *Kukkia and Vilkas: kinetic electronic garments.* **Berzowska, J. and Coelho, M.** s.l. : IEEE, 2005. Proceedings. Ninth IEEE International Symposium on Wearable Computers. 8749164.

36. *The LilyPad Arduino: Toward Wearable Engineering for Everyone.* **Eisenberg, Leah Buechley and Michael.** s.l. : IEEE Computer Society, 2008. IEEE Pervasive Computing. 9919830.

37. *Ubiquitous drums: a tangible, wearable musical interface.* **Gross, Boris Smus and Mark D.** New York, NY, USA : ACM, 2010. In Proceedings of the 28th of the international conference extended abstracts on Human factors in computing systems (CHI EA '10). 4009-4014. DOI=10.1145/1753846.1754094 http://doi.acm.org/10.1145/1753846.1754094.

38. **Chris, Well.** *Engadget - Magazine which covers everything new in gadgets and consumer electronics.* [Online] [Cited:] http://www.engadget.com/2010/03/30/t-shirt-modded-to-let-you-know-when-you-have-new-emails/.

39. Seeeduino Depot. *Seeeduino Family.* [Online] http://www.seeedstudio.com/depot/seeeduino-film-p-689.html?cPath=79_80.

40. Three Fabric Buttons. [Online] http://www.instructables.com/id/Three-Fabric-Buttons/.

41. Stroke Sensor. [Online] http://www.instructables.com/id/Stroke-Sensor/.

42. *Sensor Workshop ITP.* [Online] http://itp.nyu.edu/physcomp/sensors/Reports/ZipperSensor.

43. *Using a Wizard of Oz study to inform the design of SenToy.* **Gerd Andersson, Kristina Hook, Dário Mourão, Ana Paiva, and Marco Costa.** New York, NY, USA : ACM , 2002. In Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques (DIS '02). DOI=10.1145/778712.778762 http://doi.acm.org/10.1145/778712.778762.

APPENDICES

Processing Code

Appendix A – WearableMP3 Class: Main Code

```
/*
    Main Class

    Receives values from slave microcontroller and detects if the following
operations were selected:
        - Play, Pause, Stop music
        - Move Forward, Backward on Album's music
        - Change Volume
        - Selected a Mode

    by Cátia Sousa
  */


  import processing.serial.*;

  //Variables
  Serial mPort;
  Stablizer stab_volume;
  Stablizer stab_foward;
  Stablizer stab_backward;
  boolean firstContact = false;
  int sensors[];
  int albumPosition[] = {0,0,0,0};
  int albumPressed = 0;
  int position = 0;
  int album = 0;
  int iIndex = 0;
  int indexBack = 0;
  int albumRight = 0;
  float velocity = 0;
  float velocityBack = 0;
```

```
long aVal = 0;
long tVal = 0;
long tBack = 0;
float mResult;
float mMap;
float fowardMap = 0;
float  backwardMap = 0;

Album[] albumMusic;
Thread[] talbumMusic;

void setup() {
  size(1200,480);
  stab_volume =  new Stablizer(2,200);
  stab_foward = new Stablizer(2,200);
  stab_backward = new Stablizer(2, 200);
  mPort = new Serial(this, Serial.list()[2], 9600);
  mPort.bufferUntil('\n');
  println(mPort.list());
  albumMusic = new Album[5];
  talbumMusic = new Thread[5];

  //Threads Initialization
  albumMusic[1] = new Album(this, "Gorillaz");
  talbumMusic[1] = new Thread(albumMusic[1]);
  talbumMusic[1].start();
  albumMusic[2] = new Album(this, "Party");
  talbumMusic[2] = new Thread(albumMusic[2]);
  talbumMusic[2].start();
  albumMusic[3] = new Album(this, "Coldplay");
  talbumMusic[3] = new Thread(albumMusic[3]);
  talbumMusic[3].start();
  albumMusic[4] = new Album(this, "Bach");
  talbumMusic[4] = new Thread(albumMusic[4]);
  talbumMusic[4].start();
}

void draw() {
}

void serialEvent(Serial myPort) {
  String mString = myPort.readStringUntil('\n');
  if (mString != null) {
    mString = trim(mString);

    //Establishing handshake with Slave microcontroller
    if (firstContact == false) {
      if (mString.equals("hello")) {
```

```
      myPort.clear();
      firstContact = true;
      myPort.write('A');
      delay(200);
    }
  }
  else {
    sensors = int(split(mString, ','));  //Get the slave sensors values
    for(int i = 0; i < sensors.length; i++){
      println(sensors[i]);
    }
    println();
    println();

    //Putting the album positions in an Array
    int j = 4;
    for(int i = 0; i < albumPosition.length; i++){
      albumPosition[i] = sensors[j];
      j++;
    }

    albumPressed = sensors[3];

    if(albumPressed!=0){
      album = albumPosition[albumPressed-1];
    }

    //Console Output to check the Albums Order
    for(int i=0; i < albumPosition.length; i++){
      println("");
      println(albumPosition[i]);
    }

    //Mode Selection - Identify which mode was selected
    if(album != 0){
      if(sensors[0] == 1){
        albumMusic[album].repeat();
        println("Repeat Mode");
      }else if(sensors[1] == 1){
        albumMusic[album].randomize();
        println("Random Mode");
      }else if(sensors[0] == 0 && sensors[1] == 0){
        println("Normal Mode");
      }

      //Zip - set new volume
      aVal = stab_volume.stablize(sensors[2]);
      mResult = constrain(map(aVal, 2, 15, 400, 1000), 400, 1000);
```

```
          mMap = constrain(map(mResult,400,1000, 6, -30), -30, 6);
          println("Volume: ");
          println(mMap);
          albumMusic[album].volume(int(mMap));




          //Play - If play capacitive sensor was activated then music is played
          if(sensors[8] > 200){
            albumMusic[album].play();
            println("play");
          }

          //Pause - If pause capacitive sensor was activated then music is paused
          if(sensors[9] > 200){
            albumMusic[album].pause();
            println("pause");
          }

          //Stop - If stop capacitive sensor was activated then music is stopped
          if(sensors[10] > 200){
            albumMusic[album].stopAlbum();
            println("stop");
          }

          //Forward - If the two forward buttons are pressed and the forward stretch
sensor is stretched; goes to the next song
          tVal = stab_foward.stablize(sensors[16]);
          fowardMap = constrain(map(tVal, 470, 600, 480, 1000), 480, 1000);
            if(sensors[13] == 1 && sensors[14] == 1){
            if(iIndex == 5){
              albumMusic[album].forward();
              println("Forward");
              iIndex = 0;
            }
            else{
              iIndex++;
            }
          }

          //Backward - If the two backward buttons are pressed and the backward
stretch sensor is stretched; goes to the previous song
          tBack = stab_backward.stablize(sensors[15]);
          backwardMap = constrain(map(tBack, 470, 600, 480, 1000), 480, 1000);
            if(sensors[11] == 1 && sensors[12] == 1){
            if(indexBack == 5){
              albumMusic[album].backward();
```

```
              println("Backward");
              indexBack = 0;
            }
            else{
              indexBack++;
            }
          }
        }
      }
    }
    myPort.write('A');
    delay(200);
 }
```

Appendix B – Album Class Code

```
/*
Album Class
Responsible for the following operations in an album's music:
  -going forward and backward on album’s music
  -repeat, random or normal mode

Responsible to create an Music Object and play, pause, stop and change volume.

by Cátia Sousa
*/



import ddf.minim.*;
import ddf.minim.signals.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.Controller.*;

public class Album extends Thread{

//Variables
String name = "";
String artist ="";
String genre = "";
boolean isRepeat = false;
boolean isRandom = false;
boolean isPlaying = false;
boolean isReqStop= false;
boolean isReqForward= false;
```

```
boolean isReqBackward= false;
int myGain=6;
int nMusics=0;
int currentlyPlaying = 0;
Music[] song;

PApplet app=null;

public Album(PApplet _app, String _name){
  name = _name;
  song = new Music[10];
  app=_app;

  //Load songs from the file
  for(int i = 1; i <=9; i++){
    this.insertMusic(app,"0"+i+_name+".mp3");
  }
  this.insertMusic(app,"10"+_name+".mp3");
}

public void insertMusic(PApplet _app,String _name){
  song[nMusics] = new Music(_app,_name);
  nMusics++;
}

//Play – Plays album’s music according to mode selection that was chosen
void play(){
  isPlaying = true;
}
private void iPlay(){
  if(isRandom && (!song[currentlyPlaying].isPlaying())){ //Random Mode
    if(!song[currentlyPlaying].isPlaying()){
      song[currentlyPlaying].stop();
    }
    int result = 0;
    while(((result = int(random(nMusics))) == currentlyPlaying) ){
    }
    currentlyPlaying = result;
    if(isPlaying){
      song[currentlyPlaying].play();
    }
  }
  else if(isRepeat && (!song[currentlyPlaying].isPlaying())){ //Repeat Mode
    if(currentlyPlaying == nMusics-1){
```

```
      currentlyPlaying = 0;
    }
    else{
      currentlyPlaying++;
    }
    if(isPlaying){
      song[currentlyPlaying].play();
    }
  }
  else  if(isPlaying && (!song[currentlyPlaying].isPlaying()) && (currentlyPlaying <
nMusics)){  //Normal Mode
    if(currentlyPlaying==0){
      song[currentlyPlaying].play();
    }
    else{
      currentlyPlaying++;
      song[currentlyPlaying].play();
    }
  }
}

//Stop – Stop music
void stopAlbum(){
  isReqStop = true;
}
private void iStop(){
  if(isReqStop){
    song[currentlyPlaying].stop();
    if(currentlyPlaying!=0)
      currentlyPlaying--;
      isReqStop = false;
      isPlaying = false;
  }
}

//Forward – Goes forward on album's music
void forward(){
  isReqForward = true;
}
private void iForward(){
  if(isRandom && isReqForward){
    song[currentlyPlaying].stop();
    int result = 0;
    while(((result = int(random(nMusics))) == currentlyPlaying) ){
```

```
    }
    currentlyPlaying = result;
    if(isPlaying){
      song[currentlyPlaying].play();
    }
    isReqForward = false;
  }
  else if(isRepeat && isReqForward){
    song[currentlyPlaying].stop();
    int result = 0;
    currentlyPlaying++;
    if(currentlyPlaying == nMusics){
      currentlyPlaying = 0;
    }
    if(isPlaying){
      song[currentlyPlaying].play();
    }
    isReqForward=false;
  }
  else{
    if(isReqForward && (currentlyPlaying < nMusics-1)){
      song[currentlyPlaying].stop();
      currentlyPlaying++;
      isReqForward = false;
      if(isPlaying){
        song[currentlyPlaying].play();
      }
    }
    else{
      isReqForward = false;
    }
  }
}

//Backward – Goes Backward on album’s music
void backward(){
  isReqBackward = true;
}
private void iBackward(){
  if(isRandom && isReqBackward){
    song[currentlyPlaying].stop();
    int result = 0;
    while(((result = int(random(nMusics))) == currentlyPlaying) ){
    }
```

```
    currentlyPlaying = result;
    if(isPlaying){
      song[currentlyPlaying].play();
    }
    isReqBackward = false;
  }
  else if(isRepeat && isReqBackward){
    song[currentlyPlaying].stop();
    int result = 0;
    currentlyPlaying--;
    if(currentlyPlaying == -1){
      currentlyPlaying = nMusics-1;
    }
    if(isPlaying){
      song[currentlyPlaying].play();
    }
    isReqBackward=false;
  }
  else{
    if(isReqBackward && (currentlyPlaying > 0)){
      song[currentlyPlaying].stop();
      currentlyPlaying --;
      isReqBackward = false;
      if(isPlaying){
        song[currentlyPlaying].play();
      }
    }
    else{
      isReqBackward = false;
    }
  }
}

//Randomize – Set random mode true
void randomize(){
  isRandom = true;
}

void repeat(){
  isRepeat = true;
}

//Normal – Set normal mode true
void none(){
```

```
  isRandom = false;
  isRepeat = false;
}

//Volume - Set new volume
void volume(int _gain){
  myGain = _gain;
}
void iVolume(){
  song[currentlyPlaying].volume(myGain);
}

//Pause - Pause Music
void pause(){
  isPlaying = false;
  song[currentlyPlaying].pause();
  if(currentlyPlaying!=0)
    currentlyPlaying--;
}

public void run(){
  while(true){
    iForward();
    yield();
    iBackward();
    yield();
    iStop();
    yield();
    iPlay();
    yield();
    iVolume();
    yield();
    delay(10);
  }
 }
}
```

Appendix C – Music Class Code

```
/*
 Music Class

 Responsible for the following actions in music:
 - Play, Pause, Stop
 - Set Volume

 Uses the Minim Library.

 by Cátia Sousa
 */

import ddf.minim.*;
import ddf.minim.signals.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.Controller.*;

class Music{

  //Variables
  String name;
  AudioPlayer music;
  Voice myVoice;
  PApplet app;

  public Music(PApplet _app, String _name){

    // load the music object
    AudioPlayer musicData;
    Minim musicX;
    musicX = new Minim(_app);
    musicData = musicX.loadFile(_name);

    // load the voice over read object
    myVoice = new Voice(_app,_name);

    app = _app;
    name = _name;
    music = musicData;
  }
```

```
  public void play(){
    myVoice.play();
    music.play();
  }
  public void pause(){
    music.pause();
  }
  public void stop(){
    music.pause();
    music.rewind();
  }
  public boolean isPlaying(){
    return music.isPlaying();
  }

  public void volume(int _gain){
    music.setGain(_gain);
    myVoice.volume(_gain);
  }
}
```

Appendix D – Stablizer Class Code

```
/*
Stablizer Class

Responsible for normalize the values of the sensors, using the average between them.
Also discards values that are far away.
For instance if the new object of this class is created like this: Stablizer stab_volume =  new Stablizer(4,200)
It means that will do an average of 4 values. If each one of them differ 200 from the next one, it means that will not be incorporated in the average.
For instance, receiving [4,6,400,20,10,..], the value 400 will be discarded, only [4,6,20,10,..] will matter for the average

by Cátia Sousa
*/


class Stablizer{

 //Variables
 int[] values;
 int arrayPointer  = 0;
 int avgSampleSize = 0;
```

```
 int maxVariation  = 0;

 int valuesInside = 0;

 Stablizer(int _averageSample, int _maxVariation){
   avgSampleSize = _averageSample;
   maxVariation = _maxVariation;
   values =new int[_averageSample];
 }

 long stablize(int x){
   // reject values that do not respect the variation rules - _maxVariation
   if(valuesInside != 0){  // if it is not the first value entered
     int beforeArrayPointer = (arrayPointer == 0) ? valuesInside-1 : arrayPointer-1;
     if(x      <      values[beforeArrayPointer]-maxVariation      ||      x      >
values[beforeArrayPointer]+maxVariation){
       x = values[beforeArrayPointer];
     }
   }

   // stablizes using average
   long sum = 0;
   values[arrayPointer] = x;   // put the value inside the array
   arrayPointer ++;   // move the pointer 1 point forward
   if(arrayPointer>=avgSampleSize) arrayPointer = 0; // if it reached the max sample
size, reset the pointer
   if(valuesInside<avgSampleSize) valuesInside++;  // keep in track if the array has
all the values needed
   for(int i=0;i<valuesInside;i++){
     sum += values[i];
   }
   return sum/valuesInside;
 }
```

Appendix E – Voice Class Code

```
/*
Voice Class

Responsible to play and change volume of the music’s names.
Uses the Minim Library.

by Cátia Sousa
*/
```

```
import ddf.minim.*;
import ddf.minim.signals.*;
import ddf.minim.analysis.*;
import ddf.minim.effects.*;
import ddf.minim.Controller.*;

class Voice{

//Variables
String name;
AudioPlayer voice;
PApplet app;

public Voice(PApplet _app, String _name){

  AudioPlayer musicData;
  Minim musicX;
  musicX = new Minim(_app);
  //Load the mp3 music name file
  String voiceFileName = _name.substring(0,2)+"v"+_name.substring(2,_name.length());
  musicData = musicX.loadFile(voiceFileName);

  app = _app;
  name = _name;
  voice = musicData;
 }

 public void play(){
  voice.rewind();
  voice.play();
 }

 public void volume(int _gain){
  voice.setGain(_gain);
 }
}
```

Appendix F – Arduino Master Code

```
/*
    Master, Slave Connection

 Reads the master sensor values and send to the slave. These sensors are: Zip sensor, Mode Selection - Repeat mode sensor, Random mode sensor and Normal mode sensor values, Album sensor

 by Cátia Sousa
 */

#include <Wire.h> //Library responsible to establish connection between Master and Slave

#define ADDRESS 42

//Variables
int analogPin[4] = {0,0,0,0};
int digitalPin[13] = {0,0,0,0,0,0,0,0,0,0,0,0,0};
int analogValue1, analogValue2, analogValue3 = 0;
int albumSelected = 0;

 //Variables related with albums
int digitalBig[] = {8,5,6,7};
int digitalSmall[] = {4,3,2,13};
int albumPosition[] = {0,0,0,0};
int previousAlbumInput[] = {0,0,0,0};

void setup(){
  Wire.begin();
  Serial.begin(9600);
}

void loop(){
  //Read Zip sensor, Mode Selection - Repeat mode sensor, Mode Selection - Random mode sensor and Mode Selection - Normal mode sensor values
  for(int i  = 0; i <= 3; i++){
    analogPin[i] = analogRead(i);
  }
```

```
  //Identify if Repeat mode was selected and puts the variable to send to the slave
at 1 otherwise puts at 0
  if(analogPin[1] >= 1 && analogPin[1] <= 5){
    analogValue1 = 1;
  }
  else{
    analogValue1 = 0;
  }
  int temp = analogValue1;

  //Identify if Random mode was selected and puts the variable to send to the slave
at 1 otherwise puts at 0
  if(analogPin[3] >= 6 && analogPin[3] <= 9){
    analogValue3 = 1;
  }
  else{
    analogValue3 = 0;
  }


  //Read all ports related with album's identification and selection
  for(int i = 1; i <= 13; i++){
    digitalPin[i] = digitalRead(i);
  }


  //Identify which position was selected
  for(int i = 9; i <= 12; i++){
    if(digitalPin[i] == 1){
      switch(i){
      case 9:
        albumSelected = 1; //Position 1
        break;
      case 10:
        albumSelected = 4; //Position 4
        break;
      case 11:
        albumSelected = 3; //Position 3
        break;
      case 12:
        albumSelected = 2; //Position 2
        break;
      default:
        albumSelected = 0; //None Selected
```

```
      }
    }
  }
  Serial.println(temp);
  Serial.println(analogValue3);
  Serial.println(analogPin[2]);
  Serial.println(albumSelected);

  //Send Data to Slave
  Wire.beginTransmission(ADDRESS); //Begin transition
  Wire.send(temp); //Repeat value
  Wire.send(analogValue3); //Random value
  Wire.send(analogPin[2]); //Zip value
  Wire.send(albumSelected); //Position selected
  albumPositionFind(); // Send which album is in which position
  Wire.endTransmission(); //Ends transition
  delay(300);
}

boolean existsInArray(int arr[], int number){
  for(int i = 0 ; i < 4  ; i++){
    if(arr[i]==number) return true;
  }
  return false;
}

void albumPositionFind(){
  // check for new albums
  for(int i = 0 ; i < 4  ; i++){ //Album Position
    if(previousAlbumInput[i] != digitalPin[digitalBig[i]]){  // if there is a change
      // switch state
      if(digitalPin[digitalBig[i]]==1){
        // there is a new album
        previousAlbumInput[i] = 1;
        // search the new album
        for(int j = 0 ; j < 4 ; j++){
          if(digitalPin[digitalSmall[j]]==1){
            if(!existsInArray(albumPosition,j+1)){
              albumPosition[i]=j+1;
            }
          }
        }
      }
      else{
```

```
        // an album has been removed
        previousAlbumInput[i] = 0;
        albumPosition[i] = 0;
      }
    }
  }

  //Send which album is in each position on the pullover, for instance: [1,4,3,2]
  //So, album 1 is in position 0; album 4 in position 1; album 3 in position 2; album
4 in position 3
  for(int i = 0 ; i < 4; i++){
    Serial.println(albumPosition[i]);
    Wire.send(albumPosition[i]);
  }
}
```

Appendix G – Arduino Slave Code

```
/*
  Master, Slave, Processing Connection

  Receives the master values, add new sensor values and send all sensors data on the
pullover to Processing

  by Cátia Sousa
*/



 //Library responsible to establish connection between Master and Slave
#include <Wire.h>
//Library responsible to turn digital pins in capacitance sensors
#include <CapSense.h>

#define ADDRESS 42

//Variables
CapSense   cs_13_12 = CapSense(13,12);
CapSense   cs_11_10 = CapSense(11,10);
CapSense   cs_6_5 = CapSense(6,5);

int digitalPinMaster[13] = {0,0,0,0,0,0,0,0,0,0,0,0,0};
int analogPinMaster[4] = {0,0,0,0};
int analogResultFoward, analogResultBackward = 0;

int i=0;
```

```
int val = 0;
int masterValues, repeat, randomize, zip, pressed, firstPosition, secondPosition,
thirdPosition, fourthPosition = 0;


void setup() {
  Serial.begin(9600);
  Wire.begin(ADDRESS);
  Wire.onReceive(receiveEvent);
  cs_13_12.set_CS_AutocaL_Millis(0xFFFFFFFF);
  cs_11_10.set_CS_AutocaL_Millis(0xFFFFFFFF);
  cs_6_5.set_CS_AutocaL_Millis(0xFFFFFFFF);
  establishContact();
}

void loop() {
  int play =  cs_13_12.capSense(5);
  int pause =  cs_11_10.capSense(5);
  int sttop =  cs_6_5.capSense(5);

  //Get values of Play, Pause, Stop and Forward and Backward Buttons
  for(int j = 1; j <= 13; j++){
    digitalPinMaster[j] = digitalRead(j);
  }

  //Get values of Stretch sensor: Forward and Backward
  for(int j = 0; j <= 3; j++){
    analogPinMaster[j] = analogRead(j);
  }

  analogResultFoward = analogPinMaster[3] - analogPinMaster[2];
  analogResultBackward = analogPinMaster[1] - analogPinMaster[0];

  //Send Values to Processing after having a handshake between them.
  //Waits to send new data set until processing sends a new bit.
  if(Serial.available() > 0){
    val = Serial.read();

    Serial.print(repeat,DEC); //Repeat
    Serial.print(",");
    Serial.print(randomize,DEC); //Random
    Serial.print(",");
    Serial.print(zip,DEC); //Zip
    Serial.print(",");
```

```
    Serial.print(pressed,DEC); //Album Pressed
    Serial.print(",");
    Serial.print(firstPosition,DEC); //Album in position 1
    Serial.print(",");
    Serial.print(secondPosition,DEC); //Album in position 2
    Serial.print(",");
    Serial.print(thirdPosition,DEC); //Album in position 3
    Serial.print(",");
    Serial.print(fourthPosition,DEC); //Album in position 4
    Serial.print(",");
    Serial.print(play,DEC); //Play value
    Serial.print(",");
    Serial.print(pause,DEC);  //Pause values
    Serial.print(",");
    Serial.print(sttop,DEC);  //Stop value
    Serial.print(",");
    Serial.print(digitalPinMaster[4],DEC); //Backward Button1
    Serial.print(",");
    Serial.print(digitalPinMaster[3],DEC); //Backward Button2
    Serial.print(",");
    Serial.print(digitalPinMaster[2],DEC); //Forward Button1
    Serial.print(",");
    Serial.print(digitalPinMaster[7],DEC); //Forward Button2
    Serial.print(",");
    Serial.print(analogResultFoward,DEC); //Forward Value
    Serial.print(",");
    Serial.println(analogResultBackward,DEC); //Backward Value
  }
}

//This function is responsible to read the values that the Master sends, and save them in variables
void receiveEvent(int howMany)
{
  while(Wire.available())
  {
    masterValues = Wire.receive();

    if(i==0){
      repeat = masterValues; //repeat
      i++;
    }
    else if(i==1){
      randomize = masterValues; //Randomize
```

```
      i++;
    }
    else if(i==2){
      zip = masterValues; //Zip
      i++;
    }
    else if(i==3){
      pressed = masterValues; //Album Pressed
      i++;
    }
    else if(i==4){
      firstPosition = masterValues; //Album in position 1
      i++;
    }
    else if(i==5){
      secondPosition = masterValues; //Album in position 2
      i++;
    }
    else if(i==6){
      thirdPosition = masterValues; //Album in position 3
      i++;
    }
    else if(i==7){
      fourthPosition = masterValues; //Album in position 4
      i=0;
    }
  }
}

//This function is responsible to establish connection with Processing.
//It works as a handshake.
//Slave will send the word "hello" until the processing sends a char back.
//When this happen they establish connection, and this function is no
//more executed.
void establishContact() {
  while (Serial.available() <= 0) {
    Serial.println("hello");
    delay(300);
  }
}
```

Appendix H – User Study Questions

The personal questions made were:

1. What is your age?
    a. 25 or less
    b. 26 – 40
    c. 41 – 55
    d. 56 or more
2. What is your gender?
    a. Feminine
    b. Masculine
3. What is your nationality?
4. What your English knowledge?
    a. Elementary
    b. Good
    c. Excellent
5. Do you use mp3 players to listen to music?
6. Which frequency do you use it?
    a. Daily
    b. Weekly
    c. Monthly
    d. Other: ____________

The questions done after the user perform the tasks were:

1. How would you compare this system with other mp3 players?
2. Which was your favorite feature? And least favorite? Why?
3. Do you think that this would be good for any other specific application?
4. How you would react with the fact that other people can know what kind of music you listen to?