INTELLIGENT FURNITURE

Reducing negative health effects of prolonged sitting

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Bachelor's thesis

Electrical and Automation Engineering

Spring 2019

Aleksi Hakkarainen

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

Electrical and Automation Engineering
Valkeakoski

Author Aleksi Hakkarainen **Year** 2019

Title Intelligent furniture

Supervisors Juha Sarkula, Antti Aimo

TIIVISTELMÄ

Tämä opinnäytetyö pyrki selvittämään millä tavoin ja kuinka tehokkaasti älykäs tuoli voi auttaa vähentämään liiallisesta istumisesta koituvia terveyshaittoja ja parantamaan käyttäjän istuma-asentoa.

Opinnäytetyö koostuu kolmesta pääosasta. Ensimmäisessä osassa käydään läpi ja tutkitaan yleisiä istumiseen liittyviä terveysongelmia ja niiden ehkäisemistä sekä kirjallisuuden avulla että haastattelemalla työfysioterapeutti Auli Kaitaa. Tämän jälkeen toisessa osassa suunnitellaan, rakennetaan ja ohjelmoidaan tuolin prototyyppi ensimmäisen osan tutkimukseen pohjaavan tiedon perusteella pyrkimyksenä vähentämään istumisesta koituvia terveyshaittoja ja parantaa käyttäjän istuma-asentoa. Kolmas osa tutkii testihenkilöiden avulla onko tuolilla vaikutusta heidän istumatapoihinsa kyselylomakkeella ja tuolin tallentamilla tiedoilla heidän istumiseen liittyvistä tavoista ja tuolin kommunikoinnista.

Kolmannen osion tutkimus osoittaa että tuolilla, joka tekee päätöksiä istujasta saamallaan tiedolla ja värisyttämällä istuinta koettaa muuttaa hänen istumatapojaan paremmaksi on potentiaalisesti myönteinen vaikutus käyttäjän istumatapoihin. Luotettavampien tuloksien saamiseksi kattavampi tutkimus suuremmalla tuoli ja koehenkilö määrällä on tarpeen. Myös tuolin seuraavan version tulee kehittyä ergonomiapainotteisemmaksi ja älykkäämmäksi.

Avainsanat Älykäs kaluste, Ergonomia, Terveys, Raspberry Pi.

Sivut 59 sivua, joista liitteitä 22 sivua

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Electrical and Automation Engineering
Valkeakoski

Author Aleksi Hakkarainen **Year** 2019

Subject Intelligent furniture

Supervisors Juha Sarkula, Antti Aimo

ABSTRACT

The aim of this thesis was to learn by researching and examining whether an intelligent chair could reduce negative health effects of prolonged sitting and improve the user's sitting position. Furthermore, the thesis aimed to find the most ideal and effective implementation of such chair.

The thesis has three main parts to it: First the thesis researches common health problems regarding prolonged sitting and how it is possible to prevent these. This part of the research was done by interviewing a occupational physiotherapist Auli Kaita, by using literature in the field and online sources.

The second part focuses on designing, building and programming a chair based on this information. Lastly, the third part focuses on researching whether the chair has an impact into the sitting habits of the test users. This was done by using a questionnaire and the logs that the chair recorded.

The data acquired during the third part indicated that the chair, that makes decisions based on the sitting habits of the test users and then communicates with them by vibrating the seat with different patterns to improve their sitting habits, can potentially have a positive impact on the sitting habits and sitting position of the user. More thorough testing is still needed with greater number of test users and chairs for more reliable results. Next version of the chair should be more ergonomic oriented and intelligent with the use of artificial neural networks in detecting reliably greater variety of different sitting positions.

Keywords Intelligent furniture, Ergonomics, Health, Raspberry Pi.

Pages 59 pages including appendices 22 pages

CONTENTS

Appendices

1 INTRODUCTION

My previous degree was from the field of Industrial Furniture Design. During design studies and after that while working in the field for several years an extensive collection of my own designs was accumulated, mainly chairs. One of the main motivations when starting these current studies in Electrical and Automation Engineering was to look for ways to combine these two fields. During the Home Automation course in 2017 I was introduced to Raspberry Pi and its multiple applications, which sparked an interest in me to start developing intelligence for furniture especially for chairs.

The main idea with the chair prototype developed during the Home Automation course 2017 was that it would integrate into home- or work place automation systems or act itself as a small hub for such system. The prototype was focused on power savings at work stations. The idea being that it would always power down or hibernate the work station when the user left. Further development ideas for the prototype also included collecting simple health data and sitting habits of the organisation's employees with the aim to prevent heath issues and gain cost savings in health care (Appendix 1).

Modern day life for most of us includes too much stationary sitting for what our bodies have become used to and developed for during evolution. I wanted to look into how an intelligent chair could help us with this problem. How can the chair communicate with the user? How can the chair aid us to sit less or in a healthier manner? How effective the chair is in impacting our sitting habits?

2 RESEARCH PROJECT

This chapter offers an overall view to the history and physiology of sitting and how the human body is reacting to prolonged sitting as well as the theoretical background for why it is important for people to sit less and how it could be done.

2.1 Evolution

Throughout our long evolution we have lived a physically active life. During the last two centuries industrialization especially in the western countries has led to a radically less physically demanding life style and increased the time spent sitting for a vast majority of the population. For more and more communities the chair has become a normal part of everyday live in work places as well as in homes. (Opsvik 2009, 7) During the stone age people lived life filled with physical activity, constant walking, hunting, picking up things from the ground and trees. As people invented farming physical activity was necessity as well in order to secure their livelihood. For thousands of years our bodies have developed to fulfil our need for them to be strong, fast, flexible and active. This is simply the life style into which our bodies have developed for and are designed to. In addition to this most of us have also had very physically active childhood, when we used to play and exercise naturally physical life. (Opsvik 2009, 12)

Later developments in societies globally have led into a situation where our physical activity has in longer evolutional perspective taken very sudden decline and our bodies have not had the time they need to adjust into these new circumstances. Physical tasks that we expect our bodies to perform are significantly less than thousands of years ago. To some people this change happened 5000 years ago to some 50 years ago. Even though it is possible to see these changes as a long historical process the fact still remains that 5000 years is a very short time period in human evolution. All these changes in our life style have happened in blink of an eye if you observe them from the evolution perspective. (Opsvik 2009, 13)

2.2 Ergonomics

The world in which we live is filled with technology. Most of our actions or tasks are performed with the help of some technical device or in an environment built by us using technology. When designing this world and the technology in it we need ergonomics. (Launis & Lehtelä 2011, 17.) Ergonomics means the studying of the interaction between human and environment or technological device with this information we can improve wellbeing at homes or at work places. With ergonomics we can improve work, tools and work environment to better suit for people's needs. (Launis, M. & Lehtelä, J. 2011, 19). Ergonomics is important in a work place and the lack of it can cause numerous issues for the employees and for the company. The amount of sick leaves can be reduced by taking better care of the employees. Poor ergonomics can also have negative effect to the quality of the work and performance. In the worst-case bad work ergonomics can damage the client relations and image of the company by causing defects or inefficiency. (Launis, M. & Lehtelä, J. 2011.)

In the design of my chair to reduce the negative health effects of prolonged sitting I need to consider ergonomics as well. Not just the comfort of sitting, but also how intrusive the device is with its communication. This was one of the reasons why the chair does not include any email and SMS notifications, since people generally have already enough interruptions during their day. Instead I decided to proceed with more subtle communication such as vibrating the seat with different patterns.

2.3 Interview with a physiotherapist

Physiotherapist Auli Kaita recommends that you should have consistent breaks while sitting for longer periods for instance during work day. You should not sit for longer than one hour continuously or if you already have existing back problems this time should not exceed half an hour. She also adds that it could be beneficial to have some kind remainder in the latter case, so that the pain would not be the first indicator to get up and have a break from sitting (interview 25.10.2018).

There is also alternative or complementing option to having breaks: to shift one's sitting position. You should not stay in the same position in between the breaks. Therefore, the device or chair should notify you to shift too and it should also be able to detect different sitting positions. There are positions that are less harmful than others, as an example Auli mentions sitting on the front edge of your chair with your back straight, this resembles to the sitting position on a saddle chair that is developed to lessen the stress caused to our bodies by sitting. This is a position that the device should be able to detect and allow user to stay in longer than instead sitting at the front edge of your chair and leaning heavily forward (interview 25.10.2018).

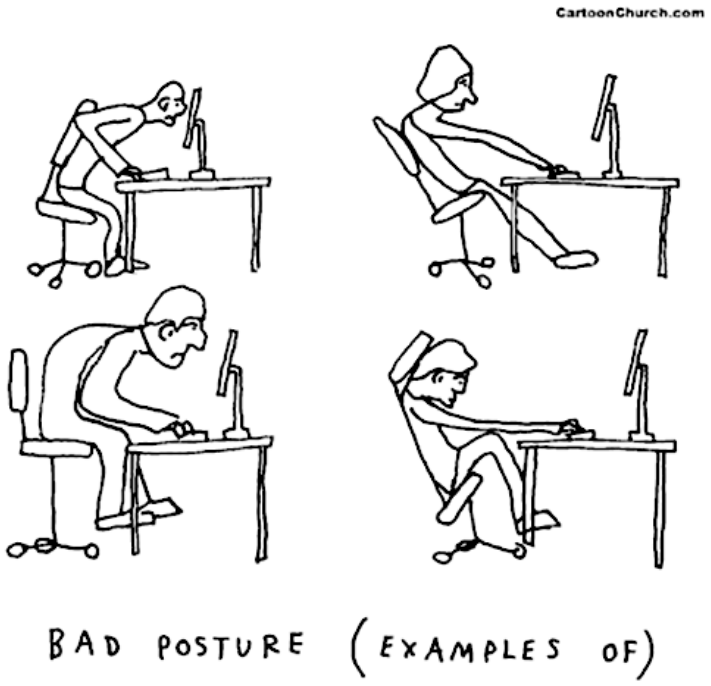


Figure 1. Picture Auli used to describe possible faulty sitting positions during our interview (Loomis, 2015)

2.4 **Sitting and short breaks**

Director of UKK institute Timo Vasankari has been researching and writing extensively about the negative health effects of sitting. The study shows that if a person spends a considerable amount of his or her day sitting even hard physical exercise after a long day of sitting will not be able to completely fix the damage caused by sitting. (Duodecimlehti 2016, 1) (Kauppalehti 2015) The research also shows that any exercise still counts if it is a break from sitting: even as little as standing up or shifting position will be enough stimulus for the body to reduce the negative health effects from sitting (duodecimlehti 2016, 5)

3 DESIGNING, BUILDING AND PROGRAMMING

At the beginning of the planning phase the idea was to build the device to provide user with comprehensive statistics about their sitting habits via cloud storage and notify them by email and SMS. As I had already built a primitive prototype of such a device on a previous home automation course in 2017. (Appendix 1) That prototype would have also provided a simple home automation system and acted as a hub or could integrate into already existing home automation system. After having interviewed physiotherapist Auli Kaita (interview 25.10.2018) about the physiology of sitting and some common issues regarding sitting, a decision was made to proceed with a more minimalistic approach trying to solve the issues people have with prolonged sitting. What Auli also suggested was my understanding as well: people generally get already enough interruptions and information during their day so I decided that the chair prototype would only communicate with the user by vibrating the seat and the program would be such that the user could either ignore it or chose to pay attention depending on the situation at work or at home.

3.1 Detecting movement

There are several different positions when it comes to sitting. Some better, some worse, but perhaps worst of all is being stationary without shifting one's position, or not standing up for a long time. This can be more harmful than the position itself. Detecting all these positions can be challenging; for instance, the distance from the persons back to the backrest can be measured with an ultrasonic or infrared proximity sensor, but they have reliability issues if person's clothes are not very fit for instance. Even though this prototype is a dedicated chair for this purpose, ultimately the goal was to build a device that should be easy to install onto any chair by just placing it on the seat, so all the sensors should be contained within the device itself.

This could be different if it would be a whole chair for this purpose, but I think it will be easier for people to start trying something like this if the cost and the effort is as little as possible. Although at this stage without any purpose-build hardware the device would be too thick with Raspberry Pi, Powerbank and vibration motor to be fitted conveniently on top of a normal chair. For these reasons I decided to embed this prototype technology into one of my chairs, to keep the system as compact as possible. While only using the load cells as sensors in the prototype, it is still capable of detecting load distribution, overall load, shift in load, sudden reduction of load and peak in load. From these I can already build a system that will help the person to reduce the negative health effects caused by too stationary or prolonged sitting.

3.2 **Hardware**

This chapter describes the overall system as well as the individual components. The intention is that the chair would maintain as much of its original look as possible and the testers could use the chair as they would use any other chair, except for having to charge it at least every other night.

3.2.1 Mona chair

Having already previous degree in Industrial Furniture Design and having accumulated extensive collection of my own designs, for this project I decided to use one of my chairs presented in Figure 2 before modifications and in Figure 3 after modifications. The name of the chair is Mona (Hakkarainen, 2017) and it has been exhibited at Finland's largest furniture fair Habitare's Ecodesign invitation exhibition in Helsinki in 2017 (Ecodesign, 2017), (Mainostoimisto valo Oy, 2017) in addition to this, Mona ranked top10 in 2018 when Ornamo was looking for sustainable Scandinavian design for UN's COP24 climate conference in Poland in Katowice. Mona was also featured on the front page of Coroflot in 2018, Coroflot is a large online community for designers (Hakkarainen, 2017). For this project one of the earlier prototypes of the chair was used that was left over. Installing this device requires modifications to the chair and I did not want to modify the exhibition quality ones.

Figure 2. Mona-chair before modifications (Hakkarainen 2017)

Figure 3. Mona-chair after modifications (Hakkarainen 2019)

3.2.2 Overview of the system

Figure 4 illustrates the overall layout of the system, later in this chapter more detailed schematics of parts of this system will be presented. As a power source generic 16000 mAh rechargeable Powerbank with two separate USB outlets was used.

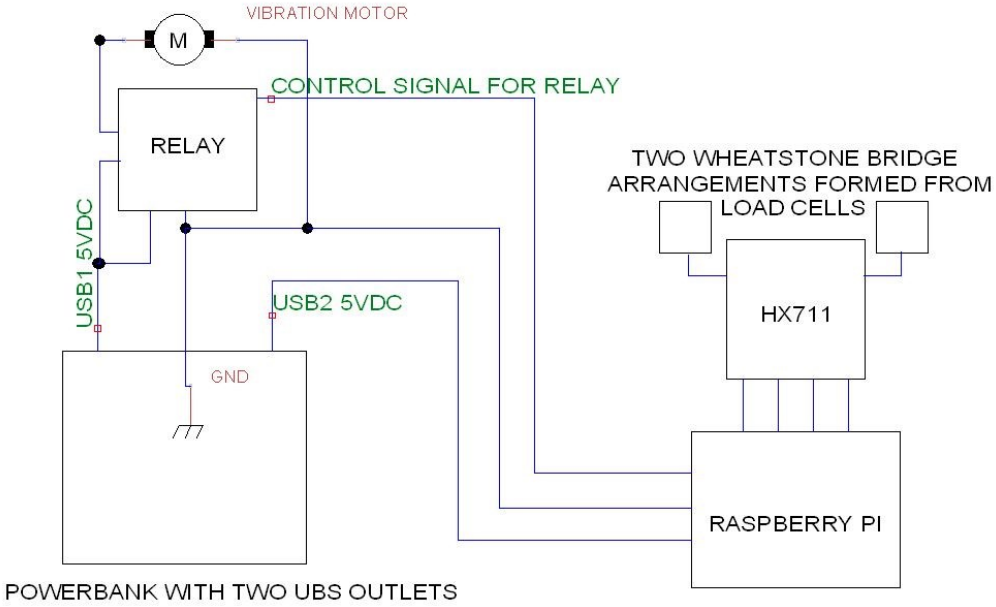


Figure 4. Overview of the system (Hakkarainen, 2019)

3.2.3 Raspberry Pi 3B

For prototyping Raspberry Pi offers a very good platform. Raspberry Pi as illustrated in Figure 5 is a one-chip computer with versatile connection options. Raspberry pi is a one chip computer with multiple digital input and output pins to communicate with the outside world with sensors and relays. In addition to that it has also USB ports, display port and one ethernet port. A 2.4GHz band Wi-Fi antenna is also included with additional hardware it is also possible to use different wireless communication frequencies and protocols such as 5 GHz band Wi-Fi and 433 MHz license free frequency band often used in home automation applications.

In this project Raspberry Pi is utilized to receive information from a load cells and then make decisions based on the information and time passed. Based on these factors Raspberry pi will communicate with the user by vibrating the seat with different patterns and keep logs of the sitting habits. One shortcoming of Raspberry Pi in comparison to Arduino (another popular one-chip computer) is the lack of analog inputs and outputs, this is possible to overcome by installing analog to digital converter for the load cells. (Raspberry Pi Foundation, 2019)

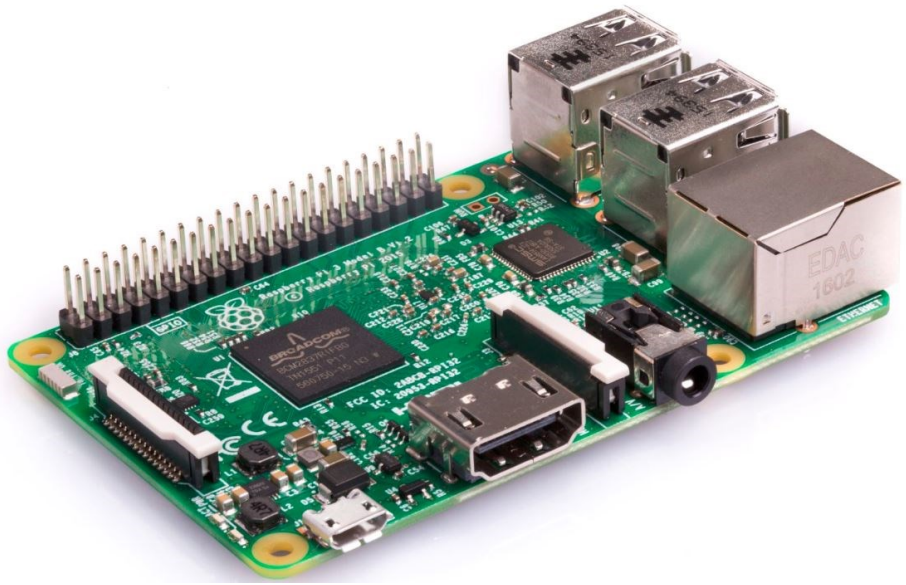


Figure 5. Raspberry Pi 3B (Raspberry Pi Foundation, n.d.)

3.2.4 HX711

HX711 is 24-bit Analog to Digital converter (ADC) that is especially designed to use with different load cells in Wheatstone Bridge arrangement. As the Figure 6 illustrates HX711 has multiplexer with two (INA+/INA- and INB+/INB-) input channels for the analog signal from the Wheatstone Bridges. From the multiplexer the signals travels to low-noise programmable gain amplifier (PGA). The first channel has two selectable options for gain 64 and 128. The other channel has fixed gain of 32 I will compensate for this later in the program to get equal readings from both Wheatstone bridges. Once the signals have been amplified, they arrive to ADC that converts them from analog voltage signal to digital signal for serial communication with Raspberry Pi via DOUT and PD_SCK pins. (Mouser 2019)

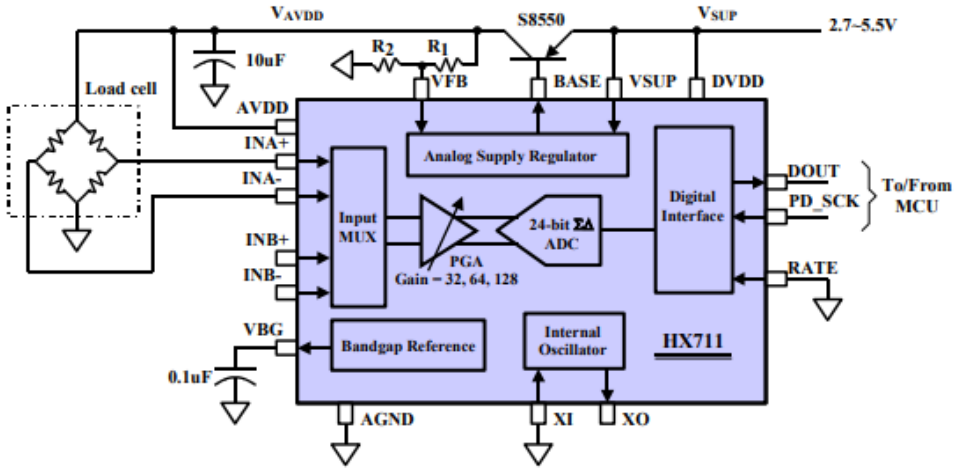


Figure 6. HX711 block diagram (Mouser Electronics, n.d.)

In this application only one HX711 was used with two separate Wheatstone Bridge arrangements. In the programming phase the HX711 was configured to use both A and B channels one for each Wheatstone Bridge arrangement.

3.2.5 Wheatstone bridge

Wheatstone bridge as illustrated in Figure 7 is named after its inventor Charles Wheatstone. It has multiple applications in electronics; it can be used to calibrate measuring instruments, since it can accurately measure resistance down in the milli-ohms range. With modern operational amplifiers such as HX711 that was used in this chair prototype the principles of Wheatstone bridge were utilised to measure changes in resistance very accurately. Each Load Cells has two strain resistors in them and when force is applied to them the strain resistors values change respectively, the other stretches and the other retracts. This can be measured as a voltage difference in the middle of the bridge between points C and D. In balanced Wheatstone Bridge the voltage difference is 0.

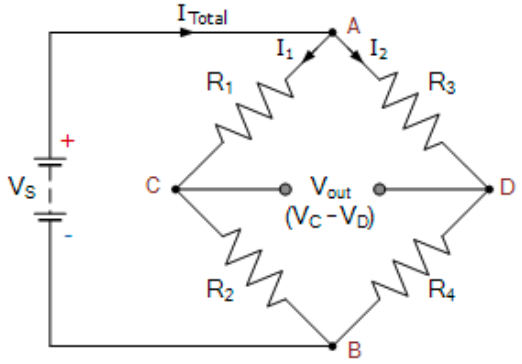


Figure 7. A Wheatstone bridge (Electronics Tutorials, n.d.)

Figure 8 presents an imbalanced Wheatstone Bridge and as a result there is voltage difference between C and D. It is possible to calculate the current going through each of the branches by using the following formula:

$I = V/R = 12V/(10\Omega + 20\Omega) = 0.4A$

Then based on the current and the second resistor on the left (R2) it is possible to calculate the voltages at points C and D as illustrated in the Figure 7 with imbalanced Wheatstone Bridge.

$V_{R2} = I*R_2 = 0.4A*20\Omega = 8$ volts
$V_{R4} = 0.4A*10\Omega = 4$ volts

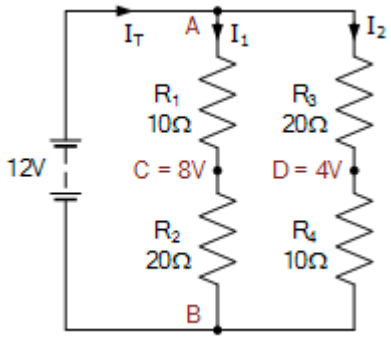


Figure 8. Imbalanced Wheatstone Bridge (Electronic Tutorials, n.d.)

3.2.6 Load cells

There are three main categories of load cells: Hydraulic, Pneumatic and electric Strain Gauge Load Cells. They all do the same task: translate pressure into an electric signal. This project uses Strain Gauge load cells.

Why load cells instead of a gyroscope or a motion sensor? In my opinion load cells in this particular case are the most reliable and simplest method to reliably measure and monitor the movements of a person. With load cells it is possible to detect whether the person has actually shifted his/her position; did the load decrease momentarily? Did the load shift from one side to another? Whereas gyroscope or motion sensor would register lighter movements as well. One option could be also to use a combination of different sensors. In the prototype there are four load cells detecting the load distribution, each load cell has two strain gauge resistors. The four load cells form two separate Wheatstone Bridge arrangements as illustrated in Figure 9. The other Wheatstone bridge arrangement is for left side of the chair and the other is for the right side of the chair. Figure 10 illustrates the installation of one of the load cells in the seat of the chair.

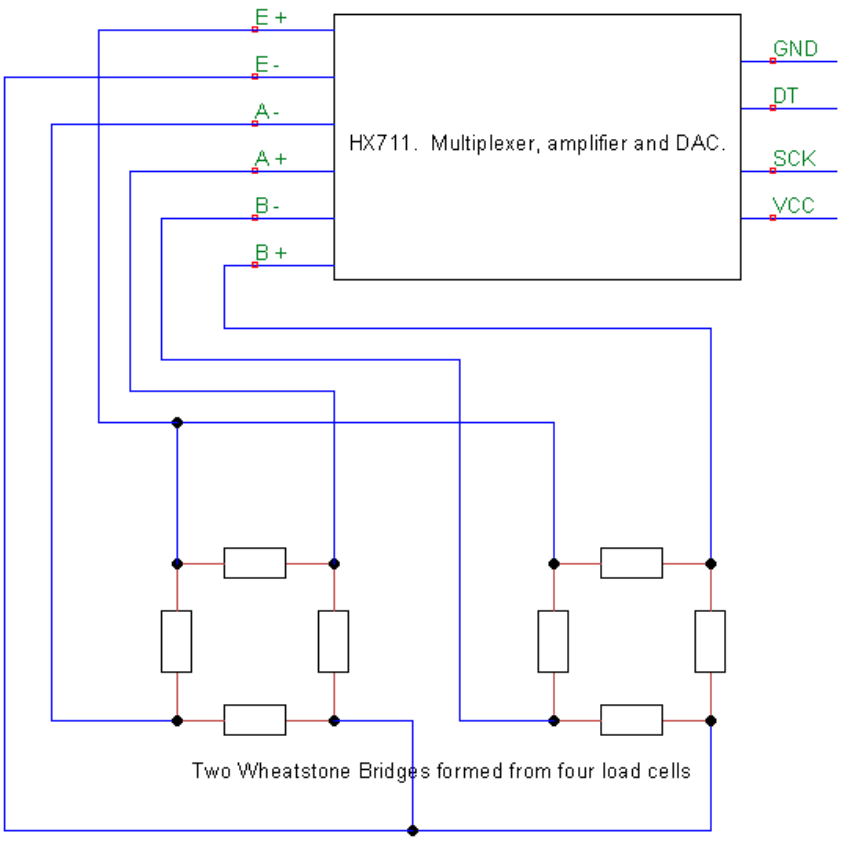


Figure 9. Two Wheatstone Bridge arrangements in the chair formed from four load cells that each contain two resistors (Hakkarainen, 2019).

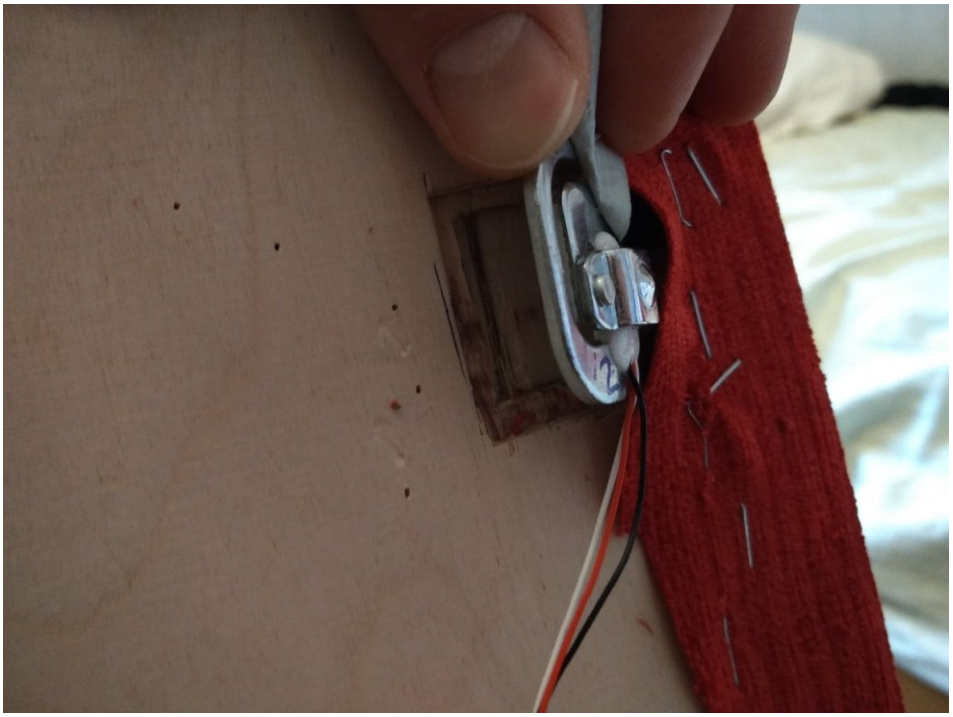


Figure 10. Installation of one of the load cells (Hakkarainen, 2019)

3.2.7 Relay

Relay is an electromagnetic switch that is used to control greater currents with a small trigger current. In this application a relay was needed to control DC vibration motor, since Raspberry Pi pins cannot provide enough current and it is also dangerous to wire a DC motor directly into Raspberry Pi since they can also act as generators for short moment after they are turned off and burn circuits in Raspberry Pi. Furthermore Raspberry Pi's pins cannot provide enough current to trigger the relay therefore the same power source PowerBank with two USB outlets was used for both Raspberry Pi and relay/vibration motor. One USB for Raspberry Pi and one USB for the relay/vibration motor circuit. Now only the control signal for the relay comes from Raspberry Pi and the test persons can simply turn on and off the vibration by connecting the USB. Schematic for controlling the vibration motor presented in the Figure 11.

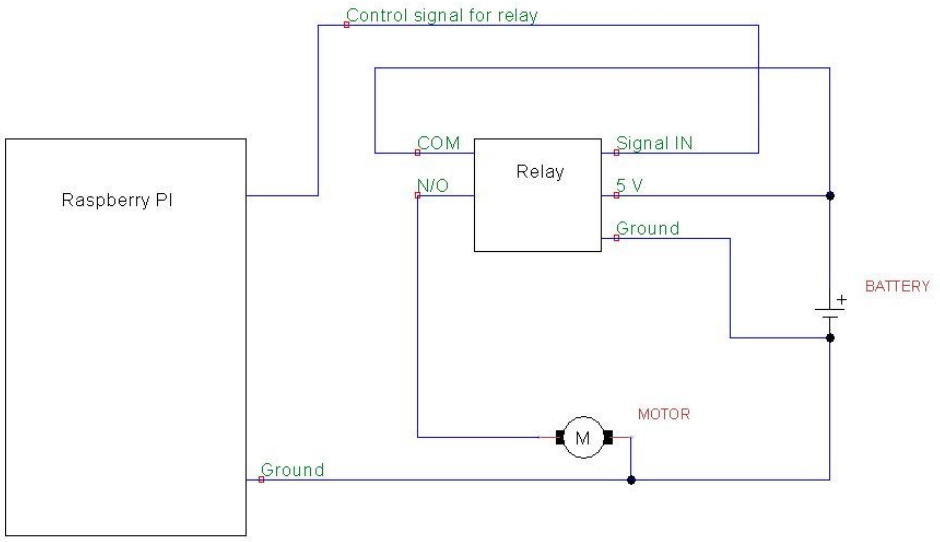


Figure 11. Schematics for the relay to control the vibration motor (Hakkarainen, 2019).

3.2.8 Vibration motor

The vibration motor as presented in figure 12 was a generic model ordered from Ebay. Its operating range is from 1.5 to 5 volts. In this application it operates at 5 volts from the Powerbank and draws 100mA of current, resulting 0.5W of power (Ebay, 2019)

Figure 12. Installation of the vibration motor (Hakkarainen, 2019)

3.3 **Programming**

Python is the most commonly used high level programming language for Raspberry pi. It is an easy to understand language for beginners. For easier reading and understanding of this programming chapter it is recommended that the reader has this text and the program it describes opened side by side. At the beginning of this project I was new to the Python and as part of the learning process two programs were developed the Program 1 and Program 2 the latter one being the final and cleaner. They both are trying to accomplish more or less the same tasks, but Program 2 is more minimalistic and it overcomes some of the issues that were encountered in the development of the Program 1. Program 2 was also the program used when testing the chair.

3.3.1 Program 1

This is an overly complex program for the user and something that was developed as a part of the learning process and testing, so that I could eventually strip it down and develop the Program 2 that is more minimalistic and user friendly so the user can either start following it at any point or choose to ignore it.

At the beginning the program calls all the different libraries that are needed for the code to run and operate Raspberry Pi pins and HX711 chip. From griozero library LED is used to send the control signal for the vibration motor relay, this is enough current to lit LED directly form Raspberry Pi's pins or in this case give control signal for the relay. (Appendix 4).

After that a (def cleanAndExit():) function is defined. The outer while loop between (try: and except) is continuously listening for keyboard interruption (Ctrl + c) if keyboard interruption is detected the while loop is exited and the (def cleanAndExit():) function is called. This will clean the Raspberry Pi's pins, so that if any of them was defined true or assigned to specific task are cleaned and ready to be assigned again when the program will start again. One of the benefits of functions is that if you have to repeatably and in multiple points in the code perform certain specific task you can write it once within the function and after that just call the function from the code making the code more compact and easier to read (Appendix 4).

After the function the specific Raspberry Pi pins are assigned for the HX711 chip and for the relay to operate the vibration motor. Then after that all the different variables are defined for the code to use. At the beginning of the code they are all defined as 0 but later on when the program is running it will modify them as needed.

The next part in the code is focused on the HX711 chip first the reading format is set. There are two options MSB and LSB these vary from python version to another since the order of bites varies. The first MSB sets the order in which the bytes are used to build the long value. The second MSB sets the order of bits inside of each byte. Then the HX711 is being reset and after that with hx.tare it is set to use both channels and identify them with A and B.

After these initial setups that will only be performed once at the beginning of the program the program will move on to the main while loop the first command after that is try: this is at the same level with the except command at the very bottom of the code. It will wait for keyboard interruption meaning CTRL + C and will kill the while loop and trigger (def cleanAndExit():) function. Once the keyboard interruption is detected otherwise the while loop will run infinitely.

Now at the first level of the while loop the program constantly reading the values from the load cells via HX711 in one second intervals. Once both of the values exceed 35000 it will move on to the next level. This next part of the code is so called welcoming part where the chair will vibrate shortly to let the user know it is working and it detected that the user is sitting on it. In this part of the code I had to also compensate for the two different gain levels between the HX711s A and B channels. The B channel has fixed gain of only 32 and the channel A can be set to either 64 or 128. The channel A is left to 128 and the channel B is multiplied by 4 always when reading the values.

Now the program will move on to the next while loop where it will again read the values from the load cells and decide on which while loop it should go next: depending on which side of the chair has more load on it.

Before the while loops there is one IF condition that is activated when certain continuous sitting interval has passed without notification vibrations and the user is still sitting on the chair. Meaning that the user has been successfully shifting his or her position avoiding the vibrations to stand up. When this FI condition activates the chair will give series of short vibrations to the user to notify to stand up instead of only shifting position.

The while loops for each of the sides have identical structure. At the beginning of each while loop the program will register starttime = time.time() and at the end stop = time.time() subtracting them and adding the result to total time spent sitting and printing it into locally stored text

file in the Raspberry Pi's memory card in the same location where the program is running.

Once the program enters either of the while loops it will sleep for 300 seconds and read again the values from the load cells and if the condition is still the same it will notify the user with quick vibration. If with in 15 seconds from that the user shifts his or her position to the other side it will only register the time spent sitting adding it to the total time and register from which side the notification came from then it would write these alongside all the other data logs into file.

In every part of the code when there is something written into file all the other data are also rewritten because in the outf = open("sittingtime.txt", "w") The program defined to overwrite the logging file every time changes are made otherwise the log file would grow extremely large. Later on this writing method was updated to: with open ("sittingtime.txt", "w") as text_file:. This is the current file writing method in Python 3 and above. The first method works too, but not always reliably.

If the user failed to shift his or her weight over to the other side within that 15 seconds the chair would vibrate more vigorously and count the total time spent sitting add one count to the more vigorous vibrations form either side. Then again it would rewrite the log file again with updated values.

These while loop structures are identical for each side. After the while loops there is only one IF condition to detect if the load cell values are under 350000, meaning that the user has stood up. This will then break the while loop and the program will return to first level while loop to scan the values at one second interval to detect when the user will sit again (Appendix 4).

As mentioned at the beginning of this paragraph this program is not very user friendly and the code itself is not very clean, but it was important part of the learning process in order to arrive into more minimalistic and user-friendly clean program.

3.3.2 Issues and problems with program 1

One of the downsides of this code is that while any of the if conditions inside either of the while loop is true the code is sleeping for the set time and then it will again check if the condition is still true or if the person has shifted position or stood up. This is not a big issue, but it can result in the chair vibrating to suggest position shift if the person stood up at certain point of the code when it is sleeping for a short while and sat down soon again in the same position and the chair was not able to notice it since the code was sleeping and when it wakes up again it appears that nothing changed and it will vibrate. This is possible to overcome by adding more IF conditions for only checking if anything has changed and if yes it will break the while loop early and proceed decision making based on the new information.

3.3.3 Issues with the Timer from threading

Attempting to implement another timer to measure the time between the vibrations to notify the person to shift position or stand up. One way to implement such timer was to use Timer thread that after certain period of no notifications it would have called function to run the vibrating motor. It also resets the timer every time vibration occurs regardless the source. The timer also will not start if it has been already started this would result in calling the function repeatedly and also vibrating repeatedly. This was achieved by simply establishing single variable that varies from 0 to 1 depending whether the timer is running or not. The status of this variable would determine will the timer start. In principle it works well. The reason why I could not leave it in was that you cannot restart threads so it was impossible to use the timer more than once. So now the person can sit on the chair infinitely with our notifications if they only remember to swift their position. Below code with the function and Timer thread parts commented. One way to go about this is to create timer instance as instructed in Figure 13.

`threading.Timer` inherits `threading.Thread`. Thread object is not reusable. You can create `Timer` instance for each call.

```
from threading import Timer
import time

class RepeatableTimer(object):
    def __init__(self, interval, function, args=[], kwargs={}):
        self._interval = interval
        self._function = function
        self._args = args
        self._kwargs = kwargs
    def start(self):
        t = Timer(self._interval, self._function, *self._args, **self._kwargs)
        t.start()

def hello():
    print "hello"

a=RepeatableTimer(3,hello,())
a.start()
time.sleep(4)
a.start()
```

Figure 13. Example from Stackoverflow how to create restart threading Timer (Stackoverflow, 2014)

In the time.time() was used to determine elapsed time without vibrations when the person sitting has successfully been shifting position. time.time() will simply mark starting time and ending time at their spots in the code and by subtracting the starting time from ending time we achieve the time interval. This can be then added into total time without shifting and once certain total value is exceeded this will trigger the vibration and also reset the total time without vibration count. Also any other vibration form not shifting your position will reset the count.

3.3.4 Alternative program ideas

In Figure 14 is presented one possible way to overcome the issue with instant detection for standing up. One alternative idea for making more faster response times for the program to instantly detect standing up could be running two simultaneous while loops within two functions:

```
from multiprocessing import Process

def loop_a():
    while 1:
        print("a")

def loop_b():
    while 1:
        print("b")

if __name__ == '__main__':
    Process(target=loop_a).start()
    Process(target=loop_b).start()
```

Figure 14. Simplified example of two while loops running simultaneously (Stackoverflow 2010)

One idea to utilize this would be to run the code I presented earlier with in the latter function loop_b and the first function loop_a would constantly check the values from the load cells and if the person stands up it would immediately kill the second function. The issue with this idea was that both loops were checking the same reading from HX711 and would either crash the code or produce very unreliable readings. The same error occurred if I had configured HX711 reading python program to start at boot in rc.local and then manually start another program reading HX711 as well this resulted in unreliable readings. Perhaps configuring turns for each programs to read HX711 at different times would help.

One way around this could be assigning the first function to simply listen to on-off switch and based on that kill the latter function. At this stage hardware was already set and there was no change to make any changes into hardware anymore.

3.3.5 Program 2

The main idea with the second and so far, the final program was to be more minimalistic and less intrusive for the user, so that the user could choose either to ignore or follow it to their liking or depending on the current situation at work (Appendix 5).

The program lets the person sitting in the chair know by vibrating on which side there is more load at two-minute intervals and after one side being more loaded continuously for six minutes it will vibrate more vigorously. This more vigorous vibration is possible to avoid by shifting our weight more often than 6 minutes. Regardless of these vibrations the program will also notify the user every 45 minutes with four vigorous vibrations to stand up and take a short break from sitting. This program does include data logging: it will log the overall time spent sitting, occurring of all the different vibrations and number of times the user sat down.

From here on in this paragraph regarding this program 2 only the parts of the code are explained that are different from the program 1 to avoid repetition.

As did the first program this too starts with importing all the needed libraries for the program to run and operate the hardware connected to IO pins. After that function is defined for writing the logs into local file. The reason why this operation was built into a function is that writing the logs is performed frequently in the code and it is several rows long operation. This way the rest of the code can be shorter and easier to read. Now it is possible to call this function from the code with single line and not having to list all the writing operations repeatedly. 100ms sleep periods were added in between every writing operation since perhaps due to worn micro SD card sometimes the writing would not register all the changes.

It was also tested to build functions in similar manner for all the vibration notifications in the code to make the code even shorter, but in my opinion the code is easier to read this way when all the vibration sections are clearly visible sections in the code.

Reading values form the load cells is also one part in the code that is relatively long and is performed often. Function was built for it too, but in testing it turned out that the IF conditions would not operate reliably with the load cell values from the function, even though the printing in terminal showed accurate readings from the load cells. Also using only one function for reading the values would mean it would not be possible to distinguish the terminal prints from each other to determinate what part of the code printed them. This is important for trouble shooting the code. For these

reasons all the reading and printing operations of the load cell values were left in their entirety into the code.

After this all the variables needed to run the code and for keeping the log of the sitting habits are defined. Then the main while loop starts and the beginning is essentially identical with the program 1 where the first values from the HX711 is being read and if they fit within the thresholds the program will proceed to the welcoming part where the chair will vibrate briefly to let the user know that the program is running.

This program is done entirely by using IF conditions within the final while loop. The first IF condition waits for the overall sitting time to exceed 45 minutes and this time is regardless of any of the other notifications, meaning that even if you shifted your weight from side to side successfully within the 6 minutes intervals avoiding all the more vigorous vibrations notifying you to shift your weight, this IF condition will give the user four strong vibrations to let them know it is time to stand up and take a small break from sitting. This IF condition will also add to the logs how many times it occurred and it will then zero its own time count.

Next follows a series of IF conditions that are essentially identical for both sides of the chair except for the different vibration patterns. Both sides start with their own notification pattern to let the user know on which side there is more weight: if left side is more loaded then user will get one short vibration and if right side is more loaded user will receive two short vibrations. After this the code will check on the load distribution in one-minute intervals and every second time it will let the user know the situation: which side is more loaded. Every one-minute interval will also update the needed variables for the logs. After six of the IF conditions have been true in a row for one side the program will give stronger vibration to let the user know it is time to shift position: if the left side was more loaded for six minutes the user will receive one long vibration and if the notification comes from right side it will be two long vibrations (Appendix 5).

3.3.6 Libraries

Gpiozero is library developed by Ben Nuttall. It is a simple interface to GPiO devices with Raspberry Pi. From Gpiozero it is possible to import controls for relays and after that it is possible to assign one of the GPIOs in the Raspberry for particular relay.

HX711 Python library allows the Raspberry Pi to drive and communicate with the HX711 chip. It provides commands for Python to communicate with HX711.

Time provides the ability to measure time it is needed for the sleep commands, timers and taking start and stop time marks at different parts of the code which you can then subtract and see how much time certain part of the code took.

Threading provides start and stop among many other functions. For this code only timers were needed. One downside or Threading timers is that they can only be started once during the code. In order to start them more than once you need to build threading timer with function.

Sys system specific parameters and functions.

4 TESTING

4.1 Testing the device

The intention is that the tester would use the chair in their everyday life as he or she would use any other chair. The battery life of the chair is about two days, during the testing users were asked to charge the chair every night so that the logs of their sitting habits would be safe. The users also received short instruction manual on how to follow the different vibration patterns of the chair (Appendix 6) and questionnaire to fill regarding their experience with the chair (Appendix 7)

4.1.1 Real VNC and Remot3.it

VNC Connect is a remote access software that is available in multiple platforms including Raspbian. It consists of a server (VNC Server) and client (VNC Viewer) application for the Virtual Network Computing (VNC) protocol to control another computer's screen remotely (realvnc 2019). At the beginning of the project VNC was only used to operate and program the Raspberry Pi from laptop eliminating the need for external display and input devices.

Later in the testing phase the Remot3.it was also utilized to gain access to Raspberry's desktop remotely over the interned. Remot3.it is versatile remote device managing software that is used in different industries to remotely manage up to thousands of devices. The installation is fairly straightforward process via terminal and it is documented well too. Once the installation was completed, next step was to register Raspberry Pi device on my account and always when it was connected to internet and had IP address it was possible to operate it from anywhere. (remot3 2019)

4.1.2 Local data logging

Because the Raspberry Pi has only 2.4 Ghz Wlan antenna embedded that restricts the real time monitoring of the sitting statistics, since most of the current wireless networks are configured to operate at 5 Ghz. The way around this was to ask the test persons to pair their mobile phones hot spot with the chair at 2.4 Ghz frequency at agreed times then it was possible to check the logs and make changes to the vibration notifications intervals if the test person requested so with the use of Remot3.it and RealVNC. This was particularly handy when the chair was in testing in Tampere and I lived in Valkeakoski. With the combination of Remot3.it and RealVNC it was possible to operate and program the Raspberry as if being next to it. This way it was possible to remotely pull the logs about the sitting habits with and without the vibration and see if the vibration made any difference.

4.1.3 Sitting habits and the possible change

The chair was tested by two people in addition to myself. The testing procedure was simple. The chair communicates with vibrations at specific intervals as explained in the programming chapter and manual (Appendix 6). First the vibrations are turned off by disconnecting the vibration circuit USB from the Powerbank while the program is still running as usual, this way it is possible to gather data of the persons sitting habits without the vibrations. After the first part the vibration circuit is turned back on and then by comparing the data from both periods it is possible to see if the chair has made any difference. The test users were also asked to fill questionnaire about their experience with the chair (Appendix 7).

4.1.4 Results

Test person 1

In the case of test person 1 the logs illustrated in Figure 15 and Figure 16 show clear difference between no vibration and with the vibration regarding how straight she sat or how evenly the load was distributed between both sides of the chair. During the first test period she sat on the chair over four-day period total of 7 hours 46 minutes. Where the vibration being turned on she would have received over 250% more notification about the load distribution from the right side than from the left, and 450% more vibrations indicating the need to shift position form the right side than from left. During the second period which lasted for four days and during these four days she sat on the chair total of 3 hours 53 minutes. With the vibration turned on and the chair communicating with her the corresponding values where: 147% and 140% indicating significant improvement in sitting straight as illustrated in figures 14 and 15. During both test periods she only got three notifications to stand up after 45 minutes meaning that she was already taking more or less enough breaks from sitting as is even before trying this chair. This was also in line with the information she provided in the questionnaire where she stated that she normally takes breaks from sitting every half an hour and also that she feels comfortable with the overall time she spends sitting in her everyday life. In the questionnaire she described how she would also be interested in improving her sitting position and how well she followed the vibration notifications from the chair. The intervals of the notifications are based on the research of the first part of this thesis and she did find them to be suitable and not annoying or something she would ignore (Appendix 8).

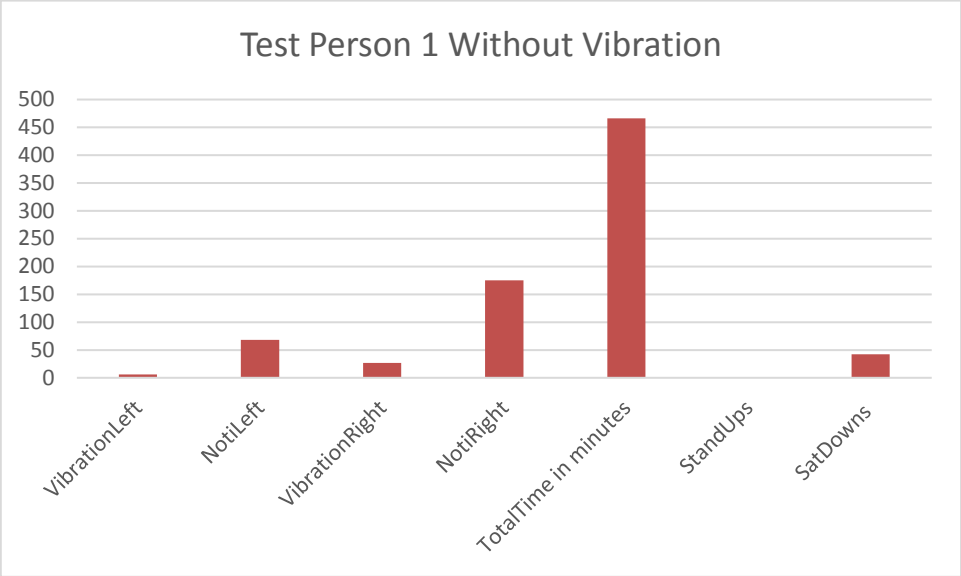


Figure 15. Test person 1 without vibration (Hakkarainen, 2019)

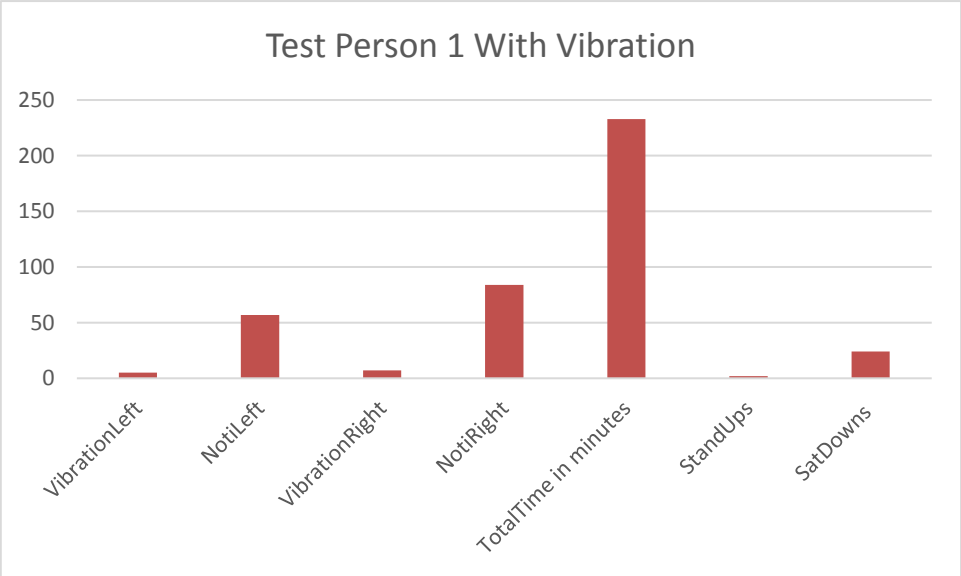


Figure 16. Test person 1 with vibration (Hakkarainen, 2019)

Test person 2

During the first test period of three days without vibration test person 2 sat for 7 hours 9 minutes and was leaning to left more than twice as much than to right. During the second phase with vibration that lasted also for three days and total sitting time was 4 hours 56 minutes. This time she was leaning to the left only 140% more than to right. In the case of test person 2 there there appears to be reduction in leaning to one side as illustrated in Figure 17 and in Figure 18.

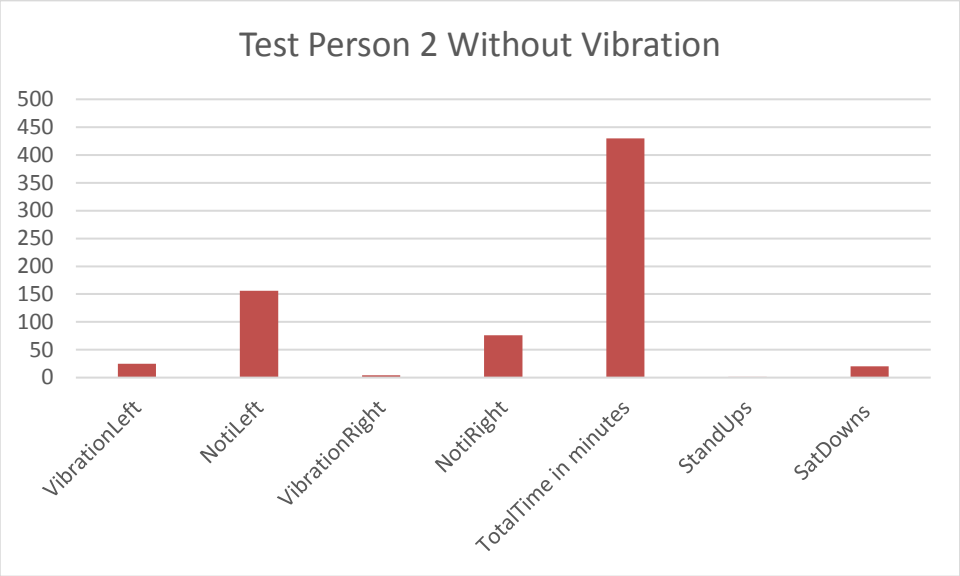


Figure 17. Test person 2 without vibration (Hakkarainen, 2019)

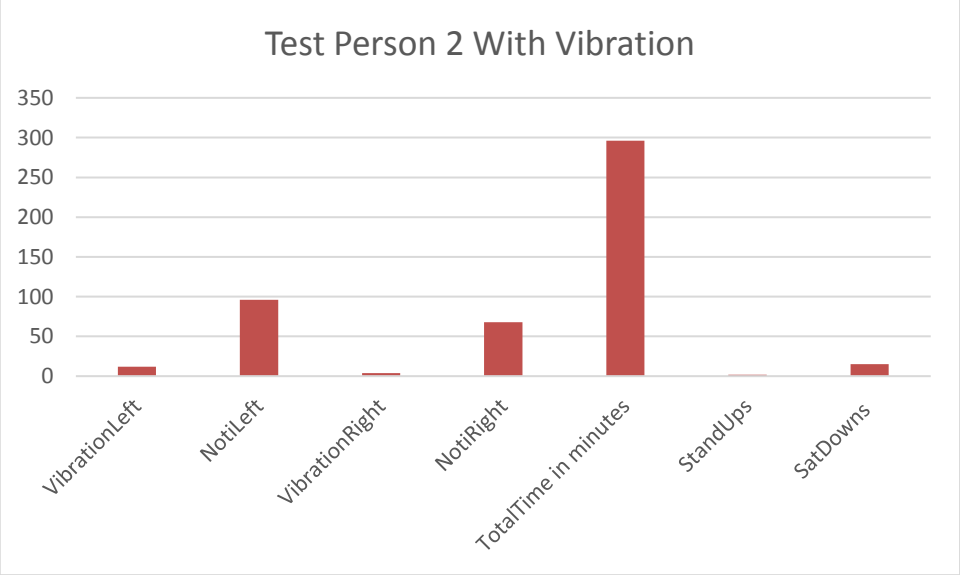


Figure 18. Test person 2 with vibration (Hakkarainen, 2019)

Test person 3

In the case of the test person 3, he sat over four-day test period for 11 hours 29 minutes without vibration and with vibration for 9 hours 51 minutes. It appears that he already sat quite straight and the weight was distributed equally already in the first part of the test without vibration and there was no significant difference in comparison to the second phase with vibration turned on as illustrated in Figures 19 and 20.

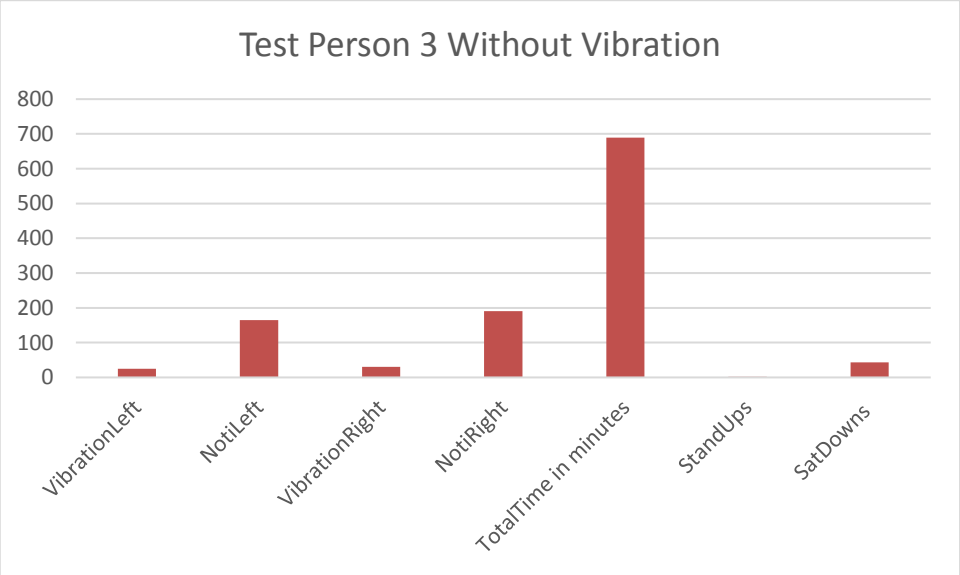


Figure 19. Test person 3 without vibration (Hakkarainen, 2019)

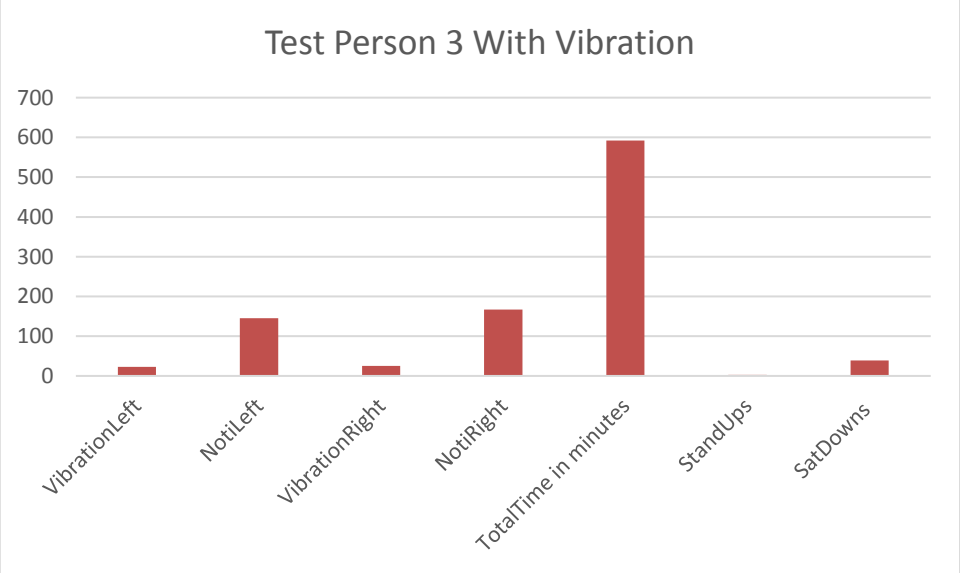


Figure 20. Test person 3 with vibration (Hakkarainen, 2019)

5 CONCLUSIONS

This research indicated that there was potential in the intelligent chair that communicates through different vibration patterns to reduce negative health effects of prolonged sitting and improves the sitting positions of the user. Even though the test period was fairly short due to a lack of time, all the different users sat on the chair in total for 45 hours and 4 minutes.

To conduct more comprehensive testing for producing more reliable results the number of testers and test time should be increased by using multiple chairs. The cost estimate for a prototype chair like this would be:

- The chair 120 EUR
- Raspberry Pi 45 EUR
- PowerBank 30 EUR
- HX711 and the load cells 5 – 10 EUR
- Vibration motor, relay and wiring 10 EUR

TOTAL APPROXIMATELY 215 EUR

For higher reliability and to detect a greater variety of different sitting positions it would be beneficial to add more sensors and to consider adding sensor into the backrest of the chair as well. Even though this prototype device was embedded into a chair to keep the system compact due to a lack of purpose specific hardware, one of the main design goals of this prototype concept was that this would be something the user could easily place on any chair, therefore the sensor and the rest of the hardware was limited into the seat only.

After this project it seems that the next logical step would be to further develop the program and to expand it into artificial neural networks (digitaltrends 2019) for identifying a variety of different sitting positions based on the information from number of different sensors and to build an ergonomically oriented chair with multiple load cells, proximity sensors and vibration motors in the seat and in the backrest as well. Although it is possible that this could introduce reliability problems with the proximity sensors as described earlier in the second part of this thesis: depending on the person’s clothing the measurement from the backrest would not necessarily always be accurate.

At the time of writing this thesis there were not many pieces of intelligent furniture in the market, but with the rising trend of home automation the demand for intelligent furniture as a part of home automation systems is expected to grow (Sapfl, n.d.). When finishing this thesis I came across another very recently launched chair with a similar purpose. At the time of writing this their website was still under construction. One conference paper came up as well where researchers in 2013 used pressure sensors to detect different sitting positions, but in their study they did not provide solutions to correct the positions.

The company that I work for in the field of warehouse automation is interested in this concept, the results of this project and the technology used in the prototype. We are starting a project to keep developing this work into applications in their field.

REFERENCES

Al-Mutlaq, S. (n.d.). Getting Started with Load Cells. **Retrieved 1 December 2018 from** https://learn.sparkfun.com/tutorials/getting-started-with-load-cells

Ebay. (2018). Strong earthquake motor silent 300 DC vibration motor 3V-5V massager. **Retrieved 2 December 2018 from** https://www.ebay.com/itm/Strong-earthquake-Vibration-motor-Silent-300-DC-vibration-motor-3V-5V-massager/323273901456?hash=item4b44a04590:g:lkcAAOSw1-5bCTuH:rk:3:pf:0

Ecodesign. (2017). Ecodesign 2017 catalog. **Retrieved 14 January 2019 from** http://www.ecodesign.fi/blog/2017/9/14/ecodesign-2017

Electronics Tutorials (n.d.). Wheatstone Bridge. **Retrieved 2 November 2018 from** https://www.electronics-tutorials.ws/blog/wheatstone-bridge.html

Finni, T., Pekkonen, M., Pesola, A. (n.d.). Miksi liiallinen istuminen on vaarallista? **Retrieved 16 November 2018 from** https://www.duodecimlehti.fi/api/pdf/duo13381

Hakkarainen, A. (2017). Mona chair. **Retrieved 3 February 2019 from** https://www.coroflot.com/aleksihakkarainen/Mona-chair

Laakso, L. (2015). Istuminen hengenvaarallista, liikunta ei korvaa. **Retrieved 1 November 2018 from** https://www.kauppalehti.fi/uutiset/istuminen-hengenvaarallista-liikunta-ei-korvaa/82f64f3f-958e-32e7-a5c8-9c1ec2718416

Launis, M. & Lehtelä, J. (2011). Ergonomia. Helsinki: Työterveyslaitos.

Loomis, J. (2015). Sitting Up-Straight: A Personal Struggle for Proper Posture. **Retrieved 25 November 2018 from** https://www.cirrusinsight.com/blog/sitting-up-straight-a-personal-struggle-for-proper-posture

Luke Dormehl. (2019). What is an artificial neural network? Here's everything you deed to know. **Retrieved 15 February 2019 from** https://www.digitaltrends.com/cool-tech/what-is-an-artificial-neural-network/

Mainostoimisto Valo Oy. (2017). Ecodesign 2017. **Retrieved 15 January 2019 from** https://issuu.com/mainostoimistovalooy/docs/ecodesign_2017_katalogi

Mouser Electronics (n.d.). HX711. **Retrieved 26 November 2018 from** https://www.mouser.com/ds/2/813/hx711_english-1022875.pdf

Opsvik, P. (2009). Rethinking sitting. 1st Edition. New York: W.W. Norton & Company Ltd.

Raspberry Pi Foundation (n.d.). Raspberry Pi. **Retrieved 14 January 2019 from** https://www.raspberrypi.org/

REALVNC. (n.d.). Connect to your Raspberry pi from anywhere. **Retrieved 3 February 2019 from** https://www.realvnc.com/en/raspberrypi/

Remot3.it. (n.d.). Remot3.it. **Retrieved 3 February 2019 from** https://www.remot3.it/web/index.html

Stackoverflow. (2010). How do I run two Python loops concurrently? **Retrieved 6 November 2018 from** https://stackoverflow.com/questions/3474382/how-do-i-run-two-python-loops-concurrently

Stackoverflow. (2014). Calling thread.timer() more than once. **Retrieved 4 November 2018 from** https://stackoverflow.com/questions/22433394/calling-thread-timer-more-than-once

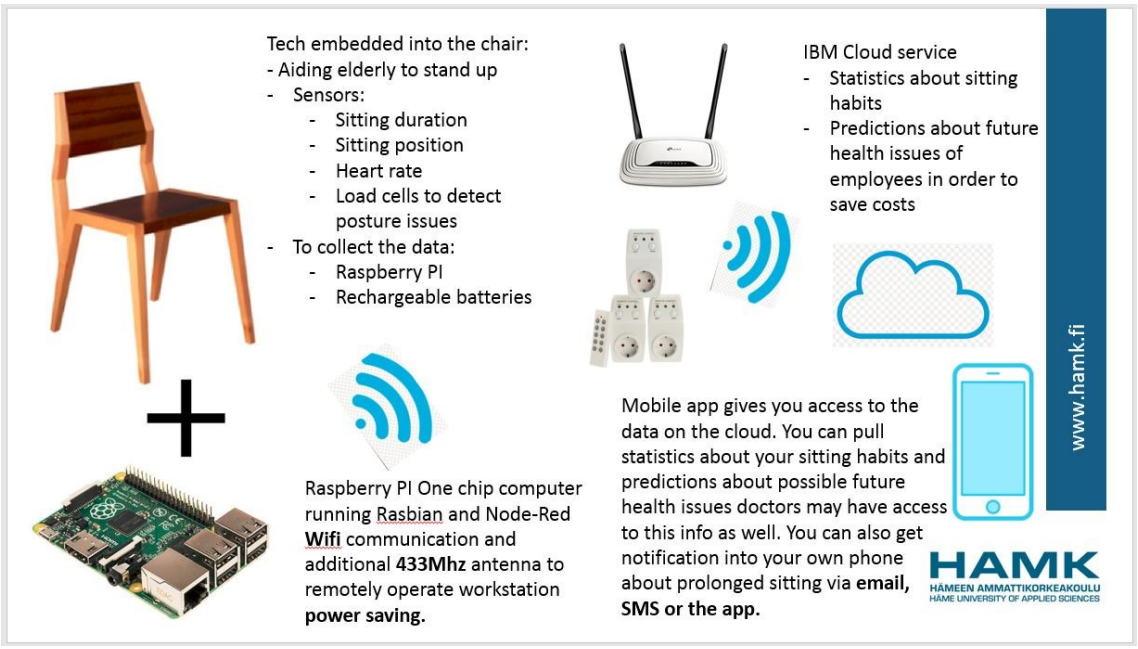
Zapfl, D. (n.d.). Why there is very little intelligent furniture for the smart home. **Retrieved 6 February 2019 from** https://www.lead-innovation.com/english-blog/intelligent-furniture

INTERVIEWS

Kaita, A. (2018). Työfysioterapeutti, Terveystalo Oyj. Haastattelu 25.10.2018.

Appendix 1

HOME AUTOMATION COURSE CHAIR PROTOTYPE



Appendix 2

INTERVIEW QUESTIONS FOR AULI KAITA

Työfysioterapeutti Auli Kaitan haastattelu

Terveystalo Valkeakoski

25.10.2018

1. Millaisia vaivoja teidän potilaissa esiintyy jotka voivat olla yhteydessä liialliseen istumiseen?
2. Onko tapoja tai keinoja ennaltaehkäistä näitä?
3. Onko mahdollista nimetä aloja jolla esiintyisi eniten istumiseen liittyviä vaivoja?
4. Mitkä ovat yleisimpiä asento/istuma virheitä?

5. Tuolin ominaisuudet

    a. Millaisia ajatuksia herää?
    b. Ominaisuudet ja ilmoitukset enemmän häiriö vai hyöty?
    c. Ehdotuksia mitä kannattaisi mitata?
    d. Tuleeko mieleen missä tuolille voisi olla käyttöä?
    e. Yhteistyö lääkäreiden/työterveyshuollon kanssa?
        i. säästöt
        ii. Ennaltaehkäisy

Appendix 3

WRITTEN PERMISSION TO USE AULI KAITA'S NAME AND PUBLISH TRANSCRIPT OF HER INTERVIEW

Intelligent Furniture

Lupa nimen ja haastettelulitteraatin käyttöön

30.05.2019

Tällä suostumuksella annan luvan, sekä nimeni käyttöön, että 25.10.2018 tehdyn istumisen terveyshaittoja koskevan haastattelu litteraatin julkaisemiseen, osana Aleksi Hakkaraisen Theseuksessa julkaistavaa opinnäytetyötä Intelligen Furniture. Nimi esiintyy tekstiosuudessa ja litteraatti työn lopussa liitteenä.

Päivämäärä, allekirjoitus ja nimenselvennys.

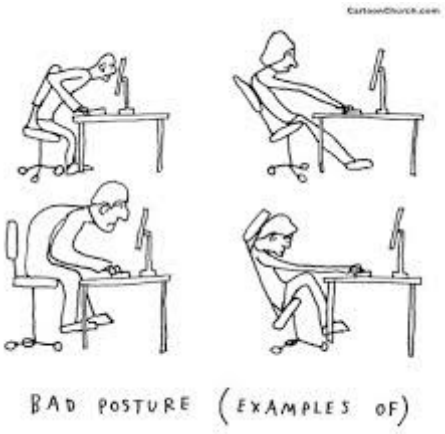
Appendix 4

TRANSCRIPTED INTERVIEW WITH PHYSIOTERAPIST AULI KAITA

Transcript of the interview with occupational physiotherapist Auli Kaita. The most essential parts are marked in green and bold.

AK: Eli sitten sinä halusit kuulla millaisia istumiseen liittyviä vaivoja meidän työntekijöillä on.
AH: joo
AK: Aika selkeältä tuntuu se yhteys että jotka paljon istuvat niin heillä alkaa niitä oireita olemaan. Osalla tulee lyhyessäkin ajassa, osalla se vaatii vuosia tai suosikymmeniä. Kyllä selänalue on se joka siellä oireilee ja alaselästä ihan ylös asti niskahartia seutuun. Istumasentoon liittyen se saattaa myös tuntua ihan sormissa, päänsärkynä ja silmäoireina. Selkeimmin se yhteys kuitenkin löytyy tuonne niskaranka-alueelle.
AK: Sitä on toimistotyötä tekevillä ja ajoneuvokuljettajilla missä esimerkiksi löytyy tärinä siihen parina.
AK: Olen ollut kovasti tyytyväinen että sähköpöydät ovat lisääntyneet työpaikoilla ja istumista on saatu vähennettyä. Siitähän tehtiin jossain vaiheessa tutkimuksia että kuinka paljon ihmiset istuvat. Tutkijat vähän yllättyivät tuntimäärästä että se keskimääräinen alkoi olemaan 6 tuntia päivässä ja minun mielestä siinä oli toimialat mukana joissa ei istuta lainkaan eli ne jotka istuvat, istuvat tosi paljon. Päätelmä oli siihen suuntaa ettei enää ole sellaista liikunta määrää jolla niitä haittoja voidaan kompensoida. Sen takia sähköpöytä on erinomainen sinä voit jatkaa sitä samaa työtä ja vaihtaa aina asentoa. Erilaisia tuoleja on kiitettävästi kehitelty: satulamallinen on ollut jo pitkään ja niiden variaatioita eli onko se istuin satulanmuotoinen vai esim. pyöreä. Se saattaa myötäillä sitä liikettä. Tai sitten että se edelleen on satulan muotoinen mutta siinä on swing toiminta niin että alavartalo ei ole passiivinen. Sitten on istuma- ja seisoma-asennon välimuoto tuoleja jotka saattavat keinua myös ja ideana että painoa siirretään selänrakenteilta tuonne jaloille. Sitten se alaselän vääntyminen koska asento muuttuu jos vertaa normaaliin istumiseen. Tuolla lannerangassa on nikamat vähän niinkuin tornissa päällekkäin ja välilevyillä on tasainen tila mutta jos ajatellaan perinteistä istumista niin ne ovat etureunalta kantillaan ja puristuvat toiselle reunalle. Satulatuolilla lonkan kulma on loivempi ja se ei pakota välilevyjä samalla lailla. **Istumista suositellaan tauotettavaksi, suositus on ettei tuntia kauempaa oltaisi kerrallaan. Jos selänalueella on oireilua missä tahansa kohtaa niin se suositusaika on lyhennetty puoleen tuntiin. Siihen on ihan tervetullut jos siihen löytyy joku joka muistuttaa sen tauottamisen ettei tarvitsisi mennä siihen että se kipu muistutta että nyt on pakko vaihtaa asentoa.** Popup ikkuna tietokoneella. Olenkohan minä jossain kuullut että se tuolikin muistuttaisi siitä.
AH: En ole kuullut varsinaisista tuotteista mutta sellaisia projekteja on rakenneltu missä se tuoli alkaa kääntymään vähän kuin munakello sitten ennepitkää sinä et pysty tekemään työtä kunnes käännät sen tuolin takaisin.

AK: Niin kuin sanoit tällainen värinä voisi olla ideana että kun sinä olet istunut tarpeeksi kauan niin se tuoli alkaa väristä muistutuksena.
AH: Tuolia ajatellen värinä olisi hyvä niin että kokonaisuus pysyisi selkeänä. Jos joka puolelta tulee muistutusta SMS ja Sposti jne. Ja jos puhelimessa on tietoa vielä sinun istumistavoista niin en tiedä onko sitä kukaan eden kiinnostunut saamaan niin paljon tietoa omasta istumisestaan.
AK: Äkkiseltään tuntuu että tämä voisi olla se erilainen juttu että se tuoli muistuttaisi siitä asennon vaihtelusta. Että sinne tuolin rakenteisiin on rakennettu niitä jotka tekisivät istumiseen vaihtelua. Seisominenkaan ei ole kaikille ratkaisu.
AH: Ennaltaehkäisy tässä tapauksessa niin ehkä se selkein ratkaisu on se tauotaminen?
AK: Kyllä. Tilastollisesti ne ihmiset jotka työssään istuvat ja seisovat ja vaihtelevat asentoa niin heillä on vähiten näitä oireita. Olen kannustanut tiettyyn levottomuuteen.
AK: Mitkä ovat yleisimpiä istuma- ja asentovirheitä? Aina välillä näytän tämmöistä:



Tässä on vähän karrikoiden näitä haettu mutta tähän tapaan. Tämä on ehkä miesten asento että liu'taan tuonne (oikea yläkulma) pöydän alle ja pitkä matka selästä on ilman tukea että tukipiste on jossain hartioissa ja pää on etukenossa kun koitetaan nähdä. Kädet on pitkästi irti vartalosta. Voi arvat että kaularangan alaosa on aika kovilla ja sitten myös alaselkä koska siellä ei ole tukea. Sitten on toinen että jos mennään kovasti kumaraan sinne pöydän ääreen niin sitten on (vasen ylä) usen syytä tarkistaa silmälasit näissä molemmissa (vasen ylä ja ala). Siten on erilaisia vino juttuja että on perinteinen näppäimistö ja hiiri käytössä niin moni nojaa vasemmalle että on helpompi liikutella oikeaa kättä. Jos se hiiri vielä karkaa tuonne kauas niin sitten ollaan vielä kiertyneinäkin, sitten on vasen lapa kipeä.
AH: Yksi idea minulla on tuohon asennon korjaukseen että käyttäisin etäisyys sensoreita selkänojassa niiden tiedon voisi yhdistää siihen painon jakaumaan istuimella ja todeta jos istuja rönöttää näin. Selkänojan yläosassa selkä kiinni mutta alaosa etäällä selästä ja istuimessa paino etujaloilla eli takamus istuimen etuosassa. Se olisi tunnistettavissa rönöttämiseksi.

AK: Tämä ei olisi ollenkaan huono vaihtoehto (Demonstroi selkä suorassa) Että istuu siinä ihan tuolin etureunalla, mutta silloin sensoreilla hankala havaita. Selkänojassa ei painoa, mutta minun mielestä tämä pitäisi sallia. Tämä on vähän niinkuin keino hakea satulatuolimaista asentoa tavallisessa tuolissa. Lonkkakulma on aika oikea ja sitten tehdään omilla lihaksilla tukemista niinkuin satulassakin tehtäisiin ylävartalossa. Mutta ei tarvitse niin paljon tukea koska lonkka ei ole alhaalta vääntynyt.

AH: Tämä olisi hyvä asento?

AK: Kyllä. Mutta sitten kun käytät selkänojaa niin siirry koko vartalolla sinne selkänojalle asti ettet vain kaada sitä ylävartaloa sinne. Eli ihan hyvä idea.

AH: Niinkuin perinteisessäkin tuolissa lonkkakulma on liian jyrkkä ja tällä saadaan lonkkakulma oikenemaan?

AK: Kyllä.

AH: Sitten haluaisin kuulla sinun ajatuksia siitä että tässä vaiheessa on paljon ominaisuuksia suunnitteilla: datan varastointia, puhelin appia istuma määrien seuraamiseen, henkilökohtaiset tilastot jne. Että siinä tulee aika paljon tietotulvaa yhden asian ympärille. Onko tämä liikaa ja vain häiritsee? Sitten sinulle ammattilaisena; onko se hyvä idea että siinä olisi yhteistyö terveysalanammattilaisten ja tämän tiedon kanssa? Että organisaatiossa olisi kaikilla työntekijöillä tällaiset tuolit ja niillä voitaisiin ennaltaehkäistä työntekijöille koituvia istumishaittoja ja säästää kuluissa?

AK: Antaisikohan se kuitenkaan lisää tietoa istumisen määrästä? Mietin että ihmiset kuitenkin aika hyvin tietää että kuinka paljon he istuvat. Kun minä sitä aika usein kysyn että ei siellä kyllä paljon sitä ihmettelyä ole että paljonkohan minä istun, että sitä joutuisi miettimään. Enemmänkin he miettivät sitä että kuinka he sen muotoilevat minulle, että onko työpäivästä mikä osa. **Kyllä he sen tuntuvat tietävän aika hyvin ja osaavat hyvin kertoa että aamupäivällä tai iltapäivällä jne. että miten se vaihteleekin. En ole ihan varma onko siinä sellaista dataa mitä ei jo tiedettäisi. Ehkä enemmän mitä tuosta istuma-asennosta sanoit niin sitä ei ehkä niinkään tiedetä, että mitenkä minä siinä tuolissa olen. Kyllä minä tiedän että istun sen aamupäivän, mutta kyllä moni sitten sanoo että mä löydän itseni sitten jostakin ihmeasennosta. Sitä halutaan korjata etten haluaisi istua niin, mutta sitten taas huomataan että olen taas valunut sinne. Luulen että siinä voisi olla sellainen että se tuoli auttaisi siinä asennon pitämisessä. Sitten minä luulen että ihmiset alkavat olemaan ylikuormitettuja siinä mitä tulee sähköpostiin tai puhelimeen. Että mielummin jokin muu kommunikaatio esim. Värähtely tai muu tuskin halutaan mitään ääniä mutta jokin viesti sieltä tuolista että vaikka värinä: että nyt on painoa liikaa jollain ääriosalla tuolia. Se saattaisi ola sellainen että huomaa että nyt voisi korjata asentoa. Palvelisi enemmän tarkoitusta.**

AH: Tuo on varmaan ihan totta ja näin minäkin vähän epäilen. Olen muutenkin koittanut saada vähän järkevämpiin mittoihin tätä projektia. Tuo voisi olla hyvä rajaus. Kun älykellotkin kehittyy huimasti niistä saa

luotettavammin ja kattavammin sen paikallaan olo määrän tietoon. Yksi tuoli on vain yksi tuoli ja sä et käytä vain yhtä tuolia. Että tuoli olisi itsenäinen kokonaisuus ja koko tämä järjestelmä rajoittuisi siihen tuoliin.
AK: Mitä helpompi sen parempi. Ettei sun tarvitse mitään asentaa tai ladata vaan kaikki tulee valmiina tuolin mukana. Monissa toimistotuoleissa on kolme vipua istuimen alla joissa on merkinnät mitä mikin tekee. Sitten tuli uudet muotoilut joissa ei ollut kunnon merkintöjä mikä tekee mitäkin etkä sinä oikeasti tiedä painetaanko sitä kierretäänkö tai mitä se tekee. Ihmiset ovat pulassa niiden tuolien kanssa ja jossain on se ohje. Jos siihen vielä tulee monta asiaa lisää niin se menee liian hankalaksi. Keep it simple.
AH: Ehkä voisi olla niin että se tulisi yksinkertaisen manuaalin kassa jossa yleisimmät virheasennot selitetty ja kun tuoli värähtää jokin näistä asennoista on kyseessä.
AK: Ja se antaa sinun olla siinä asennossa vähän aikaa, mutta kun olet ollut vartin samassa asennossa niin huomautus.
AK: Onko sinulla tuoli malli johonka näitä voisi tulla?
AH: Ei ole, mutta minulla on paljon omia tuoli malleja. Kehitän ehkä tähän oman tuolin mallin vielä. Se voisi olla myös erillinen laite esim. Levy joka tulee ihan minkä tuolin päälle vain.
AK: Sen voisi lisävarusteeksi myydä.
AH: Niin ettei tarvitsisi tuolia ostaa sitä varten.
AK: Tuo voisi olla vielä toimivampi. Joo, aika hyvä. Se tekee jo hinnassakin houkuttelevamman.
AH: Helpottaa ostopäätöstä.
AK: Monet yritykset tekee sitä että uutta juttua testataan vain muutamalla kappaleella jota kierrätetään siellä että halutaanko näitä lisää. Että voi tarjota kokeilu kappaletta sillä saa kanavia auki.
AH: Hyödyllinen keskustelu oli, auttoi löytämään suuntaa ja rajaamaan aihetta. Kiitos.

Appendix 5

PROGRAM 1

```
import RPi.GPIO as GPIO
import time
import threading
import sys
from threading import Timer
from time import sleep
from signal import pause
from hx711 import HX711
from gpiozero import LED
from gpiozero import Button

def cleanAndExit():
    print ("Cleaning...")
    GPIO.cleanup()
    print ("Bye!")
    sys.exit()

led = LED(17)
button = Button(3)
hx = HX711(5, 6)
LeaningLeft = 0
LeaningRight = 0
T1B = 0
VibrationsFromLeft = 0
VibrationsFromRight = 0
TT = 0
total = 0
TotalWithoutVibration = 0
TotalWithoutVibrationVibrations = 0
contsit = 0
comparison = 350000



hx.set_reading_format("MSB", "MSB")
hx.set_reference_unit(1)
hx.reset()
# tare (use both) channels
hx.tare()
hx.tare_A()
hx.tare_B()

while True:

    try:


#       val_LeftGain = hx.get_weight_A(5)
#       sleep(0.1)
#       val_Left = (val_LeftGain - 13500000) ## if issues with side shifting remove this
        val_Left = hx.get_weight_A(5)
        val_RightGain = hx.get_weight_B(5)
        val_Right = (val_RightGain * 4)
        print "(0)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
        print "1.0 WithoutVib: %s" % (TotalWithoutVibration)
        TotalWithoutVibration = 0
        channel='A',
        gain=64
        sleep(1)

## Tervehdysta

        if (val_Left > 350000) and (val_Right > 350000):
            led.on()
            sleep(0.5)
            led.off()
            print("pass")

            while True:

                val_Left = hx.get_weight_A(5)
                val_RightGain = hx.get_weight_B(5)
                val_Right = (val_RightGain * 4)
                print "(0)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
                print "2.0 WithoutVib: %s" % (TotalWithoutVibration)
                channel='A',
                gain=64
                sleep(1)

## Tervehdysta
```

```
if ((TotalWithoutVibration > 900) and (val_Left > 350000) and (val_Right > 350000)):
    print ("continous sitting for over 450 sec")
    led.on()
    sleep(0.3)
    led.off()
    sleep(0.1)
    led.on()
    sleep(0.3)
    led.off()
    sleep(0.1)
    led.on()
    sleep(0.3)
    led.off()
    sleep(0.1)
    led.on()
    sleep(0.3)
    led.off()
    sleep(0.1)
    led.on()
    sleep(0.3)
    led.off()
    sleep(0.1)
    led.on()
    sleep(0.3)
    led.off()
    TotalWithoutVibration = 0
    TotalWithoutVibrationVibrations += 1
    val_Left = hx.get_weight_A(5)
    val_RightGain = hx.get_weight_B(5)
    val_Right = (val_RightGain * 4)
    print "(0)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
    print "WithoutVib: %s" % (TotalWithoutVibration)


while ((val_Left > val_Right) and (val_Left > 350000) and (val_Right > 350000)):
    starttime = time.time()
    sleep(300)
    val_Left = hx.get_weight_A(5)
    val_RightGain = hx.get_weight_B(5)
    val_Right = (val_RightGain * 4)
    print "(1.0) val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
    if ((val_Left > val_Right) and (val_Left > 350000) and (val_Right > 350000)):
        led.on()
        sleep(0.5)
        led.off()
        sleep(15)
        val_Left = hx.get_weight_A(5)
        val_RightGain = hx.get_weight_B(5)
        val_Right = (val_RightGain * 4)
        print "(1.1) val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
    if ((val_Left < val_Right) and (val_Left > 350000) and (val_Right > 350000)):
        print ("next")
        sleep(4)
        LeaningLeft += 4
        stop = time.time()
        total += (stop - starttime)
        TotalWithoutVibration += (stop - starttime)
        sleep(0.2)
        outf = open("sittintime.txt", "w")
        outf.write(" LeaningLeft = %s " % (LeaningLeft))
        outf.write(" total = %s " % (total))
        outf.write(" LeaningRight = %s " % (LeaningRight))
        outf.write(" VibrationsFromRight = %s " % (VibrationsFromRight))
        outf.write(" VibrationsFromLeft = %s " % (VibrationsFromLeft))
        outf.write(" TotalWithoutVibrationVibrations = %s " % (TotalWithoutVibrationVibrations))
        val_Left = hx.get_weight_A(5)
        val_RightGain = hx.get_weight_B(5)
        val_Right = (val_RightGain * 4)
        print "(2)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
        sleep(5)
        break
```

```
        if ((val_Left > val_Right) and (val_Left > 350000) and (val_Right > 350000)):
            print("next2")
            LeaningLeft += 4
            sleep(0.1)
            led.on()  # VIBRATE!
            sleep(4)
            print ("vibrate1Left")
            led.off()
            TotalWithoutVibration = 0
            VibrationsFromLeft += 1
            hx.power_down()
            hx.power_up()
            time.sleep(0.1)
            stop = time.time()
            total += (stop - starttime)
            sleep(0.2)
            outf = open("sittintime.txt", "w")
            outf.write(" LeaningLeft = %s " % (LeaningLeft))
            outf.write(" total = %s " % (total))
            outf.write(" LeaningRight = %s " % (LeaningRight))
            outf.write(" VibrationsFromRight = %s " % (VibrationsFromRight))
            outf.write(" VibrationsFromLeft = %s " % (VibrationsFromLeft))
            outf.write(" TotalWithoutVibrationVibrations = %s " % (TotalWithoutVibrationVibrations))

            print total
            break


    while ((val_Right > val_Left) and (val_Right > 350000) and (val_Left > 350000)):
        sleep(5)
        starttime = time.time()
        sleep(300)
        val_Left = hx.get_weight_A(5)
        val_RightGain = hx.get_weight_B(5)
        val_Right = (val_RightGain * 4)
        print "(3.0) val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
        if ((val_Right > val_Left) and (val_Right > 350000) and (val_Left > 350000)):
            led.on()
            sleep(0.3)
            led.off()
            sleep(0.1)
            led.on()
            sleep(0.3)
            led.off()
            sleep(15)
            val_Left = hx.get_weight_A(5)
            val_RightGain = hx.get_weight_B(5)
            val_Right = (val_RightGain * 4)
            print "(3.1)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
        if ((val_Right < val_Left) and (val_Right > 350000) and (val_Left > 350000)):
            print ("next3")
            sleep(4)
            LeaningRight += 4
            stop = time.time()
            total += (stop - starttime)
            TotalWithoutVibration += (stop - starttime)
            sleep(0.2)
            outf = open("sittintime.txt", "w")
            outf.write(" LeaningLeft = %s " % (LeaningLeft))
            outf.write(" total = %s " % (total))
            outf.write(" LeaningRight = %s " % (LeaningRight))
            outf.write(" VibrationsFromRight = %s " % (VibrationsFromRight))
            outf.write(" VibrationsFromLeft = %s " % (VibrationsFromLeft))
            outf.write(" TotalWithoutVibrationVibrations = %s " % (TotalWithoutVibrationVibrations))
            val_Left = hx.get_weight_A(5)
            val_RightGain = hx.get_weight_B(5)
            val_Right = (val_RightGain * 4)
            print "(4)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
            sleep(5)
            break
        if ((val_Right > val_Left) and (val_Right >350000) and (val_Left > 350000)):
            print("next4")
            LeaningRight += 4
            sleep(0.5)
            led.on()  # VIBRATE!
            sleep(4)
            print ("vibrate2Right")
            led.off()
            TotalWithoutVibration = 0
            VibrationsFromRight += 1
            hx.power_down()
            hx.power_up()
            time.sleep(0.1)
            stop = time.time()
            total += (stop - starttime)
            sleep(0.2)
            outf = open("sittintime.txt", "w")
            outf.write(" LeaningLeft = %s " % (LeaningLeft))
            outf.write(" total = %s " % (total))
            outf.write(" LeaningRight = %s " % (LeaningRight))
            outf.write(" VibrationsFromRight = %s " % (VibrationsFromRight))
            outf.write(" VibrationsFromLeft = %s " % (VibrationsFromLeft))
            outf.write(" TotalWithoutVibrationVibrations = %s " % (TotalWithoutVibrationVibrations))
            print total
            break
```

```
## Tervehdysta

                if (val_Left < 350000) and (val_Right < 350000):
                    break
## Tervehdysta



        hx.power_down()
        hx.power_up()
        time.sleep(0.1)
    except (KeyboardInterrupt, SystemExit):
        print "LeaningLeft: %s" % (LeaningLeft)
        print "T1B: %s" % (T1B)
        print "VibrationsFromLeft: %s" % (VibrationsFromLeft)
        cleanAndExit()
```

Appendix 6

PROGRAM 2

```
import RPi.GPIO as GPIO
import time
import threading
import sys
from threading import Timer
from time import sleep
from signal import pause
from hx711 import HX711
from gpiozero import LED
from gpiozero import Button


def writefile():
    with open("TestPerson1WithVibration.txt", "w") as text_file:
        sleep(0.1)
        text_file.write(" VibrationLeft = %s " % (VibrationLeft))
        sleep(0.1)
        text_file.write(" NotiLeft = %s " % (NotiLeft))
        sleep(0.1)
        text_file.write(" VibrationRight = %s " % (VibrationRight))
        sleep(0.1)
        text_file.write(" NotiRight = %s " % (NotiRight))
        sleep(0.1)
        text_file.write(" TotalTime = %s " % (TotalTime))
        sleep(0.1)
        text_file.write(" StandUps = %s " % (StandUps))
        sleep(0.1)
        text_file.write(" SatDowns = %s " % (SatDowns))
        sleep(0.1)
        text_file.close
        sleep(0.1)



def cleanAndExit():
    print ("Cleaning...")
    GPIO.cleanup()
    print ("Bye!")
    sys.exit()

led = LED(17)
##button = Button(3)
hx = HX711(5, 6)
```

```
hx.set_reading_format("MSB", "MSB")
hx.set_reference_unit(1)
hx.reset()
# tare (use both) channels
hx.tare()
hx.tare_A()
hx.tare_B()


VibrationLeft = 0
NotiLeft = 0
VibrationRight = 0
NotiRight = 0
TotalTime = 0
StandUps = 0
StandUpTime = 0
SatDowns = 0

while True:

    try:


#       val_LeftGain = hx.get_weight_A(5)
#       sleep(0.2)
#       val_Left = (val_LeftGain - 12500000) ## if issues with side shifting remove this
        val_Left = hx.get_weight_A(5)
        val_RightGain = hx.get_weight_B(5)
        val_Right = (val_RightGain * 4)
        print "(0)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
        channel='A',
        gain=64
        sleep(0.5)

## Tervehdysta

        if (4000000 > val_Left > 250000) and (4000000 > val_Right > 250000):
            led.on() ##Control signal for the relay to turn vibration on
            sleep(0.5)
            led.off()  ##Control signal for the relay to turn vibration off
            StandUpTime = 0
            SatDowns += 1
            writefile() ## WriteFile
            print("pass")
```

```
            while True:

                val_Left = hx.get_weight_A(5)
                val_RightGain = hx.get_weight_B(5)
                val_Right = (val_RightGain * 4)
                print "(00)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
                channel='A',
                gain=64
                sleep(1)

## Tervehdysta
                if StandUpTime > 45 and (val_Left > 250000) and (val_Right > 250000):
                    print("StandUp")
                    led.on()
                    sleep(2)
                    led.off()
                    sleep(0.7)
                    led.on()
                    sleep(2)
                    led.off()
                    sleep(0.7)
                    led.on()
                    sleep(2)
                    led.off()
                    sleep(0.7)
                    led.on()
                    sleep(2)
                    led.off()
                    StandUps += 1
                    StandUpTime = 0
                    TotalTime += 1
                    writefile() ## WriteFile
                    sleep(60)


                if (val_Left > val_Right) and (val_Left > 250000) and (val_Right > 250000):
                    led.on()    ##NotiLeft Could be function too but in my opinion makes
                    sleep(0.2)  ##the program easier to read when when the vibrations stand out.
                    led.off()   ##NotiLeft
                    sleep(60)
                    TotalTime += 1
                    StandUpTime += 1
                    NotiLeft += 1
                    writefile() ## WriteFile
                    val_Left = hx.get_weight_A(5)
                    val_RightGain = hx.get_weight_B(5)
                    val_Right = (val_RightGain * 4)
                    print "(01)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
```

```
if (val_Left > val_Right) and (val_Left > 250000) and (val_Right > 250000):
    sleep(60)
    TotalTime += 1
    StandUpTime += 1
    writefile() ## WriteFile
    val_Left = hx.get_weight_A(5)
    val_RightGain = hx.get_weight_B(5)
    val_Right = (val_RightGain * 4)
    print "(01.1)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )

if (val_Left > val_Right) and (val_Left > 250000) and (val_Right > 250000):
    led.on()    ##NotiLeft
    sleep(0.2)  ##NotiLeft
    led.off()   ##NotiLeft
    sleep(60)
    TotalTime += 1
    StandUpTime += 1
    NotiLeft += 1
    writefile() ## WriteFile
    val_Left = hx.get_weight_A(5)
    val_RightGain = hx.get_weight_B(5)
    val_Right = (val_RightGain * 4)
    print "(02)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )

if (val_Left > val_Right) and (val_Left > 250000) and (val_Right > 250000):
    sleep(60)
    TotalTime += 1
    StandUpTime += 1
    writefile() ## WriteFile
    val_Left = hx.get_weight_A(5)
    val_RightGain = hx.get_weight_B(5)
    val_Right = (val_RightGain * 4)
    print "(02.1)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )

if (val_Left > val_Right) and (val_Left > 250000) and (val_Right > 250000):
    led.on()    ##NotiLeft
    sleep(0.2)  ##NotiLeft
    led.off()   ##NotiLeft
    sleep(60)
    TotalTime += 1
    StandUpTime += 1
    NotiLeft += 1
    writefile() ## WriteFile
    val_Left = hx.get_weight_A(5)
    val_RightGain = hx.get_weight_B(5)
    val_Right = (val_RightGain * 4)
    print "(03)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
```

```
                if (val_Left > val_Right) and (val_Left > 250000) and (val_Right > 250000):
                    sleep(60)
                    TotalTime += 1
                    StandUpTime += 1
                    writefile() ## WriteFile
                    val_Left = hx.get_weight_A(5)
                    val_RightGain = hx.get_weight_B(5)
                    val_Right = (val_RightGain * 4)
                    print "(03.1)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )

                if (val_Left > val_Right) and (val_Left > 250000) and (val_Right > 250000):
                    led.on()
                    sleep(2.5)
                    led.off()
                    VibrationLeft += 1
                    TotalTime += 1
                    writefile() ## WriteFile
                    sleep(60)
                    val_Left = hx.get_weight_A(5)
                    val_RightGain = hx.get_weight_B(5)
                    val_Right = (val_RightGain * 4)
                    print "(04)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )


##############################################################
##############################################################


                if (val_Left < val_Right) and (val_Left > 250000) and (val_Right > 250000):
                    led.on()    ##NotiRight
                    sleep(0.2)  ##NotiRight
                    led.off()   ##NotiRight
                    sleep(0.3)  ##NotiRight
                    led.on()    ##NotiRight
                    sleep(0.2)  ##NotiRight
                    led.off()   ##NotiRight
                    sleep(60)    ##NotiRight
                    TotalTime += 1
                    StandUpTime += 1
                    NotiRight += 1
                    writefile() ## WriteFile
                    val_Left = hx.get_weight_A(5)
                    val_RightGain = hx.get_weight_B(5)
                    val_Right = (val_RightGain * 4)
                    print "(11)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
```

```
if (val_Left < val_Right) and (val_Left > 250000) and (val_Right > 250000):
    sleep(60)
    TotalTime += 1
    StandUpTime += 1
    writefile() ## WriteFile
    val_Left = hx.get_weight_A(5)
    val_RightGain = hx.get_weight_B(5)
    val_Right = (val_RightGain * 4)
    print "(11.1)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )

if (val_Left < val_Right) and (val_Left > 250000) and (val_Right > 250000):
    led.on()    ##NotiRight
    sleep(0.2)  ##NotiRight
    led.off()   ##NotiRight
    sleep(0.3)  ##NotiRight
    led.on()    ##NotiRight
    sleep(0.2)  ##NotiRight
    led.off()   ##NotiRight
    sleep(60)    ##NotiRight
    TotalTime += 1
    StandUpTime += 1
    NotiRight += 1
    writefile() ## WriteFile
    val_Left = hx.get_weight_A(5)
    val_RightGain = hx.get_weight_B(5)
    val_Right = (val_RightGain * 4)
    print "(12)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )

if (val_Left < val_Right) and (val_Left > 250000) and (val_Right > 250000):
    sleep(60)
    TotalTime += 1
    StandUpTime += 1
    writefile() ## WriteFile
    val_Left = hx.get_weight_A(5)
    val_RightGain = hx.get_weight_B(5)
    val_Right = (val_RightGain * 4)
    print "(12.1)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )
```

```
                if (val_Left < val_Right) and (val_Left > 250000) and (val_Right > 250000):
                    led.on()    ##NotiRight
                    sleep(0.2)  ##NotiRight
                    led.off()   ##NotiRight
                    sleep(0.3)  ##NotiRight
                    led.on()    ##NotiRight
                    sleep(0.2)  ##NotiRight
                    led.off()   ##NotiRight
                    sleep(60)    ##NotiRight
                    TotalTime += 1
                    StandUpTime += 1
                    NotiRight += 1
                    writefile() ## WriteFile
                    val_Left = hx.get_weight_A(5)
                    val_RightGain = hx.get_weight_B(5)
                    val_Right = (val_RightGain * 4)
                    print "(13)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )

                if (val_Left < val_Right) and (val_Left > 250000) and (val_Right > 250000):
                    sleep(60)
                    TotalTime += 1
                    StandUpTime += 1
                    writefile() ## WriteFile
                    val_Left = hx.get_weight_A(5)
                    val_RightGain = hx.get_weight_B(5)
                    val_Right = (val_RightGain * 4)
                    print "(13.1)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )

                if (val_Left < val_Right) and (val_Left > 250000) and (val_Right > 250000):
                    led.on()
                    sleep(1.3)
                    led.off()
                    sleep(0.7)
                    led.on()
                    sleep(1.3)
                    led.off()
                    VibrationRight += 1
                    TotalTime += 1
                    writefile() ## WriteFile
                    sleep(60)
                    val_Left = hx.get_weight_A(5)
                    val_RightGain = hx.get_weight_B(5)
                    val_Right = (val_RightGain * 4)
                    print "(14)val_Left: %s  val_Right: %s" % ( val_Left, val_Right )


## Tervehdysta

                if (val_Left < 250000) and (val_Right < 250000):
                    break
## Tervehdysta
## Hello from the library



        hx.power_down()
        hx.power_up()
        time.sleep(0.2)
    except (KeyboardInterrupt, SystemExit):
        cleanAndExit()
```

Appendix 7

INSTRUCTIONS FOR THE USER OF THE CHAIR

Intelligent furniture

Instructions for the chair
28.01.2019

This chair uses vibrations to communicate with the user with the aim to reduce negative health effects of prolonged sitting. Its program is designed to be as little intrusive as possible so that the user can chose to either follow or ignore the vibrations depending on their current situation at work or at home. Please remember to change the chair every night.

Meaning of different vibration patterns:

- The chair will greet the user by one short vibration always when the user sits down on the chair to indicate that it is working.
- After the greeting the chair will let the user to know which side of the chair is more loaded with two-minute intervals:
    - **One short vibration** = left side more loaded
    - **Two short vibrations** = right side more loaded
- If one side is continuously more loaded for more than six minutes the chair will vibrate more vigorously to let the user know it is time to shift position:
    - **One long vibration** = left side has been more loaded for over six minutes
    - **Two long vibrations** = right side has been more loaded for over six minutes
- Regardless of any of the other vibrations the chair will also give **four long vibrations** if the user has sat continuously for over 45 minutes as a reminder to take a short brake from sitting.

Appendix 8

QUESTIONNAIRE FOR THE TEST PERSONS TO FILL

Intelligent furniture

Questionnaire for the test person
12.01.2019

How many hours do you usually sit during your day on average?
How often do you usually take breaks from sitting?
Are you interested to reduce your sitting time or change your sitting habits?

Did this chair make any difference in the overall time you spent sitting?
Did this chair help you shift your sitting position more often?
Did this chair make any difference to the intervals of your sitting breaks?

How did you utilize the vibration communication the chair provided?
How did you end up using the chair?
How did you experience the chair?

- Help
- Annoyance
- No effect
- Did you end up ignoring the vibrations?

Did the chair work reliably throughout the testing?
Do you have any improvements ideas, suggestions or comments?
Could you imagine continue using the chair and its functions in your everyday life?

Appendix 9

QUESTIONNAIRE ANSWERS FROM TEST PERSON 1

Intelligent furniture

Questionnaire for the test person 1
12.01.2019

How many hours do you usually sit during your day on average?

About 7? 10?

How often do you usually take breaks from sitting?

During workday (about 4 hours sitting) every few minutes. At home maybe once in half an hour.

Are you interested to reduce your sitting time or change your sitting habits?

Yes, I would like to sit less and/or in better position.

Did this chair make any difference in the overall time you spent sitting?

Perhaps no

Did this chair help you shift your sitting position more often?

Yes, I paid attention to shift my position usually after the small vibration already, sometimes after the big vibration.

Did this chair make any difference to the intervals of your sitting breaks?

No, I think that wasn't necessarily an issue for me. If the chair would have another type of vibration to sign the time for getting up (and perhaps would give a warning if you try to sit back too soon...? :D) I would be interested to try that and see how often do I get up. My sense of time is not that good.

How did you utilize the vibration communication the chair provided?

I usually shifted to the small vibrations, on a few occasions only at the stronger vibration. I also used the two-minute time intervals to measure time and cook pasta.

How did you end up using the chair?

I used it while eating dinner and watched TV from it. due to it being a desk chair rather than an armchair it wasn't the best for that purpose, but I barely ever sit at a desk at home, and at work I have to get up every few minutes.

Questionnaire for the test person 1
12.01.2019

How did you experience the chair?

- **Help**
- **Annoyance**
- **No effect**
- **Did you end up ignoring the vibrations?**

I paid attention to the vibrations, they didn't annoy me, I tried to follow their suggestion. After the big vibration, when I shift, there is a small vibration right away though, which I think is unnecessary. If my weight was on my right for six minutes straight and I get the warning, I shift to the left. Then I am still alert, and register the small vibration signalling my weight is on the left. That might make me automatically shift back to right, to the same position I just spent six minutes at.

Did the chair work reliably throughout the testing?

Yes. I don't know about the logs though.

Do you have any improvements ideas, suggestions or comments?

-vibration for "get up!", warning if you would want to sit back within like 3 minutes, or whatever the ideal interval is.
-no small vibration right after the big one
-couch/armchair version, perhaps with more sensors (not just left and right weight).
-something to make you sit in a correct position?

Could you imagine continue using the chair and its functions in your everyday life?

Appendix 10

QUESTIONNAIRE ANSWERS FROM TEST PERSON 2

Intelligent furniture

Questionnaire for the test person 2
12.01.2019

How many hours do you usually sit during your day on average?

- 5-9h.

How often do you usually take breaks from sitting?

- 30-60minutes.

Are you interested to reduce your sitting time or change your sitting habits?

- I'd like to sit less.

Did this chair make any difference in the overall time you spent sitting?

- I shifted position more often and stood up couple times due to notifications.

Did this chair help you shift your sitting position more often?

- Sometimes yes, sometimes I ignored the vibrations.

Did this chair make any difference to the intervals of your sitting breaks?

- Little, yes.

How did you utilize the vibration communication the chair provided?

- Shifting position and standing up.

How did you end up using the chair?

- Just sitting and trying to follow the vibrations.

How did you experience the chair?

- Help
- Annoyance
- No effect
- Did you end up ignoring the vibrations?

Did the chair work reliably throughout the testing?

-Yes.

Do you have any improvements ideas, suggestions or comments?

-

Could you imagine continue using the chair and its functions in your everyday life?

-Yes, I don't know about long term would you get used too much to the vibrations