MSOS

Simon a Campo
Simon Bavinck
Jasper Faber
Teun Keusters
Art Selbach

Assignor: Prof.Dr.Ir. A.C. Brombacher

TABLE OF CONTENTS

INTRODUCTION & EXPECTATIONS

Making sense of sensors, this elective is about learning on how to work with sensors, but more important how to analyse the data. The outcome is often not the ideal image what you see in the theory. Noise is caused by different variables where you often did not think of. This noise causes the data to differ from the actual measurements. Working with sensors is not restricted to measuring and reading, how do you know if your measurements are right? And are the measurements precise enough? We hope to get more insight in the implementation of data and with solving the problems that arise during the measurement.

MEASUREMENT GOALS

Every sport is very different, but they all have one thing in common. Before starting practise the sport you first need to do a warming up. Through warming up you reduce the chance to get an injury and increase your overall performance. Your muscles warm up and the oxygen level in your blood increases. Unfortunately, a warming up often is not optimal to the person that is performing it, to solve this we want to make a system that measures if you are warmed up enough to start exercising. This causes you to not start too early with exercising where your muscles are not ready yet or to start too late with exercising where you already tired your muscles. We want to measure this because we want to create an optimal warming up, in which you get the best performance afterwards, this optimum differs for everyone and so it is not possible to do this with a standard warming up. Our system needs to know when you are warmed up enough to start exercising. In order to detect if you are warmed up enough we searched for a relation between the increased heartbeat, the increasing temperature and the EMG, this is the signal that the muscles send out when contracting them. We tried to find out whether if your muscle signal is stronger when you are more warmed up. Our system will focus on fitness because this sport does not require any sudden moves resulting in a reduced noise ratio.

DESIGN OF THE SYSTEM

Choice Of Sensors

EMG sensor

To measure the activity of the muscle we aimed at using an EMG sensor. EMG stands for Electromyography and basically measures the electrical signals that are released when a muscle is contracted. Building an EMG sensor can be pretty complex, but luckily there are some standard sensor kits available on the internet. This, combined with the fast delivery we needed resulted in only one possible sensor. The sensor we used is an EMG breakout board produced by Sparkfun and can be found on https://www.sparkfun.com/products/13027.

Heart rate sensor

Like the EMG sensor, building a heart rate sensor can be pretty hard to manage. Basically, regarding the time we had to build the sensor system, we had two possibilities: the first was buying a standard heart rate sensor, the second was using our EMG sensor as an ECG (Electrocardiography). Since the second option brought a lot of insecurities and would demand buying a second EMG, which is pretty expensive, we chose for the standard heart rate sensor. This so-called pulse sensor[II] measures heart rate by measuring the reflection of the light it sends. This reflection is influenced by the amount of blood flowing through the finger, by determining changes in the blood flowing the sensor is able to determine the heart rate.

Thermistor

To measure the body temperature of our test person we use an Negative Temperature Coefficient thermistor, which is basically a variable resistor of which the resistance changes over temperature. A big problem with the affordable thermistors that are currently on the market is their accuracy. Most of the thermistors we found have an optimal accuracy of 1°C, which is too large for our application since the change in body temperature is rather a matter of tenth of a degree. In the end we found an thermistor with an accuracy of ±0.1°C for body temperature[I]. A big disadvantage of this sensor is its non-linearity, which makes the sensor a lot harder to calibrate, however we couldn't find any relatively cheap linear temperature sensors with a high accuracy. These two considerations forced us in choosing the NTC thermistor just mentioned.

SD module

The internal storage of Arduino is not big enough to store all our measuring points, so we needed external storage to store our data. For this we used an easy-to-use SD module[III], that comes with an Arduino library. This library makes it very easy to store data on a SD card via the serial monitor.

Calibration Of Sensors

Thermistor

Since we bought a non-linear thermistor we needed a way to calibrate it. To do so we used the so called Steinhart-Hart equation[I], which goes as follows:

$$T = \frac{1}{A + B \cdot \ln(R) + C \cdot [\ln(R)]^3}$$

For this equation counts that T is the temperature in K and R is the resistance of the thermistor in Ω. A, B and C are the Steinhart–Hart coefficients and are different for each thermistor, these values can't be found in the datasheet of the sensor so we have to calculate them. First of all the formula can easily be solved for T.

$$T = \frac{1}{A + B \cdot \ln(R) + C \cdot [\ln(R)]^3}$$

To determine the values of A, B and C we first need to know the corresponding resistance (R) of three randomly chosen temperatures (T). In the datasheet[II] of the thermistor this table with the ratio of R_T and R_{25} can be found. For the sensor we used this table was like this.

R/T No.	8016	
T (°C)	$B_{25/100}$ = 3988 K	
	R_T/R_{25}	α (%/K)
−55.0	96.3	7.4
−50.0	67.01	7.2
−45.0	47.17	6.9
−40.0	33.65	6.7
−35.0	24.26	6.4
−30.0	17.7	6.2
−25.0	13.04	6.0
−20.0	9.707	5.8
−15.0	7.293	5.6
−10.0	5.533	5.5
−5.0	4.232	5.3
0.0	3.265	5.1
5.0	2.539	5.0
10.0	1.99	4.8
15.0	1.571	4.7
20.0	1.249	4.5
25.0	1.0000	4.4
30.0	0.8057	4.3
35.0	0.6531	4.1
40.0	0.5327	4.0
45.0	0.4369	3.9
50.0	0.3603	3.8
55.0	0.2986	3.7
60.0	0.2488	3.6

Since we know R_{25} (the nominal resistance of the thermistor) is 10kΩ, it's easy to calculate the resistance at a certain temperature. This gives the following values:

Temperature °C	25	35	45
Temperature K	298.15	308.15	318.15
$\frac{R_T}{R_{25}}$	1	0.6531	0.4369
R_T in Ω	10000	6531	4369

Now, to calculate the coefficients and to keep things structured we introduce three new terms: L, Y and γ for which we take the following equations[I].

$L_1 = \ln(R_1)\,, \quad L_2 = \ln(R_2), \quad L_3 = \ln(R_3)$

$Y_1 = \frac{1}{T_1}, \; Y_2 = \frac{1}{T_2}, \; Y_3 = \frac{1}{T_3}$

$\gamma_2 = \frac{Y_2 - Y_1}{L_2 - L_1}, \; \gamma_3 = \frac{Y_3 - Y_1}{L_3 - L_1}$

$A = Y_1 - (B + L_1^2 C) \cdot L_1 \approx 1.11871312 \cdot 10^{-3}$

Using this values we can calculate A, B and C[I].

$A = Y_1 - (B + L_1^2 C) \cdot L_1 \approx 1.11871312 \cdot 10^{-3}$

$A = Y_1 - (B + L_1^2 C) \cdot L_1 \approx 1.11871312 \cdot 10^{-3}$

$A = Y_1 - (B + L_1^2 C) \cdot L_1 \approx 1.11871312 \cdot 10^{-3}$

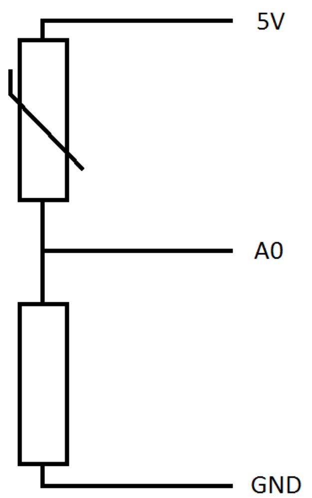
This gives the following equation for temperature.

$T_c = T_K - 273.15$

Finally to determine the temperature in Celsius we subtract 273.15 from the final answer.

$T_c = T_K - 273.15$

To be able to calculate the resistance (R) we build a voltage divider.
The figure on the right represents this:



[I] https://en.wikipedia.org/wiki/Steinhart%E2%80%93Hart_equation
[II] http://docs-europe.electrocomponents.com/webdocs/13c1/0900766b813c1d8b.pdf

The following formula can be applied to this kind of circuit.

$$U_{A0} = \frac{U_{tot} \cdot R_{const}}{R + R_{const}}$$

For which R is the resistance of the thermistor and R_{const} the resistance of the constant resistor. Now, we solve this equation for R.

$$R = \frac{U_{tot} \cdot R_{const}}{U_{A0}} - R_{const} = R_{const} \cdot (\frac{U_{tot}}{U_{A0}} - 1)$$

Since we are using Arduino we can state that the ratio of U ($\frac{U_{tot}}{U_{A0}}$) equals the ratio of A0 ($\frac{1024}{A0}$). This leads to the Arduinocode for calculating the resistance:

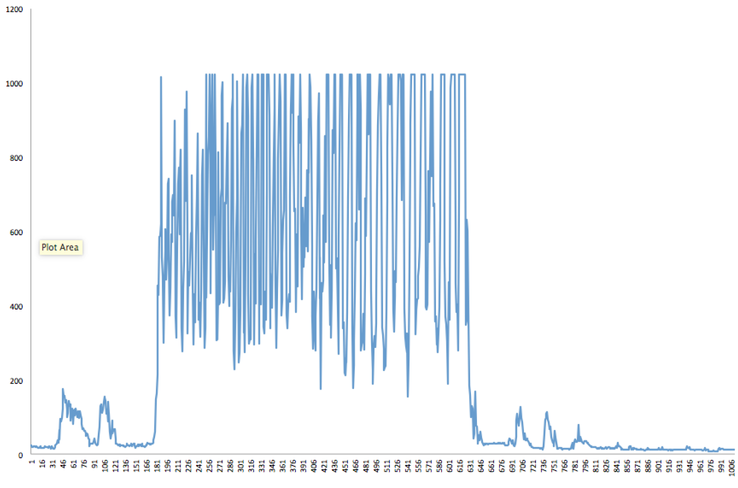
```
Resistance=constRes*((1024.0 / analogValue) - 1);
```

Now we know how to calculate the resistance in Arduino we can calculate the temperature using the Steinnhart-Hart equation.

```
Temp = log(Resistance);
Temp = 1 / (0.00111871312 + (0.00023583028 * Temp) + (0.00000008092258 * Temp *
   Temp * Temp));
Temp = Temp - 273.15;
```

EMG sensor

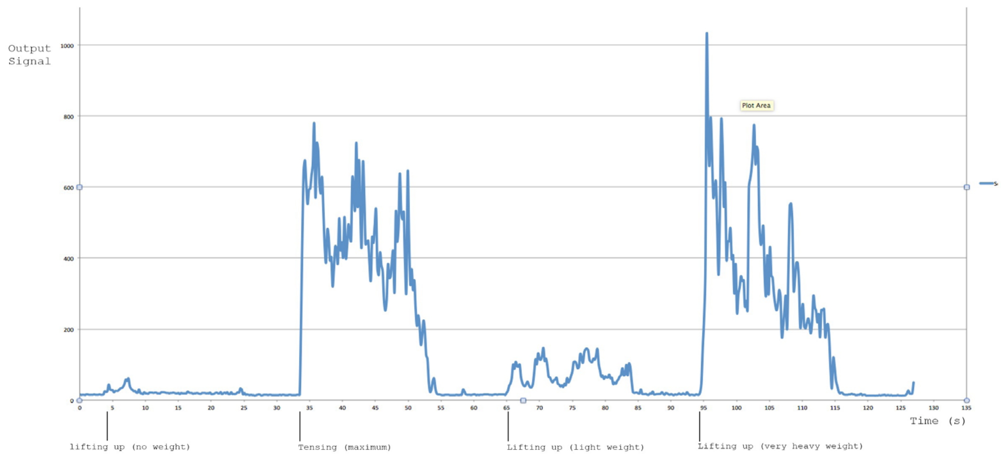
Normally muscle activity is measured in mV, however the breakout board we used in our sensor system automatically translates millivolts to an output in a range of 0-5 volts. Unfortunately no documentation is available on the factor in which this output is translated. For this project however the relative change in muscle activity is much more important than the absolute value in mV. Therefore we didn't take any action in determining the exact value of the activity, but we considered our data as ratio data and determined different types of muscle activity from relaxing to ultimate contraction.



In the dataset above you can easily analyze which data points are a result of which activity (see the graph on the next page). Because the definition of EMG signal is not an everyday signal you deal with, we had to find out what this sensor actually

measures. After a thorough analysis of the dataset we came to the conclusion that the sensor doesn't measure the actual level of warming up of the muscle. In an experiment we measured the value at rest and low intensity exercise before and after warming up (heavy exercise for 2 minutes). In the picture on the previous page you see the high peaks in the middle representing the warming up. From this dataset we can conclude that the signal doesn't has a notable change because of the warming up.

What the sensor measures is the signal that comes from the brain and tells the muscle to do something. The more power from the muscle is needed, the higher voltage the signal has. This signal only appears one time to tense a muscle. Keeping the muscle tensed doesn't result in a constant high signal, but just a peak.



Heart rate sensor

The heart rate sensor comes with a standard Arduino code that turns the output of the sensor into an heart rate. This code however filters a lot of noise and only gives a value when the sensor output is 100% reliable. Due to this filter the code only returned an heart rate a few times, and therefore is not reliable enough to use during our measurements. We edited the code to return the change in light intensity, which is a very raw kind of data. In the analyzation phase we need to determine the frequency of this signal to come to an heart rate.

on the next page the schematics of this sensor is shown

Final Arduino code

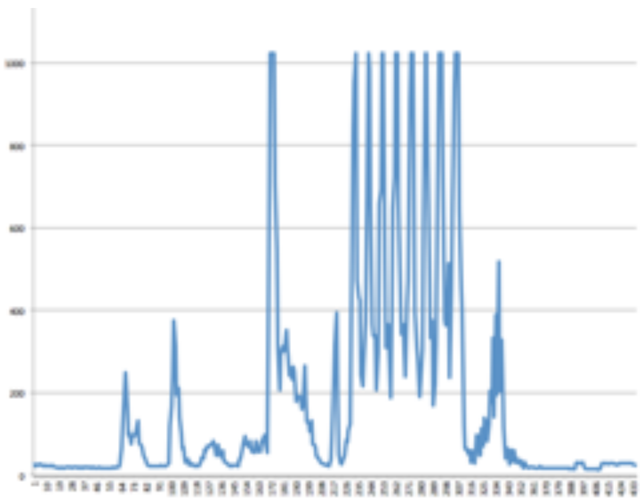
```
/* * code geschreven door studenten aan de faculteit Industrial Design, Eindhoven University of Technology * Gebruik maken van een onderdeel van een algoritme door Joel Murphy and Yury Gitman, vrij van enig eigendomsrecht, for meer informatie: zie bijgeleverde license file */

#include <math.h>
#include <SPI.h>
#include <SD.h>

char meting[] = "meting_3.txt";                                        //naam van het meetbestand

File myFile;
unsigned long ms;
float constRes = 10000;                                                //waarde van constante weerstand
int EMGpin = 2;                                                        //EMG op analoge pin 2
int pulsePin = 1;                                                      //heart rate op anologe pin 1
int thermistor = 0;                                                    //thermistor op analoge pin 0
int EMG_value;                                   //variabele om de EMG waarde in op te slaan
float temp;                                      //variabele om de temperatuur waarde in op te slaan
volatile int Signal;                             //variabele om de heartrate waarde in op te slaan

float Thermistor(int analogValue) {              //functie om de waarde in A0 om te zetten in temperatuur
  long Resistance;
  float Temp;
  Resistance = constRes * ((1024.0 / analogValue) - 1);            //weerstand berekenen d.m.v. de constante //waarde uit de
                                                                    spanningsverdeler
  Temp = log(Resistance);                                              //Neem de natuurlijke logaritme
  Temp = 1 / (0.00111871312 + (0.00023583028 * Temp) + (0.00000008092258 * Temp * Temp * Temp));
                                //Steinhart-Hart vergelijking
  Temp = Temp - 273.15;                                                //Temperatuur in Kelvin naar Temperatuur in
Celsius
  return Temp;                                                         //Return temperatuur
}

void setup() {
  pinMode(EMGpin, INPUT);
  Serial.begin(115200);                          //start seriele monitor
  interruptSetup();                              //onderdeel van algoritme door Joel Murphy and Yury Gitman, free of
                                                 copyright *zie bijgeleverde license file*

  Serial.print("Initializing SD card...");
  if (!SD.begin(4)) {                            //Check SD card
    Serial.println("initialization failed!");
    return;
  }
  Serial.println("initialization done.");

  myFile = SD.open(meting, FILE_WRITE);
  if (myFile) {                                             //Start nieuwe meting op SD card
    Serial.println("Starting new measurement");
    myFile.println("---------start new measurement--------");
    myFile.println("Time(s);Temperature(C);Muscle activity;Heart rate");
    myFile.close();
    Serial.println("New measurement started");              //Start nieuwe meting in seriele monitor (alleen ter
verificatie)
    Serial.println("Time(s);Temperature(C);Muscle activity;Heart rate");
  } else {
    Serial.println("error opening file");                   //Error als file niet geopend kan worden
  }
}

void loop() {
  ms = millis();                                            //Start tijd
  temp = Thermistor(analogRead(thermistor));                //lees de temperatuur uit
  EMG_value = analogRead(EMGpin);
  myFile = SD.open(meting, FILE_WRITE);
  myFile.print(ms);                                         //print tijd op SD card
  myFile.print(";");                                        //;
  myFile.print(temp, 2);                                    //print temperatuur op SD card, 1 decimaal
  myFile.print(";");                                        //;
  myFile.print(EMG_value);                                  //print spieractiviteit op SD card
  myFile.print(";");                                        //;
  myFile.println(Signal);                                   //print heart rate op SD card + new line
  myFile.close();
  Serial.print(ms);                                         //print tijd in seriele monitor
  Serial.print(";");                                        //;
  Serial.print(temp, 2);                                    //print temperatuur in seriele monitor, 2 dec
  Serial.print(";");                                        //;
  Serial.print(EMG_value);                                  //print spieractiviteit in seriele monitor (alleen ter
verificatie)
  Serial.print(";");                                        //;
  Serial.println(Signal);                                   //print heart rate in seriele monitor (alleen ter
verificatie)
  delay(20);                                                //delay
}
```

SIGNALS & NOISE

EMG Sensor

After the first try out of the sensor, we already noticed a very clear dataset visualising the different stages of muscle sensing we will discuss later in this report. Over all the signal contains close to zero noise from outside. We think this is because the EMG signal is very distinct signal which is not influenced by other sources. Like for example temperature, magnetic field or light intensity. There were 2 things we still need to take a look at. What if the signal intensity changes when you change the exact position of the stickers on the muscle? Should another placing lead to other values? After doing more try outs with the sensor we saw the same kind of dataset every time. In rest position, the values also remained the same. This proved that the exact placing of the stickers isn't a big noise factor. The only thing about the sensor which we weren't content with was the clipping of the sensor. When tensing the muscle very intensely or lifting very heavy objects, the sensor value clipped at 1023. Luckily we found a small 'dimmer' on the driver which functions as a resistor to lower the maximum value of the sensor.
We did a small test: Lifting something irrational heavy like a beer keg, while turning the resistor all the way up, eventually resulted in a yield of 99,8%. In addition, the lower sensor values were still readable enough. Our clipping problem was solved.

The graphs of the first clipping tests



The graphs of the first clipping tests

[I] http://docs-europe.electrocomponents.com/webdocs/13c1/0900766b813c1d8b.pdf
[II] http://www.bitsandparts.eu/Hartslagsensor-Heartbeatsensor_(Hartsensors)-p109729.html
[III] https://iprototype.nl/products/components/modules-adapters/sd-module

Heart rate sensor

As described in the calibration of the sensors, the code of the heart rate sensor normally filters out all the noise immediately and only shows the found BPM. At first the code the code looked promising since loose connections could be spotted relatively easily by steep spikes with an too high BPM.

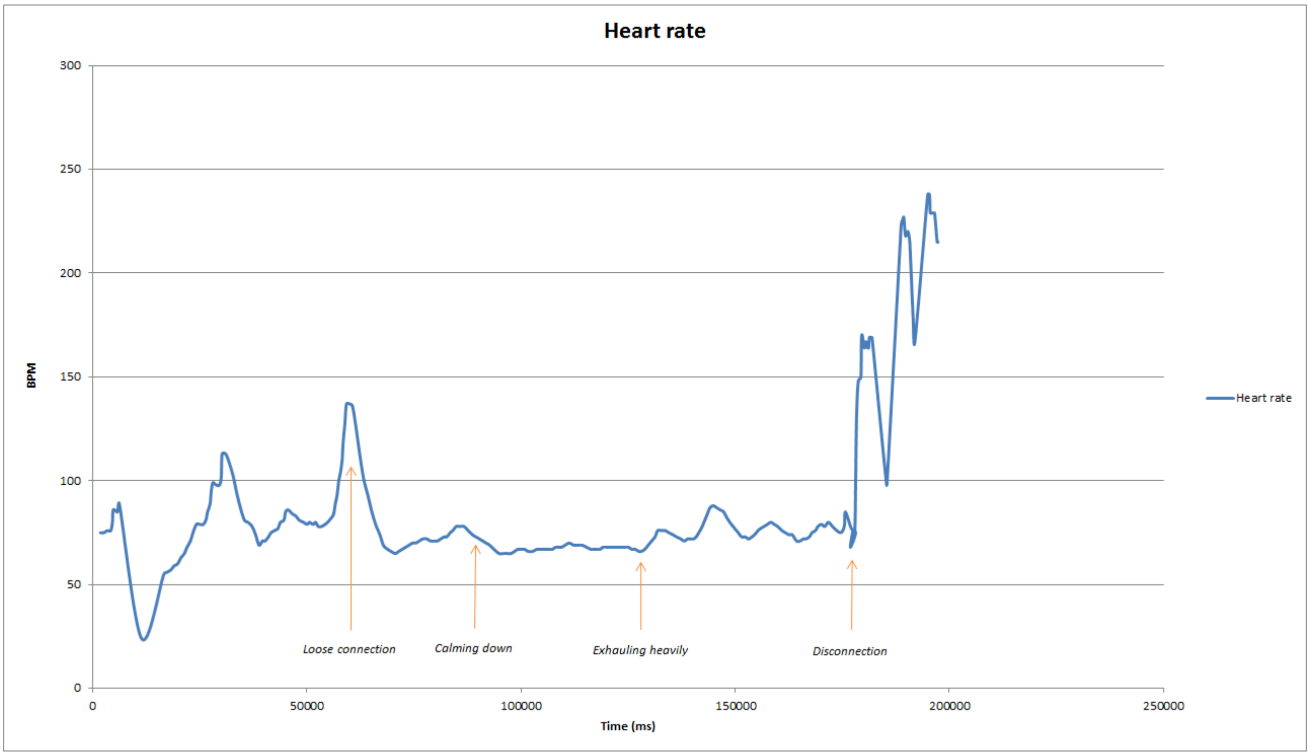


Figure 1. First measurement

When testing with the complete system however, it turned out that the sensor was sensitive to movement. The code would require a while to output a BPM and get a steady signal. When the signal was lost multiple times it wouldn't give any data which made it hard for us to understand what was happening. We found a processing code that would visualise the raw signal and help translate that to an BPM. Here it turned out that visualising the raw signal data would show much more that trusting on the BPM calculations of the code. On figure number 2 you see on the right the raw signal which shows a clear heartbeat. Since the distance between the minimum and maximums of the signal was not big enough, the program didn't recognised a BPM. On the right small box with the red line, you see all the found BPM data that was found and plotted in a graph. It shows how extreme and unreliable the code is.

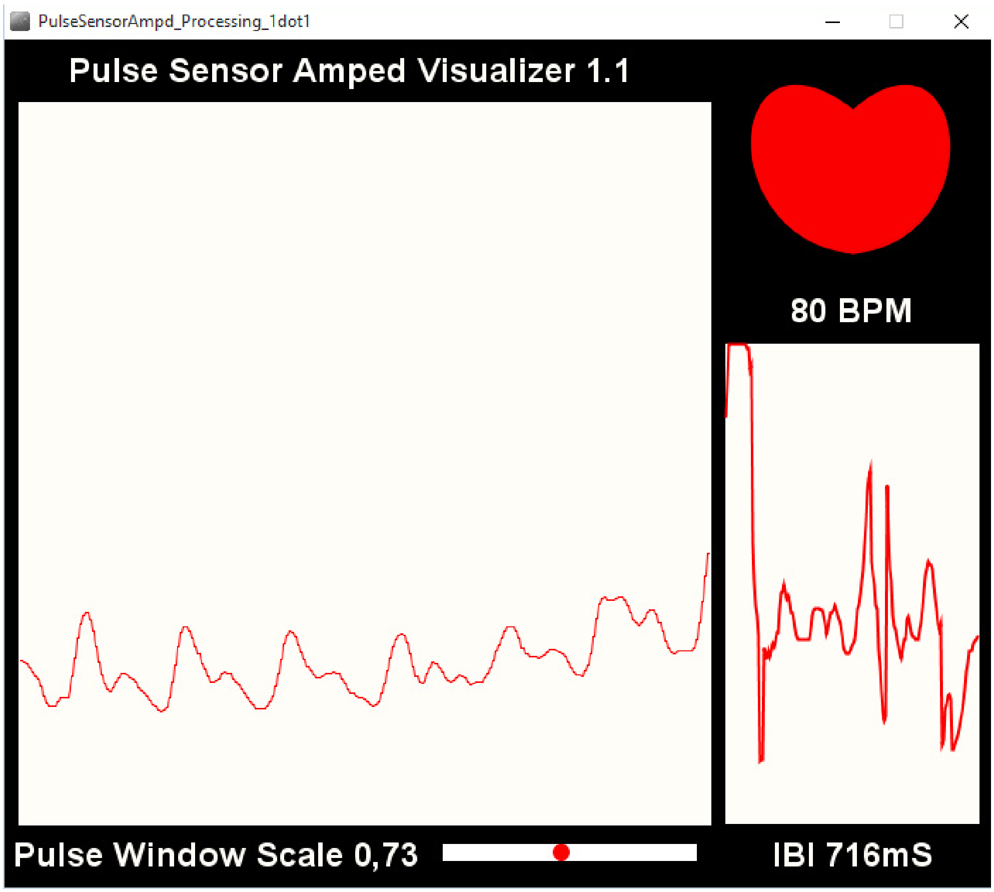


Figure 2. Left line is the raw signal data, right line in the smaller box is the calculated BPM

Unfortunately there was not quick way to see the real time data with the visualizer and store the data at same time. We decided to only use the signal output of the sensor and plot them in excel to see if they had worked and could be used in the sensor system. Also we considered that since the code would throw away data points since they would make sense for the BPM, is considered fraud. Because of these arguments we have used the raw data of the sensor in our final measurements.

Thermistor

The most important weakness of this sensor is its slowness. It takes the sensor about 40 seconds to measure a 1˚C change in temperature, which means that the output of the sensor always has some delay. Also this means that the sensor should be worn 10 to 15 minutes before we actually start measuring, to be sure that the measured temperature is equal to actual temperature. Another weak spot of this sensor are its relatively long wires. The wires (50cm) are standard multicore wires, which means that the sensor can be influenced by a lot of noise. After testing however, we found that this noise does not influence the signal in such a way that it needs to be filtered. Since the headphones we use do not isolate the ear 100% from the outside air the measured temperature can be lower than the measured temperature. From our measured signals (next chapter) we found the temperature fluctuating from 0.2 – 2 ˚C from the expected temperature, which we believe to be caused by cool air. This offset was different for each measurement, but since we are interested in the relative temperature change this was not something we needed to filter. We actually believe to have filtered a lot of extra noise by the calibration we have done.

IMPLEMENTATION

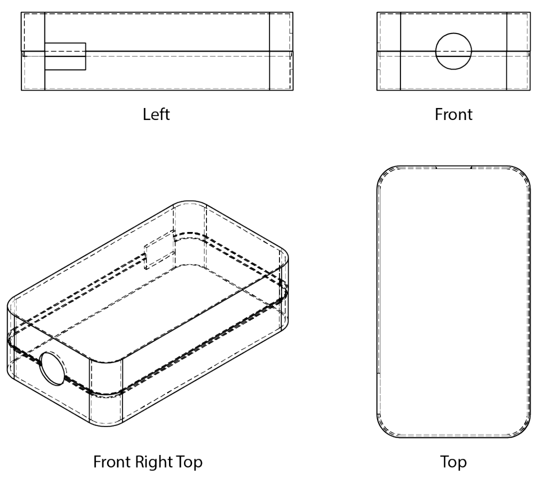
Ethics

As a designer you may not harm other people with your design.
You may not penatrate a test persons with a device.
For our system we needed to implement a thermistor, as there is a rule that you may not penetrate devices in the human body we had to carefully think about how to do this. We found out that you can use existing product that are legal to insert in the body. With this in mind we started looking for ways to implement the sensor system as effective as possible without breaking the rules. As a result we came up with the sports ear plug. This system is legal to insert into the body and gives a lot of support to the rest of the ear, making it perfect to implement the thermistor and the heart rate sensor.
To prevent accidents with the technology and the user we made a case around all the hardware. All the power comes out of this case and is further protected and secured by a belt around the arm. In this way when an accident happens with the hardware the test person is protected from injuries by the case and the belt.

Implementation of the sensors

To make sure the sensor system can be worn during sports we don't want it to interfere with the sports practiced. Therefore we want the wires we use to be as short and as close to the body as possible. Furthermore we want all our sensors to be near the Arduino we use to process our data. We decided to place the Arduino around the arm, just like runners wear their phones around their arm while running.

In order to keep all the sensors together and protected, we designed and 3D printed a case. The case is just big enough so all the electronics could fit inside and has a hole for the outgoing wires and another one for a switch. The switch was installed so the time between the start of the measuring and the desired activity was reduced. Otherwise we would have to open and close the case everytime a new measurement was done. The case was placed around the forearm since the EMG would be place on the upperarm.



EMG Sensor

The implementation of the EMG wasn't much of a trouble. As discussed earlier, we decided to implement all the hardware on the arm in an Iphone sports brace. Therefore we didn't need to take much considerations regarding the position and transportation of data. The only thing we thought that could be a problem considering the implementation are the wires. Maybe the dynamics of the body could cause noise to the sensor because the loose hanging wires course disturbances. After some tests we came to conclusion that these disturbances are hardly measured. So this shouldn't be taken into account in the implementation.

Heart rate

Measuring the heart rate could be done on the finger tips or the earlobe. Since already the thermistor would measure inside the ear, it was logical to also place the heart rate sensor on the earlobe. In this way the sensor system would remain more compact and thus more reliable and comfortable for the user. However when testing the earlobe turned out to give a too weak and inconsistent signal. When moving your head the output was completely disordered. The finger tips were more stable and the wires were long enough to relocate the heart rate sensor. With some tape we were able to hold the heart rate sensor in place at the finger tips. Still getting data from this point was hard as you can read in the information analysis

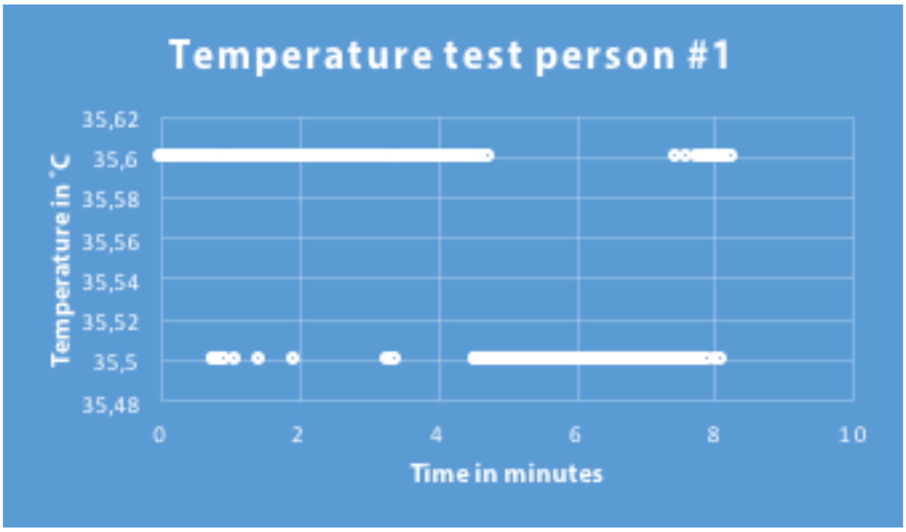
Thermistor

For the thermistor we decided to measure the temperature within the ear of our athlete. The ear is one of the best placed to measure body temperature and is really close to the arm. We implemented the thermistor in an in-ear headphone. From an ethical point of view this is a good option, because we are not allowed and don't want to pierce the human body. This headphones are designed to be brought inside the ear and therefore we can assume that it is safe to let our test persons use our thermistor in this way. For more considerations about the ethics of this project we recommend reading the chapter about ethics.

DATA ANALYSIS

Results first measurement

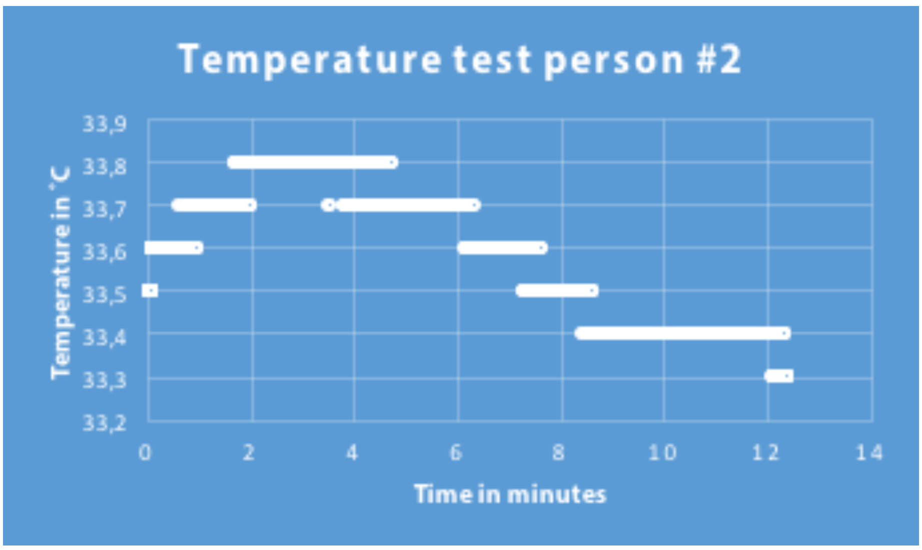
The results for temperature of the measurement with the first test person are displayed in the graph below.

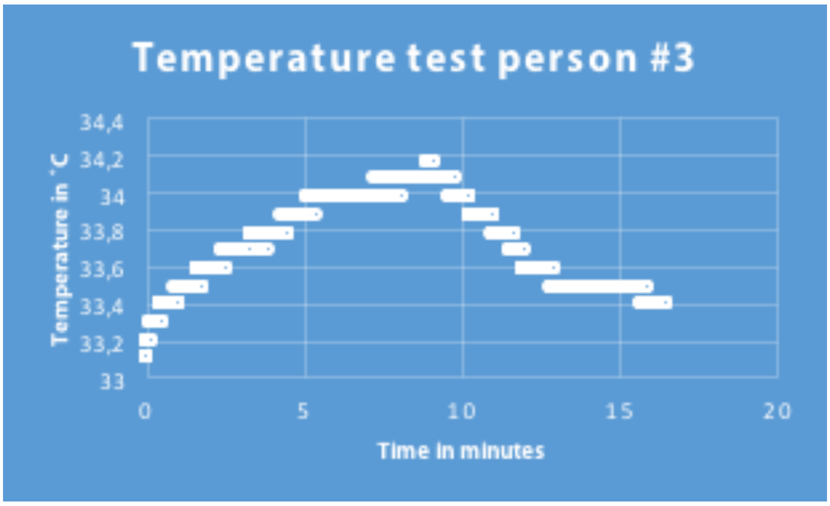


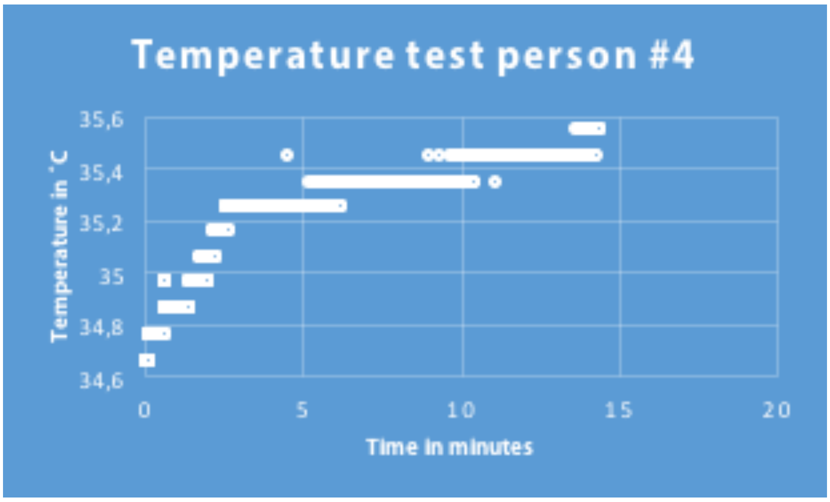
The only values found in this measurement are 35.6 and 35.5, which is caused by the Arduino settings during this measurement. The output of the sensor is rounded to only one decimal, while the change in temperature is something much more delicate. We concluded this measurement was not usable (regarding temperature) and to improve this we decided to round the sensor output to two decimals. Since the accuracy of the sensor is approximately 0.05°C we can expect a little more noise, but also more detailed data.

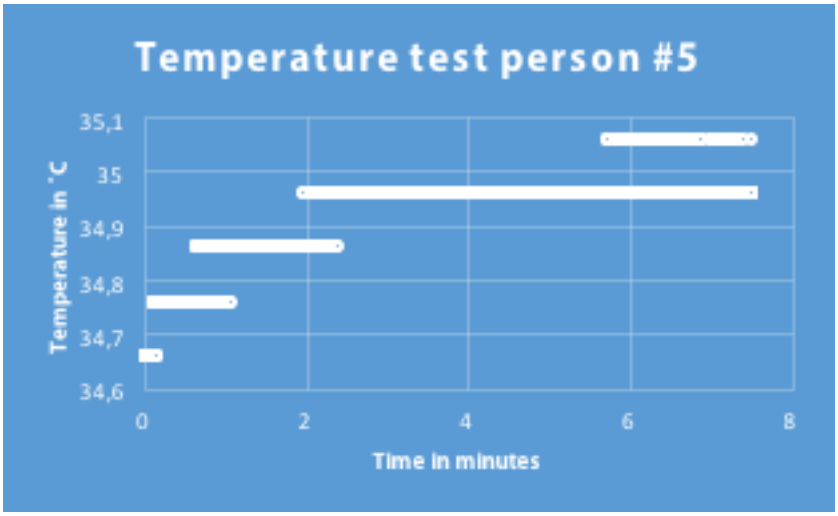
Results other measurements

The results of measurement 2 – 5 are shown below.



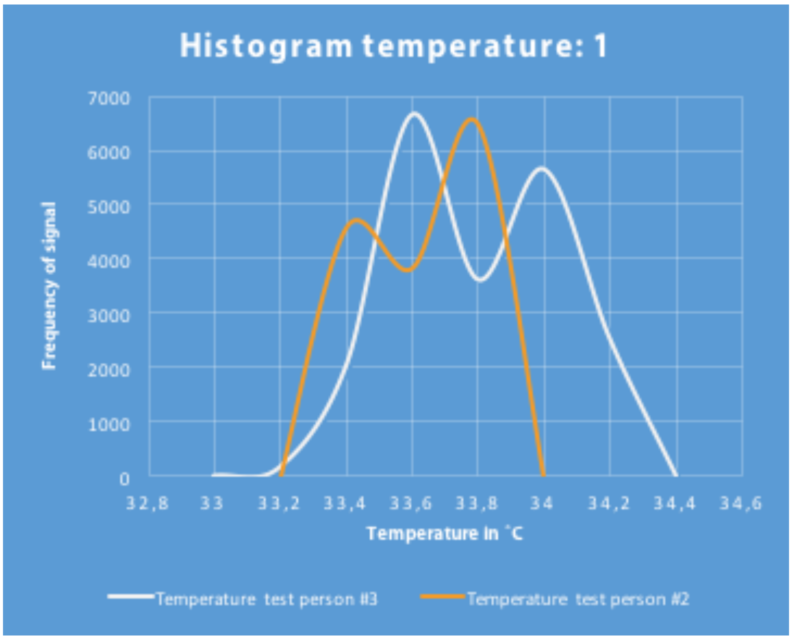

Temperature test person #3
Temperature in °C
34,4
34,2
34
33,8
33,6
33,4
33,2
33
0
5
10
15
20
Time in minutes


Temperature test person #4
Temperature in °C
35,6
35,4
35,2
35
34,8
34,6
0
5
10
15
20
Time in minutes

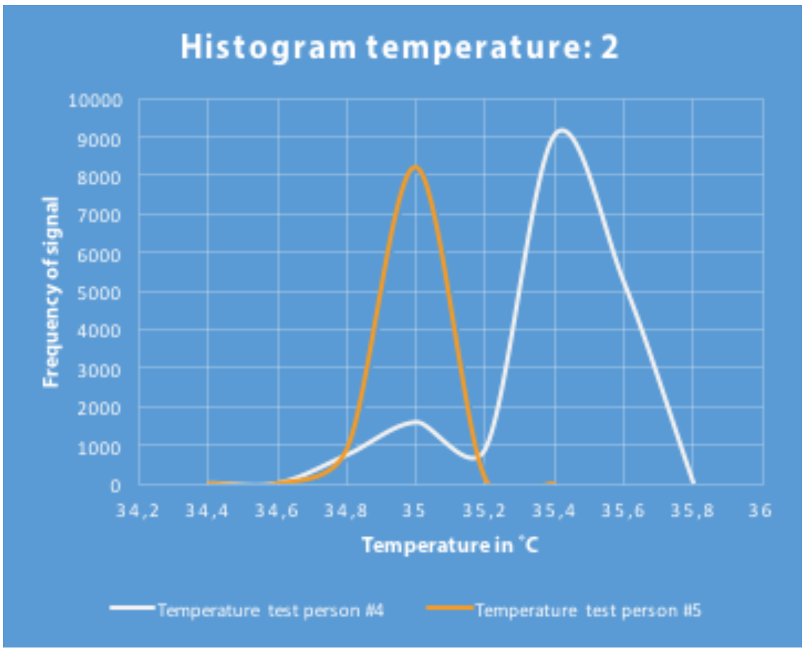

Temperature test person #5
Temperature in °C
35,1
35
34,9
34,8
34,7
34,6
0
2
4
6
8
Time in minutes

Some of these measurements contain strange aspects, which may look strange or inexplicable on first sight, therefore we will discuss them now. The first thing to notice is the slow increase of temperature, this is caused by the relatively long time that is needed for the sensor and the surrounding air to adapt the rising temperature of the human body. The graphs, which may seem like multiple horizontal line are, in reality, an enormous amount of dots, which eventually start to form a line. Knowing this we can explain the points where two different temperatures seem to be measured at the same time: At these points the sensor output is fluctuating between two different temperatures which causes the graph to display this like two lines.

Also it may catch the eye that the temperature range for none of the measurements is exactly the same, and may differ even 1.5°C. There can be multiple explanations for this, which probably all play a role to some extent. In any case we can say this difference (and also the difference with a realistic body temperature) is not caused by an error somewhere in the sensor. We can state this because the sensor was compared with a scientific thermometer after calibration and gave the same value with an accuracy of 0.05-0.1°C. A possible reason that is probably one of the main causes of this effect is the headphone, which does not make the ear entirely air tight, which enables air from the outside to influence the temperature measured by the sensor. Another influencing factor is how close the sensor is to the test persons skin, which depends on multiple things like the shape of the ear. Concluding on this, we are not able to determine the reason of this temperature difference with certainty, but we can state that the change in temperature is still accurate and therefore our measurements are still valid.
A last interesting thing in our measurements is that though we measured with Arduino's precision of 0.01°C and the sensors accuracy of about 0.05°C, the data occurs in steps of approximately 0.07 – 0.1°C. We suspect this to be the real fault of the thermistor within this system. Although the datasheet of the thermistor returns an accuracy of about 0.05°C we can't expect this to be really 0.05°C.



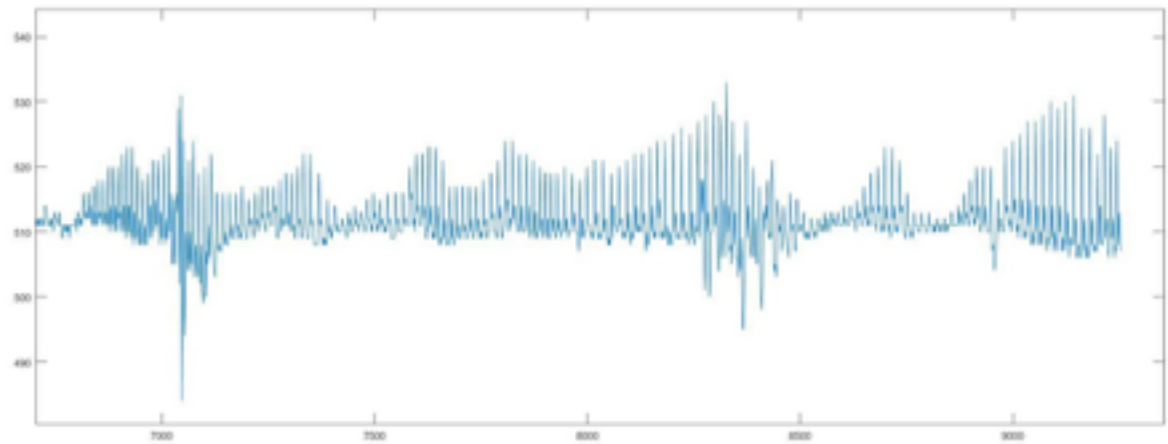
These numbers probably are reached under perfect conditions, using perfect connections and perfect equipment and therefore do not represent our situation per definition.
Especially when looking to the test person 2 and 3 we might think these lines are a bit odd. The lines start in some kind of an increasing parabolic function, but then starts decreasing almost linearly. Actually these two lines seem to represent a realistic situation perfectly: when the test persons start warming up their body temperature is slowly increasing, which causes sensor output to increase. After a while the cooling mechanism (like sweating) of the test person starts working and slowly suppresses the increase of temperature, which causes the top of the line. Then the cooling mechanism will cool down the body faster than sporting will warm it up, which results in an decreasing line that might even reach a temperature below the test persons normal body temperature. At this point the warming up was ended and therefore our measurement was. However we expect the line to rise again when our test persons would have continue exercising. We believe this to be a very plausible explanation for the odd shape of our lines, but to test it we plotted a histogram of the first two test persons (histogram temperature: 1). When looking to both lines we can clearly notice the two peaks in each line. These peaks represent the most stable body temperatures during the measurements and therefore represent the situations of a rising and a decreasing body temperature. For the other two test persons, however, we did not notice any decreasing temperature. This can be explained by many factors, for example the duration and the intensity of the warming up might not have been exactly the same for each measurement. Also the bodies of our test persons are not the same and the time until the body's cooling mechanism will start having effect might vary between different test persons. However we clearly can distinguish the temperature lines flattening which indicates the cooling mechanism is doing something. And we expect the line would have started decreasing if test persons 4 and 5 would have continued for a little while. While we are no experts within the area of warming up, the question for us raises whether the warming up of test person 4 and 5 was sufficient enough. We also plotted an histogram of the last two test persons which is 'Histogram temperature: 2'.



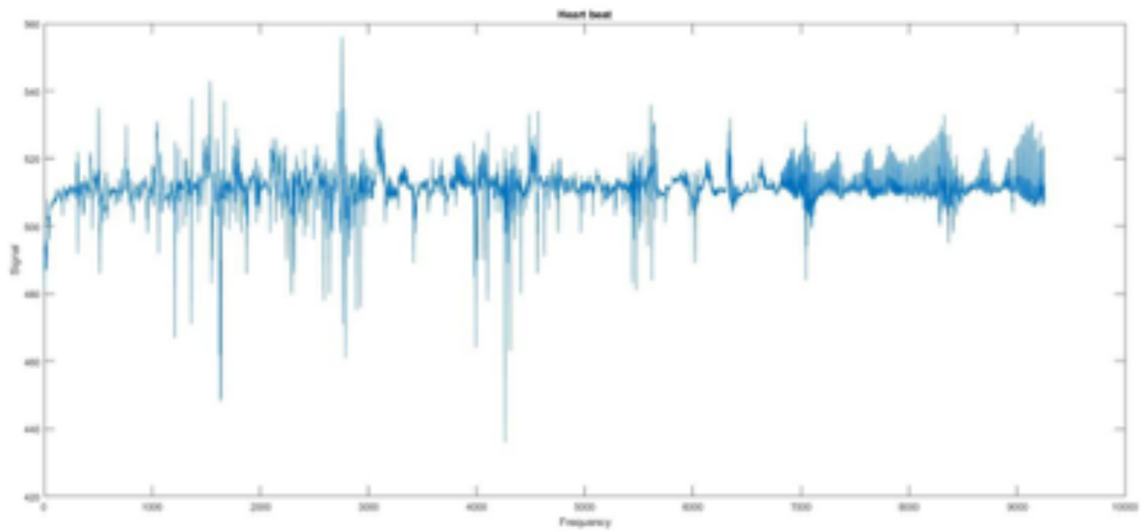
When looking to this histogram we can distinguish one big peak per test person. According to what we stated earlier this indicates no decreasing temperature, which is confirmed by graphs plotted earlier. There is however a little peak for test person #4 distinguishable at 35°C. This peak does not indicate a slight decrease in temperature, but a unexplainable variation in the speed of the rising of the temperature. This change in speed is not a big issue and is also present in the other histograms, but due to the absence of a second peak, this peak is enlarged.

Heart Rate

Unfortunately we only could analyse two of the final full sensor system test. This was because the code returned the wrong signal and it took us two measurements to figure out this was the case.

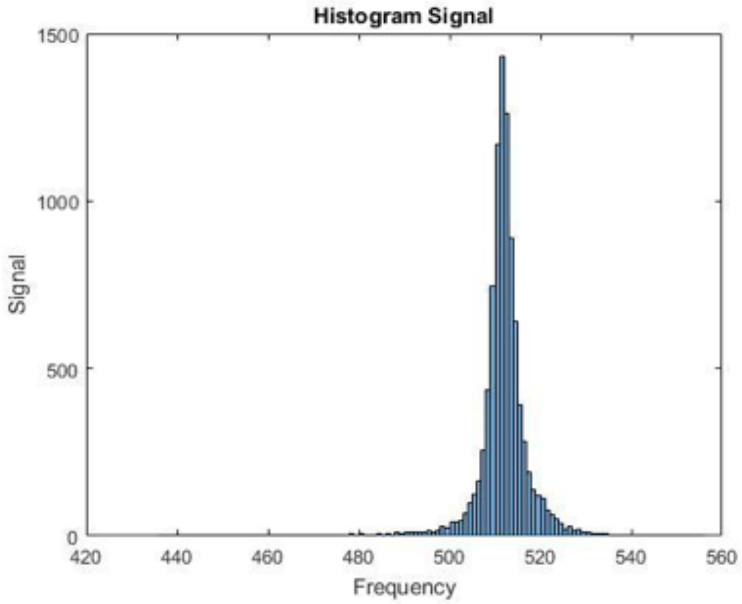


Not we've got our first measurement in the full sensor system, it is time to analyse it. In the first 2/3 of the graph, steep peaks are present and zooming more in depth to them shows it is not a heartbeat at all. This can be determined since a heart peak has a clear appearance of a high peak with a smaller peak next to it. In the first 2/3 of the data, this pattern cannot be found. Zooming in to the last 1/3 of the data shows something that looks more on a heartbeat. We tried to filter out the peaks but it showed to many gaps in order to find a proper BPM.

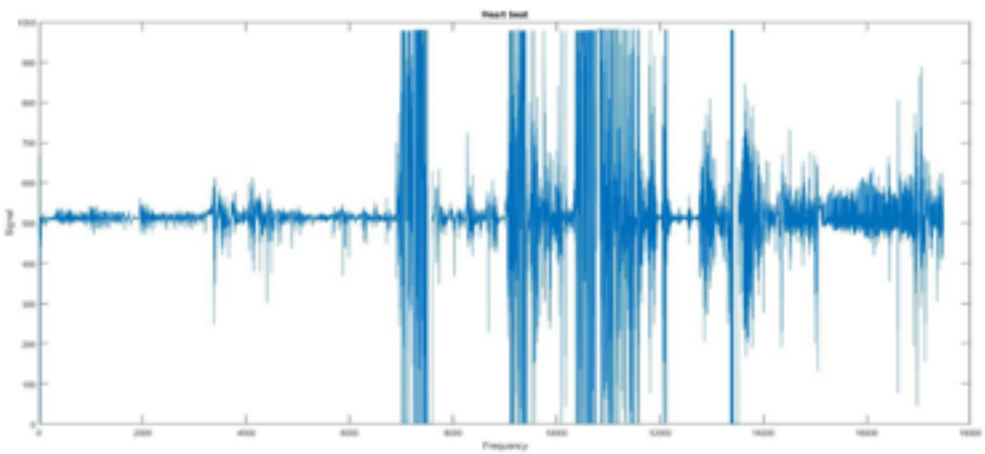


After installing and running several ECG heart rate Matlab programs, we also came to the conclusion that the sample rate wasn't high enough. The data showed up only once every 45 milliseconds which means a frequency of 22 Hertz. Normally they use a sample rate of 100 Hearts for these programs. This probably has to do with the code of the other sensors causing a delay, since the heart rate sensor alone would get a lot more data in the same time.

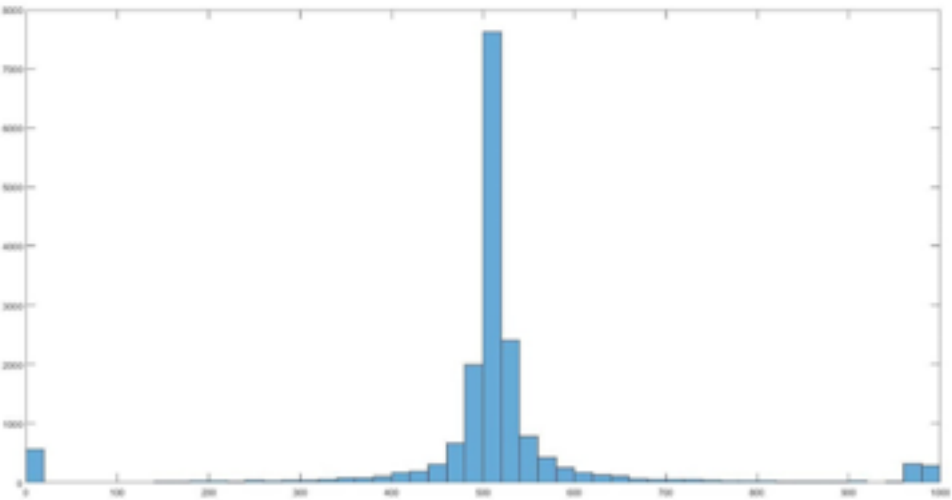
Since standard solution weren't an option, we tried to analyse the data ourselves in Matlab starting with an histogram. It was promising showing no clipping and a clear hyperbole is visible in the bars of the histogram.



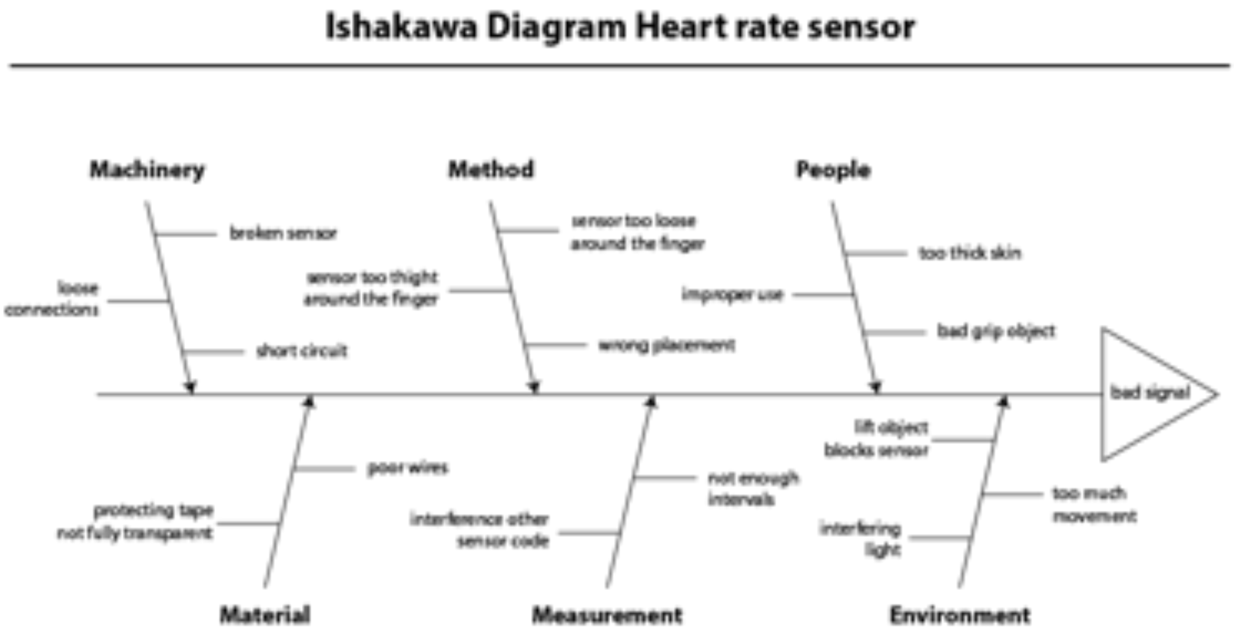
The average value of the signal should lay around 512 when looking into the Arduino code and it was at 511. This meant that the sensor itself worked well, but still data could still be distorted by the extra movement the user made by lifting objects.



The second measurement shows something went really wrong at a certain point and also from the histogram it shows clear clipping in both ways. This shouldn't be possible and also when analysing the data more closely, it showed no signs of a heart rate at all. Still the average lays around the 512 area, but this is probably because the clipping in both ways counter each other out.

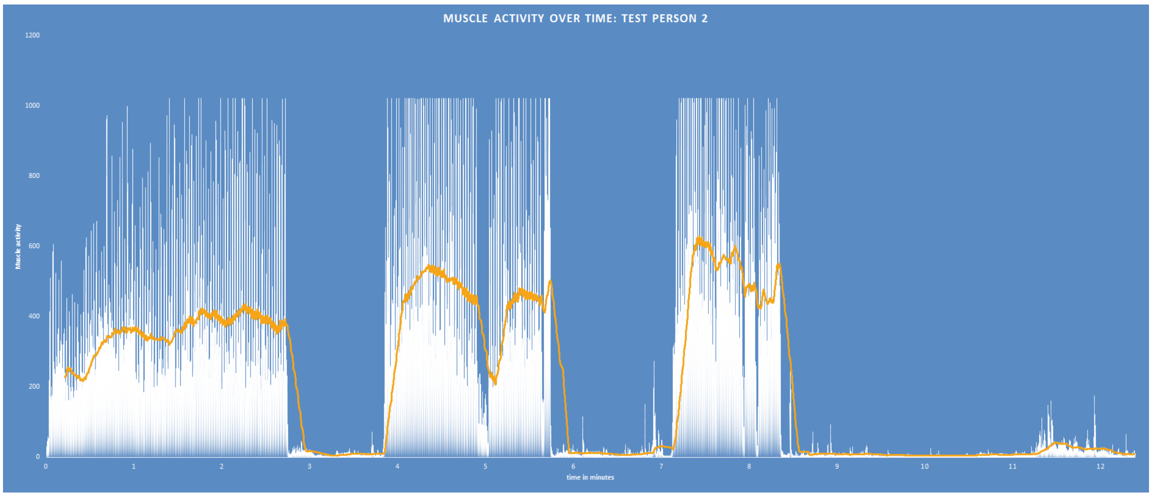


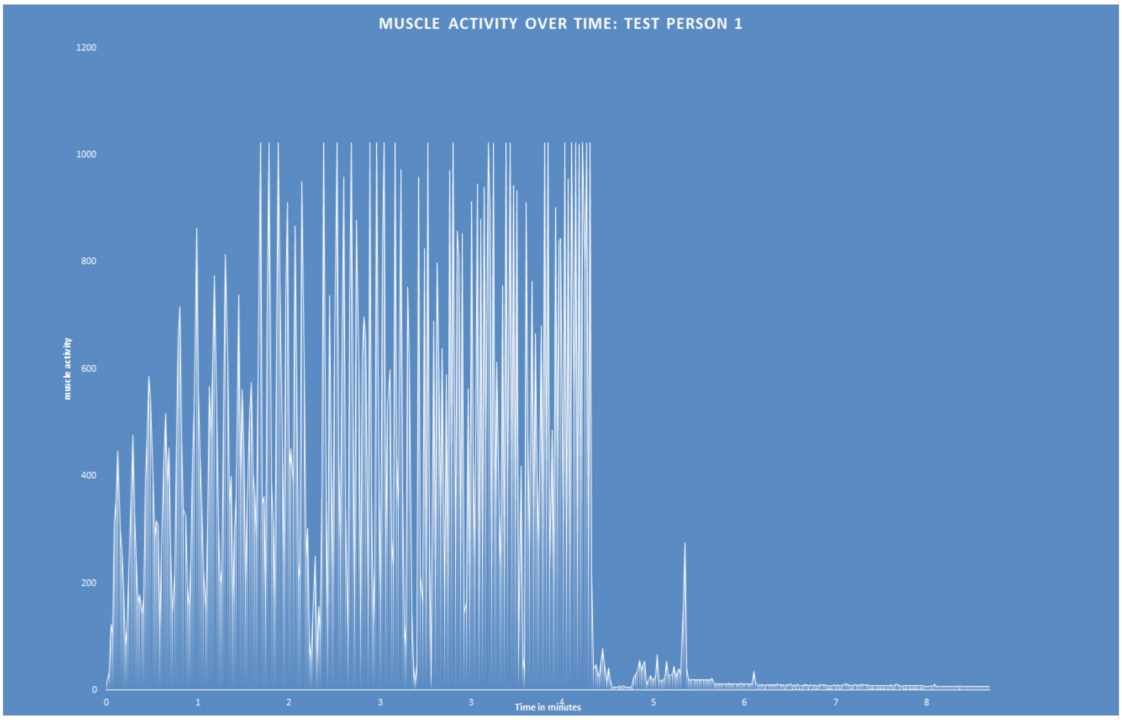
Since this could be the consequence of a lot of problems, we determined an Ishakawa Diagram (shown down below). Probably the main issues lay in the connection from the sensor to the figure tips, the rough movement of the user and the wrong frequency rate

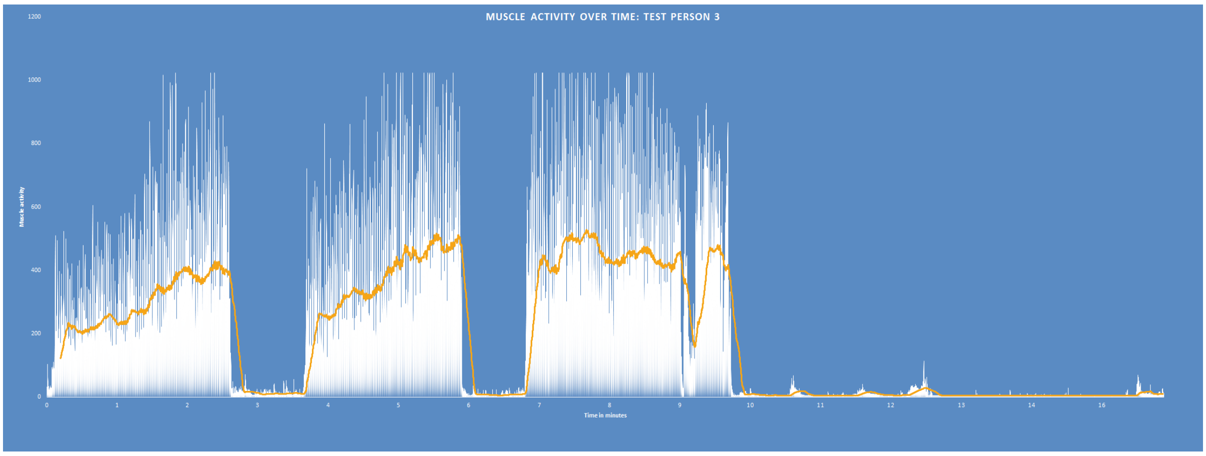


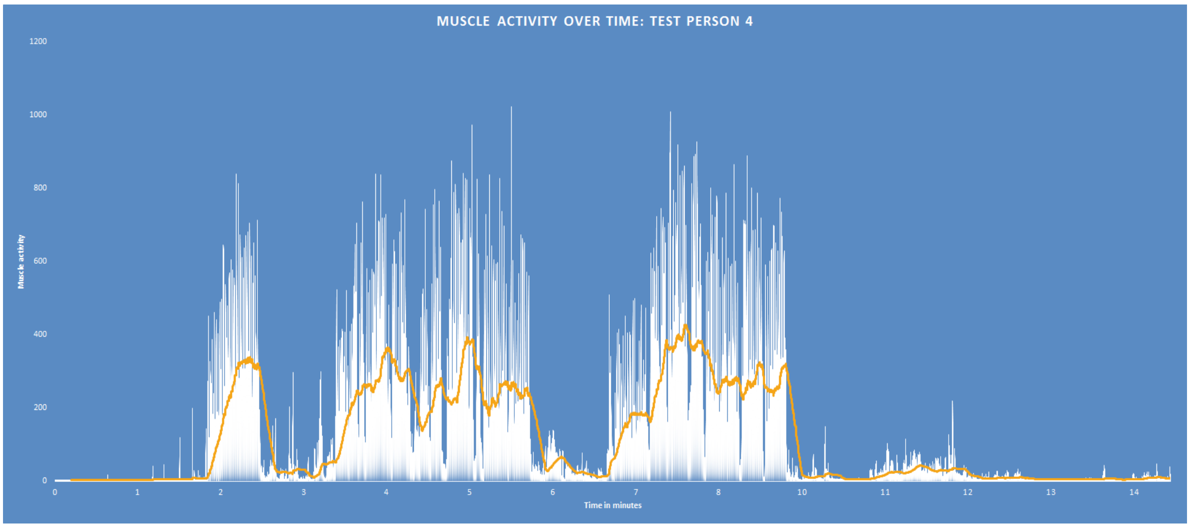
EMG Sensor

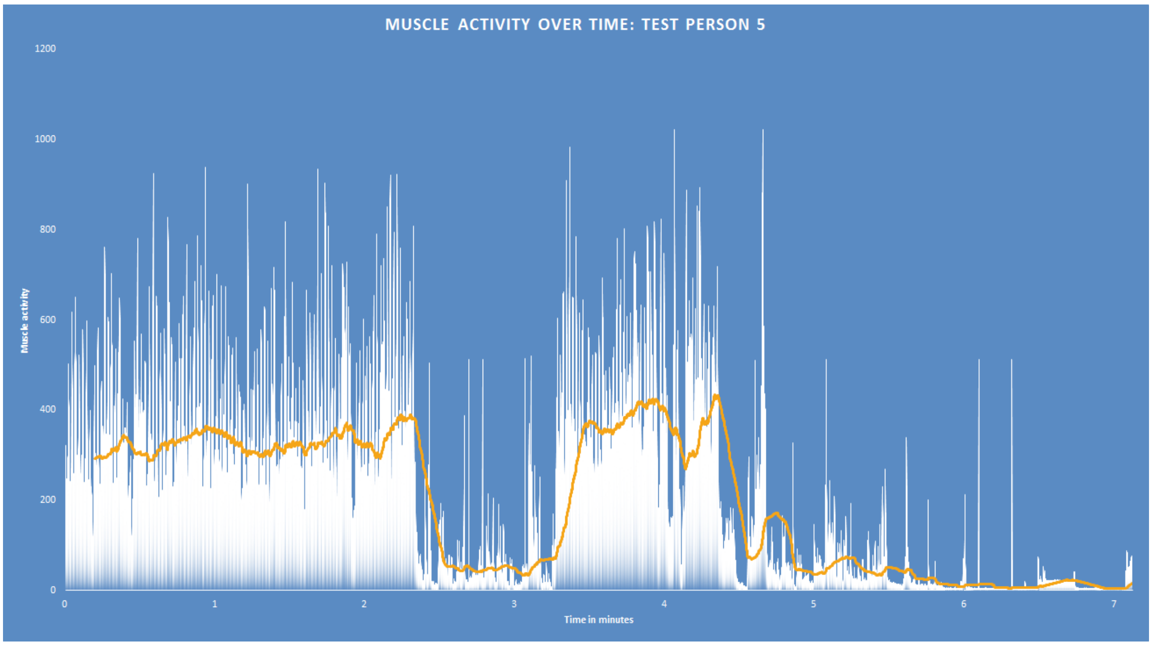
Besides measuring the temperature, we also measured the EMG values of the muscles during the exercise cycles. See the graphs below for the results





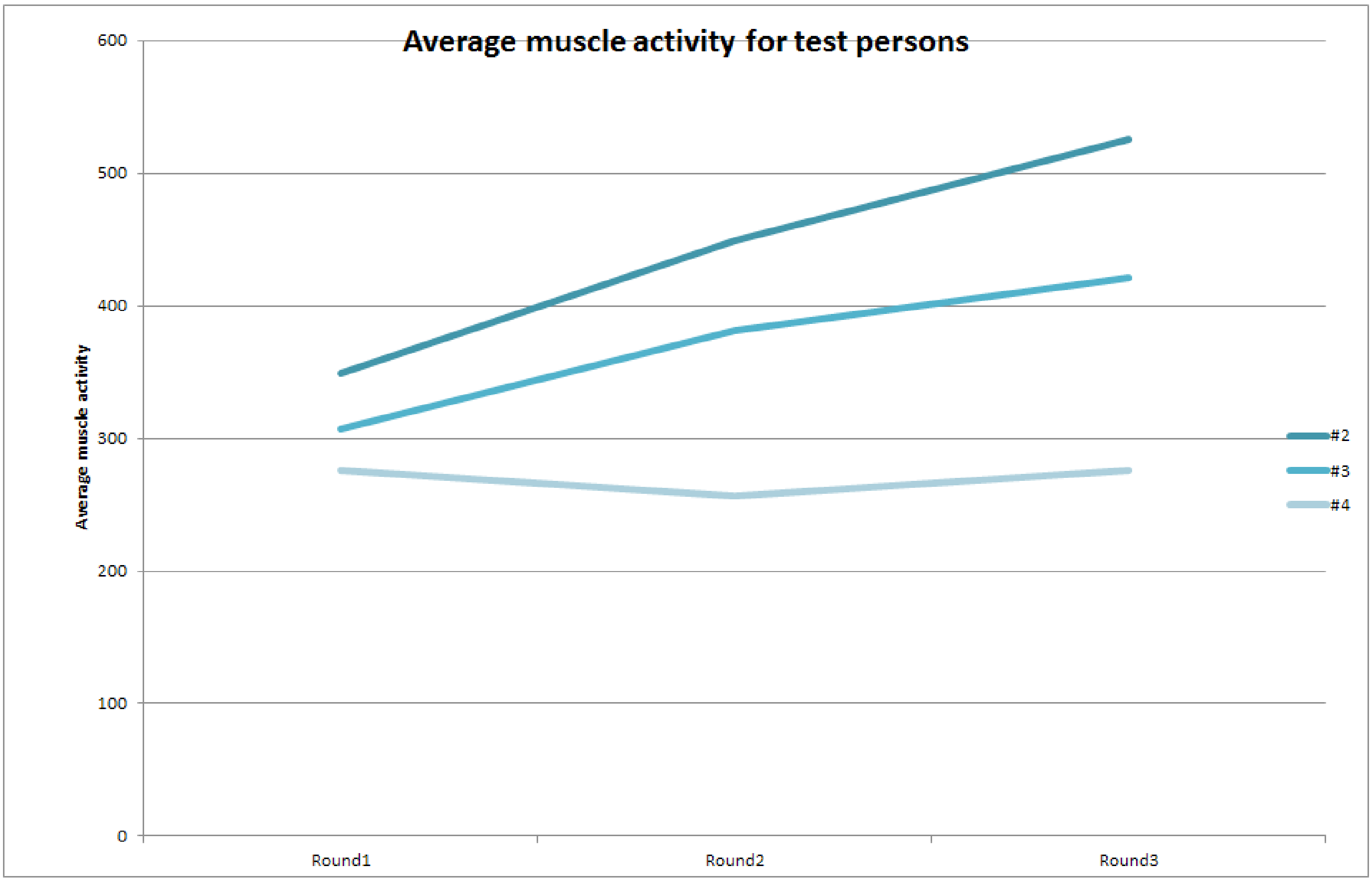

MUSCLE ACTIVITY OVER TIME: TEST PERSON 3
Muscle activity
Time in minutes


MUSCLE ACTIVITY OVER TIME: TEST PERSON 4
Muscle activity
Time in minutes


MUSCLE ACTIVITY OVER TIME: TEST PERSON 5
Muscle activity
Time in minutes

In these graphs it is very easy to distinguish the difference between the exercise and rest cycles. Our goal is to find a relationship between output signal value (OSV) and the start of exercising and OSV at the end of exercising. Therefore we can say that test person #1 and #5 aren't suitable to take into account when analyzing the data, because they went through too little exercise cycles.

While analysing the graph, there can be seen that an immense fluctuation appears during the exercise. This is very easy to explain as the exercise consists of the muscle tighening and loosening all the time. To clarify the graph, we decided to use an moving average of with a high period (400) to create a sort of average line through the fluctuation. This gives us a more evident view of whats happening. From the moving average line we can observe a possible relationship between exercise time and OSV. It seems that, the longer you exercise, the higher the OSV will become. Nevertheless the graphs aren't obvious enough to ensure that. Therefore we calculated the average value of every exercise cycle per test person, and displayed it in a graph to visualize what is happening.



As can be seen evidently, the OSV's of person #2 and #3 definitely increase over time. While looking at person #4 we only see an increase in the last 2 cycles. We think this is because the first cycle took relatively shorter then the other two cycles, which means we can't build on that measurement cycle.
This leaves us with the final conclusion that the OSV of the EMG has a relation with exercise time, because in all the 3 relevant tests we can observe an increase of the OSV.

When looking at the measurement graphs there isn’t a notable difference between before and after the exercise when looking at the OSV during rest. So for future implementations of this system we have to take into account that only during exercise measuring has it’s value.

For more data analysis in the future it would be better to change the exercise schedule. The exercises were too heavy in a too short amount of time. For a better dataset it would be convenient to have more cycles with a shorter period. In this way we can prevent that the test is exhausted before his body is even warmed up.

CONCLUSION

The relations we expected when we started this project were mainly related to heart rate. For example we could imagine a relation between heart rate and the slope of the temperature line. Also we expected a correlation between muscle activity and heart rate. However we didn't really expect any relations between muscle activity and temperature. Since the measurements with the heart rate sensor failed we are not able to investigate any correlations with heart rate, so we are forced to look into a combinations of body temperature and muscle activity.

We already found that for temperature counts that it will increase when a person starts exercising, but after a while will start decreasing. Then, we expected, when a person is warmed up properly the temperature will start increasing again. Please note that this last statement is an expectation and an hypothesis rather than a measured fact.

We also found the muscle activity to grow slowly over time, which definitely is cause by the warming up of the muscles. From our measurements we expect the muscle activity to grow also after warming, but then again, this is only a hypothesis. We expect this because some of our test persons were already pretty tired after warming up (and even experienced some muscular pain afterwards), while the muscle activity was still increasing as was the clipping of our sensor.

At this point we already can conclude that there can hardly be found any correlation between body temperature and muscle activity, since the lines of the measured data have a totally different shape that cannot be related to each other in any kind of way, without introducing more variables. However, possibly there is a correlation between this two variables after warming up.

To conclude this assignment we think the failure of our heart rate sensor was a petty for the whole project, since this sensor brought the connection between all our measured data. We, however, believe that we found very interesting data for the temperature sensor and we were happy to explain this typical behavior. We started this project with a lot of ambition and in a relatively short amount of time we were able to produce a completely working sensor system, which was really satisfying. The data measurements and data analysis could have been more elaborate, but due to our complex sensor system we were not able to focus more on this, therefore for a next project it could be wise to start with a little less ambition so we can focus more on the main objective of the course, in this case analyzing data. This said, we also believe we learned a lot from building this system in a short amount of time.

REFLECTION JASPER FABER

I chose this course because according to my vision, I want to design intelligent interactive systems which react on their environment. A smooth working sensor system is one the key elements in convenient interactive systems. I wanted to be aware of the sensors which are available nowadays, have the knowledge to pick the right sensor for my design, and have the skills to implement, analyse and tune the system. In my opinion the assignment met my expectations and I'm really glad with the final result.

During the first lessons we mostly had to listen to our teacher, explaining us different parts of the theory behind the building of a sensor system. I think this was really valuable because this knowledge has proven itself to be necessary while building the sensor system.

First we started of with a small brainstorm about the subject of our sensor system. I never knew before that the target groups for human sensor systems is so small. It was really eye opening to know that the target group to design for on the area of health is so small. I learned that a big thing to consider is ethics when working on human attached sensor system. While brainstorming we focussed on the sports section, and already really soon we all agreed on the area of measuring and analysing the warming up.

I think it is funny to see how easily you can make wrong choices in your sensor design. At the begin we wanted to measure the muscle diameter in order to determine the rate of warming up. During the presentation we got the feedback that it's going to be really hard to implement this kind of sensor. Muscles are very shaky and will almost surely disturb the accuracy of the sensor. We got the feedback to look at an EMG sensor. After evaluating our choice we decided to follow the feedback and leave the muscle diameter behind. I'm really glad we did this because the EMG sensor worked flawlessly. An improvised muscle diameter sensor wouldn't have been working out.

Because of our high ambitions, we had 3 sensors to work with. This gave us individually the opportunity to work out an entire sensor part. This was a valuable learning experience as you walk through the whole process yourself. Because I already have been working with an temperature sensor before and the EMG really got my interest, I decided to take the EMG sensor on me.

In the report there can be read that the sensor was relatively easy to work with. Due to the breakout board, no external calculations were necessary and according to our design goal, calibrating the sensor to physical units wasn't necessary either. This was, on the other hand, a nice opportunity to explore the rules of tuning a sensor. I learned to solve clipping problems to obtain a yield as high as possible, I learned how to transport data from the sensor into a datasheet using arduino and I learned how to do several analysis to obtain information out of the data by making logical graphs and histograms. Also I learned how to consider external noises and how to get rid of small technical noises using for example the moving average technique.

In conclusion I can say this course has made me more aware of the world of sensors, I achieved indispensable knowledge about creating a sensor system and due to practical application I trained skills to build one my own. In my future assignments I will be building intelligent systems. I hope to have the opportunity to apply everything I've learned from this course, and to grow more and more into designing better and better functioning systems including sensors.

REFLECTION SIMON BAVINCK

When I started the elective I wanted to focus on getting more interaction with my user, to do this I need feedback from the user. In the prototypes I have buildin the past, there were not a lot of sensors present, this means I was not able to get feedback from the user with this systems. Therefore, I wasn't able to create a mutual interaction. By following the elective Making Sense of Sensors I wanted to get more knowledge and skill in the use of sensors to get feedback from my users.

During the assignment I did two main learning activities, following the lectures of the assignor and creating and using a sensors system. At the start the lectures taught me the basics of sensors, and sensor systems, later on in the course the lectures focussed more on data analyses. The second part of the elective, creating the sensor system and analyzing data were very useful since it gave a direct opportunity to implement the theory we learned during the lectures.

On forehand I assumed that the course would mostly focus on explaining and discussing different sensors, however we did not focus to much on this. A simple explanation for this is that the field of sensors is changing so rapidly that the sensors of today can be outdated in a month. Instead of information on specific sensors we were given information on sensors in general and how to implement them. This was, looking back, way more useful.

The first thing I learned while creating the sensor system was sensor selection. It was hard not to get lost in the huge amount of sensors which is available since, I have little to no experience with finding sensors. My teammates helped me with defining the right constraints for sensors. This skill is useful for me, since it helps me to create the right kind of interaction with the right accuracy. Another important learning achievement was getting the right data, this includes calibrating sensors and analyzing sensor codes to find out which parameters they are actually collecting. I think this is useful for every interaction designer since it is very important to know which parameter you are using to base the interaction on.

The last important learning achievement was the data analyzes. In the lectures we learned some techniques which on itself were pretty straightforward. These techniques were very valuable in finding out what the collected data actually meant. This skill of analyzing data is very useful for me since I am currently working in a more researched based project. I think it is very important to look at collected data from different angles, especially when doing research.

Al these learned elements combine in the creation of interaction, as I stated in the beginning of this reflection I wanted to enable myself to create this interaction in my design. I think I have achieved a good basic knowledge of sensors to create this interaction. The next step is to implement this in a design.

REFLECTION ART SELBACH

I had two main reasons for choosing this elective. The first one was that part of my assessment feedback was I needed to improve in integrating technology. And because I noticed myself that I had a backlog in working with technology and understanding the results. My second reason was that I am focused on the health side of design For this type of design the way of interaction between the user and the product is very important. To stimulate this interaction I wanted to learn more about sensors, in my vision they act like the bridge between the user and the system. This fits perfectly in working with sensors but as I had little knowledge about these I hoped to learn more about these in this elective. At the start of this semester I had few programming experience. Last year I did not see this as a problem but during this semester I saw that my lack in programming skills could lead to not understand how the system works. Last semester the systems were very basic so an error in the program was easy to be solved. Within this elective I hoped to gain knowledge in reading and especially understanding the codes and the output the sensors give. When you understand these better you can detect the problems much more easy. And you can validate the outcome with your own experience. The second part I wanted to focus on was working with hardware. Programming is one part of the electronics but making it into a compact system is another very important aspect. For this you have to understand the functions of each part. At last I wanted to analyse the results. Within my previous project we got a lot of different data but we did not know what to do with this data. The main reason was that we did not understand what the system was measuring and what all the output meant.

During this elective we started from thinking of a sensor system to building the system and especially analysing the data. In the beginning for me it was very hard to search for the right sensors, we all agreed pretty fast on building something for warming up. But what actually happens with a warming up? It was easy to get lost in choosing between all the different sensors. There are two different variables that are pretty obvious, the increased heart rate and the increasing temperature. But in order to dive a little bit deeper in the elective and not only sticking at measuring and analysing these two variables we wanted to also research if there is a relation between the muscle signal and the more warmed up you are. This gave our elective an whole other perspective. It was not only about working with the sensors and analysing their data but also about finding a relation, comparing the data sheets of the sensors. Within this elective I learned how to apply sensors in an existing system. Together with Jasper I made the in-ear thermistor. For this sensor we needed to carefully apply it into the earplug in order to not transgress the rule of not putting sensors into the human body. As this rule sound quite logically I never thought of this before, that you are not allowed to do that. By discovering these rules I now got much more depth and reference in what you are allowed and what you are disallowed to do. When the elective advanced we had problems with the heart rate sensor. For some reason the sensor was could not find a heart rate on our bodies. In the beginning we could not find a problem but after combining the code with the corresponding Processing code I saw something very interesting. This Processing code gave a very nice heart rate signal, although this code runs on the same system as the BPM code, this code gave a nice looking signal. But still the BPM code did not recognize a heart rate. As I never worked with Processing before I did not really understand the program in the beginning. But as the elective proceeded I learned to get familiar with the language the programs have. Adding processing to the Arduino code can help you to get a much better image of what the sensors actually reads. When looking at the Arduino out put you only see numbers from 0 to 1023. Seeing all these numbers does not tell much but when adding a code that translates these numbers to a graph can help you to get a much clearer understanding of how the sensors works and what is wrong with the measurements. I learned to find the right constraints in a sensor is vital to the quality of your output. Changing these constraints can have a big impact on the accuracy of your design.

I learned a lot of different tools for analysing and ordering the data. When something is wrong with the output I now know how to apply an Ishikawa Diagram to get into the sensors and trying to discover where the error occurs. This can help me a lot in the future, where in the past an error in a system was a bit vague and I did not know where to look I now have a clearer structure of how to discover and solve the error in the system. For me it is now much clearer how sensors actually work and where the different output depends on. In the past I would search for the problem within the system itself but I understand that different noise factors and clipping can change the end result by quite a lot. Overall this elective helped me very much in understanding the process from a measurement to the output on a data graph. Getting this insights on sensors will help me very much in discovering and understanding, even prevent the problems that may occur when working when working with systems that retrieve information about different variables.

REFLECTION TEUN KEUSTERS

Most of the time, when working on my projects, I am not really aware of the (im)possibilities regarding sensors and actuators. For example within my last project I chose the wrong actuators for my driving mechanism, which almost caused my project to fail. My lack of knowledge and experience within this area drove me to choosing making sense of sensors, with the hope to improve my knowledge regarding sensors. Also I wanted to learn more on measuring things on the human body, which is very relevant for me regarding my background as professional sportsman and my ambitions for my final bachelor project.

At first, I had expected to get some kind of list of possible sensors, but soon it became clear that such a list does not exist or would be way to complex, due to the thousand different kinds of sensors. Instead, we learned how to choose the right sensor. In my opinion this could have been a little more elaborated. We, as a group, started the project full of ambition. I think we all saw some great learning opportunities which made us enthusiastic and we quickly started making plans. However, during the time we had to wait for our components to be ordered we lost track of each other and afterwards it became harder to meet and to divide tasks. We should have made some clear agreements about the team effort, which is a learning point for the future. Also I should have taken the initiative of making these agreements. This lack of good communication caused a little delay, but soon after everybody realized we had to work a little harder and we got back on track.

The technology side of this study is the thing I like the most and I am also pretty skilled within this area. Within our group I believe I had (almost) the most technology experience of us, which enabled me to do a little more than the rest in the same amount of time. I tend to do a lot of work, because I think it will go fastest when I will do it, which eventually will cost me a lot of time and limit the things that can be learned for some of my teammates, who most of the time were really eager to learn. In future projects I shall be more aware of this.

For my final bachelor project I hope to do a project with the Dutch national team of sports climbing. I already got to an agreement with the Dutch climbing federation, but with the latest changes on the department it became hard to propose your own projects. However this project would include a lot of measuring on the sportsmen, and this course made me a lot more comfortable with this. On the one side I know a lot more about the possibilities with sensors and on the other side I know more about the ethics and the restrictions that come to life when measuring on the human body.

Concluding, I believe making sense of sensors was a perfect learning experience for me. It focused on the

exact aspects of technology I am not comfortable with, while it also was connected to my interest in design for sports and the final bachelor project I hope to do.

REFLECTION SIMON A CAMPO

During my study, the choice of the right sensor has always been a struggle. A lot of time was lost in multiple projects since we tried to work with sensors that wouldn't work in the first place. Having this experience I thought it was time to get some more broader knowledge in this area. Another reason was that in my last project, I got feedback that the technology used in the project was not on the expected level. Also with my new project, a complex sensor system was probably needed since it involve AI of the product. Learning to "make sense" of all these sensors seemed necessary to implement in my current Final Bachelor project.
What I would like to get out of the elective is to better understand the sensor's data and be able to create an appropriate system of sensors. Focus is here in how I can combine the sensors within a system, since this can become quite complex within my project. Also I would like to understand better how to calibrate sensors, so useful data can be extracted in order for product to interact in the expected way. After the assignment I would like to be better in choosing the right sensor so no time loss is made here within projects.
The elective consisted of the lectures, which where all attended and a group assignment. The lectures explained the basic about how sensors work, how to select the right sensors, how to create a sensor system based on the action or reaction that you would like to measure and how to use these sensors in the correct way. The lectures gave the insights in the common pitch falls and showed to correct way of working with sensors. The main elements that where valuable for me were how to determine the interval in the time domain and what was ethically allowed and prohibited when working with users, sensors and data. The statistics theory of the lectures where the most interesting for me, since it was a completely new subject. It helped to on a more academic level analyse data and underpin conclusion based on the data. Also analysing the data could help show flaws of the sensor and see if you are measuring what want to measure.
Within the team I focused on the heart rate sensor the most. Since it was an of the shelf sensor that was made for Arduino, not much problems where expected. The opposite was true unfortunately and it required some in depth search in the provided code by the manufacturer. Analysing the raw data of the sensor was harder but didn't throw away data points and thus create a more realistic image. The sensor worked fine in a controlled situation where the user would move, but since our experiment required intense physical movement, most of the data was noise. Luckily this was not a problem for elective since it was now interesting to determine why the sensor didn't work in an appropriate way. Using Matlab to analyse data was a first for me, but it is considered a basic skill in the academic field. I understand the basic way of how

Matlab works I enjoyed experimenting with codes to see if it would give relevant graphs that could lead to better conclusions. Further I helped with the analysis of the thermistor that contrary to the heart rate sensor where almost perfect. This showed both sides of the story and by this use almost all the provided theory in practice. For the sensor system, I designed and 3D printed the case in order to create a compact system. Of course I also helped in the process of picking the right sensor and the user testing.

After the assignment I have a better understanding of sensors in general but more importantly for me, how to analyse data in an academic way. I consider this elective as one of the more valuable ones, since statistics is a domain that has never been taught within the study. Being able to use Matlab will come in handy in the future for sure and I'm curious about the possibilities it has to offer. Now I have more confidence in how I can "make sense of sensors" and create more stable sensor systems. A challenge in the future would be to write code that real-time filters the data so actuators will react but I'm sure that the new gained skills and knowledge of this elective offers the base to do this.