

Knitted smart textile sensors Appendix

Daan van der Valk

Master thesis

Integrated Product Design

Delft, June 2020

Appendix A

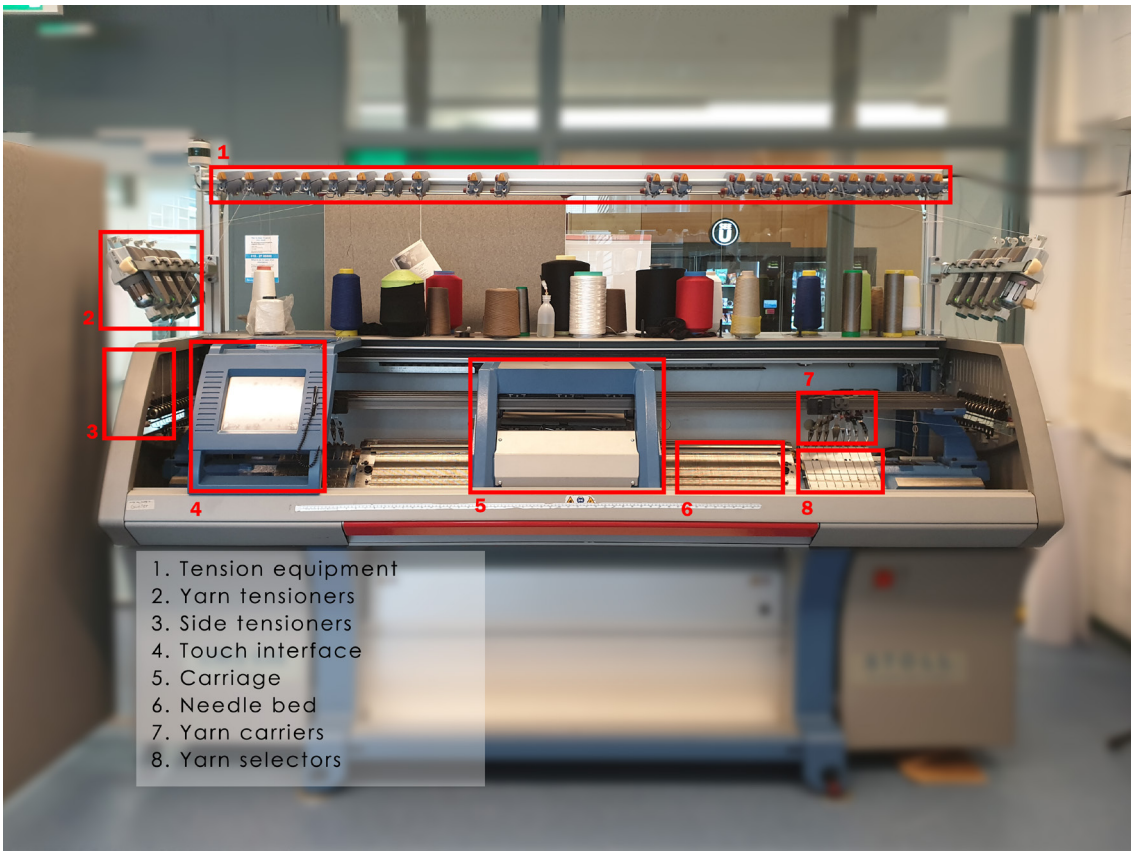
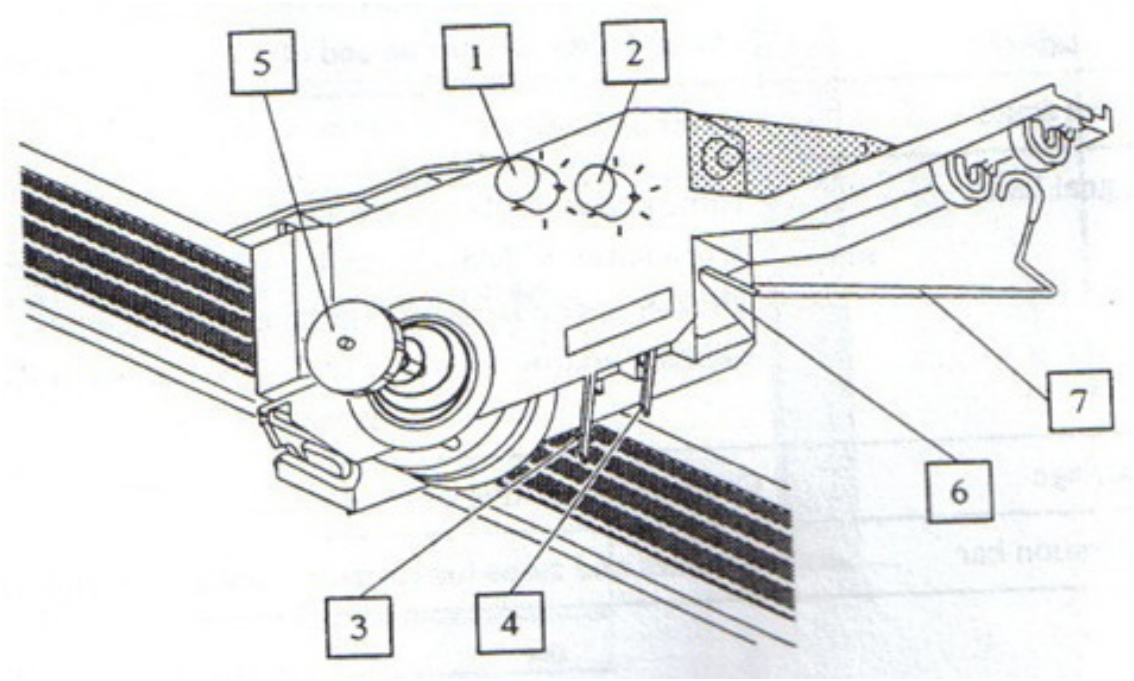


Figure 6. Stoll CMS530 multi gauge knitting machine and its parts.

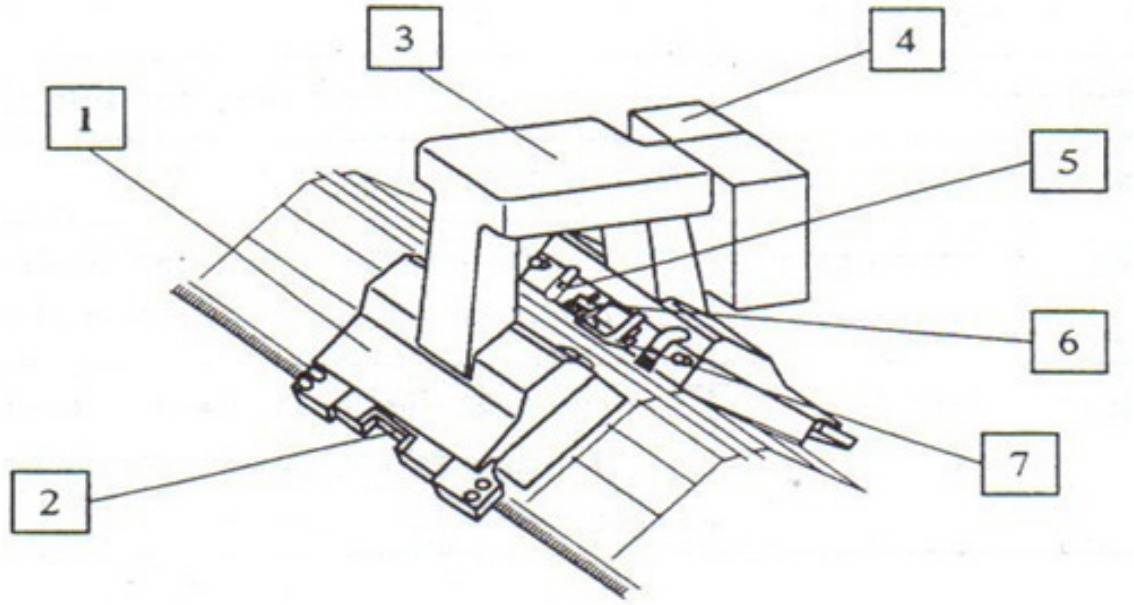
Tension equipment



1. Knot catcher adjust dial (1) -Dial for adjusting the knot catcher to the larger knots.
2. Knot catcher adjust dial (2) - Dial for adjusting the knot catcher to the smaller knots.
3. Knot catcher (a) - Detect for knot and stop the carriage.
4. Knot catcher (b) - Detect for knot. Carriage moves in low speed from the course to the line set in the screen. When a small knot is detected.
5. Tension disk – Adjusts tension of the yarn.
6. Yarn guide plate – Guides the yarn to the knot catcher rightly.
7. Tension – Detect breakage of the yarn.

Kiron, M. I. (z.d.). Main Parts of Flat Bed Knitting Machine and Their Function . Geraadpleegd op 3 april 2020, van https://textilelearner.blogspot.com/2013/02/main-parts-of-flat-bed-knitting-machine.html

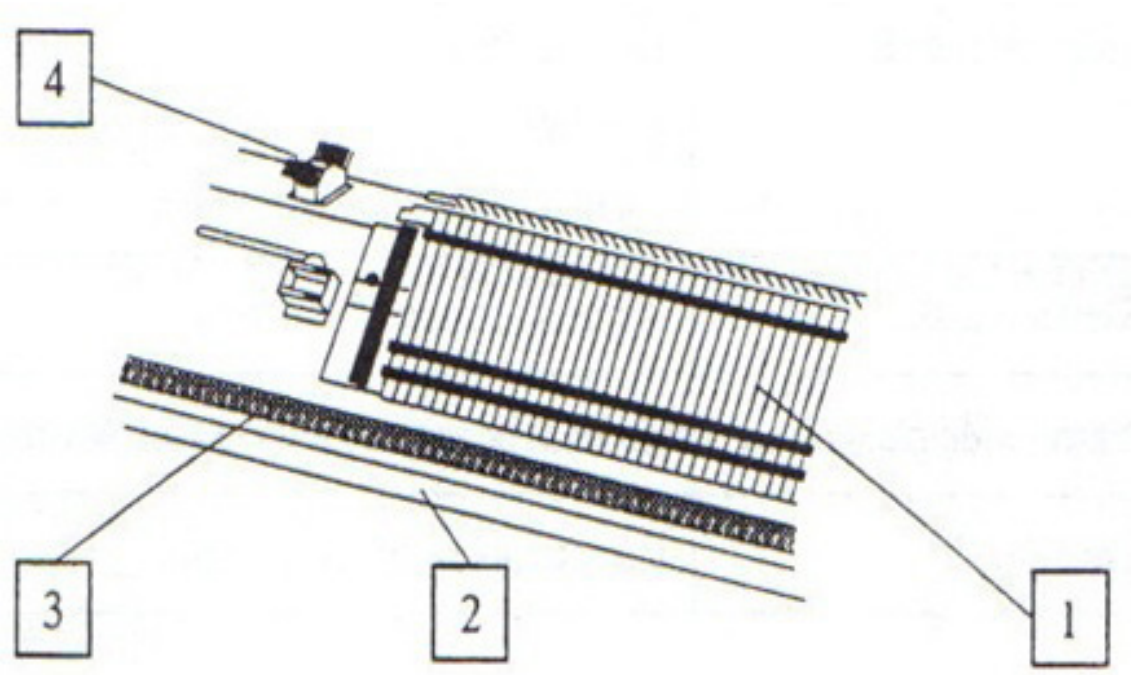
Carriage



1. Carriage – Control the needles.
2. Needle selection sensor – The needle selection gauge and needle selection sensor installed on the carriage can detect the advancing direction of carriage, as well as count the needles.
3. Carrier control box – Carrier catch pins come out from the carrier control box and carriage bring the carrier.
4. Dust cleaner – Suck the dust at the gap. Set the interval of suction and pause in controller screen to repeat exhaust, pause, and suction in sequence during knitting.
5. Fabrics swell direction – Detect the swelling of at the needle beds gap.
6. Brush – Open the needle latch. Also prevents the latch from and closing.
7. Stitch presser – Presses Fabrics into the needle beds gap.

Kiron, M. I. (z.d.). Main Parts of Flat Bed Knitting Machine and Their Function . Geraadpleegd op 3 april 2020, van https://textilelearner.blogspot.com/2013/02/main-parts-of-flat-bed-knitting-machine.html

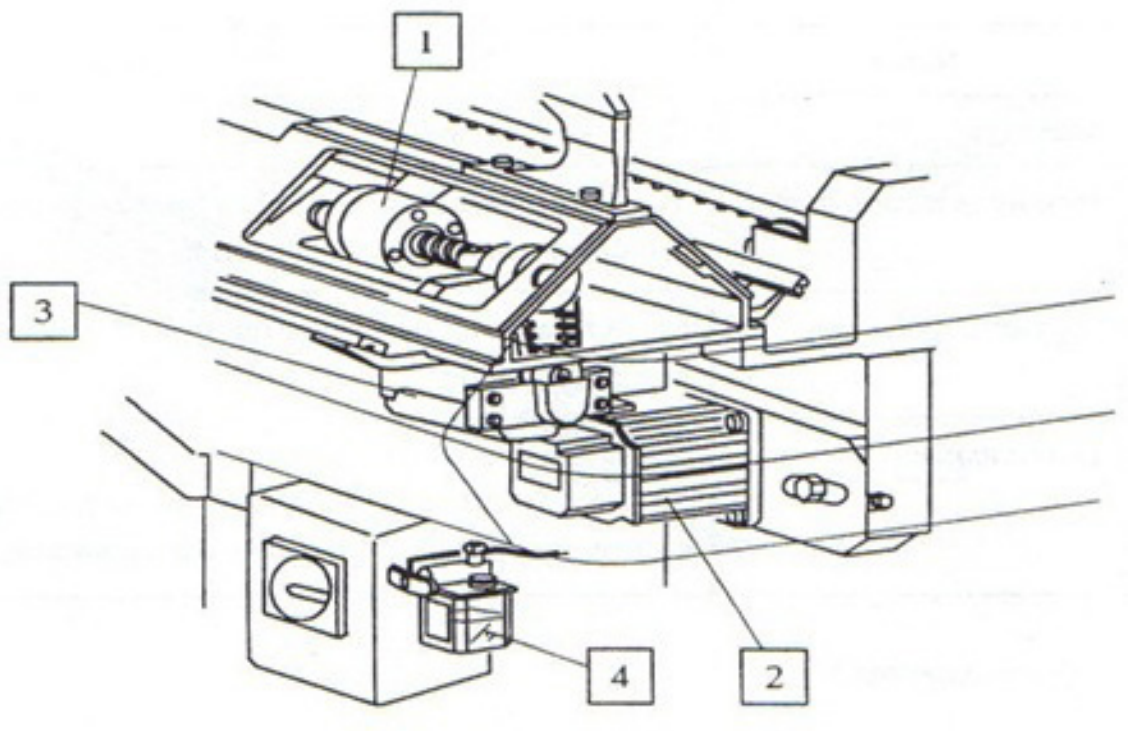
Needle bed



1. Needle bed – Needles, jack, select jack and selector are arranged on this needle bed.
2. Carriage rail – Rail for the carriage.
3. Needle selection gauge.
4. Clear brush.

Kiron, M. I. (z.d.). Main Parts of Flat Bed Knitting Machine and Their Function . Geraadpleegd op 3 april 2020, van https://textilelearner.blogspot.com/2013/02/main-parts-of-flat-bed-knitting-machine.html

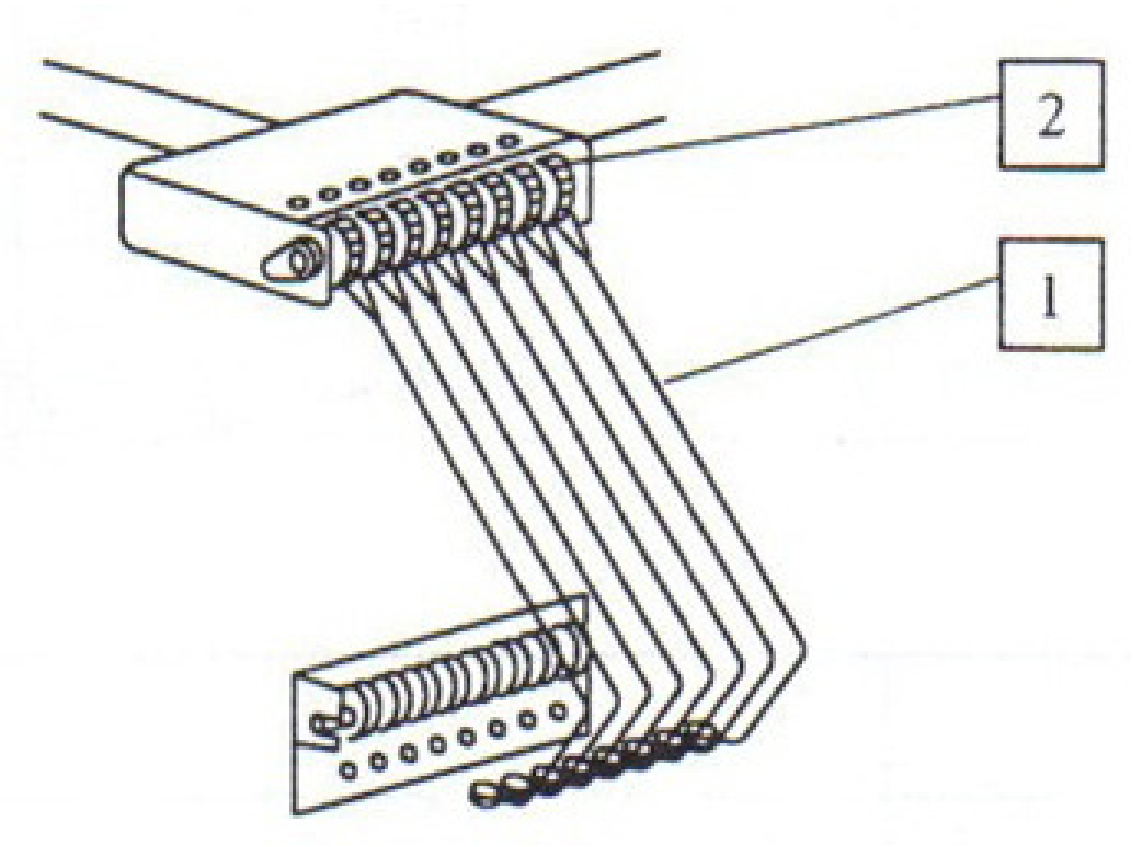
Racking device



1. Ball screw unit – converts the revolution of the racking motor to lateral motion to the needle bed.
2. Main motor – control the carriage.
3. Racking motor – used to rack the needle bed.
4. Oil pump – Supplies oil to brush holder and racking slide plate.

Kiron, M. I. (z.d.). Main Parts of Flat Bed Knitting Machine and Their Function . Geraadpleegd op 3 april 2020, van https://textilelearner.blogspot.com/2013/02/main-parts-of-flat-bed-knitting-machine.html

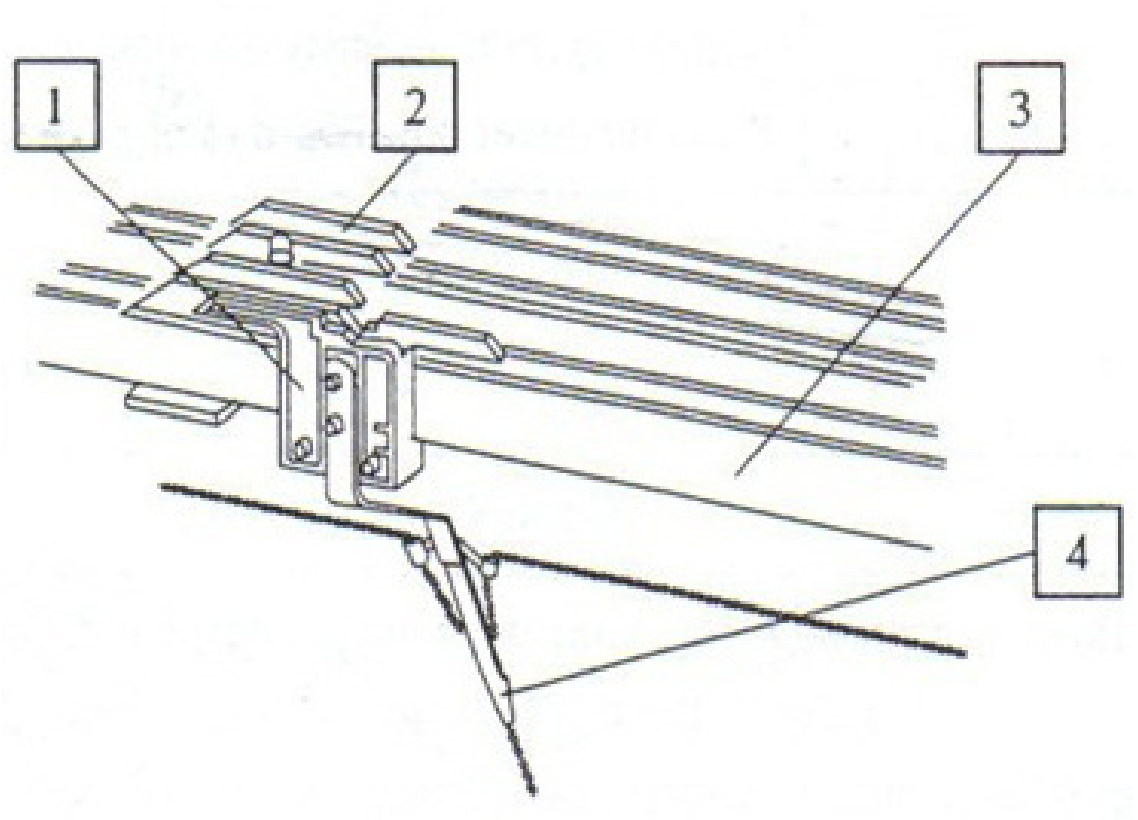
Side tension equipment



1. Side tension – Absorbs the slack of the yarn when the carriage reverses.
2. Side tension dial – Adjusts side tension.

Kiron, M. I. (z.d.). Main Parts of Flat Bed Knitting Machine and Their Function . Geraadpleegd op 3 april 2020, van https://textilelearner.blogspot.com/2013/02/main-parts-of-flat-bed-knitting-machine.html

Carrier control equipment



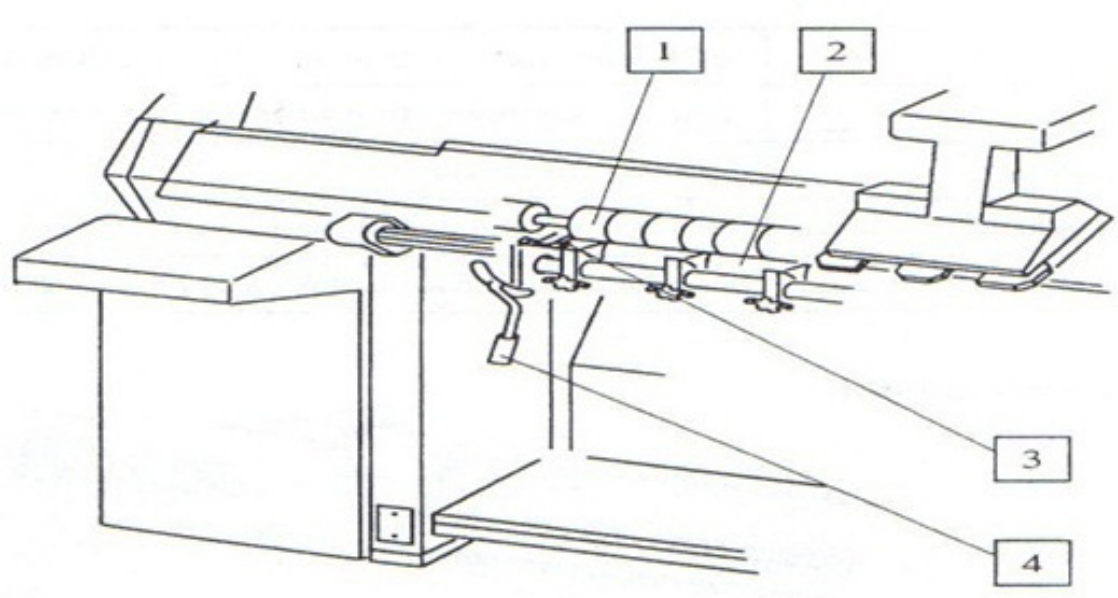
1. Carrier – Carries the yarns.
2. Carrier stopper – Stop the carriers at both ends of the carrier rails.
3. Carrier rail – Rails for the carriers.
4. Yarn feeder – Feeds yarn to the needle beds gap.

Kiron, M. I. (z.d.). Main Parts of Flat Bed Knitting Machine and Their Function . Geraadpleegd op 3 april 2020, van https://textilelearner.blogspot.com/2013/02/main-parts-of-flat-bed-knitting-machine.html

Appendix A

Knitting machine parts and explanation

Fabrics take down controller



1. Main roller – Takes the fabrics down.
2. Sub roller – Press the roll fabric between the main roller and take it down.
3. Yarn guide pin – Used to prevent yarn from twining around the main roller.
4. Lock lever – By actuating this lever, the sub roller is pressed and comes into contact with the main roller. Pull the lever toward you to release the roller from the main roller. Press the lever downward to engage the sub roller.

Kiron, M. I. (z.d.). Main Parts of Flat Bed Knitting Machine and Their Function . Geraadpleegd op 3 april 2020, van https://textilelearner.blogspot.com/2013/02/main-parts-of-flat-bed-knitting-machine.html

1. Open M1plus software

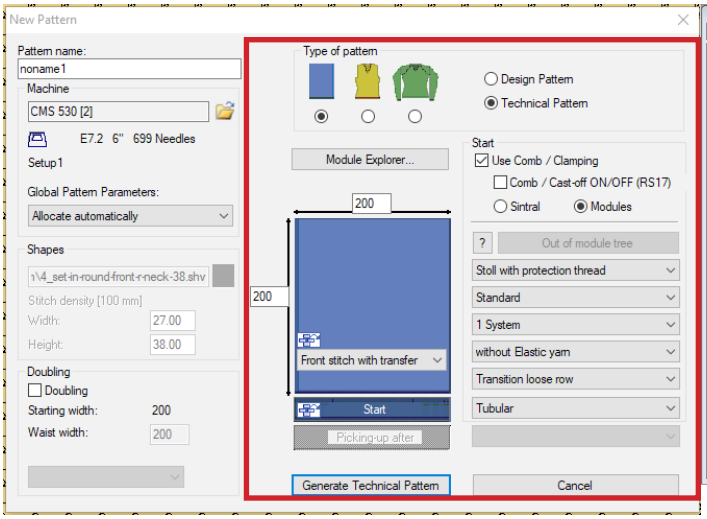
Appendix B

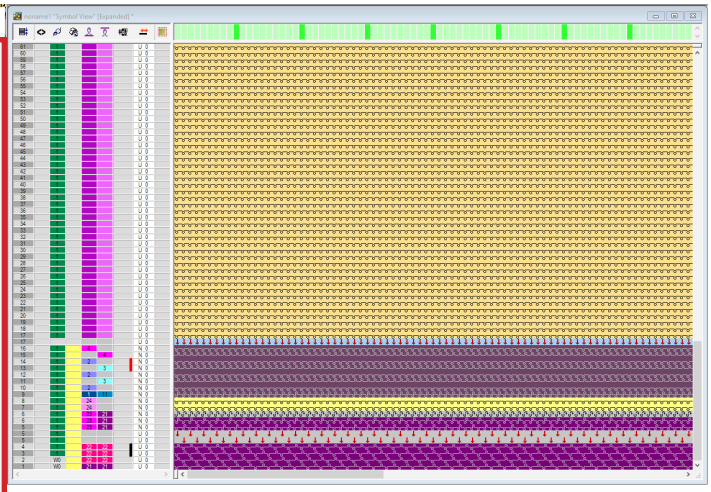
---

2. Choose File -> New Pattern

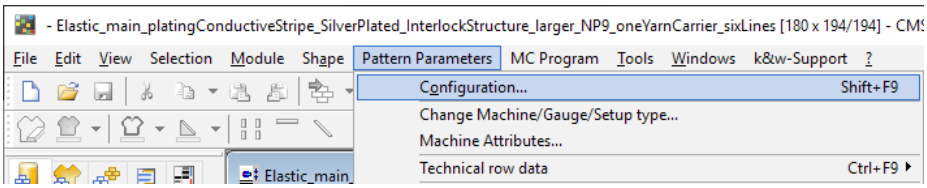
The settings on the right image boxed in red need to be the same.
The blue square has an X and Y size. These can be sized to your needs.

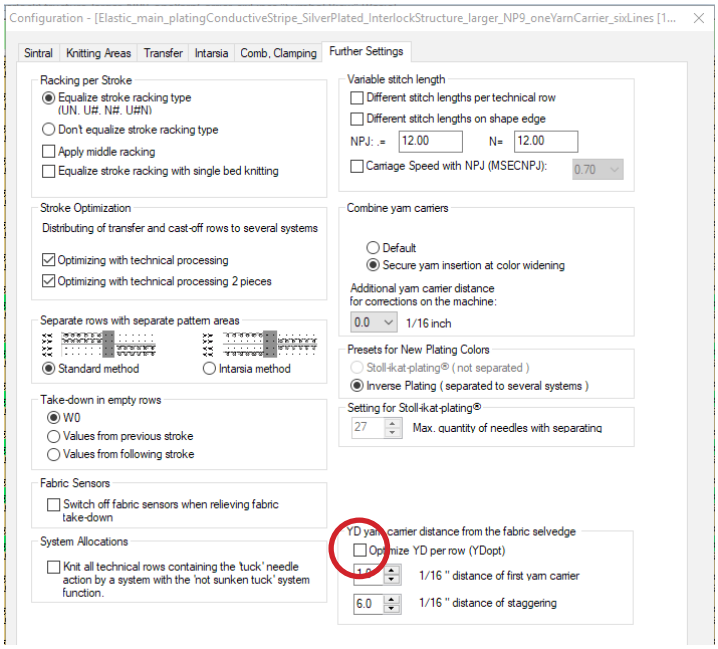
A new file should look similar as the image below.



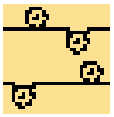


3. Before starting to edit the knitting file ‘Optimize YD per row’ needs to be unchecked.

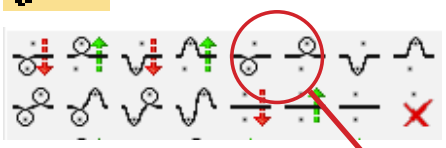




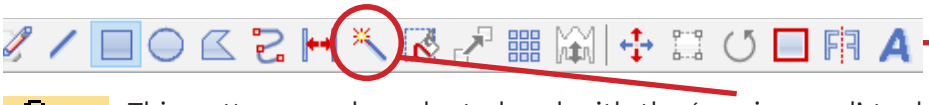
4. Now we can start to make the interlock stitch pattern, which is basically this pattern repeated in X and /Y directions

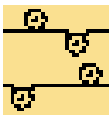


Two encircled stitches are used to create this pattern. Fill the whole fabric with this part that is yellow.



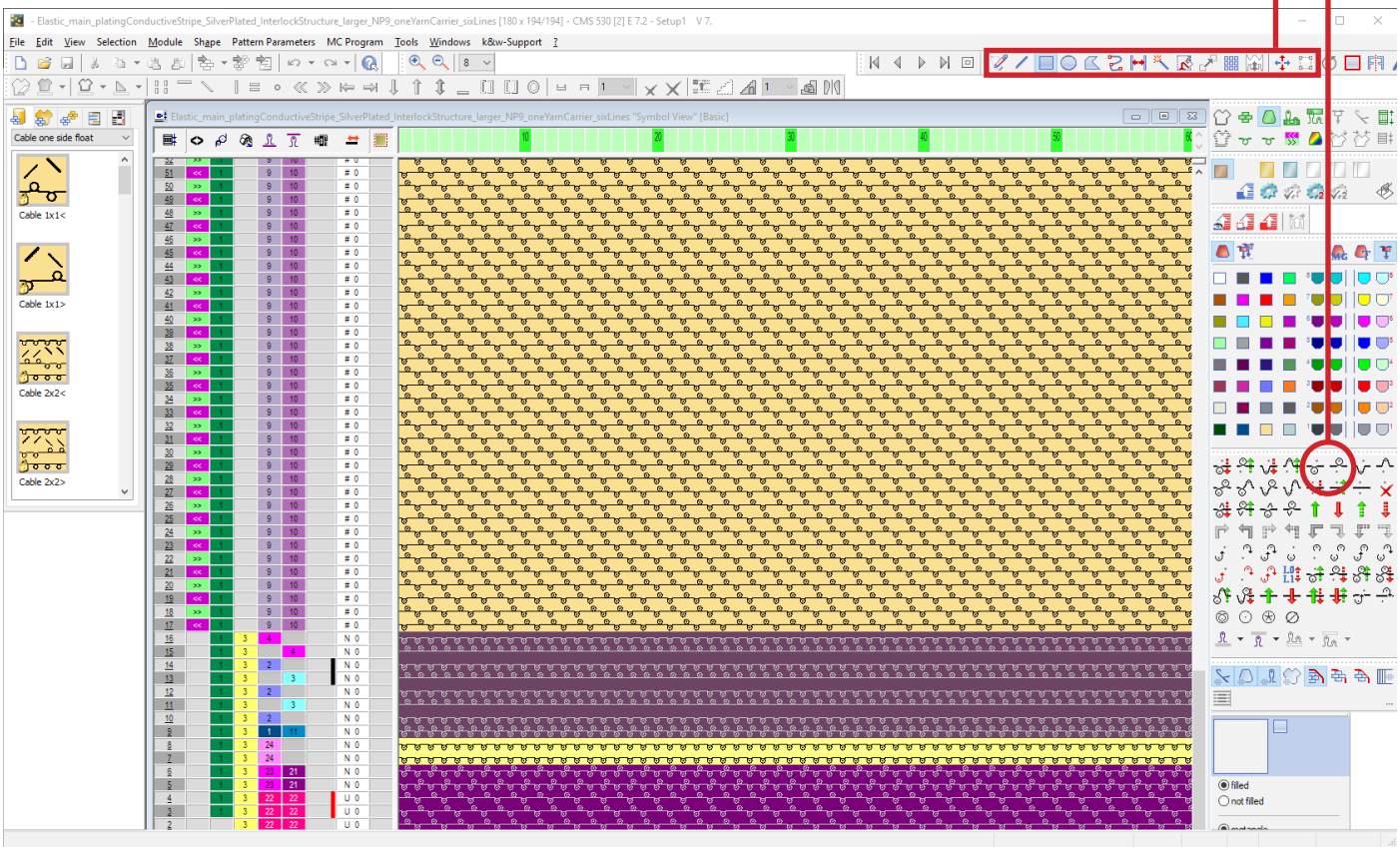
To tools to ‘paint’ this pattern are located in the top of the interface.



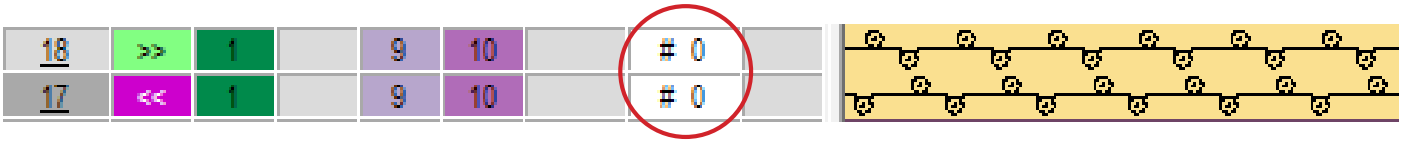


This pattern can be selected and with the ‘magic wand’ tool the whole field can be painted in one go.

The created fabric should look similar to the image below.

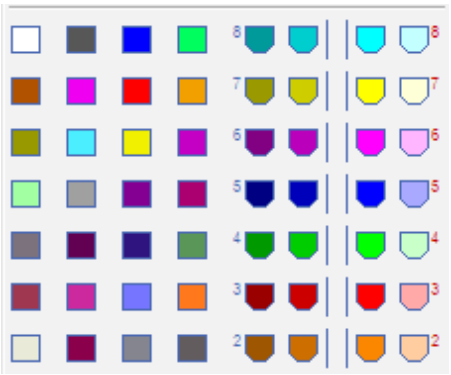


5. The column that has N0 needs to be changed to #0. This can be done by right clicking on N0 and selecting the #0 tool and left clicking the N0’s



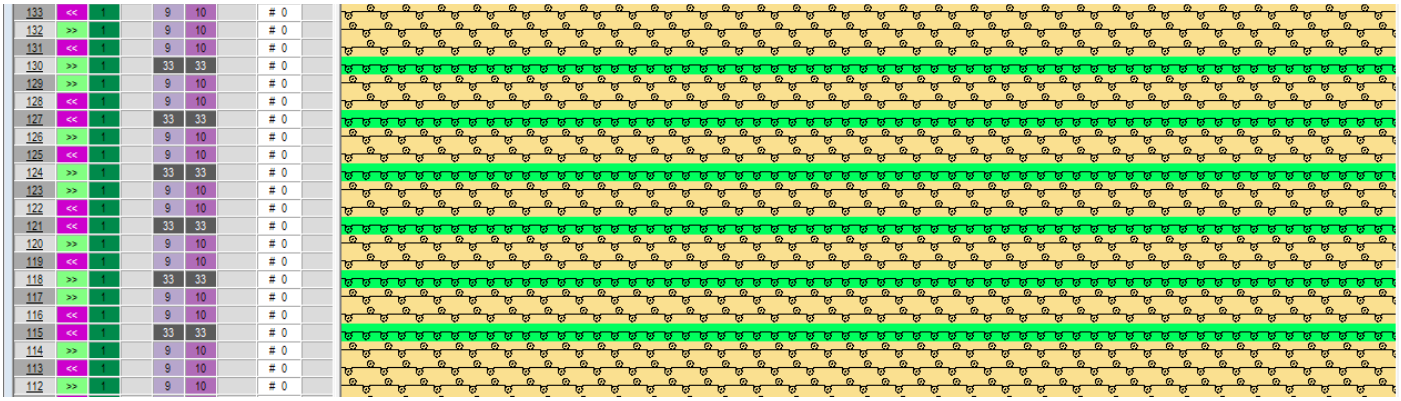
Sensor with lines in course direction

6. Now the conductive lines can be added in the fabric. For the best sensor performance 6 lines of conductive yarn should suffice.

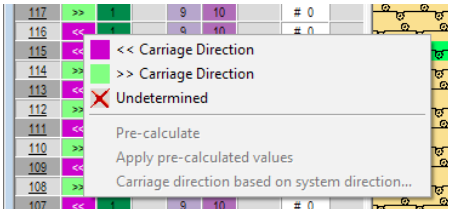


Create the lines with a different color and only the front stitch. The color makes it that the lines can be linked to a different yarn carrier. The color panel is placed above the stitch type panel and looks like the image on the right.

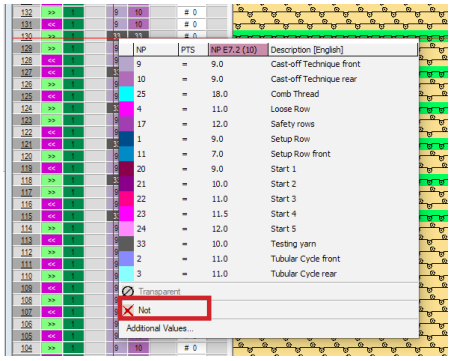
The lines should look similar to the image on the bottom



Notice the direction of the lines. The direction can be changed by right clicking on the direction sign (<< or >>) and selecting the right direction tool as seen on the image on the right.

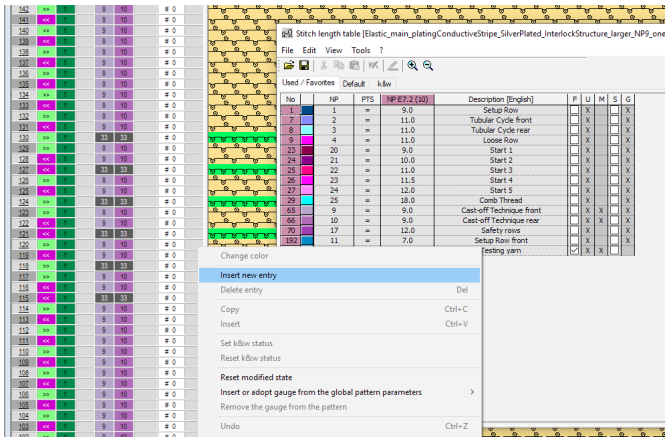


Now The stitch size needs to be adjusted. This is called the NP value in M1plus. To add a new NP value for the conductive lines, right click the NP values in your file and click additional values. Here a new NP value can be added and existing NP values can be adjusted.

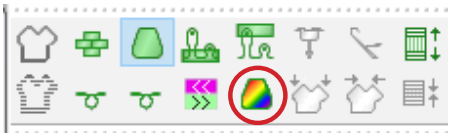


To add a new NP value right click and click add insert new entry.

The NP values can be adjusted in the fabric file the same way the direction can be changed. Right click and select the Np value tool you want (the same menu where additional values is clicked).



Now that the whole fabric is created, the yarn carriers need to be placed. They can be changed in the ‘yarn fields’, encircled on the image on the right. This icon is placed in the top right of the M1plus software.

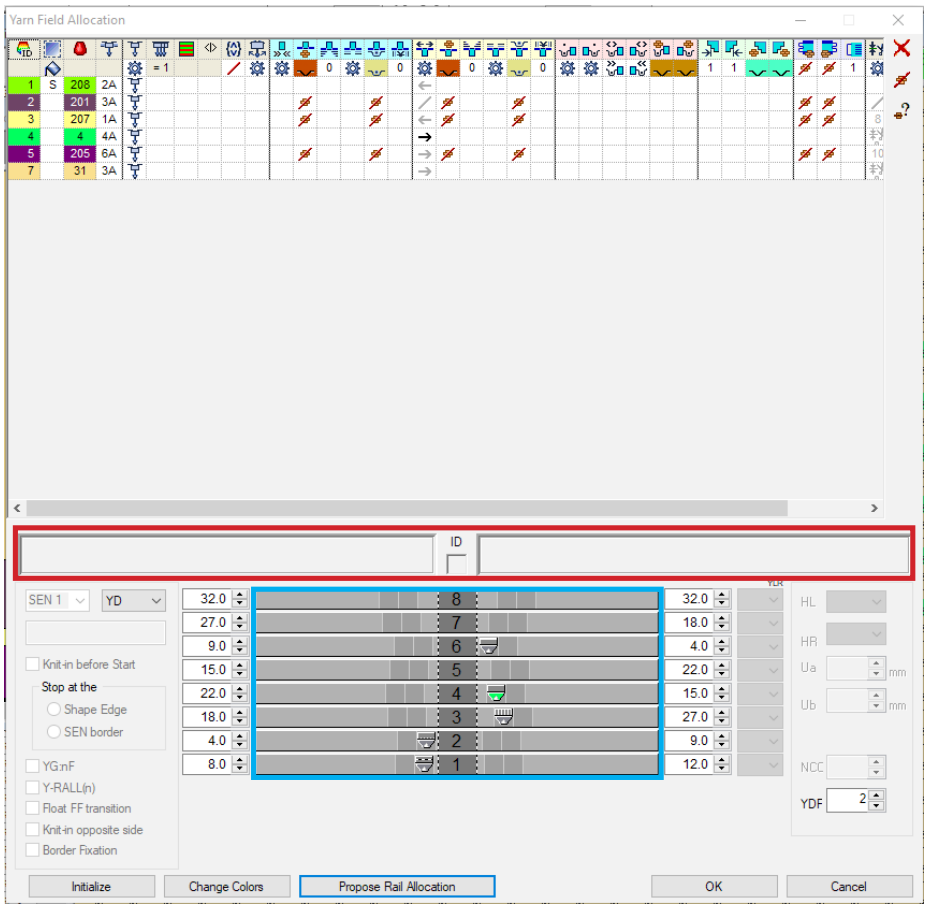


The following wind will open together with a representation window of the knitted fabric.

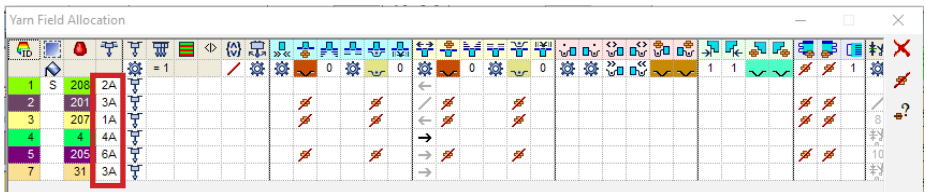
The yarn carriers are located in the red box on the image on the right. These need to be placed below on the correct yarn carrier in the blue box in the image.

These correspond to the rails on the knitting machine. When pressing a yarn carrier, the knitted part will be highlighted in the window that opens up together with the yarn fields window.

Once all yarn carriers are moved from the red to the blue box the window can be closed by pressing ‘OK’.



Some yarn carriers can be combined into one if this is wanted. This can be done by right clicking the number in the top table of the yarn fields window. The right yarn carrier can be selected as a tool and by clicking the yarn field that needs to be changed the number will change.

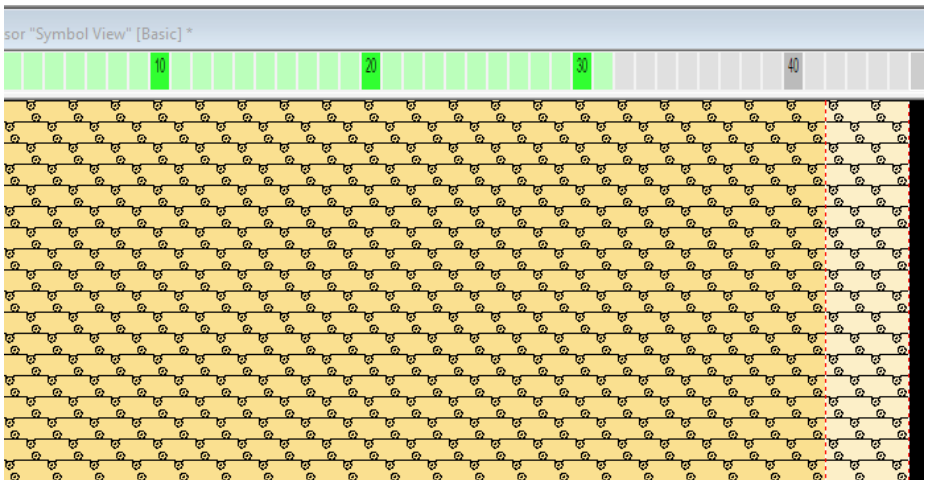


The last step can be found on the last page of the explanation. It is used for every made knitting pattern.

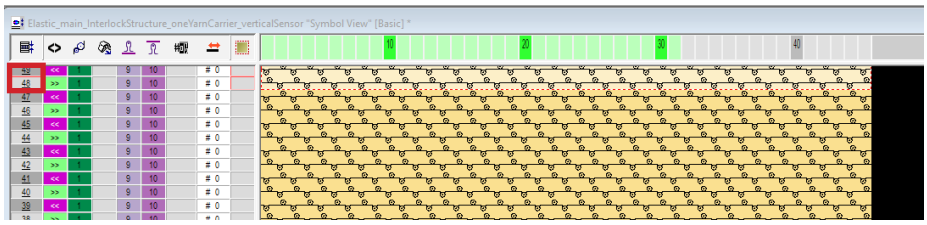
Sensor with lines in wale direction

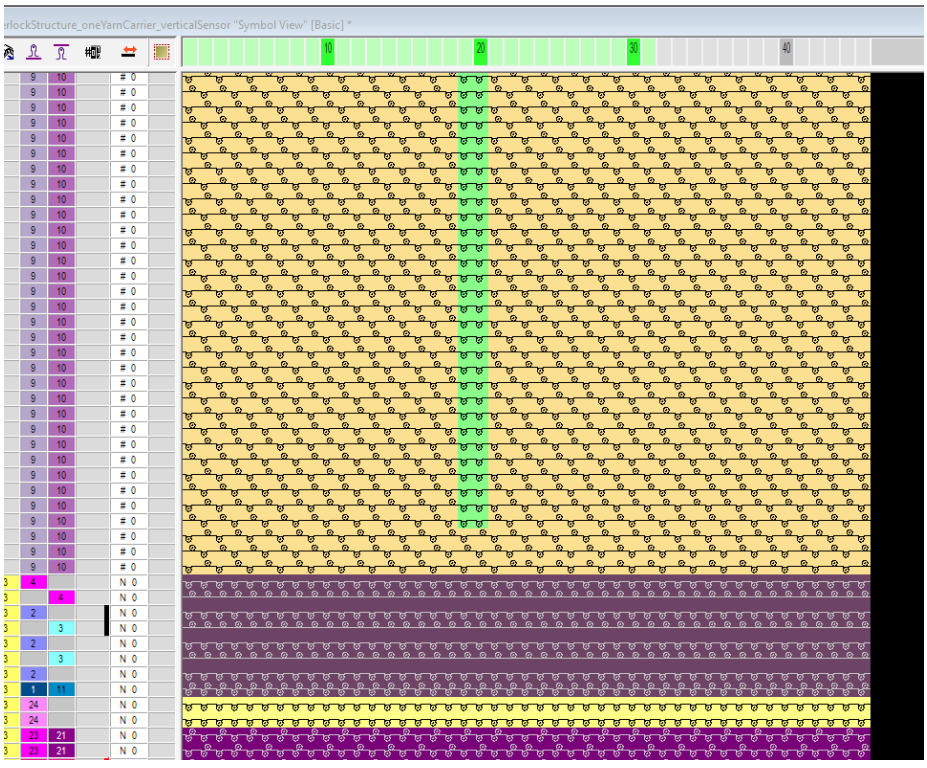
To create the sensor lines in the wale direction, a different knitting technique is used. The technique is called jacquard, which is described in chapter 2.2. In the M1plus software the structure of the knitted fabric needs to be changed from a rectangle that is lying on its long side to one that is standing on its short side, because we are going to create the sensor part in the vertical direction (wale direction).

First cut the sides of the fabric to a width of 50 to 100 needles and a height of 900 rows. This will result in a strip of around 4 by 30 centimeters. These are not a proprietary measurements of course. To get an idea of the size, this sensor size is also used in the arm movement sensor described in chapter 5.2.

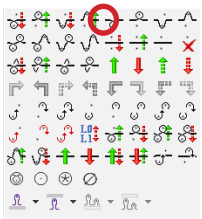


Now the height needs to be increased. This can be done by selecting the two upper rows by dragging on the numbers on the left side and pressing the ‘insert’ button on the keyboard. Increase the height to around 900 rows.

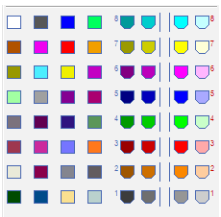




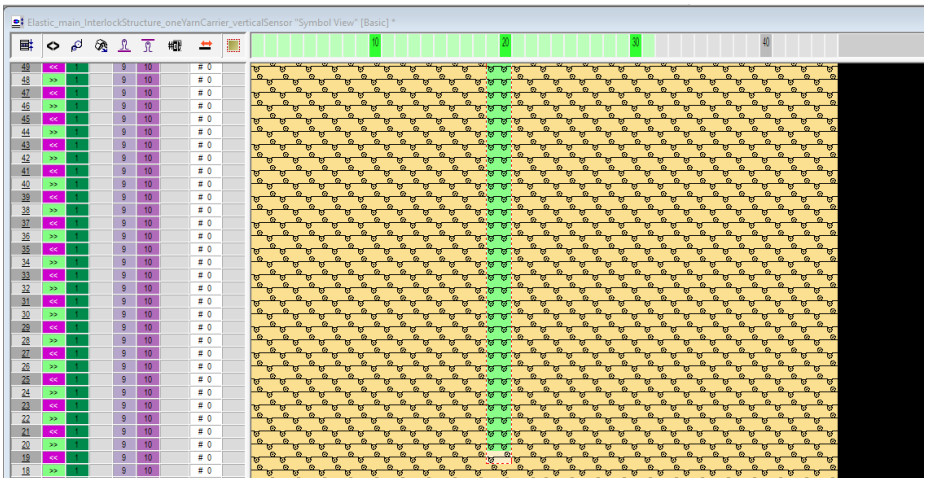
Now we are going to create the vertical line what will be the sensing part of the fabric. Select a different color (the type of color does not matter) and front stitch and draw a two needle wide line all the way up the length of the fabric.



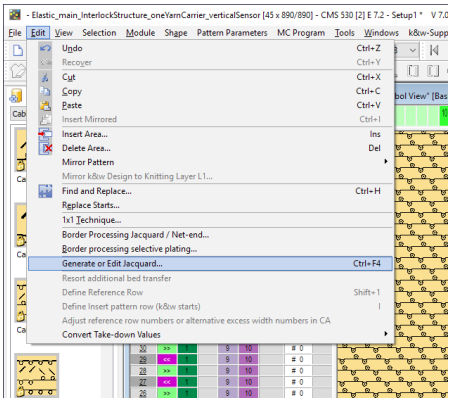
Pick front stitch

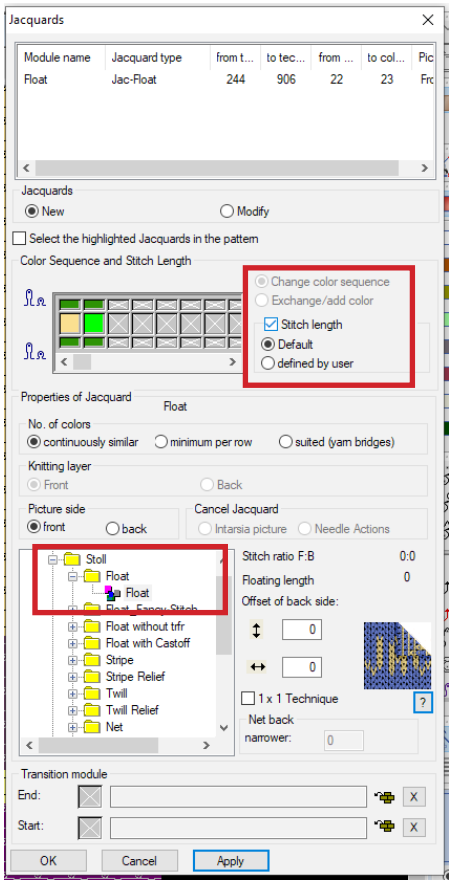


Press the ‘escape’ key a few times to remove the tool that is in use. Now select the green line and include two stitches below the green line. We do this so the software knows it is working with two yarns when doing the following step.



Now on the top left in the M1plus software press ‘Edit’ and then press ‘Generate or Edit Jacquard…’

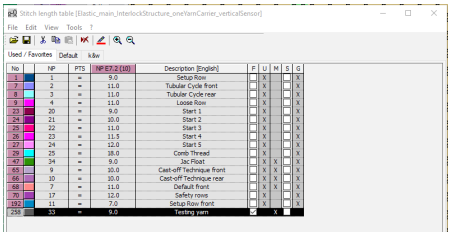




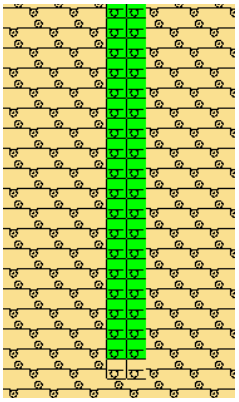
The following window will open. The smaller window on the bottom left is where the Float is selected. Open the ‘Stoll’ folder, than the ‘Float’ folder and double click the Float.

To change the NP value of this line, check the ‘Stitch length’ box in the jacquard window that you just opened. Select ‘defined by user’ and create a new NP value.

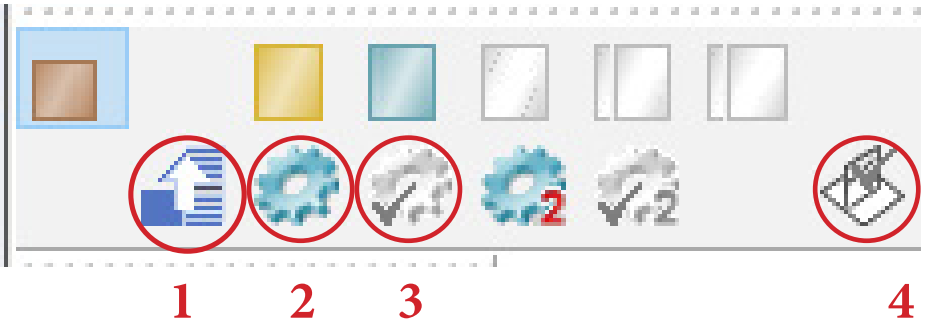
The following window will open where you can right click and press ‘insert new entry’ to create a new NP value that is used for the vertical line.



Press ‘Apply’ and ‘OK’. Now the icon in the sensor line should look like the picture on the right.



The last part is to develop the stitch file. To do this the buttons on the right from 1 to 4 need to be pressed. Between each button press the program will process the knitting file. Some warnings can appear, but these can be ignored by pressing OK.

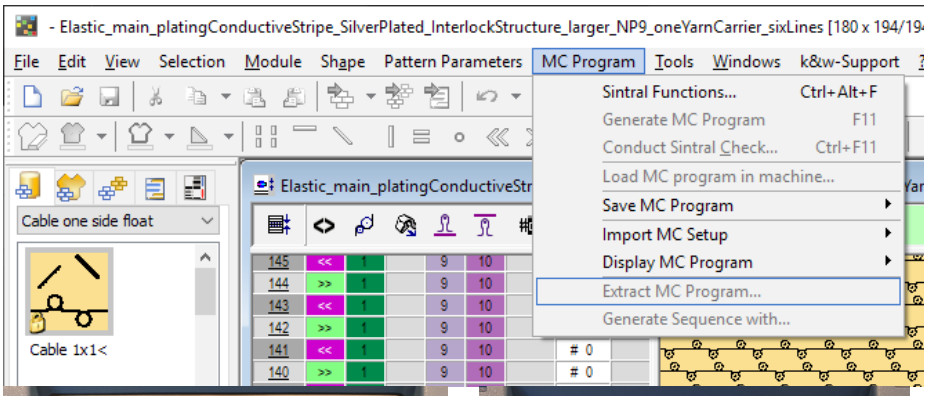


The last step (Step 4) is a simulation of the knitted fabric. This will say if the created fabric is ready to be knitted on the knitting machine.

When process 1-4 is done, the file can be exported by selecting MC program in the top of the M1plus software and choosing Extract MC program. This can only be done when steps 1-4 are finished.

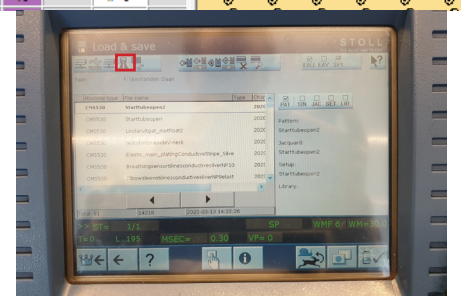
Note: When the processing of 1-4 does not work correctly there most likely is a mistake in the created file and previous steps should be rechecked.

When the files are extracted to a USB, they can be loaded into the machine.

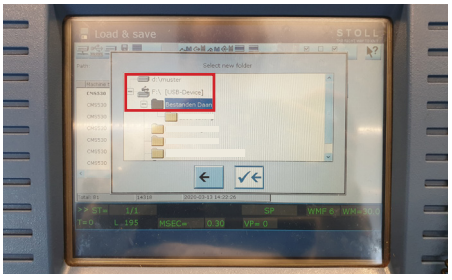




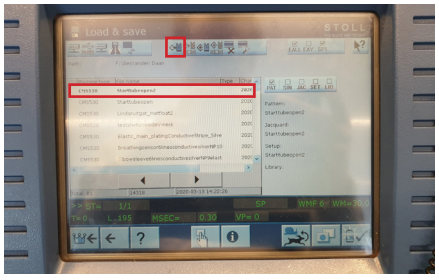
Step 1: Load program of usb



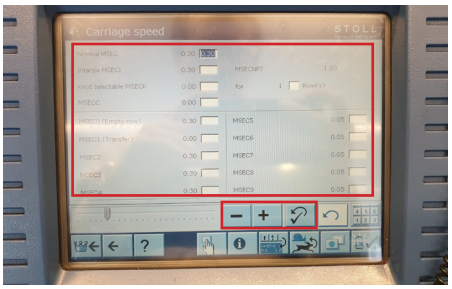
Step 2: Press the path icon



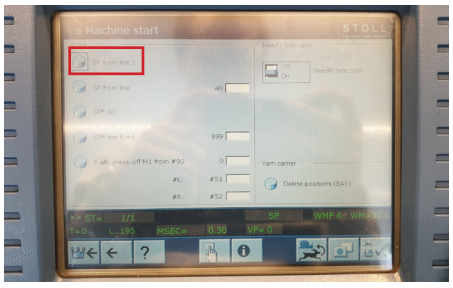
Step 3: Select correct storage path



Step 4: Select the knitting file and press load



Step 5: Change speeds if necessary

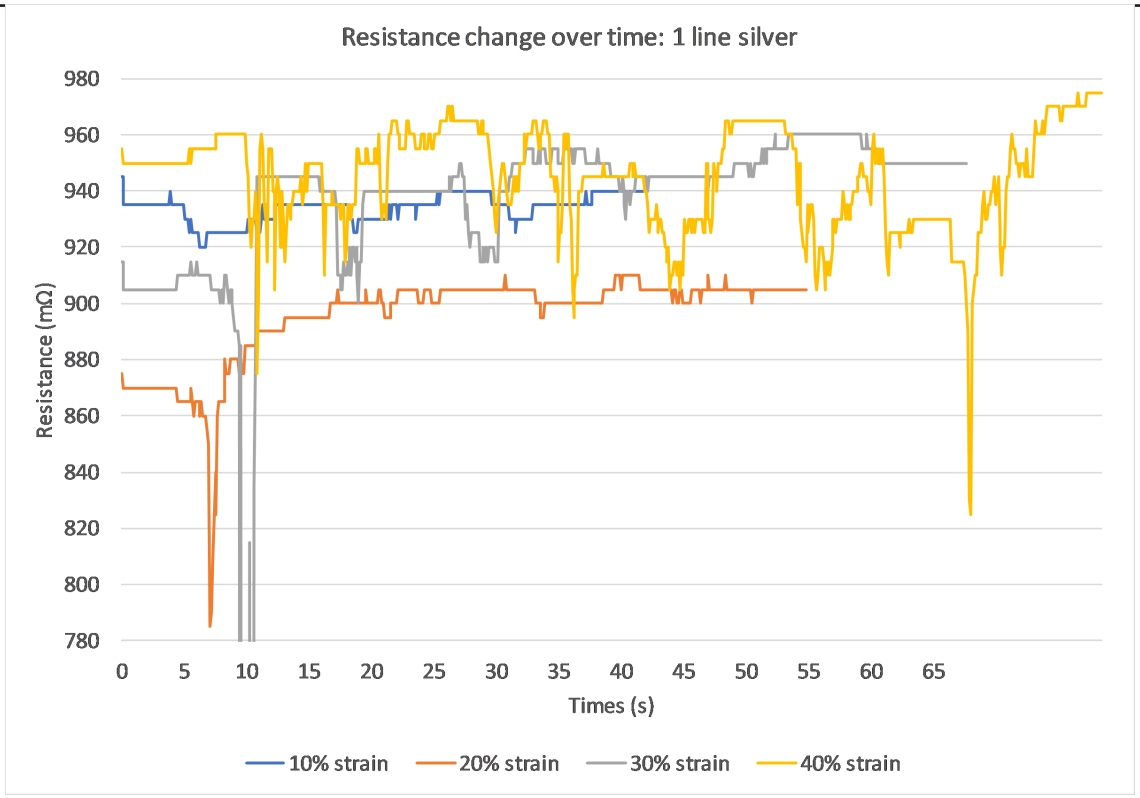


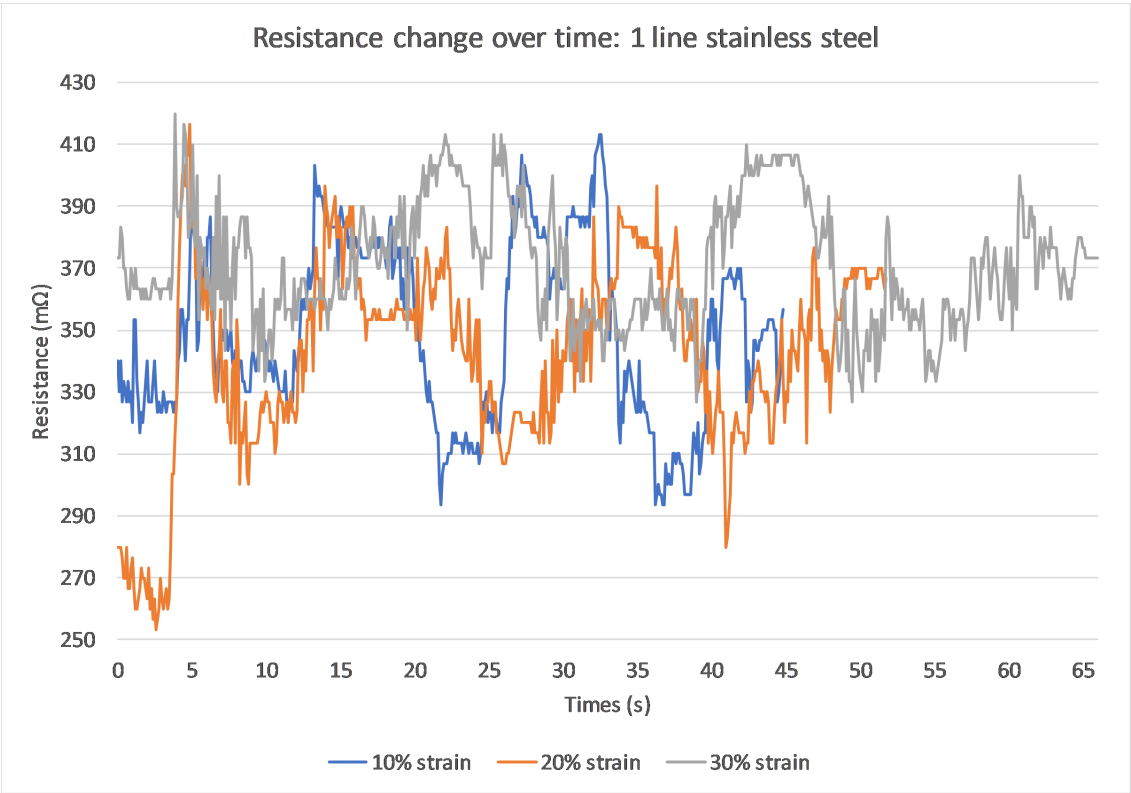
Step 6: Start the program

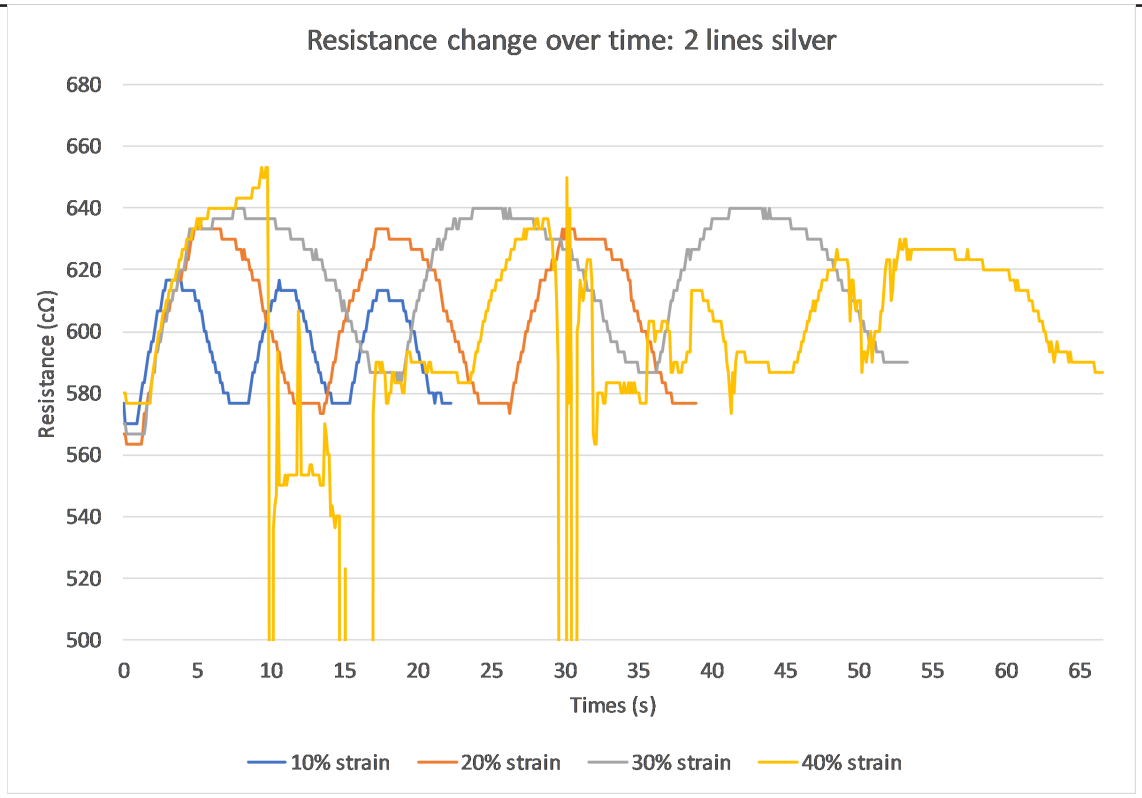
Knitting machine interface

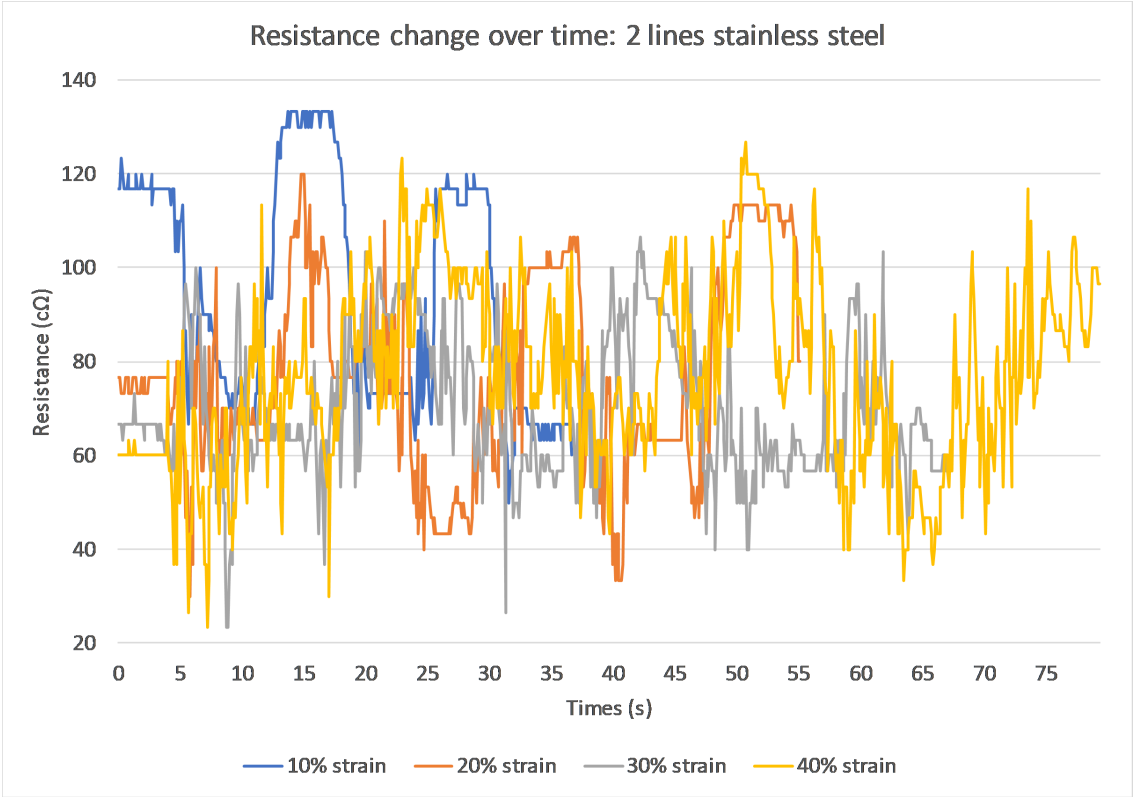
Step 1: Load the file from the usb.
Step 2: Press the road button to select the correct path of the usb.
Step 3: Select the correct path of the usb.
Step 4: Press the file that needs to be knitted and press the load button highlighted red.
Step 5: Change the speeds if necessary (See step 1, speeds).
Step 6: Start the program. Press the green traffic light (See step 1, start in the main menu, than press SP from line 1.

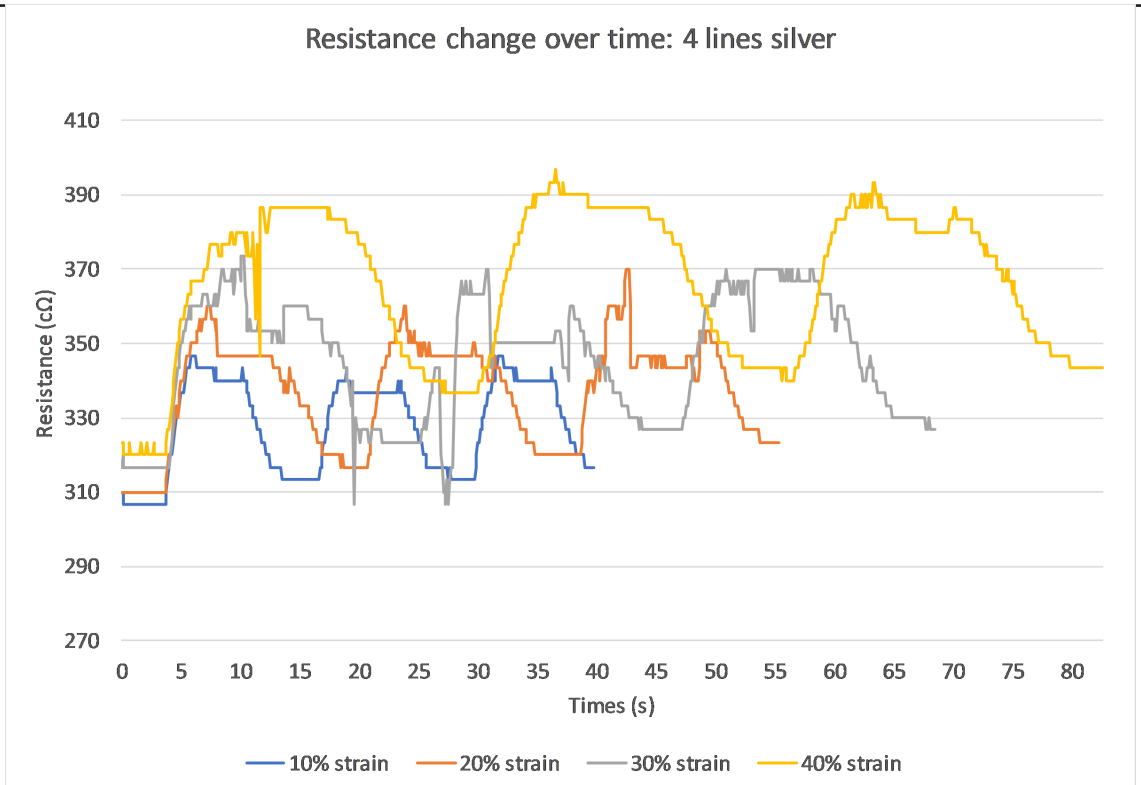
Appendix C

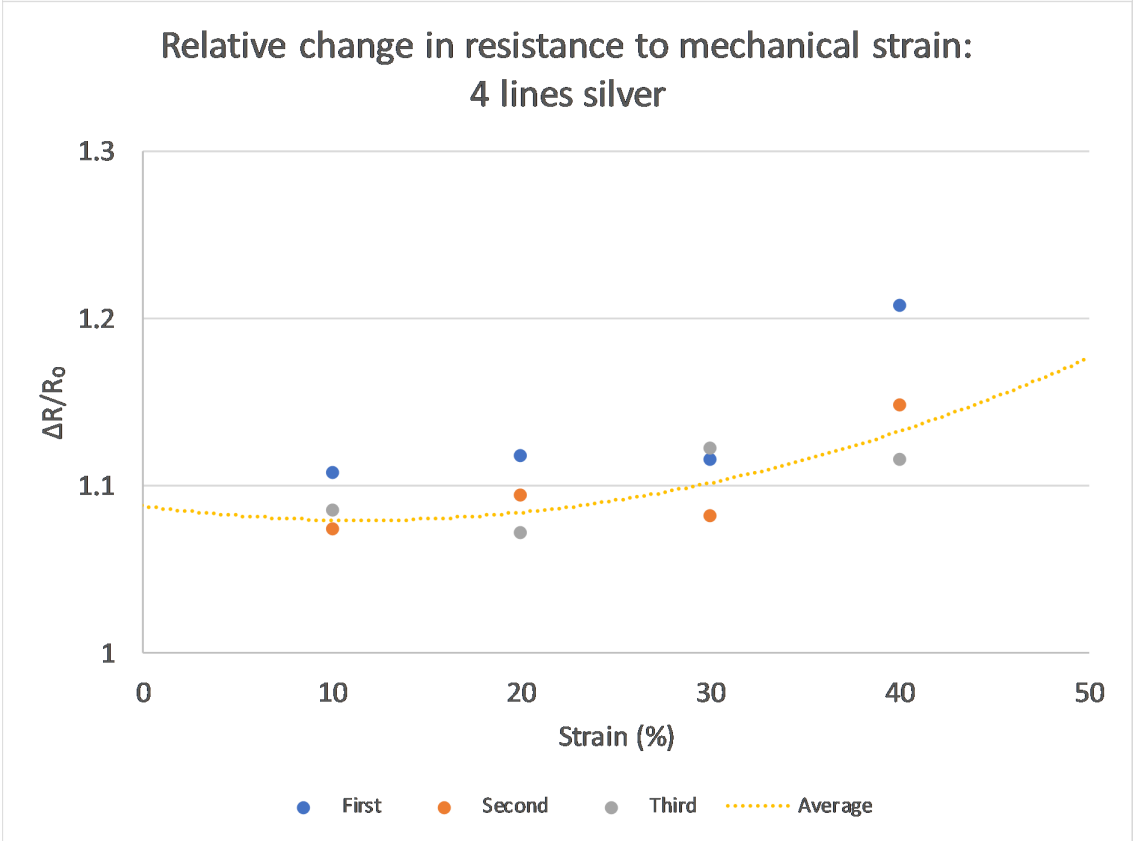

Resistance change over time: 1 line silver
Resistance (mΩ)
980
960
940
920
900
880
860
840
820
800
780
0 5 10 15 20 25 30 35 40 45 50 55 60 65
Times (s)
10% strain
20% strain
30% strain
40% strain

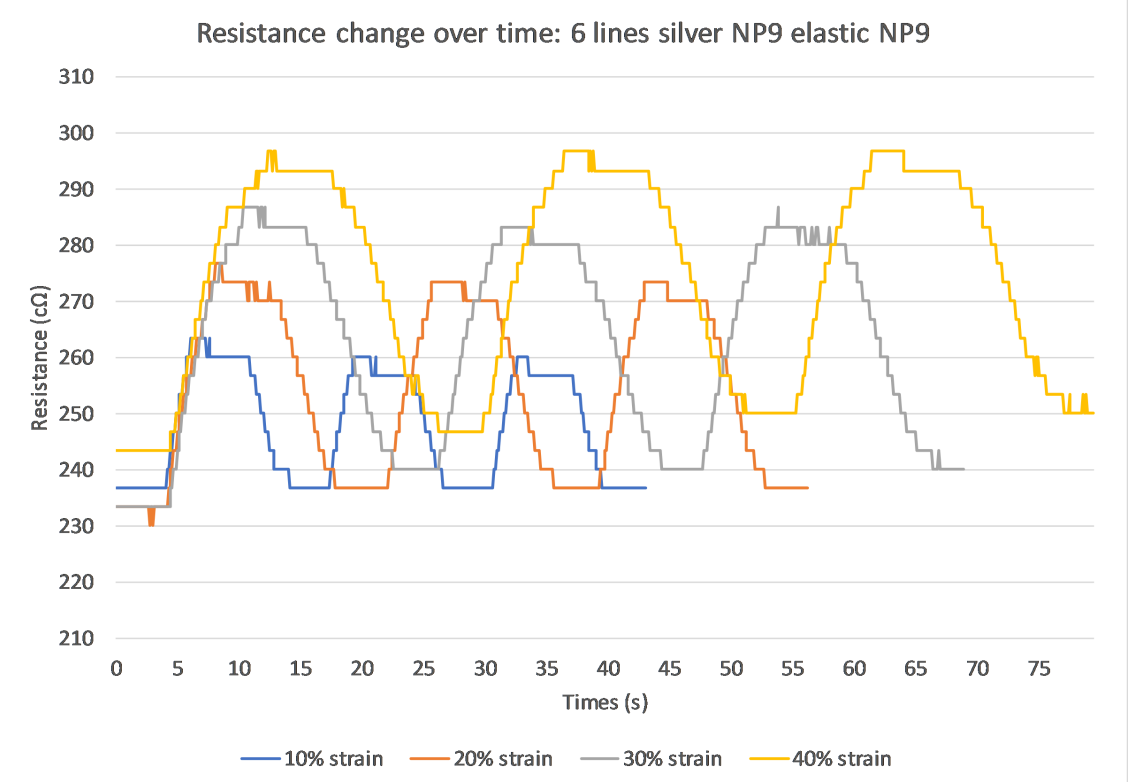

Resistance change over time: 1 line stainless steel
Resistance (mΩ)
430
410
390
370
350
330
310
290
270
250
0 5 10 15 20 25 30 35 40 45 50 55 60 65
Times (s)
10% strain
20% strain
30% strain

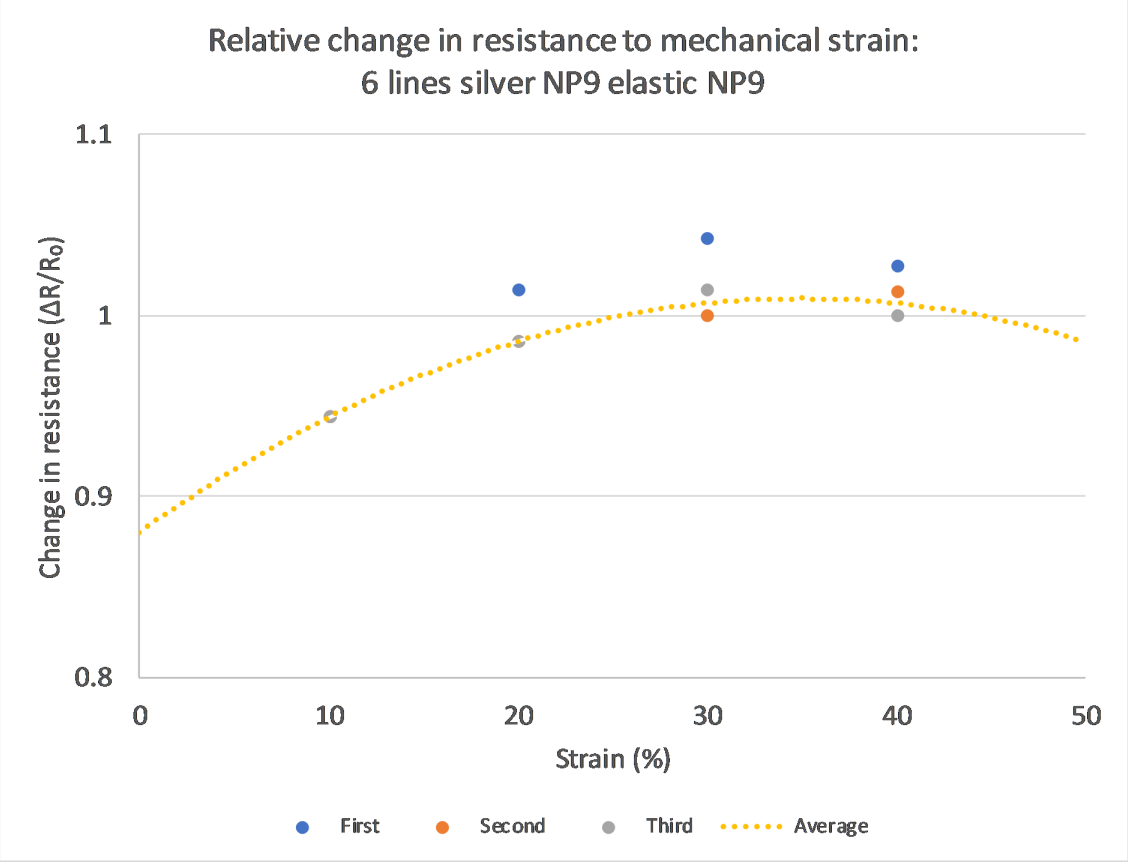

Resistance change over time: 2 lines silver
680
660
640
620
600
580
560
540
520
500
Resistance (cΩ)
0
5
10
15
20
25
30
35
40
45
50
55
60
65
Times (s)
10% strain
20% strain
30% strain
40% strain

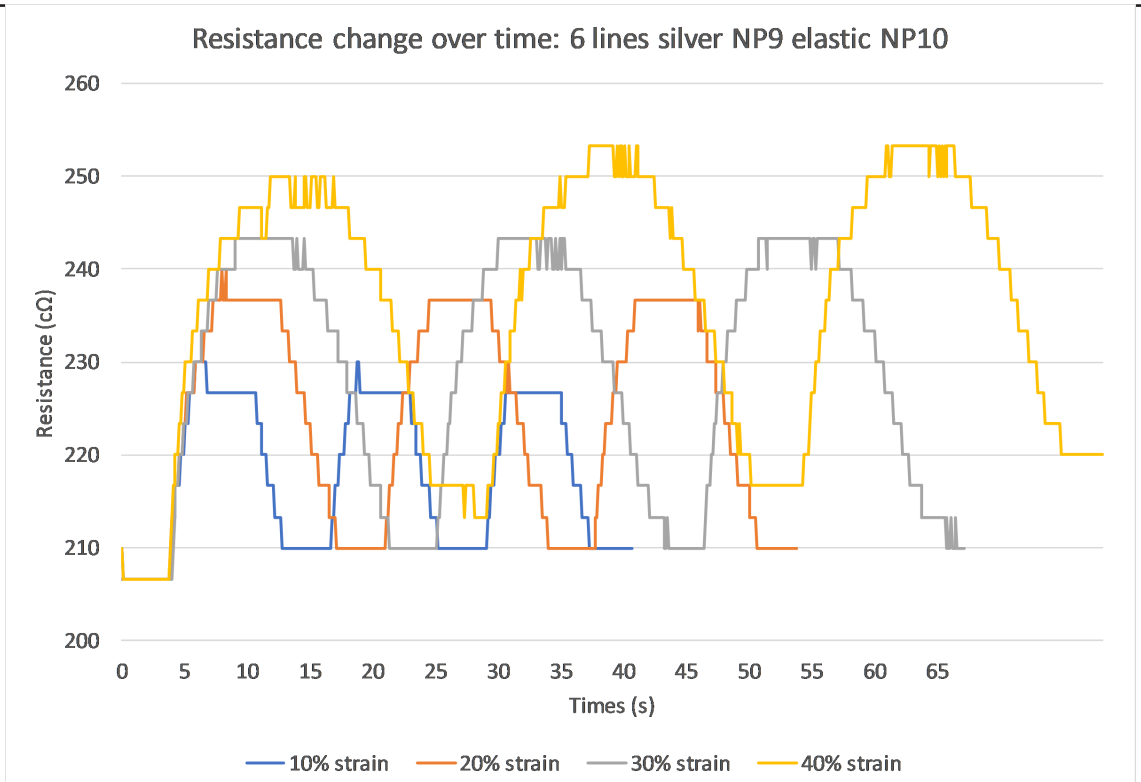

Resistance change over time: 2 lines stainless steel
140
120
100
80
60
40
20
Resistance (cΩ)
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
Times (s)
10% strain
20% strain
30% strain
40% strain

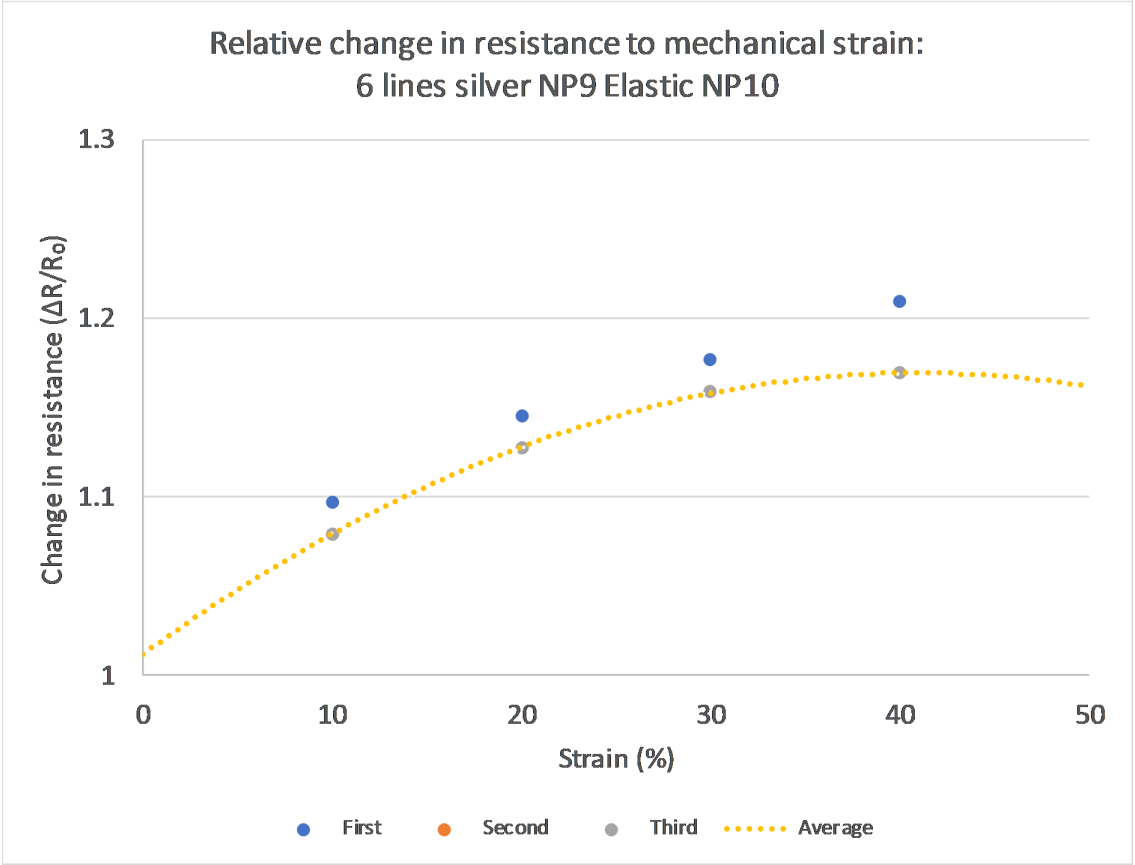

Resistance change over time: 4 lines silver
Resistance (cΩ)
410
390
370
350
330
310
290
270
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80
Times (s)
10% strain
20% strain
30% strain
40% strain

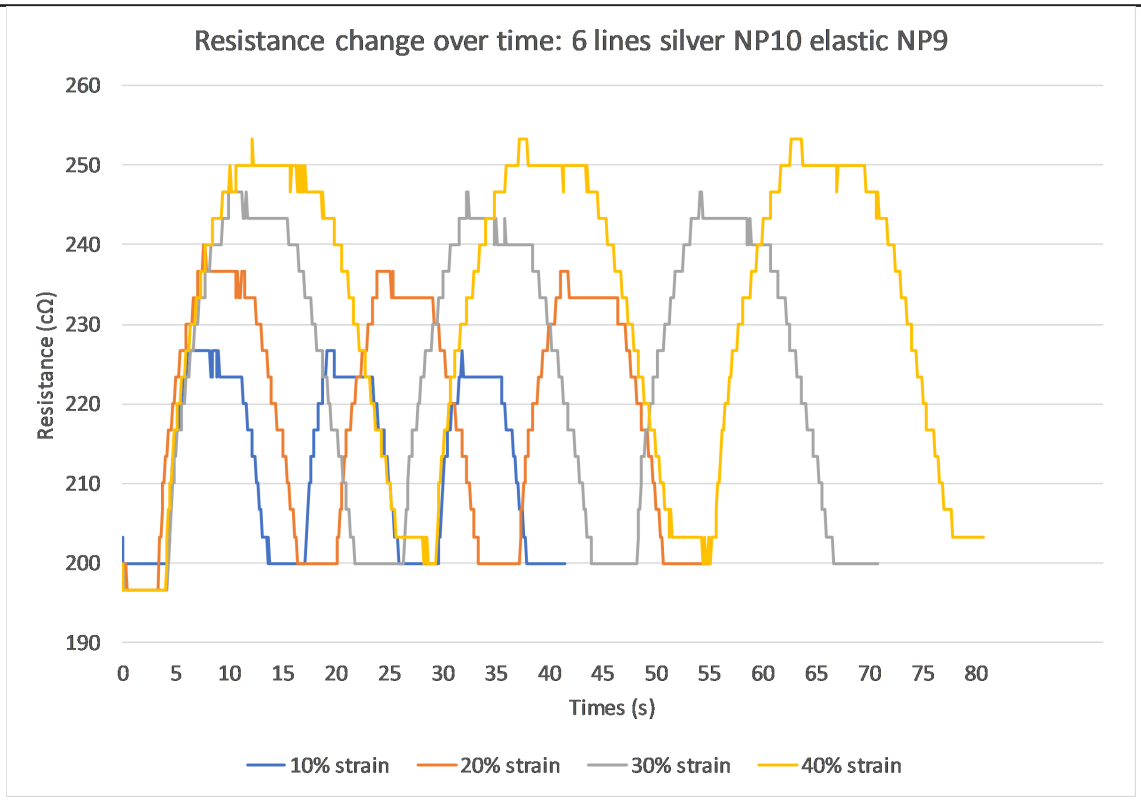

Relative change in resistance to mechanical strain:
4 lines silver
ΔR/R₀
1.3
1.2
1.1
1
0 10 20 30 40 50
Strain (%)
First
Second
Third
Average

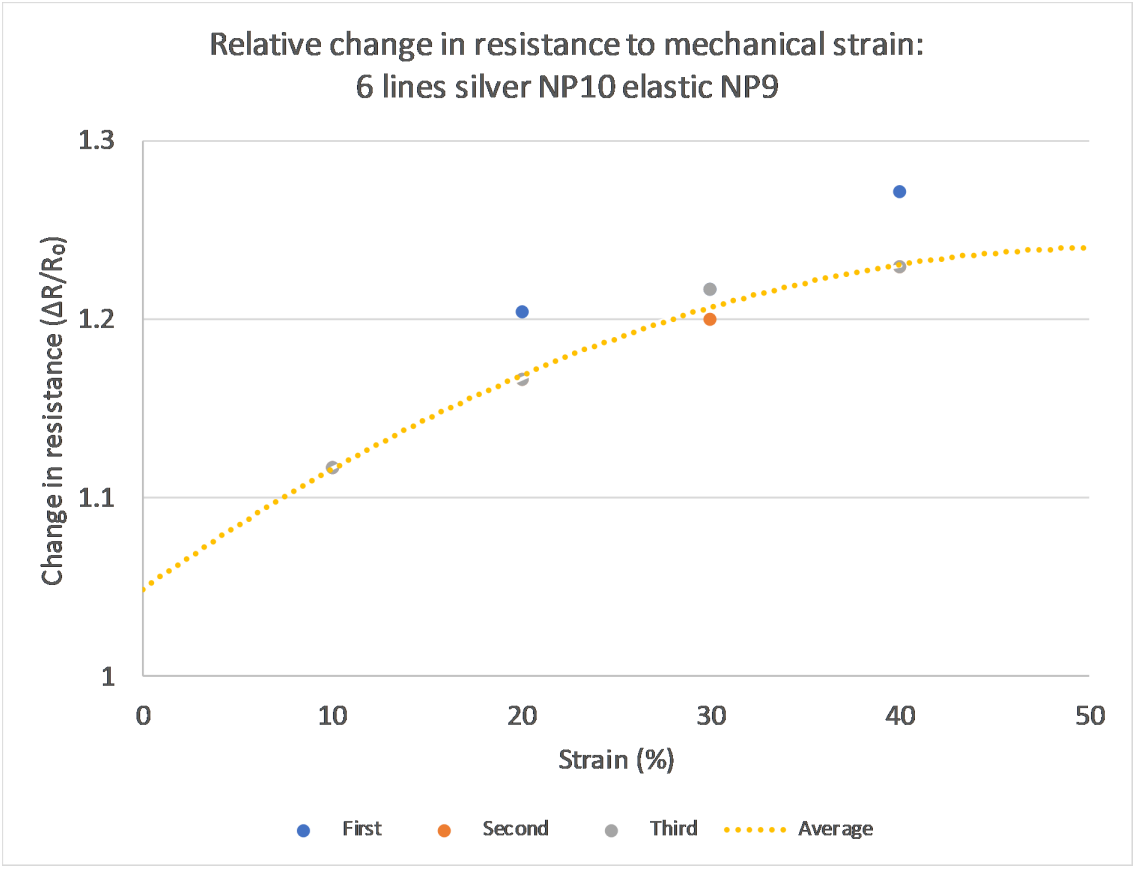

Resistance change over time: 6 lines silver NP9 elastic NP9
Resistance (cΩ)
310
300
290
280
270
260
250
240
230
220
210
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
Times (s)
10% strain
20% strain
30% strain
40% strain

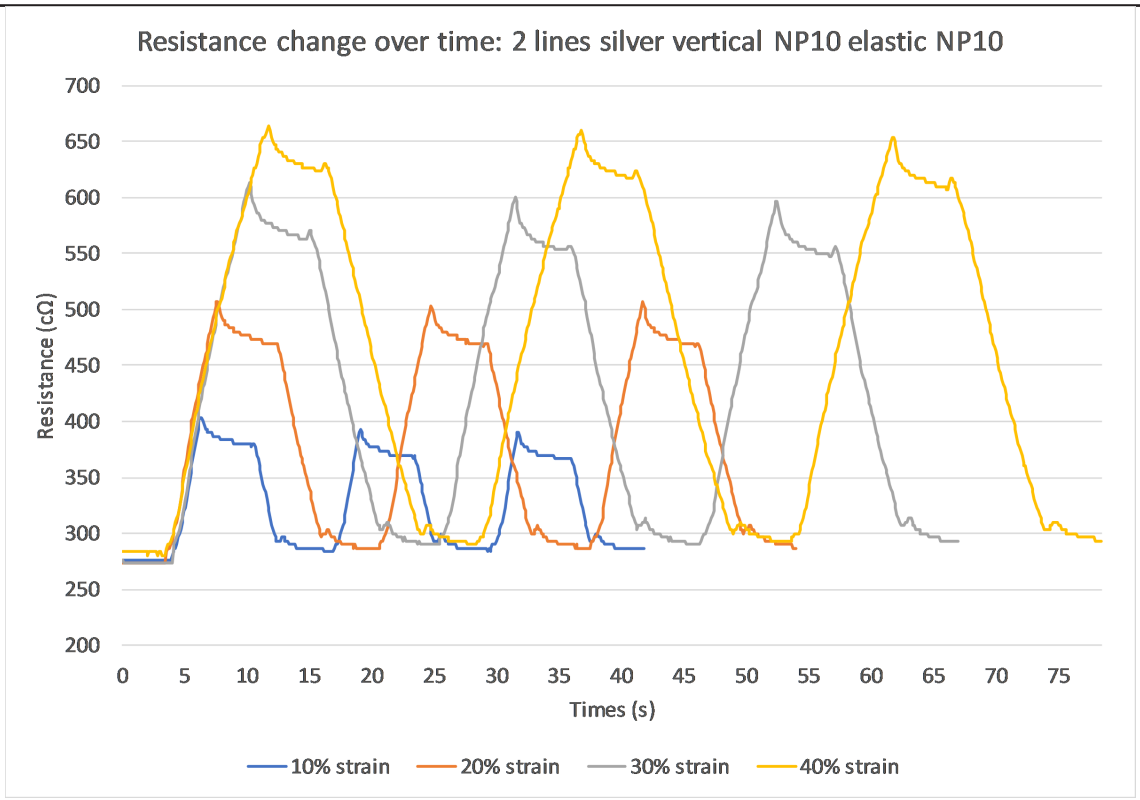

Relative change in resistance to mechanical strain:
6 lines silver NP9 elastic NP9
Change in resistance (ΔR/R₀)
1.1
1
0.9
0.8
0
10
20
30
40
50
Strain (%)
First
Second
Third
Average

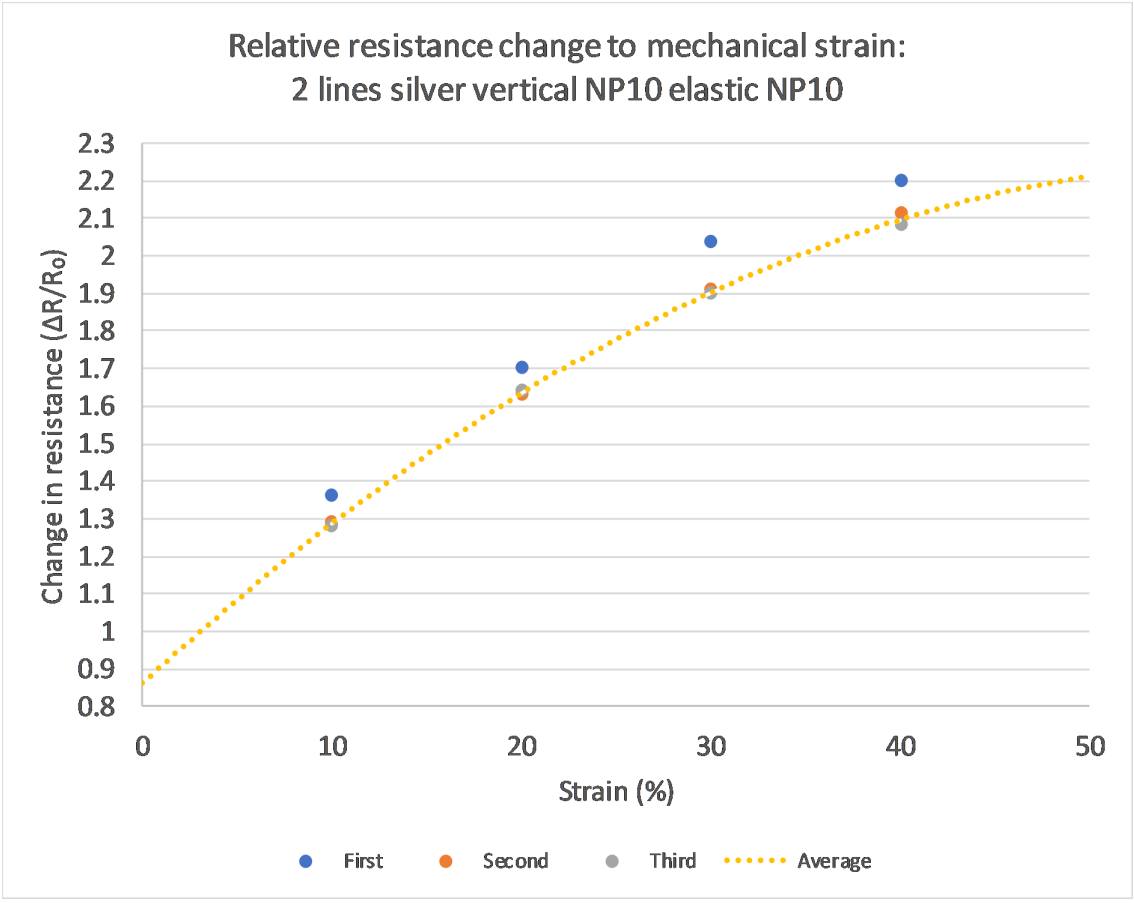

Resistance change over time: 6 lines silver NP9 elastic NP10
Resistance (cΩ)
260
250
240
230
220
210
200
0
5
10
15
20
25
30
35
40
45
50
55
60
65
Times (s)
10% strain
20% strain
30% strain
40% strain

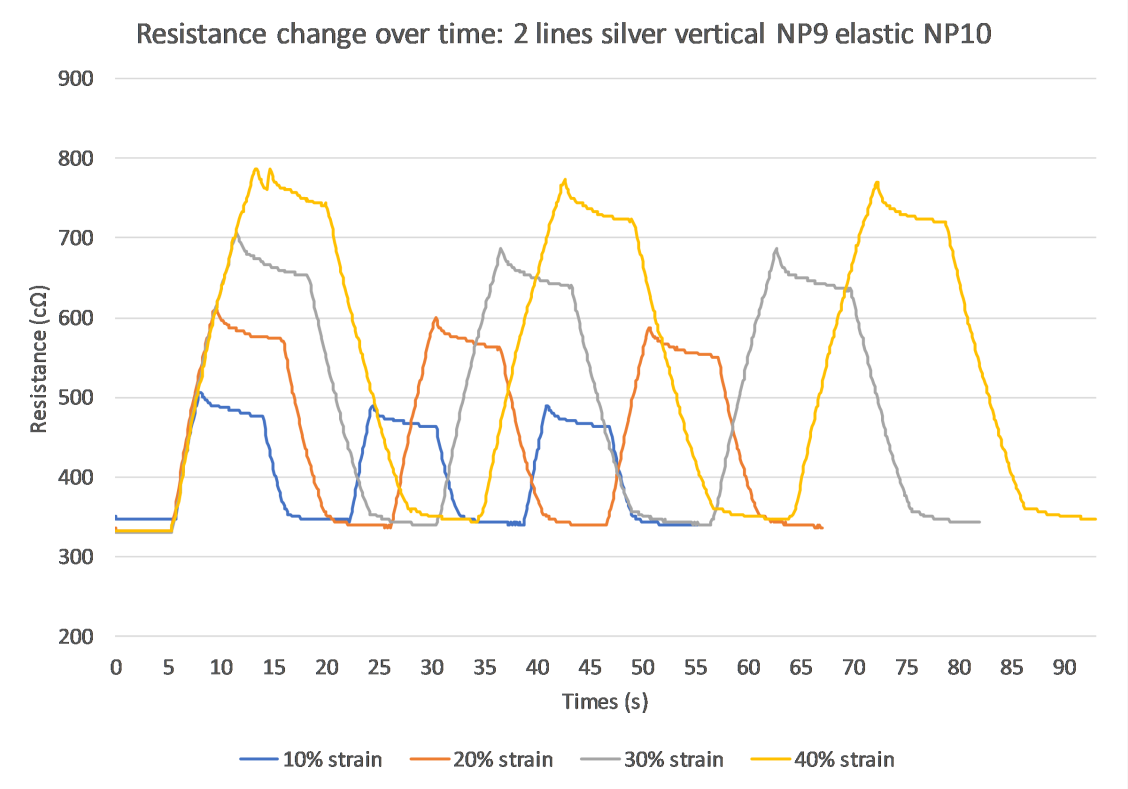

Relative change in resistance to mechanical strain:
6 lines silver NP9 Elastic NP10
Change in resistance (ΔR/R₀)
1.3
1.2
1.1
1
0
10
20
30
40
50
Strain (%)
First
Second
Third
Average

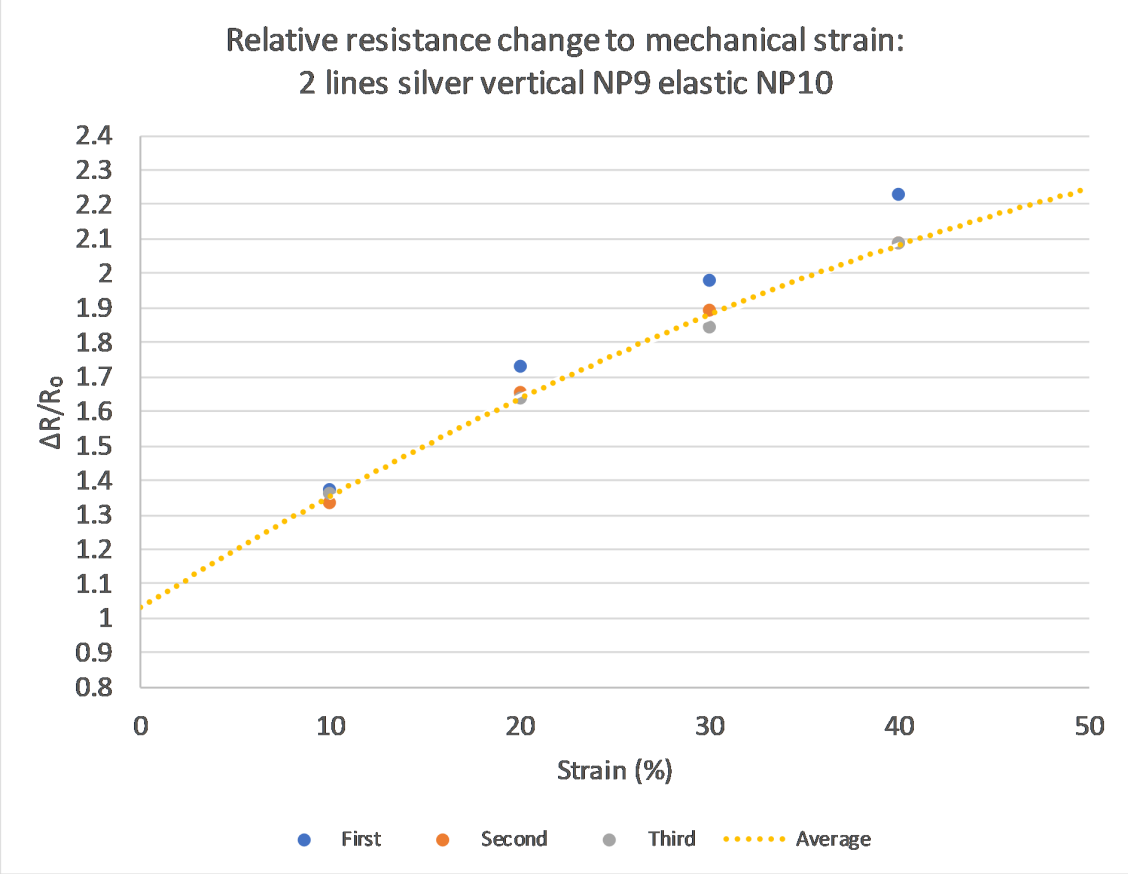

Resistance change over time: 6 lines silver NP10 elastic NP9
260
250
240
230
220
210
200
190
Resistance (cΩ)
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80
Times (s)
10% strain
20% strain
30% strain
40% strain


Relative change in resistance to mechanical strain:
6 lines silver NP10 elastic NP9
1.3
1.2
1.1
1
Change in resistance (ΔR/R₀)
0 10 20 30 40 50
Strain (%)
First
Second
Third
Average


Resistance change over time: 2 lines silver vertical NP10 elastic NP10
Resistance (cΩ)
700
650
600
550
500
450
400
350
300
250
200
0
5
10
15
20
25
30
35
40
45
50
55
60
65
70
75
Times (s)
10% strain
20% strain
30% strain
40% strain


Relative resistance change to mechanical strain:
2 lines silver vertical NP10 elastic NP10
Change in resistance (ΔR/Ro)
2.3
2.2
2.1
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0
10
20
30
40
50
Strain (%)
First
Second
Third
Average


Resistance change over time: 2 lines silver vertical NP9 elastic NP10
Resistance (cΩ)
900
800
700
600
500
400
300
200
0 5 10 15 20 25 30 35 40 45 50 55 60 65 70 75 80 85 90
Times (s)
10% strain
20% strain
30% strain
40% strain


Relative resistance change to mechanical strain:
2 lines silver vertical NP9 elastic NP10
ΔR/Ro
2.4
2.3
2.2
2.1
2
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1
0.9
0.8
0 10 20 30 40 50
Strain (%)
First
Second
Third
Average

Appendix D










RIO DE RESTAURACIÓN











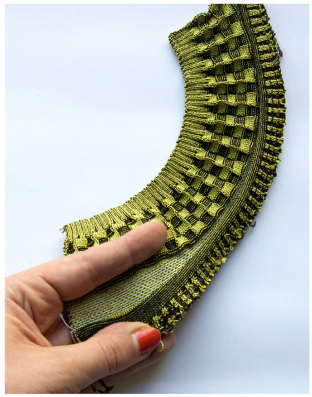


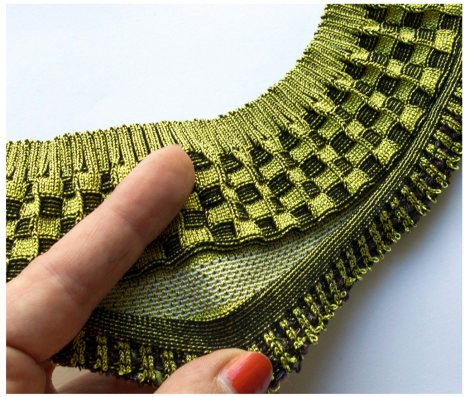
Structures: Hidden space for added functions

















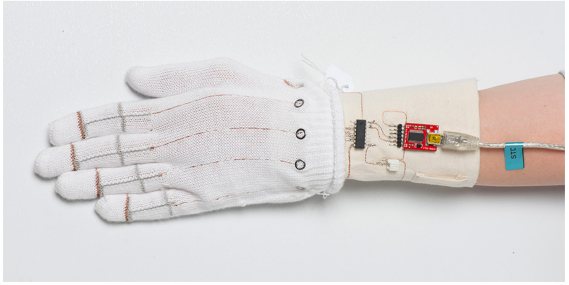






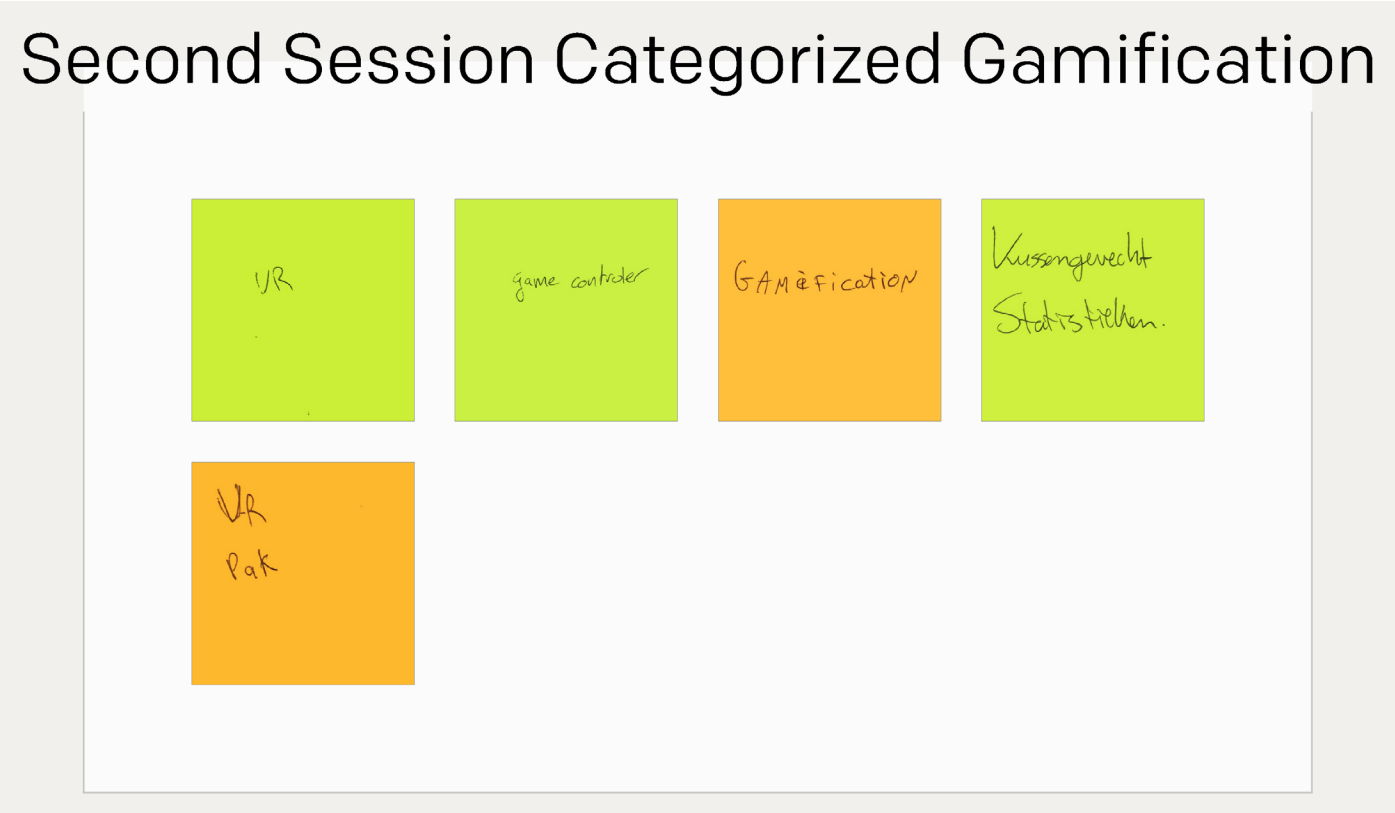








Appendix E


Second Session Categorized Gamification
VR
game controller
GAMEfication
Kussengevecht
Statistieken.
VR
Pak

Second Session Categorized Interface

Soort 3D muis applicatie.

Ergonomische muis.

Ruimtepak. ↓ bediening in handschoen.

STOFFEN MUUR INTERFACE





Knuffelbare interface e.g. Lampenkap

Stoffen knoppen in de auto.

stoffen toetsenbord

INTERACTIEF KINDER BOEK

Aansturing exoskelet (in de bouw)

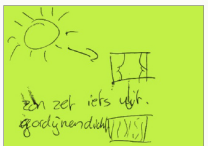
Make HARD products soft

Knuffel communicatie ↓ internet ↓ knuffel

TV REMOTE

BANK SENSOR ↓ licht gaat aan

elektrische Tekharmonica



Leip 3D muziek instrument /syntesizer.



Second Session Categorized Medical



Second Session Categorized Safety

uitzettingsalarm

autogordel, als je uit elkaar trekt is strakker/ losser

klimtouw dat aangeeft hoe ver de reep was bij een val

Safety margin

bij zoveel reep gaat het alarm af.

waterdruk

duiken.

Net op aanhangwagen

checken of het nog goed zit.

flex-meter, is een materiaal te oud en star geworden?

BUNGEE jump

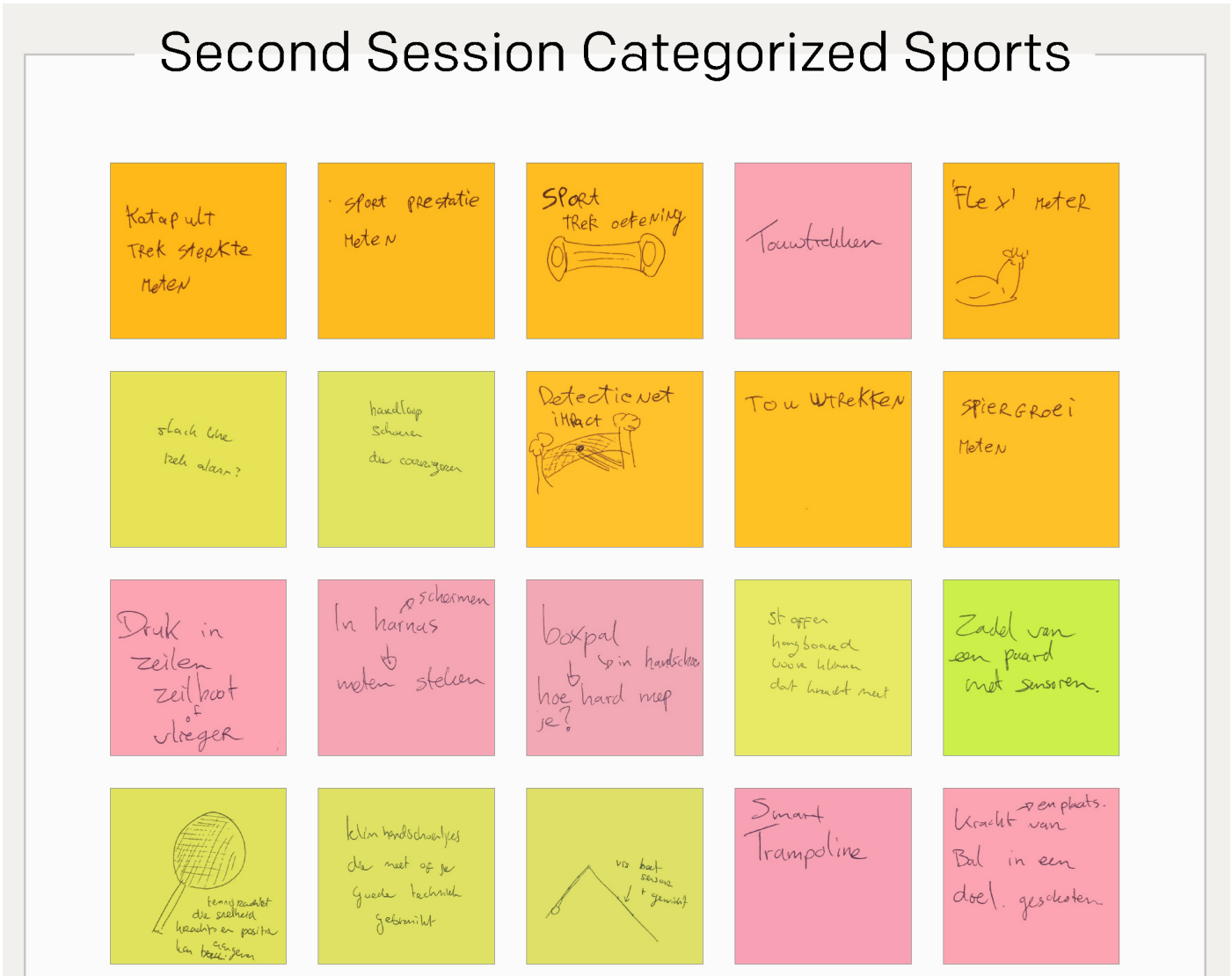
slijtage



Druk in vlieger of parachute. (lim vet)

Hgsbanden

Second Session Categorized Sports

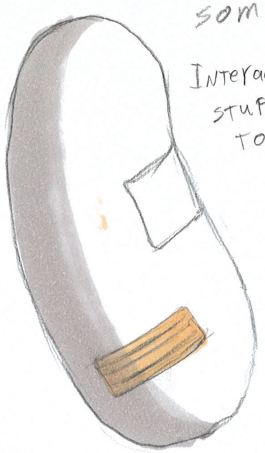


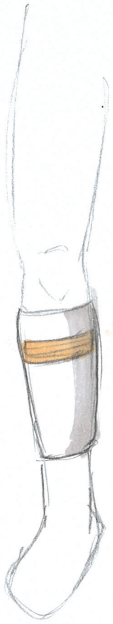
Second Session Categorized Weight/volume Measuring



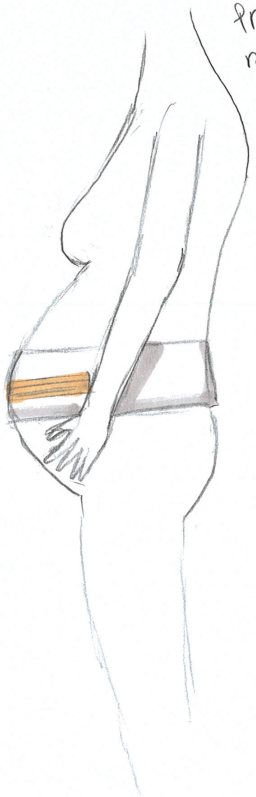
somnox

Interactive stuffed toy/sleeping buddy

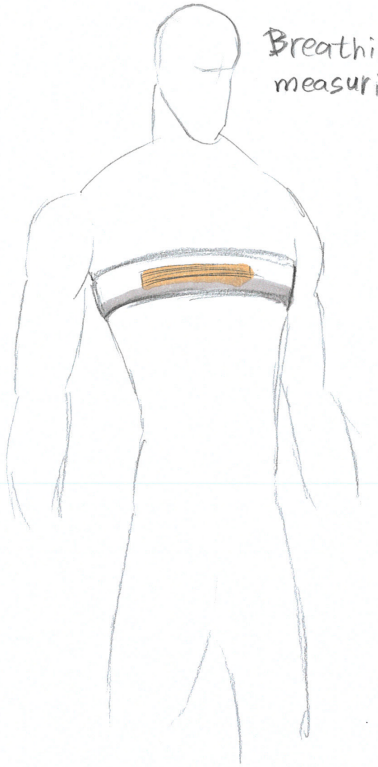




swollen leg measuring



Pregnancy measuring



Breathing measuring

stress toy with sensor





stretch it a lot

↓

measure stress

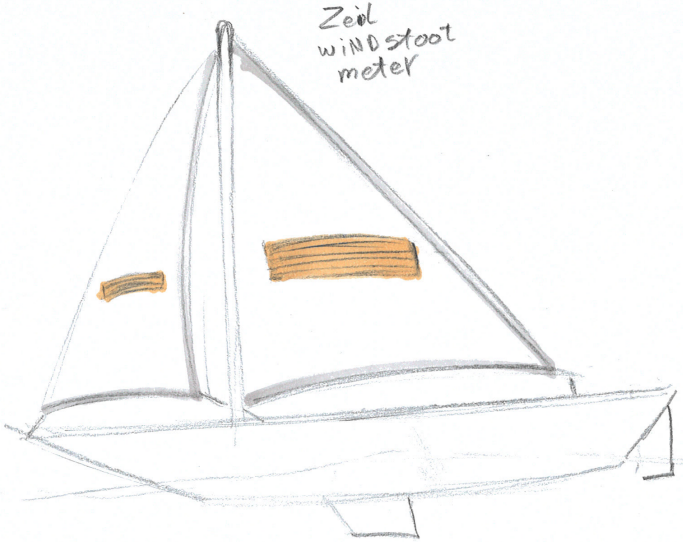


Flexibility Measuring



windsok

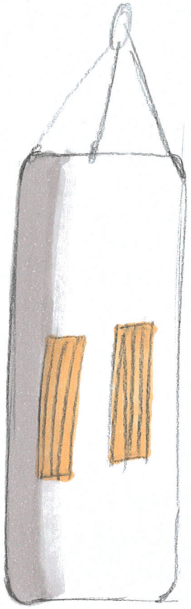
Die wind snelheid kan meten



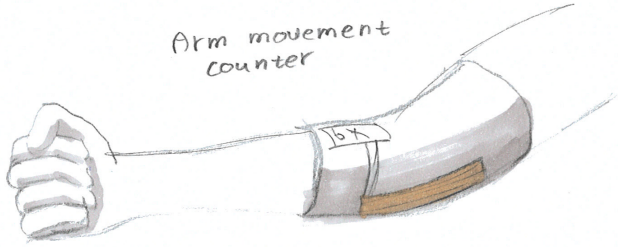
Zeil windstoot meter

Posture measuring

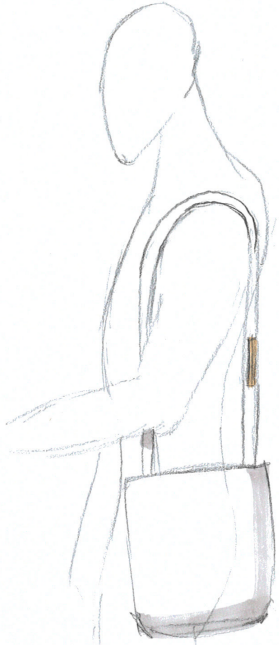




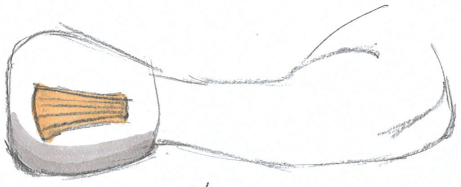
Stootzak Die stoten Meet



Arm movement counter



Tas Die Gewicht meet



Boxhandschoen Die stoten Meet.

Appendix F

Showmode_CreativeSession | Arduino 1.8.10

File Edit Sketch Tools Help

Showmode_CreativeSession

```
int analogPin = A0;
int raw = 0;
int Vin = 5;
float Vout = 0;
float R1 = 1000;
float R2 = 0;
float buffer = 0;

int LED1 = 5;
int LED2 = 4;
int LED3 = 3;
int LED4 = 6;

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

void loop()
{
  raw = ((analogRead(analogPin)+analogRead(analogPin)+analogRead(analogPin)+analogRead(analogPin)+analogRead(analogPin))/5);
  //raw = 1023 - raw;
  Serial.println(raw);
    if (raw <= 580) {
      digitalWrite(LED1, LOW);
      digitalWrite(LED2, LOW);
      digitalWrite(LED3, LOW);
      digitalWrite(LED4, LOW);
    }
    if (raw > 580) {
      digitalWrite(LED1, HIGH);
      digitalWrite(LED2, LOW);
      digitalWrite(LED3, LOW);
      digitalWrite(LED4, LOW);
    }
    if (raw > 583) {
      digitalWrite(LED1, HIGH);
      digitalWrite(LED2, HIGH);
      digitalWrite(LED3, LOW);
      digitalWrite(LED4, LOW);
    }
    if (raw > 586) {
      digitalWrite(LED1, HIGH);
      digitalWrite(LED2, HIGH);
      digitalWrite(LED3, HIGH);
      digitalWrite(LED4, LOW);
    }
    if (raw > 590) {
      digitalWrite(LED1, HIGH);
      digitalWrite(LED2, HIGH);
      digitalWrite(LED3, HIGH);
      digitalWrite(LED4, HIGH);
    }
    delay(100);
  }
```

1

Appendix G

portable_measuring_setup_2 | Arduino 1.8.10

File Edit Sketch Tools Help

portable_measuring_setup_2

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

// Set the pins used
#define cardSelect 4

#if defined(ARDUINO_SAMD_ZERO) && defined(SERIAL_PORT_USBVIRTUAL)
  // Required for Serial on Zero based boards
  #define Serial SERIAL_PORT_USBVIRTUAL
#endif

int state = LOW;
int reading = LOW;           // the current reading from the input pin
int previous;    // the previous reading from the input pin
int statePrevious = LOW;
int inPin = 12;         // the number of the input pin, button
int resistance = A0;
boolean createFile = true;

long time = 0;         // the last time the output pin was toggled
long debounce = 200;   // the debounce time, increase if the output flickers

File logfile;

// blink out an error code
void error(uint8_t errno) {
  while (1) {
    uint8_t i;
    for (i = 0; i < errno; i++) {
      digitalWrite(13, HIGH);
      delay(100);
      digitalWrite(13, LOW);
      delay(100);
    }
    for (i = errno; i < 10; i++) {
      delay(200);
      //Serial.print(i);
    }
  }
}



void setup() {
  Serial.begin(115200);
  if (!SD.begin(cardSelect)) {
    Serial.println("Card init. failed!");
    error(2);
  }
  pinMode(13, OUTPUT);

}

void loop()
{
  reading = digitalRead(inPin);
  //Serial.print(reading);

  // if the input just went from LOW and HIGH and we've waited long enough
  // to ignore any noise on the circuit, toggle the output pin and remember
  // the time
  if (reading == HIGH && previous == LOW && millis() - time > debounce) {
    if (state == HIGH)
      state = LOW;
    else
      state = HIGH;

    time = millis();
  }
  previous = reading;
```

```
  if (state == HIGH) {


    if (createFile == true) {
      //Create non existing new file
      char filename[15];
      strcpy(filename, "/TEST00.TXT");
      for (uint8_t i = 0; i < 100; i++) {
        filename[5] = '0' + i / 10;
        filename[6] = '0' + i % 10;
        // create if does not exist, do not open existing, write, sync after write
        if (! SD.exists(filename)) {
          break;
        }
      }

      logfile = SD.open(filename, FILE_WRITE);
      if ( ! logfile ) {
        Serial.print("Couldnt create ");
        Serial.println(filename);
        error(3);
      }
      Serial.print("Writing to ");
      Serial.println(filename);

      pinMode(13, OUTPUT);
      pinMode(8, OUTPUT);

      delay(100);
      digitalWrite(13, HIGH);
      delay(100);
      digitalWrite(13, LOW);
      delay(100);
      createFile = false;
    }


    uint8_t i = 0;
    //METEN VAN ANALOGE PIN
    digitalWrite(8, HIGH);
    logfile.println(analogRead(0));
    digitalWrite(8, LOW);
    delay(100);

  }

  if (state == LOW && previous == HIGH) {
    createFile = true;
    logfile.close();
    delay(100);
    digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
    delay(100);
    digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
    delay(100);
  }
  //Meting uit
}
```



Appendix H

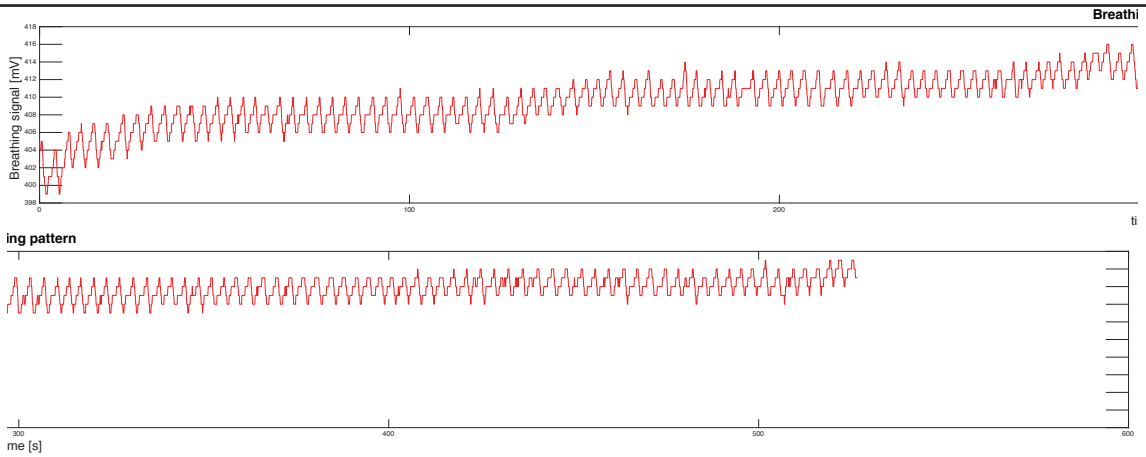



```
#include <SPI.h>
#include <SD.h>

// Set the pins used
#define cardSelect 4

#if defined(ARDUINO_SAMD_ZERO) && defined(SERIAL_PORT_USBVIRTUAL)
// Required for Serial on Zero based boards
#define Serial SERIAL_PORT_USBVIRTUAL
#endif

int state = LOW;
int reading = LOW;           // the current reading from the input pin
int previous;    // the previous reading from the input pin
int statePrevious = LOW;
int inPin = 12;         // the number of the input pin, button
int resistanceBreath = A1;    //sensor analog port
int resistanceArm = A4;
boolean createFile = true;

long time = 0;         // the last time the output pin was toggled
long debounce = 200;   // the debounce time, increase if the output flickers

File logfile;

// blink out an error code
void error(uint8_t errno) {
  while (1) {
    uint8_t i;
    for (i = 0; i < errno; i++) {
      digitalWrite(13, HIGH);
      delay(100);
      digitalWrite(13, LOW);
      delay(100);
    }
    for (i = errno; i < 10; i++) {
      delay(200);
      //Serial.print(i);
    }
  }
}
```

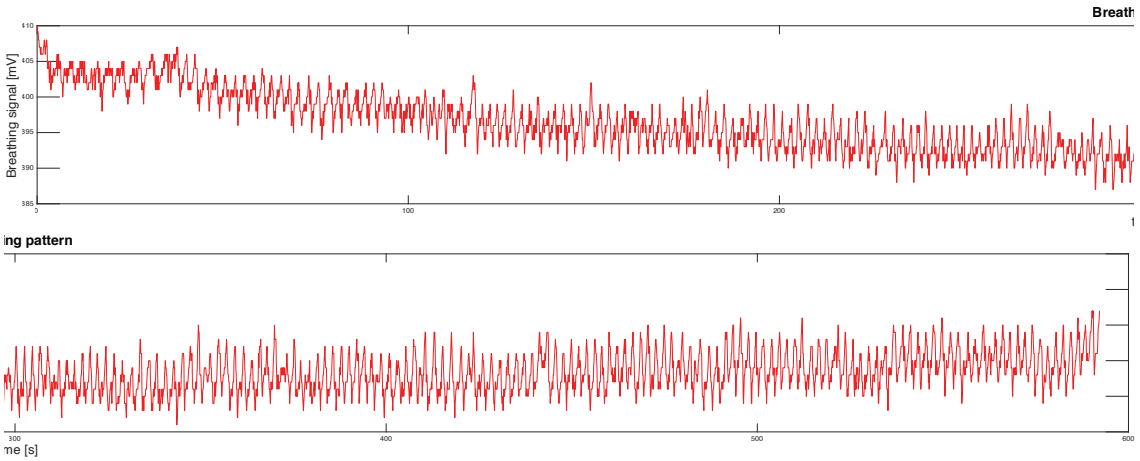
```
void setup() {
  Serial.begin(115200);
  if (!SD.begin(cardSelect)) {
    Serial.println("Card init. failed!");
    error(2);
  }
  pinMode(13, OUTPUT);

}

void loop()
{
  reading = digitalRead(inPin);
  //Serial.print(reading);

  // if the input just went from LOW and HIGH and we've waited long enough
  // to ignore any noise on the circuit, toggle the output pin and remember
  // the time
  if (reading == HIGH && previous == LOW && millis() - time > debounce) {
    if (state == HIGH)
      state = LOW;
    else
      state = HIGH;

    time = millis();
  }
  previous = reading;


  if (state == HIGH) {


    if (createFile == true) {
      //Create non existing new file
      char filename[15];
      strcpy(filename, "/TEST00.TXT");
      for (uint8_t i = 0; i < 100; i++) {
        filename[5] = '0' + i / 10;
        filename[6] = '0' + i % 10;
        // create if does not exist, do not open existing, write, sync after write
        if (! SD.exists(filename)) {
          break;
        }
      }

      logfile = SD.open(filename, FILE_WRITE);
      if ( ! logfile ) {
        Serial.print("Couldnt create ");
        Serial.println(filename);
        error(3);
      }
      Serial.print("Writing to ");
      Serial.println(filename);
```

```
    pinMode(13, OUTPUT);
    pinMode(8, OUTPUT);
    pinMode(10, OUTPUT);

    delay(100);
    /*digitalWrite(13, HIGH);
      delay(100);
      digitalWrite(13, LOW);
      delay(100);*/
    tone(10, 2500, 100);
    delay(200);
    createFile = false;
  }

  uint8_t i = 0;
  //METEN VAN ANALOGE PIN
  digitalWrite(8, HIGH);
  /*Serial.print(resistanceBreath);     //voor serial port grafiek maken tijdens filmen
  Serial.print("\t");
  Serial.println(resistanceArm);*/
  logfile.print(analogRead(1));
  logfile.print("\t");
  logfile.println(analogRead(4));
  digitalWrite(8, LOW);
  delay(100);
}

if (state == LOW && previous == HIGH) {
  createFile = true;
  logfile.close();
  delay(100);
  /*digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
    delay(100);
    digitalWrite(13, HIGH);
    delay(100);
    digitalWrite(13, LOW);
    delay(100);*/
  tone(10, 2500, 100);
  delay(140);
  tone(10, 2500, 100);
  delay(140);
}
//Meting uit
}
```

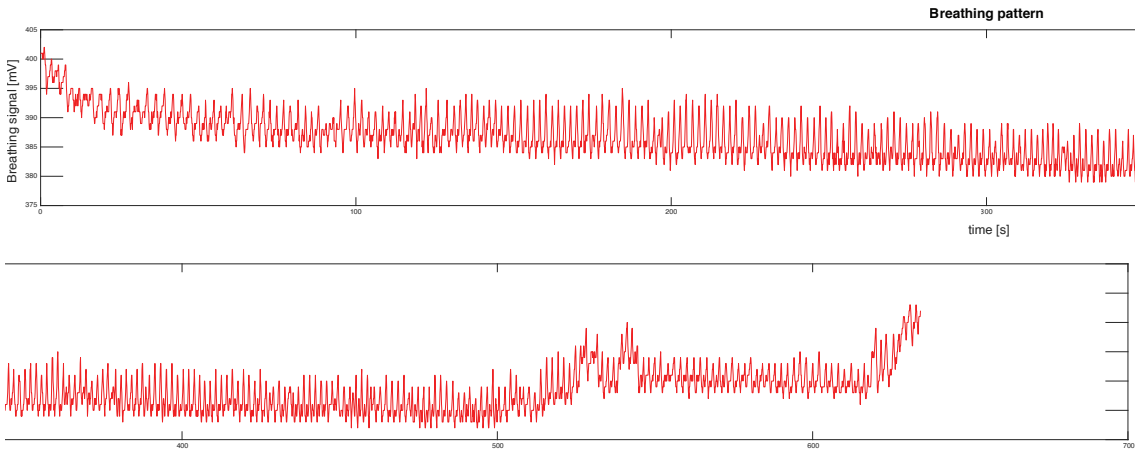
Appendix I



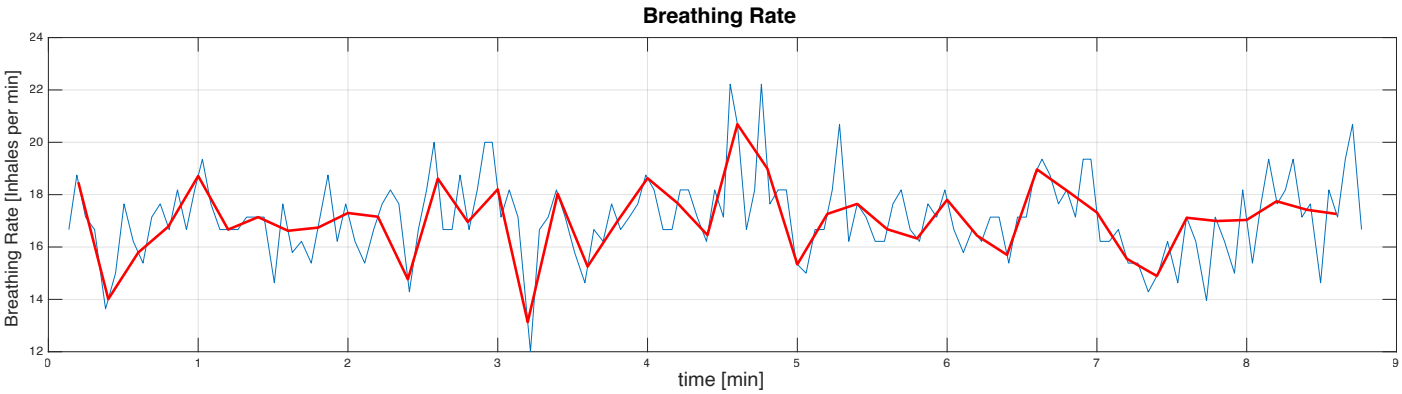
Breathing graph when sitting still for 10 minutes.



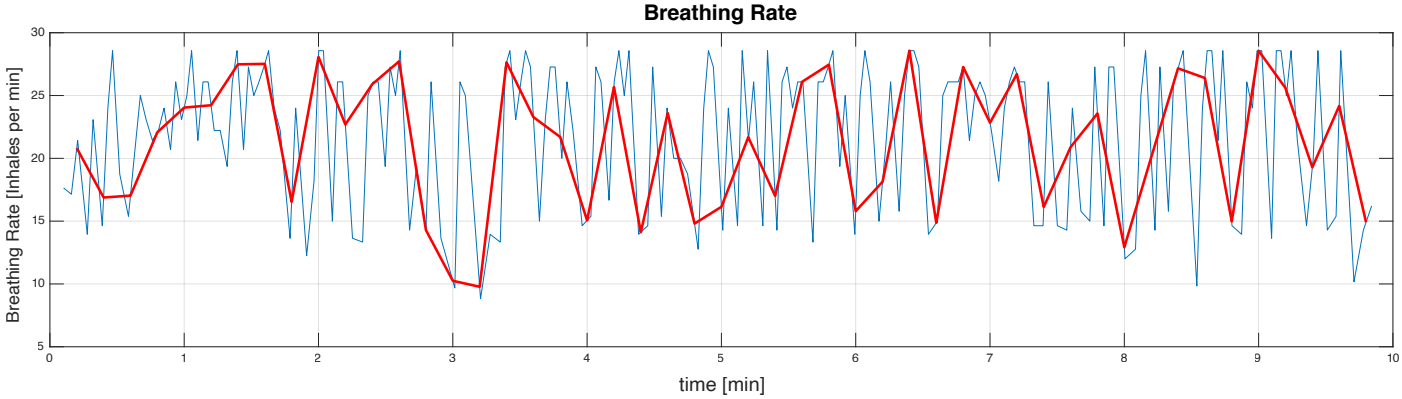
Breathing graph when walking up and down the stairs for 10 minutes.



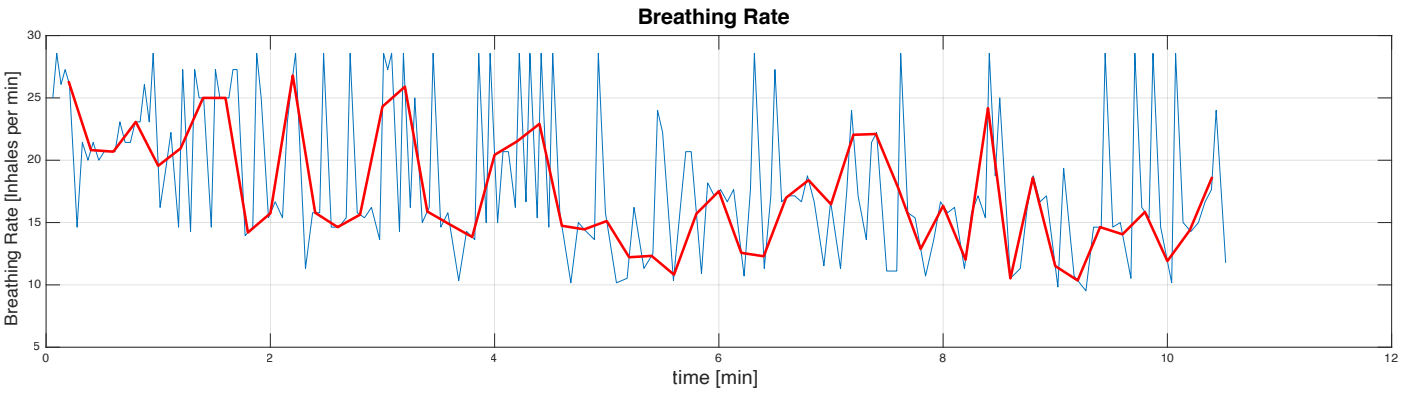
Breathing graph when cycling for 10 minutes.



Breathing rate graph when sitting still for 10 minutes.



Breathing rate graph when walking up and down the stairs for 10 minutes.



Breathing rate graph when cycling for 10 minutes.

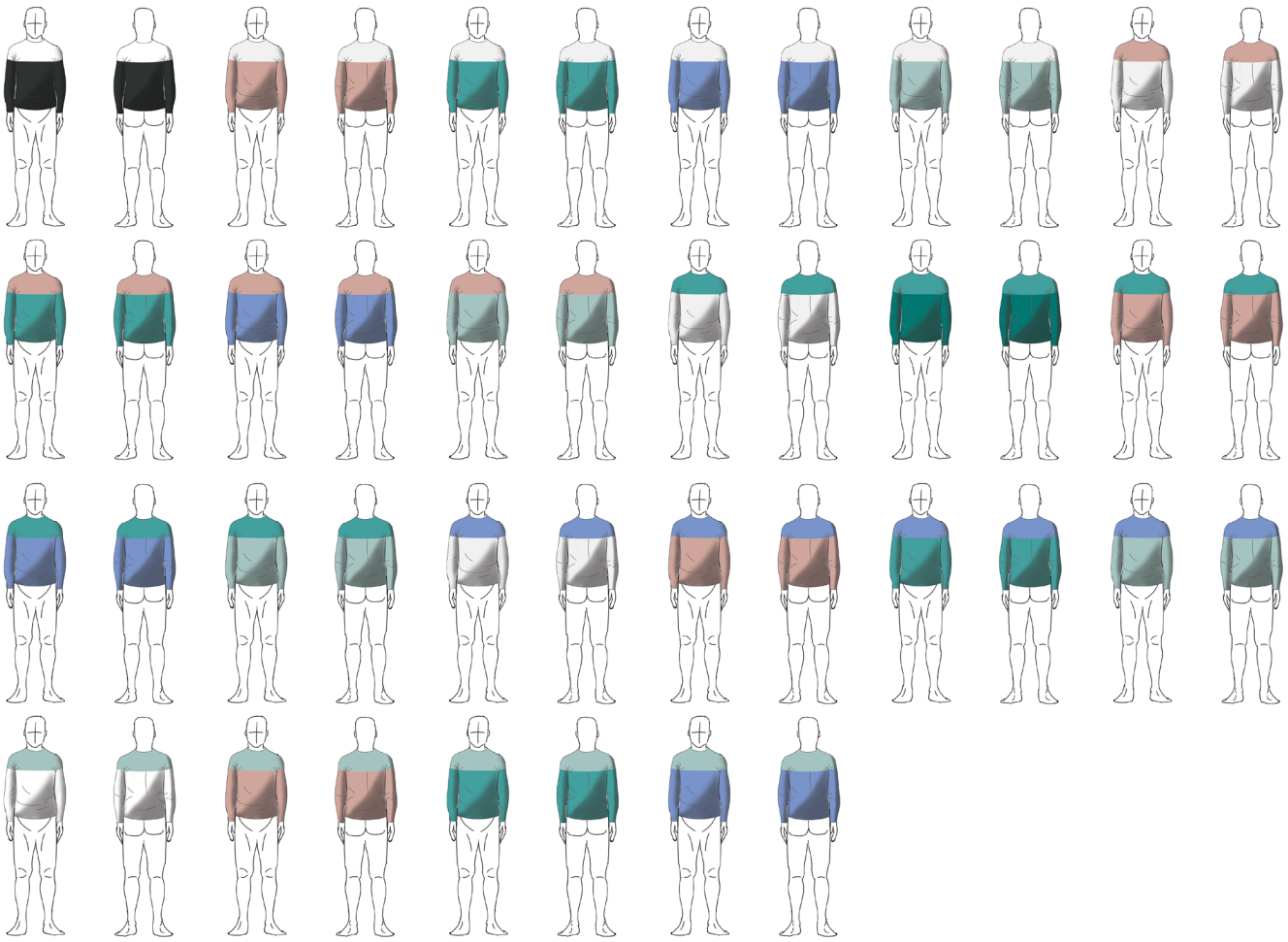
Appendix J

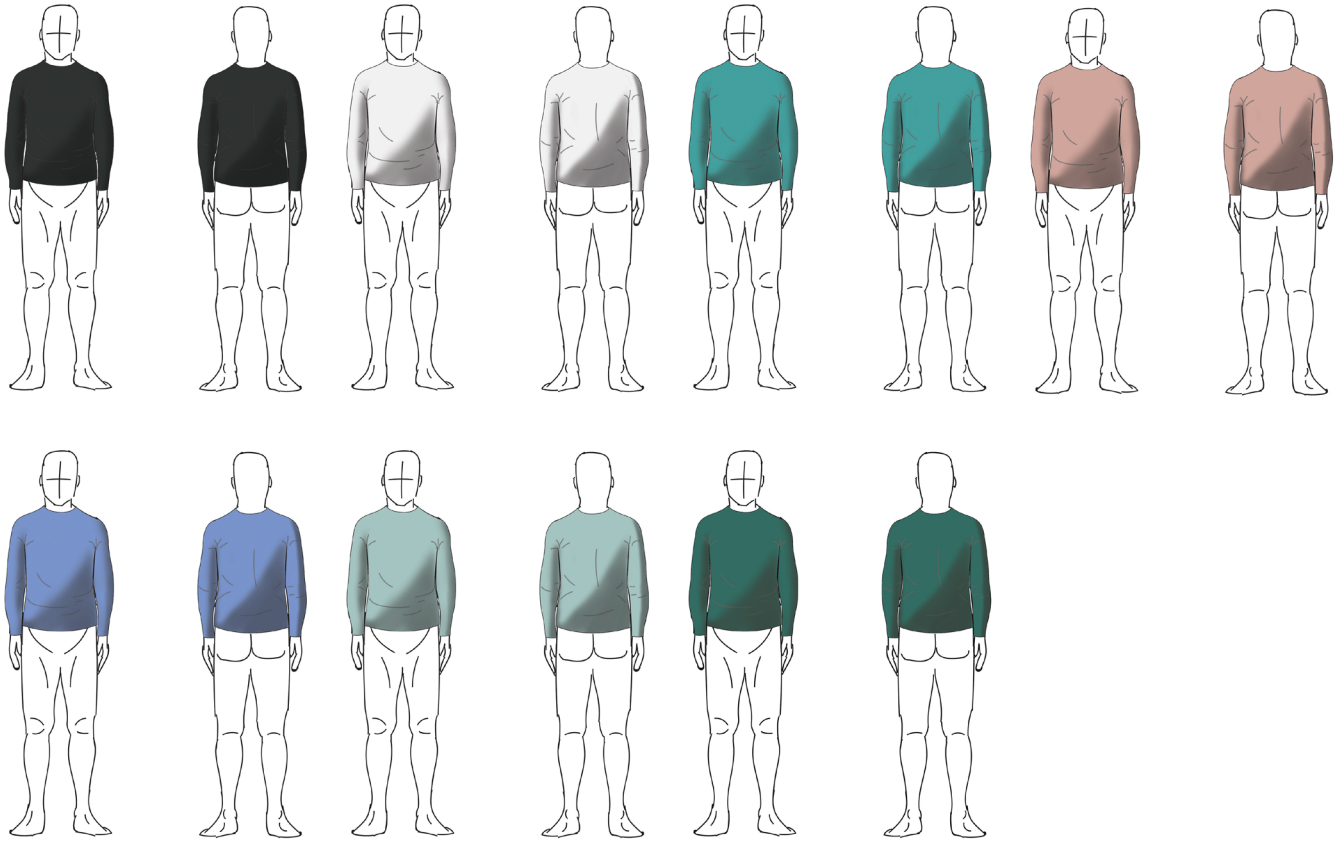
BreathingRead.m

```
% BreathingRead.m
% for ready data of the knitted breating sensor (Daan van dr Valk)
%                          KMB Jansen version 31-3-2020

clc;
clear;
close all;

filedir = 'E:\Studie\Studie\Graduation\Arduino stuff\Breathing tests\';
%file = 'Breathing test 10 minutes.xlsx';
file = '10 minutes sitting still 12-bit2.xlsx';


%% Reading of data
file_in = char( strcat(filedir,file) );
dat = xlsread(file_in);

t_s   = dat(:,1);   % time [s]
t_m   = t_s/60;     % time [min]
sig = dat(:,2);     % signal [mV]

% Plotting
figure(1); clf;
  subplot(2,1,1);
    plot(t_s,sig,'r'); hold on;
    xlabel('time [s]'); ylabel('Breathing signal [mV]');
    title('Breathing pattern')

%% Data evaluation
% First Filtering; see help filter
win = 9; a=1; b=(1/win)*ones(1,win);   % win = window size
sig_f = filter(b,a, sig);
sig_f(1:win) = sig_f(win); % adapt first values afte filtering
    plot(t_s,sig_f,'g');

% See also Matlab help on Peak Analysis
% normal peakdistance is about 3.5-4 secs [0.29-0.25/sec or 17 .. 15/min]
%         % Treshold in the minumum distance on the time axis [s]
%         % in this case the resolution itself is 1 mV
%         % MinPeakProminence, the minumum vertical distance [mV]
%         % MinPeakDistance: minimum separation between peaks
[pks, locs_f] = findpeaks(sig_f,t_s,'MinPeakDistance',2); % find peaks and locs for filtered
                %findpeaks(sig_f,t_s,'MinPeakDistance',2); % plots filtered data and peaks

% Breathing rate from peak distances
peakIntervals = diff(locs_f);     % time intervals between breathing [s]
t_s_intervals = locs_f(2:end);    % one point less
t_m_intervals = t_s_intervals/60;    % interval times in mins
BreathingRate = 60./peakIntervals;

figure(1);
subplot(2,1,2);
    hist(BreathingRate,0:2:30)
    grid on
    xlabel('Breathing Rate [Inhales per min]')
    ylabel('Frequency of Occurrence')
    title('Histogram of Breathing Rate (BpM)')

%% Now calculate the slowly changing BpM output
int_s = 12;          % output time seconds
int_m = int_s/60;    % output time in minutes
t_out = (0:int_m:t_m(end))'; % new time scale for output
bpm_out = interp1(t_m_intervals,BreathingRate, t_out);

figure(2);clf
    plot(t_m_intervals,BreathingRate); hold on
    plot(t_out,bpm_out,'r')
    grid on
    xlabel('time [min]')
    ylabel('Breathing Rate [Inhales per min]')
    title('Breathing Rate')

 %% Output

xlswrite(file_in,{'time[min]','breath/min'},'','D1');
xlswrite(file_in,[t_out,bpm_out],'','D2');
```

Appendix K

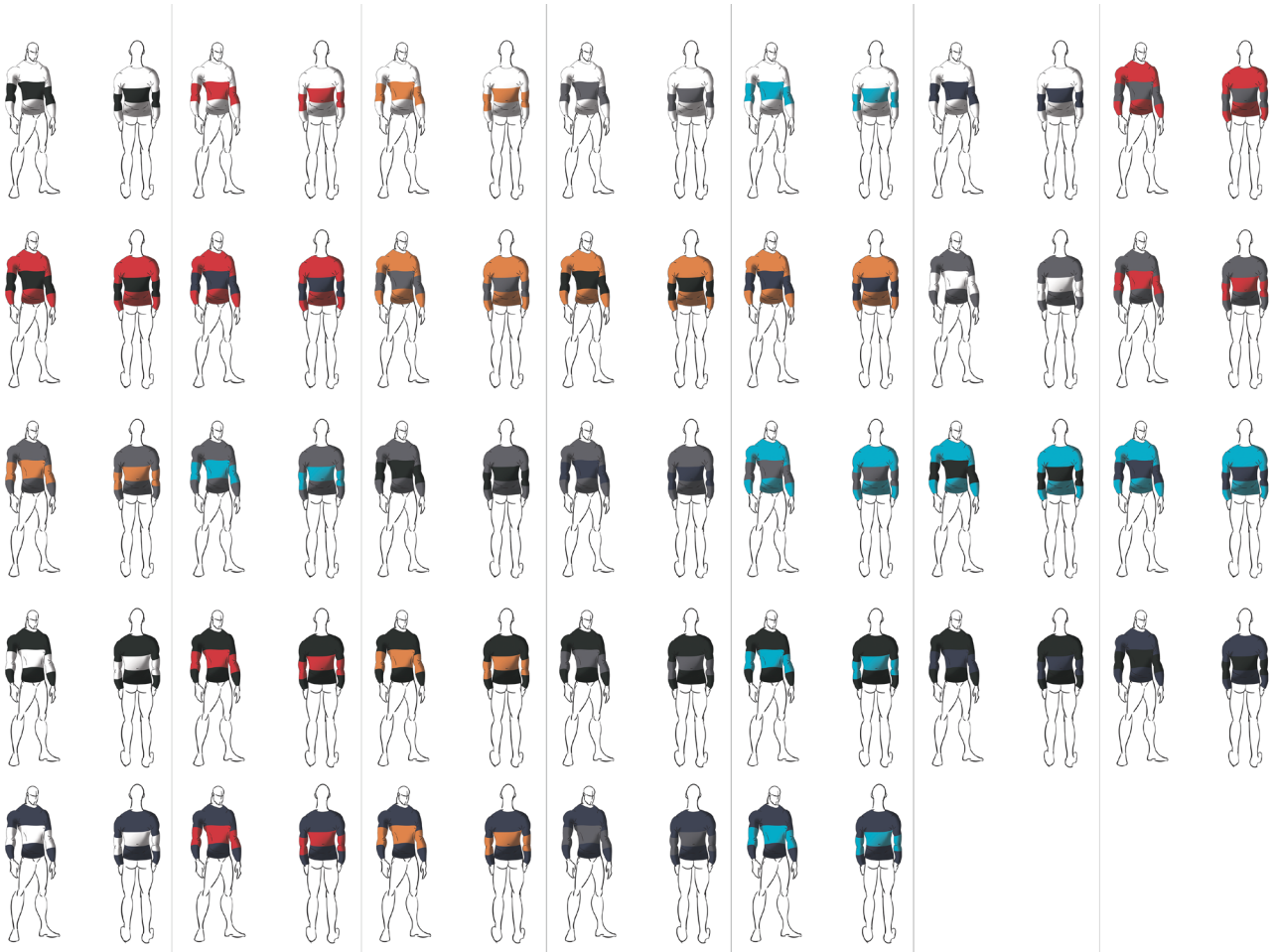




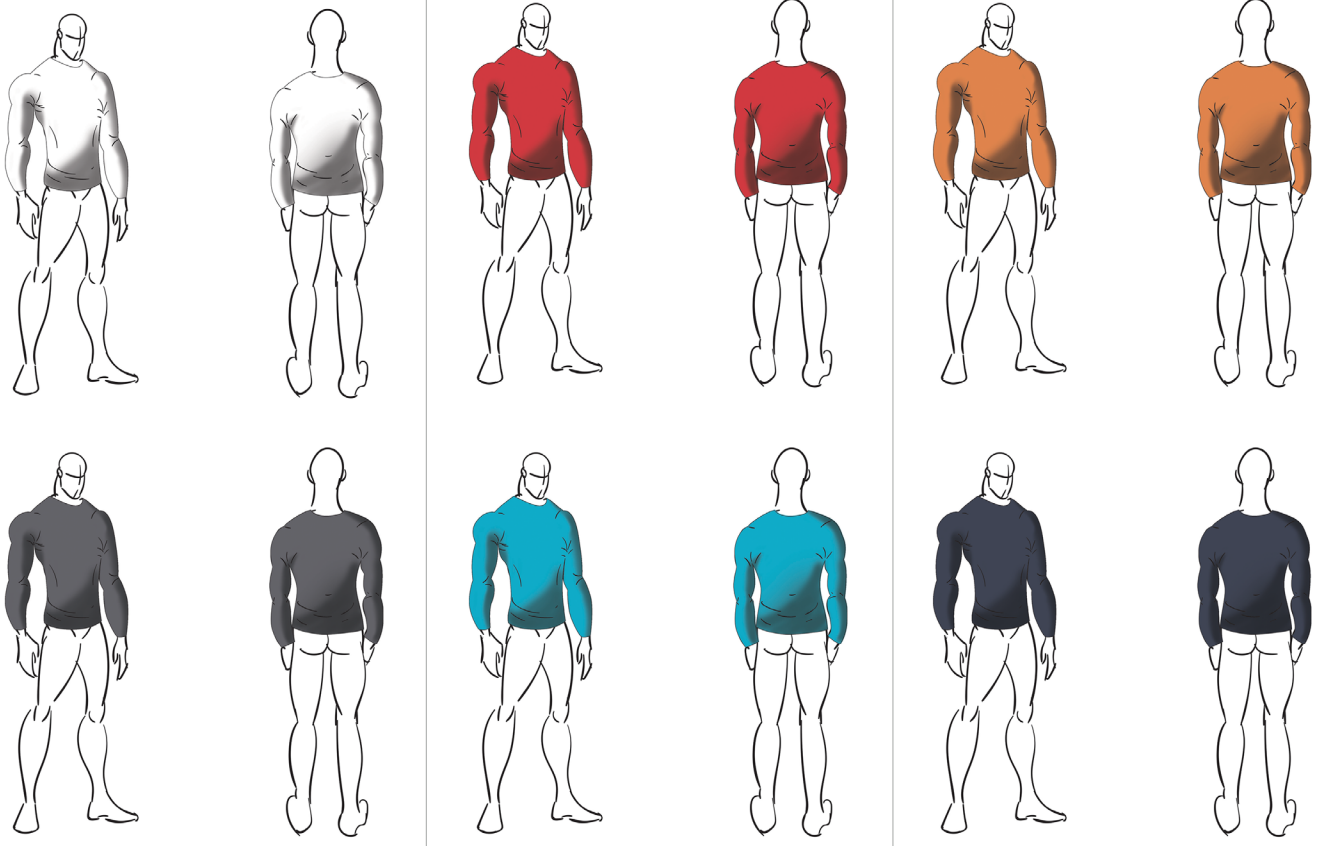


Appendix L









Appendix M

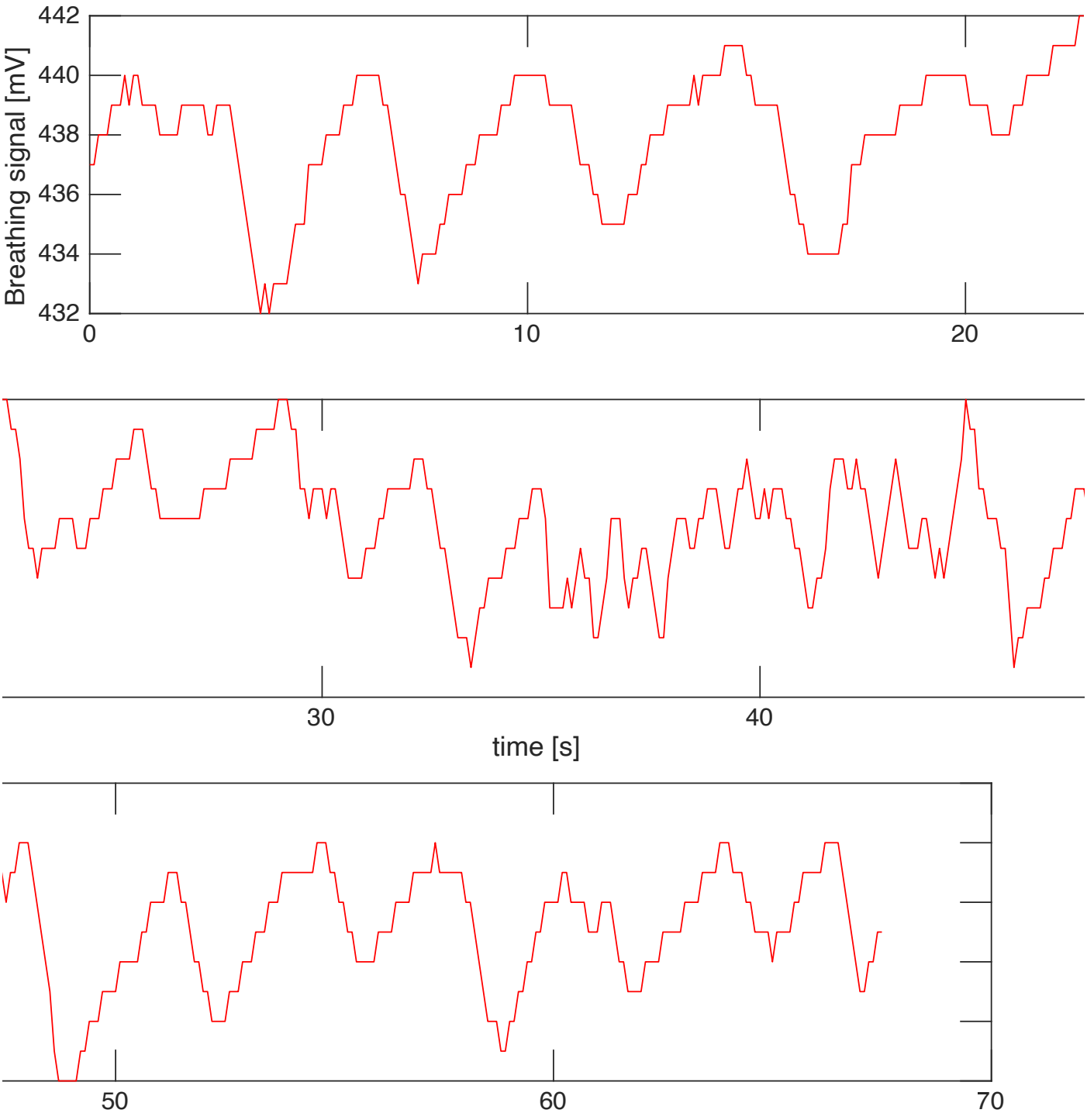


Appendix N

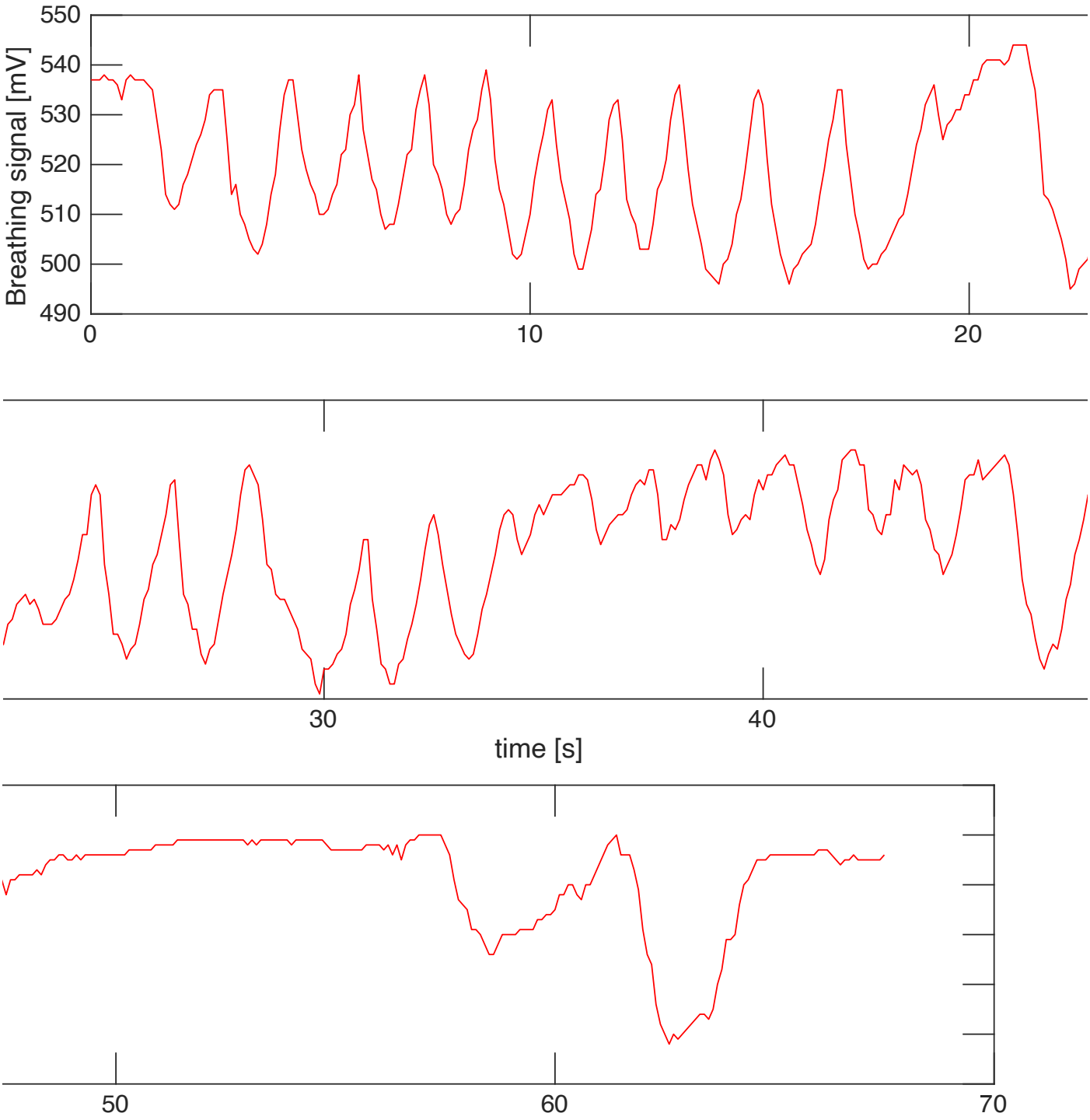
1. First, the wearer of the shirt will sit still to measure the breathing pattern
2. Secondly, the user will move the arm from the elbow up and down, starting with an extension and compression of about 25% for 5 times. This is repeated for 50, 75 and 100%. At a 100% the hand is almost touching the shoulder.
3. Arm rotation to the side, where the lower arm stays horizontal
4. Arm stretched in front of the body
5. Arm stretched to right of the body
6. Arm stretched up
7. Arm stretched to the left of the body

Appendix O

Breathing graph in movie



Arm movement graph in movie



Breathing rate graph

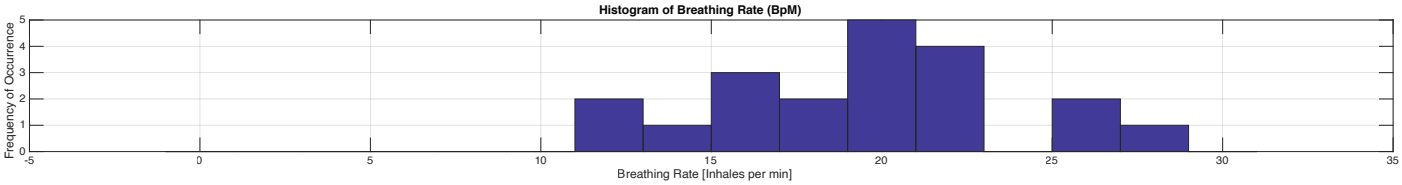




Foto: Sam Rentmeester