

Sense your heart

Giorgio Campo, Jelle Dekker, Roel van Deventer, Robbert van Vliet

Department of Industrial Design
TU/e, March 2011

Sense your heart

Giorgio Campo s109714
Jelle Dekker s070085
Roel van Deventer s052145
Robbert van Vliet s070677

Department of Industrial Design
TU/e, March 2011

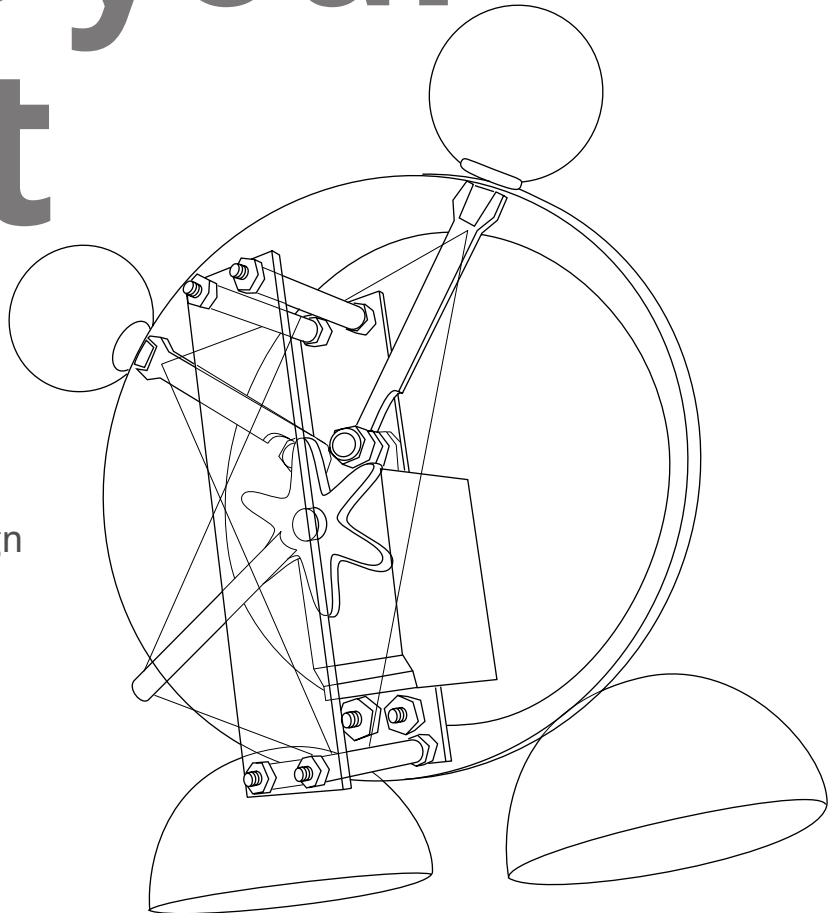


Table of Content

Introduction	...05
Theory	...06
Concept	...12
Prototype	...16
Discussion	...22
Sources	...23
Appendix	...24
- Processing code	
- Arduino code	

Introduction

The goal of this module was to get a good understanding of the workings of the human heart and how this working can be monitored. Secondly a learning objective was to implement these measurements in a concept using a self-built Photoplethysmography sensor and corresponding software. This technology can be used for measuring the heart rate, which in our case was used to determine the heart-rate variability which can be an indicator for relaxation or arousal.

Our concept makes use of the calculated heart rate variability to make an estimation of the arousal level of two related people. By making use of a robotic actuator and an internet connection two people can become aware of each other's arousal state, creating an incentive for a conversation. When person A is really relaxed or aroused in some way, this behaviour will be detected by the algorithm which will send this status to person B's laptop. At his location the robot (representing person A) will move according to the state of arousal, thus giving person B a subtle indication of how person A is feeling.

Theory

Physiology Cardio vascular system

The heart is a muscular organ found in all animals with a circulatory system. In the case of humans, the heart is a rhythmically contracting muscle weighing between 250 and 350 grams that forces blood through the vascular system providing the body with appropriate blood flow and pressure. The blood is used as a means of transport for nutrients, vital gasses, hormones and much more. Blood presence and regulation is absolutely vital for survival.

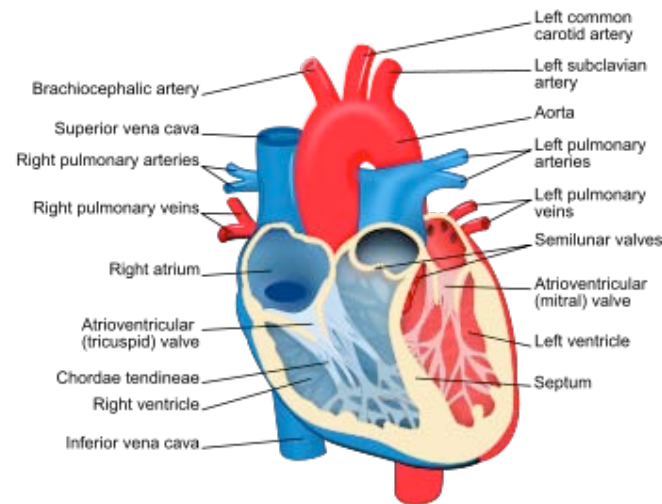


Figure 1: Cross section of the heart (Wikipedia.org)

The vascular system can be divided in two separate parts. The one that transports 'depleted blood' –oxygen poor and Carbon Dioxide rich blood – from the heart to the lungs and back. This is connected to the right side of the heart. The blood is received in the right atrium, which forces it into the right ventricle which forces it into the lungs. The other part of the vascular system transports freshly oxygenized blood from the Heart to the body. When the blood exits the lungs its received by the left atrium, which forces it into the left ventricle, which in its turn forces it through the aorta to the rest of the body.

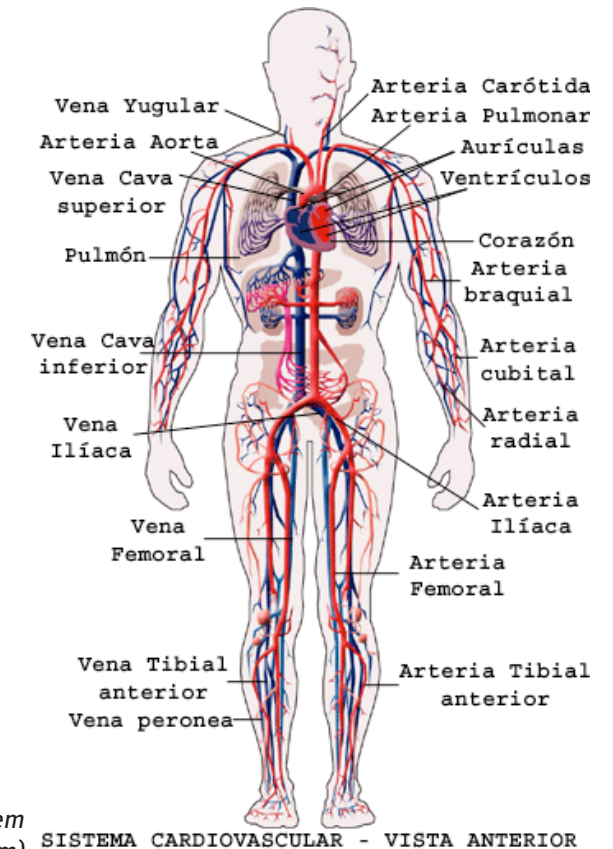


Figure 2: circulatory system (neivamigos.com)

SISTEMA CARDIOVASCULAR - VISTA ANTERIOR

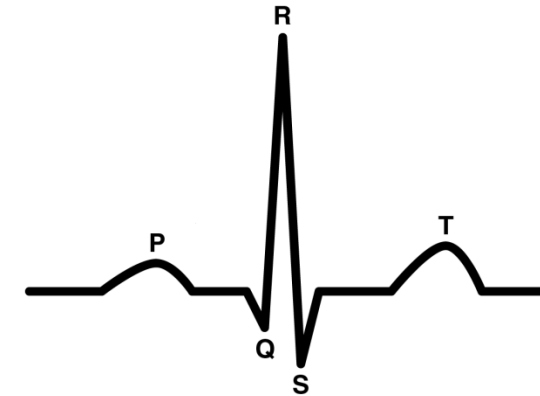
Heart rate and ‘Autonomous Nervous System’

The heart rhythm – or rate – is primarily dictated by the sinoatrial node (SA node)¹ which is a group of self-exciting cells that intrinsically fire a rhythmic pulse averaging 72 beats per minute. A system of nerve cells embedded in the heart passively regulates the shape and timing of the different contractions within the heart. Additionally, the autonomous nervous system (ANS) influences the heart-rate. The ANS – which subconsciously regulates organs and glands – can be divided into two main parts: The parasympathetic nervous system (PSNS) and the (Ortho)sympathetic nervous system (SNS) is mainly associated with fight-or-flight responses of the body and is mostly related to physical activity and stress. Very simply put the relation is reciprocal (Malliani, 1994)²; the SNS promotes the heart-rate whilst the PSNS inhibits it. They maintain a balance that shifts when the body for example digests or needs to be extremely alert. In case of the heart rate the balance is also heavily influenced by the breathing cycle. (Angelone et al. 1964)³

Electrocardiogram (ECG)

The action potential across the heart cells can be measured from outside the body. The changes in action potential as a function of time result in a graph called an ECG. Depending on the positioning of the electrodes (or; different leads) the ECG has a different shape. Despite standardization several methods are used, but they – as far as we know - are all based on Einthoven’s triangle. This means a lead is placed over the two arms (one electrode each) and the right leg. The variety within the standardization lies in the exact positioning of the electrodes, which can vary from the tip of the extremities to very close to the heart. The resulting graph is very well known as the line seen on the screen near hospital beds, and the infamous ‘flat-liner’ when the graph receives no input when a patient suffers a cardiac arrest. Specialists – typically cardiologists – can interpret these graphs as a diagnostic tool.

The ECG typically shows 5 peaks which are all produced by different events in the heart. The distance between the two main (R, See Figure 3) peaks in ECG is called the RR interval.



Agustine for Wikipedia

Figure 3: schematic representation of an ECG (single heartbeat)



Figure 4: PPG-sensor-clip attached on the hand.

Photoplethysmography (PPG)

Plethysmographs measure changes in volume (Langereis, 2010)⁴ By measuring changes in volume between an emitter and a sensor using IR-light the relative change in volume can be measured. In case of heart-rate monitoring this is done over a part of the body that is relatively thin and well blooded such as the finger (see Figure 4), earlobe or the lips. At each heartbeat the systolic pressure slightly expands the volume of the veins resulting in a (negative) peak in the Photoplethysmogram.

Heart Rate Variability

The heart rate variability (HRV) can be determined by the variation of the timespan between heartbeats in the time domain. (Malliani, 1994) This can be measured with the R-R interval or – in case of PPG – the time span between two peaks. In case of a properly working heart these intervals are very similar, albeit that the PPG peaks are slightly delayed (See Figure 5) since they are a result of the R peak in the ECG.

In normal conditions, an increased HRV is an indication of relaxed state. In a relaxed state both the sympathetic and the parasympathetic nervous system are in balance like a second order system resulting in an oscillation in heart rate frequency. When the body however is stressed the (Ortho)sympathetic nervous system allows for less variation in the frequency domain. Thus, the more even the heart-rate, the more tilted the human body is.

In medical research literature (Deboer et al., 1987)⁵ we found that to get a better picture of the HRV, scientists use spectral analyses of various data sources. Their article states variations in blood pressure are used next to the HRV calculated through R-R intervals. Their methods are mainly used for medical purposes that ask for detailed information on

the fluctuating values for example to detect certain arrhythmia.

For our purpose it wasn't important to get a very detailed view on the HRV as the goal was to keep measurements simple and non-obtrusive. By applying a self-adjusting filter in our algorithm we were able to get accurate enough results to find extreme values in the heart rate variability without analyzing the full spectrum. We only were interested in these extreme values and not especially in the spectrum of the variations, because we only defined three states: High arousal, relaxation and everything in between.

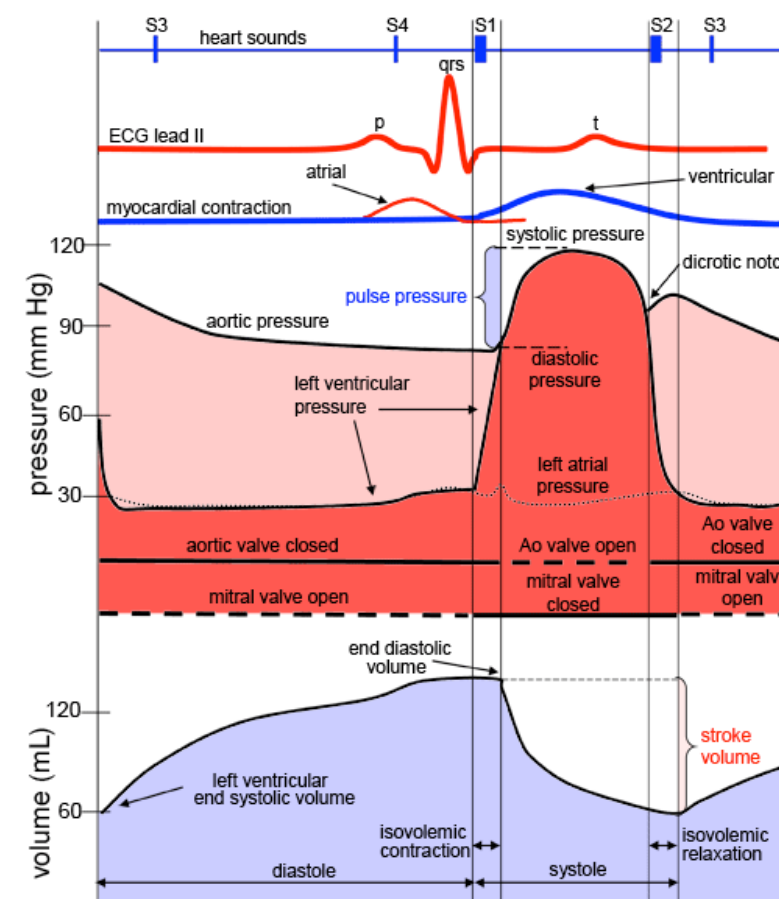


Figure 5: ECG and other heart-related measurements.

Concept

Thesis

As a matter of fact the world is getting smaller and smaller and nowadays we could fairly easy reach every country we want. In the next future possibly we'll be spread out in the world and this might also happen to your girlfriend, to your son, or to your best friends. Fortunately for many we are always connected through social networks and chat systems. By the way within this increasing trend, technology seems to be not enough personal to communicate strong emotions, sometimes it's dehumanizing and cold.

Is there a deeper way to be connected with people you love the most? A way which refers to a more personal experience? A way of communicating that starts before typing on your laptop keyboard?

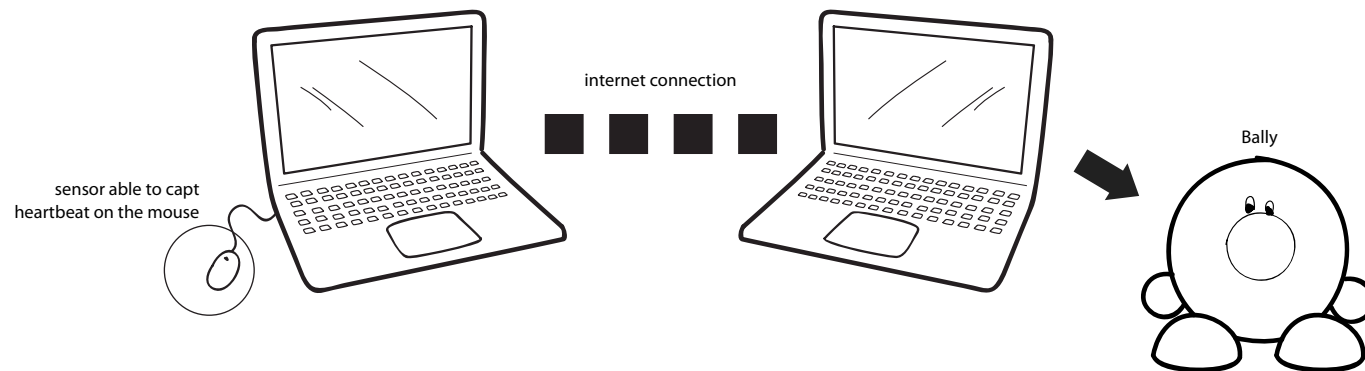


Figure 6: Concept Overview

Concept

In the module Sense your heart we learnt how to measure and keep track of the heartbeat. The heartbeat it something very personal and it could be interesting to use it and share it to keep alive a strong link between people in a very personal way.

Bally, sharing emotions

Bally is a system of a mouse and a character which are connected through the web. On the mouse is installed a sensor able to detect the heartbeat from user's fingers while the character moves her hands in different ways (see [figure 7](#)) according to the heartbeat.

The idea is that you give the mouse to a person you care about and you keep the character for yourself (or vice versa).

Bally's language

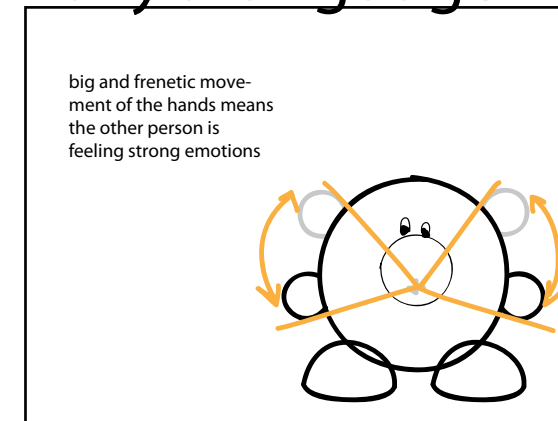
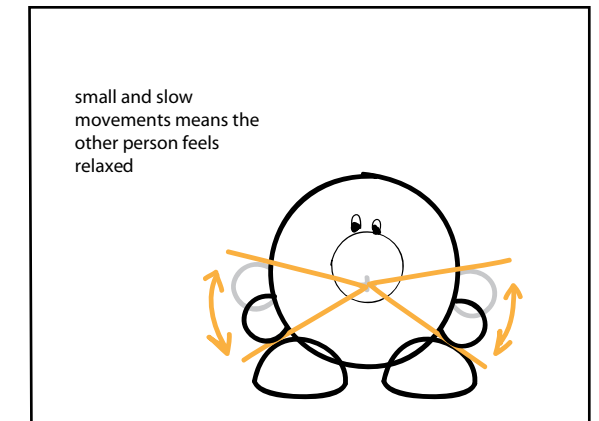


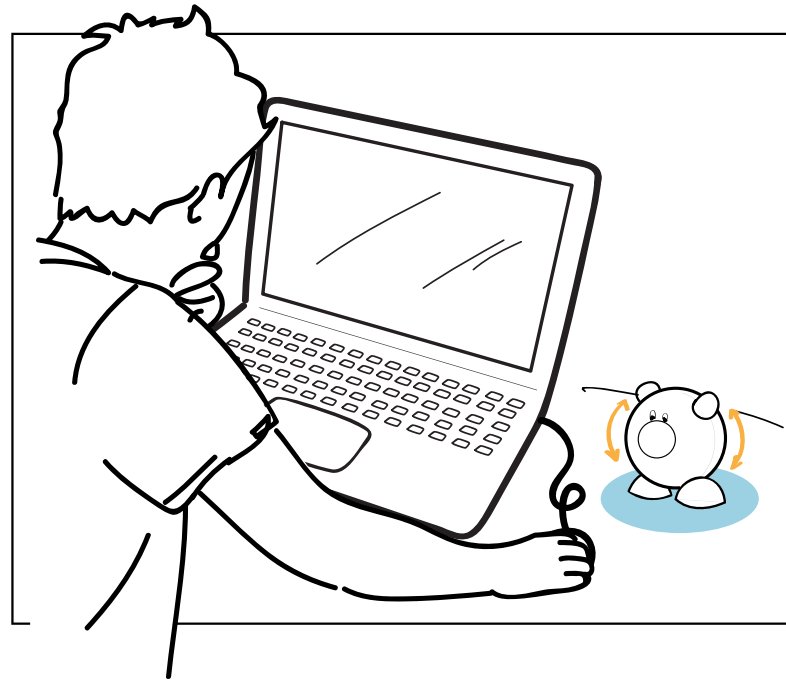
Figure 7: Bally's movement language



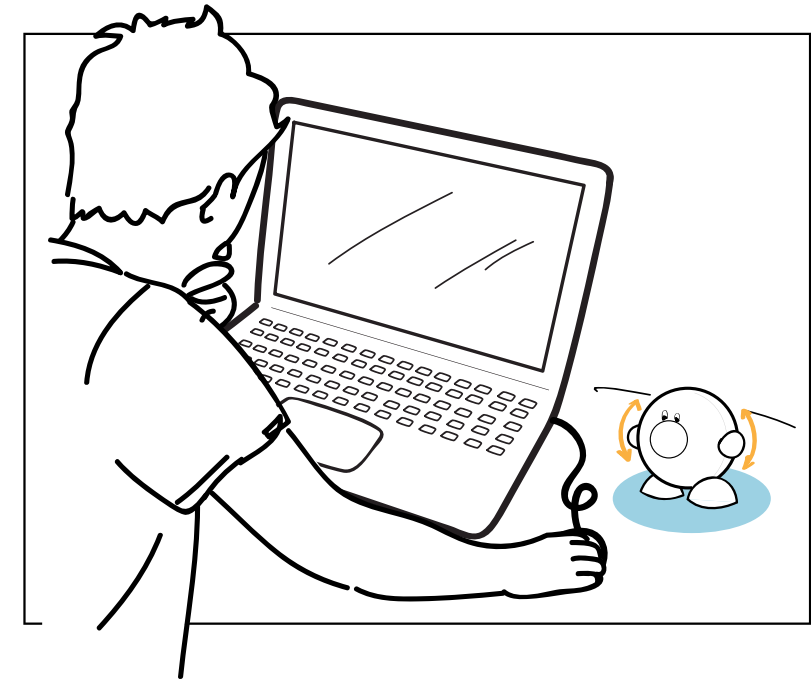
Storyboard



ooo, another boring
report to write...but I
don't wanna work



Oh! my friend seems to
be excited about some-
thing...



...let's check what's
happening there!

Prototype

Overview

The prototype connects two clients to communicate a representation of stress to each other (see [Figure 8](#)). To do this we use two computers and four Arduinos (two for sensors, two for actuators), the extra Arduinos were needed because of difficulties in communicating back to the Arduinos on the serial port. In the final demonstration one actuator was used rather than two.

The Processing sketch gets it's data from the (sensor) Arduino, which communicates the time between heartbeats (RR interval) through the serial port. Here the calculations for HRV (RRstd) are calculated as provided in the sketch for the module. This HRV is then mapped to determine a stress level, to do this we use the highest and lowest HRV of the session. These High/Low are becoming more extreme when the HRV goes beyond them, and slowly go closer to the middle when the HRV falls within them.

This is probably not an ideal way to determine stress/excitement levels. But it showed a clear difference when you were calm or active, and wasn't too irregular, the mapping made it more gradual. This meant it worked for the demonstration purposes.

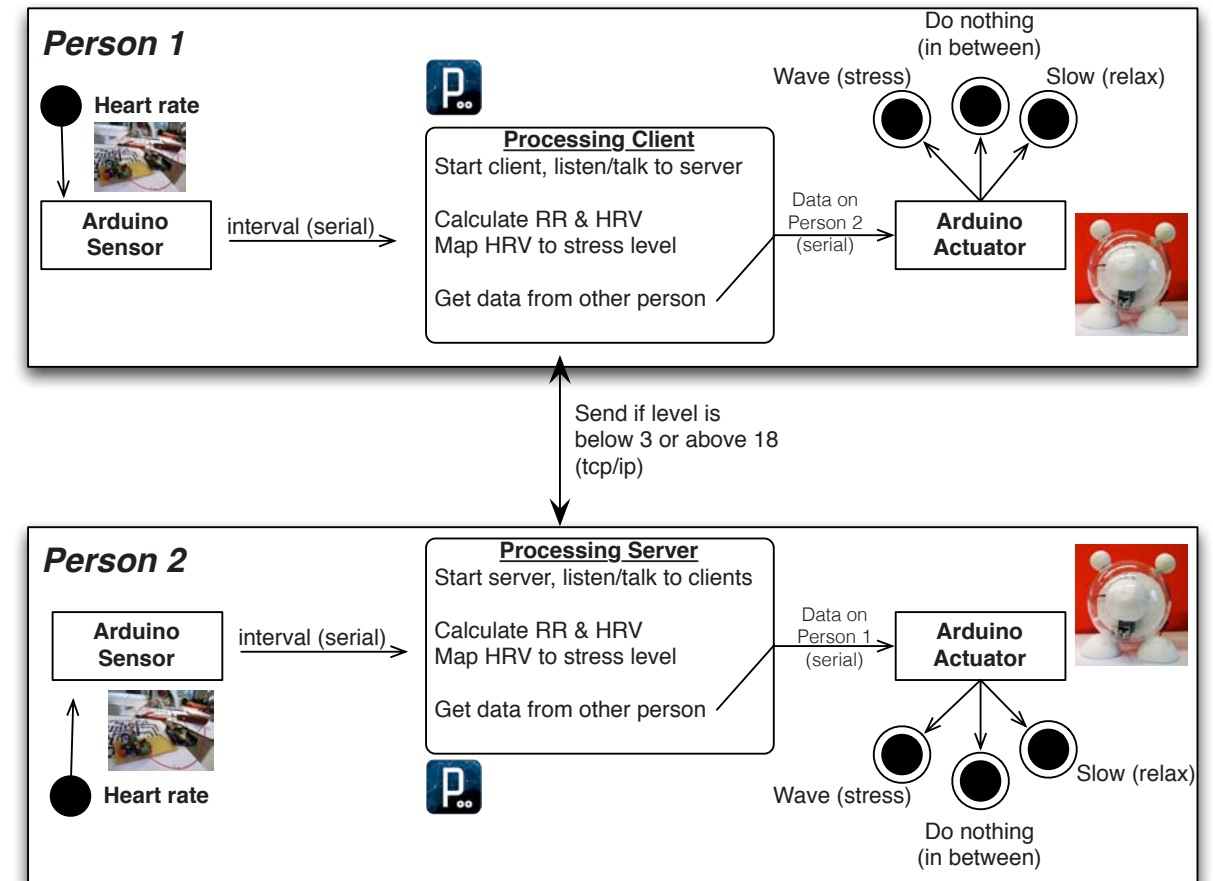


Figure 8: System communication overview

Sensor

The heart-beat sensor uses a PPG ear-lobe clip for it's measurements. This sensor can measure the difference in light passed through when the blood-vessels expand as a result of a heart beat. This signal is filtered and amplified to get a usable digital and analog output of the heart beats.

The sensor was built according to the schematic as provided (see [Figure 9 & 10](#)). Some small adjustments (other resistors) were made for a version of the sensor that could be implemented in i.e. a mouse, rather than using the ear-lobe clip. During testing we found that the ear-lobe clip could be used in more places, for example the skin between your fingers works very well.

The digital output is used to connect to the Arduino, using an interrupt port the Arduino can measure the time between heart-beats. This data is communicated to Processing through a Serial connection.

In Processing we use the method provided in the module (see [Figure 11](#)) to calculate the HRV and map this to max/min to get a stable level to be used as “relax/stress” indicator.

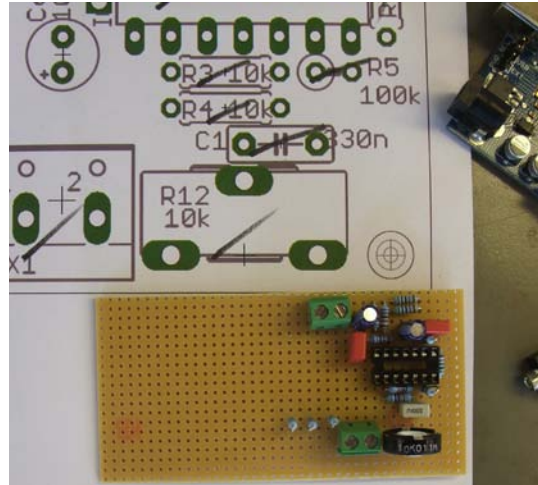


Figure 9: sensor circuit construction

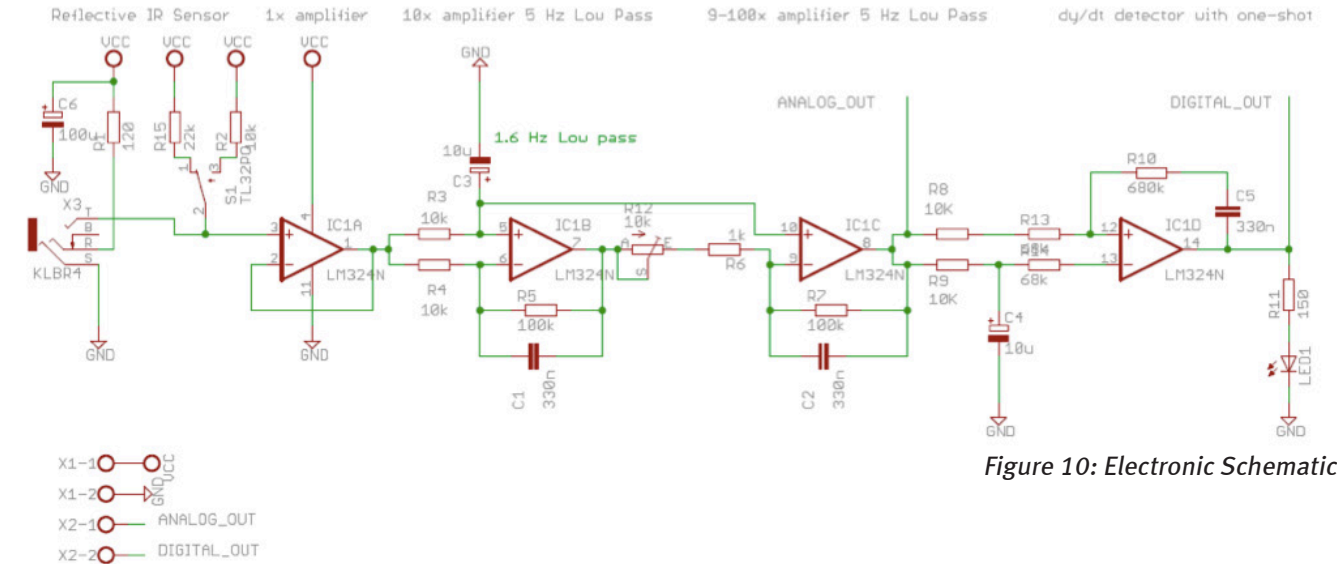
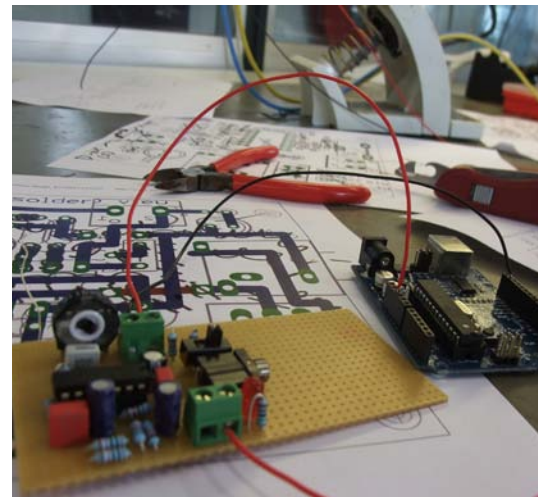


Figure 10: Electronic Schematic

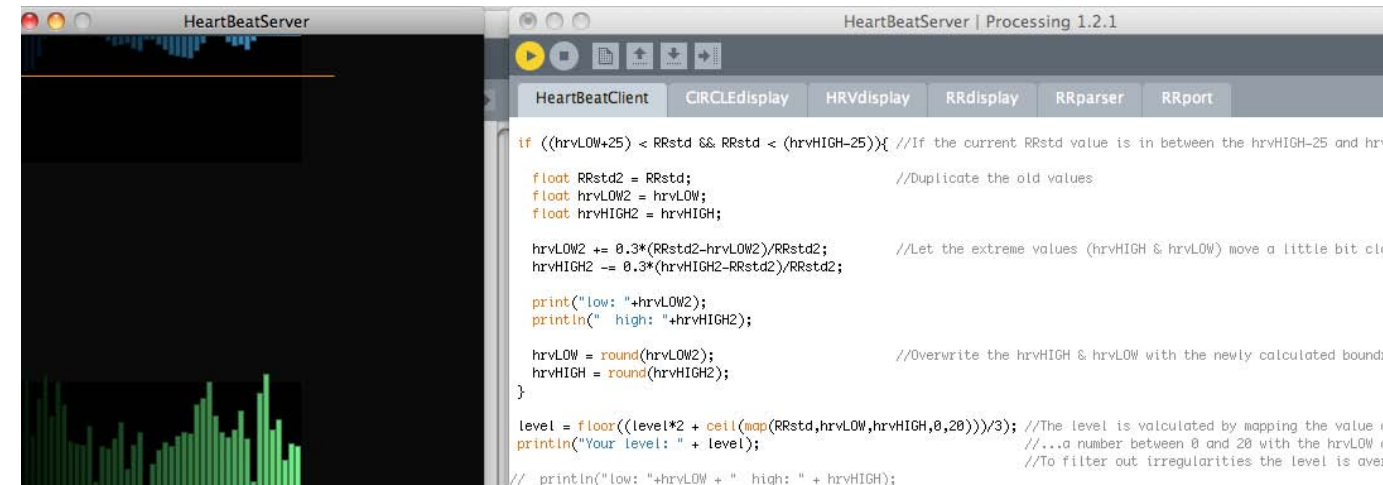


Figure 11: Processing code with HRV graph (data used) seen in the top-left

Actuator

text The Processing sketches would run on two computers, one acting as a server, the other as a client. The two measure the heartbeat of the person using it, calculate the “stress level” and when this was below/above a threshold they would send this to the other. The other sketch responds on these “1” and “0” signals by moving the arms of the puppet in a relaxed, or very excited manner.

A second Arduino served as actuator, controlling the puppet. The signal (0/1) from the other person is passed on and the Arduino translated this into the right movement for the servo controlling the puppets arms. Control for the puppet is very simple in this case, the 5v, GND and signal from the servo motor are directly connected to the Arduino. This can be improved by connecting a separate power supply to the servo.

The actuator resembles an abstract puppet, clean but able to move it’s arms in a way that can be interpreted as an “emotion” from the other person. Inside is a servo motor controlling the arms, through a magnetic connection the arms inside can move the hands on the outside of the puppet. Because of this connection the hands could be easily exchanged to personalise the puppet.

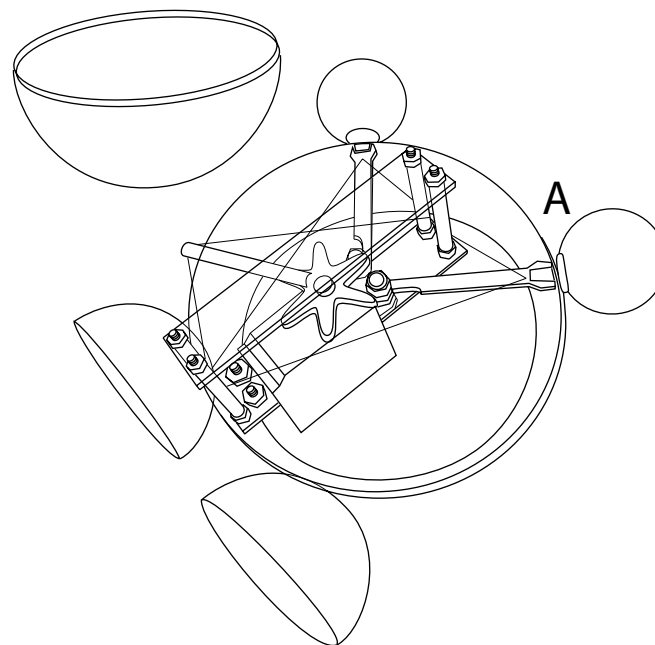
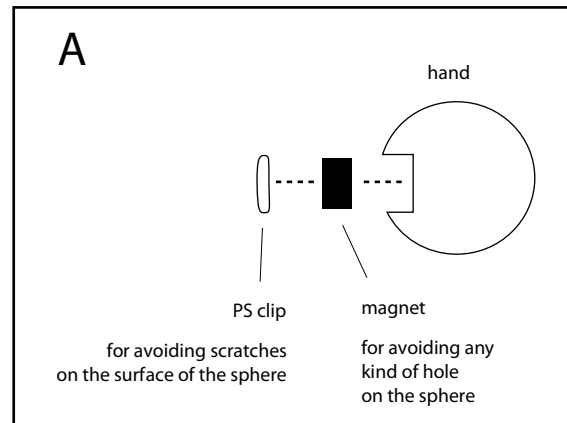


Figure 12: Bally's internals

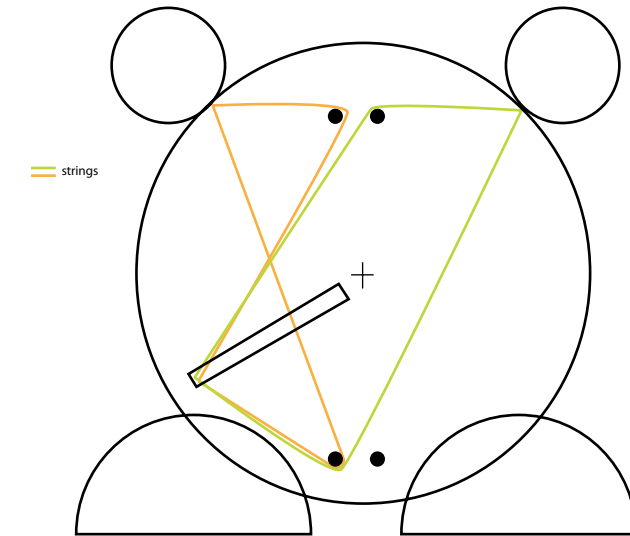


Figure 13: Bally in construction



Discussion

With Bally we looked at how we could use the heart-rate (-sensor) in a new way, exploring the possibility of sharing something more personal (and less 'digital') over a distance. But there's a lot more to explore on Bally.

This time we tried to measure a level of stress/relaxation, map this and communicate it with Bally. However this probably wasn't the ideal way of measuring stress, different sensors/techniques could be used. Also Bally might pick up different signals to communicate.

From a practical point of view it is a problem now that Bally struggles for power with the Arduino, it needs a power supply. Also the mechanism can self-destruct because of the long arms, the movement technique needs to be refined more.

We used magnets to attach the hands making the (dis)connection an easy process. This could allow to make Bally more customizable, people could expand the possibilities by adding different hands to the connection.

Furthermore there is an ethical issue. With recording and sharing this kind of private information it's important to make sure the privacy is respected and avoid making a 'big brother technology' (Daniel Berdichevsky et al, Analyzing The Ethics Of Persuasive Technology, 2005)⁶

Sources

1 <http://www.vhlab.umn.edu/atlas/phystutorial/index.shtml> (checked: 22-03-2011)

2 Malliani, Lombardi, Pagani – Power spectrum analysis of heart rate variability: a tool to explore neural regulatory mechanism – British Heart Journal 71:1-2, 1994

3 Angelone, Coulter, Respiratory sinus arrhythmia: a frequency dependent phenomenon – J Appl Physiol. 19:479-82, 1964

4 Presentation; 'Photoplethysmography (PPG) system' by Geert Langereis, Jan 2010

5 deBoer, R.W., Karemaker, J.M., Strackee, J. – Hemodynamic fluctuations and baroreflex sensitivity in human: a beat to beat model – Department of physiology and laboratory of medical physics, University of Amsterdam, 1987

6 Daniel Berdichevsky, B.J. Fogg, Ramit Sethi, Manu Kumar, Analyzing The Ethics Of Persuasive Technology / 2005, Stanford Persuasive Technology Lab

Appendix

Processing code:

Included is the code from the two processing sketches that were adjusted. Other sketches that were provided in the module have been used as-is without adjustments and are not in this text, they can be found in the package handed in.

HeartBeatServer.pde

```
import processing.serial.*; //import libraries for communicating over the serial ports
import processing.net.*; //import libraries for communicating over the tcp/ip protocol
```

```
boolean RRDISPLAY=true;
boolean HRVDISPLAY=true;
boolean CIRCLEDISPLAY=false;
```

```
RRport myport;
RRparser myparser;
```

```
Server myServer; //This is the server side of the program
Client myClient; //We also need a client for receiving data
```

```
RRdisplay rrdisplay;
HRVdisplay hrvdisplay;
CIRCLEdisplay circledisplay;
```

```
PrintWriter output;
boolean settedup=false;
```

```
void setup()
{
  int bars=75;
  frameRate(10);
  size(5*bars,5*bars);
```

```
myServer = new Server(this,5204); //Setup a server at this ip address on port 5204
```

```
//myClient = new Client(this,"131.155.171.145",5204); //Setup a client to a certain ip address on port 5204
```

```
myport = new RRport(this); //We also changed the myport class for configuring 2 arduino boards
myparser = new RRparser();
rrdisplay = new RRdisplay();
hrvdisplay = new HRVdisplay();
circledisplay = new CIRCLEdisplay();
```

```
SimpleDateFormat sdf = new SimpleDateFormat("-yyyy-MM-dd/HH-mm");
Date now = new Date();
println("log to logfiles"+sdf.format(now));
output = createWriter("logfiles"+sdf.format(now)+".txt");
```

```
fill(10,10,10);
rect(0,0,width,height);
settedup=true;
}
```

```
int RRavg = 1200;
int RRstd = 50;
```

```
int dataIn; //Variable for the incoming data from the server
```

```
int hrvLOW = 1500; //Lower hvr domain boundry
int hrvHIGH = 0; //Upped hvr domain boundry
```

```
int level = 0; //Level of arousal according to the algorithm
```

```
int time = 0;
int count =0;
```

```
void draw() {
  count++;
  count=count%10;
  if (count==9) output.flush();
  time++;
```

```
// if (myClient.available() != null) {
//   dataIn = myClient.read();
//   myport.myPort2.write(dataIn);
//   println("incoming message: " + dataIn);
// }
```

```
myClient = myServer.available(); //Create a client
if (myClient != null) { //If there is a message pending from the server
  dataIn = myClient.read(); //Read it..
  myport.myPort2.write(dataIn); //..and send it to the actuator's port
  println("incoming message: " + dataIn);
}
```

```
}
```

```

void serialEvent(Serial p) {
  if (settedup) {
    //print("!");
    //zat allemaal eerst in draw
    myport.step();
    myparser.step();
    //if (time/10 > 120) stop();
    if (myparser.event()) {
      int RR = myparser.val;
      output.println(RR);
      RRavg = int((23 * RRavg + 1*RR)/24);
      RRstd = int((15.0 * RRstd + abs(RR - RRavg)) / 16.0);

      checkLevel(); //Function containing the algorithm

      rrdisplay.next(RR);
      hrdisplay.next(RRstd);
      circledisplay.next(RRavg,RR);
      if (RRDISPLAY==true) rrdisplay.step();
      if (HRVDISPLAY==true) hrdisplay.step();
      if (CIRCLEDISPLAY==true) circledisplay.step();

      print("Heart Rate Beat per Beat= " + (60000/RR));
      println("Heart Rate running average = " + (60000/RRavg));
    }
  }
}

public void stop() {
  output.flush(); // Writes the remaining data to the file
  output.close(); // Finishes the file
  println("THANK YOU, GOODBYE");
  exit(); // Stops the program
}

void keyPressed() {
}

void checkLevel(){ //Calculates the arousal level
  if(RRstd > hrvHIGH){ //If the current RRstd (hrv) measurement exceeds the highest known value
    hrvHIGH = RRstd; //Make the highest found value the maximum RRstd
  }
  if (RRstd < hrvLOW){ //If the current RRstd (hrv) measurement is lower than the lowest know value
    hrvLOW = RRstd; //Make the lowest found value the minimum RRstd
  }

  if ((hrvLOW+25) < RRstd && RRstd < (hrvHIGH-25)){ //If the current RRstd value is in between the hrvHIGH-25 and hrvLOW+25

    float RRstd2 = RRstd; //Duplicate the old values

```

```

float hrvLOW2 = hrvLOW;
float hrvHIGH2 = hrvHIGH;

hrvLOW2 += 0.3*(RRstd2-hrvLOW2)/RRstd2; //Let the extreme values (hrvHIGH & hrvLOW) move a little bit closer to each other
hrvHIGH2 -= 0.3*(hrvHIGH2-RRstd2)/RRstd2;

print("low: "+hrvLOW2);
println(" high: "+hrvHIGH2);

hrvLOW = round(hrvLOW2); //Overwrite the hrvHIGH & hrvLOW with the newly calculated boundries
hrvHIGH = round(hrvHIGH2);
}

level = floor((level*2 + ceil(map(RRstd,hrvLOW,hrvHIGH,0,20)))/3); //The level is valculated by mapping the value of the currently
//measured RRstd to... //...a number between 0 and 20 with the hrvLOW and hrvHIGH as starting reference.
println("Your level: " + level); //To filter out irregularities the level is averaged.
// println("low: "+hrvLOW + " high: " + hrvHIGH);

if(level >= 18){ //If the level exceeds 18 the hrv is high and arousal low
// myClient.write(1);
myServer.write(1); //Send a 1 (meaning high hrv) to the client
println("sending: 1");
}
else if(level <= 3){ //If the level doesn't exceed three the hrv is low and arousal high
// myClient.write(0);
myServer.write(0); //Send a 0 (meaning low hrv) to the client
println("sending: 0");
}
}

void mouseClicked() {
// hrvLOW = 1500;
// hrvHIGH = 0;
// myport.myPort2.write(1);
// println("actuator test");
}

```

RRport.pde

```
class RRport {
  private Serial myPort, myPort2; //We need two ports, one for the PPGs Arduino and one for the actuator robot
  private int[] buffer = new int[100];
  private int bufcnt = 0;
  RRport(PApplet parent) {

    //Check if COM-ports are correct!
    String portName = Serial.list()[0];
    String actuatorPortName = Serial.list()[2]; //Also make a port for the actuator

    println(Serial.list());
    myPort = new Serial(parent, portName, 19200);
    myPort2 = new Serial(parent, actuatorPortName, 19200); //Setup the port
    while (myPort.available() == 0) {
    }
    if (myPort.read() == '>') {
      myPort.write('I');
      println(">I");
    }
    else println("ARDUINO NON > PROMPT ERROR");
  }
  void putbuf(int c) {
    if(bufcnt >= 100-1) println("OVERFLOW buffer ERROR");
    buffer[bufcnt++] = c;
  }
  int getbuf() {
    if (bufcnt <= 0) println("UNDERFLOW buffer ERROR");
    int v = buffer[0];
    //works like FIFO, shift rest again
    int i = 0;
    while (i < bufcnt-1) {
      buffer[i] = buffer[i+1];
      i++;
    } //alternative: cyclic buffer, not now
    bufcnt--;
    return v;
  }
  void step() {
    while (myPort.available() > 0) {
      int inByte = int(myPort.read());
      putbuf(inByte);
    }
  }
} // RRport
```

Arduino code:

Included is the actuator code used for the Arduino. The sensor Arduino was used without changes to the provided code from the module.

<pre>#include <Servo.h> // for the servo int incomingByte = -1; // for incoming serial data int ledPin = 13; Servo myservo; //our servo int pos = 0; void setup() { Serial.begin(19200); // opens serial port, sets data rate to 19200 bps myservo.attach(9); // attaches the servo control to output pin 9 myservo.write(120); // turn to 120 deg } void loop() { // send data only when you receive data: if (Serial.available() > 0) { // read the incoming byte: incomingByte = Serial.read(); if(incomingByte == 0){ // if the hrv is low there could be stress, so the robot moves in a disstressed way for(int i=0;i<=3;i++){ for(int r=0;r<=20;r++){ myservo.write(120+(2*r)); delay(10+(r-10)^2); } delay(50); for(int r=0;r<=20;r++){ myservo.write(160-(2*r)); delay(10+(r-10)^2); } delay(50); } } else if(incomingByte == 1){ // if the hrv is high the other person is</pre>	<pre>probably relaxed, so the robot moves in a relaxed way for(int i=0;i<=3;i++){ for(int r=0;r<=20;r++){ myservo.write(120-(2*r)); delay(20+(r-10)^2); } delay(50); for(int r=0;r<=20;r++){ myservo.write(80+(2*r)); delay(20+(r-10)^2); } delay(50); } } incomingByte = -1; Serial.flush(); } }</pre>
---	--

