2005:283 CIV

MASTER'S THESIS

BodyResT

A prototype using music responding to heart rate for stress reduction

Johan Fagerlönn

Luleå University of Technology

MSc Programmes in Engineering

Department of Computer Science and Electrical Engineering
Division of Media Technology

2005:283 CIV - ISSN: 1402-1617 - ISRN: LTU-EX--05/283--SE

ABSTRACT

The growing problem of stress in our society has provided greater motivation to seek stress solutions. This health threat demands a fresh thinking and exploration of new ways to find solutions. Music has in several studies been suggested to have a stress reducing effect. This report describes the development of a prototype that uses music to step by step help an individual to reduce stress. This is done by a combination of relaxing music and the biofeedback principle. The prototype estimates the individuals stress level by monitoring heart rate. On the basis of senor readings music is composed in real time to match the stress level. In that way a feedback loop is created where the individuals current stress level is reflected in the music. The music does not change from one song to the other; rather different parameters in the music are continuously changed such as tempo and instrumentation.

The report also includes a literature study examining heart rate as a valid stress indicator. This literature study is also complemented with an experimental study. In this study 14 participants, 7 males and 7 females, age between 21 and 54, were exposed to three different mental stressors separated with three minutes of relaxation. A version of Stroop color word test, a mental arithmetic task and a talk preparation. Heart rate was measured and the subjective estimation of stress level was given by the subjects before the first stressor, after each stressor and after each period of relaxation. A significant increase of heart rate (~10 bpm in mean) was detected in the end of each stressor compared to the periods of relaxation.

The important conclusion with the studies is that heart rate is a valid mental stress indicator, but the reliability is low since heart rate is influenced by many other factors than mental stress. In future development of BodyResT heart rate must be complemented with one or several other physiological parameters such as heart rate variability and skin conductance.

ACKNOWLEDGEMENTS

This paper presented the primary results from my Master's Thesis work carried out at the Interactive Institute's Sonic Studio during the spring 2005. It marks the closure of my Master of Science Program in Media Technology, at Luleå University of Technology.

The development of the BodyResT prototype has been a part of the Biofeedback Music project at the Sonic studio. The work of creating the prototype has involved several people at the studio.

- Music composition – Stefan Lindberg
- Music engine – Mats Liljedahl
- Open Sound Control implementation – Johan Fagerlönn (in Java), Mats Liljedahl.
- Sensor connection – Johan Fagerlönn
- User interface – Ingemar Almeros (design), Johan Fagerlönn (implementation in Java).
- Regulator development – Johan Fagerlönn, Mats Liljedahl.

I would like to thank my supervisor Kåre Synnes at Luleå University of Technology, and Mats Liljedahl who has been a co-worker and my supervisor at the Sonic Studio. I also want to thank a couple of people that has been especially helpful to me during my work. My girlfriend Lena who has supported me in my work during the whole spring, Nyssim Lefford for helping me with the English language during report writing, and Kennet Hartvik at EISlab for supervision on the MULLE platform. Finally, I would like to express my gratitude to Katarina Delsing, director at the Sonic Studio, for giving me the chance to be part of the Biofeedback project

Johan Fagerlönn

TABLE OF CONTENTS

1 Introduction

1.1 Project Background

Biological change or adaptation is something that takes exceedingly long time. From a biological point of view the human being is still a hunter, fisher and gatherer. During the last several thousands of years there have been many changes in the environment in which we live. The modern human being is now exposed to stresses never experienced as a hunter/gatherer. These stresses have been particularly noticeable during the last hundreds of years and especially during the last few years with the entrance into the information oriented era.

With globalization and modern technology our life's characteristics have dramatically changed. The strains have turned into a new kind and health risks have adopted new shapes.

Information technology makes increasing demands on individuals' availability and flexibility. This is one factor among others that increase the risk of negative stress. People frequently feel a pressure to produce more and more and feel a constant lack of time. Spare time is seen as a waste. Another factor is that in many professions, people do not have control over the working tempo or how work is performed [1]. This is particularly notable in professions demanding higher and more theoretically based educations.

In Sweden the number of people reporting in sick for work has increased steadily during the last 20 years. People take sick leave more often and are absent for longer periods. Stress related syndromes are a significant contributor to this increase. Diagnoses of depression, anxiety and burnout have also increased. At the same time, non-stress related causes of sick leave, like physical injuries, have remained constant or decreased.

It is the less serious psychologically related illnesses such as stress related symptoms and neurosis conditions that have caused an increase in sick leave. More serious illnesses, like psychosis, have remained a constant factor [2].

The increased incidents of sick leave results in a tremendous cost for the society. In 2000, in Sweden, the total cost was 74.5 billion sek. In 2001, that number was over 108 billions. Another alarming statistic is that the age group primarily affected by the rise in psychologically-related illnesses is young people that still have many working years before retirement [1,2].

The growing problem of stress in our society has provided greater motivation to seek stress solutions. This growing health threat demands a fresh thinking and exploration of new ways to find solutions.

Stress management is a catch phrase for many different strategies available for dealing with stress. One essential idea in stress management is to encourage the individual to find a balance between stress related actions and other actions in life.

Suggested lifestyle changes often include a healthier diet, more exercise and better planning. Another part of stress management tries to create increase moments of relaxation in peoples' daily lives. Listening to relaxing music is an established and well-known method to reduce stress.

1.1.1 Music and relaxation

Studies have shown that music can have a stress reducing effect [3]. In recent years, many reports in music, psychology and medicine have cited anxiolytic effects in music. These effects have been examined for different music types. Perceived relaxation elicited by sedative music, which is characterized as melodious, delicate, harmonic and romantic [4,5]. On the other hand, stimulative and excitative music, characterized as loud, dynamic and rhythmic, elicited tension and excitement [4].

Several previous studies have used classical music for developing relaxation and stress-reduction techniques [6,7]. It is assumed that this kind of music has a relaxing effect because of its harmonic consonance and low tempo [8].

Relaxing music can generate psychological and physiological changes linked to a reduction in activity in the autonomic nervous system, and this can lead to a physiological stress reduction [6]. More specifically, the physiological reaction to stress activates the sympathetic nervous system (an autonomic reaction), and this results in an increase in heart rate, respiration rate, muscle tension and blood pressure [9]. According to numerous studies, music can reduce the activation of the sympathetic nervous system and thus is capable of inducing physical relaxation [7, 10].

Patrik N. Juslin and Petri Lauka at the department of psychology, Uppsala university, Uppsala, Sweden performed a questionnaire study of everyday listening. 141 people participated in the study. On the question "Why do you listen to music?" the participants were required to talk free about the strongest motives. 47% mentioned "to express, release and influence emotions". 33 % mentioned "to relax and settle down". This can be compared to "because I like/love music" or "to evoke memories" which 12% respective 4% of the participants mentioned as motives. The questionnaire also included an item that asked subjects "if you perceive that the music expresses a certain emotion, do you also feel that emotion?" The results suggests that this may happen quite often (Always 6%, often 65%, seldom 29%, never 1%) [11].

1.1.2 The Biofeedback Music Project

Interactive Institute is an experimental research institute which focuses on interactivity between humans and modern technology. The institute has eleven studios which emphasis different areas of research scattered in different cities across Sweden. The Sonic studio in Piteå focuses primarily on interaction using music and sound.

One of the projects at the Sonic studio is Biofeedback Music. This project explores music as a tool to create relaxation, reduce stress and thereby improve the individual's well being. The project involves researches with competences in behavior, computer, music composition and technology. One of the project's goals is to create a portable device with biosensors that can measure an individual's stress level and compose customized, relaxing music that adapts to the users physiological readings. The device's is currently being designed, but variability in individuals' emotional state, personality and taste in music are included as features of the device's functionality. To help the listener relax, the product will use a combination of relaxing, synthetic music generated in real-time and the biofeedback principle.

The project hopes to answer several research questions in various disciplines. Which physiological parameters are suitable to measure when stress level is to be estimated? Which parameters in music are relaxing? How can personal preferences be included? The biofeedback project draws on different areas of expertise to find answers to these questions.

1.1.3 Problem discussion

The biofeedback music project has just completed a first prototype. This prototype will not only take the biofeedback project one step closer to a final product, but it will also be useful as an experimental tool in continued studies.

The prototype's design is much simpler than the intended final product. In the prototype, an individual's heart rate is used to monitor stress level. The heart rate is a physiological parameter assumed to have some robust stress indicating characteristics. Heart rate is also easy and fast to measure making it a good signal to measure in this implementation where the level of stress is constantly monitored.

A software application has been developed that can receive the data from the sensor. This application will also control a sound engine that generates music.

The generated music changes according to changes in the heart rate. In that way, a feedback loop is created and changes in the heart rate are reflected in the music. The generated music does not just change from song to song, but rather changes constantly by altering different variables in the music in time with physiological shifts. Musical variables include features like tempo, instrumentation and rhythmic complexity.

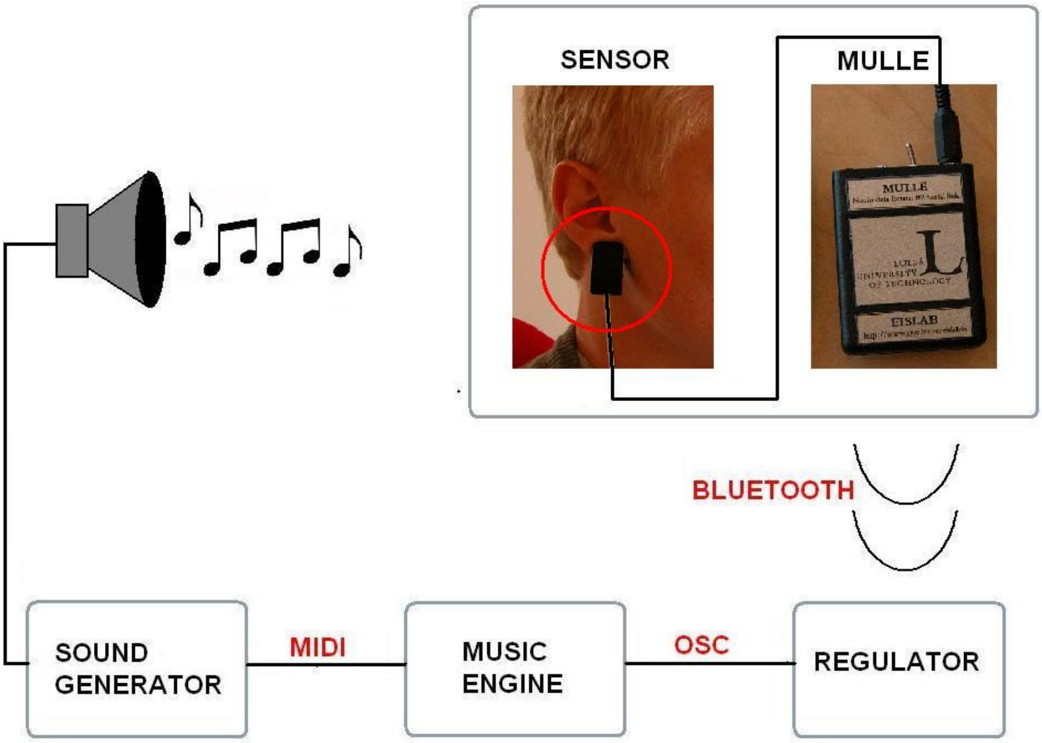


Figure 1: Conceptual outline of the BodyResT prototype.

1.1.4 Technical challenges

The following demands on technical functionality were stated before building the prototype started:

- The prototypes heart rate sensor had to meet several design criteria. It must be comfortable and light-weight. The sensor output data format should be easy to translate, and provide a suitable input for other programs. This means that the sensor should send the data in a sophisticated data format. The sensor should also be inexpensive thus making it a feasible choice for commercial applications. To avoid disturbing and unnecessary cables and to increase the mobility the connection between senor and computer must be wireless.
- An experimental regulator must be implemented in the application that decides how and when to change the music. The system must be able to change one or more musical parameters.
- A way for the regulator to communicate with the music generating system must to be implemented.
- A graphical user interface must be designed that can be used to control the prototype. This GUI must include functionality to run tests on subjects. There must also be functionality to save data recorded during the test along with information about the subject, music used during the test and time, date and length of the test. There must also be functionality to simulate incoming sensor

-readings. The GUI must also be able to display pulse rate over time in some way.

- In order to make it possible to draw conclusions from future tests using the BodyResT prototype, it must be possible to only change one or a combination of musical parameters at the time.

1.1.5 Purpose and Goal

The primary purpose of this report is to describe the creation of the first prototype. This prototype will form a solid basis for the continuous work with the biofeedback project. Secondly, the report formalizes a pre-study evaluation validating the use of heart rate and its variations as psychological parameters for determining stress level. The result of this pre-study is knowledge that can be useful in the continuous development of the prototype, and beyond this work.

The first goal for this stage of development is to create an operational prototype. This prototype should be able change one or more parameters in the music according to the heart rate. The second goal is to reach a conclusion about the validity of using heart rate and its variability as a stress level indicator.

1.1.6 Limitations

The Biofeedback Music project is expected to last a couple of years. However, the final thesis is limited to about four months. Therefore it is important to define expectations and to make sure the stated goals of the thesis are reachable in that time frame.

There are several psychological parameters that are measurable with existing sensor technology and assumed to be indicators of stress level. In the pre-study only the heart rate and heart rate variability will be examined. The time frame does not allow me to examine more than two parameters.

Prototype development has three phases. Phase one examines which sensor to use for measuring pulse and to read, synchronize and manipulate the data from the sensor. Phase two creates a regulating system to control different parameters in the music. Phase three is about creating a music engine that plays the music. The thesis research is limited to working with the sensor connection, and also together with software developer, Mats Liljedahl, to construct the regulating system. The graphical user interface for the prototype software is also part of the thesis work.

When creating a typical regulating system, one begins by examining the process that is to be regulated and then developing the regulating system. In this case, the system to be regulated is a human being which makes examination very complicated. Parallel research studies in the biofeedback project examine how different parameters in the music affect the level of stress. This study will provide the knowledge base about musical parameters and relaxation and the operational guidelines for system controls over the different parameters. The regulating system in this first prototype will therefore be relatively simple and based on design assumptions in combination with musical experiences. But the system architecture will offer a high degree of flexibility so that future changes will be easy to make.

The two goals are tightly interconnected. Both the prototype and the heart rate study will provide a basis for the continued work in the Biofeedback Music project. However, if the conclusions from the pre-study about heart rate as a stress indicating parameter do not correlate with prototype testing on real subjects, the heart rate parameter will not be omitted in futures work. But that decision is beyond the timeframe of this thesis.

2 Theoretical background

The theoretical background is divided into two chapters. The first chapter forms a literature study examining heart rate and stress. The second chapter introduces fundamental technology used in the BodyResT prototype.

2.1 Heart rate as a stress indicator

2.1.1 The Autonomic nerve system

The autonomic nerve system (ANS) is the part of the nervous system that is not under conscious control and that regulates the internal organs. It includes the sympathetic, parasympathetic, and enteric nervous systems. The first, which connects the internal organs to the brain via spinal nerves, responds to stress by increasing heart rate and blood flow to the muscles and decreasing blood flow to the skin. The second comprises the cranial nerves and the lower spinal nerves, which increase digestive secretions and slow the heart rate [12].

The ANS system is also referred to as the independent nerve system. But this definition is a bit out of date, since it has been shown that the ANS functions can be influenced by learning, although indirectly [1].

2.1.2 Heart rate

The heart contains cardiac pacemakers that spontaneously cause the heart to beat. These are controlled by the autonomic nervous system and circulating adrenaline. The heart gets its parasympathetic innervation from the vagus nerve. Sympathetic stimulation comes from the cardiac nerves.

A specialised portion of the heart, called the sinoatrial node (SA node), is responsible for the whole heart's beat. The SA node is a group of cells positioned on the wall of the right atrium, near the entrance of the superior vena cava. Cells in the SA node will naturally discharge (create action potentials) at about 70-80 times/minute. Because the sinoatrial node is responsible for the rest of the heart's electrical activity, it is sometimes called the primary pacemaker.

How fast the heart beats depends on how fast the pacemaker cells can reach their treshold potential. The resting potential of a pacemaker cell is -60mV to -70mV. But trough a inflow of sodium and an outflow of potassium the cell becomes more positive. The treshold potential of a pacemaker cell is between -40mV to -50mV.

When the SA node receives sympathetic stimulation, the stress hormone noradrenaline released from the nerve endings binds to β1-adrenergic receptors on the pacemaker cell membrane. This binding causes cyclic AMP production within the cell. This means that sodium is continually entering the cell more quickly. Cyclic AMP also activates a protein kinase, that phosphorylates the calcium channels, increasing calcium conductance into the cell. Because both sodium, and calcium can enter the cell more quickly, the continuously natural depolarisation reaches threshold more quickly. So action potentials are generated more frequently. It takes a while for the heart rate to increase after noradrenaline is released.

When the SA node receives parasympathetic stimulation Acetylcholine (ACh) is released from the vagus nerve endings, and binds to receptors on the pacemaker cells.

In the pacemaker cells, there are ACh sensitive potassium channels. These open in response to ACh binding, potassium ions leak out, and the cell gets hyperpolarised (more negative). The funny current is also reduced by ACh. This means sodium ions enter more slowly, and it takes longer for the cell to reach threshold. Thus the heart rate slows. Unlike the sympathetic mechanism, the heart will slow quite soon after vagal stimulation.

The HR is also influenced by noradrenaline and adrenaline released into the bloodstream by the adrenal medulla. This have the same action on heart rate as direct sympathetic stimulation.[13]

Several previous studies examined the connection between mental stress and heart rate [14, 15, 16, 17]. In these tests, subjects were exposed to different commonly used mental stressors such as Stroops color-word conflict test and arithmetic stressors. In the tests, heart rate was measured during the stressors and during periods of relaxation between the stressors. The results suggest that mental stress increases heart rate. In [15, 16, 17] the mean heart rate was ~10 bpm higher during the stressor compared to the period of relaxation. In [14] the raise was ~20 bpm.

However, there are many factors that influence the heart rate such as different medicines which affects the ANS function. Stimulus such as caffeine, nicotine and alcohol also influence the HR, as well as, the time of the day (highest HR is usually measured just before the awakening and about three hours after). Level of physical condition plays an important role in comparisons between different individuals. For about 6% of the population, the HR does not increase during mental stress at all. Stress in this group cannot be measures with heart rate [1].

HR as a stress indicator can be compared between different age groups because the HR does not change much with age as compared to many other physiological parameters. [1, 18]. The maximum heart rate (HRmax) has been shown to decrease some with age. A 2003 study [13] of 43 different formulae for HRmax concluded that the most accurate formula for for calculating HRmax was: Hrmax = 205.8 - (0.685 * Age). Normal resting heart rate (HRRest) are in the range from 60 to 80. Fit people generally have lower HRrest. It is also likely to increase with age [18].

Stein et al [19] found that HR is gender related. The baseline HR was shown to be higher in females. Similar results were shown in [18]. The difference decreased to some degree with age in [19].

2.1.3 Heart rate variability

The most common way to measure the activity in the ANS is to measure heart rate variability (HRV). HRV can be defined as the regularity of the heart beats [1]. In other words, it refers to the amount of variance in the time period between successive R-R spikes (time elapsed from beat to beat) in the cardiac cycle. The term "HRV" is commonly conventionally used to describe these heart rate fluctuations [20]. Figure 2 shows an electrocardiogram with two R spikes.

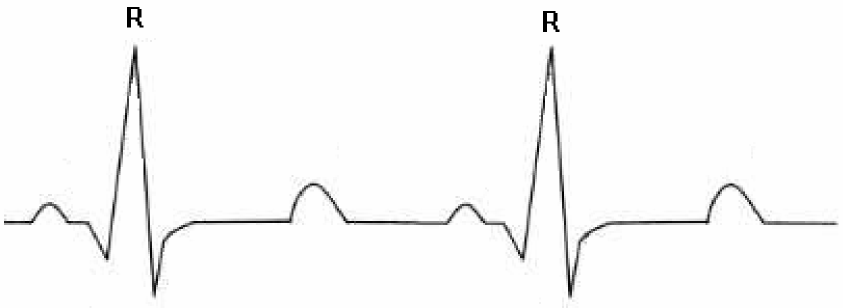


Figure 2: Electrocardiogram

Normal heart rate variation is characterized by a relationship between sympathetic and parasympathetic parts of the ANS. The balance between them is referred to as the sympathovagal balance and is believed to be reflected in the R-R changes of the cardiac cycle. Heart rate increases by slow acting sympathetic activity and decreases with fast acting parasympathetic (vagal) activity. HRV provides a non-invasive assessment of both the parasympathetic and sympathetic control of heart rate in real life conditions [1].

There are several different ways to analyse HRV. The methods can be devided into two main categories. Time and frequency domain methods. SDNN and pnn30 are two examples of time domain methods. SDNN is defined as the standard deviation of all RR intervals in a sample period and is measured in milliseconds. Pnn30 is measured in percent and is defined as the proportion of all RR intervals that have a difference of more than 50 ms from the proceeding one. More methods with short descriptions are presented in appendix 1.

HRV can by sampled at different frequencies. The two main frequency bands of interest are the Low-Frequency (LF) band (0.04 to 0.15 Hz) and the High-Frequency (HF) band (0.15 to 0.4 Hz) [21]. Sympathetic activity is believed to influence the LF component whereas both sympathetic and parasympathetic activities have an effect on the HF component [22].

HRV in resting individuals is linked with respiration, a phenomenon known as respiratory sinus arrhythmia (RSA) [23]. The RSA oscillation manifests itself as a peak in the HF band of the spectrum. For example, 15 breaths per minute correspond to a 4 second oscillation with a peak in the power spectrum at 0.25 Hz.

A high variability in heart rate is a sign of good adaptability, implying a healthy individual with well-functioning autonomic control mechanisms. Conversely, lower variability is often an indicator of abnormal and insufficient adaptability of the autonomic nervous system. [24]. Decreased HRV has been associated with both panic disorder and depression. [25, 26]. Decreased HRV has also been shown in patients with different diagnosis such as congenital heart disease [27]

Results from several studies suggests that increasing stress decreases heart rate variability both in prolonged [28] and short term [29] exposure to psychosocial stressors.

Decreased HRV has been associated with mental stress in laboratory experiments [30, 31] But there are also examples of studies that have failed to show an association between HRV and mental stress [16, 17, 33]. In the study by Wahlström et al. [17] time pressure and verbal provocation were used as stressors. In [16] and [32] the stressors were mainly cognitive (e.g. Stroops color-word conflict test (CWT). In this regard Garde et al. (2002) [32] suggested that the effects of the mental demands in a CWT were too small to elicit significant changes in HRV variables.

Lagngewitz and Ruddel [33] examined the relationship between stressor tasks, HRV and heart rate. In this study 135 healthy men were submitted to two mentally demanding tasks, reaction time and mental arithmetic plus noise. Subjects were required to say their answers to the mental arithmetic problems aloud, and heart rate reactivity and HRV data were collected during the stressor periods.

The result indicated a significant increase in heart rate and a decrease in HRV during the reaction-time task, and an increase in heart rate but no significant change in HRV during the mental arithmetic task. It was suggested that these discrepant results were because speaking their answers to the mental arithmetic problems aloud interfered with subjects' respiration patterns and made the collection of HRV data unreliable, thus leaving open the question of whether mental arithmetic actually does decrease HRV when delivered without the need for subjects to say their answers aloud.

2.2 Technology

2.2.1 Pulse Oximetry Theory

Pulse Oximetry is an optical method for measuring oxygenated hemoglobin in the blood. It is based on the ability of different forms of hemoglobin to absorb light of different wavelengths. Oxygenated hemoglobin (HbO2) absorbs light in the red spectrum (hence the red color of oxygenated blood) and deoxygenated or reduced hemoglobin (RHb) absorbs light in the near-infrared spectrum. The variations of these values is commonly referred to as the plethysmographic waveform (PPG) which can be used to determine the heart rate. Functional arterial oxygen saturation (Sa02) is defined as "the ratio of HbO2 to the total amount of arterial Hb available for reversible oxygen binding". The pulse oximetry value (Sp02) is an estimation of Sa02 measured with the pulse oximeter and is often within a few percent of the Sa02 value.

$$S_pO_2 = \frac{[HbO_2]}{[HbO_2] + [Hb]} \qquad (1)$$

The relationship between each pulse and the light measured by a pulse oximeter is shown in Figure 3. There is a delay between the electrocardiogram and the optical pulse. This comes from the time it takes for the pulse to go from the heart to where the pulse oximeter is placed. [34]

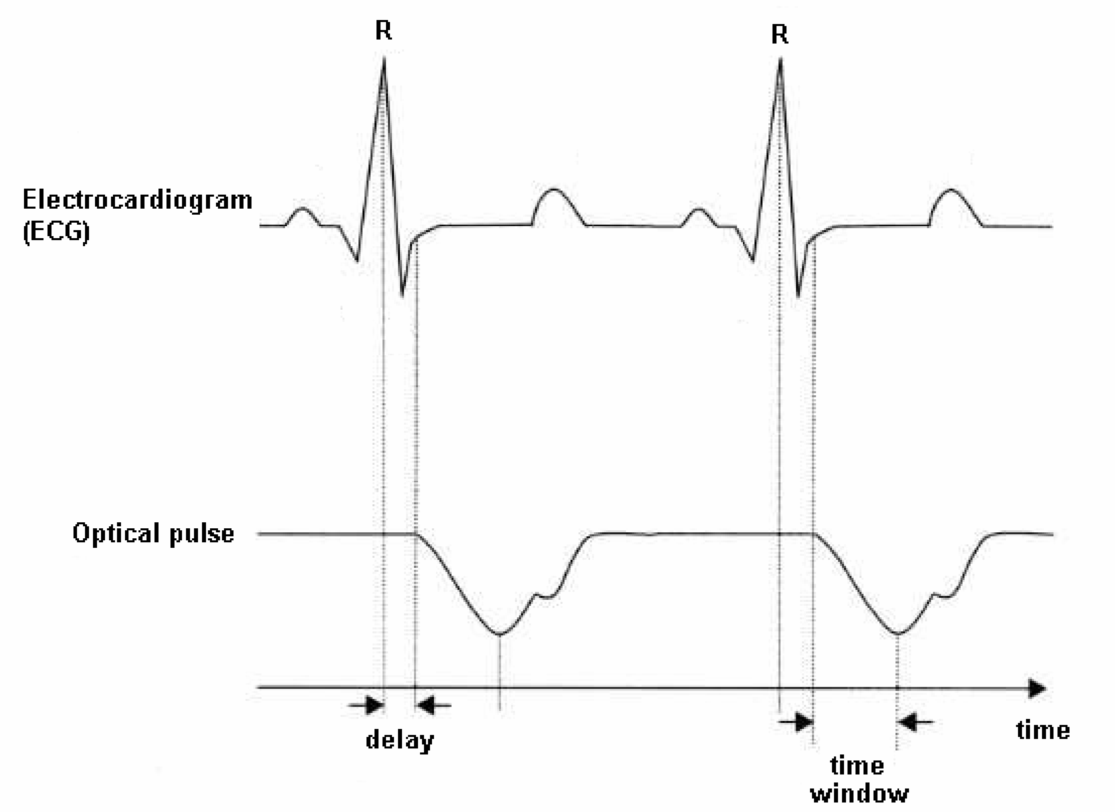


Figure 3. Optical pulse compared to electrocardiogram.

2.2.2 The OSC protocol

Open sound control ("OSC") is a protocol for communication among computers, sound synthesizers and other multimedia devices that optimally utilizes modern networking technology. The protocol was developed by Matthew Wright and Adrian Freed at the U.C. Berkeley Center for New Music and Audio Technologies in 1997 The unit of transmission in OSC is an OSC package. Any application that sends an OSC package is an OSC client and an application that receives packages is an OSC server. OSC is a high-level application protocol meaning that it does not specify which low level networking technology will be used to move the OSC packages from a client to a server. An OSC package can be encoded for network protocols such as

UDP or TCP. The underlying network protocol is responsible to deliver both the OSC packet and its size to the destination.

All OSC data is composed of the following data types: int32, float32, OSC-string, OSC-blob and OSC-time tag. The length of the first two data types is a multiple of four bytes. OSC strings always ends with a null character followed by additional null characters sufficient to make resulting packet size a multiple of four bytes.

The content of an OSC packet consists of either a single OSC message or several messages combined into an OSC Bundle. A message consists of an address pattern followed by a type tag string followed by arguments. The address pattern is a string that specifies the entity or entities within the OSC server to which the message is directed. It also provides the server information about the message type. The address pattern always begins with a forward slash and follows a URL or directory tree structure, such as `/voices/synth1/osc1/modfreq`.

The type tag string provides the data type of each argument. The string starts with a comma followed by a sequence of characters corresponding exactly to the sequence of arguments in the given message. The type tag string always ends with a null character followed by additional null characters to make the total number of bytes a multiple of four. The arguments represent the actual control messages. The OSC standard allows arguments of the data types mentioned above. But some OSC applications communicate among instances of themselves with additional, nonstandard argument types beyond those [35].

2.2.1 The UDP protocol

User Datagram Protocol (UDP) is a connectionless protocol that, like TCP, runs on top of IP networks. Unlike TCP/IP, UDP/IP provides very few error recovery services, offering instead a direct way to send and receive datagram’s over an IP network. The protocol is used primarily for broadcasting messages over a network [36].

2.2.2 MULLE

MULLE is a small and standalone EIS (Embedded Internet System) platform based on a Mitsubishi M16 microprocessor, figure 4. It was developed by EISlab at Luleå University of Technology and is one of the smallest multifunctional platforms designed for wireless, ad-hoc sensor networks. MULLE features include the following:

- Analog and digital interfaces
- Signal tracking capabilities
- Integrated web server
- Small size (25*23*5 mm)
- Low power consumption

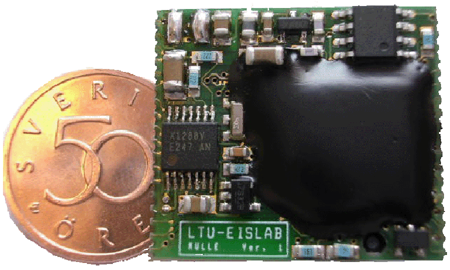


Figure 4. The MULLE platform.

MULLE is also equipped with the WML-C10AHR Bluetooth module from Mitsumi and is capable of wireless communication with Bluetooth compatible devices. The currently integrated module conforms to Bluetooth version 1.1 and supports Bluetooth class 2 [37].

2.2.3 The REMUPP music player

REMUPP is developed at the Interactive Institute and stands for Relations Between Musical Parameters and Perceived Properties, It is s a software-based tool designed for non-verbal testing of various musical functions – for use within several different disciplines of music research. It allows the user to change different parameters in the music, such as tempo, instrumentation and rhythmic complexity, in real time.

The music player used in REMUPP assembles in real time a piece of music based on two inputs. The first input is a music example object that supplies the musical raw material to the music player. The other input is a number of musical parameters that are used to influence the way the music player uses the raw material to assemble the final music.

Figure 5 shows the relation between the music player on one hand and the music example and the musical parameters on the other. It also shows the internal structure of the music engine.

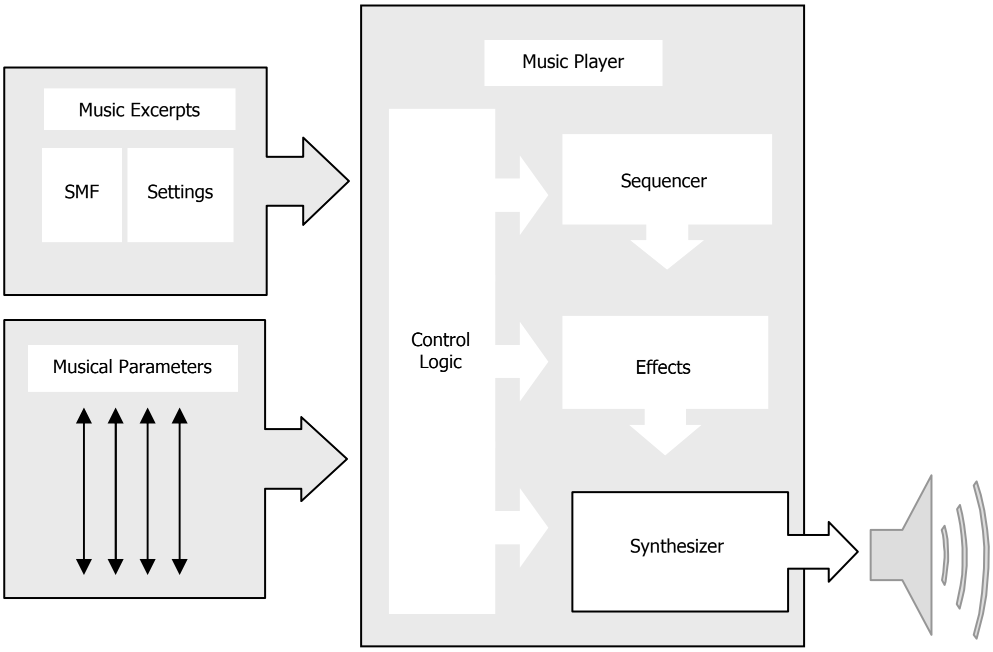


Figure 5. Conceptual outline of the Remupp music engine

The musical raw material fed to the music player from the music example, consists of MIDI data read from a standard MIDI file (SMF) pointed to by the music example. The SMF is loaded into the music players MIDI-sequencer, to manipulate the MIDI-data it contains and the way the MIDI-data finally is rendered to sound by the synthesizers. The musical parameters are of the following different kinds:

- Parameters that controls the sequencer. The tempo parameter is one example.
- Parameters that control the synthesizer used to finally render the music from the MIDI stream. Instrumentation is one example.
- Parameters that acts as filters and effects on the outgoing MIDI stream. These parameters are in turn off two different kinds. 1. Filtering out MIDI-messages by muting and unmuting tracks of the SMF, Rhythmic complexity is one example. 2. Manipulating individual MIDI messages. Articulation is one example where the length of the notes is altered. Register is another example where the pitch of notes is altered.

The different versions of the Remupp application was developed using Macromedia Director and Sibelius SequenceXtra. The MIDI music was rendered by VST sampler Halion and VST synthesizer Hypersonic from Steinberg, using the V-stack, also from Steinberg as VST host application [38].

3 Implementation

3.1 Heart rate sensor

There are a several manufacturers who produce sensors that can measure heart rate. Nonin Medical Inc, situated in Minneapolis USA, manufactures professional physiological monitoring equipment. EISlabs MULLE has used them in previous projects with great success. Since EISlab has already developed software adapted for communication between MULLE and Nonin sensors, we were able to adapt existing software for MULLE for our needs relatively quickly.

All models manufactured by Nonin Medical Inc use the oximetry technique to determine the heart rate. Nonin Medical Inc manufactures many different models of pulse oximeter sensors, each is customized for different applications. The sensors all use infrared light with 910 nm wavelength and red light with 660 nm wavelength. The following variables distinguish the different sensors.

- Placement of sensor contact.
- Children/adult model.
- Disposable/reusable sensor contact.

The sensor has two parts, the sensor contact that converts the signal emitted by the wearer and the electronics that convert the raw signal into the appropriate data format and manages data output. In some of the models, these two components are contained in one housing, but in others they are separated into two distinct parts.

Nonin Medical Inc sensors offer two, different data formats (determined by the model of external electronics). The first data format is rather simple while the second data format supplies more information about the biometric signal.

Data format 1

- Rate: 3 bytes of data per second
- Data: Heart rate, oxygen saturation

Data format 2

- Rate: 5 bytes of data 75 times per second
- Data: Heart rate, oxygen saturation, averaging data (heart rate and oxygen saturation), beat to beat value (blood pressure) and detailed plethysmographic wave data.

We evaluated two different sensor models, one ear clip sensor contact (Nonin 8000Q) and one finger mounted sensor contact (Nonin 7000J). Both models have external electronics and provide output in data format 2. We opted for external electronics and data format 2 for the following reasons.

- The external electronics provide greater design flexibility since it is possible to exchange just the sensor contact.
- External electronics allow for smaller and lighter weight sensor contacts.
- Data format 2 gives us greater opportunities for further development on the existing platform.

After evaluating the two sensor contacts, we selected the ear clip for the following reasons:

- The ear clip uses a mirror technique to receive the infrared light. This makes it less sensitive to movement and placement.
- It keeps the users hands free.

One negative aspect of the ear clip sensors is that they generally do not perform as well as sensors located on the fingers. They are therefore not recommended for applications where the best possible SpO2 accuracy is important. But since that is not the case in our application we decided to use the ear clip sensor. A technical specification of the data formats are presented in appendix 2.

3.1.1 Sensor connection

The sensor and the computer communicate via wireless connection. The sensor sends data along a serial connection using a RS232 DB9 cable. This connection has not yet been configured for wireless transmission. The EIS platform MULLE has suitable serial port capability and can be connected to the sensor. Two versions of custom software have been developed at EISlab for MULLE, both provide the necessary functionality for this prototype. In both versions the incoming sensor data is buffered.

The first version sends 125 bytes of data 3 times a second. This is preferable when very rapid sensor readings are required. The second version sends 375 bytes of data once per second. The first version of the software is preferable when very rapid sensor readings are required. The second version of the software reduces the power consumption by lowering the frequency of data transmissions. Neither version modifies the incoming sensor data. MULLE acts only as a link between serial and wireless connections.

The integrated Bluetooth module on MULLE bridges the connection between MULLE and the computer. The computer's Bluetooth interface includes a 3CREB96 USB dongle from 3COM. The 3COM Bluetooth driver supports communication via serial port from any integrated application.

3.1.2 Serial port implementation in Java

Two specially developed java classes handle the serial port communication and synchronizing of sensor data. The MULLE software handles the synchronization of data frames. In addition, to make the computer application adaptable for direct sensor connections via RS232 cable, an additional java application was implemented to handle data synchronization. This secondary level of code helps prevent problems associated with data loss in the Bluetooth link.

The java class SerialConnection opens and maintains a two way serial port connection with MULLE. A command to MULLE starts data flow to the serial port. The java class DataBlock then synchronizes the incoming sensor data. When new data is available on the serial port the constructor of the class examines each packet for new frame headers. New frames are buffered until the next new frame is discovered. The class also contains the functions getPulse and getPleth. These functions respectively identify the heart rate and plethysmographic data in the stored data frame.

3.2 Regulator design

3.2.1 Medical expertise

The regulating system utilizes medical expertise to gather sensor input data and estimate the individuals´ current level of stress. This medical expertise is quite simple in this first prototype and is much based on the theory that a high heart rate responds to a high level of stress.

We wanted as much of the prototype to be written in Java as possible. To that end we decided to incorporate the medical modeling in the Java application. Ideally we would have liked music generating functionality in Java, as well. But Java does not yet offer high-quality music control on the level needed in this implementation. Besides, there was already a suitable music system available created for the REMUPP project. Therefore we decided to use this system in the first version of BodyResT.

The Java application communicates with the music generating system through the REMUPP music engine application. This music engine is developed at the Sonic studio and is a result of several research projects.

The regulator analyzes the sensor data and sends a message to the music engine when the measured level of stress changes. The regulator only sends one parameter to the music engine, the stress level. In practice, this value is of the data type float and between zero and one.

The music psychological competence of the prototype is located in the music engine. The music engine receives and interprets the information from the regulator and changes the parameters in the music.

3.2.2 Regulator technology

The Proportional regulator is perhaps the most common technique in feedback regulating systems. One characteristic of this kind of regulator technique is its handling of large error in the signal (i.e. difference between reference signal and actual system output). A large error signal will result in a strong control signal to compensate for the error and reach the preferred system output. In this prototype the system output is the individual's current level of stress, and the control is changes of different parameters in the music. Preferred system output is achieved when the individual is relaxed.

A hypothesis was formed suggesting that the above described regulator techniques are not ideal for this implementation. Results from previous research studies suggest that people tend to choose music to match their current mode [11]. These results are also supported by the results of a pre-study investigation about relaxing music and well being, performed at the Interactive Institute. Music that is very calming may not be an optimal treatment for a stressed person. On the basis of this suggestion the regulating system must be able to adapt the music to the individual's current mood to work effectively.

On the basis of this hypothesis a new design was developed based on a stepwise, or step ladder, model of change. The regulator estimates the individual's level of stress based on heart rate data. The music engine then adjusts different parameters in the music to be calming according to the individual's current stress level. When a new, lower level of stress is reached, the parameters are changed again to a new and more calming level. In this way the regulator helps the individual to reduce the level of stress and finally reach a relaxed state in incremental steps, figure 6**.**

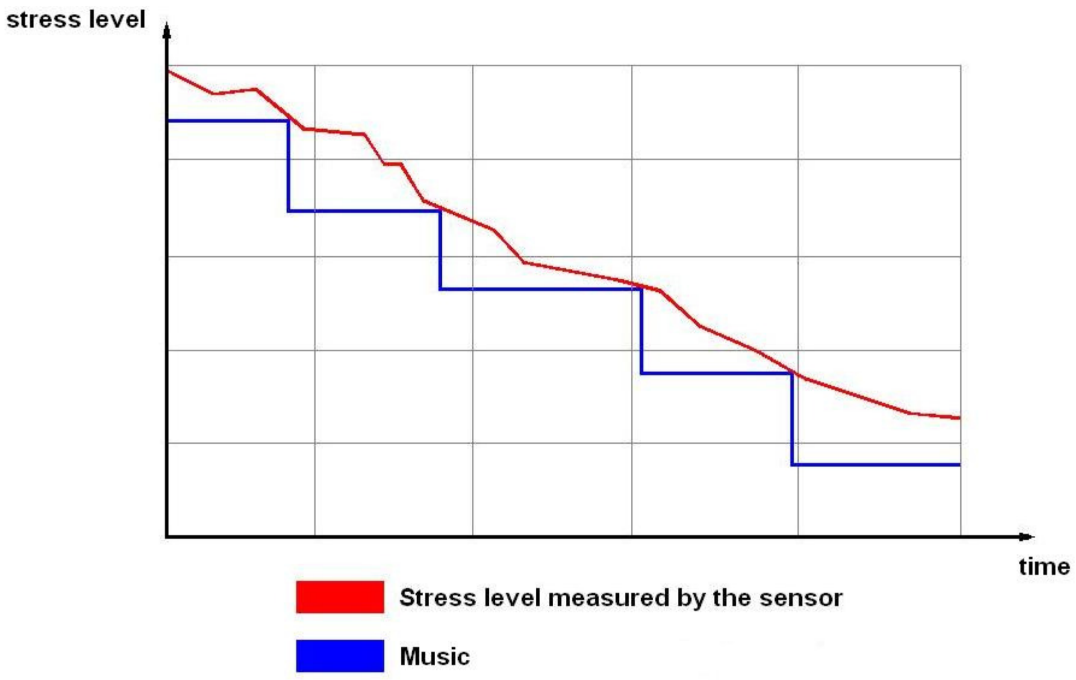


Figure 6: The regulator's way to calibrate the music on the basis of estimated stress level.

5.4.3 Regulator implementation in Java

The regulator is implemented as a single Java class. This class contains two essential functions, StepLadder and FilterStream. The FilterStream function is a low pass filter for incoming sensor data built on the floating mean value principle. This function prevents jumps in values in the control output signal by making the system react slower to changes in pulse rate. The StepLadder function decides the level of stress based on the heart rate mean value. It is possible to define three intervals with different resolutions of steps.

To determine the level of stress, the function first determines the resolution interval of the heart rate mean value. If the mean value lies within, for example, the first resolution interval, the function decides within which step in the first resolution interval the mean value lies. If this step is not the same as the previous interval, an OSC message will be sent to the music engine to inform it that a new step has been entered. Each time new heart rate data enters the regulator class, a new mean value will be calculated which will be passed to the StepLadder function.

3.2.3 Changing musical parameters

The ongoing study in the biofeedback project examines what musical parameters are useful for stress reduction and yield parameters that can be implemented and tested in the BodyResT prototype. Since these new signals have not yet been completely identified it is important that the BodyResT design is flexible enough to allow for new musical parameters to easily be implemented. To accommodate this requirement, settings for the musical parameters are located in the music file. A special designed musical composition, composed by Stefan Lindberg, was created for testing the BodyResT prototype. In this music piece two dynamic parameters were implemented.

The music has been customized for BodyResT in two ways. Firstly, the music works as a monitor for the biofeedback principle helping the individual to learn how to control an autonomic body function, in this case the heart rate. To make the changes readily apparent to the listener, instrumentation was selected as a variable musical parameter. Secondly, the music has a stress reducing effect. Tempo is a parameter that is assumed to have stress related characteristics [8]. The pre-study performed at the Interactive institute also implies that people tend to relate a low tempo with calm music and high tempo with exciting music. For this reasons, tempo was selected as a second parameter.

3.2.4 OSC or MIDI?

A two way communication was implemented between the Java application and the music engine. The java application must be able to communicate current stress level as well as commands for controlling basic functions to the music engine such as play, stop and load music. The music engine must be able to provide the java application with necessary information about current settings and available options for a specific music piece. The communication only requires a message based connection; no information needs to be streamed between the two applications.

Open Sound Control (OSC) and MIDI are two standard protocols used in message-based communication between music devices. Overall, the OSC protocol can be seen as a more modern and powerful standard than the older MIDI protocol. The OSC standard was implemented instead of MIDI for the following reasons:

- OSC standard allows arguments of four data types while MIDI only allows numbers between 0 - 127.
- MIDI has a set of standard message destination addresses, adapted for functions in music devices. OSC allows the application developer to define an unlimited number of addresses.
- OSC is a network technology independent protocol while MIDI must be used together with MIDI or USB cable.

3.2.5 Implementation of OSC in Java

Functionality for both receiving and sending OSC packages, with UDP as an underlying network protocol, was implemented in the Java application. A three step guide how the classes are used to send an OSC message is shown in figure 7.

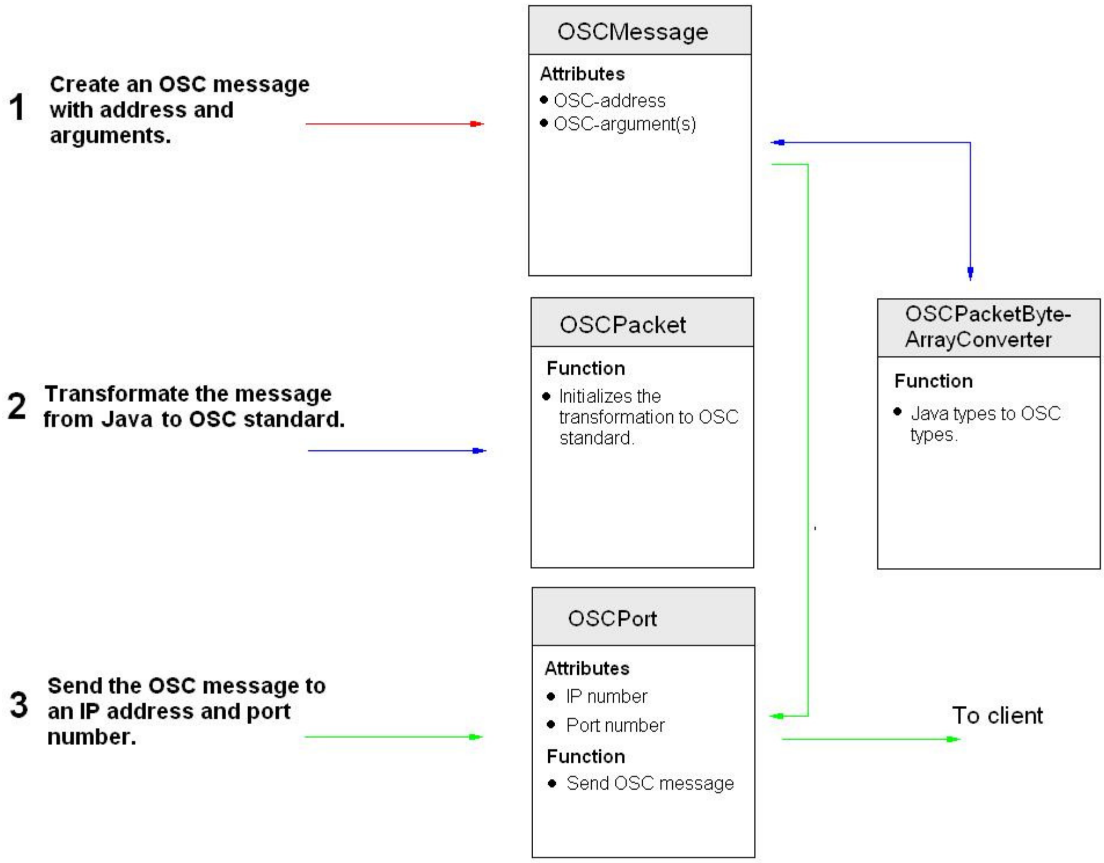


Figure 7: A conceptual outline of Java classes used for sending OSC messages.

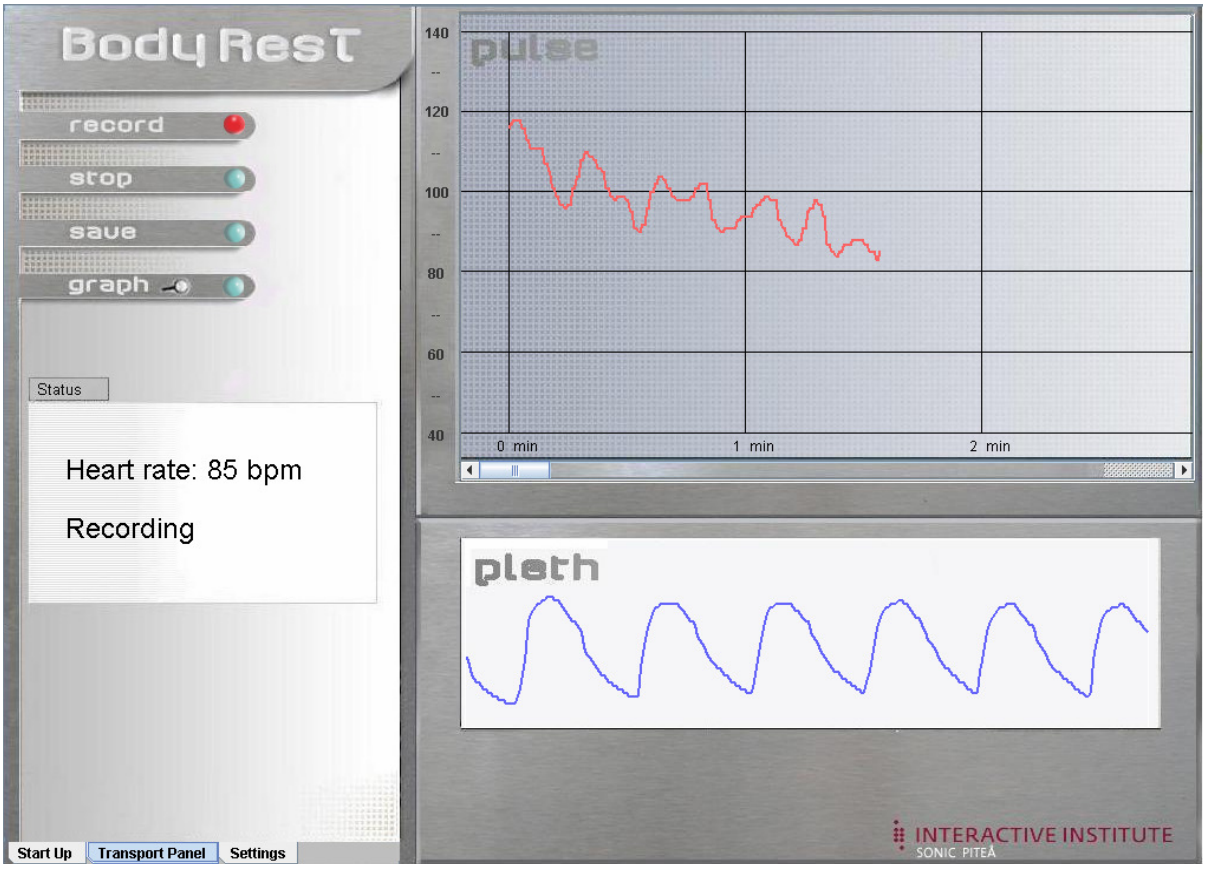
The first step in sending an OSC message is to create an OSCMessage. An OSC message has two attributes, an OSC address and an array of attributes. OSC Message is a subclass of OSCPacket. By calling the initialization function in the OSCPacket class the OSC message is transformed to OSC standard. This is done by using functions in the OSCPacketByteArrayConverter class and results in the OSCPackage an array of bytes. By calling the send method in the OSCPort class a UDP datagram is created containing the array of bytes along with its size, destination IP address and port. Finally, the UDP datagram is sent.

The class OSCInPort handles incoming UDP datagrams on a specified port. By using functions in the OSCInDatagram class the incoming OSC Message is encoded to Java standard. Depending on the OSC address, a specified function in the OSCInPort class is called that directs the arguments to its final destination.

Complete documentation of all implemented OSC messages used in the communication between the java application and music engine are included in appendix 2.

4 BodyResT Interface

In collaboration with the graphic designer, Ingemar Almeros, a graphical user interface design was created for the BodyRest prototype, figure 8. This GUI includes controls for accessing a variety of functions including start up of sensor connection, adjustments of music engine and running the system. The GUI consists of a control panel and a display area for displaying sensor readings.



Heart rate: 85 bpm

Recording

Figure 8: The BodyResT graphical user interface

The different controls are arranged on three, separate, tabbed panels, figure 9. These tabbed panels are labeled “Start Up”, “Settings” and “Transport Panel”.

The Start Up panel includes the functionality needed to start up the regulating system and make it ready for a session including sensor connection and loading of music example. When the sensor is connected properly the readings appear on a status display. The user or experimenter can select which musical parameters that will be changed by the music engine.

The Settings panel includes the following three panels for changing default settings of the prototype:

- **OSC Settings**
  Here the user can change destination address and port for outgoing OSC messages for cases when the music generating system is run on a separate computer.

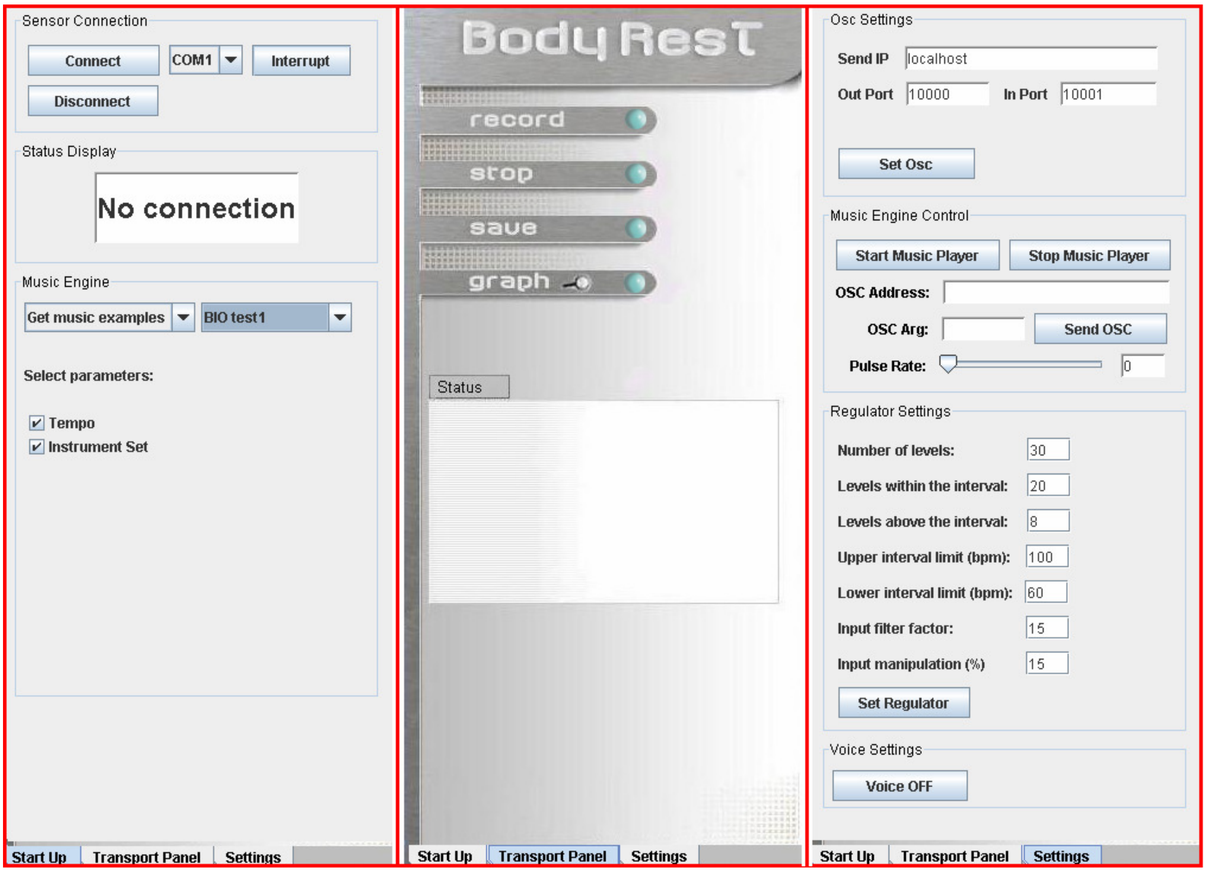
- **Music Engine Control**
  This panel includes manual controls for the music engine. This gives the user the ability to listen to the music synthesized by the music engine and modify the music engine functionality without performing a session. OSC messages can be manually typed and sent to the music engine and sensor readings simulated.

- **Regulator settings**
  This panel allows the user to adjust the properties of the regulator.

- **Voice settings**
  For turning on and of a synthetic generated voice. This voice gives the user/experimenter information regarding different processes at the same time as the information is shown in the display.

The test panel is used when the user is performing a session. On the panel are the three buttons associated with performing the session, a button for enlarging the graph displaying sensor readings and a system status display.

- **Record**
  This button initializes a test. When the regulator has finished collecting enough sensor readings to estimate the initial stress level the music starts playing and the pulse rate is displayed on the graph.
- **Stop**
  This button stops a running test.
- **Save**
  This button saves a session. Information about test subject, music piece, used musical parameters, date and time of test session, test length and sensor readings are saved to file.



COM1

No connection

Get music examples
BIO test1

Select parameters:

☑ Tempo
☑ Instrument Set

Send IP: localhost
Out Port: 10000
In Port: 10001

OSC Address:
OSC Arg:
Pulse Rate: 0

Number of levels: 30
Levels within the interval: 20
Levels above the interval: 8
Upper interval limit (bpm): 100
Lower interval limit (bpm): 60
Input filter factor: 15
Input manipulation (%) 15

Figure 9: The different views of the control panel.

4.1 Display area

The display area includes two graph windows. The first window displays heart rate data over time during a test. The second window displays raw plethmyographic readings as soon as the sensor is connected. This plethmyographic waveform indicates that the system is working properly and gives the continuous status of the link between sensor and application.

5 Tests

5.1 Prototype functionality

Three different tests were performed examining functionalities of the prototype. These tests do not examine whether the prototype has any medical effect but whether the technical solutions of the system are reliable.

5.2 Sensors ability to detect heart rate on different individuals.

In this test the Nonin ear clip oximeter sensors ability to continuously measure heart rate for 20 minutes and on different individuals was examined. According to the directions for use the ear lobe must be rubbed vigorously for at least 5 seconds before the sensor is applied, otherwise the sensor will not be able to perform well. For future test runs using the BodyResT it is crucial to know whether the sensor is capable of measuring heart rate for a certain time.

5.2.1 Method

15 subjects participated in the study, 7 women and 8 men. Their age ranged from 21 to 54 years. Before the pulse rate sensor was attached, the ear lobe was rubbed for 10 seconds to increase the blood flow. After the sensor was attached to the ear lobe the sensors ability to instantly measure heart rate was examined. Heart rate was then measured continuously during 20 minutes.

5.2.2 Results

The sensor had problem to detect the heart rate for one of the subjects. This subject was a female, age 27. When this subject rubbed the earlobe for additionally 10 seconds heart rate was detected by the sensor, but the pleth waveform displayed in the interface indicated a weak signal (low amplitudes of the peaks).

The sensor had no problem to continuously measure the heart rate for any subject during the 20 minutes.

5.3 Reliability of the Bluetooth connection.

In this test the reliability of the Bluetooth connection between MULLE and computer was examined. Three different aspects of the connection were examined. Connection initialization (a), connection reliability when running the connection for a longer time (b), maximum range (c).

5.3.1 Method

A 3 COM Bluetooth USB dongle was used as Bluetooth interface on the computer side. In (a) 30 connections were initialized between MULLE and computer. In (b) the connection was run 15 times for 20 minutes. There was also one long time test for 6 hours. In both (a) and (b) the distance was approximately one meter. In (c) 10 connections were initialized. The MULLE device was then moved slowly away from the computer until connection was broken. The distance was then measured with a tape measure. There was always clear visibility between computer and MULLE. To reduce the problem with available space the range test was run outdoors.

5.3.2 Results

There was a problem to initialize the bluetooth connection between computer and MULLE in 8 cases of 30. In 4 cases also a second attempt to connect failed. In (b) two of the 15 connections were broken during the 20 minutes. In (c) the average range was 12.5 meters (max 13 , min 11 meters).

5.4 UDP connection

In this test the reliability of using UDP as underlying network protocol for sending OSC messages was examined. The UDP connection between the java application and the music engine in BodyResT is essential for the functionality. Since UDP is a connectionless network protocol it is crucial to know whether packets are so often lost between retriever and receiver.

5.4.1 Method and results

A java application was created that sends OSC packages in UDP datagrams in a specified frequency. OSC messages were sent over network to an application created in Macromedia Director. This Director application printed all incoming messages in a message box along with the number of total incoming messages. In that way the number of sent messages could be compared to the number of received messages. During the test 650 000 messages were sent, one every 20 ms. The network was frequently burdened with other traffic during the rest.

None of the OSC messages were lost during the test. This suggests that UDP is a sufficient secure protocol to use in BodyResT.

5.5 Heart rate as a stress indicator

This study examines heart rate as a valid indicator of stress. Earlier studies, presented in the theoretical framework chapter, suggest that heart rate is affected by increasing stress. This study complements and confirms earlier findings, and focuses on increases in heart rate during mental stress as compared to periods of relaxation.

5.5.1 Method

Thirteen subjects participated in the study, 7 females and 6 males. Their age ranged from 21 to 54 (mean age 34). The subjects were exposed to three different and commonly used mental stressors during the test session. A variant of Stroops color stress test (A), a mental arithmetic task (B) and a talk preparation stress task (C).

In (A) different cards with a word written on each card are shown to the subject. If the word is a substantive the subject is supposed to read aloud the word. If the word is a color the subject is required to name the color of which the word is printed in. The experimenter points out each error. There is a one minute time limit to complete as many cards as possible.

In (B) the subject is required to count backwards, starting at 306 and subtracts 17 until 0 is reached. There is a two minute time limit.

In (C) the subject is given a text (weather forecast). The subject then has two minutes to prepare a speech summarizing the text.

During the session the test subject sat in a comfortable arm chair. The experimenter did not give the subject any information about the purpose of the test. Subjects were given verbal instructions for each task. The BodyResT prototype was used to measure the HR. A relaxation period of three minutes before each stressor allowed HR to stabilize. The HR was recorded after each relaxation period and immediately following each stressor. After the last stressor there was an additional three minutes rest, followed by a last HR notification. The heart rate was recorded continuously by the BodyResT prototype for visual analysis of HR changes during the test. The participants verbally reported their subjective experiences of stress seven times: once before the beginning of the first stressor, once at the end of each of the three stressors, and once after each period of relaxation. A seven point scale (1=not at all stressed, 7 = extremely stressed) was used.

The measured heart rate and the subjective response after the first period of relaxation were set as a baseline for each subject. The variations from this baseline were then calculated for each subject and relaxation/stressor period. The mean difference was also calculated for the total group, males only and females only.

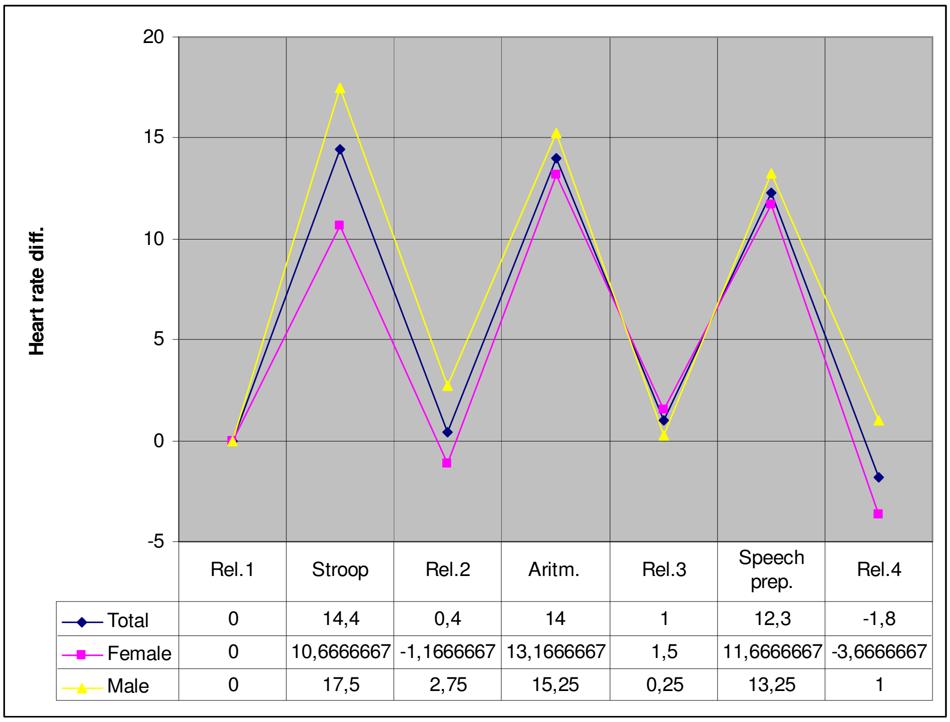
5.5.2 Result

The results show the mean difference in heart rate is higher in the end, after each stressor, compared to the periods of relaxation, figure 10. The subjective stress responses follow the same pattern, figure 11. The difference in the mean subjective stress is about two points higher after all of the stressors than during relaxation periods for the entire group. This indicates that subjects experienced only mild stress from the tasks.

The heart rate as well as the subjective stress response was close to the baseline after each period of relaxation. This indicates that three minutes of relaxation between the stressors was enough.

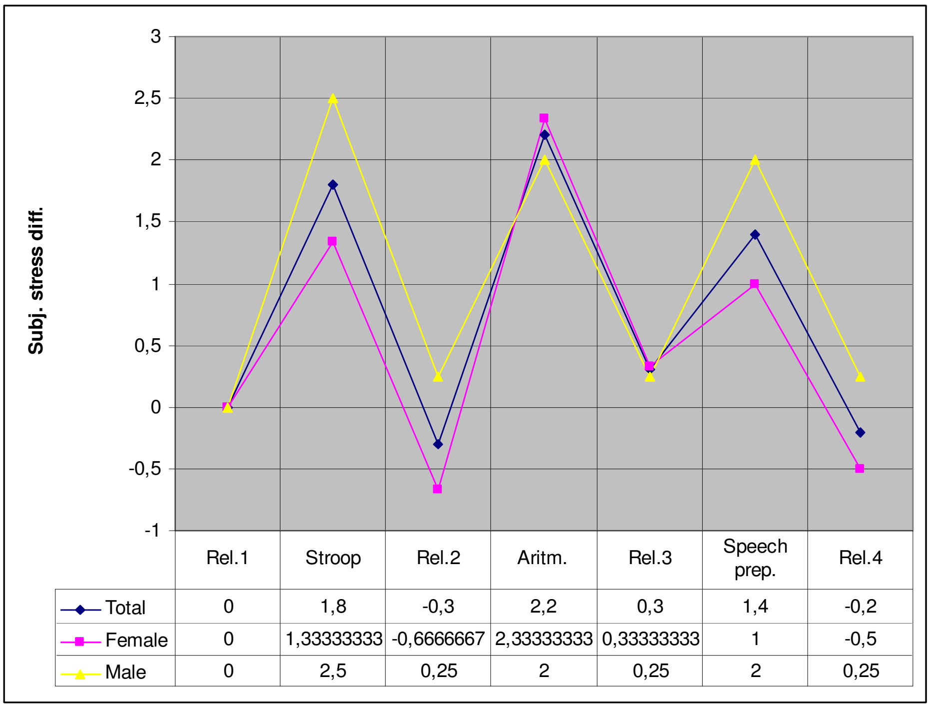
For the entire group, the mean difference in heart rate was 10-15 bpm higher after all the stressors compared to the baseline. The results reveal no substantial difference in patterns of heart rate changes between males and females. However, on the Stroops test males showed a ~7 bpm higher heart rate in mean. The male subjective stress response was also higher compared to the baseline after this stressor.

In the analysis of the visual data, where heart rate was measured continuously during the test sessions, sudden changes in heart rate corresponded with one of several events such as yawns, laughs or when the subject started to talk. During one session a telephone rang which resulted in a rapid increase of ~20 bpm.



	Rel.1	Stroop	Rel.2	Aritm.	Rel.3	Speech prep.	Rel.4
Total	0	14,4	0,4	14	1	12,3	-1,8
Female	0	10,6666667	-1,1666667	13,1666667	1,5	11,6666667	-3,6666667
Male	0	17,5	2,75	15,25	0,25	13,25	1

Figure 10: Mean differences in heart rate.



	Rel.1	Stroop	Rel.2	Aritm.	Rel.3	Speech prep.	Rel.4
Total	0	1,8	-0,3	2,2	0,3	1,4	-0,2
Female	0	1,33333333	-0,6666667	2,33333333	0,33333333	1	-0,5
Male	0	2,5	0,25	2	0,25	2	0,25

Figure 11: Mean differences in subjective stress responses.

6 Discussion

6.1 Prototype functionality

The functionality tests described earlier examined suspected weak links in the system. There was one known connection problem. The Bluetooth connection between computer and MULLE has known problems, but generally not to the degree observed in the functionality tests.

In two of the sessions examining the sensor functionality the sensor was not able to detect the heart rate instantly. This problem was alleviated by having the subject rubbed the earlobe for a couple of more seconds before attaching the sensor. Apparently, the sensor functionality is sensitive to low blood flow.

6.2 Heart rate as a stress indicator

The studies mentioned earlier in the theoretical background section suggest that HR is influenced by mental stress. In two of the studies the mean value of the heart rate increased about 10 bpm during mental stress [15,16]. In the Freyschuss et.al [14] study an increase of ~20 bpm was reported. There was no obvious explanation for greater increases in HR in this study.

The study presented in this report on heart rate and mental stress shows results similar to [15,16] even though the method used here differs from the earlier studies. The mean value of the heart rate during the stress session was not calculated. Instead, the heart rate was measured after the stressor ceased. This may have influenced the results since the pre-experiment heart rate tended to be higher than the heart rates post-stressor and post-relaxation period.

One consideration in interpreting the results is the reliability of HR as a measure of stress. Although HR seems to be influenced by mental stress, it is also affected by many other factors. Without any other measurements it is difficult to tell whether an increase in heart rate is a result of stress or something else. In the analysis of the visual data, heart rate measured continuously over time during the test sessions, sudden changes in heart rate corresponded with one of several events such as yawns, laughs or when the subject started to talk. Whether these “spikes” in heart rate actually are connected to these events are to be left outside this discussion, but the observation suggests that short time changes in heart rate are common and occur for many reasons. An application that uses heart rate as a stress indicating parameter must be able to filter out these temporarily changes and focus on the mean value changes of the HR.

Previous studies have suggested that HRV is a robust indicator on the relation between the sympathetic and the parasympathetic parts of the ANS, and therefore may be a more reliable measure of stress. It has been shown that increasing stress of different kind reduces HRV. However, the relation between the sympathetic and parasympathetic parts of the ANS has shown to be influenced by other factors than stress. But overall, HRV is a more reliable physiological parameter for measuring stress than HR.

7 Conclusion and recommendations

7.1 Heart rate as a stress indicator

Both the literature survey and findings from the experimental study examining heart rate as a stress indicator suggest that heart rate is a valid stress indicator. But it is an indicator with low reliability since the heart rate seems to be influenced by many factors other than mental stress. In future versions of BodyResT heart rate will need to be combined with one or more other physiological parameters to increase the system's reliability for measuring stress.

Heart rate variability is an interesting complementary measurement, and has greater reliability than heart rate. Although some tests has failed to bind a low HRV to stress. Also it will not require additional sensors. HRV can be measured with the existing pulse oximeter. As mentioned in the theoretical background there are two categories of methods for measuring HRV, time and frequency domain. The time domain methods are easier to implement. These methods give a more overall measurement of HRV, or are specified for a certain frequency. Frequency domain methods are more complicated but offer the possibility to analyze different frequencies of HRV at the same time. My advise is to start examine the time domain methods and then look more into the frequency models if necessary.

There are other interesting physiologic parameters to investigate such as skin conductance and blood pressure. These parameters can both be measured non-invasively and have shown to have stress indicating characteristics. Skin conductance is especially interesting to investigate since this is a parameter that overall is believed to be unresponsive to voluntary control.

Blood pressure in combination with heart rate can also be an interesting parameter to examine further. The product of these combined parameters is called Rate Pressure Product (RPP). In comparison with other established measurements on the ANS the validity of this measurement is ranked highly [1].

7.2 Prototype development

The investigation of complementary physiological parameters extends to the sensors used for measuring these parameters, and further development of MULLE software enabling it to handle several sensors. The MULLE platform currently allows several inputs, but of only one sensor type.

The Bluetooth module in MULLE also requires further refinements. The results from the functionality tests suggest that there is a major problem with the initialization of the Bluetooth connection.

The music used in BodyResT will change with subsequent iterations of the system design. The piece currently in use is based on results from a preliminary study about musical parameters and stress performed at The Interactive Institute studio Sonic. More detailed analysis of the data collected from this study will greatly inform the composition of new music for the BodyResT prototype.

Subsequent versions of BodyResT will also be more adaptable to individual music preferences. Results from studies examining the connections between music and relaxation suggest that self selected music is important to the efficacy of the relaxing effect [3, 9, 11]. This could partly be realized and tested by implementing more user control of the musical parameters, by for example letting the user self adjust the parameters to a level perceived as relaxing.

The regulating system of the prototype will also undergo modifications. The current regulator technique allows several adjustments. However, to explore how the overall regulating technique behaves BodyResT needs to be tested on real subjects.

An important question is how the regulator should be designed. Is it possible to combine different physiological parameters to a "stress level" that controls the changes in the music? Or should physiological parameters be individually mapped to musical parameters in some way? To increase the understanding about this stress and physiological parameters has to be further studied. How exactly is stress measured in other areas such as in stress tests?

8 References

1. Währborg, Peter., *Stress och den nya ohälsan*. 2003, Stockholm: Natur och kultur.
2. Socialstyrelsen. *Folkhälsorapport 2005,* 2005. Stockholm: Edita Nordsteds Tryckeri.
3. Schneider N, Schedlowski M, Schürmeyer T.H. ,Becker H, Stress reduction through music in patients undergoing cerebral angiography*, Neuroradiology* 2001. 43(6): p 472-476.
4. Iwanaga, M.,Tsukamoto, M,1997. Effects of excitative and sedative music on subjective and physiological responses. *Journal of Music Therapy,* 1997*,* 33: p 219-230.
5. Scartelli, J.P., The effects of EMG biofeedback and sedative music, EMG biofeedback only and sedative music only on frontalis muscle relaxation ability, *Journal of Music Therapy*, 1984, 21: p 67-78.
6. Hanser, S.B. Music therapy and stress reduction research, *Journal of Music Therapy*, 1985, 22: p 193-206.
7. White, J.M., Music therapy: an intervention to reduce anxiety in the myocardial infarction patient, *Clinical Nurse Specialist*, 1992, 6: p 58-63.
8. Myskja, A., & Lindbaek, M. Hvordan virker musikk på menneskekroppen? *Tidskriftet Norska Laegeforen*, 2000, 120: p 1182-1185.
9. Thaut, M.H., Davis, W.B., The influence of subject-selected versus experimenter-chosen music on affect, anxiety, and relaxion, *International Journal of Music Therapy*, 1993, 30: p 210-223.
10. O´Sullivan, R. A musical road of recovery: music in intensive care. *Intensive Care Nursing*, 1991, 7: p 160-163.
11. Juslin, P.N., Laukka P, Expression, perception, and induction of musical emotions: A review and a questionnaire study of everyday listening. *Journal of New Music Research,* 2004, 33: p 217-238.
12. Encyclopedia Britannica Online, 2005, http://www.britannica.com/ebc/article?tocId=9356196&query=br, {Acc. 2005-05-01}
13. Wikipedia, 2005, http://en.wikipedia.org/wiki/Cardiac_pacemaker, {Acc. 2005-08-28}.
14. Freyschuss, U.,P. Hjelmdahl, A. Juhlin-Dannefelt, and Hjemdahl P, Cardiovascular and sympathoadrenal responses to mental stress: a study of sensory intake and rejection reactions. *Acta Physiol. Stand*, 1990, 139: p 173-183.
15. Bohlin, and Perski. A. Muscle symphathetic activity and and norepinephrine release during mental challenge in humans, *American Journal of Physiol*ogy, 1989, 257 (Ednoricol metab. 20): p 654-664.
16. Hoshikawa Y, Yamamoto Y, effects of Stroop color-word conflict test on the autonomic nervous system response. *American Journal of Physiology*, 1997, 272: p 1113-1121
17. Wahlström K, Hageberg M, Johnsom PW, Svensson J, Rempel D., Influence of time pressure and verbal provocation on physiological reactions during

work with a computer mouse. *European Journal of Applied Physiology,* 2002, 87: p 257-263

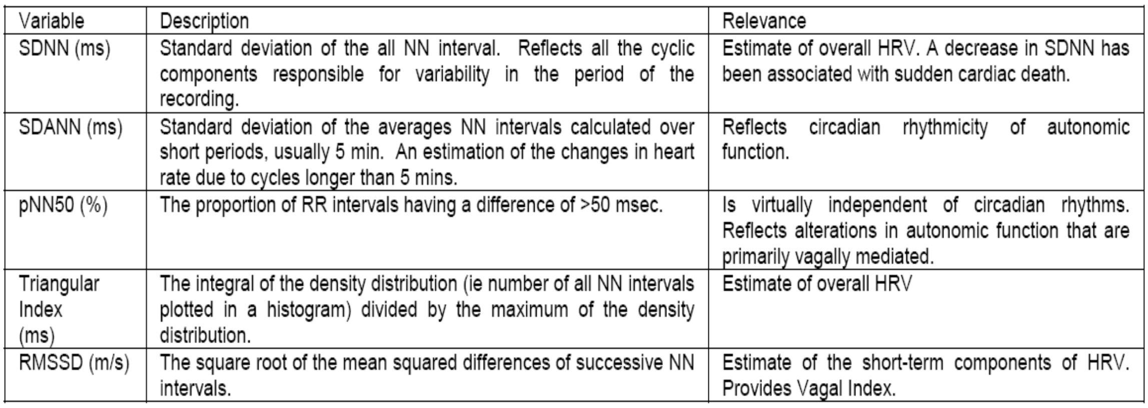
18. Fagard, Robert H. 1,2; Pardaens, Karel 1; Staessen, Jan A. 1 Influence of demographic, anthropometric and lifestyle characteristics on heart rate and its variability in the population. *Journal of Hypertension*, 1999, 17(11): p 1589-1599.
19. Stein PK, Kleiger RE, Rottman JN, Differing effects of age on heart rate variability in men and women. *American Journal of Cardiology*, 1997, 80: p 302–305.
20. Sharpley CF, Kamen P, Galatsis M, Heppel R, Veivers C, Claus K. An examination of the relationship between resting heart rate variability and heart rate reactivity to a mental arithmetic stressor., *Appl Psychophysiol Biofeedback*, 2000, 25(3): p 143-153.
21. Task Force of the European Society of Cardiology and the North American Society of Pacing and Electrophysiology. Heart rate variability: standards of measurement, physiological interpretation, and clinical use, Circulation, 1996, 93: p 1043-1065.
22. M. Malik and A. J. Camm, *Heart Rate Variability*, Futura Publishing, Armonk, NY, 1995
23. Junichiro, H., Fumihiko, Y., Akiyoshi, O., Seiji, M., Takao F, Respiratory Sinus Arrhythmia, 1998, http://circ.ahajournals.org/cgi/content/full/94/4/842 {Acc. 2005-05-03}
24. Evrengül H, Dursunoglu D, Cobankara V. Heart rate variability in patients with rheumatoid arthritis, *Rheumatology International*, 2004, 24(4): p 198-2002.
25. Yeragani, V.K., Pohl, R., Balon, R., Ramesh, C., Glitz, D., Jung, I., & Sherwood, P., Heart rate variability in patients with mayor depression, *Psychiatry Research*, 1991, *37I*(1): p 35–46
26. Yeragani, V. K., Pohl, R., Berger, R., Balon, R., Ramesh, C., Glitz, D., Srinivasan, K., & Weinberg, P. Decreased HRV in panic disorder patients: a study of power-spectral analysis of heart rate, *Psychiatry Research*, 1993, *46*(1): p 89–103.
27. Massin, M., Von Bernuth, G., Clinical and haemodynamic correlates of heart rate variability in children with congenital heart disease, *European Journal of Pediatrics*, 1998, 157(12): p 967-971.
28. Bernardi L., Wdowczyk-Szulc J., Valenti C., Castoldi S., Passino C., Spadacini G, Sleight P, Effects of controlled breathing mental activity and mental stress with or without verbalization on heart rate variability, *J Am Coll Cardiol,* 2000, 35: p 1462–1469.
29. Geus EJ, van Doornen LJ, de Visser DC., Orlebeke JF., Existing and training induced differences in aerobic fitness: the relationship to physiological response patterns during different types of stress, *Psychophysiology*, 1990, 27(4): p 457-478.
30. Myrtek M, Weber D, Brügner G, Müller W, Occupational stress and strain of female students: results of physiological, behavioural, and psychological monitoring. *Biol Psychol,* 1996, 42: p 379–391.
31. Sloan RP, Shapiro PA, Bagiella E, Boni SM, Paik M, Bigger JT, Steinman RC, Gorman JM, Effect of mental stress throughout the day on cardiac autonomic control. *Biol Psychol*, 1994, 37: p 89–99.

32. Garde AH, Laursen B, Jørgensen AH, Jensen BR, Effects of mental and physical demands on heart rate variability and blood pressure variability during computer work, *European Journal of Applied Physiology,* 2002, 87: p 456–461.
33. Gail D. Baura. Improved pulse oximetry. In System Theory and Practical Applications of Biomedical Signals, *Series in Biomedical Engineering*, 2002, IEEE Press, Piscataway, New York.
34. Langewitz, W., & Rudell, H. Spectral analysis of heart rate variability under mental stress, *Journal of Hypertension,* 1989, 7: p 32–33.
35. Matt Wright*, Open sound Control specification, 2002-03-26,* http://www.cnmat.berkeley.edu/OpenSoundControl/OSC-spec.html, {Acc. 2005-02-18}.
36. Webopedia, 2005, http://www.webopedia.com/TERM/U/User_Datagram_Protocol.html, {Acc. 2005-04-02}.
37. Völker, M., Design rules development for chip-on-board and flip-chip mounting technologies, 2004, Technische Universität Berlin: Berlin.
38. Wingstedt, J.,Lliljedah, M., Lindberg S., & Berg, J: REMUPP – An Interactive Tool for Investigating Musical properties and Relations. Accepted for presentation at The International Conference on New Interfaces for Musical Expression, Vancouver, Canada, May 2005.

9 Appendix

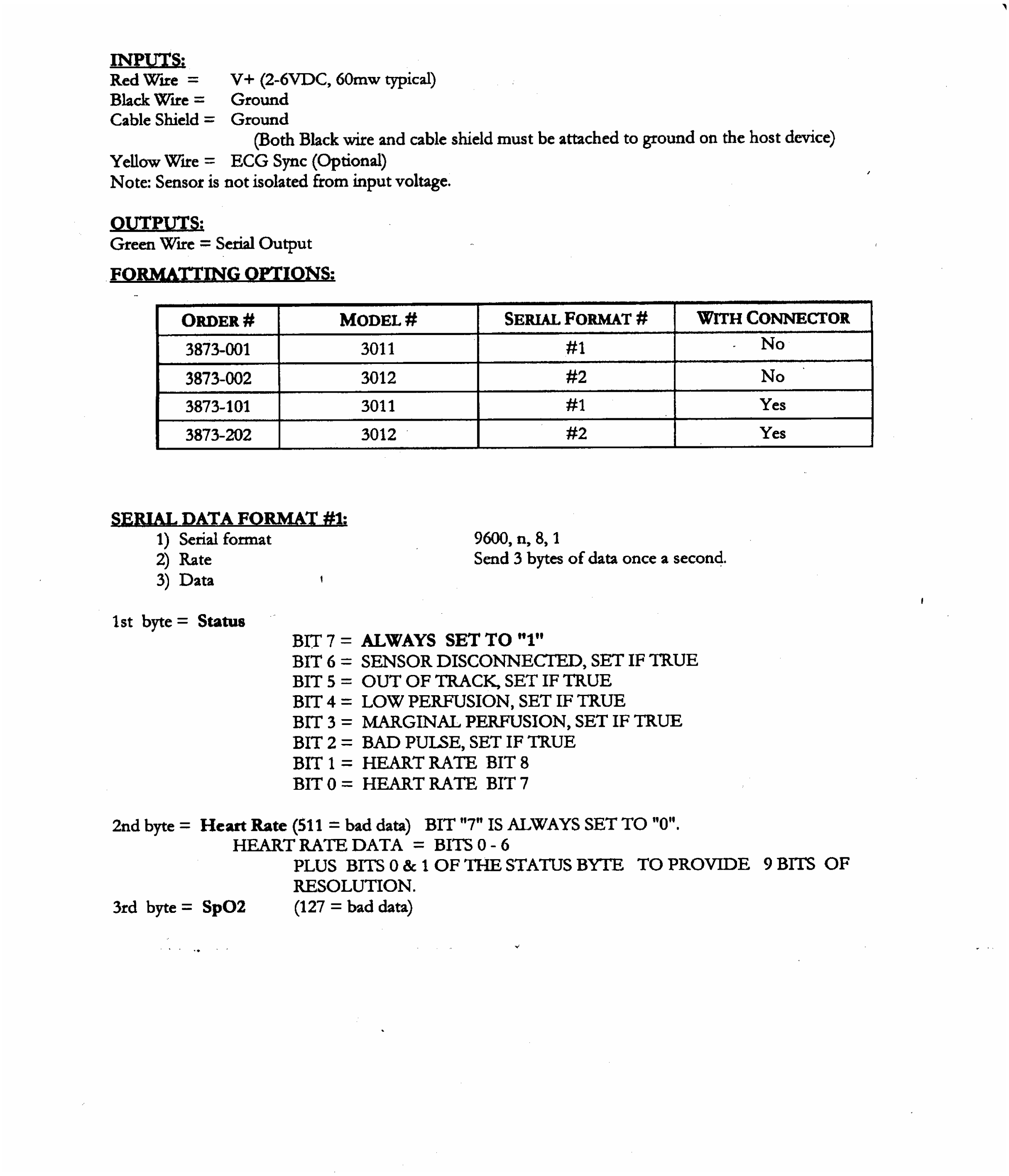
Appendix 1. HRV time domain methods.

Time domain measures are the means and standard deviations of R-R intervals recorded by the continuous ECG, where NN (normal-to-normal) intervals represents all the R-R intervals. The variables of time domain measures are shown in the following table.



Variable	Description	Relevance
SDNN (ms)	Standard deviation of the all NN interval. Reflects all the cyclic components responsible for variability in the period of the recording.	Estimate of overall HRV. A decrease in SDNN has been associated with sudden cardiac death.
SDANN (ms)	Standard deviation of the averages NN intervals calculated over short periods, usually 5 min. An estimation of the changes in heart rate due to cycles longer than 5 mins.	Reflects circadian rhythmicity of autonomic function.
pNN50 (%)	The proportion of RR intervals having a difference of >50 msec.	Is virtually independent of circadian rhythms. Reflects alterations in autonomic function that are primarily vagally mediated.
Triangular Index (ms)	The integral of the density distribution (ie number of all NN intervals plotted in a histogram) divided by the maximum of the density distribution.	Estimate of overall HRV
RMSSD (m/s)	The square root of the mean squared differences of successive NN intervals.	Estimate of the short-term components of HRV. Provides Vagal Index.

Appendix 2. Nonin pulse oximeter data formats.



INPUTS:

Red Wire = V+ (2-6VDC, 60mw typical)
Black Wire = Ground
Cable Shield = Ground
(Both Black wire and cable shield must be attached to ground on the host device)
Yellow Wire = ECG Sync (Optional)
Note: Sensor is not isolated from input voltage.

OUTPUTS:

Green Wire = Serial Output

FORMATTING OPTIONS:

ORDER #	MODEL #	SERIAL FORMAT #	WITH CONNECTOR
3873-001	3011	#1	No
3873-002	3012	#2	No
3873-101	3011	#1	Yes
3873-202	3012	#2	Yes

SERIAL DATA FORMAT #1:

1) Serial format 9600, n, 8, 1
2) Rate Send 3 bytes of data once a second.
3) Data

1st byte = **Status**

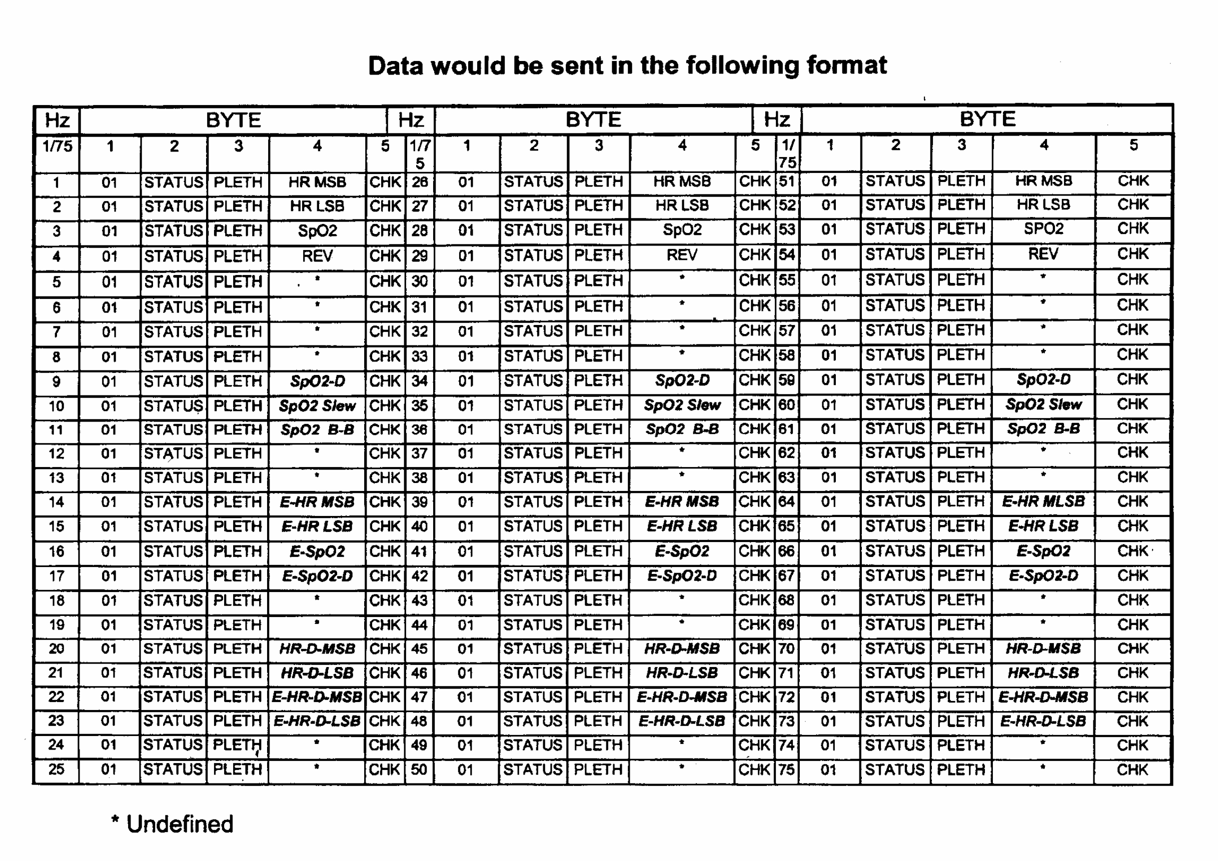
BIT 7 = **ALWAYS SET TO "1"**
BIT 6 = SENSOR DISCONNECTED, SET IF TRUE
BIT 5 = OUT OF TRACK, SET IF TRUE
BIT 4 = LOW PERFUSION, SET IF TRUE
BIT 3 = MARGINAL PERFUSION, SET IF TRUE
BIT 2 = BAD PULSE, SET IF TRUE
BIT 1 = HEART RATE BIT 8
BIT 0 = HEART RATE BIT 7

2nd byte = **Heart Rate** (511 = bad data) BIT "7" IS ALWAYS SET TO "0".
HEART RATE DATA = BITS 0 - 6
PLUS BITS 0 & 1 OF THE STATUS BYTE TO PROVIDE 9 BITS OF RESOLUTION.

3rd byte = **SpO2** (127 = bad data)

SERIAL DATA FORMAT #2:

1) Serial format	9600, n, 8, 1	
2) Rate	Send 5 bytes of data 75 times a second.	
3) Data		
a. HR value bits 7&8 (128-511), 511 = bad data	1 byte	3 times a second
b. HR value bits 0-6 (0-127)	1 byte	3 times a second
c. SpO2 value 0 - 100	1 byte	3 times a second
d. Firmware revision level	1 byte	3 times a second
e. Status byte 128 - 255	1 byte	75 times a second
Bit 0 frame Sync, set for 1 of 25, clear for 2-25 of 25		
Bit 1 green perfusion, set if true only during pulse		
Bit 2 red perfusion, set if true only during pulse		
Bit 3 sensor alarm, set if true		
Bit 4 out of track, set if true		
Bit 5 bad pulse, set if true		
Bit 6 sensor disconnected, set if true		
Bit 7 always set		
Note: bits 1 & 2 are set for yellow perfusion.		
f. Plethysmographic pulse value 0 - 254	1 byte	75 times a second
g. Sync character (01)	1 byte	75 times a second
h. Checksum = byte 1 + byte 2 + byte 3 + byte 4	1 byte	75 times a second
Extended Averaging Data		
i. E-HR value bits 7&8 (128-511), 511 = bad data	1 byte	3 times a second
j. E-HR value bits 0-6 (0-127)	1 byte	3 times a second
k. E-SpO2 value 0 - 100	1 byte	3 times a second
Non-Slew Limited with Standard Averaging		
l. SpO2 Slew value 0 - 100, 127 = bad data	1 byte	3 times a second
Beat to Beat Value (No Averaging or Slew Limiting)		
m. SpO2 B-B value 0 - 100, 127 = bad data	1 byte	3 times a second
Display Data		
SpO2-D Display Value with Standard Averaging		
n. 0-11, 127 = bad data	1 byte	3 times a second
E-SpO2-D Display Value with Extended Averaging		
o. 0-100, 127 = bad data	1 byte	3 times a second
HR-D-MSB Display Value with Standard Averaging		
p. HR Value bits 7&8, 511 = bad data	1 byte	3 times a second
HR-D-LSB Display Value with Standard Averaging		
q. HR Value bits 0-6 (0-127)	1 byte	3 times a second
E-HR-D-MSB Display Value with Extended Averaging		
r. HR Value bits 7&8, 511 = bad data	1 byte	3 times a second
E-HR-D-LSB Display Value with Extended Averaging		
s. HR Value bits 0-6 (0-127)	1 byte	3 times a second



Data would be sent in the following format

Hz	BYTE					Hz	BYTE					Hz	BYTE				
1/75	1	2	3	4	5	1/75	1	2	3	4	5	1/75	1	2	3	4	5
1	01	STATUS	PLETH	HR MSB	CHK	26	01	STATUS	PLETH	HR MSB	CHK	51	01	STATUS	PLETH	HR MSB	CHK
2	01	STATUS	PLETH	HR LSB	CHK	27	01	STATUS	PLETH	HR LSB	CHK	52	01	STATUS	PLETH	HR LSB	CHK
3	01	STATUS	PLETH	SpO2	CHK	28	01	STATUS	PLETH	SpO2	CHK	53	01	STATUS	PLETH	SPO2	CHK
4	01	STATUS	PLETH	REV	CHK	29	01	STATUS	PLETH	REV	CHK	54	01	STATUS	PLETH	REV	CHK
5	01	STATUS	PLETH	*	CHK	30	01	STATUS	PLETH	*	CHK	55	01	STATUS	PLETH	*	CHK
6	01	STATUS	PLETH	*	CHK	31	01	STATUS	PLETH	*	CHK	56	01	STATUS	PLETH	*	CHK
7	01	STATUS	PLETH	*	CHK	32	01	STATUS	PLETH	*	CHK	57	01	STATUS	PLETH	*	CHK
8	01	STATUS	PLETH	*	CHK	33	01	STATUS	PLETH	*	CHK	58	01	STATUS	PLETH	*	CHK
9	01	STATUS	PLETH	***SpO2-D***	CHK	34	01	STATUS	PLETH	***SpO2-D***	CHK	59	01	STATUS	PLETH	***SpO2-D***	CHK
10	01	STATUS	PLETH	***SpO2 Slew***	CHK	35	01	STATUS	PLETH	***SpO2 Slew***	CHK	60	01	STATUS	PLETH	***SpO2 Slew***	CHK
11	01	STATUS	PLETH	***SpO2 B-B***	CHK	36	01	STATUS	PLETH	***SpO2 B-B***	CHK	61	01	STATUS	PLETH	***SpO2 B-B***	CHK
12	01	STATUS	PLETH	*	CHK	37	01	STATUS	PLETH	*	CHK	62	01	STATUS	PLETH	*	CHK
13	01	STATUS	PLETH	*	CHK	38	01	STATUS	PLETH	*	CHK	63	01	STATUS	PLETH	*	CHK
14	01	STATUS	PLETH	***E-HR MSB***	CHK	39	01	STATUS	PLETH	***E-HR MSB***	CHK	64	01	STATUS	PLETH	***E-HR MLSB***	CHK
15	01	STATUS	PLETH	***E-HR LSB***	CHK	40	01	STATUS	PLETH	***E-HR LSB***	CHK	65	01	STATUS	PLETH	***E-HR LSB***	CHK
16	01	STATUS	PLETH	***E-SpO2***	CHK	41	01	STATUS	PLETH	***E-SpO2***	CHK	66	01	STATUS	PLETH	***E-SpO2***	CHK
17	01	STATUS	PLETH	***E-SpO2-D***	CHK	42	01	STATUS	PLETH	***E-SpO2-D***	CHK	67	01	STATUS	PLETH	***E-SpO2-D***	CHK
18	01	STATUS	PLETH	*	CHK	43	01	STATUS	PLETH	*	CHK	68	01	STATUS	PLETH	*	CHK
19	01	STATUS	PLETH	*	CHK	44	01	STATUS	PLETH	*	CHK	69	01	STATUS	PLETH	*	CHK
20	01	STATUS	PLETH	***HR-D-MSB***	CHK	45	01	STATUS	PLETH	***HR-D-MSB***	CHK	70	01	STATUS	PLETH	***HR-D-MSB***	CHK
21	01	STATUS	PLETH	***HR-D-LSB***	CHK	46	01	STATUS	PLETH	***HR-D-LSB***	CHK	71	01	STATUS	PLETH	***HR-D-LSB***	CHK
22	01	STATUS	PLETH	***E-HR-D-MSB***	CHK	47	01	STATUS	PLETH	***E-HR-D-MSB***	CHK	72	01	STATUS	PLETH	***E-HR-D-MSB***	CHK
23	01	STATUS	PLETH	***E-HR-D-LSB***	CHK	48	01	STATUS	PLETH	***E-HR-D-LSB***	CHK	73	01	STATUS	PLETH	***E-HR-D-LSB***	CHK
24	01	STATUS	PLETH	*	CHK	49	01	STATUS	PLETH	*	CHK	74	01	STATUS	PLETH	*	CHK
25	01	STATUS	PLETH	*	CHK	50	01	STATUS	PLETH	*	CHK	75	01	STATUS	PLETH	*	CHK

* Undefined

Appendix 3. Music engine OSC commands

This is a description of the commands that the OSCMusicEngine understands. The first part lists commands that is recognized by the OSCMusicEngine. The second part lists the messages that the OSCMusicEngine sends in return to some of the messages it recognizes.

Commands recognized by the OSCMusicEngine

/getMusicExampleList

Arguments:	None
Description:	This message returns a list of available music examples.
Returns:	/musicExampleList and a variable number of strings as arguments, each with the name of an available music example.

/loadMusicExample, (string) aMusicExampleName

Arguments:	aMusicExampleName (string). The name is obtained by /getMusicExampleList message.
Description:	Starts loading of a music example. The music example becomes “current” with this call, even when it’s not fully loaded yet. All subsequent messages to the music engine refers to the current music example.
Returns:	/musicExampleLoading, (int) aResult, (float) aLoadtime, (string) aMusicalParameter. aResult is 1 if the music example was able to start the loading process and 0 if the loading process failed. If aResult is 0 then it is not followed by any more arguments. If aResult is 1 it is followed by a float giving the estimated remaining load time to fully load the musical example and a variable number of strings, each giving the name of an available musical parameter.

/getMusicalParameters

Arguments:	
Description:	Returns the available musical parameters of the current music example set with /loadMusicExample. The parameters are returned as strings, with one parameter per string.
Returns:	/availableMusicalParameters with a variable number of strings, each string giving the name of an available musical parameter.

/getMusicalParameterInfo, (string) aParameterName

Arguments:	A string holding the name of the parameter to get information about.

Description: This message returns information about one of the available musical parameters. The information return tells if what type of parameter it is, stepped or linear, if stepped how many steps, and finally min- and max values.
Returns: /parameterInfo with type of parameter, number of steps (if applicable) and min and max values.

/useMusicalParameters, (string) aParameterName…
Arguments: A variable number of strings stating which of the available parameters to use.
Description: This message takes a variable number of strings as arguments. Each string must be the name of an available musical parameter and tells the music engine to use it when manipulating the musical intensity.
If this message is not sent to the music engine, the music engine will default to manipulate all the parameters. If on the other hand, an available parameter is not present in the argument list, this parameter will not be used. Allowed values for aParameterName are: “Tempo", "Instrument Set", "Harmonic Complexity", "Rhythmic Complexity", "Register", “Articulation".
Returns: Nothing

/setMusicIntensityLevel, (float) anIntensityLevel
Arguments: (float) anIntensityLevel 0.0 – 1.0.
Description: A value between 0.0 and 1.0 giving the level of music intensity to set. The music engine has the responsibility to interpret this value and set the used musical parameters to values giving a musical intensity level that corresponds to the argument of this message.
Returns: Nothing

/setMusicalParameter, (string) aParameter, (integer) aValue
Arguments: (string) aParameter is the name of the musical parameter to set.
(integer) aValue is the new value of the parameter.
Description: If the parameter is a used parameter, then it is set to the new value.
Returns: Nothing.

startMusicPlayer, (float) anInitialIntensityLevel
Arguments: (float) anInitialIntensityLevel to set, 0.0 to 1.0.
Description: Starts the music player on the musical intensity level given as argument. The value of this argument is between 0.0 and 1.0. See OSC message /setMusicIntensityLevel for details.
Returns: nothing

stopMusicPlayer
Arguments: none
Description: Stops the music playing immediately.
Returns: nothing

Commands sent by the OSCMusicEngine

This is a description of the commands sent by the OSCMusicEngine as return values to some of the commands that the music engine recognizes.

/musicExampleList, (string) aMusicExampleName…
Arguments: a variable number of strings each with the name of an available music example.
Returns from: /getMusicExampleList
Description: Returns the names of the available music examples

/musicExampleLoading, (int) aResult, (float) aLoadtime, (string) aMusicalParameter...
Arguments: aResult is 1 if the music example was able to start the loading process and 0 if the loading process failed. aLoadTime is the estimated remaining loading time. After the load time comes a variable number of strings each giving the name of an available musical parameter.
Description: If aResult is 0 then it is not followed by any more arguments. If aResult is 1 it is followed by a float giving the estimated remaining load time to fully load the musical example and a variable number of strings, each giving the name of an available musical parameter. Possible strings are: “Tempo", "Instrument Set", "Harmonic Complexity", "Rhythmic Complexity", "Register", “Articulation".

/availableMusicalParameters, (string) aParameterName…
Arguments: A variable number of strings giving the names of the available musical parameters.
Returns from: /setMusicExample
Description: This message reports back to the regulating system which musical parameters are available for the music example set as current.

/parameterInfo, (string) aParameterName, (string) aParameterType, (int) aStepCount, (int) aMinValue, (int) aMaxValue
Arguments: (string) aParameterName is the name of the parameter that has the following settings.

(string) aParameterType is “linear” for a continous parameter and “step” for a stepped parameter.
(int) aStepCount. For a linear parameter this argument is always 0. For a stepped parameter this value holds the number of steps of the fader.
(int) aMinValue is the minimum value the parameter can take.
(int) aMaxValue is the maximum value the parameter can take.
Returns from: /getMusicalParameterInfo
Description: This message returns the settings made for an available musical parameter.

/musicExampleLoaded, (string) aMusicExampleName
Arguments: The name of a music examples that has finished loading.
Returns from: /loadMusicExample
Description: This message is sent by the music engine when a music example is finished loading.

Deprecated or discussed but never implemented OSC messages

/getTempoMaxMin
Arguments: none
Description: gets the max and min tempo from the current music example, set with command setMusicExample.
Returns: OSC message “tempoMaxMin” and a list of arguments “maxTempo”, aMaxTempo, “minTempo”, aMinTempo where aMaxTempo and aMinTempo is integer values. If no music example is set, this function returns -1 for both maxTempo and minTempo.

/getNumInstrumentSets
Arguments: none
Returns: OSC message “numInstrumentSets” and an integer value with the number of instrument sets available

/getNumRhythmicComplexityLevels
Arguments: none
Returns: OSC message “numRhythmicComplexityLevels” and an integer value with the number of rhythmic complexity levels.

/getNumHarmonicComplexityLevels
Arguments: none
Returns: OSC message “numHarmonicComplexityLevels” and an integer value with the number of harmonic complexity levels.

/numInstrumentSets
Arguments: an integer value with the number of instrument sets available
Returns from: OSC message getNumInstrumentSets

/tempoMaxMin

Arguments:	a list of arguments “maxTempo”, aMaxTempo, “minTempo”, aMinTempo where aMaxTempo and aMinTempo are integer values.
Returns from:	OSC message getTempoMaxMin

/numRhythmicComplexityLevels

Arguments:	an integer value with the number of rhythmic complexity levels.
Returns from:	OSC message getNumRhythmicComplexityLevels

/numHarmonicComplexityLevels

Arguments:	an integer value with the number of harmonic complexity levels.
Returns from:	OSC message getNumHarmonicComplexityLevels