Master Thesis

in the Master’s Program
Master of Science in Autonomous Systems

Combined Measurement System
for the Evaluation of Multi-Causal Strain at Mobile Workplaces

by
Holger M. Steiner

First Examiner: Prof. Dr. Dietmar Reinert
Second Examiner: Prof. Dr. Norbert Jung

Handed in on: 31st of October 2008
**Declaration**

I hereby declare that this thesis is my own work and effort and that it has not been submitted anywhere for any award. Where other sources of information have been used, they have been acknowledged.

Hiermit versichere ich, die vorliegende Arbeit selbständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Sankt Augustin, 31st of October 2008

Holger Steiner
Abstract

This work addresses the problem of measuring and distinguishing psychological and physical strain in humans by the use of physiological data. The aim of the work is the research, development and evaluation of a measurement system for the acquisition of such data from humans with a focus on employees at their workplaces. Additionally, an approach to automatically distinguish between psychological and physical strain is developed and the feasibility of this approach is evaluated.

In order to develop a physiological measurement system, appropriate physiological parameters and sensors that are suited to acquire these parameters have been researched. The electrocardiogram (ECG), the respiratory effort, the skin conductance response and the muscle activity in the trapezius muscle, which is measured by an electromyogram (EMG), have been selected due to their importance for the measurement of strain.

On the basis of intelligent sensor systems, which have been developed in previous research projects by the author, a combined measurement system for the acquisition of the proposed physiological parameters is designed and implemented. It consists of two small and lightweight sensor modules, which are powered and connected via USB interfaces. In addition, a PC-based software tool is developed for the control of the sensor modules.

By the use of this combined measurement system in conjunction with a measurement system for the assessment of physical strain, which has been developed at the BGIA Institute for Occupational Health and Safety, algorithms for the differentiation between physical and psychological strain are researched. For this purpose, a specifically designed measurement setup is performed by several subjects in order to gain training data for the evaluation of different machine learning algorithms.

Finally, the proposed measurement system is extensively evaluated in terms of the theoretical and technical implementation of the combined sensor system, as well as by a functional evaluation with respect to the practical use of the system.
## Contents

1 **Introduction**  
1.1 Scope and Aim of the Work ........................................... 1  
1.2 Problem Description .................................................. 2  

2 **Context and Basics** ................................................... 3  
2.1 Measurement of Strain ................................................ 3  
  2.1.1 Parameters of the Physical Activity .......................... 3  
  2.1.2 Physiological Parameters ........................................ 4  
  2.1.3 Results of Recent Work ......................................... 6  
2.2 State of the Art ...................................................... 9  
  2.2.1 Measurement Systems ............................................ 9  
  2.2.2 Physiological Sensor Technology .............................. 10  
2.3 The CUELA-System .................................................... 13  
  2.3.1 Description of the “Standard” System ....................... 13  
  2.3.2 The Analysis-Software WIDAAN ............................... 14  
  2.3.3 Recent and Current Development ............................. 15  
    2.3.3.1 CUELA Activity ........................................... 16  
    2.3.3.2 CUELA Digital ............................................ 17  

3 **Problem Analysis** ................................................... 19  
3.1 Division of the Main Problem ...................................... 19  
3.2 Requirements on the Solution ....................................... 20  
3.3 Assumptions ............................................................ 21  

4 **Approach** ............................................................. 23  
4.1 Basic Idea .............................................................. 23  
4.2 Theoretical Details .................................................. 24  
  4.2.1 Signal Processing Chain ....................................... 24  
  4.2.2 System Concept for Hardware and Software ................ 26  
  4.2.3 Sensor System Framework ..................................... 28  
  4.2.4 Details of the Sensors ......................................... 29  
    4.2.4.1 ECG Sensor ............................................... 29  
    4.2.4.2 Respiration Sensor ..................................... 31  
    4.2.4.3 Skin Conductance Sensor ................................. 33  
    4.2.4.4 Blood Oxygen Saturation Sensor ......................... 35  
    4.2.4.5 EMG Sensor ............................................... 38  
  4.2.5 Classification of Data .......................................... 39  
    4.2.5.1 Classification by the Use of Learning Algorithms .... 40  
    4.2.5.2 Generation of Training Data ............................ 42  
4.3 Technical Details ................................................... 45  
  4.3.1 Hardware Setup .................................................. 45  
    4.3.1.1 Selection of Components for the Sensor Systems ...... 45  
    4.3.1.2 The Different Sensor Systems .......................... 48  
    4.3.1.3 The Combined Sensor Module ............................ 56  
    4.3.1.4 Data Storage and Control Device ....................... 60  
  4.3.2 Software Setup .................................................. 63  
    4.3.2.1 Software Setup of the Sensor Systems ................. 63
List of Figures

2.1 Schematic representation of the ECG waves¹ ........................................ 5
2.2 The CUELA System (translated and adapted from [Ell06]) .......................... 14
2.3 Screenshot of the WIDAAN user interface ........................................... 15
2.4 The CUELA Activity System ([Web08]) ............................................. 16
2.5 Combined acceleration and gyroscope sensor of the Activity system ... 17

4.1 Signal processing chain of the sensor systems .................................... 25
4.2 System concept and data flow chart ............................................... 27
4.3 Correct placement of the ECG electrodes .......................................... 30
4.4 Respiratory Effort Transducer ......................................................... 32
4.5 Circuitry of a Wheatstone bridge ¹ .................................................. 32
4.6 Placement of the skin conductance electrodes on the foot ................. 34
4.7 Biopac OXY100C amplifier module and TSD123b oxygen saturation transducer ... 37
4.8 Actual oximeter module of the Biopac OXY100C ................................ 37
4.9 Correct placement of the EMG electrodes on one shoulder ............... 38
4.10 EMG sensor system ................................................................. 39
4.11 Example of a decision tree to decide whether or not to play tennis, adapted from [Wit05] 42
4.12 Screenshot of the Weka workbench, showing the Explorer and the KnowledgeFlow Environment ......................................................... 43
4.13 Comparison of the size of a PSoC in DIP package and a similar PSoC in SSOP package 46
4.14 Input / output pin configuration of the PSoC as specified by the sensor system framework ... 49
4.15 Circuit diagram of the PSoC chip as specified by the sensor system framework .... 49
4.16 Circuit diagram of the ECG sensor system ..................................... 50
4.17 Circuit diagram of the respiration sensor system ............................. 51
4.18 Voltage output of the respiration sensor with respect to the resistance of the transducer ... 53
4.19 Circuit diagram of the skin conductance sensor system ...................... 53
4.20 Voltage output of the skin conductance sensor with respect to the skin conductance value .......................................................... 55
4.21 Circuit diagram of the PIC USB controller ...................................... 57
4.22 Additional circuitry of the sensor modules ....................................... 57
4.23 Layout of the PhysioModule’s circuit board ...................................... 59
4.24 Layout of the OxyModule’s circuit board ......................................... 59
4.25 The completely assembled PhysioModule in its casing ....................... 61
4.26 The completely assembled OxyModule in its casing .......................... 61
4.27 Asus Eee-PC compared to the size of a hand ................................... 62
4.28 UML class diagram of the PSoC firmware framework .......................... 64
4.29 UML class diagram of the ECG sensor module .................................. 66
4.30 UML class diagram of the respiration sensor module ......................... 66
4.31 UML class diagram of the skin conductance sensor module ............... 67
4.32 UML class diagram of the oxy sensor module ................................... 67
4.33 Structure of the USB data packets [Sch07] ...................................... 69
4.34 Structure of the USB information and control packets [Sch07] ............. 69
4.35 Structure of the SPI command packets ......................................... 70
4.36 Structure of the SPI data as encapsulated in the USB data packet of the OxyModule ... 71
4.37 Structure of the SPI data as encapsulated in the USB data packet of the PhysioModule 72
4.38 Class Diagram of the PhysioLogger software ................................... 73
List of Tables

4.1 Strain classes ................................................................. 40
4.2 Test scenarios for the generation of training data .................. 44
4.3 SPI commands and arguments .......................................... 71
1 Introduction

This work addresses the problem of measuring and distinguishing psychological and physical strain in humans by the use of physiological data. It continues the work that has been done in previous research projects by the author (see [Ste07] and [Ste08]) and uses the results that were proposed there to present an approach to solve this problem.

1.1 Scope and Aim of the Work

The aim of this work is the research, development and evaluation of a measurement system for the acquisition of physiological data from humans and the differentiation of psychological and physical strain. The main purpose of this measurement system is to assess both types of strain in employees at their workplaces, as this is not easily possible with currently available measurement systems, but may help to reduce or prevent this strain (this aspect will be explained in detail in the following section). In the previous work of the author, a respective physiological measurement system has been outlined and four microcontroller-based intelligent sensor systems have been developed for the use in this measurement system. In the scope of this work, a way to combine these sensor systems and the physiological data they acquire, as well as data of the physical activity, is researched and implemented. Furthermore, algorithms for the differentiation between physical and psychological strain are examined and the feasibility of this approach is evaluated.

The basic knowledge that is required to understand the problem at hand, as well as the relevant medical and technical aspects of the measurement of strain are summarised in chapter 2 of this work. Additionally, the results of recent work in the field of (psychological) strain measurement are presented and an updated overview over the technical state of the art in this field, as well as the field of sensor technology in specific, is given. The analysis of the problem, which has been presented in the previous work, is improved and complemented in chapter 3. The focus of this analysis is on those parts of the problem, which have not been solved yet.

For the measurement of physical activity and strain, a well-suited measurement system is available that has been developed at the BGIA Institute for Occupational Health and Safety: the “CUELA” system. This system is described as a part of this work (see chapter 2.3) and used for the acquisition of physical data in addition to the physiological data.
The approach for the development of the physiological measurement system is described in detail in chapter 4. Its focus is on the development of hardware and software for the combined measurement system on the basis of the previous work, as well as on the research and implementation of an appropriate learning algorithm that can be used to autonomously analyse and classify the measured data with respect to the differentiation of psychological and physical strain. For this purpose, different scenarios to induce psychological and physical strain in order to gain training data for the learning algorithm have been set up and performed.

The research of the feasibility of an integration of the measured physiological data into the WIDAAN software, which has also been developed at the BGIA and is used for the analysis of the data recorded by the CUELA system (see chapter 2.3), as well as a complete combination of the physical and the physiological measurement systems concludes the proposed approach.

Finally, the implemented measurement system is evaluated in terms of the theoretical and technical implementation of the combined sensor system, as well as a functional evaluation with respect to the practical use of the system.

1.2 Problem Description

The measurement and analysis of strain in employees at different (or even mobile) workplaces is the main problem of this work. Strain can have multiple causes and consequences, which might be a risk for the health of the employees and result in high costs for health insurance companies (see [Ste06]). Researching the strain employees encounter during their work can help to prevent or reduce the strain, for example by modifying the working environment or the tools used during the work, leading to safer working conditions for the employees and reduced costs for the insurance companies.

In this context, different types of strain have to be distinguished, as strain can be physical (that is, the body of the employee is affected) or psychological: high mental workload or an emotionally stressful working environment might lead to psychological problems, such as depression, but can also be the cause for physical problems, such as illness or back pain.

To distinguish between the different types of strain and to classify their magnitude and their influence on the employee, a complex measurement system is necessary, as different parameters for physical and psychological or emotional strain have to be acquired simultaneously. As it is shown in the research of the state of the art in chapter 2.2, there is no convincing “universal” measurement system available yet that can be used at different kinds of workplaces and on different employees without the need to adapt and specialise its sensors and algorithms on the specific task and working environment. Therefore, this work deals with the research and development of a such measurement system.
2 Context and Basics

This part of the work presents the context of this work with respect to the measurement of strain in general, including parameters that can be used to assess the physical activity of a subject, as well as physiological parameters that reflect the overall strain the subject is encountered with. Recent work and studies that have been published in this field are described and their results and drawbacks are presented. Afterwards, the state of the art regarding the technical developments in the fields of strain measurement and physiological sensor systems is described. Furthermore, the CUELA system is described, which serves as a basis for the approach that is presented in the later parts of the work.

2.1 Measurement of Strain

It has already been described in the introduction that there are different kinds and multiple causes of strain, which an employee might experience at work. The two main categories that have to be differentiated in this context are physical strain and psychological strain, which can be further distinguished into mental (or cognitive, respectively) and emotional strain (see [Ste06]). In order to acquire both types of strain in a subject, different physiological and physical parameters have to be measured and monitored simultaneously. Some of the parameters that are described in the following are useful for the monitoring of just one single type of strain, while others can be used to monitor both types of strain, but do not allow differentiating between them. As a result, it is obvious that a combination of such parameters must be used to analyse and evaluate the strain situation of a subject in detail.

2.1.1 Parameters of the Physical Activity

Measuring the physical activity of subjects is an indirect way to assess the strain they encounter, as it does not measure the strain itself, but allows to “predict” or estimate the strain that would “typically” result from the measured activity. However, as important aspects like the personal fitness of the subjects, for example, are not taken into account in these measurements, the results are completely objective and do not reflect the personal perception of the subjects.

Important parameters for the assessment of the physical activity include the speed and frequency of movement (that is, of the arms and legs, as well as the upper body), the weight of possibly existing
“payload”, the body posture and the actual type of activity. All of these parameters can be acquired and assessed by the CUELA system (see [Ell06] and [Her03]) and, with the exception of the payload-evaluation, by the more portable CUELA Activity system (see [Web06] and [Web07]), which are both described in detail in chapter 2.3.

The measurement of the physical activity has been used by several studies for the evaluation of strain. Myrtek repeatedly proposed the use of acceleration sensors as motion detectors and combined them with physiological sensors in order to distinguish physical and psychological strain (compare [Myr96], [Myr99], [Myr01a] and [Myr01b]). Together with others, he developed the “Freiburger Monitoring System”, which implements such a combination of sensors. The system is described in greater detail in the chapter State of the Art (2.2). A similar approach has been used by Zeier et al. (see [Zei01]), who also used a single acceleration sensor to assess the physical activity of a subject in combination with physiological sensors. Another example for the use of acceleration sensors in combination with physiological sensors is the work by Wilhelm et al. ([Wil06]). Although this aspect has not been researched in this work, it seems that acceleration sensors are currently the state of the art for the measurement of physical activity.

2.1.2 Physiological Parameters

In contrast to the parameters of the physical activity, physiological parameters represent a more direct way to measure and evaluate the strain of a subject. In general, these parameters measure the reaction of the human body to both physical and psychological strain of any kind. The results that can be achieved by the assessment of these parameters are always subjective to a certain extend, as the subject’s personal fitness and individual perception of the stressor have an influence on the physiological parameters (compare [Myr01b]). A simple example to explain this is the comparison of the heart rates of an exercised athlete and an untrained person during an uphill bicycle ride: the trained subject will have a much lower heart rate than the untrained subject.

The heart rate, which is more commonly known in the public as any other physiological parameter, is strongly correlated with the overall strain of the body (compare to [Myr01b] or [Rau98], for example) and can be acquired by an electrocardiogram (ECG). An ECG is recorded by the use of electrodes, which are applied directly on the skin of the subject, and measures the electrical activity that occurs during the depolarisation of the heart muscles over the time (see [Gon99]). There are several ways to record an ECG (so-called “leads”) with different numbers and placements of the electrodes, but in a non-medical context, a bipolar lead with two electrodes (plus a third reference-electrode) is the most commonly used method. In the ECG curve, five typical waveforms that are called P-, Q-, R-,
2.1. Measurement of Strain

S- and T-wave can be noticed. A schematic of these waveforms is shown in figure 2.1. It is easy to see that the R-wave is the highest peak in the curve. The R-wave marks the exact moment of the heart beat and can therefore be used to calculate the heart rate by measuring the time interval between two successive R-waves (the so-called “RR-interval”).

Besides the heart rate, there are several additional parameters that are useful for the assessment of strain. The heart rate variability, which can also be calculated by means of the ECG signal, as it represents the variations of the heart rate from beat to beat, can be used to assess mental strain ([Myr01b]). It is typically assessed in terms of the “root mean square of successive differences” (RMSSD) and the standard deviation of the RR-intervals (SDRR), which are both measured in milliseconds. The respiration rate (measured in breathing cycles per minute) and the respiration amplitude are, similar to the heart rate, strongly correlated with the overall strain of the body, but can also serve as an indicator for mental strain (see [Wie92], [Wil01]).

In contrast to this, the skin conductance response allows the detection of emotional strain and arousal (e.g. [Zha06], [Wil06]). The skin conductance (that is, the ability of the skin to conduct electricity) is measured in Siemens, which is the reciprocal of the resistance in Ohm, and depends on the amount of sweat in the sweat glands of the skin. A “spontaneous” filling of the sweat glands can be caused by emotional strain and excitement and influences the skin conductance noticeably, although no sweat might be visible on the skin. A dense population of sweat glands can be found in the region of the palms (that is, the inner sides of the hands), as well as at the soles of the feet (see [Cut07]).

---

1Created by Anthony Atkielski, published under Creative Commons license
The activity of the trapezius muscle, which is situated in the shoulder and neck region of the human body, can be measured by recording an electromyogram (EMG). It is also known to be an indicator for strain, especially when activity is detected during phases without body movements (e.g. [Tae07], [Vog01]). Similar to the ECG, the EMG uses a pair of electrodes that are applied on the skin directly above the muscle that has to be measured. It records the electrical voltages occurring in the muscle over the time.

Finally, the blood’s oxygen saturation might be an indicator for strain as well: in situations that are stressful for the human body, a lower oxygen saturation of the blood might be noticeable. As no references are available for the use of the blood oxygen saturation in this context, the question whether or not this parameter can be useful for the assessment of strain is part of the research of this work and is addressed in the evaluation (see chapter 5).

2.1.3 Results of Recent Work

In the recent years, a number of studies were published that researched the psychological strain of subjects in an occupational environment. This section presents the results of these studies in chronological order, beginning with the oldest of the studies, which was published in the year 1996.

In 1996, a research group from the University of Freiburg, Germany, around Myrtek investigated the subjective stress and objective strain of 50 female students during their everyday university life ([Myr96]). They recorded an ECG lead together with the physical activity, which was acquired by two motion detectors, and analysed the heart rate, heart rate variability and the ST-segment amplitude “online” (that is, during the running measurement) in order to reduce the amount of data by storing only one mean value for each minute and to be able to request feedback from the subjects. This was done by displaying predefined questions regarding the subject’s current situation and feeling on the display of the used device that had to be answered by pressing a button. The complete system weighed 2.5kg and recorded 23 hours of data. The results of the study were the following: the heart rate variability showed clear differences for studies and leisure time and therefore indicates the amount of mental strain that the subjects were encountered with. Additionally, it was shown that the heart rate of “chronically stressed” students was significantly higher than that of non-stressed students. The opposite applies to the height of the ST-segment amplitudes: they were lower for the stressed students.

Rau published a paper in 1998, which reports the result of two studies on job analysis and strain evaluation ([Rau98]). The first study was performed on 50 male operators, who were employed in the electric utility industry, and analysed the heart rate and the blood pressure of the subjects during work. In addition, a pocket computer was used to present questions to the subjects concerning the
subjective strain during the measurement. The results of this study showed a higher blood pressure and higher heart rate for the more stressed (according to the self-reported strain) employees. In the second study, the effect of strain during work on the nightly sleep was investigated. It was shown that the blood pressure was still elevated during the sleep period.

In 1999, Myrtek et al. published a new study, in which they investigated the strain of 29 blue and 57 white collar workers ([Myr99]). In this context, blue collar workers refers to physically working persons, while white collar workers refers to office-workers, for example, who have a higher mental than physical workload. Except for the ST-segment amplitude, which is also assessed by means of the ECG signal, they used the same measurement set up as before ([Myr96]). Although the total heart rate was higher for the blue collar workers due to more physical activity, they found no clear indication for higher emotional or mental strain in one of the groups.

Two years later, in 2001, Myrtek and Foerster described a new method of emotional strain measurement: the “additional heart rate” ([Myr01a]). The additional heart rate expresses the increase of the heart rate that is not caused by physical activity and is assumed to indicate emotional strain. Together with Brügner, they published a book with the title “Freiburger Monitoring System” in the same year, which proposed a measurement system that is based on this method and explained its use in detail ([Myr01b]). Furthermore, they presented the results of several studies that had been performed with the system. The results showed that the additional heart rate is a good indicator of emotional strain.

In the same year, Vrijkotte et al. published a study that investigated work-related stress by measuring the ambulatory blood pressure, the heart rate and the heart rate variability over 24 hours ([Vri01]). Their conclusion was that all these parameters produced reliable information about the physiological responses to the strain that the subjects experienced during work. Additionally, they proposed the heart rate reactivity as an indicator to differentiate between high and low amounts of chronic work stress.

Wilson also published a paper in 2001, which describes an in-flight psychophysiological monitoring on pilots [Wil01]. From the previously explained physiological measures, the ECG, respiration, skin conductance response and EMG were used in this study. Additionally, electrooculography (EOG, assesses the movement and blink frequency of the eye) and electroencephalography (EEG, measures the electrical activity of the brain) were acquired. According to Wilson, ECG and EOG produced the best results in this study, but the skin conductance, which was highly correlated with the heart rate, also proved to be a good indicator of stress.

A more common situation was investigated by the study of Zeier et al. in the same year: both heart rate and physical activity were monitored for 24 hours in 14 male students. It showed that the
heart rate was slightly increased during an academic test situation and even significantly increased during a public speaking situation.

In 2002, Vogt and Kastner published the results of a study that was performed in 1996. This study investigated the strain of air traffic controllers at Düsseldorf Airport during their working day. Coincidentally, on that day a fire was caused at the airport due to which all arriving air traffic had to be diverted to Cologne airport. This exceptional situation caused additional stress and strain in the traffic controllers, who were monitored by means of blood pressure, ECG, respiration and skin conductance. The results showed a high increase in blood pressure, heart rate and respiration rate after the fire alarm. The skin conductance response also was slightly increased.

In 2004, an “emotion recognition system” was developed by Kim et al., who used the skin temperature variation, the skin conductance response, the heart rate and the heart rate variability to assess emotions. The respective emotions were induced in a total of 50 subjects, who were children in the age from seven to eight years, by a specified emotion induction protocol that used audio, visual and cognitive elements such as storytelling and sad background music. Their recognition system was able to achieve a correct classification rate of 62 to 78 percent, depending on the number of categories used.

Wilhelm et al. performed a study, which investigated the ability to distinguish between emotional and physical strain in 2006 ([Wil06]). Besides the physical activity, the ECG, the skin conductance response, respiration and skin temperature were acquired from 28 subjects during a resting phase, physical activity and a short commercial flight. Their conclusion was that although heart rate, respiration and skin conductance are affected by both physical activity and emotional strain, they still provide information that helps to distinguish them, as long as additional “context awareness” is given in the form of data acquired by acceleration sensors or self-reports of the subjects.

Another work was published in 2006 by Zhai and Barreto, who investigated the stress detection in computer users by monitoring the skin conductance response, the blood volume pulse, the pupil diameter and the skin temperature. Their work did not aim on differentiating physical from psychological strain, but on the detection of mental stress alone. They used a computerised test to induce mental strain in a total of 32 subjects. The results were analysed by the use of three different learning algorithms (provided by the Weka software, see section 4.3.3 in the chapter Technical Details): Naive Bayes Classifiers, Decision Tree Classifiers and Support Vector Machines. Their results showed that Support Vector Machines produce the best results in predicting stress, followed closely by the Decision Tree Classifiers, while the results of the Naive Bayes Classifiers were significantly worse.

In 2007, Hwang et al. published a study about the mental workload of operators in a nuclear power plant ([Hwa07]). They measured the heart rate, the blood pressure, the heart rate variability and the
2.2 State of the Art

In this chapter, the state of the art in the field of strain measurement systems and physiological sensor systems is presented and discussed.

2.2.1 Measurement Systems

In the previous chapter, the work by Myrtek and his “Freiburger Monitoring System” (FMS) ([Myr01b]) has already been mentioned. Myrtek and his research group from the University of Freiburg have a long experience with ambulatory monitoring, as their first study in this field of research was published in 1982. This new measurement system has been the result of their experience. It acquires the physical activity of the subjects by the use of a three-axis acceleration sensor applied to the chest and a second one-axis acceleration sensor applied on one thigh, as well as the heart rate and the heart rate variability by the use of two ECG leads. Myrtek et al. specified a standard protocol to evaluate the current activity of the subject. This protocol is able to differentiate between lying, sitting, standing, walking, climbing stairs and bicycling. It is used to predict the increase of the heart rate on the basis of the classified activity. Given this predicted increase in heart rate and the actually measured increase, the proposed parameter “additional heart rate” can be calculated as the difference between them. According to Myrtek, this new parameter is able to express the emotional strain of the subject. In addition to this, the FMS assesses the heart rate variability as a measure of mental strain.

The system consists of a standard data recorder for physiological signals, which is connected to a handheld computer. This handheld computer is running a software developed by the Freiburg research group, which is used to request feedback from the subjects during the measurement: predefined questions are presented to the subjects every 10 to 20 minutes, which have to be answered using the computer. The questions are meant to acquire background knowledge about the respective subject’s feelings, environment or activity and are requested each time a significant increase in emotional strain is detected by the use of the additional heart rate. In order to avoid a sensitisation of the subjects,
an additional “random” feedback is requested after 20 minutes without an “emotional event”. The acquired data is analysed by a specifically developed software package on a regular personal computer.

Although there are several additional studies that used the combination of physical and physiological sensors, none of them developed an own sensor system or analysis tool - therefore, the Freiburger Monitoring System is still the only example of a combined measurement system. However, there are some approaches for physical activity measurement systems that could be combined with a physiological sensor system in order to extend them to combined measurement systems. The first example for this is the work by Tanaka et al. ([Tan04]), who presented a portable device for the monitoring of human posture and walking velocity. This device is, similar to the system proposed by Myrtek, based on acceleration sensors that are applied to the chest, thigh and on the shank, but uses an additional gyroscope, which is also positioned on the thigh. This system is able to differentiate between lying, sitting, standing and walking, but also allows measuring the walking velocity of the subject.

The “CUELA Activity” system, which was developed by Weber ([Web06], [Web07] and [Web08]), is also based on acceleration sensors and gyroscopes. In contrast to the previously described measurement systems, the “Activity” system provides a combined three-axis acceleration- and gyroscope sensor for both legs, one arm, the thoracic spine and the lumbar spine and allows to assess both the body posture and the movements made by the subject without the need to specify a fixed activity protocol. Therefore, the “CUELA Activity” system can be considered to be the current state of the art in the field of physical activity measurement systems. It is described in detail in chapter 2.3, together with its “predecessor”, the “CUELA Classic” system, which is based on mechanical sensors.

2.2.2 Physiological Sensor Technology

Besides the state of the art for complete measurement systems, the state of the art in the field of physiological sensor technology is of interest for this work. It has to be researched in order to combine the most advanced and most promising sensors to develop a new physiological measurement system. Therefore, a short overview over the recent work in this field is given in the following.

ECG Measurement

According to the work of Jovanov et al. ([Jov99]), which was published in 1999, the use of intelligent “Holter monitors” for the monitoring of the ECG during every day activities has become a common standard procedure since many years (the oldest reference mentioned in the work dates back to 1986). These devices are able to analyse the ECG in real time, for example by detecting
the QRS-complex in order to calculate the heart rate. Typically, such Holter monitors are based on digital signal processing (DSP) microcontrollers.

In addition to the hardware itself, the software algorithms for the Holter monitors have been developed further: in 1997, for example, Ruha et al. demonstrated a microcontroller-based device, which is specifically designed for the measurement of the heart rate variability and achieves a timing accuracy of 1ms by the use of a complex QRS-detection algorithm ([Ruh97]). A different QRS-detection algorithm, yet similar in terms of accuracy, has been proposed by Meissimilly et al. in the year 2003 ([Mei03]).

A relatively new approach concerning the hardware setup of an ECG measurement system has been shown by Luprano et al. in 2006 ([Lup06]): they developed a shirt with integrated textile electrodes for the acquisition of the ECG without the need for additional electrodes. The acquired ECG signals are analysed in real time by an on-body signal processing device and transmitted over a “body sensor network” to a data storage device. This device is capable of further transmitting the data by the use of a Bluetooth interface to a connected mobile phone, which forwards the data to a specific Internet portal. Additionally, the proposed body sensor network allows to add an acceleration sensor for the (very basic) measurement of physical activity to the system.

Despite the fact that new approaches have been presented recently, it can be concluded that microcontroller-based devices, which acquire the ECG by standardised leads and which analyse the ECG signal in real-time by the use of advanced QRS-detection algorithms, are still the state of the art in the field of ECG measurement, as they are most frequently used: nearly all of the studies mentioned in section 2.1.3 use similar ECG measurement systems (for example [Hwa07], [Myr01b], [Vog01]).

**Skin Conductance Measurement**

For the measurement of the skin conductance response (or electrodermal activity (EDA), respectively), similar (often the same) devices are used as for the measurement of the ECG (see [Vog01], [Wil06]). Theoretically, the same kind of electrodes can be used as well, although there are specific EDA-electrodes available, which have an increased contact area and are provided with a special contact paste (see [Cut07]).

From the electronic side of view, developing a skin conductance measurement system is comparably easy (see [McC06]). Nevertheless, there are some more advanced approaches, for example the one by Strauss et al.: this group developed a very small skin conductance sensor which does not store the data on the device, but can be connected via a wireless Bluetooth connection to a personal computer ([Str05]). It is based on a microcontroller that controls the amplification circuitry and can
adapt the gain factor of the amplifier in order to maximise the quantisation resolution. The complete device is smaller than the 9V block battery that is used to power it and that provides a runtime of approximately 10 hours, which is enough to cover a complete working day. Because of its small size, the device can be attached to the arm of the subject by the use of a small wrist band. This allows to keep the wiring to the electrodes, which are attached either to the subject’s fingers or palm, short and does avoids hindering the subjects in their movement.

**EMG Measurement**

Holter devices similar to the ones used for the ECG measurement and standard electrodes are also used for the EMG measurement (see [Cut07]). However, the applied filtering algorithms are somewhat different, as there is nothing like a QRS-complex in the EMG signal that has to be detected. Usually, an RMS (root mean square) value of the measured muscle activity is calculated after the measurement (see [Kel06]).

In their work from 2007, Taelman et al. presented a “biofeedback shirt” with textile EMG sensors embedded in the shirt ([Tae07]). This approach is to a large extend similar to the approach by Luprano et al., who developed a similar shirt for the measurement of the ECG.

**Respiration Measurement**

For the measurement of the respiration rate (and in a limited way, the respiration amplitude), chest straps with integrated strain gauges, which increase their resistance linear with the chest cavity expansion, are commonly used (see [Cut07], [Vog01]). If they are placed appropriately, they allow to detect both thoracic and abdominal respiration. It is also possible to combine two such chest straps in order to differentiate between thoracic and abdominal respiration.

The changing resistance of the strain gauges can be converted into a voltage, which can be digitalised by a microcontroller, for example, by the use of a Wheatstone bridge (see [Cut07]). A Wheatstone bridge is used to compare an unknown resistance with a reference resistance. If the reference resistance is properly selected and the Wheatstone bridge has been calibrated correctly, it also allows to measure a changing resistance. Further details of its workings, as well as a respective circuitry is described in section 4.2.4.2 of chapter Theoretical Details.

**Oxygen Saturation Measurement**

To measure the oxygen saturation of the blood in a non-invasive way (that is, without penetrating the skin), optical sensor systems are used (see [Cut07], [Bar02]). This principle is called “pulse oximetry”. It is explained in detail in section 4.2.4.4. Typically, the used optical sensors are controlled by
intelligent, microcontroller-based systems, which calculate the oxygen saturation based on the sensor readings with respect to the detected pulse rate.

According to Barker and Shah, as well as to Cutmore and James, the pulse oximeter technology is very susceptible to motion ([Bar96], [Cut07]), although there are examples of respective measurement systems that seem to be able to solve this issue up to a certain extend by applying advanced filtering techniques.

2.3 The CUELA-System

The CUELA system is a measurement system for the assessment of physical strain of employees at their workplaces. It was developed at the BGIA and has been used in a large number of applications since the year 1997 (see [Her03]). CUELA is a German abbreviation and stands for “Computer-unterstützte Erfassung und Langzeit-Analyse von Belastungen des Muskel-Skelett-Systems”, which can be translated as “computer-assisted measurement and long-term analysis of strain on the musculoskeletal system”. Today, there are different variants of the CUELA system, which are all based on the original “standard” system, but use different sensors and assess different types of strain. The variants and current developments, which are interesting for this work, are presented in the following sections. However, until now there is no variant of the CUELA system that is able to measure or evaluate psychological strain.

The data that is measured by the system is recorded on a so-called “datalogger” and analysed at a later time with a software called WIDAAN (which stands for “Winkel-Daten-Analyse”, or “angle data analysis”, respectively). WIDAAN runs on Windows computers and allows to display the recorded data, as well as to perform calculations on it. The software is described in detail in section 2.3.2.

2.3.1 Description of the “Standard” System

The standard CUELA system is a sensor suite that can be worn over the working clothes. It is able to measure physical strain of a subject on the basis of the body posture and movements. It uses mechanical sensors to measure the angles of the thoracic and lumbar spine, the torsion of the back, as well as the angles of the hip and knee joints. Additionally, it is able to measure pressure on the feet by the use of special measurement soles, which allows to evaluate the weight of any load the subject might be carrying (see [Her03]). In figure 2.2, the components of the CUELA system are described.

As shown in the figure, the system is based on a centralised digitalization of the analogue sensor readings and data storage on a “CUELA Datalogger”. Every measurement is simultaneously recorded on video for reasons of documentation and in order to provide appropriate context data for the
measurement. By the use of the WIDAAN software, the recorded video and the sensor readings can be synchronised subsequently. This way, if any alarming or otherwise interesting sensor data is found, the information about the subject’s exact activity and environment is available in form of the video recording.

An additional extension of the CUELA system is available that further allows to assess the angles and movements of the arms, shoulders and the head (see [Ell06]). This extension can also be used independently from the rest of the measurement system as a different variant of it. For the assessment of sedentary work, for example, the pressure on the feet or the angles of the legs are usually not of interest. Therefore, the respective sensors could be omitted. Furthermore, “force handles” have been developed in the BGIA, which allow to measure the forces that are caused by pushing or pulling of objects ([Ell06]). These handles have, for example, been used in conjunction with the evaluation of the strain of stewards during flights, who have to push and pull heavy trolleys through the plane.

Another example of an extension to the standard CUELA system is the EMG sensor system: a sensor box, to which up to four pairs of electrodes can be connected, filters and amplifies the EMG signal and uses the standard CUELA datalogger to store the signal simultaneously with the rest of the CUELA data.

2.3.2 The Analysis-Software WIDAAN

The WIDAAN software is used to display the recorded data in the form of graphs over time, as well as in the form of an animated 3D-skeleton figure. WIDAAN is also able to calculate information about the movement of the subject on the basis of the recorded (angular) data by integrating and combining it. It handles every sensor or calculation result as a “channel”, which can be viewed as a chart or graph over the time, and allows to synchronize, mark and edit single or multiple data.
channels at once ([Her03], [Ell06]). By the use of markers, specific elements of the recorded data can be used for further calculations: when using the EMG sensors, for example, markers can be set for a short time interval with no activity (low EMG) and a second interval with very high activity (high EMG) in order to “normalise” the recorded data.

The design of the WIDAAN software allows to extend it with new data channels or new “plug-in” components for the calculation of strain data easily. Figure 2.3 shows the graphical user interface of WIDAAN with the lists of sensor channels and applied interval markers on the left, the 3D-skeleton and the synchronised video recording on the right, as well as the graphical representation of the currently selected sensor channel at the bottom.

2.3.3 Recent and Current Development

In the last two years, a modified version of the CUELA system, called CUELA Activity, has been developed that is strongly reduced in the terms of size, weight and complexity by using different sensors. Besides this, a completely new basis for the standard CUELA system is currently developed. This new approach is called CUELA Digital, as it will replace the analogue sensors that are used up to now in all versions of the CUELA system by digital sensor systems. In the following, both systems are described shortly.
2.3.3.1 CUELA Activity

On the basis of the standard CUELA system, a modified measurement system called “CUELA Activity” has been developed in the last two years by Weber (see [Web06], [Web07] and [Web08]). It uses accelerometers and gyroscopes to measure the actual body posture and movement instead of the mechanical sensors of the standard CUELA system, thus reducing the weight, size and complexity of the sensor suite significantly. The main purpose of the Activity system is to evaluate the actual activity of a subject in terms of both the type of the performed activity and the speed of movement. The sensor data is stored on the same datalogger which is used in the standard CUELA system and the analysis algorithms have been integrated into the WIDAAN software as a plug-in. Among others, these algorithms provide the functionality to calculate a physical activity intensity (PAI) index, which can be used to objectively evaluate the physical activity of a subject, as well as an activity detection, which recognises the type of activity that is performed (e.g. climbing stairs or bicycling). Figure 2.4 shows a schematic of the CUELA Activity system.

Similar to the standard CUELA system, all sensor data is stored on a datalogger and transferred to a computer after the measurement has been completed, where it can be synchronised with a video recording. In total, the Activity system uses seven sensor modules, which are connected to the datalogger by the use of a specially developed sensor box that combines (multiplexes, respectively) the
2.3. The CUELA-System

Figure 2.5: Combined acceleration and gyroscope sensor of the Activity system

sensor readings. Each of these sensor modules consists of a 3D accelerometer (that is, the acceleration sensor can detect movement on three axes) and a gyroscope. The sensors are attached to the human body by the use of elastic bands, which can be worn over or under the clothes. Figure 2.5 shows one of the Activity sensors, as well as the elastic band of the arm-sensor.

2.3.3.2 CUELA Digital

A new version of the CUELA measurement system is currently developed (see [Sch07] and [Sch08b]), which uses intelligent sensor units that are connected to a new type of datalogger by digital universal serial bus (USB) interfaces. These intelligent sensor units are meant to replace the previously used analogue sensors. Additionally, as there is no need for centralised digitalisation anymore, the new “datalogger” can be any device that is equipped with an USB root hub and running the necessary software, for example a mobile phone or a handheld computer (PDA). The new system will therefore be called “CUELA Digital”.

CUELA Digital allows to connect different kinds of sensors to it, including the analogue and mechanical sensors of the standard CUELA system, as “adapters” are provided for this purpose. However, the idea about this new system is to focus on the use of acceleration and gyroscope sensors ([Sch08b]), as it was shown by the CUELA Activity system that this kind of sensors are able to provide similar results while being less hindering for the subjects. Nevertheless, this new approach is very interesting especially for this work, as it makes the integration of new digital sensor systems into the CUELA system easier.

The selection of a PDA, mobile phone or similar computer-like device for the use as datalogger allows to perform a real-time analysis of the data and to show the currently recorded data on the device’s display during the measurement. This way, it is also possible for the measurement system to give direct feedback to the personnel supervising the measurement or to the subject.
3 Problem Analysis

This part of the work describes and analyses the problem that this work is dealing with. In chapter 3.1, the work’s main problem is divided into several subproblems. These subproblems can be solved individually in order to solve the main problem as a whole. The following chapter 3.2 describes the requirements for a successful solution of the individual subproblems, as well as for the main problem. Finally, chapter 3.3 specifies assumptions that have been made for the proposed solution.

3.1 Division of the Main Problem

As it has already been described in the previous chapters, the acquisition of physiological data from employees working at different kinds of workplaces and the use of this data to evaluate the type and amount of the strain that the employees experience is the main problem of this work. In previous work (see [Ste08]) it was described that this main problem consists of a number of subproblems, which can be solved individually, although they build up on each other. These subproblems are:

i. Identification of useful physiological data (parameters, respectively) for the detection of strain, including the analysis and evaluation of suited sensors and transducers.

ii. Reliable acquisition of the physiological data. This requires also the selection or development of sensors, sensor systems and associated circuitry, which can be used for this purpose.

iii. Combination of the different sensors in one measurement system to acquire all of the proposed physiological parameters simultaneously.

iv. Analysis of the respective physiological signals and extraction of the parameters of interest.

v. Storage and pre-analysis of the acquired data during the measurement.

Except for the last one, all of these subproblems have already been addressed by previous work (compare [Ste06], [Ste07] and [Ste08]). However, as a respective combined measurement system has not been implemented and no detailed measurements have been performed so far, none of these subproblems can be considered to be finally solved. In addition, four new subproblems have to be specified for this work:

vi. Combination and synchronisation of simultaneously measured physiological and physical data.
vii. Automatic analysis of the combined physical and physiological data using clustering and ma-
machine learning algorithms.

viii. Well-defined generation of strain in a subject in order to gain training data for the used learning
algorithm.

ix. Integration of the resulting analysis algorithm into the WIDAAN analysis software.

For the solution of these new subproblems, the five previously presented problems have to be solved in
advance, as the new subproblems require a reliable physiological measurement system to be available.
Therefore, the implementation and evaluation of that measurement system is still the most important
aspect about this work, although it addresses the new subproblems as well.

3.2 Requirements on the Solution

The proposed measurement system has to be bound to the subjects without hindering them in any way
in their work, as otherwise their behaviour might differ from their regular (working day) behaviour.
This includes movements and body postures as well as the subjective experience of psychological
strain. Such an unusual behaviour might falsify the measurement, as situations that are usually not
stressful for the subjects in any way, might be wrongly classified as stressing due to the fact that the
subjects are hindered or distracted by the measurement system.

On the one hand, this means that the measurement system has to be “portable” and “wireless” by
using small and lightweight components, which can be carried around easily, and avoid the need of
being connected by cables to any external components. On the other hand, the system has to be
easy to use in order to keep the amount of personnel needed to set up and operate the system low,
as any personnel supervising the measurement might influence the subjects’ behaviour. It also has
to work autonomously to avoid the need for any “user-interactions” that might disturb the subjects
in the same way.

By using both physiological and physical sensors, the combined measurement system must be able to
evaluate and classify the subjects’ activity sufficiently exact. This classification is required in order
to distinguish between physical and psychological strain and has to be either subject-independent or
easily adjustable to the actual subject.

Apart from these functional requirements, there are also technical requirements, which have to be
mentioned: for the development of the measurement system, standardised hardware-components and
interfaces should be used in order to simplify both the maintenance and future enhancements of the
system.
3.3 Assumptions

The following assumptions were made for the development of the measurement system:

- All physical activity is detected and evaluated appropriately by the CUELA Activity system, as this work is meant to provide an extension to the current CUELA system. Although only the Activity system is used in this work, the developed physiological measurement system can also be used in combination with the standard CUELA system, as the acquired physiological parameters are independent from the actual type of the physical parameters.

- In later implementations of the measurement system, a PDA or similar device will be used as “datalogger” (that is, the data recording device that stores the data acquired by the sensor systems on non-volatile memory) for both the CUELA Activity and the new physiological sensors. For this work, a temporary solution for the physiological sensor systems is used, as this new datalogger is not available yet.

- The measurement system (the physiological sensors in particular) is powered by a battery to ensure the personal safety of the subject. No electrical connection is established between parts of the measurement system and other (e.g. mains powered) devices.
4 Approach

This part of the work describes the approach that has been developed in order to solve the problem which has been described and analysed in the previous chapters. First, the basic idea of the approach is described. Afterwards, the theoretical and the technical details of the approach are presented. Finally, the details of the software implementation are described and discussed.

4.1 Basic Idea

As described in chapter 2.1, there are several parameters that are of interest for the measurement of both physiological and physical strain. The physical activity can be acquired and analysed in detail by the use of the CUELA-system in its different variants. Therefore, the CUELA-system is a very good basis for the development of the proposed multi-causal strain measurement system. The implementation of microcontroller-based intelligent sensor systems for the physiological parameters, which can not be acquired by the CUELA-system, and the integration of these sensor systems into the CUELA-system is the basic idea of the approach of this work.

The sensor systems have already been developed in previous work (see [Ste07] and [Ste08]). The idea about them was the following: they should be able to calculate all the required parameters directly and without the need for an additional device in order to keep the complexity of the system low and to be able to exchange and re-combine the sensor systems if required by the specific application. Furthermore, the hard- and software design process was simplified by setting up a general sensor system framework consisting of similar hardware components and software modules that was used for all of the different sensor systems. This framework was implemented and extended with specific hardware components and software functions for the particular sensor system. Additionally, the development of hardware (electrical circuitry in particular) was avoided wherever possible by the use of “system-on-a-chip” microcontrollers, which allowed to implement both digital and analogue signal processing modules and circuits by software and as a part of the microcontroller chip. To make sure that components or devices can be exchanged easily, the use of standardised hardware components, communication protocols and software modules was preferred. By simplifying the design process, both time and costs of the development were reduced.
In this work, the developed sensor systems are optimised and combined in an easy to use physiological sensor module consisting of a printed circuit board, which contains the complete hardware of all the sensor systems. This sensor module is mounted in a small and lightweight, but robust casing with external power supply to enable a wide range of applications. The data acquired by the individual sensor systems is gathered and transmitted digitally and in the form of data packets to one centralised data storage and measurement control device. Simultaneously, but independent from the physiological data, the physical activity is recorded by using the CUELA-system.

The analysis of the recorded data is done “offline”: both the physical and the physiological data is imported into the CUELA analysis software WIDAAN. Using this software, the data of each sensor can be represented as a single data channel and visualised by a graph over time. Furthermore, sequences of data can be marked and labelled. Finally, a classification algorithm is developed that allows classifying and evaluating the data over time: the type of strain and its impact on the subject can be calculated.

In order to set up the required categories and decision rules, a suited machine learning algorithm is researched and used. Test scenarios that comply with the requirements of the measurement system for its use in practice are designed and performed by several subjects to gain training data that can be used by the learning algorithm.

### 4.2 Theoretical Details

This chapter describes the theoretical details of the proposed approach. First, the signal processing chain of the physiological sensor systems is described, followed by an explanation of the concept for the complete measurement and analysis system with respect to hardware, software and the flow of data. Afterwards, the sensor system framework is presented, which leads on to a description of the used sensors themselves. The chapter is concluded by a section about the classification of the acquired data by the use of learning algorithms.

#### 4.2.1 Signal Processing Chain

The signal processing chain, which is depicted in figure 4.1, is almost identical for all of the sensor systems. The steps of the chain differ only in the details and in their implementation. The signal processing chain always begins with the acquisition of the physiological signal directly on the subject. In the second step, the analogue signal is “conditioned”, which means that it is amplified and filtered in order to remove noise and to improve the signal-to-noise ratio. The signal-to-noise ratio describes the relationship between the strength of the useful signal and the strength of the noise. A higher ratio
improves the “readability” of the signal: it can be digitalised and analysed more accurately without getting erroneous results. Another important aspect about the filtering process and the digitalisation of the signal is the bandwidth limitation: according to the Nyquist-Shannon sampling theorem, the sampling rate, which is applied during the digitalisation, has to be more than twice as high as the highest frequency occurring in the signal (which is called the Nyquist frequency, see [Smi99]). If this theorem is broken, disturbing aliasing effects are the consequence. These unwanted effects can only be avoided if the signal is bandwidth limited before it is digitalised. However, this is not completely possible in practice, as there are always frequencies above the Nyquist frequency included in the signal. These signal components might be a part of the useful signal, but can also be caused by noise. To limit the unwanted aliasing effects to an amount that suffices for the given application, these signal components have to be sufficiently attenuated. This can, for example, be done by applying a low pass filter on the signal that provides a sufficiently strong attenuation of the signal’s amplitude with respect to the (increasing) frequency: a simple first-order filter, which can be realised easily by using just one resistor and one capacitor, achieves a rolloff of 6dB per octave (that is, it reduces the amplitude by half each time the frequency doubles, see [Tie99]) and a more complex filter of the n-th order achieves a rolloff of n*6dB per octave, respectively. As the attenuation, which is needed to achieve a sufficiently bandwidth limited signal, is depending on the actual signal, as well as on the characteristic and amount of noise, this will be further evaluated individually for each sensor system in chapter 4.3.
The digitalisation of the signal is the next step in the signal processing chain. As mentioned before, the digitalisation has to be performed by using a sampling rate which is adapted to the bandwidth limitation of the signal. If necessary for the respective sensor, the resulting digital signal is filtered again (digital signal conditioning) before the signal analysis is performed. The signal analysis is done by an algorithm that detects specific attributes of the signal, depending on the particular physiological parameter. Based on these attributes, the physiological parameters, which are used for the evaluation of the strain that the subject is encountered with, are calculated in the next step.

After the calculation of the physiological parameters, the resulting values are either transmitted directly over a serial interface or encapsulated in data packets that are transmitted over a serial peripheral bus interface (SPI) at a regular time interval. The SPI bus, which has originally been developed by Motorola, has been selected as it is a very popular interface for microcontrollers (see [Kal02]). It can be seen in figure 4.1 that most of the steps of the signal processing chain up to this point are realised “inside” of one single integrated circuit (IC) chip, starting from parts of the analogue signal conditioning up to the transmission of the resulting data. These steps are marked by a yellow background in the figure. As already mentioned in the section Basic Ideas, this is achieved by using “system-on-a-chip” microcontrollers.

If the data is transmitted via the SPI bus interface, the signal processing chain has two additional steps, as a bridging device is used that passes the SPI data packets on to a data storage device via an additional universal serial bus (USB) interface (see [Axe05] for further details on the USB interface). These two steps are not performed separately for each sensor system, but for the combined data. This is explained in more detail in the next section, which describes the system concept.

### 4.2.2 System Concept for Hardware and Software

While the concept for the physiological sensor systems can be deduced easily from the signal processing chain that has been presented in the previous section, the overall system concept regarding the hard- and software is more complex and therefore is described in this section in greater detail. The starting points for this description are the physiological sensor systems. On the hardware side, they consist of the actual sensor for the acquisition of the physiological signal, electrical circuitry and components for the analogue signal conditioning, as well as one microcontroller IC-chip. On the software side, several functions are implemented on the microcontroller to deal with all following signal processing steps that are applied on the digitalised signal, as shown in figure 4.1.

For the complete measurement system, all of the physiological sensor systems are combined, together with data of the physical activity. The concept for this combined measurement system is shown in figure 4.2.
Figure 4.2: System concept and data flow chart

The figure shows all components of the measurement system and the flow of data between them. Hardware components are coloured in yellow, software components in green and the data flow is shown in blue. The components are arranged in the order of the data flow, starting with the physiological sensor systems in the upper left corner. It can be seen that the physiological sensor systems are combined on one hardware module (represented by the area marked in light yellow), together with an USB interface controller. The sensor systems transmit their data to the USB controller by using a SPI bus interface. The USB controller unites the data and transmits it over its USB bus interface to a host device, which stores the data and controls the measurement process by sending control commands back to the physiological sensor systems (via the USB controller). This host device, which is called “datalogger” in conjunction with the CUELA system, can be any personal computer or notebook, as well as a personal digital assistant (PDA) or handheld computer, or even a mobile phone with USB host capabilities and sufficient memory to store the entire measurement data. Therefore, only a software tool that is able to manage the storage of the acquired data and that allows a user to control the measurement system has to be developed in order to implement this component of the system.

* Recorded simultaneously by the CUELA system
From this point on, the data is stored and available for “offline” use, which means that it can be viewed and analysed after the measurement has been completed. All further components of the system work with this stored data. They are implemented completely in software, which runs on typical computer hardware like a desktop PC or a notebook. The basis for the software implementation is the WIDAAN analysis software of the CUELA system. As described in chapter 2.3, it allows viewing, synchronising, editing and analysing of the data by using a graphical user interface. Furthermore, it can be extended with additional software components in the form of “plug-ins”, which makes it perfectly suited for the use in this work. The stored physiological data is imported into WIDAAN as a new data channel and every sensor system is shown as a sensor device with its own graphical representation. In the same way, the data of the physical activity, which has to be recorded simultaneously and in the same time interval with the physiological data, is also imported into WIDAAN. By the use of time stamps and markers in the data stream, both data channels are synchronised. This synchronisation process is necessary to build a set of data elements containing all available data for one single point in time. WIDAAN allows to set synchronisation markers in the data streams and automatically calculates the offset and the scale factor of each channel. When all channels are synchronised correctly, arbitrary time intervals containing the data of all sensors can be selected and marked with a description. This allows to specify time intervals of resting phases (for example at the beginning of a measurement procedure) and of phases with maximum strain. The values of these intervals can then be used to normalise the data.

Afterwards, a decision tree algorithm is used to classify the data in the course of time with respect to the type and amount of strain that the subject experienced. A new data channel is created in WIDAAN and the results of the algorithm’s calculations are stored and graphically represented in this channel. In order to develop appropriate decision rules for the classification of the strain data, a learning algorithm is used on suited learning data. The resulting strain classification method is the final step of the system concept and represents the result of the approach that is proposed in this work.

4.2.3 Sensor System Framework

The framework for the sensor systems provides a common basis for the development of all the individual sensor system units. It makes use of the fact that the signal processing steps are almost identical for all sensor systems (as described in section 4.2.1 and shown in figure 4.1). This way, the development process can be simplified and it is possible to re-use hard- and software designs and implementations of “general” functions that are needed for every single sensor system.

The framework is based on a programmable system-on-a-chip microcontroller, which allows to avoid
4.2. Theoretical Details

many steps of the hardware development process by providing functions that usually require a certain hardware component (e.g. the amplification of an analogue signal by an operational amplifier) directly inside of the controller. These functions can be set up and configured with the help of a special integrated development environment (IDE). Therefore, the hardware design can be kept relatively simple and in a wide range identical for all sensor systems, as only sensor-specific components and some parameters have to be added or adapted. These common hardware components are a part of the framework as well and are described in section 4.3.1 as part of the Technical Details.

Besides the hardware design, a basic software implementation is the second and even more important part of the sensor system framework. Functions that are needed by every sensor system unit, independent from the actual sensor it is using, are provided and can be used with a minimum of necessary modifications and effort. Additionally, the framework provides a clear structure for the implementations of the sensor-specific functions. This structure makes sure that the contents of the framework are not mixed up with the individual functionality and that changes in either one part do not affect the other: if changes are made in the framework-functionality of one sensor system, for example, they can easily be copied into the implementations of all other sensor systems without any modifications. The structure is explained in detail in the section Implementation Details (4.4.1).

4.2.4 Details of the Sensors

In the following sections, the details of the sensors that are used in this work are presented. As these sensors have been explained in detail in previous work already (see [Ste08]), this is done only briefly. An exception to this is the EMG sensor, which has not been integrated into the measurement system until now.

4.2.4.1 ECG Sensor

The ECG sensor consists of a pair of electrodes which are attached to the skin of the subject. According to the literature (e.g. [Myr01a]), silver/silver chloride (Ag/AgCl) electrodes are the best choice for this purpose and therefore, they are used in this work. They acquire the small voltages that are produced by the heart muscles during contraction and relaxation (see section 2.1.2 of the chapter Context and Basics). Two electrodes are sufficient for this work, as it focuses on the detection of the R-waves and a complete medical analysis of the ECG is not necessary. The first electrode is placed in the middle of the chest and the second electrode is placed beneath the left armpit. This way, the electrodes “surround” the subject’s heart similar to a precordial lead and the best possible ECG signal strength is achieved. In the very rare case that a subject’s heart is situated on the right side
of the body ("Dextrocardia"), the second electrode has to be placed beneath the right armpit instead of the left one. Figure 4.3 shows an example of correctly placed electrodes.

The two electrodes produce a small differential signal that is amplified by an instrumentation amplifier to make it more "readable". This type of amplifiers is best suited for differential signals, as it only amplifies the voltage difference between the two signal lines and rejects any voltage that is common to them (common mode rejection). This is a great advantage, as the useful signal (that is, the voltage difference) is usually much smaller than the absolute voltage of each signal line (that is, offset to ground). Additionally, as noise that might be caused by irradiations, for example, influences both inputs of the amplifier similarly, it is rejected by the amplifier, too. This effect can be increased, if identical electrodes and electrode leads are used in order to achieve similar physical properties for both inputs (compare to [Cut07]).

To remove noise that might disturb or even blanket the comparatively weak ECG signal, the signal has to be filtered before it can be used further. The filtering process usually consists of multiple steps: first, very low frequencies are removed (high pass filtering) to stabilise the "baseline" of the signal. The baseline is the zero-level of the undistorted signal and should always be steady, but due to low-frequent noise it might drift up and down if no high pass filter is applied. Second, high-frequent noise that might be caused by irradiation, for example by mobile phones, as well as other electrical components on the same circuit board or with an electrical connection to it, for example switching noise generated by the used microcontroller, is removed by applying a low pass filter. This filter also limits the bandwidth of the signal, which is necessary to avoid disturbing aliasing effects during
4.2. Theoretical Details

the digitalisation of the signal in a later step as described in section 4.2.1. Additionally, it might be necessary to use a notch filter that removes noise with a certain frequency, which can not be removed by one of the other filters, because it is inside of the useful frequency band of the signal. This is the case for noise caused by irradiations from main power lines. This noise has a frequency of 50Hz (in Europe; in some other countries, e.g. the USA, the frequency is 60Hz) and is easily adopted by the human body, if any main power lines are near to it - and therefore, this noise is nearly always included in the ECG readings.

The physiological parameters that are calculated on the basis of the ECG signal are the Heart Rate (HR), which is measured in beats per minute, and the Heart Rate Variability (expressed by the “root mean square of successive differences” (RMSSD) and the standard deviation (SDRR), see 2.1.2), which is measured in milliseconds. To calculate these parameters, the time intervals between every two successive R-waves, which are detected by a specialised peak detection algorithm, are used in conjunction with the following equations:

\[
HR_i = \frac{60000}{RR_i} \quad \text{with} \quad RR_i = \text{time between two successive R-waves in ms} \quad (4.1)
\]

\[
RMSSD = \sqrt{\frac{\sum_{i=2}^{N} (RR_i - RR_{i-1})^2}{(N - 1)}} \quad \text{with} \quad N = \text{number of averaged samples} \quad (4.2)
\]

\[
SDRR = \sqrt{\frac{1}{N - 1} \sum_{i=1}^{N} (RR_i - \overline{RR})^2} \quad \text{with} \quad \overline{RR} = \frac{1}{N} \sum_{i=1}^{N} RR_i \quad (4.3)
\]

4.2.4.2 Respiration Sensor

In general, two different kinds of sensors are available for the measurement of the respiration rate and effort (see section 2.2.2): masks enclosing mouth and nose that measure the amount and frequency of air breathed in and out on the one side and chest straps that measure the raises and falls of the subject’s chest (or stomach, respectively) during breathing on the other side. The former would be a huge handicap for the subject if he is supposed to work and to behave as usual during the measurement process. Therefore, the latter kind is the better choice for this work. These chest straps are equipped with strain gauges and work as resistive transducers that change their resistance when their length is increased and vice versa. They can be used to acquire either thoracic or abdominal breathing and, by using a combination of two straps, both can be acquired simultaneously as well. In figure 4.4, the “Respiratory Effort Transducer” that is used in this work is presented.
As the resistance change of the transducer can not be used as an input signal for the sensor system directly, it has to be converted into a measurable voltage first. This is done with the help of a Wheatstone bridge. It is used to measure the electrical resistance of an unknown resistor by the voltage difference between two sides of a bridge circuit (see [Nor12]), consisting of three fixed resistors and one variable resistor (the one that has to be measured), which are arranged as shown on figure 4.5. The voltage that can be measured between the sides of the bridge, which are powered by a constant voltage supply, changes depending on the value of the variable resistor: it is 0V if the resistance on both sides is identical and increases with the difference of the resistance values. If the values of the fixed resistors are chosen appropriately and the Wheatstone bridge has been properly calibrated, the output of the Wheatstone bridge covers the full input voltage range of the sensor system with the minimum voltage being applied when the transducer is completely relaxed and the maximum voltage being applied when the transducer is stretched to its maximum length.

The following equation describes the behaviour of the Wheatstone bridge ($U_0$ is the constant supply voltage):

$$U_W = U_0 \frac{R_1 \cdot R_4 - R_2 \cdot R_3}{(R_1 + R_2) \cdot (R_3 + R_4)}$$ (4.4)

---

1Created by Horst Sauer, published under a Creative Commons License
If the resistors R2 and R4 are chosen identically, the equation can be further simplified:

$$U_W = U_0 \cdot \frac{(R_1 - R_3) \cdot R}{(R_1 + R) \cdot (R_3 + R)}$$

(4.5)

The differential output voltage $U_M$ is fed into an instrumentation amplifier in order to decouple the impedance of the Wheatstone bridge from the sensor system and to transform the differential voltage into an absolute (ground-related) voltage signal. To remove possible noise from the signal, a simple resistor-capacitor (RC) low pass filter is used. It is configured with a cut-off frequency of about 10Hz, as the useful respiration signal does not have any frequency components with a frequency higher than 10Hz: the highest breathing rate that can be expected amounts to approximately two to three cycles per second. As the shape of the signal is very similar to a sine, the highest important frequency that is included in the signal is in the same dimension.

For the description of the respiratory effort, the two parameters Respiration Rate (RR) and Respiration Amplitude (RA) are used. The respiration rate is measured in breathing cycles per minute and can be compared with a respiration rate acquired during a different measurement and/or measured by any other sensor system. In contrast to this, the respiration amplitude is measured as the percentage of the sensor range between the minimal and the maximal extension of the transducer during the last breathing cycle. It depends strongly on the specific sensor, as well as on the position and setting of the respiratory effort transducer. Therefore, it is very important for the later analysis of the recorded data that the respiration amplitude is “normalised” by calculating the offset of the minimum amplitude (with respect to the value measured for the completely relaxed transducer) and the scale between the minimum and the maximum amplitude occurring in the acquired data. Similar to the ECG sensor, a peak detector algorithm is used to measure the duration of each breathing cycle by detecting the exact time of the maximum extension of the transducer. Afterwards, the following equations are used to calculate the parameters:

$$RR = \frac{60000}{\Delta T} \quad \text{with} \quad \Delta T = \text{duration of one respiration cycle in ms}$$

(4.6)

$$RA = \frac{100 \cdot (MAX_i - MIN_i)}{\text{RANGE}}$$

(4.7)

### 4.2.4.3 Skin Conductance Sensor

The basis for the skin conductance sensor is a pair of electrodes that are attached to the skin of the subject at a position that is well suited for this kind of measurement. As described in section 2.1.2 of the chapter Context and Basics, the best positions for the electrode placement are the palms of
the hands or the soles of the feet - from a general point of view, both positions can be used for this work. Depending on the environment and the performed activity, the one that does least hinder the subject can be chosen. As the placement at the foot is more comfortable and less hindering than the placement at the hands in most situations, it is used during the evaluation measurements of this work and it is taken as an example in the following description. The respective placement of the electrodes is shown in figure 4.6.

Silver / silver chloride (Ag/AgCl) electrodes with an increased contact area and isotonic wet gel are used for the measurement of the skin conductance (Electrodermal Activity (EDA), respectively). A low voltage is applied on the two electrodes and the resistance between these two electrodes is measured. As the resistance is the reciprocal of the conductance, it can easily be converted. Again, a Wheatstone bridge is used in order to acquire the resistance with a circuitry similar to the respiration sensor, which was described in the previous section. The main difference is that the supply voltage of the Wheatstone bridge has to be limited to a maximum of 0.5V, because the conductivity of the human skin is only (almost) linear up to this voltage (see [McC06]). Additionally, voltages applied to the human skin should always be as low as possible for safety reasons. Therefore, an appropriate voltage divider has to be added to the circuitry to reduce the (typically much higher) supply voltage of the sensor system accordingly. The differential output signal of the Wheatstone bridge is amplified by an instrumentation amplifier and filtered with a low pass filter afterwards to remove possible high frequent noise. As the skin conductance is a very slowly changing signal, the cut-off frequency of the
low-pass filter can be set equally low: 0.5Hz proved to be a good value for this sensor.

The parameters that are extracted out of the skin conductance signal are the Skin Conductance Level (SCL), which simply describes the current level of the measured skin conductance (typically averaged over a certain time interval, for example over the last ten seconds, by using a sliding window averaging algorithm), and the Skin Conductance Response (SCR), which is calculated by deriving the skin conductance level with respect to time. The skin conductance response expresses all changes in the signal independent from the current conductance level and therefore allows to detect even short “events” that might, for example, be caused by some kind of emotional arousal of the subject. The two parameters are calculated and averaged over the last N skin conductance values (SCV) by the following equations:

\[ SCL = \frac{\sum_{i=1}^{N} (SCV)}{N} \] (4.8)

\[ SCR = \frac{\sum_{i=1}^{N} (SCV')}{N} \quad \text{with} \quad SCV' = \frac{dSCV}{dt} \] (4.9)

### 4.2.4.4 Blood Oxygen Saturation Sensor

The oxygen saturation of the blood can be measured in different ways, as described in the chapter Context and Basics. For this work, the pulse oximetry with an optical transducer is used, because it is a non-invasive method. The advantage of this method is that the sensor is comparably easy to apply and does not hinder the subject in his mobility. The optical transducer consists of two parts: a photodiode is located on the one side of the transducer, measuring the amount of light that is emitted by two LEDs with different, specially chosen wavelengths, which are located opposite the photodiode on the other side of the transducer. The first LEDs wavelength is situated in the range of red light, while the wavelength of the second LED is situated in the range of infrared light (see [Bar02]).

The transducer, which typically has the form of the letter Y, is attached to the subject’s fingers, toes, earlobes or another (thin) part of the subject’s body with the two parts of the transducer facing each other through the skin. This variant is called “transmission method”, because the amount of light that is transmitted through the skin is measured by the photodiode. A second variant, called “reflection method”, can alternatively be used that allows for different placement positions. In contrast to the first variant, here the photodiode and the LEDs are attached side by side, for example on the forehead, and the amount of light that is reflected by the skin is measured by the photodiode.
On the light’s way through the skin of the subject, some of it is absorbed. The absorption characteristics for the red and the infrared light are different for different blood oxygen saturations ([Bar02]): infrared light is absorbed stronger than red light by oxygenated haemoglobin, which is a protein in the blood cells that is responsible for the transport of oxygen, while the red light is absorbed stronger by deoxygenated haemoglobin. Therefore, the ratio between the absorbed red and infrared light can be used to calculate the blood’s current oxygen saturation.

In modern pulse oximeters, the optical transducer is actively controlled by a digital signal processor (DSP) and a large amount of associated circuitry. Besides the control of the LEDs and the analysis of the photodiodes readings, this DSP also calculates the parameters of interest, namely the percentage of the blood’s oxygen saturation. As the complexity of the development of a complete pulse oximeter would have been too high to be included in this work, it was decided to buy and use one of the pulse oximeter systems available on the market for this purpose: the OXY100C sensor and amplifier module from Biopac Systems in combination with the Biopac TSD123b saturation transducer, which are both shown in figure 4.7. The OXY100C was disassembled in order to find a way of reducing its size and to integrate it into the physiological measurement system. It was found that the control of the transducer and the calculation of the oxygen saturation is performed by a comparably small module consisting of two stacked circuit boards, which communicates digitally with the rest of the amplifier’s components. The module is shown in figure 4.8 and has a size of 8.5 x 7.5 x 2 centimetres (length x width x height). As the rest of the components are needed to convert the digital signal to an analogue voltage only, they are not necessary for this work and can be left aside. The digital output of the oximeter module is fed directly into the sensor system’s microcontroller.

Similar to the skin conductance sensor, this sensor system’s parameters of interest are the Oxygen Saturation Level (OSL) and the Oxygen Saturation Response (OSR). The oxygen saturation level represents the value of the oxygen saturation calculated by the oximeter module, averaged over a certain time interval by the use of a sliding window averaging algorithm. The oxygen saturation response is calculated by deriving the saturation level with respect to the time and emphasises changes of the blood’s oxygen saturation. In the following, the equations used to calculate the OSL and the OSR over the last N measurements of the current saturation value (SV) are shown:

\[
OSL = \frac{\sum_{i=1}^{N}(SV)}{(N)}
\]

(4.10)

\[
OSR = \frac{dOSL}{(dt)}
\]

(4.11)
Figure 4.7: Biopac OXY100C amplifier module and TSD123b oxygen saturation transducer

Figure 4.8: Actual oximeter module of the Biopac OXY100C
4.2.4.5 EMG Sensor

Activity in the trapezius muscle, which is situated in the shoulder and neck region, is measured by an EMG sensor. This sensor consists of five silver/silver-chloride (Ag/AgCl) electrodes that are attached to the subject’s shoulders (two electrodes on each side) and neck (one electrode centred on the neck as reference/ground electrode). This way, two EMG derivations are recorded simultaneously, one for each shoulder. Depending on the type of activity that is performed by the subject, the number of electrodes can be reduced to three by recording just one single shoulder side. For many activities, it can be expected that the muscle activity is identical in both shoulders, so that this reduced setup is sufficient. Figure 4.9 shows the correct placement of the EMG electrodes on one shoulder.

Each of the electrode pairs used for a single EMG derivation creates a differential signal, similar to the signal generated by the ECG electrodes, caused by the electrical activity in the muscles (see section 2.1.2 in chapter Context and Basics for a more detailed description of the electromyography). The voltages occurring in the EMG signal are very low (less than 1mV, see [Eic97]). Therefore, the signal is easily distorted by irradiations and noise, making it absolutely necessary to filter the signal by well-designed high-pass and low-pass filters. In contrast to the ECG sensor, which focuses on the detection of the highest peaks in the signal only, even small voltage changes have to be detected accurately during the EMG measurements, as especially the phases of low muscle activity are interesting for this work: if, for example, a subject waves his arms, high peaks are visible in the

![Correct placement of the EMG electrodes on one shoulder](image-url)
4.2. Theoretical Details

For this work, an EMG sensor system manufactured by BioMed Jena is used that is already available in the BGIA and that has proven to be sufficiently accurate for the measurement of trapezius muscle activity (for example in the work by Keller [Kel06]). The device is shown in figure 4.10. It has to be connected to one of the CUELA dataloggers in order to record the measured signal for a later analysis. As the physiological sensor system is used together with the CUELA Activity system in most cases, it is possible to use just one single CUELA datalogger for both the EMG and the activity data. In contrast to all other sensor systems that are used in this work, the EMG sensor systems stores the complete signal instead of calculating the parameters of interest directly. This step is done afterwards by using the WIDAAN analysis software.

4.2.5 Classification of Data

The data that is recorded by the physiological sensor systems and, simultaneously, by the CUELA Activity system has to be analysed as a whole in order to evaluate the type and amount of strain that the subject experienced during the measurement. This can be done manually by viewing and
evaluating the data in numbers or as a graph, but this approach is much too time-consuming to be performed for long time measurements or for a series of measurements with different subjects. Therefore, an automatic and autonomously working algorithm has to be developed that is able to perform this task in less time. In the following, a classification by the use of a suited algorithm is explained and, consecutively, the generation process of the necessary training data is described.

### 4.2.5.1 Classification by the Use of Learning Algorithms

In order to classify the data, a sufficient number of appropriate and discrete classes (or categories, respectively) have to be specified for the data at first. Every data set consisting of the data of each sensor system at one point in time must be analysed and classified with respect to the specified classes. The idea about this approach is that each class represents a well-defined strain situation. The two outer limits for the set of appropriate classes are relatively obvious: they have to be “high physical strain without any psychological strain” and the exact opposite “high psychological strain without any physical strain”. Additionally, a class representing a resting phase with neither physical nor psychological strain has to be included. However, some additional classes, representing the situations with a combination of physical and psychological strain, are required that are not obvious and that have to be specified appropriately. The number of these classes must be sufficiently high to allow a detailed analysis of the strain that the subject encountered during the measurement, but also sufficiently low to make a reliable and fast classification of the data possible. To suffice both requirements, a reasonable compromise had to be found. For this work, a (limited) set of classes was specified, which is presented in table 4.1.

<table>
<thead>
<tr>
<th>Class</th>
<th>Physical Strain</th>
<th>Psychological Strain</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>2</td>
<td>No</td>
<td>Low</td>
</tr>
<tr>
<td>3</td>
<td>No</td>
<td>High</td>
</tr>
<tr>
<td>4</td>
<td>Low</td>
<td>No</td>
</tr>
<tr>
<td>5</td>
<td>Low</td>
<td>Low</td>
</tr>
<tr>
<td>6</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>7</td>
<td>High</td>
<td>No</td>
</tr>
<tr>
<td>8</td>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td>9</td>
<td>High</td>
<td>High</td>
</tr>
</tbody>
</table>

*Table 4.1: Strain classes*
As it can be seen, the amount of strain is distinguished in three very rough steps within this set of classes: no strain, low strain or high strain. This is done due to the fact that every additional separating step increases the number of classes exponentially: let the number of classes be \( N_C \) and the number of steps for the amount of strain be \( N_S \), then \( N_C = 2^{N_S} + 1 \). Choosing three steps therefore seemed to be the highest reasonable number for a first test of the approach. The results of the evaluation of the measurement system will show if this set of classes proves to be a good compromise between the level of detail of the analysis and the reliability of the classification or if it has to be redesigned to meet the requirements.

Besides the specification of the classes, another important aspect about the classification of data is the type and structure of the data itself. The data that the classification algorithm has to deal with is characterised as follows: every single data item consists of a fixed set of attribute-value pairs, in which the attributes represent the individually acquired physiological or physical parameters, each having an associated numerical and (for most of the parameters) integer value. Additionally, the data may contain errors or missing values due to incorrect sensor readings, as well as errors or loss of data during the data transmission from the sensor systems to the data storage device.

Each of the data sets can be classified into exactly one of the previously defined (disjunct) categories. This means, that the algorithm has a discrete output function. Nevertheless, finding a correct output function for the classification algorithm is very complex, as there are up to ten attribute-value pairs, depending on the number of sensors used, each having a wide range of possible values (but always less than 256, as the parameters are reduced to 8-bit precision before transmission). Due to these reasons, it is almost impossible to set up an appropriate classification algorithm for the data manually. Therefore, a machine learning algorithm is used instead.

A decision tree learning algorithm promises to be the best choice for this application, as the characterisation of the data and the type of the required output function perfectly meets the requirements for this kind of learning algorithm (compare to Mitchell [Mit97], chapter 3.3, and to Russel and Norvig [Rus04], chapter 18.3). In decision tree learning, the data is split up by a number of simple decisions that together form a tree: for every data set, decision rules are applied along one path along the edges of the tree until a leave, which represents one of the specified classes, is reached. An example for a decision tree is shown in figure 4.11.

An appropriate decision tree for a given problem can be learned comparably easy by recording typical training data for every needed class and the resulting decision rules can be implemented by a set of simple if-then-else blocks. Therefore, it is (in the following step) possible to implement these rules directly in the analysis software WIDAAN, for example in the form of a plug-in. To achieve the best possible results with this approach, a state of the art learning algorithm is applied by using
Figure 4.11: Example of a decision tree to decide whether or not to play tennis, adapted from [Wit05]

the Weka workbench, which is shown in figure 4.12. Weka, which stands for “Waikato Environment for Knowledge Analysis”, is a software tool, which is written in Java and has been developed by the University of Waikato in New Zealand. It is published free of charge under the terms of a GNU General Public License. Weka is proposed by Witten and Frank and allows to make use of a number of state of the art learning algorithms ([Wit05]). With the use of Weka, a “knowledge flow” can be configured, in which, for example, data is read from a file, automatically classified by selectable algorithms and attributes, visualised and written to a second file together with the results of the classification. Weka allows to experiment with different algorithms in order to find the one that best fits the individual problem. Additionally, Weka provides an interface that allows to integrate its routines in own software code.

4.2.5.2 Generation of Training Data

In order to use a learning algorithm to derive appropriate decision rules that can be applied successfully on every data item, it is absolute necessary to have a sufficient amount of characteristic training data for every of the previously specified classes. To gain respective training data, at least one “test-scenario” for each class is needed. Such a scenario has to be designed in a way that the intended type and amount of strain of the respective class is induced (or “generated”, respectively) in the subject. For the generation of physical strain, the design of an appropriate scenario is relatively easy, as this type of strain can be induced by most kinds of sports. By the use of treadmills or ergometers, the induction of physical strain is possible even in laboratory environments.

In contrast to this, the generation of psychological strain is much more complex. Though mental or cognitive strain can be induced successfully by mathematical or logical exercises, for example, the important component of emotional strain can not be induced easily. Therefore, most of the work in this field was done by using “real-life” data: Myrtek et al. [Myr96], for example, investigated the
strain of female students during their typical day at the university and combined the measurements with a questionnaire to classify his data. Another interesting, yet expensive, approach is the one by Wilhelm et al. [Wil06]: he recorded data on a number of subjects during different scenarios, including a short commercial flight, and assumed that the flight phase would induce a high amount of emotional strain. However, it is obvious that these measures can not be used for this work. A more practical approach is the one of Kim et al. [Kim04], who induced emotions in children aged from seven to eight years in a laboratory environment by using audio, visual and cognitive stimuli, for example storytelling in combination with suited background music and illumination. The drawback of this approach is that it is very likely to fail on adults.

As a result, this work focuses on the induction of cognitive strain. By using dual task scenarios, which are proposed by Wickens et al. [Wic98], for example, the workload and the induced strain can be increased to an amount that most of the subjects will experience as very “stressful”. The idea about the dual task scenario is that the subjects will try to perform well on both tasks simultaneously, but sooner or later they will fail as the workload increases. This way, at least a certain amount of emotional strain can be induced additionally to the cognitive strain.

For the induction of this kind of psychological strain, a specially developed software called “Stress-Generator” is used in this work. The software combines mathematical exercises, which have been used in several studies, for example by Myrtek et al. [Myr01b], and a “Stroop Test” in an adapted
version, which is proposed by Zhai and Barreto [Zha06]. In the Stroop test, the subject has to name the colour of the letters of a written word, which designates a different colour. For example, the word “yellow”, written in red letters, is shown to the subject. The correct answer in this case would be “red”, even though the word reads “yellow”. Zhai and Barreto proposed a software-based interactive version of this test, which they called “Paced Stroop Test”. In this version, the subject has to click on a button with the correct answer in the software’s user interface instead of stating the answer verbally. To intensify the strain induction, the subject is forced to answer within a limited amount of time. The software that is developed in this work allows to choose if the mathematical test, the paced Stroop test or both should be used. Additionally, the difficulty of the mathematical exercises and the amount of time available for the answer can be chosen: with the “easy” setting, it will produce only simple exercises dealing with additions and subtractions, while it will also produce more complicated exercises dealing with divisions and multiplications with the “difficult” setting.

By the use of these tools, a number of test-scenarios (one for each of the specified strain classes) were designed. They are described in table 4.2 and performed in the order of the class numbers by each subject. To ensure that a sufficient amount of data is recorded for every class, each scenario has to be performed for five to ten minutes at least.

<table>
<thead>
<tr>
<th>Covered Class</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 - no strain</td>
<td>Resting phase, also used to determine baseline values for reference. At least ten minutes.</td>
</tr>
<tr>
<td>2 - low psychological strain only</td>
<td>Combination of simple mathematical and colour exercises with a fair solving time of 15 seconds.</td>
</tr>
<tr>
<td>3 - high psychological strain only</td>
<td>Combination of difficult mathematical and colour exercises with a stronger limited solving time of 10 seconds.</td>
</tr>
<tr>
<td>4 - low physical strain only</td>
<td>Cycling with a low workload of approx. 50 Watts.</td>
</tr>
<tr>
<td>5 - low physical and low psychological strain</td>
<td>Cycling with low workload and solving of simple mathematical and colour exercises, 15s solving time.</td>
</tr>
<tr>
<td>6 - low physical and high psychological strain</td>
<td>Cycling with low workload and solving of difficult mathematical and colour exercises, 10s solving time.</td>
</tr>
<tr>
<td>7 - high physical only</td>
<td>Cycling with higher workload of approx. 150 Watts.</td>
</tr>
<tr>
<td>8 - high physical and low psychological strain</td>
<td>Cycling with high workload and solving of simple mathematical and colour exercises, 15s solving time.</td>
</tr>
<tr>
<td>9 - high physical and high psychological strain</td>
<td>Cycling with high workload and solving of difficult mathematical and colour exercises, 10s solving time.</td>
</tr>
</tbody>
</table>

Table 4.2: Test scenarios for the generation of training data
4.3 Technical Details

The technical details of the proposed approach are described in this chapter. On the basis of the previously presented signal processing chain and the system concept for the hardware and software, the following aspects of the approach are explained: the setup of the hardware and of the different software components, as well as the research process of the data classification rules.

4.3.1 Hardware Setup

This section covers the selection and implementation of the required hardware components for each of the different sensor systems, as well as for the combined physiological sensor module. Additionally, the (preliminary) hardware for the measurement control and data storage device is described.

4.3.1.1 Selection of Components for the Sensor Systems

Although it is one of the main ideas of this approach to avoid the use of hardware development by implementing the respective functionality in software wherever possible, there is still a number of hardware components needed for the implementation of the measurement system, in particular for the several sensor system units. During the development of the sensor system prototypes in the previous work ([Ste07] and [Ste08]), only components in dual inline packages (DIP) and wired components were used, as handling and soldering components in through-hole-technology (THT) is easier compared to surface mounted device (SMD) components. Nevertheless, it was made sure that all of the components are also available in SMD technology, for example in small outline (SOT) or shrinked small outline packages (SSOP), as the wired components should be replaced by the (much smaller) SMD components in this work. Figure 4.13 shows the DIP and the SSOP version of the same microcontroller to allow a comparison of the actual differences in size.

The basis for each of the sensor systems is a PSoC (Programmable System on a Chip) microcontroller by Cypress Semiconductors. More specifically, the latest generation of the MSC series, the CY8C29x66, is used, because it provides more freely configurable user modules than any other available microcontroller: 12 analogue modules and 16 digital modules can be configured and used simultaneously. Additionally, the CY8C29x66 provides a higher amount of flash memory (32kB) and supports more kinds of user modules than any other PSoC microcontroller does. This can be easily seen from the overview of the different PSoC’s capabilities, which is shown in an application note by Basinger [Bas07]. Therefore, by choosing this controller for the implementation of the sensor systems, the highest possible flexibility for current and future development is achieved.
As shown in the signal processing chain (see section 4.2.1 in the chapter *Theoretical Details*), the instrumentation amplifier is a very important component for nearly all of the sensor systems (the only exception is the blood oxygen saturation sensor), because it is used to amplify the differential input signal before the digitalisation. The PSoC provides an instrumentation amplifier module, which could be used for this purpose. Unfortunately, its noise characteristics are very bad when compared to state of the art stand-alone instrumentation amplifiers. Therefore, a number of instrumentation amplifiers that were specifically designed for applications like this (medical instrumentation or similar) were compared in previous work (see [Ste07], chapter 4.4) and the Analogue Devices AD623 instrumentation amplifier was found to be best suited. It features a typical common mode rejection rate of 100dB for a gain factor of G=10 and even more than 110dB for gain factors above G=100 (see [AD99]), as well as an input voltage noise of typically $U_n = 35nV/\sqrt{Hz}$. This is much better than the characteristics of the PSoC’s instrumentation amplifier module, which has a typical common mode rejection rate of 60dB at a gain factor of G=48 and a noise level of $U_n = 175nV/\sqrt{Hz}$. This appreciable difference might be explained by the fact that the AD623 was specifically designed for applications which require very good noise characteristics, but which do not need a high performance, while the PSoC’s instrumentation amplifier module was designed as a general purpose amplifier. This theory is supported by the fact that, in contrast to the noise characteristics, the performance values of the AD623 are much worse than those of the PSoC module: the PSoC module’s slew rate is much higher ($6.0V/\mu S$ compared to $0.3V/\mu S$) and the settling time much slower ($3\mu S$ compared to $20\mu S$) than that of the AD623. However, for the application of medical instrumentation, high performance rates are not necessary, as the signals’ frequencies are comparably low.
The most accurate analogue-to-digital converter module available in the selected PSoC microcontroller features a nominal resolution of $N = 14$ bit, which would require a signal-to-noise ratio (SNR) of approximately 86dB for the instrumentation amplifier, according to equation (4.12), which is stated in the following (see [Tie99]):

$$SNR_{dB} = N \cdot 6dB + 1,8dB$$

(4.12)

For the common mode voltages, which would also be noticed as noise if they were amplified and digitalised together with the useful signal, this requirement is met by the AD623, as long as it is used with a gain factor of at least $G=10$ ([AD99]). For the signal-to-noise ratio of the actual input signal, the ECG is used as an example, as the input voltages of the ECG signal are by far the smallest of all signals analysed by any of the proposed sensor systems. The input voltage noise of the AD623 is in inverse proportion with the frequency of the noise: for a frequency of 1Hz, which can be assumed to be close to the lower limit of the ECG signals frequency range, the AD623 produces typically $U_n = 35nV$ of noise. If the peak amplitude of the ECG signal, which is the only feature of the signal that is used in this work, is assumed to be $U_S = 1mV$ (compare to [Gon99]), the resulting SNR according to equation (4.13), which says that

$$SNR_{dB} = 20 \cdot \log \left( \frac{U_{signal}}{U_{noise}} \right),$$

(4.13)

is approximately 89dB, which also meets the requirements for a quantisation with 14-bit accuracy. This requirement would not nearly have been met by the PSoC’s instrumentation amplifier module. However, it is important to notice that this estimation is only valid for the highest peaks (that is, the R-waves) of the ECG signal. This is adequate in this case, as the R-waves are the only parts of the signal that have to be detected properly and therefore, other features of the ECG signal with lower amplitudes can be left out in this evaluation.

The next components after the instrumentation amplifier (with respect to the signal processing chain) are resistors and capacitors for the implementation of the high and low pass filters. For this purpose, standardised parts with low tolerance (that is, with high accuracy) are used. The same applies to the capacitors that are used to stabilise the supply voltage of the amplifier and the PSoC chip. Two capacitors are placed besides each supply pin of the chips, implementing low pass filters without additional resistors, resulting in cut off frequencies of about $\approx 2MHz$ and $\approx 20kHz$. This way, switching noise, which is caused by the controllers with frequencies within this range due to the oscillator and clock frequencies, is reduced.

The last component that has to be mentioned here is the USB controller, which is needed for the communication with the measurement control device. A Microchip PiC18F4550 controller is used in this work, which was selected and programmed for this purpose by Schiefer [Sch07] within the scope of a research project at the BGIA.
4.3.1.2 The Different Sensor Systems

Most parts of the hardware setup of the sensor system framework and the different sensor system units have already been described in detail in previous work ([Ste08]). Therefore, only a summary of these is given here for the sake of completeness, complemented by a detailed description of the aspects that are new in this work. The EMG sensor is not described in this section, as it has neither been developed nor analysed in detail as a part of this work.

The sensor system framework, which is the basis for every sensor system unit, specifies only a few aspects of the hardware setup. These aspects are the configuration of the PSoC chip with respect to the selection of input and output pins and the associated circuitry, including the filters for the supply voltage, a light emitting diode (LED) for the control of the PSoC’s status and a connector for the in-circuit programming adapter, which allows to (re-)program the PSoC although it has already been soldered onto the circuit board. The pin configuration of the PSoC is shown in figure 4.14 and the circuit diagram of the associated circuitry is shown in figure 4.15. The input and output pins were selected with the intention to position pins with related functions close to each other. Unfortunately, due to specific restrictions of the chosen PSoC chip, this was not always possible.

It can be seen in both the pin configuration and the circuit diagram that there are two digital communication interfaces implemented on the PSoC: a universal asynchronous receiver/transmitter (UART) module, which allows to connect the PSoC to an RS232-interface of a personal computer, for example, by the use of an additional voltage level converter (e.g. a MAX232), and a serial peripheral interface (SPI), which is used to communicate with the above mentioned USB interface controller. However, there are no connectors specified in the framework for either one of the interfaces. The reason for this is that multiple PSoC chips can be connected to a single USB controller via conducting paths on the circuit board. This is explained in detail in the next section. In the following paragraphs, the setup of the individual sensor system units is described and the respective circuit diagrams are shown. In these individual circuit diagrams, the components of the framework are left out, as they are identical for every sensor system.

The ECG Sensor

Figure 4.16 shows the circuit diagram of the ECG sensor’s circuitry, which is explained in the following.

The disposable single-use electrodes that are used to acquire the differential ECG signal are connected to standard medical electrode wires with snap fasteners, which allow to attach them fast and easy while still providing sufficient mechanical strength. The electrode wires are equipped with robust Lemo plugs, which are used to connect them to the sensor system. The two signal lines are connected
Figure 4.14: Input / output pin configuration of the PSoC as specified by the sensor system framework

Figure 4.15: Circuit diagram of the PSoC chip as specified by the sensor system framework
directly to the differential inputs of the AD623 instrumentation amplifier, as can be seen in figure 4.16. In the previous work, a reference voltage of $+2.5V$ was added to the voltage coming from the electrodes via $1M\Omega$ resistors on both inputs. This was done in order to ensure that the allowed input range of the amplifier was not violated (compare to [AD99]), because otherwise, negative voltages might have been applied to the input pins of the amplifier. Applying negative voltages to the input pins would lead to undefined operating conditions of the AD623, as it is supplied with $+5V$ and $0V$ only (single supply) and therefore not able to output negative voltages. However, due to the fact that in this work the ECG sensor system is used simultaneously with the skin conductance sensor, which needs to place two additional electrodes to the skin and to apply a voltage to them, the previously proposed setup can not be used anymore: any voltage that is added to the ECG input pins is pulled down again by the skin conductance electrodes, which are connected to ground and $+0.5V$ without additional resistors. Fortunately, the application of $+0.5V$ to the skin is still sufficient to ensure that no negative voltages are applied to the input pins of the instrumentation amplifier and therefore, the ECG sensor works correctly as long as the skin conductance sensor is attached simultaneously.

The reference voltage pin of the instrumentation amplifier is supplied with the reference voltage of $+2.5V$ that is generated by the PSoC in order to shift the output of the amplifier: if a differential input voltage of $0V$ is applied, the output is shifted to $+2.5V$, making it possible for the amplifier to output also the negative parts of the differential input signal correctly. The input amplifier and the analogue to digital converter of the PSoC chip use the same reference voltage to ensure the correct digitalisation of the signal. To remove possible switching noise from this reference voltage, it is filtered by two low-pass filters, which are implemented by a pair of capacitors (similar to the supply voltage filtering of the PSoC chip, see section 4.3.1.1). Identical filters are applied on the supply voltage pin of the amplifier, too. The gain factor of the amplifier is set to $G=100$ by connecting the two gain selector pins of the amplifier via an $1k\Omega$ resistor.
After it has been amplified, the signal is filtered by a simple first order high-pass filter with a very low cut-off frequency of 0.5Hz in order to remove possible base line drift. Then, the signal is fed into the PSoC chip, where it is amplified again by an internal Programmable Gain Amplifier (PGA) module, filtered by a pair of two pole second order low pass filters with a cut off frequency of 35Hz and converted to a digital signal by an analogue-to-digital converter with a nominal resolution of 14bit and a sampling rate of 200Hz.

The Respiration Sensor

The respiratory effort transducer, which has been described in section 4.2.4.2 of the chapter Theoretical Details, is connected to the respiration sensor system by a Lemo plug of the same kind than the one used for the ECG sensor. As can be seen on the circuit diagram of the sensor system, which is presented in figure 4.17, the two signal lines are not connected directly to the instrumentation amplifier, as the respiratory effort transducer replaces a fixed resistor in the (previously described) Wheatstone bridge.

The Wheatstone bridge is driven from the same supply voltage as the following instrumentation amplifier. Again, the supply voltage inputs are filtered by a set of low pass filters similar to the ones described before. The values for the fixed resistors of the Wheatstone bridge and the optimal gain factor for the instrumentation amplifier were calculated by using an Excel spreadsheet based on the following formula, which is derived from equation (4.5), which was described in section 4.2.4.2:

\[ U_W = 5V \times G \times \frac{(R_{Sensor} - R_{2W3}) \times R_{2W1/2}}{(R_{Sensor} + R_{2W1/2}) \times (R_{2W3} + R_{2W1/2})} \] (4.14)

In this formula, G stands for the gain factor of the following amplifier, \( R_{Sensor} \) is the (variable) resistance value of the respiratory effort transducer, \( R_{2W3} \) is the value of the resistor to which
$R_{\text{Sensor}}$ is compared (called R3 in figure 4.5) and $R_{2W1/2}$ is the (identical) resistance value of the “upper” two resistors of the Wheatstone bridge (called R2 and R4 in figure 4.5).

The range for the resistance value of the respiratory effort transducer begins at approximately 2kΩ and goes up to 125kΩ. According to the specifications of the transducer (see [BIO07]), the increase in resistance is in linear proportion with the increase in length. However, it was found out in previous work (compare to [Ste08]) that the opposite is the case: the resistance of the transducer is actually in inverse proportion with the length. Nevertheless the increase in resistance seemed to be linear, so it was tried to design the Wheatstone bridge and the following instrumentation amplifier in a way that would keep up this linearity by generating an output voltage that is in linear proportion with the resistance value. The result was an almost linear output voltage curve with a minimum of 0V output voltage for a resistance value of 2kΩ and a (theoretical) maximum of 4.3V for a resistance value of 125kΩ.

During the use of the sensor system and the resulting gain of experience with the transducer, it became obvious that the voltage response of the sensor was not optimal designed for the use in practice. The reason for this is that the transducer has to be attached to the subject in a way that it is not slipping down. Therefore, it has to be stretched a little in order to be tight enough even if the subject exhales completely, leading to a maximum resistance value of approximately 30 to 40kΩ in practice, as well as to a certain non-linear behaviour: the resolution is much higher when the transducer is less stretched and decreases significantly when it is stretched almost to the maximum. To react on these findings, the design of the Wheatstone bridge is changed for this work: the response curve is designed less linear to increase the resolution of the sensor for lower resistance values compared to higher resistance values and the upper limit for the voltage output range is set to 60kΩ instead of 125kΩ. This is achieved by choosing $R = 47k\Omega$ and $R_{\text{Compare}} = 2.7k\Omega$ in combination with a gain factor of $G=2$ configured for the instrumentation amplifier. The resulting voltage output curve is shown in figure 4.18.

The output signal of the instrumentation amplifier is filtered by a simple first order low pass filter with a cut off frequency of 10Hz. The filtered signal is than fed into a programmable gain amplifier module of the PSOC chip, which can be used to increase or decrease the signal amplitude to adapt it to changing operating conditions if necessary, before it is digitalised by an analogue-to-digital converter similar to the one used in the ECG sensor system.

**The Skin Conductance Sensor**

Similar to the ECG sensor system, standard medical electrode leads with snap fasteners and a Lemo-plug are used to connect the used EDA electrodes to the skin conductance sensor system. The circuit diagram of the sensor system is shown in figure 4.19.
It can be seen that the sensor replaces a resistor in a Wheatstone bridge, similar to the respiration sensor system, but in contrast to the respiratory effort transducer, the resistance that has to be measured and converted into a voltage is not an attribute of the sensor, but an attribute of the subject’s skin. Therefore, it is important to limit the maximum voltage that is applied to the electrodes to 0.5V, as described in section 4.2.4.3 of the chapter *Theoretical Details*. This is done by using a simple voltage divider that is implemented by two resistors (R3-D1 and R3-D2 in the circuit diagram) in a row between the supply voltage, which is filtered similar to the ECG and respiration sensor systems, and ground. Usually, in order to divide the voltage to one tenth, the values of the resistors would be chosen in a way that the equation $R_{3D2} = 9 \times R_{3D1}$ holds, which is derived from the
voltage divider equation (4.15). Unfortunately, as the Wheatstone bridge itself represents a second way two ground with a (roughly) similar resistance value, it influences the voltage divider strongly. It would have been possible to avoid this problem by choosing much smaller or much higher resistor values for the voltage divider, but this would have caused different problems: if the resistor values would be lower, the current flowing through the skin of the subject would have been increased as well as the power consumption of the complete sensor system, and if higher values would be used, the sensitivity to noise would rise significantly. Therefore, the problem was solved by choosing \( R_{3-D1} \) larger than \( \frac{R_{3-D2}}{6} \): if the Wheatstone bridge is assumed to be completely unbalanced (which would be the case for open ended electrodes and is the best assumption in this case, as it ensures that even for very high skin conductance values the voltage applied to the skin never exceeds the specified value), its resistance value, measured between supply and ground, would be 24.6\( k\Omega \) (calculated by formula (4.16)). Now, the corrected value for \( R_{3-D1} \) can be calculated by formula (4.21), which is derived from the formula for the calculation of the division factor (4.18). The resulting resistance value for \( R_{3D1} \), given a value of 100\( k\Omega \) for \( R_{3D2} \), is 15\( k\Omega \).

\[
U_{WB} = U_0 \times \frac{R_{3D1}}{R_{3D1} + R_{3D2}} \quad (4.15)
\]

\[
R_{WB} = \frac{1}{\frac{1}{R_{3W2} + R_{3W3}} + \frac{1}{R_{3W1} + R_{3\text{sensor}}}} \quad (4.16)
\]

\[
R'_{3D1} = \frac{1}{\frac{1}{R_{3D1}} + \frac{1}{R_{WB}}} \quad (4.17)
\]

\[
\text{Divider} = \frac{U_{WB}}{U_0} = \frac{R'_{3D1}}{R'_{3D1} + R_{3D2}} \quad (4.18)
\]

or

\[
R'_{3D1} = \frac{R_{3D2} \times \text{Divider}}{1 - \text{Divider}} \quad (4.19)
\]

and finally:

\[
R_{3D1} = \frac{1}{\frac{1}{R_{3D1}} + \frac{1}{R_{WB}}} \quad (4.20)
\]

\[
= \frac{1}{\frac{R_{3D2} \times \text{Divider}}{1 - \text{Divider}} - \frac{1}{R_{WB}}} \quad (4.21)
\]

The values of the fixed resistors that are a part of the Wheatstone bridge can be calculated by the use of an Excel spreadsheet, similar to the one used for the respiration sensor system and based on the same equation (see previous paragraph). In the previous work, the range of the skin conductance
Figure 4.20: Voltage output of the skin conductance sensor with respect to the skin conductance value

was specified as 2 to 40 $\mu$S (see [Ste08]). It was shown during the further development of the sensor system that the upper limit of 40 $\mu$S is much too high if the electrodes are placed on the side of the foot. Therefore, the range is changed in this work to a lower limit of 1$\mu$S and an upper limit of 20$\mu$S in order to increase the resolution of the measurement. The resistance values and the appropriate gain factor for the following instrumentation amplifier are calculated with the objective to achieve an almost linear output voltage curve with a maximum of +5V output for the lower limit of the input range and a minimum of 0V for the upper limit of the input range. This objective can be met by selecting $R_{3W1} = R_{3W2} = 2.2k\Omega$ and $R_{3W3} = 47k\Omega$ in combination with a gain factor of G=240. The resulting output voltage curve is shown in figure 4.20.

Due to limitations given by the design of the used AD623 instrumentation amplifier (compare to [AD99]), a gain factor of G=240 can not be used with the given differential input and common mode voltages. It was calculated in previous work that the maximum gain factor that can be used without violating the operating conditions of the amplifier is approximately G=30. Therefore, this gain factor is set by using a resistor with a value of 3.3k$\Omega$ and the programmable gain amplifier module of the PSoC chip is used to further amplify the signal after it has been filtered by a simple low pass filter with a cut off frequency of 0.5Hz. This setup has the additional advantage that the lower limit of the input range can be further decreased by decreasing the PGA’s gain factor during the operation of the sensor system.
The Oxygen Saturation Sensor

The hardware setup of the blood oxygen saturation sensor system is much simpler than the setup of the previously described sensor systems, as the sensor itself, as well as the control and analysis circuitry is taken from the Biopac pulse oximeter, which was described in section 4.2.4.4 of the previous chapter. Therefore, no additional hardware components besides the PSoC chip are needed, except for a three-pin connector, which is used to connect the sensor system with the Biopac oximeter module. The oximeter module is powered by the +5V supply voltage of the sensor system and transmits the oxygen saturation value back to the sensor system by the use of an asynchronous serial interface line. To receive and use this data, the PSoC is configured with an additional UART-receiver module.

4.3.1.3 The Combined Sensor Module

One of the main problems of this work was to combine the different sensor systems in order to reduce the size and complexity of the system, as well as to make the usage of the system easier. The approach to this problem is to integrate all sensor systems into one sensor module, using a single printed circuit board and sharing a single power supply and a single USB interface controller.

On the hardware side, the main task that had to be performed was the design of a combined sensor module in terms of an appropriate circuit diagram and the layouting of the respective circuit board. All problems that might have come up when using more than just one sensor system simultaneously had already been avoided during the design of the individual sensor systems: the power consumption of the individual sensor systems has been kept low and every single power supply pin has been equipped with filters in order to avoid possible distortions from the other microcontrollers on the circuit board, therefore the use of a shared power supply is possible without any disadvantages. Additionally, the SPI bus, which is used to transmit data from the sensor systems to the USB controller chip, is a bus system that can be extended to multiple devices easily by adding one “slave-select” line (that is, one additional conducting path on the PCB) for each sensor system. This way, the USB controller, which is the SPI bus master, while the PSoC chips of the sensor systems are the SPI slaves, is able to “collect” the data from all of the sensor systems and to transmit a combined data package to the data storage device. The complete sensor module appears as a single USB device to the data storage device.

Due to the problem that the Biopac oximeter module is relatively big (8.5 x 7.5 x 2 centimetres) and that it might be replaced by different hardware in some future work, it was decided to develop two separated sensor modules: the first one is called “PhysioModule” and combines the ECG sensor, the respiration sensor and the skin conductance sensor on one circuit board that fits exactly in a
casing with a size of 10 x 5 x 2.3 centimetres, while the second one (called “Oxymodule”) contains only the oxygen saturation sensor system on a circuit board with a size of 5.5 x 5 centimetres, which is integrated in a second casing with a size of 12 x 10 x 3 centimetres, together with the Biopac oximeter module. Both sensor modules have their own USB controller chip and can be connected to the data storage device either both simultaneously or separated from each other. In the following, the designed circuit diagrams, board layouts and the completely assembled boards are presented.
Figure 4.21 shows the circuit diagram of the PIC USB controller chip with its directly associated circuitry, while the circuitry concerning the PIC’s program adapter, LEDs and the power supply are combined in figure 4.22. These components are identical for both sensor modules. It can be seen that the PIC microcontroller is clocked by an external oscillator with a frequency of 20MHz. It is supplied over two supply voltage pins and two ground pins, which are each filtered by a low pass filter, which is implemented by a single capacitor, to protect them from switching noise and similar disturbances. The PIC controller drives two LEDs (“LED1” and “LED2”) in total to indicate specific operation states; a third LED (“LED\textsubscript{PWR}”) is added to indicate the power state of the board. The board can be powered either by the Mini-USB connector or by an additional 7.2V battery, which can be optionally connected by a two pin Lemo plug. By setting a three pin jumper (called “\textit{V\textsubscript{CC Selector}}”), the power supply can be switched between the two inputs. Both power supply inputs are filtered and stabilised by a set of capacitors and an inductor coil each. The specific design for these components was taken from the reference design of the PIC controller as proposed by Schiefer [Sch08b]. For the power supply by battery, a MIC2950 voltage regulator is added, which converts the input voltage from 7.2V to 5V. Choosing a battery with less than 7.2V (e.g. a 5V battery, this way the voltage regulator could have been spared) was rejected, because this might have caused problems when the battery loses power: the voltage of a battery could drop below the specified voltage, leading to undefined states of the electronic components. By using a battery with 7.2V and a voltage regulator, this issue was resolved. An additional advantage of this is that there are well engineered batteries with this voltage available for low cost, as these batteries are the usual power supply for video cameras (camcorders). The same type of battery is used as power supply for the CUELA system, too.

The circuit board layout of the PhysioModule is shown in figure 4.23 and the layout of the OxyModule can be seen in figure 4.24. Both circuit boards have been designed with the Eagle layout editor on the basis of the previously presented circuit diagrams. For their development, only small SMD components were used: for the PSoCs, the already described SSOP packages were chosen and the PIC chip is implemented by a “Thin Quad Flat Pack” (TQFP) package. If available, all resistors and capacitors were implemented in 0805-packages with a size of 2 x 1.3 millimetres. These package sizes were selected, because they can still be soldered by hand - if even smaller packages should be used, the use of a mounting machine would have been necessary. However, this means that the size of the circuit board could still be reduced if necessary, especially if a circuit board with more than two layers would be used. Except for the components and the conducting paths, both sides of the circuit board are completely filled with conducting areas. On the upper side, the area is connected to the supply voltage, while the area on the lower side of the board is connected to ground. This design was chosen as it shields the circuit board from irradiations to a certain extend and therefore
makes it less susceptible to noise than a design in which supply voltage and ground are simple paths instead of complete areas.

It is easy to see in the images of the board layout that the components, which belong either to the power supply or to the PiC controller, are arranged almost identically on the right side of the circuit board for both sensor modules. The only difference is the design of the sensor system specific hardware and the complexity of the SPI bus system. On the PhysioModule’s circuit board, the Lemo plugs that are used to connect the sensors are arranged on the left side, while on the OxyModule’s board, the connector for the oximeter module is situated at that place. As the hardware setup of the oxygen saturation sensor system does not require additional components, its PSoC chip is directly connected to the oximeter module connector. In contrast to this, the PhysioModule is much more complex. The circuitry associated directly with the sensors (that is, the Wheatstone bridges and voltage dividers) is situated close to the respective sensor connectors in order to keep the wires short. Next to these components (to the right) are the instrumentation amplifiers arranged, followed by the components of the filters and finally by a PSoC chip for each sensor system. Each of the PSoCs is connected to the data lines of the SPI bus interface, which are “Master In Slave Out” (MISO),
“Master Out Slave In” (MOSI) and the data clock (CLK). They are easily visible in the image of the board layout, as they are situated on the back side of the board and therefore are shown in blue, right in the middle of the board. Additionally, a “slave select” (SSx) line is leading from every single PSoC to the PIC chip. This line is used by the PIC to select the PSoC he is transmitting or receiving data to and from. The jumpers and the five pin connectors, which are situated close to each of the PSoCs and the PIC chip, are used to (re-)program the respective microcontroller.

Images of the completely assembled circuit boards are presented in figure 4.25 and 4.26, respectively.

4.3.1.4 Data Storage and Control Device

It has already been described that a measurement control and data storage device is necessary for the use of the development measurement system. This device, which is called “datalogger” in this work, following the nomenclature of the CUELA system (see [Her03]), needs to be USB host capable (that is, it must be able to recognise and power USB devices, as well as to communicate with them) and to have sufficient non-volatile storage memory available to store all data it receives from the USB device(s) during the measurement period.

The idea about this new datalogger is to be able to connect both the new (and currently developed) USB-based CUELA system and the physiological sensor modules to it. This way, only one datalogger is necessary and both size and weight of the measurement system are reduced. It has been proposed in the work by Linder [Lin07], as well as the one by Schiefer [Sch07] that a portable digital assistant (PDA) that is running the Windows CE operating system should be used for this purpose. Such a handheld computer has several advantages for the use as datalogger: it is both small and lightweight and it allows the subject and the supervisors of the measurement to view and check the status and validity of the measurement process “online” on the PDA’s display, as well as to manipulate the sensor configuration or to control the measurement system by using the PDA’s buttons or touchscreen. Additionally, wireless communication interfaces such as Bluetooth or WLAN can be used to transmit the recorded data to an external computer for a more advanced and more detailed online-analysis. However, there is one (temporary) problem with this approach: both the necessary sensor device drivers and the datalogger software for the PDA is still in development and not yet available to be used in this work. Therefore, a temporary solution is proposed in this work, which can be replaced as soon as the PDA is fully functional: a Windows-PC-based software is used, which runs on a standard, yet as small and lightweight as possible, notebook computer. In order to make the change from the notebook to the future PDA device as easy as possible, the Microsoft .net-Framework is used to implement this software. This framework is available for both Windows-based PCs and Windows-CE-based PDAs and allows to use any device-independent programs similarly on both devices. For
Figure 4.25: The completely assembled PhysioModule in its casing

Figure 4.26: The completely assembled OxyModule in its casing
this work, this means that only the device drivers and the graphical user interface have to be adapted, while the rest of the developed software will work without further changes.

As the interesting attributes of a notebook computer for this purpose are mainly its size, weight, power consumption and robustness, while high performance is not necessary for the given task, the Asus Eee-PC 900 was found to be the optimal choice for the following reasons (see [ASU08]):

- It is running on Windows XP and supports the .net Framework, enabling it to run the developed software.
- It is comparably small for a notebook computer (approximately 22.5 x 17 x 3.5 centimetres) and weighs less than 1kg when equipped with the standard battery.
- The resolution of the screen is sufficiently high and the screen itself sufficiently large to reasonably display all sensor values during the measurement.
- Due to its solid state disk drive, it is much more shock-resistant than notebooks with conventional harddisks.
- It provides enough memory on the solid state disk to store several hours of measurement data.
- Its low voltage (7.4V) battery ensures the safety of the subject and runs up to 2.5 hours, which is sufficient for the type of measurements planned for this work.

Additionally, it is available for a comparably low price, which made it even more attractive. Figure 4.27 shows the Eee-PC besides the author’s hand in order to give an impression of the notebook’s actual size.

Figure 4.27: Asus Eee-PC compared to the size of a hand
4.3.2 Software Setup

In this section, an overview over the structure and setup of the software, which was developed as a part of this work, is given. This includes the firmware of the different sensor systems, the firmware of the USB controller that connects each sensor module to the datalogger, the measurement control software running on the proposed notebook computer, the strain generation software and the software used for the data analysis and classification.

4.3.2.1 Software Setup of the Sensor Systems

The software setup of the different sensor systems is almost completely based on the proposed sensor system framework. Both the setup of the framework and the differences of the individual sensor systems have already been described in previous work in detail (see [Ste08]) and have not changed since then. Therefore, this section is only a short summary.

The software running on the PSoC controllers (the firmware) provides all functionality to digitalise, filter and analyse the input signal, as well as to calculate the physiological parameters and to transmit the results of the performed calculations to the datalogger. As the necessary setup for these functions is the same for all sensor systems, except for the oxygen saturation sensor, it has been added to the PSoC sensor system framework in the form of a pre-configured software project. The project is configured to use +5V as power supply voltage as well as the highest possible clock rate of 24MHz and includes the following components:

- one DeltaSigma-ADC with a nominal resolution of 14bit and up to 7812Hz sampling rate,
- two programmable gain amplifiers (PGA): the first one is used for the generation and output of a reference voltage of +2.5V (which is also used as conversion reference for the ADC), while the second one is used as an input amplifier (or voltage follower) for the physiological signal,
- one SPI slave module (SPIS), which is used to communicate with the PIC USB microcontroller in order to transmit data to the datalogger,
- one UART transmit module (TX8) for development or debugging purposes,
- and a combination of an 8-bit timer (TIMER8) and a 32-bit counter module (CNTR32), which are used as a real time clock.

The software itself is build up in a modular way, allowing to exchange the sensor-specific parts without the need to change or modify the “general” functions. These software modules and their corresponding functions are shown in figure 4.28 in the form of an UML class diagram. This representation allows
to survey the software’s structure easily, although the PSoC’s firmware has not been developed by using object-oriented programming.

The “Main” module of the software contains a “general structure”, represented by pre-defined, yet not fully implemented, functions that have to be adapted and extended according to the specific sensor system. It defines the operating mode of the sensor system and starts a respective operating function. This operating function implements the actual digital signal processing chain and performs an endless loop, in which it acquires data using functions from the “PSoC Basics” module, process this data using sensor-specific functions from the respective “Sensor” module and, finally, transmit the resulting physiological parameters by using the “PSoC Basics” module again. Additionally, the “Main” module provides a sensor specific reset-function, which allows to reset all parameters and the calibration data, which has been stored by the signal analysis algorithms.

In contrast to the “Main” module, the module “PSoC Basics” is independent from the specific sensor system and contains all (already completely implemented) functions provided by the framework. Therefore, if any changes are made to the code of this module, the updated module can be copied into the software project of all different sensor systems without any modifications. The module contains functions to initialise, reset and shutdown the microcontroller and its components, to start and stop the measurement, to change the gain factor of the input amplifier, to handle the real time clock and to acquire new data values by the use of the analogue-to-digital converter. It also handles the complete communication protocol for both the SPI bus and the serial UART interfaces and provides functions to transmit single characters or entire strings over the serial interface, as well
as functions to receive and transmit commands and 8- or 16-bit data words over the SPI interface. Additionally, it provides a digital IIR notch filter, which can be used to remove the 50Hz noise caused by main power lines that is contained in the ECG and (less disturbing) in the skin conductance signal. This IIR filter has been implemented by following a design approach proposed by Smith (see [Smi99]) and is described in previous work (see [Ste07]).

Finally, the “Sensors” module contains the functions that are specific to the respective type of sensor. As the requirements of the sensors are different, no special structure for this module is defined for it by the framework - the methods that are shown in the UML diagram are only an example of typical methods for this module. Therefore, the setup of the individual sensor system is described in the following.

The ECG sensor system is configured with three additional components (user modules, respectively): two low pass filter modules (LPF1 and LPF2) and one 8-bit pulse width modulator (PWM8), which generates the clock frequency that is needed by the low pass filters. The filters are arranged sequentially between the input amplifier and the analogue-to-digital converter. They are implemented as two-pole second order filters with a cut off frequency of 35Hz and Butterworth characteristic, which seemed to be best suited for this purpose (compare to [Smi99]). After passing the filters, the signal is digitalised with a sample rate of 200Hz, which is sufficiently high to comply with the Nyquist-Shannon sampling theorem, as it was shown in the evaluation of the filters in previous work (see [Ste07]) that the attenuation at the Nyquist frequency \( f_N = \frac{f_{\text{sample}}}{2} = 100 \text{Hz} \) is high enough to avoid disturbing aliasing effects. The software module “Sensor”, which is called “ECG Sensor” in this case, contains a function to initialise the sensor specific modules, as well as algorithms that are able to detect the R-waves in the ECG signal and to calculate both the heart rate and the heart rate variability. Furthermore, a specialised (in terms of the data format and buffer length) Median filter algorithm is contained in the module, which is used to filter the calculated parameters before they are transmitted in order to remove outliers caused by incorrect sensor readings, for example. An UML diagram of the ECG module is shown in figure 4.29.

In contrast to the ECG sensor system, the respiration sensor system does not require any additional components, as the input signal is already filtered by an external low pass filter with a cut off frequency of 10Hz before it is fed into the PSoC chip. The input amplifier, which is included in the sensor system framework, is used as a voltage follower in this setup, but can be used to increase or reduce the amplitude of the signal by changing its gain factor at runtime. This might be useful in situations where the chest strap can not be attached appropriately, resulting in a reduced amplitude of the signal. The following analogue-to-digital converter is configured with a sample rate of 100Hz, which is sufficient to comply with the Nyquist-Shannon theorem, even though the used low pass filter is only a first order filter with a relatively low attenuation. The associated software module
“Respiration Sensor” contains functions to initialise the sensor specific configuration, to detect the peaks in the signal, which is acquired from the respiratory effort transducer, as well as to calculate the respiration rate and the amplitude of the last breathing cycle, which can be filtered by applying a Median filter on them. In figure 4.30, the UML diagram of this sensor module is presented.

The skin conductance sensor also does not require additional components, as its input signal is filtered by an external low pass filter, too. The cut off frequency of the low pass filter is 0.5Hz; therefore, a sample rate of only 10Hz is sufficient for this sensor system. The calculation of the resulting parameters skin conductance level (SCL) and skin conductance response (SCR) is performed by a function, which is provided by the “Skin Conductance Sensor” module, which is presented in figure 4.31, again in the form of an UML diagram. This function uses the also provided implementation of a Median filter to remove outliers from the data stream. Finally, a function for the initialisation of the sensor specific algorithms and variables is included.

As the oxygen saturation sensor system is different from the other sensor systems in the terms of data acquisition, the configuration of the PSoC’s components differs from the configuration proposed
4.3. Technical Details

<table>
<thead>
<tr>
<th>Skin Conductance Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>wSCL : unsigned int</td>
</tr>
<tr>
<td>wSCR : unsigned int</td>
</tr>
<tr>
<td>initialiseSC() : void</td>
</tr>
<tr>
<td>calcSC(ein wData : unsigned int) : void</td>
</tr>
<tr>
<td>applyMedianFilter(ein wData : unsigned int) : unsigned int</td>
</tr>
<tr>
<td>getSCR() : char</td>
</tr>
<tr>
<td>getSCL() : char</td>
</tr>
</tbody>
</table>

**Figure 4.31:** UML class diagram of the skin conductance sensor module

<table>
<thead>
<tr>
<th>Oxy Sensor</th>
</tr>
</thead>
<tbody>
<tr>
<td>bRX8Data : char</td>
</tr>
<tr>
<td>bRX8DataNew : bool</td>
</tr>
<tr>
<td>initialiseOxy() : void</td>
</tr>
<tr>
<td>RX8_ISR_handler() : void</td>
</tr>
<tr>
<td>applyMedianFilter(ein bData : char) : char</td>
</tr>
<tr>
<td>getDataNew() : bool</td>
</tr>
<tr>
<td>getData() : char</td>
</tr>
<tr>
<td>getRAWData() : char</td>
</tr>
</tbody>
</table>

**Figure 4.32:** UML class diagram of the oxy sensor module

in the framework: all analogue user modules have been removed, including the analogue-to-digital converter and the input amplifier. Instead, a UART receiver module (RX8) and an associated pulse width modulator for the generation of the necessary data clock are added. The UART receiver allows to receive data bytes that are transmitted over a serial data line with a known data clock, in this case: the data that is transmitted by the Biopac oximeter module. The oximeter module’s data clock was examined by measuring the length of the data pulses on the signal using an oscilloscope, as described in previous work (see [Ste08]), and found to be approximately 4800Hz. With this configuration, it is possible to receive and analyse the data that is transmitted by the oximeter module without the need to use the analogue components of the Biopac amplifier module and to digitalise the signal afterwards. On the software side, the “Oxy Sensor” module provides a function to initialise the UART receiver and the algorithms, as well as an interrupt handler method, which is used to acquire the data transmitted by the oximeter module, which can additionally be filtered by a Median filter. The arrival of new data is signalled by a flag. Both the status of this flag and the data itself can be queried out of the “Main” module by calling the respective get-functions of the “Oxy Sensor” module. The UML diagram of the module is shown in figure 4.32.

### 4.3.2.2 Software Setup of the Sensor Modules

For the two physiological sensor modules, the software setup is almost identical: the PSoC microcontrollers communicate with the datalogger by the use of a SPI-to-USB bridge, which is represented by the PiC USB controller. The only difference is that the OxyModule contains only one PSoC microcontroller (sensor system, respectively), while the PhysioModule contains three of them. Therefore,
the USB controller has to request data from all three sensor systems after each other before it can transmit a (combined) data packet to the datalogger. The communication with the datalogger itself is done by using the USB-HID device class. The human interface device (HID) class is the same class that is used by USB keyboards or a mouse, for example, and allows to simplify the development of a device driver. The software for the USB controller and the respective USB communication protocol have been developed by Schiefer in a project contemporary with this work (see [Sch07] and [Sch08b]).

On the datalogger side, any device driver can be used that fulfils the following requirements: it must support HID class devices and allow the developer to specify the “Vendor” and “Product” IDs of the device that shall be used. These IDs uniquely specify a device and are assigned by its manufacturer. In this case, the IDs have been specified by Schiefer and allow the device driver to identify the sensor modules as soon as they are connected to the datalogger. The driver also must be able to receive and transmit data packets independent of the actual data type, as the datalogger software has to access single data bytes directly.

The USB communication protocol deals with packets with a fixed length of 32 bytes and differentiates between data packets and information and control packets ([Sch07]). The structure of these two types of packets is shown in figures 4.33 and 4.34. As can be seen, the packet type is specified by the first bit of the packet: this bit is zero for data packets and one for information and control packets. In case of a data packet, this bit is followed by a 15-bit packet counter that is used to ensure that no packet is lost during the communication, while in the case of an information and control packet 8 of this 15-bit are used to specify the type of the information packet. For the communication with the PSoC sensor systems, the type is always set to “0x0A”, which means that the packet has to be relayed (or bridged, respectively) on to the SPI bus. The third byte is in both cases used as a segmentation counter, which allows to transmit data that is longer than 28 bytes (which is the actual payload length of each packet) by splitting it into up to 15 segments. The following byte contains the sensorbox-ID, which is used to specify the sensor system to or from which the packet is transmitted.

While the sensorbox-ID is always one for the OxyModule, as there is only one sensor system on that module, it can be between one and three for the PhysioModule, where one stands for the ECG sensor system, two for the respiration sensor system and three for the skin conductance sensor system. This header is followed by the 28 bytes of actual payload.

The 28 bytes of payload in the USB packets are used to encapsulate the SPI packets that are sent to or from the sensor systems. The structure of the SPI packets is modelled after the example of the USB data packets: similar to the USB packet types, the SPI packets are differentiated into data packets and command packets. In the case of a command packet, the USB information and control packet type is used to encapsulate the SPI packet and in the case of a data packet the USB data packet
Figure 4.33: Structure of the USB data packets [Sch07]

Figure 4.34: Structure of the USB information and control packets [Sch07]
type is used, respectively. To indicate the beginning of a new data packet, one byte with a value of 0xFA (decimal: 250) is transmitted previously of each data packet. This “start-marker” byte ensures that the receiving device is able to correctly detect the beginning of the new packet, as it happens frequently that the transmitting device repeats an “old” byte before it begins to transmit the new data bytes. These marker bytes are removed by the receiving device immediately and are not treated as a part of the data packet. The SPI packets themselves are transmitted completely unmodified as the payload of the USB packet. Therefore, the datalogger (or the user, who is controlling the measurement system by the use of the control software) is able to communicate with the sensor systems as if they were directly connected to it - the USB communication is transparent to both the datalogger and the user.

![Figure 4.35: Structure of the SPI command packets](image)

The structure of the SPI command packets is presented in figure 4.35. In contrast to the SPI data packets, the command packets’ structure is identical for both the PhysioModule and the OxyModule. It can be seen in the figure that the packets start with one byte indicating the length of the payload (that is, the number of the following bytes), followed by the command, which also uses one byte. All following bytes (up to a maximum of 26 bytes) are optional arguments to the command. Table 4.3 presents all specified commands in combination with the possible arguments and their length on the way to the sensor system (“In”) and back to the datalogger (“Out”).

The structure of the SPI data packets is different for each sensor system, as the length of data to transmit is different according to the number of physiological parameters that the respective sensor system calculates. Additionally, the ECG sensor system and the respiration sensor system transmit three bytes of technical parameters for reasons of both debugging and measurement control, namely the minimum and maximum signal amplitude, which was measured in the last time interval, as well as the last threshold value. However, the SPI data packets are always arranged in the same order: the first byte indicates the number of following bytes, which contain one physiological or technical parameter each. In the case of the oxygen saturation sensor on the OxyModule, these are three bytes in total, which are simply encapsulated in one USB data packet by the USB controller. Figure 4.36 shows the structure of the SPI data packet.
4.3. Technical Details

<table>
<thead>
<tr>
<th>Command</th>
<th>Arguments In</th>
<th>Arguments Out</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 = Stop</td>
<td>None</td>
<td>Acknowledgment (1 byte)</td>
</tr>
<tr>
<td>1 = Start</td>
<td>None</td>
<td>Acknowledgment (1 byte)</td>
</tr>
<tr>
<td>2 = Get Sensor Type</td>
<td>None</td>
<td>Type descriptor (up to 26 bytes)</td>
</tr>
<tr>
<td>3 = Set Gain</td>
<td>New gain factor (1 Byte)</td>
<td>Acknowledgment (1 byte)</td>
</tr>
<tr>
<td>4 = Get Gain</td>
<td>None</td>
<td>Current gain setting (1 byte)</td>
</tr>
<tr>
<td>5 = Calibrate</td>
<td>None</td>
<td>Acknowledgment (1 byte)</td>
</tr>
<tr>
<td>6 = Reset</td>
<td>None</td>
<td>Acknowledgment (1 byte)</td>
</tr>
<tr>
<td>Default</td>
<td>Do not care</td>
<td>Error code (1 byte)</td>
</tr>
</tbody>
</table>

Table 4.3: SPI commands and arguments

For the PhysioModule, things are a little different, as in this case the USB controller collects and transmits the data of all three sensor systems on that module sequentially in one USB data packet. Nevertheless, the SPI data packets are not modified in any way and even the bytes indicating the length of the single SPI packets are simply added to the payload of the USB data packet, as can be seen in figure 4.37.

The measurement control software, which is running on the datalogger, has to implement this protocol exactly in order to correctly “unpack” the received data. Therefore, the setup of this software is explained in the following section.

4.3.2.3 Setup of the Measurement Control Software

The measurement control software, which has been named “PhysioLogger”, performs two important tasks: it handles the detection of the USB-based sensor modules as well as the communication with them, including the operation control of the sensor systems, and it stores the data that is transmitted by the sensor modules in a file on non-volatile memory. For practical reasons, it also performs a third task: it allows the user of the measurement system to view the received data of all connected sensor systems in real time (that is, when they just have been received) in order to verify that the sensors are applied and working correctly without the need to investigate the stored data file.
The PhysioLogger has been developed in Microsoft VisualStudio and is implemented in the object-oriented programming language C#. This programming language has been chosen for personal preference as it combines the advantages of both Java and C, which could have been used as an alternative. The software is based on the Microsoft “.net 2.0” framework, allowing it to be ported comparably easy to PDA handheld computers, which will be used as dataloggers in future versions of the CUELA system. Additionally, it is (theoretically) possible to use the software without modifications on different operating systems, for example on Mac OS X or Linux, as the .net framework has been designed to be independent from the operating system, as well as from the device it is running on. Unfortunately, Microsoft did not release final versions of the .net framework for other operating systems than Windows, yet. However, it should be possible with low effort to use the recently released version 2.0 of Mono\(^2\), which is a free and open source alternative to Microsoft’s implementation of the .net framework, in order to run the PhysioLogger on Mac OS X or Linux platforms.

The PhysioLogger is based on a simple USB HID-device driver, which was proposed by Linder ([Lin07]). The driver is integrated into the software in the form of a library and treated similar to a component of the user interface. This allows to use event-triggered methods for the communication over the USB interface, but has the drawback that the component can only be used in the main class (that is, the “form”) of the software and can not be encapsulated in an additional object. Therefore, all communication related methods had to be implemented in the PhysioLogger class, together with the methods that are related with the user interface. The class provides functions to send SPI and USB commands, to react on the arrival or removal of the sensor modules, as well as for the receipt

\(^2\)http://www_mono-project.com
4.3. Technical Details

The PhysioLogger class holds a list of strings called “dataBuffer”, in which all received data items are stored, as well as a “tempFile”, into which the same data is written sequentially directly after new data has been received by the use of a respective function in order to ensure that no data is lost, even if the software crashes or if the power of the computer’s battery runs out. Furthermore, the PhysioLogger class holds an instance of two additional classes, which have been specified for the handling of command packets: a Command Packet Generator and a Command Queue Handler. The Command Queue Handler holds a list of Command Packets, which are objects that are generated by the Command Packet Generator and that contain the ID of the target for the respective command as well as the command itself and its associated arguments, and allows to enqueue and dequeue Command Packets from the list, as well as to query the number of remaining commands in the list. It is used to ensure the correct transmission and acknowledgment of each command packet. A more
detailed description of the implementation of these classes and the program flow is given in the next chapter, *Implementation Details*.

The graphical user interface of the PhysioLogger is optimised for the resolution of the Eee-PC’s screen and displays all received physiological data values and technical parameters. Additionally, it provides several graphical control elements such as buttons and drop down menus to control the operation of the sensor modules. Figure 4.39 shows a screenshot of the user interface.

### 4.3.2.4 Setup of the Strain Generation Software

As described in chapter *Theoretical Details*, an interactive software has been developed that prompts the user (or the subject during the measurements, respectively) to solve mathematical and logical exercises in a limited amount of time in order to generate stress. Therefore, this software is called “Stress Generator”. The Stress Generator has been developed with VisualStudio and in C# for the .net framework, similar to the PhysioLogger software.

The implementation of the Stress Generator uses four classes in total, as can be seen in the class diagram that is presented in figure 4.40. The main class, which is also the main form of the project (that means, all elements and components of the user interface are a member of this class), contains all event-triggered functions related to the graphical user interface, as well as a function to display the next exercise and a timer-based function, which indicates the remaining time to solve this exercise. The Stress Generator differentiates between math exercises, which the user has to solve by mental arithmetic and typing in the solution, and colour exercises, in which the user has to click a button naming the colour of the letters of a displayed word. These exercises are implemented in form of the additional classes “ColorExercise” and “MathExercise”. Both classes provide a function to check
if the solution entered by the user is correct, as well as functions to get information about the respective exercise (the colour or the operands, for example). Both types of exercises are generated by a fourth class, the “ExerciseGenerator”. In this class, the available colours for the colour exercises are specified, as well as the difficulty of the math exercises. This feature is currently limited to the two settings “easy” and “difficult”, but could be extended to more steps easily, if necessary. The “ExerciseGenerator” also contains a random number generator, which is used to generate new exercises. For this purpose, it provides functions to generate new exercises of a specified type, which are called whenever the user either solved the exercises correctly or did not manage to solve them before the time has run out.

The design of the graphical user interface, which is shown in figure 4.41, has been kept as simple as possible and uses large fonts and buttons in order to ensure that the user does not have to waste time searching for the actual exercise or for the button with the right solution.
4.3.2.5 Setup for the Data Analysis and Classification

For the analysis of the recorded data, the file that has been created by the PhysioLogger software has to be analysed in combination with the EMG and activity data that was recorded by the CUELA datalogger simultaneously. For this purpose, the EMG and the activity data have to be loaded into the CUELA analysis software WIDAAN. Using this software, the EMG data has to be normalised in order to be able to compare the recorded data of different subjects and measurement sessions. WIDAAN is also used to calculate a physical activity index (physical activity intensity, PAI) out of the recorded activity data. Both the normalised EMG data and the physical activity index are then used together with the data recorded by the PhysioLogger to classify the strain of the subject over the time.

The idea about this classification process is to implement the necessary decision rules in WIDAAN, as this allows to use just one software for the complete analysis of the data and to re-use the existing functionality, for example the synchronisation with the video stream and the graphical representation of the data. For this purpose, the physiological sensor data can be loaded into WIDAAN and treated as new sensor channels, which are automatically viewed as a graph over the time. Based on the synchronisation markers, which have been added by the PhysioLogger during the measurement, the

![Graphical user interface of the Stress Generator software](image)

**Figure 4.41:** Graphical user interface of the Stress Generator software
4.3. Technical Details

Figure 4.42: Measurement setup for the recording of training data

new channels can be linked with the other channels and the video recording. Afterwards, the different strain situations can be calculated directly in WIDAAN by using a decision tree algorithm, which has been learned with well-defined training data sets. The resulting strain classification is then also displayed as a graph and can be summed up over a selected time frame, for example.

4.3.3 Research of Classification Rules

In order to research the design of rules for the classification of the sensor data with respect to the previously specified strain classes (see section 4.2.5.1), a measurement setup was designed and performed by different subjects to gain appropriate data for the analysis. This measurement setup is described in the following section (4.3.3.1). Afterwards, the actual machine learning process is described in section 4.3.3.2.

4.3.3.1 Recording of Training Data

As described in section 4.2.5.2 of the chapter Theoretical Details, a bicycle ergometer and a computer-based math and colour test are used to induce both physical and psychological strain in the subjects. A respective measurement setup has been designed that allows to gain training data for each of the proposed strain classes. Figure 4.42 shows a photograph of the measurement setup.
It can be seen that two “workplaces” have been prepared for the subject. The first one (on the right side) is a typical office workplace. At this workplace, the scenarios that are associated with the classes one to three are performed, beginning with the resting phase, followed by the low and high psychological strain induction phases. In these latter phases, the subject has to use the provided laptop computer to solve the mathematical exercises and the colour test provided by the Stress Generator software.

For the following phases, the subject has to exchange his workplace for the bicycle ergometer. For this purpose, the height of the table is increased electronically to a height of approximately 130cm, which positions the additionally provided screen on eye-level directly in front of the subject. After phase four has been completed, the Stress Generator software is used again to induce additional psychological strain in the subject. As it is not possible for the subject to enter the solutions of the exercises himself while he is riding the bicycle, an assistant is needed to operate the laptop computer. Figure 4.43 shows one of the subjects at the bicycle ergometer workplace.

4.3.3.2 Machine Learning Process

Before the acquired data can be used by the machine learning algorithms, it must be combined appropriately in the form of a single data table in the “comma-separated-value” (CSV) format. As the acquired data does not contain any information about the type of the strain that the subject
was encountered with, this information has to be added by hand by entering the respective strain class (that is, the numbers one to nine) in each row of the exported table. In a future version of the WIDAAN software, it will be possible to use its graphical user interface and its marker-feature for this purpose (see section 4.4.4), however, this is currently not possible.

The CSV-formatted data table can be loaded into the pre-processor of the Weka Explorer, which is shown in figure 4.44. It displays all available attributes and information about the instances of each attribute, such as the mean value and the distribution of the values. In order to use a classification-algorithm on this data, the classifying attribute has to be specified as “nominal”, which means, the range of possible values for this attribute has to be specified. As the strain class is used as classifier in this work, the possible values are the numbers one to nine. All other attributes can be specified as “numerical”, which means that they are treated as real numbers with an arbitrary value.

On the “Classify” tab of the Weka Explorer, a classification algorithm for the use on the data can be selected and configured, as well as the method used to evaluate the quality of the resulting classification. The basic method to evaluate the performance of the learned algorithm is to use separated training and test data sets. The decision tree, for example, is then learned on the training data set and its performance is tested with the test data set (see [Wit05]). However, according to
Witten and Frank, the “stratified 10-fold cross-validation” is a better way to predict the error rate of the chosen learning algorithm. In contrast to the typical setup with two sets of data (a training data set and a test data set), with cross-validation the data is partitioned randomly into ten sets, which are called folds in this context, and each fold is used in turn as test data set, while the nine remaining sets in each turn are used as training data. The folds are generated in a way that each class value is represented in the folds in the same proportion as in the complete data (that is, the folds are “stratified”). After the ten turns are complete, the error estimates of each turn are averaged to achieve an overall error estimation. In this work, the stratified 10-fold cross-validation is used to compare the performance of the learned classification algorithms.

A large number of algorithms that can be used to classify the data are available in Weka ([Wit05]). Figure 4.45 shows the output of a decision tree algorithm that has been applied on the data recorded with the previously described measurement setup. As already stated in section 4.2.5.1, a decision tree learning algorithm seems to be the best choice for this work due to the characteristics of the used data. This argumentation is supported by the shown results: the algorithm classifies 98.59% of the data instances correctly. Nevertheless, a comparison of the results of different algorithms is given in the chapter Evaluation. Additionally, the results of the decision tree algorithm are discussed in detail and with respect to the selected attributes and configurations in section 5.3.2.

4.4 Implementation Details

In this chapter, the implementation of the developed software for the sensor systems, the PhysioLogger and the Stress Generator software are explained in detail. As it would be too complex to show and explain the source code directly, flow charts are used instead to explain the developed algorithms and functions. Additionally, this chapter covers the implementation of the data classification rules on the basis of a pre-learned decision tree, as well as the integration of both the data itself and the analysis algorithms in the WIDAAN software.

4.4.1 Implementation of the Sensor Systems

As it has been said in the previous chapters, the PSoC-based sensor systems are implemented by using and extending a common sensor system framework, which has already been proposed and described in previous work (see [Ste08]). Therefore, the focus of this section is more on the algorithms, which have been strongly improved and extended in this work, and on their workings than on the implementation of the basic functions. However, if the exact implementation is of interest, refer to the completely commented source code that is available as a part of the Appendix.
Figure 4.45: Screenshot of the Weka Explorer showing a classification result

Figure 4.46 shows the flow charts of the three most important functions of the sensor system framework: the main() function, which starts up the sensor system and defines the operating mode, the processSPI() function, which checks if commands have been received and reacts on them, and finally the SPIS-ISR-handler() function, which handles the actual data transmission and receiving process. All other functions of the framework, which were shown in its UML diagram in figure 4.28, are left out here, as their implementation is comparably simple and does not need to be explained in detail.

The main() function is called as soon as the sensor system is supplied with power. At first, it initialises the PSoC components, which are included in the framework, including the timers used to implement the real time clock and the amplifiers used to amplify the input signal, as well as to output the reference voltage of +2.5V. Afterwards, it calls the initialisation-function provided by the respective sensor module. Then, it reads the variable, which specifies the operating mode for the sensor system, and starts the program function associated with this operating mode. Operating mode five indicates the program that is used in practice, while the modes from zero to four are used
for debugging purposes only. If the program flow ever returns from the respective program function, which might be the case if a critical error occurred, for example, the main() function tries to get to a “safe state” by disabling all user modules and ends the program flow.

During the operation of each sensor system, the function \( \text{processSPI()} \) is called frequently in order to check for activity on the SPI bus and to react on incoming commands. The function checks, if new commands have been received and performs the respective actions. Finally, it transmits a response (which might be an acknowledgment or the answer to a query) and returns.

The \( \text{SPIS_ISR_handler()} \) implements the interrupt service routine of the SPIS module and gets called each time the “Receive-Buffer-Full”-interrupt is triggered, indicating that one complete data byte has been transmitted and received (which happens simultaneously) on the SPI bus. It differentiates between two main situations: the receipt of a new byte of an incoming command packet and

![Flow charts of the most important functions of the framework](image)

**Figure 4.46:** Flow charts of the most important functions of the framework
the receipt of a “zero”-byte, which does not belong to a new packet and is used by the SPI bus master (the PiC USB controller) to indicate its request for the sensor system to transmit data. When called, the function first checks if the receive-buffer is full and clears the interrupt bit. If the check was successful, it reads the received data and stores it at the current position of an input buffer. Now it checks, if the received data byte has been the start-marker byte (which has the value 0xFA) of a new incoming packet and if this is the case, it sets a flag to indicate that the following bytes belong to an incoming data packet.

If a receive-process is started or still ongoing, which is indicated by the respective flag, the position of the input buffer is increased and the function returns. If the in-receive-flag is not set, the receipt of any byte on the SPI bus is assumed to be a request to transmit data. Therefore, the function checks if the first byte in the output buffer is the start-marker of a new packet to ensure that the data packet is ready to be transmitted. Then, it transmits the byte at the current position of the output buffer and checks if this was the last byte of the current data packet. If not, the output buffer position is increased and the function returns; otherwise, the output buffer position is reset to zero and the function checks, if a new data packet is available. If this is the case, it loads the new data in the output buffer before it returns.

In the following sections, the implementation of the sensor system specific operating functions, which are set up on this framework and which use the described communication functions, is explained.

4.4.1.1 Implementation of the ECG Sensor

The ECG sensor system’s main operating mode is defined by the function calcHR_writeSPI(). The flow chart of this function is presented in figure 4.47, together with the flow charts of the two most important algorithms of this sensor system: detectRwave() and calcHR(). It can be seen that an empty data packet is generated at the beginning of the sensor systems generation. This is done to ensure that well defined data (in this case all values are set to zero) is transmitted right from the beginning, since the ECG analysis algorithm needs some time to generate the first set of “true” data and several requests for data might already be received from the USB controller before valid data is available. Afterwards, the main operation loop is entered. This loop is never left unless a critical error occurs.

The first operation of every cycle is the call of the processSPI() function, which has been explained in the description of the framework implementation: it checks for and reacts on commands received on the SPI interface. If the sensor system is not in measurement (or operation, respectively) mode, the loop ends at this point and is repeated immediately. Otherwise, the getDelSigValue() function is called to request a new sample of the input signal. In this mode, the cycle time of the loop is
specified by this function call, as it waits actively until a new data sample has been acquired by the analogue-to-digital converter. In the case of the ECG sensor system, this is done with a sample rate of 200Hz, which means that the main operation loop runs with a frequency of 200Hz, too. As soon as the new data sample has been acquired, the \texttt{filter50Hz()} function is called, which implements a digital IIR notch filter with a rejection band between approximately 49Hz and 51Hz to reduce disturbances caused by main power lines (see [Ste07] for a detailed description of the implementation of this filter).

After being filtered, the signal is handed over to a peak detection algorithm, which is implemented in the function \texttt{detectRWave()}. This algorithm uses a dynamic threshold to detect the R-waves in

\textit{Figure 4.47: Flow charts of the ECG related algorithms}
the ECG signal, which mark the exact time of a heart beat. If the ECG electrodes are placed as proposed in section 4.2.4.1 of the chapter *Theoretical Details*, the R-waves are by far the highest peaks in the signal and can be precisely detected by the dual-edge detection principle: both the time of exceeding and dropping below the threshold are stored and the time of the peak is determined by calculating the middle of this time interval. This principle is, for example, proposed by Ruha et al. [Ruh97] and is well suited for the calculation of the heart rate variability. The threshold itself is adjusted dynamically on the basis of the last maximum and minimum values of the ECG cycle. For this purpose, the (temporarily) stored maximum and minimum values are updated at the beginning of the function, if the new sample is either higher or lower than the current maximum or minimum value. Afterwards, the algorithm differentiates between the two states “waiting for peak to begin” and “waiting for peak to end”. In the first case, the algorithm compares the new data sample with the current threshold: if the sample is higher, the current time is stored and the beginning of the peak is marked and if the threshold has not been reached yet, it is decreased. The amount, about which the threshold is decreased, is dynamic, too: it is proportional to the difference between the minimum and the maximum value of the previous ECG cycle. As these values are updated only when a detected peak ends, the decrease rate is linear, but its slope changes.

In the case that the algorithm has already detected a peak and is waiting for the peak to end, it again compares the new data sample with the (unchanged) threshold. If the signal has not dropped below the threshold after two seconds, the algorithm assumes that it missed the falling edge of the peak, for example due to a drift of the baseline, marks the peak as ended without counting it and returns to the “waiting for peak to start”-state. However, the normal case is that the signal drops below the threshold very quickly. When that happens, the algorithm marks the peak as ended and calculates the time of the middle of the peak in compliance with the dual edge principle. It stores the calculated peak time and the used threshold, as well as the maximum and minimum values that occurred in this last ECG cycle and adjusts the threshold for the next cycle to \( \frac{3}{4} \) of the maximum value. Then it resets the maximum and minimum values and returns “true”, indicating that a peak has been detected.

If a peak has been detected, the function `calcHR()` is called out of the main operation loop. `calcHR()` reads the data stored by the peak detection algorithm, as well as a buffer containing the last 16 calculation results, and calculates the time interval between the new R-wave and the previous one (RR-interval). By building the reciprocal of this time interval and “adjusting” the unit, the heart rate in beats per minute is calculated in the next step (see equation (4.1) in section 4.2.4.1). A validity check is performed on the result of this calculation in order to remove false positive R-wave detections: the heart rate has to be within the range from 30 to 240 beats per minute to be valid. Afterwards, the heart rate is filtered with a Median filter to remove the (natural) beat-to-beat variations, as well
as outliers caused by remaining false negative or false positive detections, from the heart rate value. Then, by using a buffer of the 16 previously recorded RR-intervals, both the RMSSD and the SDRR are calculated and Median filtered with a window size of nine samples. All resulting values are stored and the buffer is updated before the function returns.

Now, the final steps of the main operation loop are performed: the heart rate and heart rate variability calculated by \texttt{calcHR()} is added to the transmit buffer, which is then handed over to the transmit function provided by the framework. Afterwards, a new cycle of the loop begins.

4.4.1.2 Implementation of the Respiration Sensor

The software implementation of the respiration sensor system is to a large extend similar to the implementation of the ECG sensor system. Therefore, no additional flow chart is presented for the respiration sensor related functions. Instead, the differences to the functions of the ECG sensor system are explained in the following.

The function \texttt{calcRespWriteSPI()} corresponds to the \texttt{calcHRwriteSPI()} function of the ECG sensor system and works the exact same way, except for the acquisition of new data samples: in contrast to the ECG sensor system, the values that are acquired by the analogue-to-digital convertor are unsigned and get inverted by subtracting them from 16383, which is the maximum value of the used 14bit ADC, before they are handed over to the peak detector algorithm. This is due to the fact that the output of the Wheatstone bridge is proportional to the resistances value of the respiratory effort transducer, while the resistance itself is in inverse proportion with the increase of the transducer’s length. As the respiratory effort transducer is not vulnerable to irradiations from main power lines, the 50Hz notch filter is left out in this function, too.

The \texttt{peakDetector()} function, which is called out of the main operation loop, is also similar to the one used at the ECG sensor system, except for the fact that the decrease rate of the threshold and the time out for the detection of the falling edge of the peak have been adapted to the (longer) respiration cycles. The same applies to the \texttt{calcRR()} function, which corresponds to the \texttt{calcHR()} function: its validity check accepts respiration rates between 5 and 240 cycles per minute. In addition to the respiration rate, the \texttt{calcRR()} function also calculates the respiration amplitude in terms of the difference between the minimum and the maximum value that occurred in the last breathing cycle. Both the respiration rate and the respiration amplitude are filtered by a Median filter that uses a comparably small window size of five samples.
4.4. Implementation Details

4.4.1.3 Implementation of the Skin Conductance Sensor

For the skin conductance sensor, the software implementation differs from the implementations of the ECG and the respiration sensor systems, as there is no peak detection or similar algorithm necessary for this type of sensor. The `calcSCwriteSPI()` function contains the main operation loop for the skin conductance sensor system. Similar to the corresponding functions of the ECG and the respiration sensor system, this function transmits an empty data packet before it enters the operation loop. This loop processes the SPI bus interface and checks if the sensor system is in measurement mode. If this is the case, it acquires a new data sample from the analogue-to-digital converter and hands it over to a function called `calcSC()`, which is used to calculate both the skin conductance level and the skin conductance response. As this function also differs from the parameter calculation algorithms of the previously described sensor system, its flow chart is presented in figure 4.48.

```
calcSC(wData) : bool
Start
Store last SCL, update SCL with new data
Calculate derived SCR from last and current SCL
Store SCR in ringbuffer
Calculate averaged SCR
Apply Median filter
on SCL and SCR
applyMedianFilter()
Store output values
Return
Store SCR in ringbuffer
```

Figure 4.48: Flow charts of the skin conductance calculation algorithm

The `calcSC()` function reads the skin conductance level (SCL) that has been calculated in the last cycle and stores it temporarily in a different variable, before it updates the SCL with the new data sample. Then it calculates the skin conductance response (SCR) by building the first derivation of the SCL. For this purpose, it calculates the (absolute) slope between the last and the current SCL value and stores the result in a buffer with a window size of ten samples. Afterwards, the samples in the buffer are averaged to get the mean SCR value. Finally, both the SCL and the SCR are filtered by a Median filter with a window size of five samples in order to remove possible outliers, which might be caused by errors during the measurement (e.g. pulling at the electrode cables).
Similar to the other sensor system’s main operation loops, the resulting data is added to the transmit buffer and handed over to the respective framework function.

### 4.4.1.4 Implementation of the Blood Oxygen Saturation Sensor

Due to the fact that the oxygen saturation sensor system acquires its data not by the use of an analogue-to-digital converter, but receives it on an UART-compatible serial data interface, the software implementation of this sensor system differs strongly from the other sensor systems. The flow charts of the important functions are shown in figure 4.49.

The function `processDataUseSPI()` is the equivalent of the `calcXXwriteSPI()` functions of the other sensor systems and is similar to them in its structure, but differs from them when it comes to the acquisition of the sensor data. In the main operation loop, it also processes the SPI bus interface and checks if the sensor system is in measurement mode, in which it reads a flag that indicates if new data is available. If new data has been received, it calls the `getData()` function, which is provided by the “Oxy Sensor” module, to get the new data sample. The data is added to a transmit buffer and transmitted by the usual `sendDataSPI()` function.

The flag, which indicates that new data has been received, is set by the `RX8-ISR-handler()` function, which implements the interrupt service routine of the RX8-module that is used to receive the data transmitted by the oximeter module. After being called, the function first checks if the receive process has been completed to ensure that the data is not changed while it is read. Afterwards, it validates the value of the data, as “zero”-bytes are not counted as new data values. Valid data values are stored in a buffer and the new-data-ready flag is set.

The `getData()` function, which is called when this flag has been set, reads the stored data and clears the flag. It applies a Median filter with a comparably large window size of 25 samples. This large window size is used to remove the outliers that are caused by direct or even environmental influences on the oxygen saturation transducer, such as movement of the subject or a change of lighting conditions.

### 4.4.2 The PhysioLogger Software

The PhysioLogger software is implemented in the form of an event-triggered, object-oriented software. Therefore, its main function is kept very short: it creates the necessary object instances and initialises the application. The methods, which implement the functionality of the software, are then called by events that are triggered either by the user (i.e. by clicking on buttons) or by the USB communication library, which is used by the software: this library provides all methods and settings that are necessary
to specify the used USB device, as well as to communicate with it, and generates events when data packets are sent or received, as well as when the specified device is connected to or disconnected from the computer. PhysioLogger is able to operate simultaneously with both sensor modules, but does also work if only one of the modules is connected.

For the transmission of command packets to the sensor modules, a queue is used to avoid the loss of command packets due to a buffer overflow, which might occur when a new USB packet is received by the sensor system while the controller is still processing a previous packet. For this purpose, specifically defined command objects are generated and added to the queue. The queue itself is processed by an event-triggered (non-blocking) queue management routine, which sequentially transmits the commands and waits for them to be acknowledged by the sensor module before the next command is transmitted. If an error message is received from the USB controller or if no response has been received for 200ms, the last command is repeated (watchdog-like behaviour).

It was already shown in the class diagram of the software, which was presented in figure 4.38, that a
large number of functions has been implemented for the PhysioLogger software. Approximately half of them are used to handle the events triggered by the user in the graphical user interface (GUI), while the other half handles the communication over the USB bus and the storage of the received data. Not all of these functions are explained here, as this would go beyond the scope of this section. Instead, a typical sequence of actions (events, respectively) is described with the help of the flow charts shown in figure 4.50.

The top left flow chart shows the workings of the previously mentioned main() function, which creates the main application form and initialises all objects and variables, including the data buffer for the incoming data packets and the temporary file, into which all received data is written. Afterwards, the software waits for events, which can be thought of as “software interrupts”. Typically, the first event that occurs is the arrival of one of the specified USB devices. This event is triggered by the USB object, which calls the function specifiedDeviceArrived(). The function differentiates between the two sensor modules by analysing the sender object of the event and sets a flag, which indicates that the respective sensor module is connected to the computer. Then, it enables the GUI elements for the control of the respective sensor systems and returns. A similar function has been implemented for the removal of a specified device: this event calls the function specifiedDeviceRemoved(), which has the exact same program flow, except for the fact that it clears the flag and disables the respective GUI elements.

If the user clicks on one of the buttons, for example the “Start” button, an event is triggered by the button object that calls its respective event handler function. In case of the “Start” button, this function adds a start command to the command queue and calls the handleQueue() function. As can be seen in the flow chart of this function, it first ensures that the “watch-timer”, which causes a command to be repeated if it has not been acknowledged for 200ms, is disabled. Then it checks if there are unsent commands in the queue. If this is the case, it checks if there has been a positive response to the last command and removes that command from the queue, if this is the case. Afterwards, it transmits the first command in the queue, which is either a repetition of the last command or a new command. Finally, it enables the watch-timer and returns.

As soon as the reaction of the sensor module to the command is received on the USB interface, a data-received event is triggered and the function onDataReceived() is called. This function creates a temporary data string, into which either the command or the data is written. It differentiates between data packets and information and command packets. The response of the sensor system to the previously sent command is the latter one. After the function has checked that the sender ID and the command are valid, it notifies the queue handler of the arrival of a response by calling the handleQueue() function with the command as parameter. handleQueue() then removes the
Figure 4.50: Flow charts of the PhysioLogger software
command and transmit the next one, if there is a command object left in the queue. Then, `onDataReceived()` reacts on the response according to the type of command: an acknowledgment message to the “Start” command, to follow the previous example, would simply be printed in the message box of the graphical user interface, while a response to the “Get Gain” command, for example, would cause the function to update the gain-selector combobox in the user interface in addition.

If the measurement system is in the running state, the sensor module transmits ten new data packets per second, which each trigger a call of the `onDataReceived()` function. In this case, the function splits up the received data according to the specified data packet protocol (see section 4.3.2.2 in the chapter *Technical Details*). If the OxyModule was the sender of the data packet and the PhysioModule is connected, too, the function returns directly. Otherwise, the updated data string is complemented by a time stamp and stored in the data buffer and (simultaneously) in a temporary file on the hard disc, which is automatically deleted as soon as the data is correctly saved to a user specified file.

To synchronise the physiological sensor readings with other simultaneously recorded data, such as CUELA Activity measurements or a video recording, the PhysioLogger provides a synchronise button. If it is clicked by the user, a synchronisation marker with a time stamp is written to the data file immediately. Pressing a similar synchronisation button at the CUELA system simultaneously or recording the moment of the button press on video allows to synchronise the different data streams.

**4.4.3 The Stress Generator Software**

Similar to the PhysioLogger, the implementation of the Stress Generator software is based on event-triggered methods. Once the user has configured and started the software, it displays one single or two exercises simultaneously at a time (that is, one math and one colour exercise) and waits for either the user to enter the correct solutions or the timer to run out, before it generates and displays the next set of exercises. Therefore, the two important functions of the software are `newExercise()` and `timerTick()`. The flow charts of these two functions are presented in figure 4.51. The other functions, which were shown in the class diagram of the Stress Generator in figure 4.40, are used to handle the graphical user interface only and are not described here.

The function `newExercise()` is called when the user starts the program by clicking the “Start” button. It checks if the exercise type “Math” is enabled and generates a respective exercise. Afterwards, both the randomly generated operands and the associated operation are displayed on the GUI and the focus for keyboard inputs is set to the solution text box. Then, it checks if colour tests are enabled and if this is the case, it generates a new colour exercise and displays the randomly selected colour name with an also randomly selected font colour, as well as the randomly ordered solution buttons.
In every case, it sets the solution time according to the user defined value, resets the progress bar and starts the timer.

Each second that passes, the timer generates a “tick” event and calls the `timer_tick()` function. The function decreases the remaining solution time and checks, if it is still greater than zero. If this is the case, it updates the progress bar and returns. If the remaining time has just run out, it queries the solution of the math-exercise and presents it. Otherwise (if the remaining time is already less than zero), it calls `newExercise()` and starts a new cycle.

However, in most cases the user will have entered a correct solution before the solution time runs out. In this case, `newExercise()` is called immediately. As the software is intended to generate stress in the user, it does not allow him to rest.
4.4.4 Integration into WIDAAN

This section describes the approach to integrate the acquired physiological data into the WIDAAN software. As it was not possible to implement this approach during the scope of this work, this section is only an analysis of the feasibility of the approach.

The physiological data, which has been stored by the PhysioLogger software, can be imported as new sensor channels in WIDAAN. A respective input filter algorithm, which is necessary for this purpose, is already available and in use for the integration of the heart rate data that is used in the CUELA Activity system: in the current version of the Activity system, the heart rate is acquired by the use of a Polar watch, which stores the data in a file with the “comma-separated-values” (CSV) format. As an equivalent file format is used by the PhysioLogger, only slight modifications are necessary on the import filter algorithm.

To synchronise the different data channels with each other or with a video recording, synchronisation-markers can be set in each channel. If the sample rate of each data channel is known, one such marker suffices to calculate the offsets of the channels. Otherwise, two markers have to be set (e.g. at beginning and end of measurement) in order to synchronise the channels: in this case, WIDAAN automatically resamples the new data channels according to the selected time interval. In addition to the synchronisation process, the markers can be used for other purposes, too: single markers can be set in order to mark a specific point in time or a pair of markers can be used to mark a time interval of arbitrary length. This feature can be used to name and specify (or classify, respectively) certain time intervals according to the type of activity that has been performed, as well as to further analyse the respective time interval (see [Her03]). Figure 4.52 shows the use of this feature in WIDAAN.

For this work, the marker feature can be used to calculate the baseline of each measurement by measuring the subject’s physiological data at a (sufficiently long) resting period and classifying this time interval manually as a resting period. A similar interval can be specified for an interval with a maximum of physical activity (or physical strain, respectively). By using both intervals, the physiological data can be “normalised” in order to allow a person-independent analysis of the data. For this purpose, a functionality of the EMG-plug-in of WIDAAN can be used without much effort, as it performs almost the same operation.

After the normalisation of the physiological data, the classification of the data has to be performed. For this purpose, a completely new plug-in will be integrated into WIDAAN, which implements the decision rules that have been calculated by the used decision tree learning algorithm as described in chapter 4.3.3.2. By applying the decision tree algorithm on the acquired data, a new “user-defined” data channel will be created in WIDAAN, which contains the resulting strain classification according to the classes that were specified in chapter 4.2.5.1.
Figure 4.52: Setting of markers in the WIDAAN software
5 Evaluation

In this part of the work, the developed measurement system is evaluated with respect to the technical implementation of the different sensor systems, as well as to the functionality of the complete system in practice. Additionally, the proposed approach for the classification of the acquired data is evaluated by presenting and discussing the results of the performed measurements as well as the performance of the different learning algorithms on this data.

5.1 Technical Evaluation of the Components

This section evaluates the technical implementation of the different sensor systems, which are the basis of the measurement system. First, the used analogue low-pass filter module is evaluated by systematic tests, followed by a practice-oriented evaluation of the digital notch filter algorithm. Afterwards, the timing of the time-critical modules and algorithms of the sensor systems is evaluated by the use of an oscilloscope. Finally, the reliability of the sensor modules’ data transmission is tested.

5.1.1 Evaluation of the Filters

The analogue low-pass filter of the ECG sensor system has been implemented by the use of the PSoC’s LPF user modules. Two such modules are positioned consecutively. The performance of these filter modules has been evaluated in detail in previous work (see [Ste07]) and was found to be sufficient for the use in the sensor system. In figure 5.1, the output signals of the two filter stages are shown for two different input signals: a sine wave with a frequency of 70Hz, which is twice the filter modules’ cut-off frequency, and a square wave with a frequency of 20Hz. It is shown that the sine wave is strongly attenuated in the output signal. In contrast to this, the amplitude of the square wave is maintained, although the sharp edges of the square wave are smoothed.

The digital infinite impulse response (IIR) notch filter algorithm that is used in the ECG and the skin conductance sensor systems has been implemented on the basis of formulas proposed by Smith ([Smi99]). In previous work ([Ste07]), it has been compared to a filter algorithm developed by the use of Matlab. It was shown that the performance of the Matlab-derived algorithm was slightly better
with respect to the attenuation of the 50Hz noise, but unfortunately, the filter algorithm changed the typical waveform of the ECG and made it more difficult to detect the R-waves. Therefore, the algorithm proposed by Smith was the better choice for the sensor systems. Figure 5.2 presents a typical input and a filtered output signal in order to demonstrate the performance of the filter algorithm: the output signal shown in figure 5.2 (b) is noticeably smoothed, as the 50Hz noise, which is visible between the peaks in the signal shown in figure 5.2 (a), has been removed.

5.1.2 Evaluation of the Sensor Systems’ Timing

Besides the filtering of the physiological signals, the timing of the digitalisation (that is, the sample rate) has to be absolutely exact in order to reliably acquire the signal and to calculate the time-dependent parameters. Additionally, the real-time clock of the sensor systems needs to be sufficiently
5.1. Technical Evaluation of the Components

Figure 5.3: Evaluation of the sensor systems’ timing conditions

Figure 5.3 shows that the pin, which is driven by the ADC of the ECG sensor system, is toggled with a frequency of 100.5Hz, indicating a sample rate of 201Hz. The sample rate of the respiration sensor system, which is evaluated in part (b) of the figure, amounts to 98.6Hz, and the sample rate of the skin conductance sensor system (shown in part (c)) to approximately 10Hz. These values differ slightly from the specified sample rates, which are 200Hz for the ECG sensor system, 100Hz for the respiration sensor system and 10Hz for the skin conductance sensor system. However,
as the deviations are constant (the sample rates did not vary during the measurement), they can be neglected. The evaluation of the real-time clock frequency is shown in part (d) of figure 5.3. It can be seen that the desired frequency of 1kHz is closely approximated, as the actual clock frequency amounts to 1.002kHz. Due to this slight deviation, the time measured by the sensor systems differs about 0.2% from the actual time, resulting in a difference of 0.12 seconds per minute. This deviation can also be neglected for the proposed use of the sensor systems.

5.1.3 Evaluation of the Data Transmission

The reliability and the timing of the data transmission, which is controlled by the PiC USB controller, has been evaluated by counting the number of received data packets over a time of 44 minutes and 27 seconds. As a data packet should be received every 100ms, ten data packets should be received per second, resulting in a total of 26670 data packets over the complete time interval. This number was perfectly met, as the data file contained 26670 lines of data after the additional synchronisation markers had been removed.

The evaluation of the data transmission concludes the technical evaluation of the sensor modules. In the following section, a practice-oriented test of the complete measurement system is presented in order to evaluate the functionality and operation of the system.

5.2 Functional Evaluation of the System

In order to perform a functional evaluation of the developed measurement system, a measurement in the FIVIStress environment was arranged. FIVIS, which stands for the German title Fahraddrahsimulator in der Immersiven Visualisierungsumgebung “Immersion Square”, is a bicycle simulator in an immersive virtual reality environment that has been developed at the University of Applied Sciences Bonn-Rhein-Sieg. FIVIStress is an application for this environment that has been developed by Scherfgen ([Sch08a]). FIVIStress allows a user to ride the bicycle through a virtual copy of the city Siegburg and is able to simulate traffic and to generate obstacles in order to induce psychological (and especially important: emotional) strain in the user. As different levels of physical strain can be induced simultaneously by the activity on the bicycle, FIVIStress is a well-suited environment to evaluate the combined strain measurement system.

For the measurement, an experimentee (male, 23) was chosen, who already knew the FIVIS simulator. This way, additional strain that might have been caused in a subject that uses the simulator for the first time (e.g. excitement or anxiety) could be avoided. As it was shown that the blood oxygen saturation sensor fails to acquire reliable sensor readings (see section 5.3.1), it was not used in this
5.2. Functional Evaluation of the System

Figure 5.4: Measurement setup for the evaluation with the FIVIS system

measurement setup. The FIVIS environment and the measurement setup are shown in figure 5.4. Furthermore, the figures 5.5 and 5.6 present the results of the measurement, which are explained in the following paragraph.

In the first minutes of the measurement, the subject was sitting quietly on a chair. Afterwards, it started riding the bicycle until the end of the measurement. As can be seen in figure 5.5 (a), the physical activity of the subject changed during the measurement due to different cycling speeds. It can be noticed that the heart rate, which is shown in figure 5.5 (b), is strongly correlated with the physical activity intensity (PAI) index. The heart rate variability is varying strongly during the resting phase. In contrast to this, it decreases significantly as soon as the subject begins to cycle and it remains constantly low until the end, except for a few short peaks, which are correlated to a decrease of the heart rate at the same time.

The results of the EMG sensor, which are presented in figure 5.5 (c), show only low activity on both channels (that is, on both shoulders) for the short resting phase at the beginning of the measurement. During the cycling phase, the muscle activity seems to be correlated with the physical activity to a certain extend. The same applies to the results produced by the respiration sensor, which are shown in figure 5.6 (a): the progress of the respiration rate is at a large extend similar to the physical activity intensity. The respiration amplitude is almost constant throughout the cycling phase. This shows
(a) Results of the Activity system

(b) Results of the ECG sensor system

(c) Results of the EMG sensor system

Figure 5.5: Results of the measurement with the FIVISstress system (1)
that the transducer was properly fixed and did not change its position during the measurement.

The output of the skin conductance sensor (figure 5.6 (b)) shows an almost linear increase of the skin conductance level during the measurement. This was most likely caused by sweating of the subject. However, the skin conductance response is not influenced by this and shows a clear difference between the resting phase and the cycling phase, but in contrast to the other parameters it does not seem to be correlated with the physical activity.

Altogether, the measurement results indicate that the developed physiological sensor systems are working reliably, except for the oximeter, which can not be used when the subject is moving intensively. Additionally, the FIVIS system proved to be suited for the induction of strain in the subject, although the physical component seemed to be clearly superior to the psychological component. This might be compensated by reducing the physical workload of the subject, for example by lowering the pedal’s resistance, which was very high at the bicycle used and could not be lowered at that time.
5.3 Evaluation of the Data Classification

This section presents and evaluates the results of the measurements that have been performed in order to research algorithms for the classification of the acquired data with respect to the proposed strain classes (see section 4.2.5.1 in the chapter Theoretical Details). In addition, the results of the applied machine learning algorithms are presented and discussed.

5.3.1 Results of the Measurements

By the use of the previously described measurement setup (see section 4.3.3.1 in the chapter Technical Details), four measurements have been performed within the scope of this work. Unfortunately, one measurement, which was performed on a 27-year old female, did not produce usable data, as it was not possible to find a suited place for the electrodes and the ECG signal was too weak to properly be analysed. The reason for this failure and possible solutions are discussed in the Conclusion. Of the remaining three measurements, two have been chosen to be presented and discussed in the following.

Results of Subject A

The first measurement was performed on subject A (27, male). The acquired sensor data of this measurement is shown in figures 5.7 and 5.8. For each sensor an own graph is shown that presents the values of the different parameters, which have been calculated by the respective sensor system, with respect to the time. Due to a failure in the oxygen saturation sensor system at the time of approximately 35 minutes, a break of almost 14 minutes appears in the time line. After this break, the measurement was resumed and the different sensor systems were synchronised again. The transparent blue areas, which can be seen on each graph, mark the time intervals of the different strain induction scenarios. The numbers in the top left corners of these areas represent the respective strain class. In the following, when it is spoken of “interval two”, for example, this means the time interval in which the scenario for the induction of strain class two was performed. It can be seen in the figures that the classes six and eight were left out during this measurement due to a limited timeframe of the subject and the delay, which was caused by the failure in the sensor system.

Figure 5.7 shows the graphs for the physical activity intensity (PAI), which has been calculated by the WIDAAN software using the data that has been acquired by the CUELA Activity system, the ECG parameters heart rate (HR) and heart rate variability (HRV:RMSSD and HRV:SDRR), as well as the muscle activity recorded by the EMG sensor system. The muscle activity is expressed by the summed up and normalised EMG readings for the left and the right shoulder.

It is easy to see that the physical activity intensity (figure 5.7 (a)) is well suited to differentiate between quiet sitting, sitting and working, as well as bicycling. To distinguish between low and high
5.3. Evaluation of the Data Classification

(a) Results of the Activity system

(b) Results of the ECG sensor system

(c) Results of the EMG sensor system

Figure 5.7: Measurement results of subject A (1)
additional mental strain is, however, not reliably possible. The actual physical strain is also not visible in the results: when comparing the time intervals five and nine, a noticeable difference should have been shown, as in interval nine the (physical) workload has been much higher for the subject, but no such difference can be seen. The reason for this is that the CUELA Activity system assesses the activity of the subject in terms of the intensity of motions - unfortunately, a higher workload does not automatically lead to more intense motions.

The readings of the ECG sensor, which are presented in figure 5.7 (b), show a minimum for the heart rate and a maximum for the heart rate variability during the resting phase (class one). The readings for the intervals two and three, in which psychological strain was induced, are very similar to each other, but show noticeable differences in comparison to the resting phase: the heart rate is (in average) about 20 beats per minute higher, while the heart rate variability is about 20ms lower. When physical strain is induced instead of psychological strain (class four), the heart rate stays almost the same as in the intervals two and three, while the heart rate variability is slightly increased. This changes in interval five, in which light physical and psychological strain are combined, resulting in an increased heart rate and a lower heart rate variability. With the further increase in physical strain, beginning with interval seven, the heart rate is increased again, while the heart rate variability is decreased even more. The maximum for the heart rate is reached in interval nine with more than 130 beats per minute. All of these readings correspond to the literature (compare to section 2.1.2).

The normalised EMG sensor readings (figure 5.7 (c)) show almost no activity in the trapezius muscle during the resting phase. In contrast to this, the activity is much higher during the interval two and increases even more in interval three. During the performance of scenario four, the activity is very low, except for some short peaks. With the increase of both physical and psychological strain during the intervals five to nine, the muscle activity acquired by the EMG also increases again.

The measurement results of the respiration, the skin conductance and the oxygen saturation sensor systems are presented in figure 5.8. In part (a) of the figure, the respiration rate and the respiration amplitude, which were both acquired by the respiration sensor, are shown. It can be seen that the respiration amplitude is varying over almost the complete range (approximately from 5 to 95 percent). The reason for this effect is that the readings of the respiratory effort transducer are strongly influenced by the exact position of the chest strap, as well as the body posture of the subject. Unfortunately, both the body posture and the position of the chest strap change during the measurement due to the motions of the subject, leading to completely useless measurements for the respiration amplitude, as well as to errors in the measurement of the (more important) respiration rate. Two obvious examples for errors in the respiration rate can be seen in the time range of approximately 26 to 30 minutes, as well as from 52 minutes until the end of the measurement, as it is very unlikely that the subject actually had a (constant) respiration rate of 120 breathing cycles per
5.3. Evaluation of the Data Classification

(a) Results of the respiration sensor system

(b) Results of the skin conductance sensor system

(c) Results of the oxygen saturation sensor system

Figure 5.8: Measurement results of subject A (2)
minute and more. Due to these problems, the validity of the respiration rate is questionable, although there are time intervals, in which it seems to be convenient. It is slightly increased during scenario three compared to scenario two. During the bicycle riding (intervals four to nine), it is increased even more and varies much, but unfortunately, this might as well be caused by the subject’s motion.

The readings of the skin conductance sensor system, namely the parameters skin conductance level (SCL) and skin conductance response (SCR), are shown in figure 5.8 (b). It can be seen that the skin conductance level slightly increases during the first (approximately) 20 minutes of the measurement. This can be lead back to the consecutively increasing contact of the electrodes with the subject’s skin due to the increasing warmth of the electrode’s isotonic contact gel, as well as sweating of the subject. However, the derived response of the skin conductance is more of interest, as (according to the literature) it is better suited to show the reactions of the subject’s body to psychological strain. During the resting phase (interval one), no high responses are visible in the graph. This changes in the time interval two: three high peaks are shown between the minutes 12 and 13. As the video recording does not show any motion or behaviour of the subject during this time that might have caused erroneous sensor readings, these peaks seem to be accurate. During the performance of scenario three, the SCR value varies more and is slightly increased in comparison to the values that were measured during scenario two. The readings in interval four are also slightly increased, but do not show any noticeable peak values. In contrast to this, very high peak values are shown in the readings during interval five. Again, no obvious cause for erroneous sensor readings can be found by reviewing the video recording. During the intervals seven and nine, the readings are still increased when compared to interval four, but no further “high peaks” have been recorded.

In figure 5.8 (c), the results of the blood oxygen saturation sensor system are shown. While the subject was sitting on the chair, the readings were very constant at about 98% oxygen saturation. Unfortunately, as soon as the subject starting riding the bicycle, the sensor failed to acquire the oxygen saturation correctly. Depending on the way the subject held his left hand (to which the sensor was attached), readings that could be assumed to be correct were shown. However, the sensor was still not able to work reliably. In the time range from (approximately) 33 minutes until the end of the measurement, the readings seem to be correct and show an oxygen saturation of between 91% and 96%, but unfortunately it can not be determined if they are actually valid.

**Results of Subject B**

For the measurement of subject B (male, 35), the blood’s oxygen saturation was not acquired anymore, as it was shown in the previous measurement that the used oximeter is too easily distorted by the subject’s movement. Additionally, it proved to be useless for the detection of psychological strain, as it did only react to the subject’s physical strain. However, physical strain can be acquired
better by the use of the CUELA system. Besides this reduction of sensors, the measurement setup was identical and all proposed strain induction scenarios were performed for approximately five minutes each. The results are shown in the figures 5.9 and 5.10. Again, the respective strain classes are marked in the graphs.

The physical activity intensity, which is shown in figure 5.9 (a), shows (almost) no activity during the resting phase (interval one) and only minor activity during the induction of psychological strain (intervals two and three). This activity results mainly from the fact that the subject had to use the keyboard and the mouse of the computer to solve the presented exercises. During the cycling phase (intervals four to nine), the activity is at a significantly higher level. It is noticeable that the activity slightly increases between the intervals of the different strain induction scenarios, four and five. This increase is most likely caused by a slip of the upper leg sensor straps towards the knee. During interval seven, the subject drew up the sensor straps again, resulting in an immediate decrease of the activity value to the previous level. Between the intervals six and seven, a gap can be noticed in the graph, which resulted from the short pause that was necessary to increase the pedal resistance of the bicycle ergometer.

Figure 5.9 (b) presents the graphs of the heart rate and the heart rate variation. This time, only one of the two parameters for the HRV is shown, as the second parameter does not provide any additional information. During the time intervals of the induction of psychological strain (intervals two and three), the heart rate is slightly increased when compared to the resting phase. In contrast to this, the heart rate variation is (against expectation) not lower than during the resting phase. This changes with the beginning of the cycling phase: while the heart rate is slightly increased throughout the intervals four to six, the heart rate variation is significantly lower than during the first three intervals. During the intervals seven to nine, the heart rate almost constantly increases up to a maximum of more than 140 beats per minute in interval nine, while the heart rate variation decreases more and more.

The results of the EMG sensor (figure 5.9 (c)) show no significant activity during the time intervals for the classes one to six. Only for the last three time intervals an increase in activity can be noticed. The highest peak in the graph, which can be seen at the end of interval seven, was most likely caused by the previously mentioned correction of the upper leg sensor straps’ positions by the subject.

The measurement results of the respiration sensor, which are presented in figure 5.10 (a), show (on average) a slight increase of the respiration rate during intervals two and three, as well as a significantly higher variation, when compared to the resting phase. In interval four, the respiration rate is comparably constant on an elevated level. This changes again beginning with interval five: on average, the rate is not noticeably increased, but it varies strongly. In the intervals seven to nine,
(a) Results of the Activity system

(b) Results of the ECG sensor system

(c) Results of the EMG sensor system

Figure 5.9: Measurement results of subject B (1)
the variation remains almost the same, but the actual rate increases more and more. The respiration amplitude is very small during the first three intervals compared to the rest of the measurement. The reason for this effect is the position of the chest strap, which obviously was much better positioned during the cycling phase.

Similar to the results of the measurement with subject A, the skin conductance level increases slowly during the measurement, as can be seen in figure 5.10 (b). The skin conductance response is noticeably increased during the cycling phase. Besides this, it does not seem to provide information that can be used to distinguish the different levels of psychological strain.

On the basis of these two measurements, different classification algorithms were tested in order to find a way to classify the data with respect to the (induced) strain class. The parameters heart rate, heart rate variation (RMSSD), respiration rate, skin conductance response, root mean square (RMS) of the EMG signal and the physical activity intensity were used for this purpose. The standard deviation (SDRR) of the heart rate variation was left out as it does not provide additional information.
compared to the parameter RMSSD, while the respiration amplitude, the skin conductance level and
the oxygen saturation level did not provide any interesting information at all. The results of this
approach are presented and discussed in the following section.

5.3.2 Results of the Learning Algorithm

The measurement results that were presented and discussed in the previous section are now used to
evaluate different learning techniques. This is done in order to find the optimal classification algorithm
for the provided data. The optimal algorithm is the one that correctly classifies a maximum of input
values, while requiring a minimum of classification rules. In addition, the algorithm should not learn
the measurement-specific details and possibly untypical data items of the training data. This effect
is called “overfitting” (see [Mit97]). Due to the characteristics of the data at hand, which has been
described in section 4.2.5.1 of the chapter Theoretical Details, these specific details will most likely
contain errors and misclassified data items. If these errors are learned by the algorithm, it will
reproduce them every time it is applied on new data.

Besides the already mentioned decision tree learning algorithms, three other learning techniques are
applied on the data of both measurements: the Bayesian Network model, the Naive Bayes classifier
and the Multilayer Perceptron model, which is a special type of a neural network. For the decision
tree learning algorithms, two different kinds of algorithms are chosen: the first is the J48 algorithm,
which is a Java implementation of the state-of-the-art algorithm C4.5 revision 8 (see [Wit05]). It
is applied with three different levels of pruning in order to gain decision trees with different sizes.
The second algorithm is the REPtree algorithm, which implements a simpler decision tree learning
algorithm with a selectable maximum tree depth. It also is applied three times with different settings:
with unlimited depth, with a maximum depth of eight levels and finally with a maximum depth of
five levels. Each learning technique is tested with the stratified 10-fold cross-validation method,
as described in section 4.3.3.2 of the chapter Technical Details. The resulting correct classification
rates as well as the tree size (only for the decision tree algorithms) of each algorithm are shown and
compared in figure 5.11.

The results indicate clearly that the decision tree algorithms provide the best results for the clas-
sification of this kind of data. The difference in the results of the two subjects can be explained
by the fact that no data for the classes six and eight was acquired in the measurement of subject
A. Nevertheless, the J48 algorithm configured with maximum tree size clearly is in the first place
for both subjects and consists of 705 nodes for subject A and 781 nodes for subject B. It is closely
followed by the REPtree algorithm, which produces slightly smaller trees with the used setting “un-
limited depth” (461 nodes for subject A and 763 nodes for subject B). However, the complexity of
these tree models is very high, as can be seen in figure 5.12, which shows the resulting tree of the J48 algorithm applied on the data of subject A. Due to the complexity of these trees, it is very likely that too specific details of the data had been learned and the effect of overfitting will be noticeable when data from a different subject will be classified with the resulting rules. Therefore, smaller trees might be preferable, although their correct classification rates are slightly lower. The smallest trees are produced by the REPtree algorithm when it is configured to use a maximum depth of five levels. This way, the resulting tree has a size of 35 nodes for subject A with a correct classification rate of 89.8% and a size of 49 nodes with a correct classification rate of 81.7% for subject B. Figure 5.13 shows the resulting tree for subject B in order to give an impression of the actual size in comparison to the much more complex tree in figure 5.12.

To further evaluate the different learning techniques, it is tested how well the classification algorithms learned with the data of one of the subjects perform on the data of the other subject. For this purpose, a Bayesian network, a full J48 tree and a six-level REPtree are learned on the basis of the data acquired from subject A. Then, the data of subject B is used as “test set” for the learned algorithms. The results are the following: the Bayesian network classifies 37.5% of the data items correctly, the full J48 algorithm 42.5% and the REPtree achieves 45.2% correctly classified data items. The reason for the better result of the REPtree might be that the (much bigger) J48 tree
suffered from the overfitting-effect, as it learns too many details of the data acquired from subject A. To validate this theory, the test is run again with a bigger REPtree with unlimited depth-levels. As expected, this bigger tree achieves only 42.3% correctly identified data items.

It is important to notice that the actual performance of the algorithms in this test is not representative, as the data set of subject A does not include data items for the classes six and eight, as well as only a minor amount of data items for class seven. Additionally, none of the physiological parameters has been normalised for this test and therefore, the data items of the two subjects differ strongly and cannot be compared easily. Nevertheless, the results can be used to compare the different learning techniques with each other.
Finally, the data sets of both subjects are combined in order to find a classification that can be used for both of them with better results than before. This time, only the decision tree algorithms J48 and REPtree are applied on the joint data set. The results are presented in figure 5.14. It can be seen that the correct classification rate is almost identical to the results of the algorithm for just one single data set. Unfortunately, this is achieved by a significant increase of the trees’ sizes. The J48 algorithm configured with maximum tree size produces a tree with an almost doubled number of nodes. The same applies to the REPtree algorithm with unlimited depth. However, if the depth size of the REPtree algorithm is limited, the size of the resulting trees is only moderately increased: with a maximum depth of eight levels, the size increases from 179 to 231 nodes, and with a maximum depth of five levels, the size increases only slightly from 49 to 53 nodes.

The results of this evaluation show that decision tree learning algorithms are well suited to classify the acquired data with respect to the class of the induced strain with a sufficiently low error rate. Even if the size of the trees is limited to a comparably low number of nodes that can easily be implemented (for example directly in the WIDAAN software), their performance is still above that of naive Bayes classifiers or Bayesian and neural network models. The combination of data sets acquired from different subjects does not lower the correct classification rate noticeably, although the parameters have not been normalised. By applying a decision tree learning algorithm on a combination of normalised data sets from a sufficiently large number of subjects it might be able to produce a tree that will achieve high correct classification rates even on data from new subjects that it has not been trained with.


6 Conclusion and Recommendations

This work addressed the problem of measuring both psychological and physical strain in humans, especially in employees at their workplaces, and of differentiating between the types of strain the subject encountered. A literature review was performed in order to research the state of the art in the fields of strain measurement systems in general and the current sensor technology for the measurement of strain in specific. Additionally, an overview over the results of recent studies in this field was given and it has been shown that there is still room for improvement in this field of research.

On the basis of four intelligent sensor systems, which were developed in previous research projects by the author, a combined measurement system for the acquisition of physiological parameters has been designed and implemented. For this purpose, the intelligent sensor systems were further improved and combined on two small and lightweight sensor modules, which are embedded in robust casings and can be connected to any PC or to a handheld computer via a simple USB interface. The “PhysioModule” contains the sensor systems for the acquisition of the ECG, the respiration and the skin conductance, while the “OxyModule” contains the sensor system for the measurement of the blood oxygen saturation only. These sensor modules can be used either independently or simultaneously.

In addition, a PC-based software tool has been developed for the control of the sensor modules. This software, which is called “PhysioLogger”, is also used to store the data that is acquired and analysed by the individual sensor systems on the computers harddisk.

The results of the technical evaluation showed that the sensor modules are working correctly. Furthermore, a number of tests with a more practical orientation were performed with the complete measurement system and showed that all sensor systems produce good results and are working reliably. However, it was found that the chest strap, which is used for the respiration sensor system, gets out of place easily, as it has to be attached very loose in order to provide a sufficiently wide output range. Therefore, it has to be fixated by safety pins or similar means. Additionally, due to the loose fixation it is often influenced by the subject’s movement, resulting in erroneous measurements. The use of a different chest strap, which allows to be tightened stronger without limiting the output range too much, might remedy this problem.

A similar problem was found in the blood oxygen saturation sensor: the Biopac oximeter, which has been used for the acquisition of this parameter, fails to measure the oxygen saturation level
reliably if the subject is in motion. Therefore, the oxygen saturation can not be assessed in the given context of this work. Fortunately, this is not an issue, as the oxygen saturation level does not seem to be influenced by the psychological strain of the subjects. Therefore, no useful information for the differentiation of psychological and physical strain can be gained by the measurement of the oxygen saturation level.

Another problem was met during the measurement of a female subject: it was not possible to position the ECG electrodes on the subject’s chest in a way that would produce a usable ECG signal. This might be a general problem when dealing with female subjects, as the amount of connective tissue in the chest area, which attenuates the electric voltages of the ECG signal, is (typically) much higher for female than for male subjects. However, it should be able to solve this problem by optimising the used filters and the QRS detection algorithm with respect to this specific situation.

The proposed measurement system can also be used in conjunction with the CUELA system in order to acquire both physiological and physical parameters of a subject. Additionally, the EMG sensor system, which is available as an extension of the CUELA system, can be used to assess the activity of the subject’s trapezius muscle. All acquired data can easily be combined by the use of synchronisation markers that are added to the data streams. In order to automatically classify the combined data with respect to the type of strain that the subject encountered, the use of machine learning algorithms has been proposed. For this purpose, nine different strain classes have been specified and respective strain induction scenarios have been included in the design of a measurement setup that allows to induce psychological and physical strain separately, as well as simultaneously. This measurement setup has been performed by several subjects in order to gain training data with a known combination of strain for the use in the learning algorithm.

During these measurements, two minor drawbacks of the measurement setup were found, that are recommended to take care of for future measurements: first, all subjects complained that the keyboard of the notebook computer, which was running the StressGenerator software, did not have a numeric keypad. According to the subjects, the necessary time to enter the solution to the mathematical exercises would have been lower if they could have used such a keypad. In this context, one of the subjects mentioned that his way of solving the math exercises was slightly different when he had to speak out the result instead of typing them. To avoid this, an assistant that enters the results for the subjects should be used in both the sitting and the cycling phase.

By the use of the Weka software, which provides a collection of state of the art machine learning algorithms, different types of algorithms were applied on the acquired training data. It was shown in the evaluation that the decision tree learning algorithms deliver the best performance for the classification of the data at hand. Correct classification rates of more than 90% can be achieved
even by comparably small decision trees, which could easily be implemented in an analysis software. However, to achieve reliable classification results for unknown data that has been acquired from a new subject, using a tree that has been learned with the data from only one subject is not sufficient. This has been shown in evaluation, as the correct classification rate of a decision tree that was learned from only one subject’s data was decreased to 45% when it was applied to the data of a different subject. In contrast to this, a decision tree that was learned with data from both subjects achieved a constantly high classification rate, although the data of the two subjects differed strongly and had not been normalised. In order to find a decision tree that can be applied successfully to the data of arbitrary subjects, training data from a large number of subjects has to be collected. In addition, this training data has to be normalised appropriately to ensure that the data of different subjects can be compared independently from the personal fitness of the subject. Otherwise, the variations in the physiological parameters of the subjects might be too strong to maintain a high correct classification rate, as the distinctive features of each class will be softened more and more with an increasing amount of training data.

It has been described in this work that the proposed measurement system will be integrated into the CUELA Digital system, which is currently developed. This new system is also based on intelligent sensor modules that are connected via USB interface to a handheld computer, which serves as data-logger and control device, and allows connecting the physiological sensor modules simultaneously. It was shown that it is easily possible to import the data that is acquired by the physiological sensor system into the WIDAAN analysis software. It is also possible to implement the proposed data classification algorithm directly in WIDAAN by the use of a plug-in. This can be done in two different ways: one approach is to implement a static decision tree in WIDAAN that has been learned with a sufficient amount of appropriately normalised training data from different subjects. This approach requires additional research in order to find convenient ways of normalising the data, as well as the design of respective calibration phases that have to be performed by each new subject. The second approach is to implement a complete decision tree learning algorithm that learns a new decision tree for every new subject. This would require to extend the calibration phase that has to be performed by each subject, as it has to include all specified strain induction scenarios, but in contrast to the first approach, the data does not have to be normalised and it is not necessary to collect training data from a large number of subjects in advance. Additionally, the results of the classification algorithm might be better, as the distinctive features of each class are based on the specific subject’s data only. However, the feasibility of both approaches has to be evaluated in detail to find the better suited way of integrating the classification algorithms in WIDAAN.

With the results of this work, an appropriate basis has been provided for the further development of this combined measurement system. Nevertheless, there are still some tasks left for following work.
A very promising aspect, for example, is the (more intensive) use of the FIVIStress system for the further research of the classification algorithms. Theoretically, the FIVIStress system provides the possibility to directly induce emotional strain in subjects, which is not easily possible by different measurement setups like the one used in this work. However, if this can be used in practice has still to be evaluated.
Appendix

Contents of the enclosed CD-ROM

<table>
<thead>
<tr>
<th>Directory</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Circuit-diagrams</td>
<td>EAGLE-Project files with the circuit diagrams and board layouts of the sensor systems</td>
</tr>
<tr>
<td>Software</td>
<td>Source code and binaries of the following software:</td>
</tr>
<tr>
<td>– PhysioLogger</td>
<td>Setup and source code of the measurement control software</td>
</tr>
<tr>
<td>– PSoC Firmware</td>
<td>Source code of the developed software for the PSoC sensor systems</td>
</tr>
<tr>
<td>– StressGenerator</td>
<td>Setup and source code of the psychological strain induction software</td>
</tr>
<tr>
<td>Thesis</td>
<td>Electronic version of the master thesis in pdf format</td>
</tr>
</tbody>
</table>
List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADC</td>
<td>Analog-to-Digital-Converter</td>
</tr>
<tr>
<td>AGND</td>
<td>Analog Ground (internal reference voltage of the PSoC)</td>
</tr>
<tr>
<td>AHR</td>
<td>Additional Heart Rate</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BGIA</td>
<td>Institut für Arbeitsschutz (Institute for Occupational Safety and Health)</td>
</tr>
<tr>
<td>CUELA</td>
<td>Computer-unterstützte Erfassung und Langzeit-Analyse</td>
</tr>
<tr>
<td>DAC</td>
<td>Digital-to-Analog-Converter</td>
</tr>
<tr>
<td>DIP</td>
<td>Dual Inline Package</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processing</td>
</tr>
<tr>
<td>ECG</td>
<td>Electrocardiogram</td>
</tr>
<tr>
<td>EDA</td>
<td>Electrodermal Activity</td>
</tr>
<tr>
<td>EEG</td>
<td>Electroencephalography</td>
</tr>
<tr>
<td>EMG</td>
<td>Electromyogram</td>
</tr>
<tr>
<td>EOG</td>
<td>Electrooculography</td>
</tr>
<tr>
<td>FMS</td>
<td>Freiburger Monitoring System</td>
</tr>
<tr>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>HR</td>
<td>Heart Rate</td>
</tr>
<tr>
<td>HRV</td>
<td>Heart Rate Variability</td>
</tr>
<tr>
<td>IC</td>
<td>Integrated Circuit</td>
</tr>
<tr>
<td>IDE</td>
<td>Integrated Development Environment</td>
</tr>
<tr>
<td>IIR</td>
<td>Infinite Impulse Response (Filter)</td>
</tr>
<tr>
<td>LED</td>
<td>Light Emitting Diode</td>
</tr>
<tr>
<td>MISO</td>
<td>Master In, Slave Out</td>
</tr>
<tr>
<td>MOSI</td>
<td>Master Out, Slave In</td>
</tr>
<tr>
<td>OSL</td>
<td>Oxygen Saturation Level</td>
</tr>
<tr>
<td>OSR</td>
<td>Oxygen Saturation Response</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>PDA</td>
<td>Personal Digital Assistant (handheld computer)</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>PGA</td>
<td>Programmable Gain Amplifier</td>
</tr>
<tr>
<td>PSoC</td>
<td>Programmable System on a Chip (by Cypress)</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse-Width-Modulator</td>
</tr>
<tr>
<td>QRS</td>
<td>Complex in the ECG signal consisting of the Q-, R- and S-waves</td>
</tr>
<tr>
<td>RA</td>
<td>Respiration Amplitude</td>
</tr>
<tr>
<td>RMS</td>
<td>Root Mean Square</td>
</tr>
<tr>
<td>RMSSD</td>
<td>Root Mean Square of Successive Differences</td>
</tr>
<tr>
<td>RR</td>
<td>Respiration Rate</td>
</tr>
<tr>
<td>SCL</td>
<td>Skin Conductance Level</td>
</tr>
<tr>
<td>SCR</td>
<td>Skin Conductance Response</td>
</tr>
<tr>
<td>SDRR</td>
<td>Standard deviation of the RR-intervals</td>
</tr>
<tr>
<td>SMD</td>
<td>Surface Mounted Device</td>
</tr>
<tr>
<td>SNR</td>
<td>Signal to Noise Ratio</td>
</tr>
<tr>
<td>SoC</td>
<td>System on a Chip</td>
</tr>
<tr>
<td>SOT</td>
<td>Small Outline Package</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial Peripheral Interface</td>
</tr>
<tr>
<td>SSOP</td>
<td>Shrinked Small Outline Package</td>
</tr>
<tr>
<td>UART</td>
<td>Universal Asynchronous Receiver Transmitter</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>Vcc</td>
<td>Supply Voltage</td>
</tr>
<tr>
<td>WIDAAN</td>
<td>Winkel-Daten-Analyse (software for the analysis of the CUELA data)</td>
</tr>
</tbody>
</table>
Bibliography


