



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorarbeit

**Oliver Quast**

**Berührungslose Schlafphasenerkennung zur Integration in ein  
Smart-Home**

*Fakultät Technik und Informatik  
Studiendepartment Informatik*

*Faculty of Engineering and Computer Science  
Department of Computer Science*

Oliver Quast

**Berührungslose Schlafphasenerkennung zur Integration in ein  
Smart-Home**

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Technische Informatik  
am Department Informatik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Gunter Klemke  
Zweitgutachter: Prof. Dr. Birgit Wendholt

Eingereicht am: 31. August 2012

**Oliver Quast**

**Thema der Arbeit**

Berührungslose Schlafphasenerkennung zur Integration in ein Smart-Home

**Stichworte**

Dehnungsmessstreifen, Living Place Hamburg, Schlafphasen, berührungslos, Smart-Home

**Kurzzusammenfassung**

Dieses Dokument beschäftigt sich mit der berührungslosen Schlafphasenerkennung im Kontext des Smart-Home, Living Place an der Hochschule für Angewandte Wissenschaften, zur Realisierung eines intelligenten Weckers.

**Oliver Quast**

**Title of the paper**

Non-Contact based sleep tracking for integration to a smart-home

**Keywords**

resistance strain gauge, Living Place Hamburg, sleep cycle, non contact, smart home

**Abstract**

This document deals with non contact sleep tracking in context with the smart home Living Place at the Hamburg University of Applied Sciences to implement an intelligent alarm clock

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Aufgabenbeschreibung . . . . .	1
1.3	Gliederung . . . . .	2
<b>2</b>	<b>Grundlagen</b>	<b>3</b>
2.1	Elektronische Grundlagen . . . . .	3
2.1.1	Widerstand eines Drahtes . . . . .	3
2.1.2	Widerstandsänderungen bei Temperaturänderungen . . . . .	4
2.2	Dehnungsmessstreifen . . . . .	5
2.2.1	Prinzipieller Aufbau des Dehnungsmessstreifens . . . . .	5
2.2.2	Grundlegende Funktion des Dehnungsmessstreifens . . . . .	6
2.2.3	Eigenschaften der Dehnungsmessstreifen . . . . .	6
2.3	Wheatstone'sche Brückenschaltung . . . . .	8
2.4	Schlafphasen und Schlafhygiene . . . . .	10
<b>3</b>	<b>Analyse</b>	<b>13</b>
3.1	Analyse der Schlafphasen . . . . .	13
3.2	Analyse existierender und alternativer Verfahren . . . . .	13
3.2.1	Existierende Verfahren . . . . .	14
3.2.2	Alternative Ideen . . . . .	19
3.2.3	Zusammenfassung der Verfahren . . . . .	21
3.3	Szenarien . . . . .	22
3.4	Anforderungen . . . . .	22
3.4.1	Generelle Anforderungen . . . . .	22
3.4.2	Softwareanforderungen . . . . .	24
3.4.3	Hardwareanforderungen . . . . .	26
3.4.4	Tabellarische Zusammenfassung . . . . .	26
<b>4</b>	<b>Design</b>	<b>27</b>
4.1	Hardwaredesign . . . . .	27
4.1.1	Mechanischer Aufbau . . . . .	28
4.1.2	Verstärkerschaltung . . . . .	32
4.1.3	Spannungsversorgung . . . . .	37
4.1.4	Offsetschaltung . . . . .	38
4.1.5	Schaltplan und Board . . . . .	41

<b>5</b>	<b>Realisierung und Schlafanalyse</b>	<b>44</b>
5.1	Realisierung des Aufbaus . . . . .	44
5.2	Softwarerealisierung . . . . .	46
5.2.1	Arduino Softwarerealisierung . . . . .	48
5.2.2	Server - Verbindung von Arduino zu ActiveMQ . . . . .	51
5.2.3	Aufbau des Clients . . . . .	55
5.2.4	Google API . . . . .	64
5.2.5	Day-View-Kalender . . . . .	66
5.3	Schlafanalyse . . . . .	67
5.4	Auswertung und Probleme . . . . .	70
5.4.1	Störung durch Funk . . . . .	70
5.4.2	Mechanischer Verschleiß . . . . .	71
5.4.3	Nicht abgeglichene Brückenschaltung . . . . .	71
5.4.4	Probleme beim Abgleich der Brückenschaltung . . . . .	71
5.4.5	Overhead an Daten . . . . .	72
<b>6</b>	<b>Fazit</b>	<b>73</b>
<b>7</b>	<b>Ausblick</b>	<b>75</b>
	<b>Glossar</b>	<b>80</b>

# Tabellenverzeichnis

2.1	Leitfähigkeit und Temperaturkoeffizient . . . . .	4
4.1	Verstärkungseinstellung des PGA204 . . . . .	37
5.1	Netzwerkbelegung Dip 9 und 10 . . . . .	49
5.2	MySQL-Ansicht einer verketteten Nachricht . . . . .	52

# Abbildungsverzeichnis

2.1	Stromröhre . . . . .	3
2.2	Aufbau eines Dehnungsmessstreifens . . . . .	5
2.3	Deformation des DMS . . . . .	6
2.4	Wheatstone'sche Brückenschaltung . . . . .	8
2.5	Abgeglichene Brückenschaltung . . . . .	9
2.6	Gehirnwellenform . . . . .	11
3.1	Druckpunktgrafik . . . . .	16
3.2	aXbo mit Armband . . . . .	17
3.3	Platzierung des iPhones zur Schlafanalyse . . . . .	19
3.4	Vergleichsanalyse auf Grundlage der Anforderungen . . . . .	22
3.5	Anforderungsanalyse . . . . .	26
4.1	Schema des Aufbaus . . . . .	28
4.2	Modell des Bettes mit DMS auf den Latten . . . . .	29
4.3	Modell einer Latte mit Dehnungsmessstreifen . . . . .	30
4.4	Verwendete Dehnungsmessstreifen . . . . .	31
4.5	Schema der Verstärkerschaltung . . . . .	32
4.6	Aufbau einer Differenzverstärkung mittels Transistoren . . . . .	33
4.7	Aufbau des PGA204 . . . . .	34
4.8	Eingangsruschen des PGA204 . . . . .	35
4.9	Grundlage der Verstärkerschaltung des PGA204 . . . . .	35
4.10	Verstärkerschaltung Teil 1 . . . . .	36
4.11	Verstärkerschaltung Teil 2 . . . . .	36
4.12	Spannungsversorgung . . . . .	38
4.13	Offsetschaltung aus dem Datenblatt . . . . .	39
4.14	Offsetschaltung . . . . .	40
4.15	Schaltplan für das Layout . . . . .	42
4.16	Platinen-Layout . . . . .	43
5.1	Bett mit Wake-Up-Scene . . . . .	44
5.2	Präparierte Latte aus dem Lattenrost . . . . .	45
5.3	Verwendeter Dehnungsmessstreifen . . . . .	45
5.4	Verstärkerschaltung . . . . .	46
5.5	Schema des Aufbaus . . . . .	47
5.6	Kommunikationsschema: Kommunikation Arduino -> Server -> Client . . . . .	51

5.7	Kommunikationsaufbau ActiveMQConnector . . . . .	53
5.8	Klassendiagramm ActiveMQConnector . . . . .	54
5.9	Klassendiagramm der Schlafphasenerkennungssoftware . . . . .	56
5.10	Erkennung eines Weckereignisses . . . . .	57
5.11	Ablauf der Weckzeitpunktermittlung . . . . .	58
5.12	Ansicht der Weckerform . . . . .	59
5.13	Hauptansicht der Clientsoftware zur Schlafphasenerkennung . . . . .	60
5.14	Kalenderansicht der Clientsoftware . . . . .	61
5.15	Komplette Graphenansicht der Clientsoftware zur Schlafphasenerkennung . . . . .	62
5.16	Komplettes Bewegungsprofil der Clientsoftware zur Schlafphasenerkennung . . . . .	63
5.17	Modell-View-Control . . . . .	64
5.18	Komplettes Schlafprofil einer Nacht . . . . .	68
5.19	Stärke der Bewegungen des Schlafenden einer Nacht . . . . .	68
5.20	Stärke der Bewegungen in Wachphase morgens als Ausschnitt . . . . .	69
5.21	Bewegungsprofilausschnitt, in Wachphase morgens . . . . .	69
5.22	Atmung mit Aussetzer . . . . .	70



# Listings

5.1	Benötigte Parameter für das Ethernet-Shield (vgl. <a href="#">Watterott</a> ) . . . . .	49
5.2	Ethernet-Shield starten ( <a href="#">Watterott</a> ) . . . . .	49
5.3	Clientrolle definieren ( <a href="#">Watterott</a> ) . . . . .	49
5.4	Client connect ( <a href="#">Watterott</a> ) . . . . .	50
5.5	Benötigte Parameter RedFly-Shield ( <a href="#">Watterott</a> ) . . . . .	50
5.6	init Redfly-Shield ( <a href="#">Watterott</a> ) . . . . .	50
5.7	Scan des Netzwerks mittels RedFly-Shield ( <a href="#">Watterott</a> ) . . . . .	50
5.8	Verbindungsaufbau zum Netzwerk mit RedFly-Shield ( <a href="#">Watterott</a> ) . . . . .	50
5.9	Verbindungsaufbau abschließen mit RedFly-Shield ( <a href="#">Watterott</a> ) . . . . .	51
5.10	Datentransfer RedFly-Shield (vgl. <a href="#">Watterott</a> ) . . . . .	51
5.11	Service anmelden für Google-Kalender . . . . .	65
5.12	Query-Erstellung für Google-Kalender . . . . .	65
5.13	Query-Übertragung zu Google . . . . .	65
5.14	Über EventFeed iterieren . . . . .	65
5.15	Initialisieren des DayView-Kalenders . . . . .	66
5.16	Über EventFeed iterieren . . . . .	66

# 1 Einleitung

## 1.1 Motivation

In der heutigen Zeit, in der alles schnell gehen und funktionieren muss, ist es wichtig, einen Ruhepunkt zu besitzen. Bei vielen Menschen ist dieser Punkt das Bett. Der Schlaf eines jeden Menschen ist überlebenswichtig. So verbringt jeder Mensch  $\frac{1}{3}$  seines Lebens im Bett mit Schlafen. In dieser Zeit findet die Aufarbeitung des Erlebten statt. Um diesen Schlaf analysieren zu können, gibt es die Möglichkeit, sich im Schlaflabor untersuchen zu lassen. Diese Untersuchung deckt jedoch nur einen Teil des Schlafens ab, da nicht jede Nacht in einem Schlaflabor unter Laborbedingungen geschlafen werden kann. Es wäre besser, eine deutlich längere Zeit zu überwachen. So entstand die Idee zu dieser Arbeit. Die Idee, einen Weg zu finden, der es ermöglicht, eine Person jede Nacht zu überwachen und diese Überwachung so zu gestalten, dass die zu überwachende Person hierfür nichts anderes tun muss, als normal ins Bett zu gehen. Die Schwierigkeit besteht darin, einen Mittelweg zwischen Genauigkeit der Messung zu finden sowie die Person nicht zu stören und im Schlaf zu beeinträchtigen.

Das primäre Ziel dieser Arbeit ist eine Möglichkeit zu schaffen, um auf Bewegungen im Bett zu reagieren, ohne den Schlaf einer Person zu beeinflussen. Dies ermöglicht die Ausarbeitung von Projekten wie dem Wecker 2.0.

## 1.2 Aufgabenbeschreibung

Diese Arbeit beschäftigt sich mit dem Erkennen von Bewegungen, die in einem Bett stattfinden. Hierzu soll ein modifiziertes Lattenrost zum Einsatz kommen, auf dem Sensoren aufgebracht sind. Mittels dieser Sensoren sollen die Bewegungen erkannt und an einen Computer weitergeleitet werden. Mithilfe der Erkennung, zu welchem Zeitpunkt Bewegungen aufgetreten sind und wie stark die Bewegungen waren, soll analysiert werden, ob die Person schläft oder wach ist.

## 1.3 Gliederung

In **Kapitel 2** wird auf die elektronischen Merkmale eingegangen. Es wird erklärt, was Dehnungsmessstreifen sind, und ihre Funktionsweise wird aufgezeigt.

Anschließend wird in **Abschnitt 2.4** ein Einblick in die Schlaftheorie und die physiologischen Eigenschaften gegeben, um ein Verständnis der Arbeitsweise des Bettes zu vermitteln. Anschließend wird in **Kapitel 3** auf verschiedene bereits vorhandene und alternative Techniken eingegangen, die zur Inspiration dienen, und erklärt, warum die vorgestellten Verfahren (**Abschnitt 3.2**) nicht geeignet sind. Zudem wird ein Szenario aufgezeigt, das in Verbindung mit den zuvor erwähnten Verfahren die Anforderungen (**Abschnitt 3.4**) bildet. In **Abschnitt 3.4** wird der genaue Rahmen dieser Arbeit abgesteckt und geklärt, was diese Arbeit in ihrem fertigen Zustand leisten soll.

**Kapitel 4** diskutiert den geplanten Hardwareaufbau, gibt einen Einblick in die verwendeten Komponenten und bildet die Grundlage für **Kapitel 5**. In diesem Kapitel wird der Aufbau gezeigt und diskutiert. Anschließend wird besprochen, welche Teile der Anforderungen eingehalten wurden und welche nicht realisierbar sind und erklärt, aus welchem Grund dieser Ansatz verworfen wurde.

In **Abschnitt 5.4** werden die Probleme weiter besprochen und die Lösungsansätze beschrieben. Eine abschließende Zusammenfassung der erbrachten Leistungen findet in **Kapitel 6** statt. Zudem erfolgt ein Vergleich der erbrachten Leistungen mit den Anforderungen, um eine abschließende Bewertung vorzunehmen. Alle Ideen zur Verbesserung, Erweiterung und dem Ausbau dieser Arbeit werden anschließend in **Kapitel 7** aufgezeigt und vertieft.

## 2 Grundlagen

In diesem Kapitel werden sowohl das Prinzip der Dehnungsmessstreifen als auch der Aufbau, die Eigenschaften und Funktionen behandelt. Ergänzend wird die Wheatstone'sche Brückenschaltung behandelt.

### 2.1 Elektronische Grundlagen

Zuerst werden die elektronischen Grundlagen besprochen, die die Funktionsweise der Dehnungsmessstreifen beschreiben und zum Verständnis des Prinzips dienen.

#### 2.1.1 Widerstand eines Drahtes

Als Teilausschnitt eines Drahtes wird eine Stromröhre betrachtet. Die Stromröhre hat eine Länge  $l$  und einen Querschnitt  $A$  (siehe [Abbildung 2.1](#)).

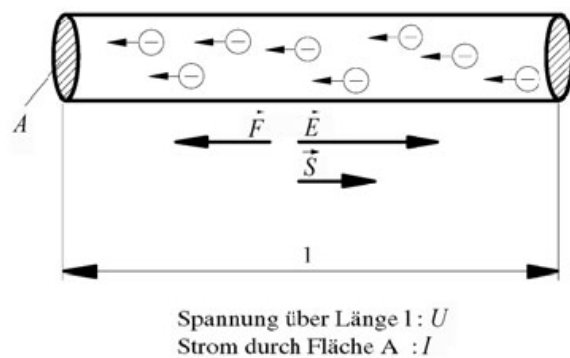


Abbildung 2.1: Stromröhre  
[Busch und Busch \(2011\)](#)

Die Stromröhre setzt dem Elektronenfluss einen Widerstand entgegen. Dieser Widerstand ist abhängig vom Querschnitt, der Länge der Stromröhre sowie dem spezifischen Widerstand  $\rho$ . Somit ergibt sich diese Formel:

$$R = \rho * \frac{l}{A}$$

Busch und Busch (2011)

Hieraus ist ersichtlich, dass sich der Widerstand bei einer Querschnittverringern erhöht.  $R$  ist der elektrische Widerstand in  $\Omega$ , der elektrische Leitwert ist  $\frac{1}{\Omega}$  und heißt Siemens.

### 2.1.2 Widerstandsänderungen bei Temperaturänderungen

Der zuvor angesprochene Widerstand ist durch die Temperatur beeinflussbar, da sich die elektrische Leitfähigkeit durch Erhöhung der Temperatur verringert. Unter dieser Tatsache ist die folgende Formel wichtig.

$$R(t) = R(t_0) * (1 + \alpha_{t_0} * (t - t_0))$$

Wobei  $t_0 = 20^\circ C$

Material	$\rho$ in $\frac{\Omega * mm^2}{m}$	$\alpha_{20}$ in $\frac{1}{K}$
Silber	$16,5 * 10^{-3}$	$3,8 * 10^{-3}$
Kupfer	$17,8 * 10^{-3}$	$3,9 * 10^{-3}$
Silizium	$2,3 * 10^9$	$-75 * 10^{-3}$

Tabelle 2.1: Leitfähigkeit und Temperaturkoeffizient  
(vgl. Klaus Tkotz (2005))

## 2.2 Dehnungsmessstreifen

In diesem Kapitel wird auf den Aufbau und die Funktionsweise von Dehnungsmessstreifen (DMS) eingegangen, sodass die Funktionsweise beim Einsatz im Bett erkennbar wird.

### 2.2.1 Prinzipieller Aufbau des Dehnungsmessstreifens

Der DMS besteht aus Kunststoff, auf dem in Schlangenlinien (siehe [Abbildung 2.2](#)) in Verformungsrichtung ein dünner leitfähiger Streifen aufgetragen wurde. An einem Ende befinden sich zudem Lötunkte an denen Kabel zur Messung angeschlossen werden können.

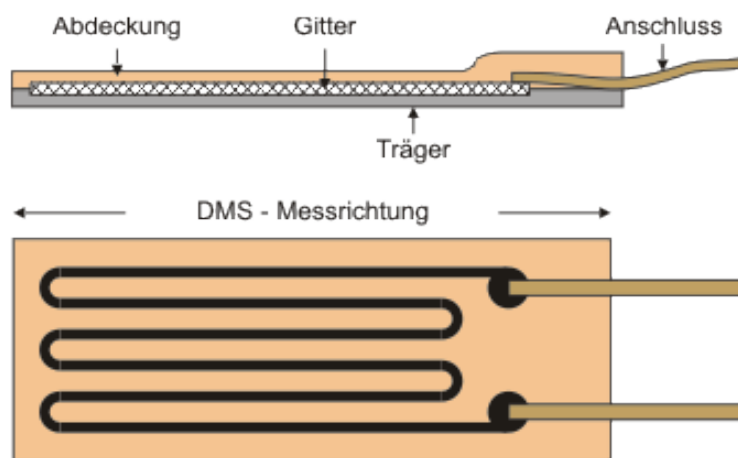


Abbildung 2.2: Aufbau eines Dehnungsmessstreifens  
disynet Data Interface Systems und GmbH

### 2.2.2 Grundlegende Funktion des Dehnungsmessstreifens

Der auf einem Trägermaterial fest verbaute DMS wird durch die Verformung des Trägermaterials gestreckt bzw. gestaucht (siehe [Unterabschnitt 2.2.2](#)). Durch diese Änderung des DMS

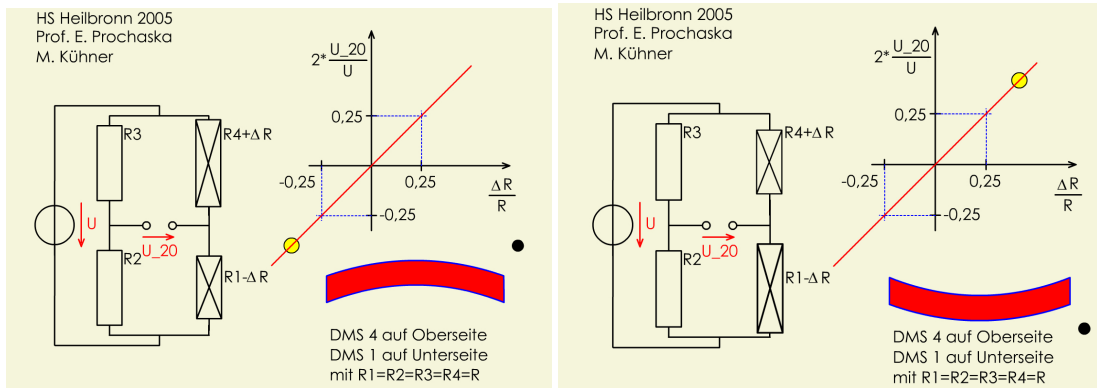


Abbildung 2.3: Deformation des DMS  
Prochaska (2012)

wird die auf ihm aufgebraute Leiterbahn eben so verformt. Die Verformung bewirkt eine Querschnittänderung der Leiterbahn, wodurch eine Widerstandsänderung erzielt wird. Diese Widerstandsänderung wird im vorliegenden Aufbau (DMS auf der Ober- und Unterseite) noch durch die entgegengesetzte Deformation verdoppelt.

### 2.2.3 Eigenschaften der Dehnungsmessstreifen

Für DMS gibt es verschiedene Kenngrößen, die für die Verwendung wichtig sind.

**Der Widerstand** eines Dehnungsmessstreifens wird in Ohm ( $\Omega$ ) angegeben. Er bezieht sich auf den elektrischen Widerstand eines unbelasteten, nicht aufgeklebten Dehnungsmessstreifen bei Raumtemperatur (vgl. [Messtechnik](#)).

**Der k-Faktor** drückt das zahlenmäßige Verhältnis der Wandlung der mechanischen Größe  $\Delta l$  in die elektrische Größe  $\Delta R$  aus und wird unter Nennung der Poissonzahl des verwendeten Testobjekts angegeben.

$$k = \frac{\Delta R}{\varepsilon} \quad \begin{array}{l} k: \text{ k-Faktor} \\ \Delta R: \text{ Widerstandsänderung} \\ R: \text{ DMS-Widerstand} \\ \varepsilon: \text{ Dehnung} \end{array}$$

(vgl. **Messtechnik**)

**Korrektur des k-Faktors** Um die Verringerung des k-Faktors und damit die Empfindlichkeit des Dehnungsmessstreifens zu vermeiden, können verschiedene Verfahren angewendet werden. Ein Verfahren ist die rechnerische Korrektur.

$$A = \frac{R}{R+rL} \quad \begin{array}{l} R: \text{ Normwiderstand des DMS} \\ r: \text{ Gesamtwiderstand der Zuleitung } (\Omega/m) \\ L: \text{ Länge der Zuleitung (m)} \\ k: \text{ angegebener k-Faktor} \end{array}$$
$$k_0 = \frac{R}{R+rL} * k = Ak$$

**Gitterlänge** Die angegebene Gitterlänge beschreibt die aktive Messlänge. Die aktive Messlänge ist die Länge zwischen den Umkehrschlaufen (vgl. **Messtechnik**).

**Querempfindlichkeit** Aufgrund ihrer Beschaffenheit reagieren DMS auch auf Dehnung orthogonal zur eigentlichen Messrichtung.

$$k_t = \frac{\Delta R}{\varepsilon_t} \quad \begin{array}{l} k_t: \text{ k-Faktor in Querrichtung} \\ \Delta R: \text{ Widerstandsänderung} \\ R: \text{ DMS-Widerstand} \\ \varepsilon_t: \text{ Dehnung in Querrichtung} \end{array}$$

$$C = \frac{\text{Empfindlichkeit in Querrichtung}}{\text{Empfindlichkeit in Längsrichtung}} = \frac{k_t}{k}$$

(vgl. **Messtechnik**)

**Dehnungsgrenze** Die Dehnungsgrenze oder zulässige Dehnung eines DMS hängt von den Eigenschaften des Gittermaterials, des Trägers und des zur DMS-Installation verwendeten Klebstoffes ab. Im Allgemeinen ist die Dehnungsgrenze eines DMS mit kurzem Messgitter



im Verhältnis kleiner als die eines DMS aus der gleichen Serie mit längerem Messgitter. (Messtechnik)

### 2.3 Wheatstone'sche Brückenschaltung

Nachdem die elektronischen Grundlagen dargelegt wurden, wird auf die Wheatstone'sche Brückenschaltung eingegangen. Die Wheatstone'sche Brückenschaltung wird zur Aufnahme kleinerer Widerstandsänderungen in der Messtechnik verwendet. Die Brückenschaltung ist eine Gruppenschaltung aus Widerständen, die zwei parallele Reihenschaltungen aus je zwei Widerständen beinhaltet (siehe [Abbildung 2.4](#)). Im abgeglichenen Zustand herrscht zwischen den Widerständen kein Potentialunterschied, somit eine Spannung von 0V.

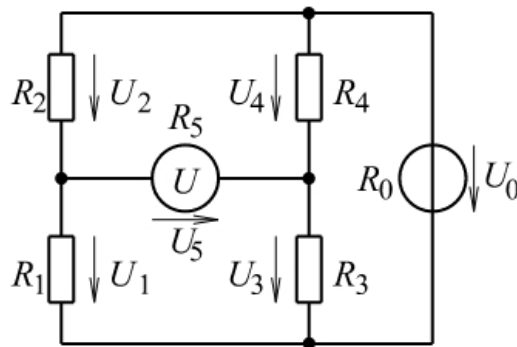


Abbildung 2.4: Wheatstone'sche Brückenschaltung  
Dvorak (2012)

In einer Brückenschaltung redet man von einem abgeglichenen Zustand genau dann, wenn die Mitte der linken Seite der Messbrücke dasselbe Potential, ungleich Null, besitzt wie das der rechten Seite (siehe [Abbildung 2.5](#)).

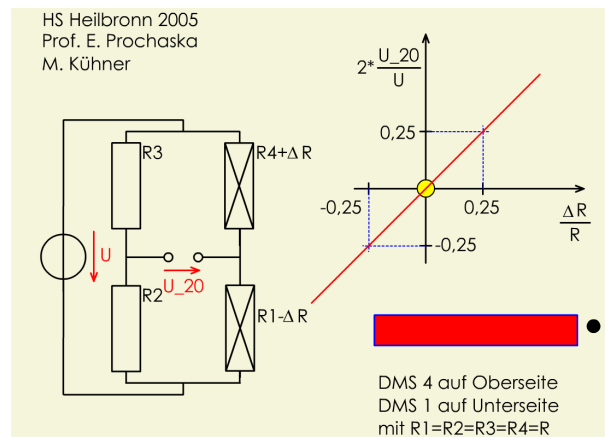


Abbildung 2.5: Abgegliche Brückenschaltung  
Prochaska (2012)

Hier werden zur Berechnung die Widerstandsverhältnisse zueinander betrachtet.

$$\frac{R_1}{R_2} = \frac{R_3}{R_4} = \frac{U_1}{U_2} = \frac{U_3}{U_4}$$

### Brückenschaltungsarten

Es existieren drei verschiedene Brückenschaltungsarten, die alle Vor- und Nachteile haben.

**Die Viertelbrücke** besitzt drei feste, unveränderbare Widerstände (R2, R3, R4) und nur ein Widerstand (R1) dient zur Potentialänderung zwischen den Widerständen. Hierbei treten jedoch gerade durch Temperaturänderungen Probleme auf. Bei einer unterschiedlichen Beschaffenheit der Widerstände ändert sich der Wert im Verhältnis zu den anderen Widerständen unterschiedlich stark. Eine abgegliche Brückenschaltung wird hierdurch gestört. Im hier behandelten Versuch verstärkt sich der Effekt durch das Trägermaterial. Würde lediglich eine Viertelbrücke zum Einsatz kommen, könnte die Temperaturänderung des DMS und die Ausdehnung des Trägermaterials, auf dem der DMS aufgebracht ist, Störungen verursachen. Der Aufbau gewährleistet eine  $\frac{1}{4}$ -fache Änderung der Spannung im Verhältnis zur Widerstandsänderung.

**Die Halbbrücke** besitzt zwei feste (R3 und R4), unveränderbare Widerstände und zwei Widerstände (R1 und R2), die sich verändern lassen, die Widerstandspaare sind jeweils in einer Reihenschaltung angeordnet sind. So können die Verhältnisse auch bei Temperaturschwankungen beibehalten werden und die Brückenschaltung bleibt ausgeglichen. Mittels

dieser Anordnung kann auch bei der Oberflächenveränderung durch Temperaturunterschiede des Trägermaterials das Verhältnis der Widerstände eingehalten werden. Bei Änderungen der Brückenschaltung schlagen die Änderungen nur mit dem  $\frac{1}{2}$ -fachen der Widerstandsänderung zur Spannung durch.

**Die Vollbrücke** besitzt vier veränderbare Widerstände (R1-R4). So bleiben alle Vorteile der Halbbrücke zur Viertelbrücke bestehen. Zusätzlich wird aber der Nachteil, dass nur die Hälfte der Änderung detektiert wird, ausgeglichen.

Jedoch entsteht hier der doppelte Materialaufwand im Vergleich zur Halbbrücke und der 4-fache im Vergleich zur Viertelbrücke.

## 2.4 Schlafphasen und Schlafhygiene

Während des Schlafens durchläuft der Mensch mehrere Schlafphasen. Um die Messbarkeit dieser Schlafphasen zu analysieren, werden im folgenden die Schlafphasen aufgezeigt.

Es existieren zwei grundlegende Schlafphasen. Diese werden von **Fischer (2012)** folgendermaßen unterteilt.

### REM-Schlafphase

REM steht für Rapid-Eye-Movement (schnelle Augenbewegungen). In dieser Schlafphase findet eine sehr hohe Hirnaktivität statt. Die Muskulatur wird in dieser Phase nur minimal beansprucht, einzig die Gesichtsmuskulatur wird noch vom Gehirn aktiv kontrolliert.

### Non-REM-Schlafphase

Die Non-REM-Phase lässt sich in vier Phasen unterteilen:

- **Phase 1: Übergang von Wach- zu Einschlafphase**

In dieser Phase verändern sich die Hirnwellen langsam von Betawellen in Alphawellen (siehe **Abbildung 2.6**), die langsamer und gleichmäßiger sind.

Die Augen sind geschlossen und die Muskulatur entspannt sich.

Die Atmung und Herzfrequenz werden langsamer. Die Sinneswahrnehmung verändert sich und die Reaktion auf akustische Reize nimmt ab. In dieser Phase ist man noch leicht zu wecken.

- **Phase 2: Einschlafphase**

Diese Phase dauert nur wenige Sekunden bis Minuten. In dieser Schlafphase kommt es

oft zu krampfartigem Muskelzucken. Die Augenbewegung ist langsam, meist horizontal ausgerichtet und die Bewegungen sind ziellos. Diese Phase wird auch als SEM (Slow-Eye-Movement) bezeichnet.

- **Phase 3: leichter Schlaf**

In dieser Phase hat sich der Schlaf schon vertieft und es wird in der Regel nicht oder sehr realitätsbezogen geträumt. Die Muskulatur ist kaum noch angespannt und auch die Augenbewegung hat nahezu aufgehört.

- **Phase 4: Tiefschlafphase**

Diese ist die tiefste Schlafphase. In dieser Schlafphase findet die Erholung des Körpers und des Geistes statt. Daher ist diese Schlafphase die wichtigste.

Die Atmung und die Herzfrequenz sind sehr langsam und gleichmäßig. Aus dieser Schlafphase ist es nur sehr schwer aufzuwachen. Oft werden mehrere Minuten benötigt, um sich zurecht zu finden, wenn man in dieser Phase aufgeweckt wird. Im Schnitt dauert diese Phase beim ersten Eintreten 90 Minuten. Beim zweiten Eintreten ist die Dauer im Schnitt um 25 Minuten verkürzt.

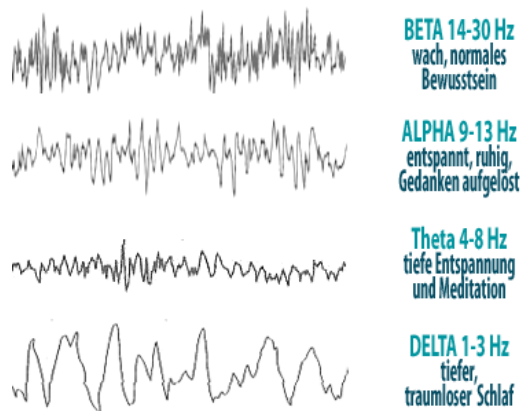


Abbildung 2.6: Gehirnwellenform

Kuonen

Damit die Schlafphasen in der beschriebenen Form durchlaufen werden können, muss die Schlafhygiene eingehalten werden. Was unter Schlafhygiene verstanden werden kann, wird im Folgenden besprochen.

Erwachsene Menschen schlafen in der Regel 7,5 Stunden und durchlaufen alle Schlafphasen

viermal (vgl. **Baumgart**). Die folgenden fünf Punkte sollten im Laufe des Tages berücksichtigt werden, damit eine Unterteilung der Schlafphasen stattfinden kann.

- **Stimulatoren**

In den vier Stunden ist vor Antritt des Schlafes auf koffeinhaltige Lebensmittel, Getränke und Medikamente zu verzichten, da diese die Hirnaktivität anregen und Entzugserscheinungen in der Nacht stören können. Ebenso sollte Nikotinkonsum am Abend nicht stattfinden, da dies auf Grund von Entzug ebenso zu Schlafproblemen in der Nacht führen kann.

- **Alkohol**

Auf Alkohol ist zu verzichten. Alkohol setzt die Hirnaktivität herab. Dies führt zum leichteren Einschlafen, bewirkt jedoch im weiteren Verlauf Störungen im Übergang zwischen den Schlafphasen und kann zum Aufwachen führen.

- **Sport**

Sport bewirkt einen gesünderen Schlaf, jedoch sollte auch dieser nicht zu spät ausgeübt werden. Sport führt zur Aktivierung des Körpers, so kann es am Abend zu Einschlafstörungen kommen.

- **Die Schlafumgebung**

Ein bequemes Bett ist in einem dunklen, gut gelüfteten Raum zu stellen und frei von Reizen wie Straßenlärm, elektronischen Geräten oder dem laufenden Fernseher sein. Die Raumtemperatur muss niedriger sein als in anderen Bereichen der Wohnung, aber nicht zu kalt. Erhöhte Temperatur führt zum Schwitzen. Auf die Feuchtigkeit reagiert der Körper mit Wälzen im Bett.

- **Ernährung**

Fettes Essen und übermäßige Mengen können zu Ein- und Durchschlafstörungen führen. Eine leichte Mahlzeit hingegen kann den Schlaf fördern. Milchprodukte eignen sich besonders als kleine Mahlzeit vor dem Einschlafen, da diese die Substanz Tryptophan beinhalten, die schlaffördernd wirkt.

(vgl. **Penzel (b)**)

### **Überleitung**

In diesem Kapitel wurden die Grundlagen zum Aufbau dieser Arbeit gelegt.

Als nächstes werden sowohl die Schlaftheorie als auch vorhandene und existierende Verfahren analysiert und unter Zuhilfenahme eines Szenarios die Anforderungen formuliert.

## 3 Analyse

Im Rahmen dieser Arbeit soll eine Möglichkeit entwickelt werden, Bewegungen im Bett zu detektieren und diese zur Auswertung zur Verfügung zu stellen. Die Bewegungen sollen möglichst präzise erkannt werden und nicht von anderen Entwicklungen im Living Place beeinflusst werden. Zur Detektion sollen Sensoren im Bett verbaut werden, die es ermöglichen, kleinste Bewegungen zu registrieren und die Bewegungen an einen Server zu melden.

Als Prototyp soll zudem ein Wecker implementiert werden, der die detektierten Bewegungen nutzt, um eine zu weckende Person in einer vordefinierten Weckzeit zu wecken. Dies soll das Aufstehen erleichtern.

Ebenso ist ein Client zu entwickeln, der die Messwerte grafisch aufarbeitet und die Möglichkeit zur Analyse der Messwerte bietet.

### 3.1 Analyse der Schlafphasen

Aus den unter [Abschnitt 2.4](#) gezeigten Schlafphasen und den physiologischen Eigenschaften ergibt sich, dass nur Phase 1 und Phase 2 durch die Bewegung des gesamten Körpers erkannt werden können, da in den letzten Phasen kaum noch bis gar keine Bewegungen stattfinden. So ergibt sich die Definition Wachphase und Schlafphase als Gruppenbildung. Die Wachphase fasst die Gruppe aus Phase 1 und Phase 2 zusammen. Wachphase bedeutet hierbei, dass Bewegungen detektierbar sind. Die Schlafphase beschreibt hingegen die Gruppe aus Phase 3, Phase 4 und der REM-Phase und fasst damit die Phasen, in denen keine Bewegungen detektiert werden können, zusammen.

Die Annahme, dass Bewegungen mit den Schlafphasen übereinstimmen und bei allen Personen dieselben Ergebnisse liefern, muss durch weitere Untersuchungen sichergestellt werden.

### 3.2 Analyse existierender und alternativer Verfahren

Um die zuvor aufgezeigten Schlafphasen analysieren zu können, existieren mehrere Ansätze. In diesem Abschnitt werden bereits existierende Ansätze und deren Vor- bzw. Nachteile

aufgezeigt. Die erwähnten Verfahren und ihre Vor- und Nachteile dienen anschließend zur Formulierung der Anforderungen an diese Arbeit. Zudem wird diskutiert, ob die vorgestellten Verfahren im Einsatz praktikabel realisierbar sind und vom Benutzer angenommen werden. Nachdem bereits existierende Ansätze gezeigt wurden, wird auf eigene Ideen eingegangen. Diese Ideen werden mit den bereits existierenden Ansätzen verglichen.

#### 3.2.1 Existierende Verfahren

Weltweit existieren verschiedene Ansätze zur Erkennung von Schlafphasen. Die Ansätze sind nach ihrer Professionalität sortiert. Als Erstes wird das professionelle Verfahren zur Schlafanalyse gezeigt. Dieses soll ein Maß zum Vergleichen bieten und den Aufwand zeigen, der erbracht werden muss, um eine genaue Analyse der Schlafphasen zu erhalten.

**Schlaflabor:** Die professionelle Schlafanalyse findet in einem Schlaflabor statt. Hier werden sowohl die Hirnströme und deren Ausdehnung als auch Form und Aktivität des Gehirns analysiert. Zusätzlich werden die Atmung und die Herzfrequenz gemessen, um den Vitalstatus der zu überwachenden Person festzustellen. Muskelkontraktionen an Armen, Beinen, dem Torso und im Gesicht werden zudem aufgezeichnet, um Bewegungen zu registrieren. Hierfür wird der ganze Körper mit Elektroden am Kabel bestückt (vgl. [Penzel \(a\)](#)). Dies ist im alltäglichen Leben nicht realisierbar. Dieses Verfahren bietet die einzige genaue Überwachung des Körpers und der Körperfunktionen und sollte nicht mit dem in dieser Arbeit zu entwickelnden Verfahren gleichgesetzt werden. Es wird lediglich erwähnt, um aufzuzeigen, welches Maß an Aufwand nötig ist, um eine genaue Messung zu gewährleisten.

##### **Vorteile:**

Die Messung bietet eine Bestimmung der Schlafphasen.

##### **Nachteile:**

Auf Grund des hohen Aufwands durch die Verkabelung ist dieses Verfahren nicht praktikabel. Zudem findet keine automatische Auswertung statt.

**Luftmatratze mit Luftdruckmessventil:** Es existieren im Internet ([JEPICO](#)) Ansätze, den Körper mittels einer Luftmatratze und eines Luftdruckmessers zu überwachen. Hierbei wird eine Luftmatratze auf die eigentliche Matratze gelegt. Die Luftmatratze verfügt über einen

Luftdruckmesser am Ventil der Matratze. Dieser soll die Bewegung mittels der dadurch resultierenden Luftdruckschwankungen erkennen und messbar machen. Dieses Verfahren behindert die Funktion des Lattenrostes, da die überwiegende Belastung bereits in der Luftmatratze abgefedert wird. Dieses Verfahren bietet keine dauerhafte Lösung zur generellen Erkennung des Schlafverhaltens, da der Aufbau regelmäßig kontrolliert werden muss.

**Vorteile:**

Dieser Aufbau ist kostengünstig realisierbar. Die Auswertung der Druckschwankungen erfolgt direkt, ist leicht digitalisierbar und somit leicht zu interpretieren.

**Nachteile:**

Der Aufbau muss regelmäßig kontrolliert werden.

Schwankender Luftdruck in der Umgebung verfälscht das Ergebnis.

Die Funktion des Lattenrostes wird durch den Aufbau beeinträchtigt.

**Druckpunktmatten:** Druckpunktmatten haben eine hohe Anzahl an Sensoren, die jeder für sich den Druck, der auf ihnen lastet, messen und die Messwerte in einem 3D-Diagramm darstellt. Die Druckpunktmatte kann kaum merkbar verwendet werden, da sie so dick ist wie eine Decke und einfach auf der Matratze aufgebracht werden kann. Die Druckpunktmatte liefert ein exaktes Bild der Silhouette der Person, die von der Druckpunktmatte erfasst wird (siehe [Abbildung 3.1](#)).



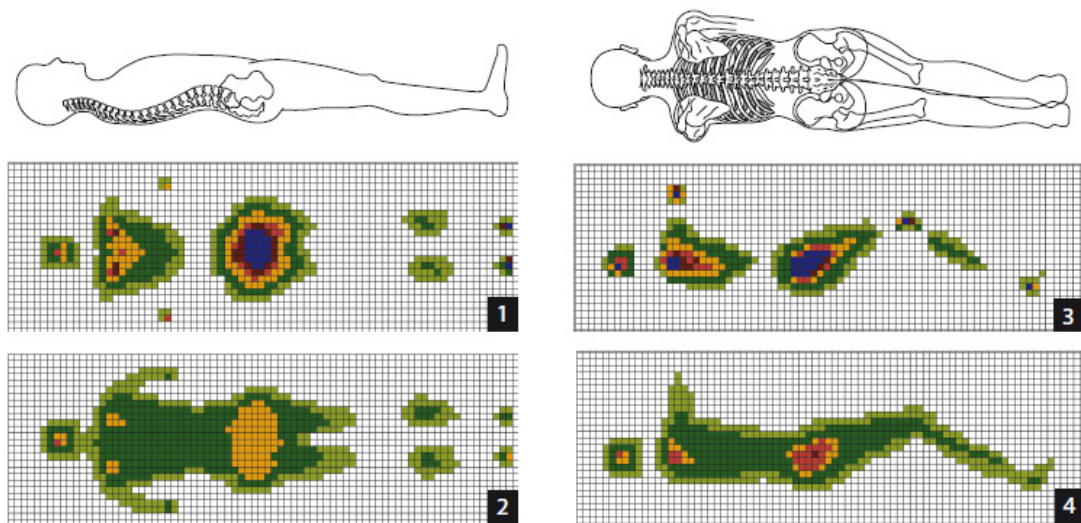


Abbildung 3.1: Druckpunktgrafik  
Walter

Druckpunktmatten werden hauptsächlich zur Bestimmung der richtigen Matratze oder zur Anpassung des Sattels im Reitsport verwendet. Das hier erzeugte Bild liefert für das Ziel dieser Arbeit eine zu genaue Messung. Im Rahmen dieser Arbeit sind die Bewegung an sich, die Bewegungsstärke und der Abstand der Bewegungen interessant, nicht jedoch die Position der Körperteile oder die Druckpunkte (vgl. [MattressLab](#) und [Fachverband](#)).

#### **Vorteile:**

Kleine Bewegungen sind detektierbar.

Die Druckpunktmatte bietet die Möglichkeit der Zuordnung der Bewegung zu Körperteilen und deren Positionen.

#### **Nachteile:**

Es ist ein kostenintensiver Aufbau nötig.

Die Auswertung der Daten ist umständlich und erfordert viel Rechenleistung.

**Bewegungssensoren am Armgelenk:** Eine Umsetzung der Idee zum intelligenten Wecker ist der aXbo, ein Wecker, der die Bewegungen der zu überwachenden Person mittels eines Armbandes ([Abbildung 3.2](#)) misst, um diese im Fast-Wach-Moment zu wecken (vgl. [werner](#)

**eis company austria**). Im Armband befinden sich Lage- und Beschleunigungssensoren, diese registrieren jede Bewegung des Armes und übermitteln sie an einen Wecker neben dem Bett. Dieser Wecker übernimmt die Auswertung und analysiert die Bewegungen. Dieses Konzept basiert auf einem Armband und nutzt die Übermittlung der Messwerte mittels Funk. Ein



Abbildung 3.2: aXbo mit Armband  
**werner eis company austria**

ähnliches Konzept verfolgt der Wecker von SleepSmart, dieser verwendet ein Stirnband, um die Hirnströme zu messen (**sleepsmart** und **afp**). Diese Verfahren können nicht in das bestehende Netz des LP integriert werden, da sie keine Schnittstelle beinhalten, um eine Steuerung außerhalb des Produktes vorzunehmen.

#### **Vorteile:**

Der Aufbau und die Anwendung sind einfach gehalten und leicht zu verwenden. Das Verfahren bietet die Möglichkeit, mehrere Personen zu überwachen und die Bewegungen den jeweiligen Personen zuzuordnen.

#### **Nachteile:**

Aufgrund fehlender Schnittstellen bietet dieses Verfahren keine Möglichkeit, dieses in das Living Place zu integrieren.

Es benötigt das Anlegen von Messinstrumenten und ist ohne das Anlegen der Messinstrumente nicht einsatzbereit.

**Kapazitive Druckmessstreifen:** In der Masterarbeit „Das intelligente Bett“ ([Hardenack](#)) wurde das Bett mit kapazitiven Druckmessstreifen ausgestattet. Diese Sensoren ermitteln den Druck auf den Latten, nicht die Verformung. Die Drucksensoren sind mittels eines mehrschichtigen Modells aus Schirmung, Isolatoren und einer kapazitiven Schicht aufgebaut. Dieses Verfahren mit der implementierten Software benötigt einen PC neben dem Bett und verbraucht die vorhandenen Ressourcen nahezu vollständig.

**Vorteile:**

Das Verfahren bietet die Möglichkeit der berührungslosen Erkennung von Bewegungen.

**Nachteile:**

Es wird zur Auswertung der Messwerte ein PC in der Nähe des Bettes benötigt.

Durch den Aufbau unter Zuhilfenahme eines PCs entstehen Geräusche, die den Schlaf stören können.

Durch das kapazitive Messen entstehen Strahlungen, die den Körper beeinflussen können.

Das Auswerten der Messwerte ist rechenintensiv und erfordert viele Ressourcen.

**App auf dem Smartphone:** Als eines von vielen Beispielen für Smartphone-Apps wurde das Programm Sleep Cycle getestet. Hierbei wird wie in [Abbildung 3.3](#) zu sehen das iPhone neben dem Kopfkissen auf der Matratze platziert. So sollen die Bewegungen der Person erkannt und im internen Speicher des iPhones gespeichert werden. Das Hamburger Schlaflabor des Jerusalem Krankenhauses hat mit Spiegel Online diese App in einem Versuch getestet. Die Ärztin Andrea Iwansky räumte maximal die Chance ein, zwischen Traumschlaf und Nichttraumschlaf zu unterscheiden, jedoch bezweifelt sie, anhand der Bewegungen die Tiefe des Schlafes und die einzelnen Schlafphasen unterscheiden zu können. Diese Vermutung wurde durch den Versuch bestätigt. Ob dies auf Grund der App oder des Funktionsprinzips nicht funktionierte, kann hierbei nicht nachvollzogen werden (vgl. [Klettke](#)).

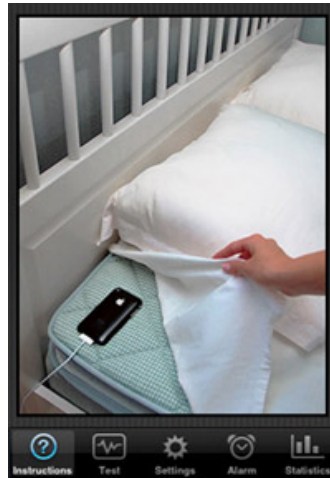


Abbildung 3.3: Platzierung des iPhones zur Schlafanalyse  
**SleepCycle**

#### **Vorteile:**

Einfacher Aufbau der Messeinrichtung.

Kostengünstiger Aufbau, wenn ein Smartphone vorhanden ist.

#### **Nachteile:**

Die Anwendung strapaziert den Akku im Smartphone.

Es existiert keine zufriedenstellende Funktion festzustellen.

Die Strahlung des Smartphones beeinträchtigt die Schlafhygiene.

### **3.2.2 Alternative Ideen**

Da die hier vorgestellten herkömmlichen Verfahren keine zufriedenstellende Lösung erbracht haben, werden im Folgenden alternative Lösungsansätze diskutiert und bewertet.

**Ultraschallmessverfahren:** Hier handelt es sich um die Idee, den Platz unter dem Bett zu nutzen und so das Bett komplett unverändert zu lassen. Hierfür wird unterhalb des Bettes ein Ultraschallsensor verbaut, der den Abstand zum Bett misst. Wird nun die Matratze durch einen auf ihr liegenden Körper belastet, sinkt die Matratze auf dem Lattenrost ein, sodass sich der Abstand zu dem unter dem Bett befestigten Ultraschallsensor verringert. Die Messwerte

des Sensors können von einem PC ausgelesen werden. Dies hat den Vorteil, dass keine Verschleißteile anfallen, da die Messeinrichtung nicht durch die Bewegungen im Bett direkt beeinflusst wird.

**Vorteile:**

Der Aufbau der Hardware ist unkompliziert zu realisieren und zudem kostengünstig.

**Nachteile:**

Mehrere Sensoren stören sich gegenseitig.

Die Beeinflussung von Haustieren und der zu überwachenden Person ist nicht auszuschließen. Der Aufbau benötigt viel Raum unter dem Bett, um eine genaue Messung zu ermöglichen.

**Infrarotmessverfahren:** Bei dem Infrarotmessverfahren handelt es sich um eine Umstellung des zuvor erwähnten Verfahrens. Hierbei wird der Ultraschallsensor durch einen Infrarotsensor ersetzt. Dies eliminiert die negativen Einflüsse auf den Körper und die Haustiere durch die Verwendung von Ultraschall.

Hier entstehen Probleme durch Staub, der sich auf dem Infrarotsensor absetzt und auf Dauer die Messungen beeinflusst.

**Vorteile:**

Der Aufbau der Hardware ist unkompliziert zu realisieren und zudem kostengünstig.

**Nachteile:**

Es sind Wartungsarbeiten nötig, um ein Verdrecken der Sensoren und die damit verbundene Beeinträchtigung der Messung zu verhindern.

Der Aufbau benötigt viel Raum unter dem Bett, um eine genaue Messung zu ermöglichen.

**Bewegungsmelder:** Die einfachste Art, Bewegungen zu detektieren, ist, einen Bewegungsmelder einzusetzen. Jedoch würde das im Versuchsaufbau heißen, dass keine Decke verwendet werden darf. Die Decke würde die Infrarotabstrahlung, die durch den Bewegungsmelder detektiert wird, zu stark abgeschwächt. Durch diesen Versuchsaufbau wären nur starke Bewegungen detektierbar.

**Vorteile:**

Der Aufbau der Hardware kann kostengünstig realisiert werden.

**Nachteile:**

Durch die Bettdecke können keine oder nur starke Bewegungen detektiert werden.

**Dehnungsmessstreifen:** Bei diesem Messverfahren handelt es sich um die Detektion der Deformationen der Latten des Lattenrostes. Hier werden Dehnungsmessstreifen direkt auf der Latte verklebt, so ist es möglich, die Bewegungen indirekt zu messen. Hier ergeben sich Vorteile im Bereich der Störanfälligkeit, da die zu verwendenden Dehnungsmessstreifen nicht durch Licht, Staub oder Geräusche beeinflusst werden können. Ebenso sind die Dehnungsmessstreifen durch die Matratze geschützt und können hierdurch nicht beschädigt werden.

**Vorteile:**

Der Aufbau der Hardware ist kostengünstig zu realisieren.

Mittels des Aufbaus ist die Detektion kleiner Bewegungen möglich.

Die Auswertung kann dezentral mit wenig Rechenaufwand realisiert werden.

Durch den Aufbau wird die Schlafhygiene der zu überwachenden Person nicht gestört.

Ein Wartungsfreier Aufbau kann realisiert werden.

Umwelteinflüsse wie Temperatur oder Luftdruck beeinflussen das Messsystem nicht.

#### 3.2.3 Zusammenfassung der Verfahren

Aufgrund der Abwägung der Vor- und Nachteile (siehe [Abbildung 3.4](#)) fiel die Entscheidung auf das Messen der Dehnung der Latten mittels Dehnungsmessstreifen. Dieses Verfahren gewährleistet die genaue Messung der Bewegungen im Bett, da diese direkt auf die Latten durch die Matratze weitergegeben werden. Dieses Verfahren bietet zudem einen möglichst guten Schutz vor thermischen Veränderungen, da diese durch den Aufbau ausgeglichen werden können.

Verfahren	berührungslos	lautlos	aufwand	strahlungsfrei	Wartungsfrei	Kosten	Praktikabilität	Akzeptanz
Schlaflabor	-	o	-	+	-	-	-	-
Luftmatratze	+	o	o	+	-	+	-	-
Druckpunktmatte	+	+	+	+	+	-	o	o
kapazitive Sensoren	+	o	+	-	o	-	+	o
Bewegungssensoren (Arm)	o	+	o	-	o	+	o	o
Smartphone App	+	+	o	o	+	+	o	o
Ultraschall	+	o	+	o	+	+	+	o
Infrarot	+	+	+	+	-	+	+	+
Bewegungsmelder	+	+	+	+	+	+	-	o
Dehnungsmessstreifen	+	+	+	+	+	+	+	+

Legende	Zeichen	Beschreibung
	+	trifft zu
	o	neutral
	-	trifft nicht zu

Abbildung 3.4: Vergleichsanalyse auf Grundlage der Anforderungen

### 3.3 Szenarien

Als Modell möglicher Ereignisse, zur Analyse der Aufgabe und Erstellung der Anforderungen wurde das folgende Szenario betrachtet.

Eine Person vereinbart einen Termin und trägt diesen mittels Smartphone in einen Kalender ein. Dieser Termin liegt früh morgens zu einer Zeit, in der die Person normalerweise schläft. Die Person legt sich zum Schlafen ins Bett. Das Bett analysiert den Schlaf anhand der Bewegungen und berechnet mittels Zugriff auf den Kalender den idealen Zeitraum zum Aufstehen. Das Bett registriert Bewegungen kurz vor dem eigentlichen Zeitpunkt zum Aufstehen. Die Person wird in dem Moment geweckt, in dem sie sich bewegt.

Als weitere Analysegrundlage dienten die Ausarbeitung von Jens Ellenberg ([Ellenberg](#)).

Mittels der zusammengestellten Informationen werden im Folgenden die Anforderungen formuliert, die an diese Arbeit gestellt werden.

### 3.4 Anforderungen

In diesem Abschnitt werden die funktionalen und nicht funktionalen Anforderungen an die Hard- und Software ermittelt und der Rahmen dieser Arbeit wird abgesteckt.

#### 3.4.1 Generelle Anforderungen

Auf Grundlage der zuvor analysierten Verfahren haben sich die folgenden Punkte ergeben, die zu erfüllen sind. Die Bewegungserkennung, die im Rahmen dieser Bachelorarbeit entwickelt werden soll, wird nach Fertigstellung einer Reihe von Bedingungen genügen müssen. Die

Anforderungen lassen sich wie folgt unterteilen.

#### **Anforderungen an die Grundfunktion**

Das wichtigste Kriterium, das es zu erfüllen gilt, ist die Analyse der Bewegungen und die darauf aufbauende automatische Erkennung, zu welchem Zeitpunkt die schlafende Person geweckt werden sollte.

#### **Anforderungen an die Bedienungsfreundlichkeit**

Der Nutzer soll das System verwenden können, ohne einen zusätzlichen PC neben dem Bett zu positionieren.

Ebenso soll das Bett so konzipiert werden, dass keine aufwendigen Installationen und Einmessungen nötig sind.

#### **Anforderungen an die Strahlungsreduzierung**

Es soll auch keine zusätzliche Funktechnologie erforderlich sein, hierfür ist die vorhandene Infrastruktur – WLAN oder LAN – zu nutzen.

#### **Anforderungen an die Ressourcenschonung**

Es ist ein möglichst ressourcenschonender Aufbau zu realisieren, um die Verwendung anderer Anwendungen im Living Place nicht zu stören. Hierbei ist besonders auf die CPU-Last sowie die Beanspruchung des Netzwerks und des Speichers zu achten.

#### **Anforderungen an die Mobilität**

Das Bett ist nicht als mobiler Gegenstand konzipiert, der regelmäßig bewegt und umgebaut werden soll, daher ist es nicht nötig, einen mobilen Einsatz zu gewährleisten.

#### **Anforderungen an die Umwelteinflüsse**

Die Messungen sollen zudem unabhängig von Licht, Feuchtigkeit und Temperatur fehlerfrei funktionieren und trotz unterschiedlicher Matratzenhöhe bzw. unterschiedlicher Lattenroste exakt erfolgen.



#### **Anforderungen an die Schlafhygiene**

Das System soll den menschlichen Organismus möglichst wenig beeinflussen. Um dieser Anforderung gerecht zu werden, darf die Hardware beim Aufbau keine störenden Geräusche verursachen und der Messaufbau darf keine Strahlung durch das Messverfahren verursachen, um den Körper nicht zu beeinflussen, damit die Schlafhygiene nicht beeinträchtigt wird. Für die Anwendung ist außerdem wichtig, den Anwender nicht durch Kabel, Sensoren, Armbänder oder ähnliches beim Schlafen zu stören. Die Benutzung des Bettes ist daher nicht nur strahlungsarm und lautlos, sondern auch berührungslos zu konzipieren.

#### **Anforderungen an die Kosten**

Das letzte Kriterium, das es zu erfüllen gilt, ist der Preis. Das System soll kostengünstig im Aufbau sein, damit es einer größtmöglichen Zahl von Anwendern zur Verfügung gestellt werden kann. Hieraus ergibt sich die folgende Zusammenfassung ([Abbildung 3.5](#))

#### **3.4.2 Softwareanforderungen**

Aufgrund der Struktur des Living Places und der dort eingesetzten Softwarekomponenten ergeben sich die folgenden Softwareanforderungen in Bezug auf die Kompatibilität.

##### **Kompatibilität und Erweiterbarkeit**

Mittels Software soll eine Server-Client-Kommunikation mit einer serverseitigen Vermittlungsstelle aufgebaut werden. Diese soll zum Halten der Daten der letzten 24 Stunden dienen und macht die Steuerung der Hard- und Software möglich. Es soll der im Living Place verwendete ActiveMQ zum Einsatz kommen, damit die Zustandsänderungen der Person im Bett erfasst werden und es ermöglicht wird, dass Anwendungen hierauf reagieren.

Ein weiterer Punkt, den diese Software erfüllen muss, ist, dass die Verwendung eines Arduinos erleichtert wird und die Kommunikation mittels einfacher TCP-Nachrichten erfolgen kann. Dieses ist nötig, da der Arduino nur mit einem erhöhten Aufwand mit dem ActiveMQ kommunizieren kann. Alle erfassten Messwerte sollen in einer Datenbank abgespeichert werden. Dies ermöglicht es Clients, die sich später verbinden, Zugriff auf alle bisherigen Daten zu gewähren.

##### **Clientanwendung**

Auf der Clientseite soll eine Software zur Verfügung stehen, die sowohl die aktuellen Messwerte darstellt als auch den Verlauf der Messwerte in den letzten 24 Stunden grafisch aufbereitet,

damit diese leichter ausgewertet werden können. Ebenso soll die Möglichkeit der Kommunikation mit einem serverbasierten Kalender realisiert werden, um eingetragene Termine auf der GUI des Clients darzustellen und die Möglichkeit bietet die Termine von jedem Gerät zu verwalten. Im Rahmen einer Ausarbeitung kann dies zum Wecker 2.0 erweitert werden. Der Wecker 2.0 ist ein Projekt im Living Place und wurde in einer Projektarbeit entworfen (siehe [Ellenberg](#)).

#### **Weckfunktion**

Als weiterer Punkt soll eine prototypische Weckfunktion implementiert werden, der nicht den vollen Funktionsumfang des Weckers 2.0 aufweisen. Jedoch im Fall von Bewegung Personen in einem vordefinierten Zeitraum, vor einem eingetragenen Weckereignis, weckt. Diese Art des Weckens geschieht mit der Absicht, dass die Person leichter aufstehen kann.

#### **Ressourcen**

Die Software darf im Betrieb nicht übermäßig viel der zur Verfügung stehenden Ressourcen, zur Auswertung für beide Betthälften beanspruchen, damit ein reibungsloser Ablauf anderer Programme nicht beeinträchtigt wird.

### 3.4.3 Hardwareanforderungen

Die Hardware soll störungsfrei arbeiten und nicht von Licht, Temperaturschwankungen und Feuchtigkeit beeinflusst werden. Sie darf keine Störungen durch Lautstärke oder Strahlung auf Menschen und Tiere verursachen und soll ohne zusätzliche Hardware, wie einen PC, funktionieren. Die Messungen sollen Ergebnisse liefern, die frei von Störungen sind. Hierbei ist zu beachten, dass im Living Place eine erhöhte Strahlungsbelastung durch Funkverbindungen vorherrscht. Die Hardware soll wartungsarm sein und einen kostengünstigen Aufbau gewährleisten.

Aufgrund des Aufbaus und der Verwendung der DMS ist die Widerstandsänderung minimal. Die daraus resultierende Spannung ist somit zu verstärken, um die Signale messbar zu machen.

### 3.4.4 Tabellarische Zusammenfassung

Nr.	Anforderung	Priorität	Stabilität	Komplexität	Risiko
1	Benutzerfreundlichkeit	mittel	hoch	mittel	niedrig
2	Strahlungsreduzierung	mittel	hoch	niedrig	niedrig
3	Ressourcenschonend	hoch	hoch	mittel	mittel
4	Mobilität	niedrig	niedrig	niedrig	niedrig
5	Umwelteinflüsse	mittel	mittel	mittel	mittel
6	Schlafhygiene	hoch	hoch	mittel	niedrig
7	Kosten	mittel	mittel	mittel	mittel
8	Kompatibilität	hoch	mittel	mittel	mittel

Abbildung 3.5: Anforderungsanalyse

Auf Grundlage dieser Anforderungen wird nun das geplante Design besprochen.

## 4 Design

In diesem Kapitel werden die verwendeten Komponenten gezeigt und es wird diskutiert, warum sie im Aufbau verwendet werden.

### 4.1 Hardwaredesign

Im diesem Kapitel werden die verwendeten Bauteile sowie die Alternative besprochen, um herauszustellen, welche Vorteile sich durch die Verwendung der eingesetzten Bauteile ergeben. Um die Verformungen der Latten zu registrieren, wurde nach der Auswertung und dem Vergleich verschiedener Verfahren das Messen mittels Dehnungsmessstreifen ausgewählt. Die hieraus resultierende Spannung ist zu niedrig, um sie direkt auszuwerten. Damit die Spannung digital messbar gemacht werden kann, muss eine Verstärkerschaltung entworfen werden. Damit die verstärkte Spannung digitalisiert werden kann und für verschiedene Anwendungen zur Verfügung steht, wird ein Arduino mit einem Netzwerkanschluss eingesetzt, damit dieser die anliegende Spannung messen und an einen Server übermitteln kann.

Die Entscheidung, einen Arduino zu verwenden, wurde getroffen, da der Arduino mittels eines Ethernetshields kostengünstig und geräuschlos verwendet werden kann und nur einen geringen Stromverbrauch aufweist.

Diesen Aufbau zeigt die folgende Grafik schematisch.

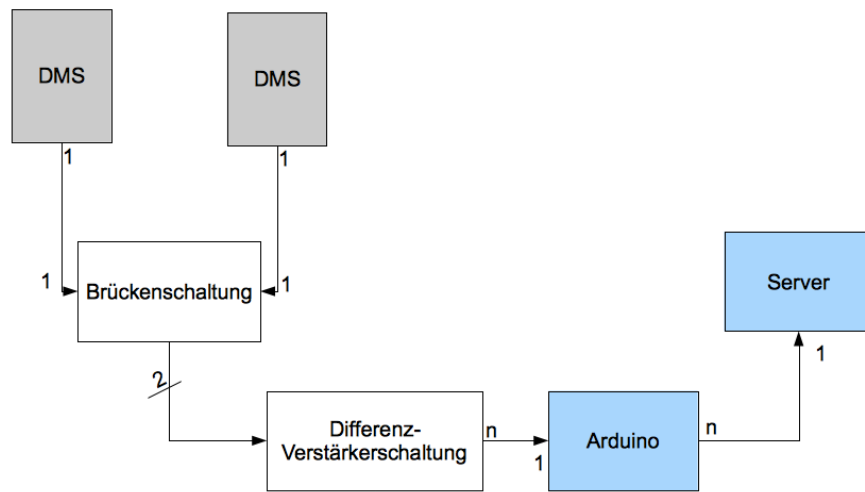


Abbildung 4.1: Schema des Aufbaus

#### 4.1.1 Mechanischer Aufbau

Auf jeder Bettseite sollen wie in der [Abbildung 4.2](#) zu sehen drei Dehnungsmessstreifenpaare je auf Schulter-, Hüft- und Wadenhöhe angebracht werden. So sollen möglichst alle Bewegungen und damit verbundene Deformationen erkannt werden können. Da diese Bereiche das höchste Gewicht tragen, wird davon ausgegangen, dass bei Bewegungen hier am einfachsten zu messen sind.

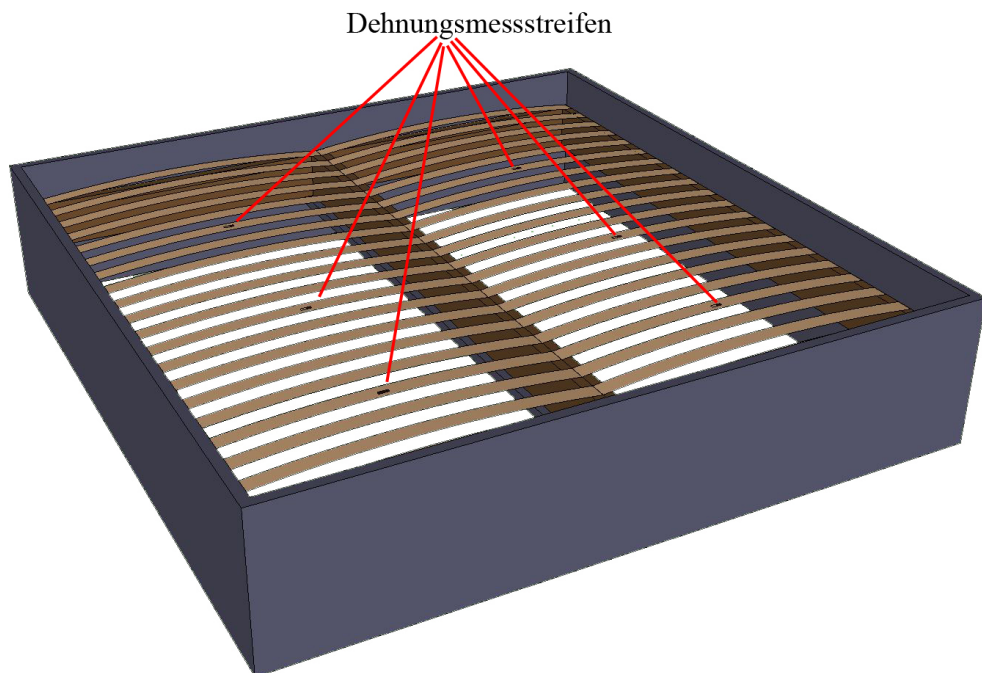


Abbildung 4.2: Modell des Bettes mit DMS auf den Latten

Die Positionierung an diesen Stellen bietet einen möglichst kurzen Weg zur Verkabelung beider Betthälften und deren Dehnungsmessstreifenpaare mit den zugehörigen Verstärkerschaltungen und dem dahinter liegenden Arduino.

Auf Grund der kurzen Wege ist gewährleistet, dass der Eigenwiderstand der Kabel, die zur Verbindung verwendet werden, nicht das Ergebnis verfälscht.

**Dehnungsmessstreifen auf den Latten** Die Dehnungsmessstreifen werden fest mit einem Trägermaterial, hier der Latte aus dem Lattenrost, verbunden. Hierbei ist es wichtig, einen speziellen Kleber zu verwenden, damit die DMS nicht anfangen zu kriechen. Der Wert des Kriechens wird bei speziellen Klebern mit angegeben und durch den k-Faktor beschrieben. Hierbei ist unbedingt auf Sauberkeit und das plane aufbringen zu achten.

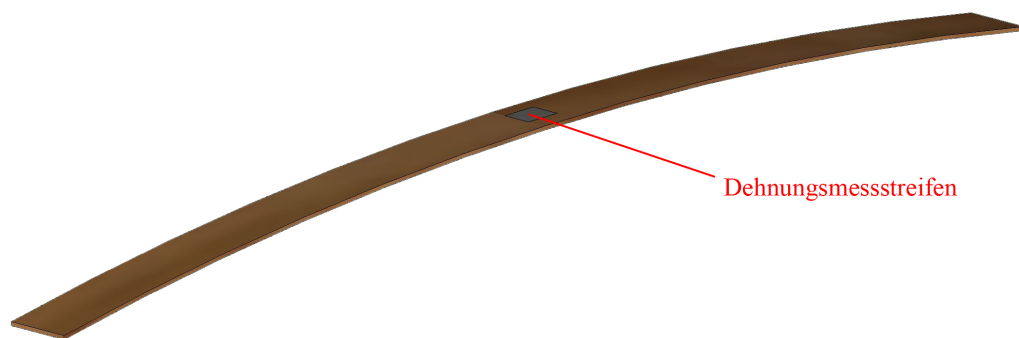


Abbildung 4.3: Modell einer Latte mit Dehnungsmessstreifen

**Dehnungsmessstreifen** Im endgültigen Aufbau werden die Dehnungsmessstreifen des Typs Y41 (Abbildung 4.4) der Firma HBM verwendet. Die Entscheidung für die Universal-Dehnungsmessstreifen ist aus den folgenden Gründen gefallen.

Die Dehnungsmessstreifen sind auf Grund ihres kompakten Aufbaus leicht zu verarbeiten. In Rücksprache mit dem Hersteller hat sich herausgestellt, dass die Verwendung eines längeren Messgitters keine erhöhte Genauigkeit im Messverhalten bringt, da durch die einheitliche Oberflächenspannung die Dehnung gleichmäßig auf der Oberfläche der Latten des Lattenrostes verteilt ist. Eine Erhöhung der Größe würde lediglich Nachteile bei der Verarbeitung bringen, da die DMS hierdurch schwerer zu applizieren wären.

Zur Befestigung der Dehnungsmessstreifen auf dem Lattenrost wurde der kalthärtende Universal Kleber X60 verwendet, da der Kleber ohne eine Vorrichtung zur Fixierung der Dehnungsmessstreifen verwendet werden kann. Würde hier ein heiß-härtender Kleber verwendet, würden bei der Aushärtung Temperaturen bis zu  $180^{\circ}\text{C}$  entstehen. Zudem härtet der Kleber X60 in wenigen Minuten vollständig aus und ist somit schnell voll belastbar.

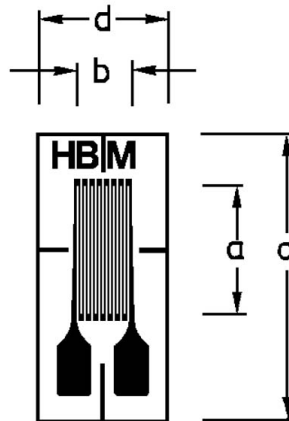


Abbildung 4.4: Verwendete Dehnungsmessstreifen

Der Dehnungsmessstreifen hat mit einer Gesamtlänge von knapp 20 mm und einer Gesamtbreite von rund 10 mm genügend Platz, um auf einer Latte verklebt zu werden. Der Eigenwiderstand im unbelasteten Zustand beträgt  $120 \Omega$ .

Dies ist ein Standardwiderstandswert und wurde gewählt, da dies durch die Massenfertigung einen besonders günstigen Preis gewährleistet. Es wäre theoretisch möglich, einen anderen Wert zu wählen, hierfür müsste die Wheatstone'sche Brückenschaltung angepasst werden. Zur Verwendung kann jeder beliebige Dehnungsmessstreifen kommen, der die Anforderungen an das Dehnungsverhalten erfüllt. So sollte darauf geachtet werden, dass der Dehnungsmessstreifen für die starken Deformationen konzipiert ist.

Um die Signale, die von den DMS auf den Latten erzeugt werden, auswerten zu können, wird eine Verstärkung der Signale benötigt. Die Signalstärke beträgt lediglich 100-800 mV. Der hieraus resultierende Digitalwert wäre zu gering, um eine signifikante Änderung der Messwerte zu erkennen.



### 4.1.2 Verstärkerschaltung

In diesem Unterkapitel wird die Verstärkerschaltung und deren verwendete Komponenten besprochen.

Für die Kozeptionierung der Verstärkerschaltung wird von einer maximalen Spannung von 1 V ausgegangen, die von der Brückenschaltung geliefert wird. Bei der Spannungsänderung wird von 0,001 V ausgegangen. Hier wird eine Verstärkung von  $\times 1000$  benötigt, um eine klare Änderung zu messen. Die Verstärkerschaltung lässt sich in die folgenden drei Teile unterteilen.

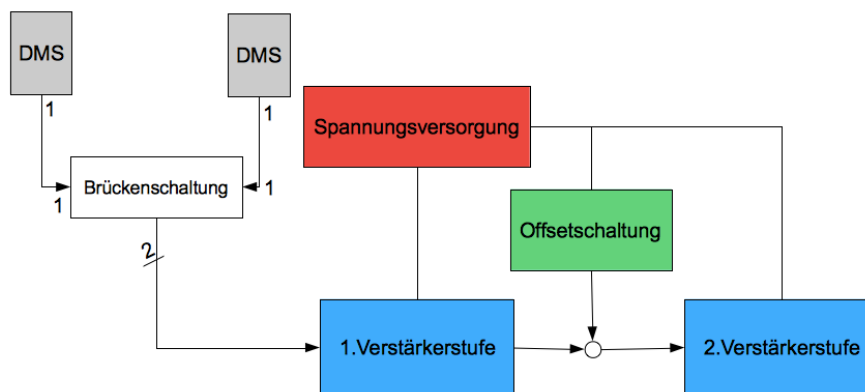


Abbildung 4.5: Schema der Verstärkerschaltung

In die Verstärkerstufen, die die Signale aus der Wheatstone'schen Brückenschaltung verstärken sollen, sodass an ihren Ausgängen eine klar definierte, rauschfreie, deutlich messbare Spannung anliegt. Die Offset-Schaltung um die Verstärkerschaltung abgleichen zu können, damit ein möglichst großer Wertebereich zur Messung und Spannungsänderung zur Verfügung steht und der Spannungsversorgung die sowohl für die Versorgung der Verstärkung als auch der Offsetschaltung dient.

**Aufbau der Verstärkerschaltung** Eine Möglichkeit, die Verstärkung der Differenzspannung zu realisieren, ist das Verstärken mittels Transistoren wie hier ([Elektronik-Kompendium](#)) beschrieben zu realisieren. Hier würde die Schaltung „Differenzverstärker im Gleichtaktbetrieb“, wie in [Abbildung 4.6](#) gezeigt, verwendet werden. Dies hätte den Vorteil eines kostengünstigen Aufbaus.

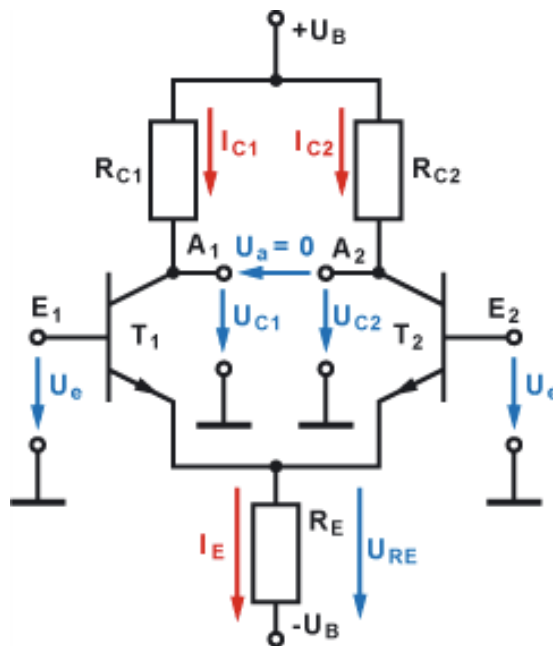


Abbildung 4.6: Aufbau einer Differenzverstärkung mittels Transistoren  
Elektronik-Kompodium

Diese Schaltung bietet die Möglichkeit, auf der linken und rechten Seite der Schaltung eine Eingangsspannung anzulegen, die die Transistoren verstärkt.

Mittels dieser Transistoren wird die Auswirkung der Widerstandsänderung verstärkt, so liegt in der Mitte der Schaltung die verstärkte Spannung als  $U_a$  an. Hier wurde jedoch befürchtet, dass auftretendes Schwingen das Ergebnis zu stark beeinflusst und somit die Messung unbrauchbar machen würde.

Zusätzlich ist eine so hohe Verstärkung nur durch eine Modifizierung der Schaltung realisierbar, was im Fehlerfall zur Zerstörung des verwendeten Arduino führen könnte. Ebenso wäre die Schaltung durch thermische Einflüsse zu stark beeinflussbar, so dass dieser Ansatz verworfen wurde. Die Verstärkerschaltung ist aus diesen Gründen mittels einem PGA204 (siehe [Abbildung 4.7](#)) realisiert worden. Hierbei handelt es sich um einen Instrumentenverstärker. Instrumentenverstärker werden in der Musikindustrie zur Verstärkung von Musikinstrumenten verwendet. Dieser Einsatzzweck benötigt eine hohe Verstärkung und gleichzeitig ein geringes Rauschen.

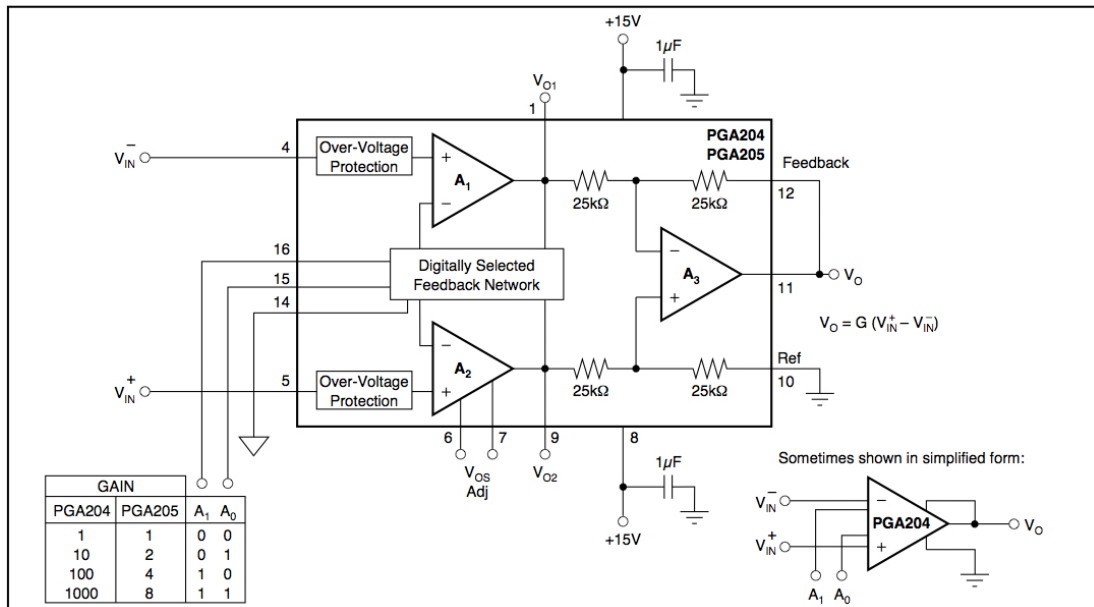


Abbildung 4.7: Aufbau des PGA204

TI

Die Entscheidung fiel auf den hier verwendeten PGA204, da das Eingangsrauschen, wie in [Abbildung 4.8](#) zu sehen, sehr gering ist. Das Rauschen beträgt bei einer Versorgungsspannung von  $\pm 15V$  und einer Raumtemperatur von  $25^\circ C$  maximal  $0.4\mu V$ . Die Eingangsspannung aus der Wheatstone'schen Brückenschaltung beträgt  $100 - 800mV$ , so ist das Rauschen nicht signifikant und schlägt nicht bis zum Ausgang durch, da das SNR-Verhältnis  $250 : 1$  bis  $2000 : 1$  beträgt.

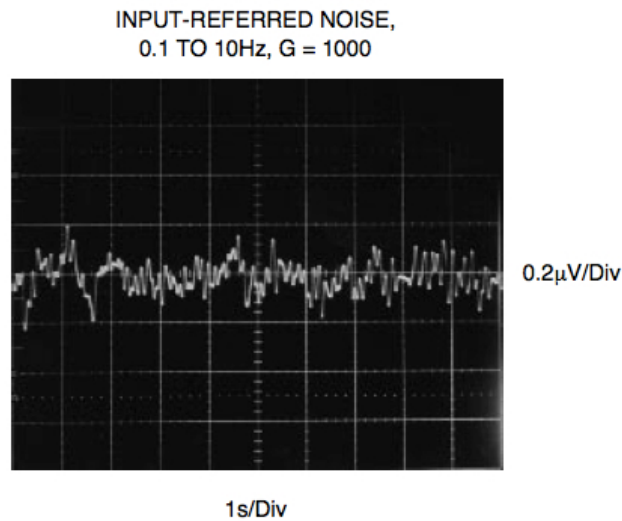


Abbildung 4.8: Eingangsruschen des PGA204

TI

Der PGA204 erfüllt somit die Anforderungen der hohen Verstärkung bei geringem Rauschen. Er ist mittels vier Operationsverstärkern aufgebaut, die zur Trennung und Verstärkung dienen. Die nachfolgende Beschreibung zeigt den Einsatz dieses Bausteins und die verwendeten Möglichkeiten zur Justierung.

Der Aufbau der Verstärkerstufen sieht wie in [Abbildung 4.9](#) veranschaulicht aus. Es wird eine doppelte Verstärkung verwendet, da durch verschiedene Einflüsse eine erhöhte Verstärkung benötigt werden kann.

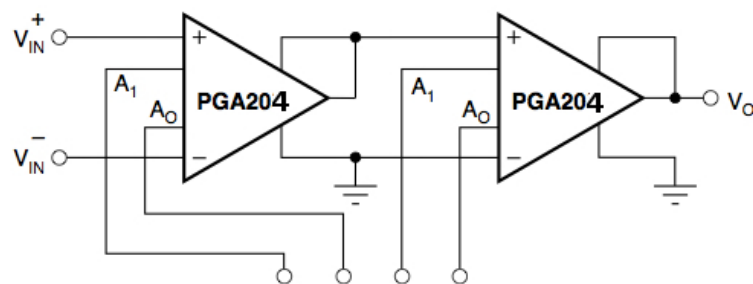


Abbildung 4.9: Grundlage der Verstärkerschaltung des PGA204

(vgl. TI)

Die Verstärkerschaltung besteht aus zwei Teilen. Der erste dient zur Verstärkung der Dif-

ferenzspannung zwischen den Widerständen in der Wheatstone'schen Brückenschaltung. Dessen Ausgang liefert eine Spannung von 0-15 V. Dieser Ausgang wird an den einen der zwei Eingänge des zweiten PGA204 geschaltet und der zweite Eingang wird auf Masse gezogen. Der zweite Teil kann zusätzlich dieses Signal um das 1000-fache verstärken, wenn dies benötigt wird. Dies kann bei dicken Matratzen oder bei starken Lattenrosten der Fall sein, da diese sich nicht so leicht verformen lassen. Die maximale Ausgangsspannung beträgt +15 V. In der ersten

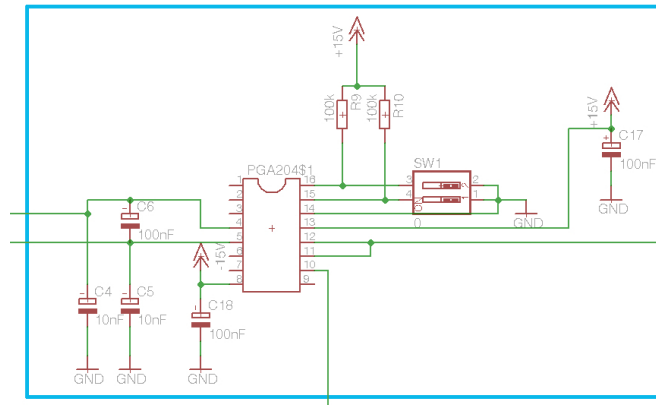


Abbildung 4.10: Verstärkerschaltung Teil 1

Verstärkerstufe dient der Eingang 10 auf der Ausgangsseite zur Offsetsteuerung. Hierfür wird die Offsetschaltung (Abbildung 4.14) verwendet.

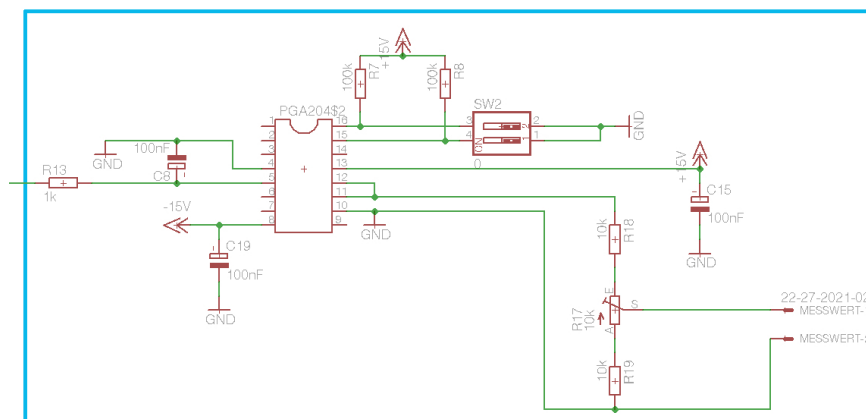


Abbildung 4.11: Verstärkerschaltung Teil 2

In der zweiten Verstärkerstufe wird der Eingang 10 auf Masse gezogen und es findet keine erneute Offsetsteuerung statt. Dies dient zum einfacheren Messen mittels Arduino. Hier ist keine Referenzspannung nötig und die Messung findet in Bezug zur Masse statt.

**Konfiguration der Verstärkerschaltung** Der PGA204 ([Abbildung 4.7](#)) besitzt vier Eingänge auf der Eingangsseite (4, 5, 15, 16), die für diese Schaltung relevant sind.

Die Eingänge 4 und 5 bilden die zu verstärkende Differenzspannung. Eingang 15 und 16 dienen zur Steuerung der Verstärkung wie in [Abschnitt 4.1.2](#) beschrieben. Auf der Ausgangsseite muss der Ausgang 12 auf den Eingang 11 zurückgeführt werden, damit der Operationsverstärker richtig regelt und ein Übersteuern verhindert wird. Der Ausgang 10 wird als Referenzspannung verwendet und kann zur Offsetsteuerung dienen.

Verstärkung	A0	A1
1	0	0
10	0	1
100	1	0
1000	1	1

TI

Tabelle 4.1: Verstärkungseinstellung des PGA204

### 4.1.3 Spannungsversorgung

Die Schaltung benötigt eine Spannungsversorgung von  $\pm 15V$ , dies gewährleistet ein Schaltnetzteil mit einer Festspannung von  $\pm 15V$ . Die Spannung dient als Versorgungsspannung für die Impedanzwandler und die Instrumentenverstärker PGA204. Als Brückenspeisung und zur Versorgung der Offsetschaltung ([Unterabschnitt 4.1.4](#)) wird eine Spannung von  $\pm 2.5V$  benötigt, hierfür sorgt die folgende Spannungsversorgung.

**Aufbau der Spannungsversorgung** Die Spannungsversorgung ([Abbildung 4.12](#)) lässt sich in zwei Teile gliedern. Der erste besteht aus einem ADR291 und liefert eine Spannung von 2.5 V. Der zweite Teil dient zur Stabilisierung und zum Wandeln der Impedanz. So kann eine Spannung von  $\pm 2.5V$  abgegriffen werden.

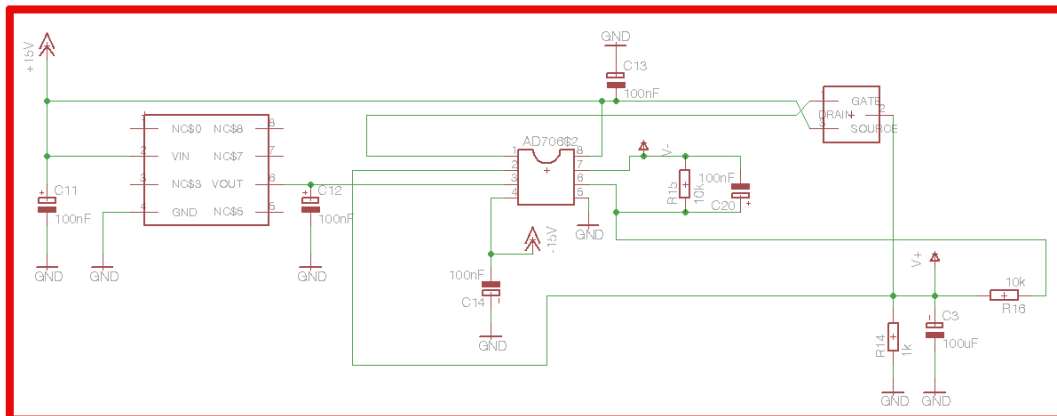


Abbildung 4.12: Spannungsversorgung

Der hier eingesetzte ADR291 wurde verwendet, da dieser eine sehr stabile Spannung ohne die Gefahr des Schwingens liefert. Hier wäre jeder andere Festspannungsregler ebenso geeignet. Die Entscheidung fiel auf den ADR291, da der ADR291 bereits früher mit guten Erfahrungen verwendet wurde. Ebenso ist der zweite Teil durch jeden Operationsverstärker realisierbar. Der hier verwendete Operationsverstärker ist wie die Spannungsversorgung nach einer Veröffentlichung von **Neuendorf** konzipiert worden.

#### 4.1.4 Offsetschaltung

Um die Verformung der Dehnungsmessstreifen durch das Gewicht der Matratze auszugleichen und um die Schaltung im durch die Matratze belasteten Zustand auf null abzugleichen, wird eine Offsetschaltung benötigt.

**Realisierung der Offsetschaltung** Die Offsetschaltung (**Abbildung 4.14**) besteht aus einem Spannungsteiler, der eine Spannung von  $\pm 2.5V$  liefert. Der Ausgang des Spannungsteilers, wird in einen AD706 gegeben, dieser wird mit einem hochohmigen Eingang als Spannungsfolger eingesetzt. Der AD706 ersetzt den in der folgenden Schaltung gezeigten OPA177. Die Schaltung (**Abbildung 4.13**) wurde als Vorlage zum Entwurf der Schaltung verwendet. In dieser Schaltung wird der Spannungsteiler, der zum Einsatz gekommen ist, mit der anschließenden Einfachverstärkung gezeigt.

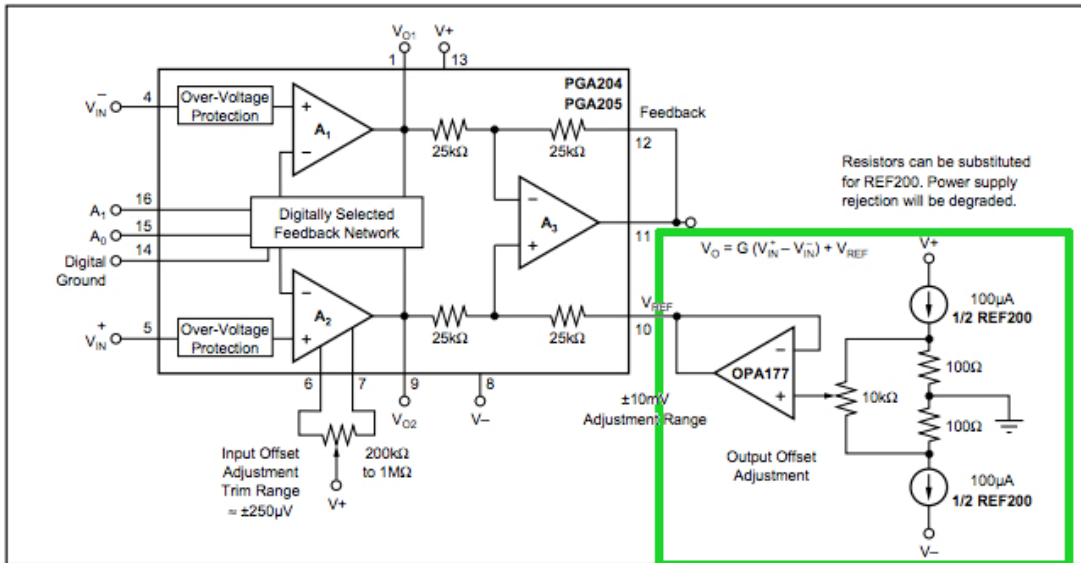


Abbildung 4.13: Offsetschaltung aus dem Datenblatt

TI

Das hier gezeigte Design wurde leicht verändert in die fertige Schaltung übernommen, da dieser Aufbau ein sehr gutes Ergebnis geliefert hat. Lediglich der OPA177 wurde durch den AD706 ersetzt, da dieser ähnliche Eigenschaften besitzt und bereits in anderen Teilen der Schaltung verwendet wurde. Die folgende Schaltung (siehe [Abbildung 4.14](#)) zeigt den Aufbau im Schaltplan.



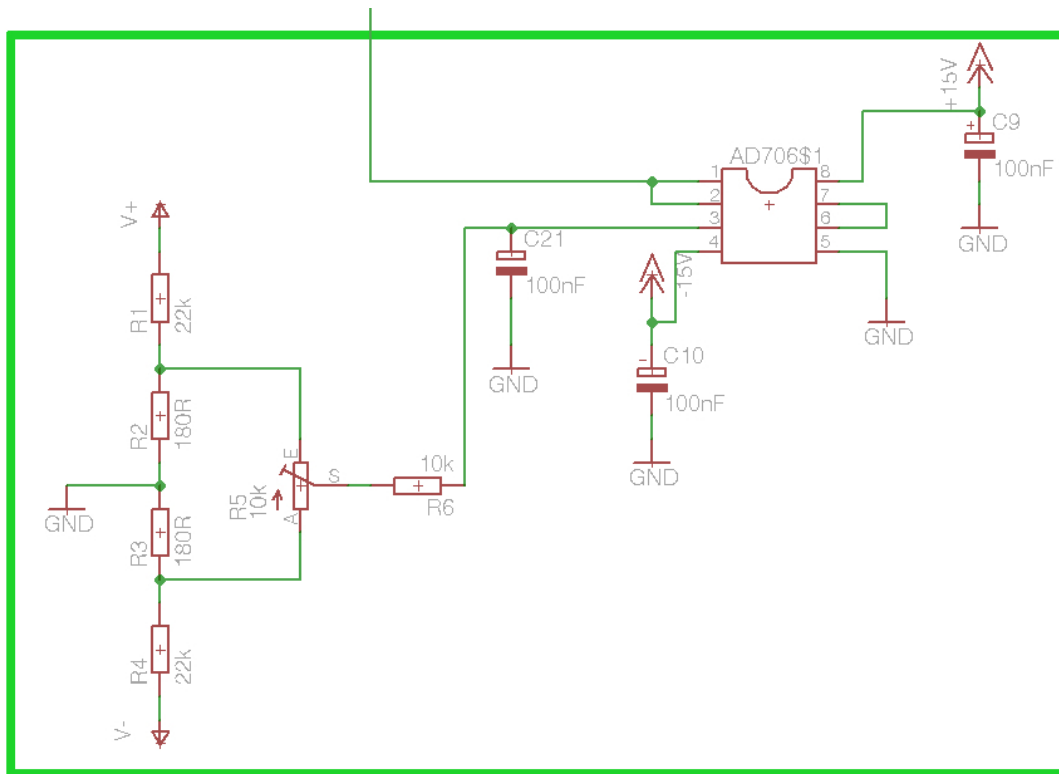


Abbildung 4.14: Offsetschaltung

Die Möglichkeit der Eingangsoffsetsteuerung über die Eingänge 6 und 7, wie in der Schaltung in [Abbildung 4.13](#) beschrieben, wurde nicht genutzt, um eventuelle Störeinflüsse zu vermeiden. Die verwendete Offsetsteuerung wirkt direkt auf die Spannung am Ausgang 11.

### 4.1.5 Schaltplan und Board

Der hier gezeigte Schaltplan stellt den Entwurf der Schaltung dar. Auf Grundlage dieses Schaltplans ([Abbildung 4.15](#)) wurde das Board ([Abbildung 4.16](#)) entworfen. Hierbei handelt es sich um einen in Eagle entwickelten Schaltplan sowie das dazugehörige Board. Die hier gezeigte Schaltung wurde in dieser Form in der fertigen Ausfertigung produziert und wird im Bett eingesetzt werden.

**Stromversorgung der Schaltung** Die Versorgungsspannung von  $\pm 15V$  wird an einen 3-Pin-Molex-Anschluss mittig auf der Platine angelegt.

**Anschluss der Dehnungsmessstreifen an die Schaltung** Auf der Platine an der rechten Seite befindet sich ein 4-poliger Molex-Anschluss, dieser liefert  $+2.5 V$  als Versorgungsspannung für die Messbrücke, den Massebezug sowie zwei Pins für die Spannungsanalyse der Messbrücke. Der Massebezug dient zusätzlich für die Schirmung der Kabel, sodass äußere Einflüsse, wie WLAN-Sender, minimiert werden können.

**Mechanische Einstellmöglichkeiten auf der Platine** Auf der Platine befinden sich zwei Potentiometer, einer zur Steuerung der Offsetschaltung und einer, der den Spannungsteiler für die Ausgangsspannung reguliert. Das Potentiometer zur Steuerung der Offsetschaltung befindet sich vertikal auf der Platine. Das Potentiometer zur Regulierung der Ausgangsspannung ist waagrecht auf der Platine angebracht.

## 4 Design

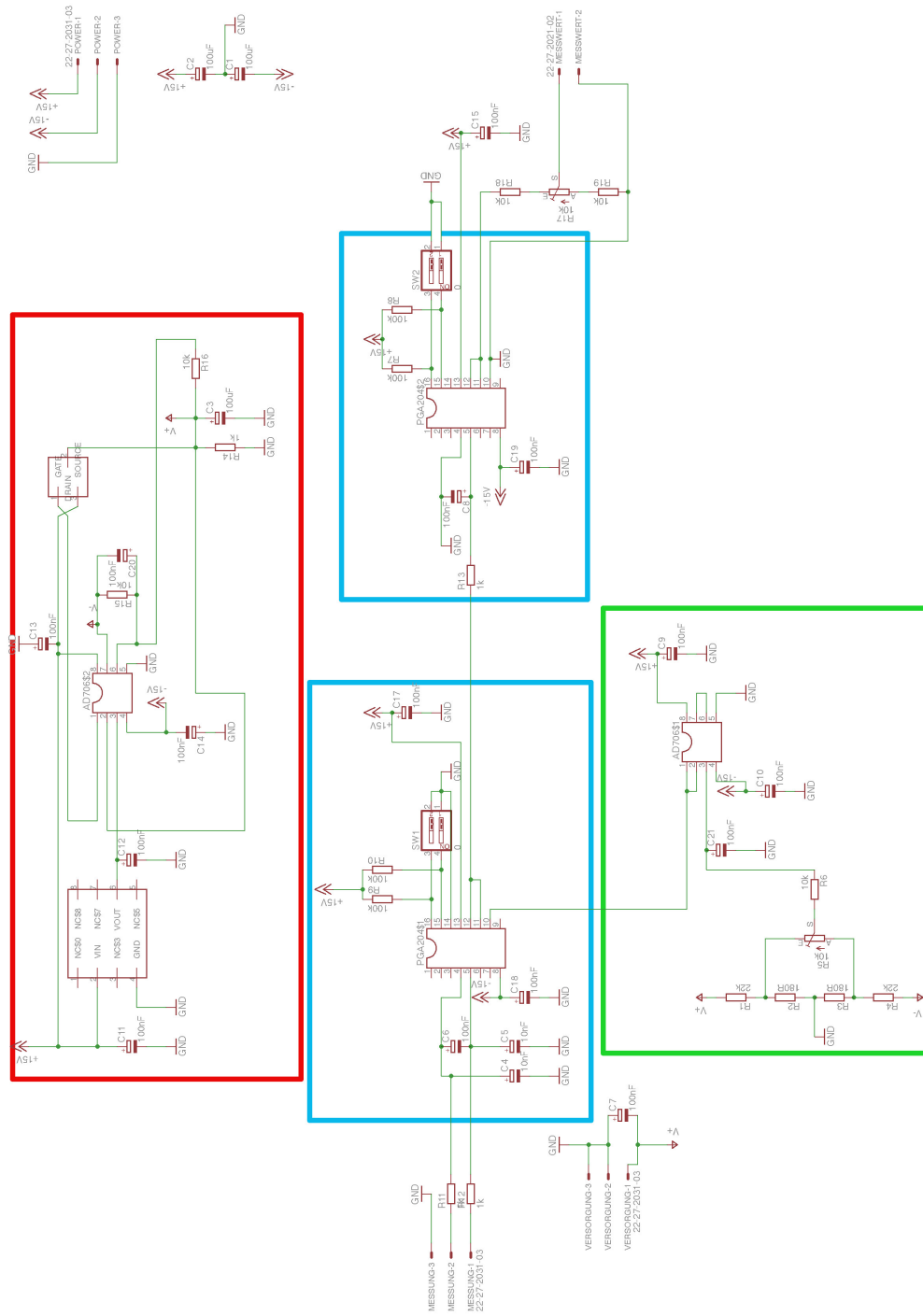


Abbildung 4.15: Schaltplan für das Layout

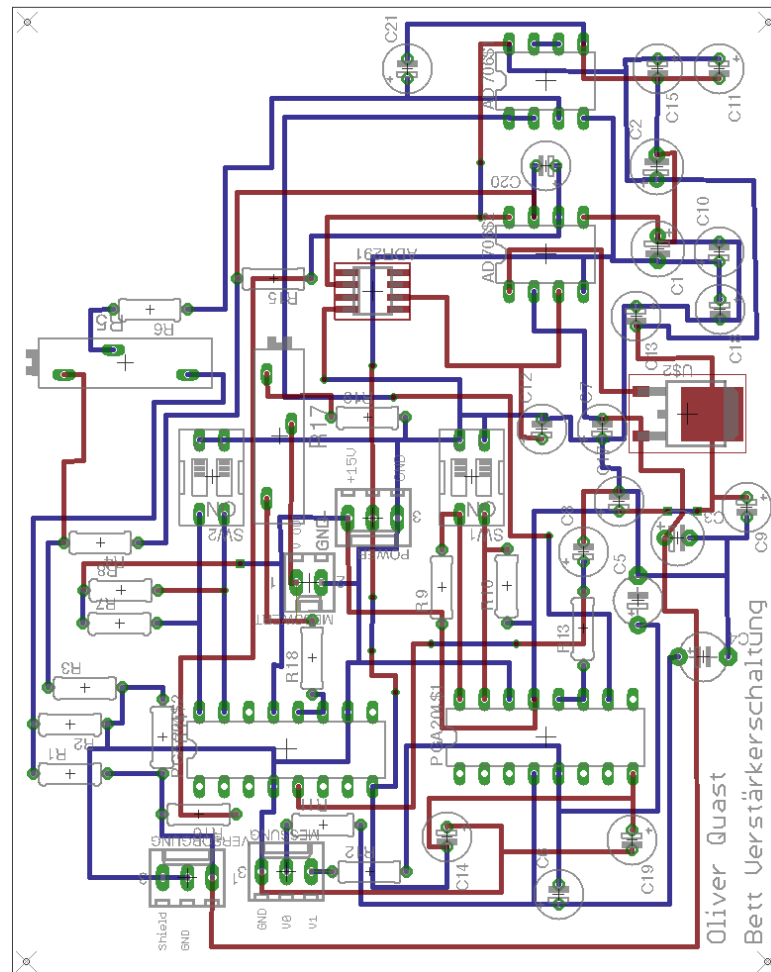


Abbildung 4.16: Platinen-Layout

### Überleitung

Nachdem die einzelnen Komponenten besprochen und ihre Funktionsweise erläutert wurden, wird nun der Aufbau, der auf Grundlage dieses Designs getroffen wurde, gezeigt.

## 5 Realisierung und Schlafanalyse

Dieses Kapitel beschäftigt sich mit dem Aufbau der Hard- und Software. Es wird auf die Realisierung des Aufbaus eingegangen und die entwickelte Software vorgestellt. Zudem wird eine Schlafkurve analysiert und gezeigt, wie der Prototyp eines Weckers entwickelt wurde. Im Anschluss wird gezeigt, welche Probleme bei der Entwicklung aufgetreten sind und welche Lösungen für diese Probleme entwickelt wurden.

### 5.1 Realisierung des Aufbaus

Wie im Designkapitel besprochen, wurden auf den Latten des Bettes die Dehnungsmessstreifen aufgebracht und so verteilt, dass die Dehnungsmessstreifen auf den Latten angebracht wurden, auf denen das meiste Gewicht liegt. Dieser Aufbau gewährleistet eine größtmögliche Deformationsänderung bei Bewegungen. So wurde je ein Paar Dehnungsmessstreifen auf Höhe der Schultern, der Hüfte und der Waden angebracht und mit der unter dem Bett liegenden Verstärkerschaltung verbunden.



Abbildung 5.1: Bett mit Wake-Up-Scene

Auf der [Abbildung 5.2](#) ist der Dehnungsmessstreifen auf der Latte zu sehen. Dies stimmt mit dem im Designkapitel angenommenen Aufbringen überein. Jedoch mussten die Anpassungen, mittels Lötstützpunkte die Anschlüsse zu verstärken, mit aufgenommen werden. Das Bild zeigt die endgültige Version mit allen eingeflossenen Verbesserungen, die im Testzeitraum entworfen wurden. Unter dem schwarzen Klebeband ist, farblich markiert, die in einer Schlaufe liegende Zuleitung. Am Ende dieser Zuleitung ist der Anschluss aufgeklebt, der zum Anschluss der Wheatstone'schen Brückenschaltung dient.

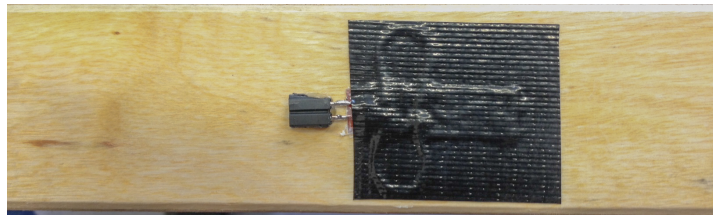


Abbildung 5.2: Präparierte Latte aus dem Lattenrost

In der [Abbildung 5.3](#) ist der Dehnungsmessstreifen zu sehen der auf der Latte appliziert wurde.

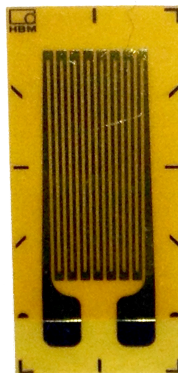


Abbildung 5.3: Verwendeter Dehnungsmessstreifen

In [Abbildung 5.4](#) ist die verwendete Schaltung zu sehen.

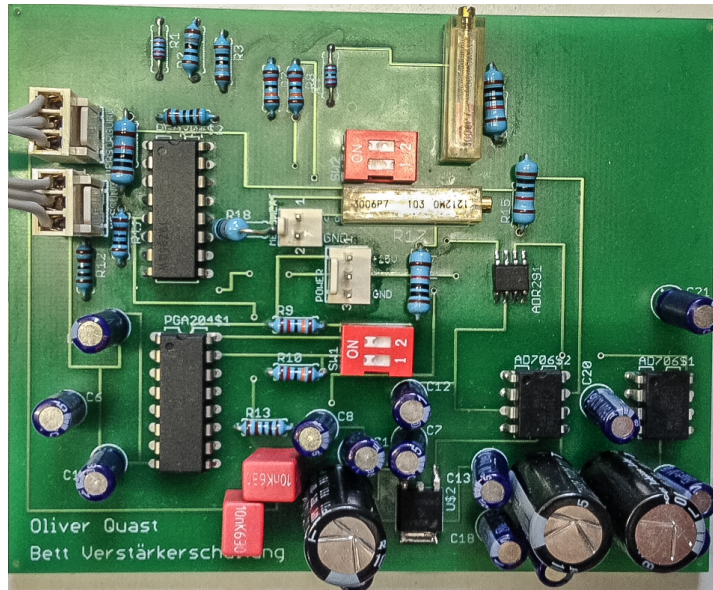


Abbildung 5.4: Verstärkerschaltung

## 5.2 Softwarerealisierung

### Arduino

Auf dem Arduino läuft eine Firmware, die in der Hauptroutine Messwerte vom auf dem Arduino-Board integrierten A/D-Wandler ausliest. Diese Messwerte werden an den Server gesendet.

Die Verbindung mit dem Server kann wahlweise über ein Ethernet- oder ein WiFi-Shield erfolgen. Der Aufbau dieser Komponente ist im [Unterabschnitt 5.2.1](#) beschrieben.

### Server

Der Server dient als Kommunikationsschnittstelle zwischen dem Arduino und der Client-Software. Zudem dient der Server als Informationsstelle. Hier können die Clients sich alte Messwerte holen, um einen Graphen zu erstellen.

Die Komponente Server ermöglicht es einer beliebigen Anzahl an Arduinos und Clients, die

Verbindung aufzubauen sowie Messwerte zu senden und zu empfangen. Der Aufbau dieser Komponente ist im [Unterabschnitt 5.2.2](#) beschrieben.

### Client

Die implementierte Client-Software dient zur visuellen Darstellung der Messwerte. Diese werden auf einem Graphen im Hauptfenster angezeigt. Zusätzlich ist zudem als Prototyp ein Wecker integriert worden. Dieser lässt sich mittels eines Google-Kalenders mit jedem beliebigen Gerät, das Zugriff auf den Kalender besitzt, stellen. Der Wecker nutzt die Messwerte, um Bewegungen zu erkennen und im Fall einer Bewegung das Weckereignis in einem definierten Zeitraum vor dem eigentlichen Weckereignis auszulösen. Der Aufbau dieser Komponente ist im [Unterabschnitt 5.2.3](#) beschrieben.

Die folgende schematische Darstellung ([Abbildung 5.5](#)) zeigt die Verbindung der einzelnen Komponenten.

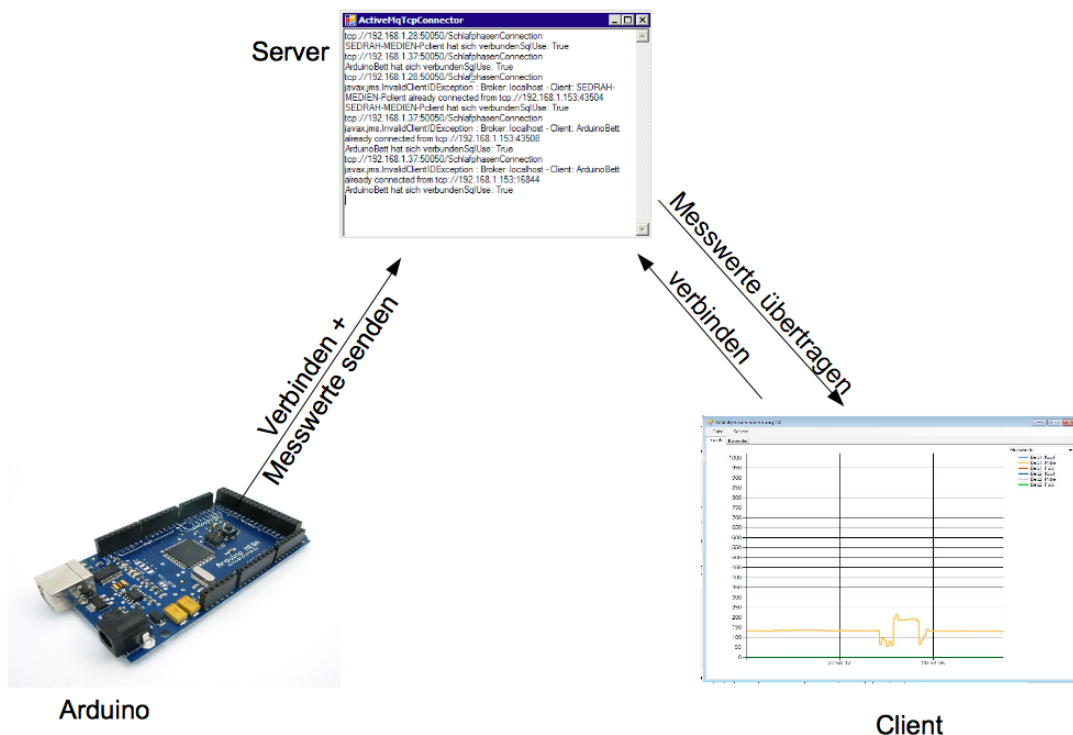


Abbildung 5.5: Schema des Aufbaus



### 5.2.1 Arduino Softwarerealisierung

Zur Digitalisierung der DMS-Daten kommt ein Arduino zum Einsatz. Dieser nutzt den integrierten A/D-Wandler, um die Messungen durchzuführen.

**Messung** Auf der Arduino-Seite wird mittels der Funktion `AnalogRead(PIN)` der analoge Wert des angegebenen Pins gelesen und in einen digitalen Wert zwischen 0 und 1023 umgewandelt. Es werden 16 Messwerte / sec aufgenommen und daraus jeweils der Mittelwert gebildet. Dies ermöglicht kleinere Schwankungen ohne Qualitätsverlust des Signals herauszufiltern und dies möglichst ressourcenschonend zu realisieren.

**Nachrichtenaufbau** Die Nachricht wird im folgenden Format an den ActiveMQConnector übermittelt:

```
BETT<ID>:<KOPFMITTELWERT>:<MITTEMITTELWERT>:<FUSSMITTELWERT>
```

Im Rahmen der Kommunikation wurde die Alternative der Nutzung des JSON-Nachrichtenformates betrachtet.

Dieses schied aufgrund einer fehlenden Kommunikationsmöglichkeit auf der Arduino Seite aus. Hier existiert ein entsprechendes Projekt ([Nowotny](#)), jedoch läuft dieses für die hier geforderte Stabilität nicht zufriedenstellend. Aus diesem Grund wurde auf die Verwendung des JSON-Formates verzichtet.

Um die so zusammengestellten Nachrichten zu übertragen, kommt wahlweise ein Ethernet- oder WiFi-Shield zum Einsatz.

**Steuerung des Arduinos über Hardware** Auf der Oberseite des Arduinos ist ein Shield aufgebracht worden, auf dem sich zwei Dip-Schalterreihen mit jeweils zehn Dip-Schaltern befinden. Diese Schalter dienen zur Steuerung der Netzwerke und der Dauer der Messzyklen. Die erste Reihe der Dip-Schalter dient zur Steuerung der Messdauer, hier sind die Dip-Schalter als Binärwerte zu betrachten. Der eingestellte Wert beschreibt die Dauer zwischen den gelieferten Messwerten von Arduino.

Die zweite Reihe beschreibt mittels der Dip-Schalter 9 und 10 das entsprechende Netzwerk. Die Belegung ist unter [Abschnitt 5.2.1](#) beschrieben.

Dip-9	Dip-10	SSID	IP
0	0	PrivatesNetz1	192.168.1.x
0	1	PrivatesNetz2	192.168.0.x
1	0	LivingplaceBuero2G	172.16.11.x
1	1	LivingplaceSensor	172.16.0.x

Tabelle 5.1: Netzwerkbelegung Dip 9 und 10

Die Dip-Schalter 1 bis 8 dienen der Bestimmung des letzten IP-Blocks des zuvor ausgewählten Netzwerks. Die so erstellte IP-Adresse gibt die Adresse des Servers an, mit dem sich der Arduino verbinden soll.

**Ethernet-Shield** Die Verwendung des Ethernet-Shields von Arduino ist dank der Ethernet-Library möglich. Hierbei ist ein bestimmter Ablauf wichtig und ein paar Punkte müssen beachtet werden.

```
1 byte mac [] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
2 byte eigeneip [] = { 192, 168, 1, 10 };
3 byte server [] = { 192, 168, 1, 153 };
```

Listing 5.1: Benötigte Parameter für das Ethernet-Shield (vgl. [Watterott](#))

Es ist wichtig, dem Ethernet-Shield eine eigene, eindeutige MAC-Adresse zuzuweisen. Ebenso wird die IP-Adresse des Shields angegeben, und um eine Client-Server-Verbindung aufzubauen ist die Zieladresse des Servers wichtig.

Das Ethernet-Shield wird mit der folgenden Befehlszeile aktiviert.

```
1 Ethernet.begin(mac, eigeneip);
```

Listing 5.2: Ethernet-Shield starten ([Watterott](#))

Hier werden die MAC-Adresse und die eigene IP-Adresse übergeben und das CS für den Ethernet-Controller auf dem Board wird aktiviert.

```
1 Client client(server, port);
```

Listing 5.3: Clientrolle definieren ([Watterott](#))

Mittels dieser Zeile wird eine Clientrolle erzeugt, die es ermöglicht, Daten von dem angegebenen Server unter dem Port zu lesen. Die Verbindung zu diesem Server wird über die Codezeile aufgebaut:

```
1 client.connect();
```

Listing 5.4: Client connect (Watterott)

Nun ist es möglich, mittels der Methoden `client.read()` und `client.write(char*)` Daten zu lesen beziehungsweise zu schreiben. Über die Methode `client.available()` kann geprüft werden, ob sich Daten im Empfangspuffer des Controllers befinden (vgl. [Company](#)).

**WiFi-Shield** Als WiFi-Shield wurde das RedFly-WiFi-Shield verwendet, da hiermit im Living Place gute Erfahrungen gemacht wurden und es mit den meisten Arduino-Boards kommunizieren kann.

```
1 byte ip [] = { 192,168, 0, 30 }; //ip from shield (client)
2 byte netmask [] = { 255,255,255, 0 }; //netmask
3 byte gateway [] = { 192,168, 0,100 }; //ip from gateway/router
4 byte dnsserver [] = { 192,168, 0,100 }; //ip from dns server
5 byte server [] = { 0, 0, 0, 0 }; //ip from server
```

Listing 5.5: Benötigte Parameter RedFly-Shield (Watterott)

Hier ist der Ablauf jedoch ein wenig anders.

Das RedFly-Shield wird mittels

```
1 RedFly.init();
```

Listing 5.6: init Redfly-Shield (Watterott)

initialisiert. Hierbei wird das CS-Signal auf High gesetzt, damit alle folgenden Befehle interpretiert werden. Durch das einfache `init` ohne Parameter wird mit dem RedFly-Shield eine Kommunikation mit einer Baudrate von 9600 ausgehandelt und die Signalstärke des Shields wird auf High gesetzt. Anschließend muss mittels des Befehls

```
1 RedFly.scan();
```

Listing 5.7: Scan des Netzwerks mittels RedFly-Shield (Watterott)

das Netzwerk nach Accesspoints gescannt werden. Ist dieser Befehl erfolgreich durchgeführt, wird die Verbindung mit einem gewünschten Netzwerk aufgebaut.

```
1 RedFly.join("wlan-ssid", "wlan-passw", INFRASTRUCTURE);
```

Listing 5.8: Verbindungsaufbau zum Netzwerk mit RedFly-Shield (Watterott)

Jetzt kann die Verbindung zu diesem Netzwerk aufgebaut werden und es werden IP-Adresse, DNS und Gateway ausgehandelt oder übergeben.

```
1 ret = RedFly.begin(); //DHCP nutzen
2 //oder
3 ret = RedFly.begin(ip,dnsserver,gateway,netmask); //kein DHCP
```

Listing 5.9: Verbindungsaufbau abschließen mit RedFly-Shield (Watterott)

Nachdem dieser Befehl erfolgreich durchgeführt wurde, kann nun mittels

```
1 http = RedFly.socketConnect(PROTO_TCP, server, 80);
2
3 RedFly.socketSendPGM(http, PSTR("GET / HTTP/1.1\r\n"));
4 RedFly.socketRead(&sock, &len, buf, sizeof(buf));
```

Listing 5.10: Datentransfer RedFly-Shield (vgl. Watterott)

socketConnect eine Socket-Verbindung aufgebaut werden und den dazu passenden socketSendPGM und socketRead Nachrichten übermittelt beziehungsweise gelesen werden.

### 5.2.2 Server - Verbindung von Arduino zu ActiveMQ

Für die Kommunikation der einzelnen Softwaremodule wurde eine Anwendung benötigt, die als Vermittlungsstelle zwischen dem Arduino, dem ActiveMQ, einer Datenbank und den Clients dient (siehe [Abbildung 5.6](#)).

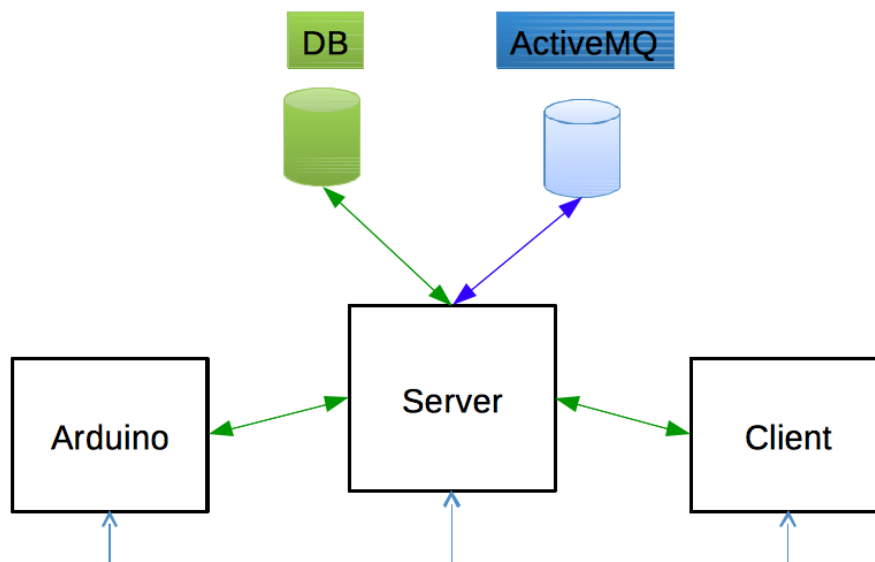


Abbildung 5.6: Kommunikationsschema: Kommunikation Arduino -> Server -> Client

Der Server bietet die Möglichkeit, direkt vom Arduino TCP-Nachrichten zu senden, die mit Hilfe der Serveranwendung im ActiveMQ abgelegt werden. Zusätzlich kann die Nachricht in einer MySQL-Datenbank gespeichert werden. Jeder verbundene Client, der auf Nachrichten wartet, wird mittels Remoting über die Nachrichten informiert und muss nicht auf neue Ereignisse pollen. Dies dient zur Graphenerstellung, ohne einen enormen Overhead zu verursachen. Im MySQL Server sind alle Nachrichten der letzten 24 Stunden gehalten. So kann jeder Client einen kompletten Graphen erstellen und der ActiveMQ-Server bleibt lediglich zur Kommunikation und zum Statusupdate.

Um diesen Dienst nutzen zu können, muss lediglich ein Protokoll eingehalten werden, sodass die Kommunikation erfolgreich abgewickelt werden kann.

Beim Verbindungsaufbau ist es wichtig, dass die erste Nachricht das folgende Format einhält: <ID des Senders>:<ActiveMQ nutzen>:<Topicname>:<Response erwartet>:<SQL nutzen>:<IP des SQL Servers>

mit den folgenden Datentypen:

String:String:(0/1):(0/1):String

So können alle folgenden Nachrichten unter dem Topicnamen und in einer Datenbank gespeichert werden, zusätzlich wird jeder Client über Neuerungen informiert.

Es ist zudem möglich, mehrere Nachrichten auf einmal an den ActiveMQ zu übertragen. Das Trennzeichen für Nachrichten ist das Semikolon „;“. So kann beispielsweise die folgende Nachricht erfolgreich interpretiert werden.

Bett1:1:2:3;Bett2:3:4:5 Diese Nachricht wird in zwei Teile zerlegt und getrennt gespeichert. So ergibt sich die folgende Ansicht (Tabelle 5.2.2) in MySQL:

DeviceID	Nachricht	Timestamp
Bett1	1:2:3	yyyy-MM-dd HH:mm:ss
Bett2	3:4:5	yyyy-MM-dd HH:mm:ss

Tabelle 5.2: MySQL-Ansicht einer verketteten Nachricht

Ein ähnliches Modul ist bereits im Living Place im Einsatz. Das Modul bietet die Möglichkeit, einen Client mit einer MongoDB und dem im Living Place verwendeten ActiveMQ zu verbinden.

Diese Möglichkeit wurde nicht genutzt, da sie für die Kommunikation mit dem Arduino ungeeignet gewesen wäre und zu viele Änderungen erfordert hätte. Aus diesem Grund wurde eine neue Anwendung entworfen, die die nötigen Funktionen für die Aufgabe liefert und ohne großen Aufwand erweitert werden kann. So ist nur an dieser Anwendung eine Ände-

rung nötig, wenn eine andere Art von Datenbank verwendet werden soll als die verwendete MySQL-Datenbank.

**Netzwerkaufbau und Kommunikation der Komponenten** Um eine erfolgreiche Kommunikation zwischen dem Client und dem Arduino aufzubauen, existieren mehrere Möglichkeiten. Die Schlafphasenerkennungssoftware kann entweder direkt als Server dienen oder hat nur die Aufgabe, als Client zu fungieren. So ist es möglich, dass der Arduino sich direkt mit der Schlafphasenerkennungssoftware verbindet oder aber auch mittels ActiveMQConnector kommuniziert.

Für Letzteres ist die Verwendung einer weiteren Komponente nötig, dem ActiveMQConnector, siehe [Unterabschnitt 5.2.2](#). Der ActiveMQConnector übernimmt die Verteilung aller Nachrichten. So meldet sich jeder Client bei dem ActiveMQConnector an. Der ActiveMQConnector empfängt die Nachrichten vom Arduino und gibt diese an die angemeldeten Clients mittels Remoting weiter. Das Remoting beschreibt eine Möglichkeit, Methoden über das Netzwerk in anderen Anwendungen aufzurufen. Die Möglichkeit des Remoting wird genutzt, um Messwerte in den Client einzutragen. Zusätzlich speichert der ActiveMQConnector alle eingehenden Nachrichten, so es gewünscht ist, in einer MySQL-Datenbank und leitet sie an das ActiveMQ weiter. Diese Option lässt sich, wie zuvor unter [Abbildung 5.2.2](#) beschrieben, mittels Flags beim Verbindungsaufbau steuern. Die Kommunikation findet wie in der Grafik in [Abbildung 5.7](#) statt.

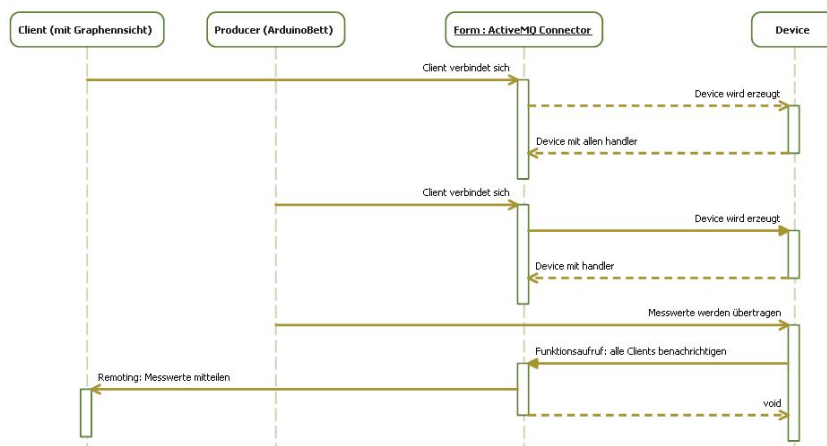


Abbildung 5.7: Kommunikationsaufbau ActiveMQConnector

Der ActiveMQConnector ist wie in der Grafik in [Abbildung 5.8](#) beschrieben aufgebaut. Die Klasse SchlafphasenConnection dient zur Kommunikation mit den Clients. Sie besitzt die

Remotingmethoden und implementiert die für das Marshalling nötigen Funktionen. Die Klassen SimplePublisher und SimpleSubscriber dienen zur Kommunikation mit dem ActiveMQ. SimpleSubscriber enthält eine delegate OnMessageReceived-Methode, die durch eine eigene Methode, die zur Deklaration void METHODENNAME(ITextMessage) passt, überladen werden kann. Diese Methode wird aufgerufen sobald eine neue Nachricht im Topic des ActiveMQ eingetragen wird. Die Klasse SimplePublisher dient zum Veröffentlichen von Nachrichten im ActiveMQ. Diese Klassen benötigen eine Verbindung zum ActiveMQ, diese wird in der Klasse TopicConnectionFactory erstellt. Der Aufbau auf dem Prinzip eines ver-

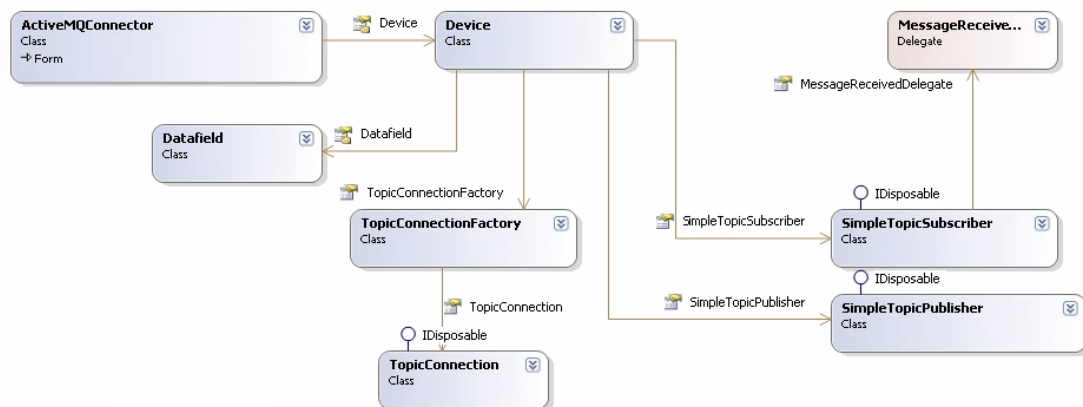


Abbildung 5.8: Klassendiagramm ActiveMQConnector

teilten Systems gewährleistet die Möglichkeit, dass jederzeit ein neuer Client hinzukommen oder verschwinden kann und jeder Client Zugriff zu den identischen Daten hat sowie die Möglichkeit bietet, die Anwendung nur zu starten, wenn sie benötigt wird. Einzig auf dem Server muss der ActiveMQConnector laufen, um die Daten für die Clients zu sammeln.

### 5.2.3 Aufbau des Clients

Die Einstiegsklasse ist die Schlafphasenerkennung.cs (siehe [Abbildung 5.9](#)). Diese beinhaltet die folgende Oberfläche und realisiert sowohl die Steuerung der Oberfläche als auch die Kommunikation. Sie stellt die Funktionen für das Marshalling bereit. Das Marshalling dient zur Nachrichtenmitteilung. Mittels des Marshalling können sich beliebig viele Clients am Server anmelden und werden von diesem mit Messwerten versorgt, ohne zyklisch nach neuen Messwerten zu pollen. Die Klasse Kalender beinhaltet die Funktionen zur Steuerung des Kalenders und nutzt die Funktionen aus der Google API ([Unterabschnitt 5.2.4](#)). Ebenso wird in dieser Klasse der Prototyp eines Weckers ([Abschnitt 5.2.3](#)) implementiert. Das Objekt der Klasse Bett repräsentiert eine Bettseite. In diesem Objekt werden die zu der Bettseite gehörenden DMS verwaltet. Das Objekt der Klasse Bett dient zur Kommunikation mit der Klasse Dehnungsmessstreifen. Alle Messwerte werden in der Klasse Dehnungsmessstreifen verarbeitet. Als Speicherung der Messwerte dient die Klasse Messwert. Alle in der Klasse Dehnungsmessstreifen eintreffenden Messwerte werden zur Anzeige im Graphen an die Klasse Graph weitergeleitet.



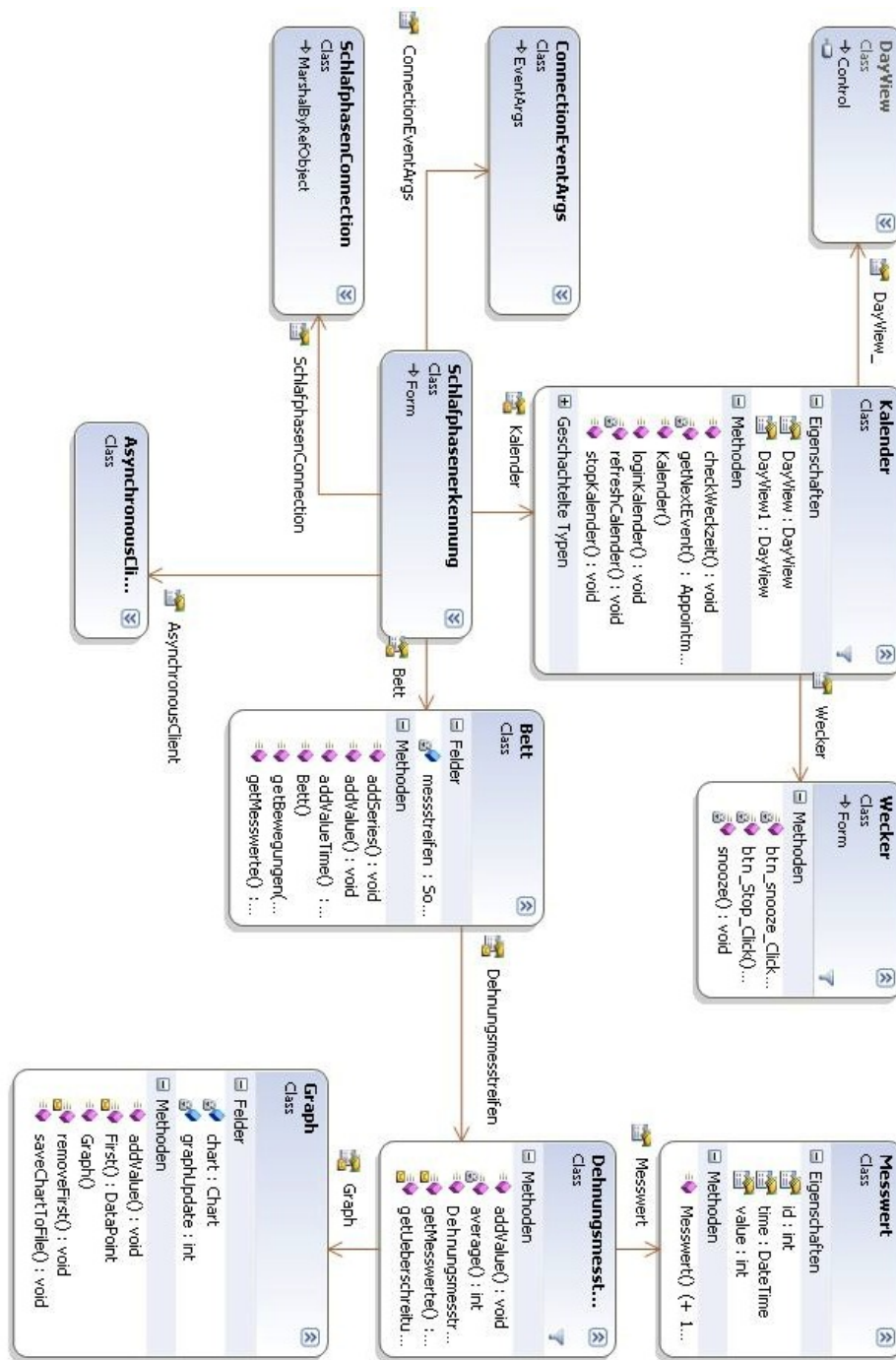


Abbildung 5.9: Klassendiagramm der Schlafphasenerkennungssoftware

**Weckfunktion** Mittels des Google-Kalenders und des Wissens ob eine Person zuhause ist, kann eine Weckfunktion implementiert werden. Die Erkennung, ob eine Person zuhause ist, kann mittels der im Living Place verwendeten Positionstags stattfinden (siehe **UbiSense**). Das sich eine Person in der Wohnung wird im Rahmen dieser Arbeit aber vorausgesetzt. Die Weckfunktion bietet die Möglichkeit, über die Steuerung des Lichtes unter Zuhilfenahme des ActiveMQ eine Lichtszene zu starten, die dem Sonnenaufgang nachempfunden wurde. Zusätzlich ist es möglich, einen Weckton abzuspielen. Hierfür kann jede beliebige wav-Datei verwendet werden.

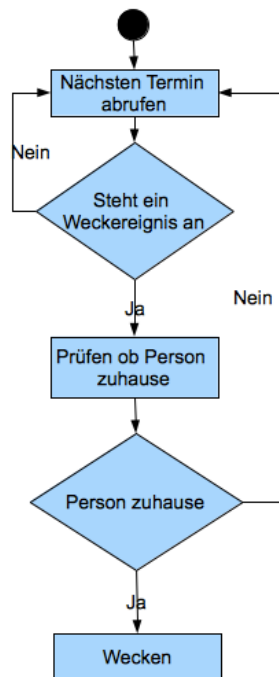


Abbildung 5.10: Erkennung eines Weckereignisses

Ob ein Weckereignis ansteht, wird mittels des folgenden Ablaufs überprüft.

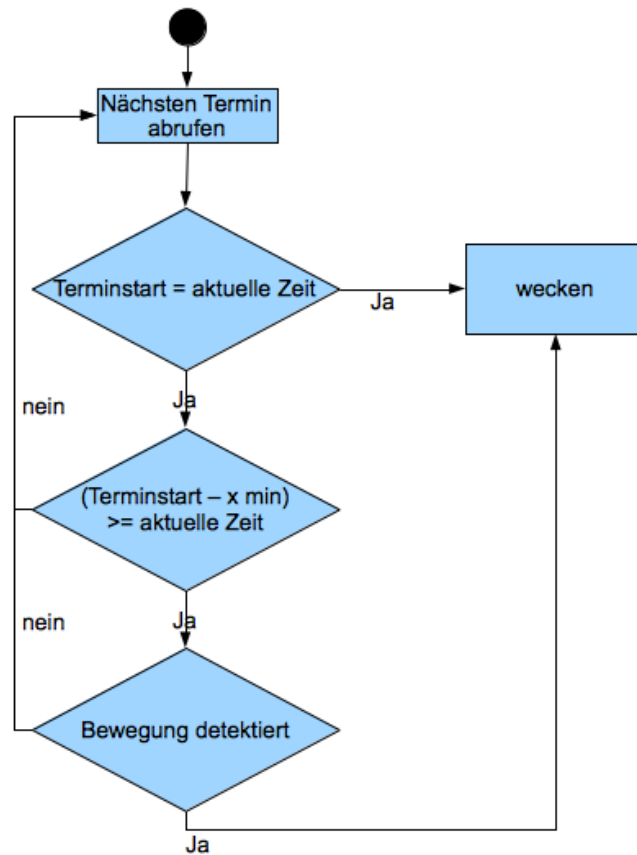


Abbildung 5.11: Ablauf der Weckzeitpunktermittlung

Das Living Place bietet eine Vielzahl von Möglichkeiten, die zum Wecken einer Person genutzt werden können. Eine dieser Möglichkeiten wurde im Rahmen dieser Arbeit zur Entwicklung eines Weckerprotoyps genutzt.

Mittels des verwendeten Kalenders ist es möglich, einem Termin ein Weckereignis zuzuordnen. Wenn die Person sich in einer definierten Zeit vor dem eigentlichen Ereignis bewegt, spätestens jedoch zum eigentlichen Weckereignis, wird das Ereignis Wecken ausgelöst. Dies führt dazu, dass die im Living Pace verbaute Lichtinstallation eine vordefinierte Lichtszene abspielt, in der der natürliche Sonnenaufgang, wie auf dem Bild in [Abbildung 5.1](#) zu sehen, simuliert wird. Anschließend wird eine Form geöffnet und es wird nach einer vordefinierten Zeit ein Weckton abgespielt.

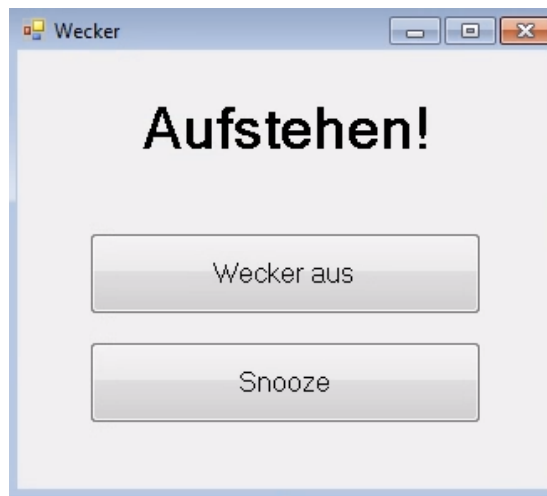


Abbildung 5.12: Ansicht der Weckerform

Die Oberfläche ([Abbildung 5.12](#)) dient zur Steuerung des Weckers. Hier ist eine Schlummerfunktion oder das Abschalten des Weckers realisiert worden. Dieses Zusammenspiel von Ereignissen soll dazu führen, dass das morgendliche Aufstehen erleichtert wird und die zu weckende Person sich entspannter auf den Tag vorbereitet fühlt.

Der hier zu entwickelnde Wecker dient lediglich als Prototyp und umfasst nicht den Funktionsumfang des Wecker 2.0. Ebenso reagiert dieser sofort auf Bewegung und erkennt nicht exakt, in welcher Schlafphase sich die zu weckende Person befindet.

**Aufbau der Benutzeroberfläche des Clients** Die Schlafphasenerkennungssoftware hat im Hauptfenster (Abbildung 5.13) einen Graphen, der die Messwerte der angeschlossenen Latten der letzten 60 Sekunden anzeigt. Die Messwerte, die angezeigt werden, sind die Analogwerte direkt aus dem Arduino und liegen im Wertebereich zwischen 0 und 1023. Mittels der

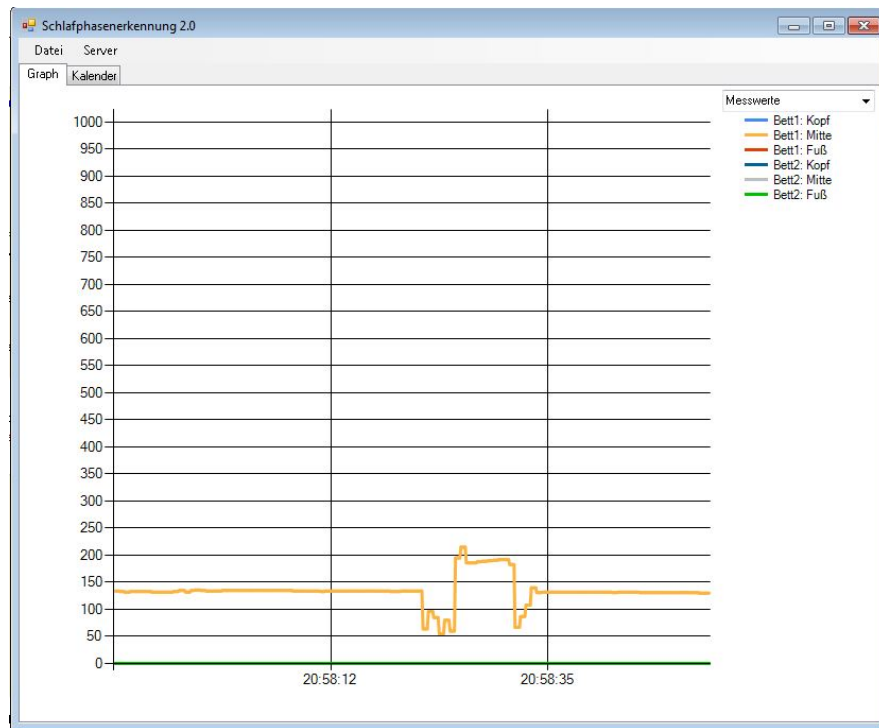


Abbildung 5.13: Hauptansicht der Clientsoftware zur Schlafphasenerkennung

Option „Schlafprofil laden“ beziehungsweise „Schlafprofil speichern“ lassen sich die Schlafprofildaten in eine XML-Datei exportieren und es wird zusätzlich ein Bild des kompletten Graphen gespeichert. Die so gespeicherten XML-Dateien können ebenso wieder importiert werden, um das Schlafprofil zu einem späteren Zeitpunkt auswerten zu können und gegebenenfalls Anomalien aufzuspüren.

Die Kalenderansicht ([Abbildung 5.14](#)) zeigt den Kalender der nächsten sieben Tage. Im unteren Bereich des Fenster ist eine Einlog-Maske zu finden. Nach dem erfolgreichen Einloggen wird im Abstand von zehn Minuten über die Google API ([Unterabschnitt 5.2.4](#)) geprüft, ob neue Termine vorhanden sind, und diese werden im Kalender angezeigt.

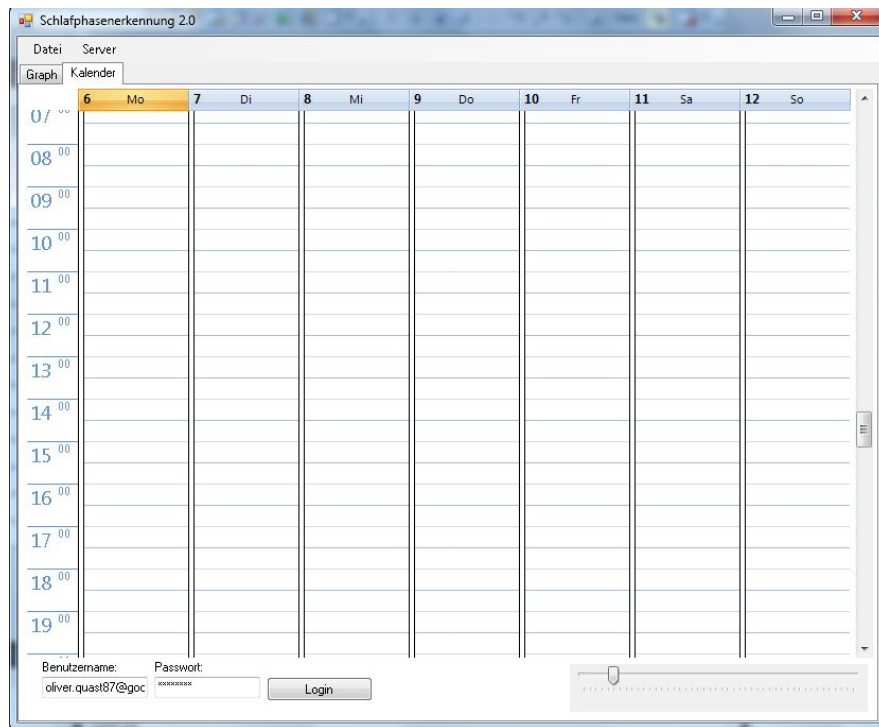


Abbildung 5.14: Kalenderansicht der Clientsoftware

In der Graphenansicht ([Abbildung 5.15](#)) sind alle Messwerte der letzten 24 Stunden aufgetragen. So kann ein genaues Schlafprofil erstellt werden. In der Option, die im unteren Bereich des Graphens eingestellt werden kann, kann zusätzlich auf einen Graphen gewechselt werden, der die Anzahl der Bewegungen beziehungsweise die Stärke der Bewegungen zeigt.

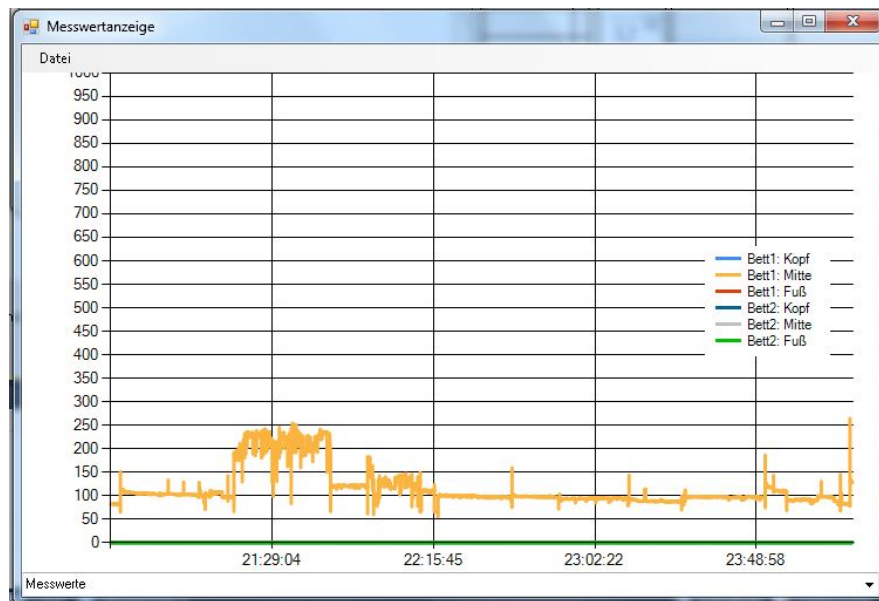


Abbildung 5.15: Komplette Graphenansicht der Clientsoftware zur Schlafphasenerkennung

In dieser (Abbildung 5.16) Ansicht wird die Stärke der Bewegung aufgezeigt. Die Stärke der Bewegung wird als Abweichung zum errechneten Mittelwert aufgetragen.

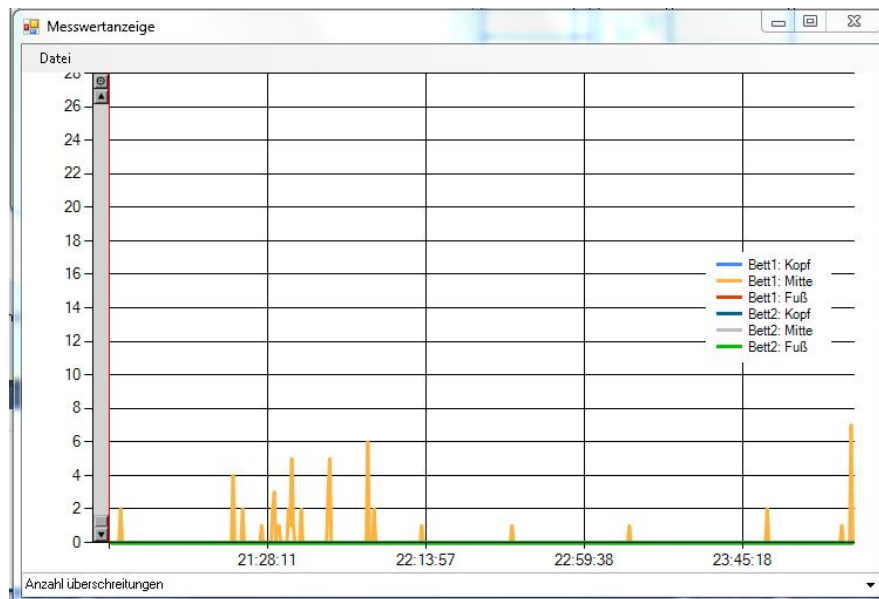


Abbildung 5.16: Komplettes Bewegungsprofil der Clientsoftware zur Schlafphasenerkennung

In jeder Graphenansicht ist es möglich, durch Klicken und Ziehen in horizontaler und vertikaler Richtung in den erstellten Graphen hereinzuzoomen. Dies bietet die Möglichkeit der Analyse kleinerer Bereiche des Graphen unter Ausschluss der übrigen Messwerte. Beim hereinzuzoomen werden die Daten nicht interpoliert, so kommt es nicht zu Verfälschungen der Messwerte.

Beim Design der Klassen und der Anzeige der Daten ist strikt das Prinzip des Modell-View-Control eingehalten worden. Der Aufbau wird in [Abbildung 5.17](#) gezeigt.



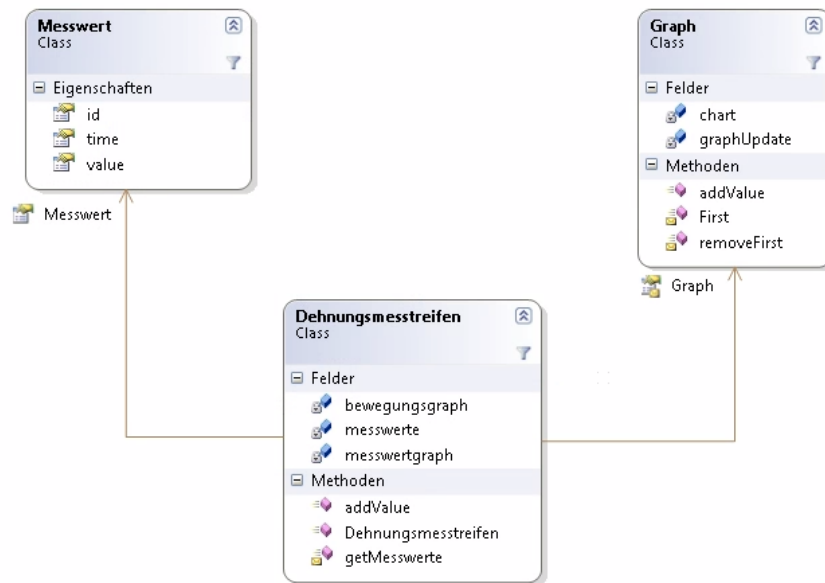


Abbildung 5.17: Modell-View-Control

In der Klasse Dehnungsmesstreifen werden die Messdaten mittels Objekte der Klasse Messwert gesammelt und beim Eintreffen neuer Daten wird die Funktion AddValueToChart, der Graph-Klasse, aus der Klasse Dehnungsmesstreifen aufgerufen, um die neuen Daten zu visualisieren.

## 5.2.4 Google API

Zur Anbindung des Google-Kalenders wurde die hauseigene API verwendet, um an Termine aus dem Kalender zu gelangen und hieraus gegebenenfalls einen Weckzeitpunkt zu bestimmen, falls dieser Termin ein weckzeitrelevanter Eintrag ist. Der Google-Kalender wurde verwendet, da dieser frei von jedem nutzbar ist und die Möglichkeit bietet, von einer unbegrenzten Anzahl von Geräten bedient zu werden. So ist es möglich, sowohl mit dem Touch-Tresen des Living Places Termine in den Kalender einzutragen als auch Termine mittels eines Smartphones oder eines Laptops, zu verwalten und somit indirekt das Bett zu steuern. Als mögliche Alternative wurde beispielsweise ein Exchange-Kalender betrachtet, dieser ist nicht durch frei verfügbare Software steuerbar und steht nicht jedem frei zur Verfügung. Als weitere Option gäbe es eine lokale Lösung, diese bietet jedoch nicht den Komfort, Termine von mehreren Geräten synchron zu halten und Termine von überall anzupassen und zu erstellen.

**Benötigte Methoden** Zur Verwendung des Google-Kalenders ist es wichtig, einen Service anzumelden, der auf den Google-Kalender zugreifen soll.

Dies erfolgt über die folgenden Befehlszeilen:

```
1 CalendarService myService = new CalendarService("ServiceName");
2 myService.setUserCredentials(user, password);
```

Listing 5.11: Service anmelden für Google-Kalender

Hierbei kann jeder beliebige Name für den Service verwendet werden. <user> und <password> müssen zu einem existierenden Google-Account gehören und der User muss sich aus der Google-Mail-Adresse zusammensetzen.

Jetzt kann auf den Google-Account zugegriffen werden. Der Zugriff und die Steuerung erfolgen mittels Query.

Im Folgenden wird eine Query für den Abruf des Kalenders und aller Kalenderdaten der nächsten sieben Tage ab dem aktuellen Zeitpunkt in aufsteigender Reihenfolge erzeugt.

```
1 EventQuery myQuery =
2     new EventQuery("http://www.google.com/calendar/feeds/"
3         + user + "/private/full");
4 myQuery.StartTime = System.DateTime.Now.Date;
5 myQuery.EndTime = System.DateTime.Now.AddDays(7.0);
6 myQuery.ExtraParameters = "orderby=starttime&sortorder=ascending";
```

Listing 5.12: Query-Erstellung für Google-Kalender

Die so erstellte Query muss nun an Google übertragen werden und liefert als Ergebnis ein EventFeed. Die Übertragung geschieht folgendermaßen:

```
1 EventFeed myResultsFeed = myService.Query(myQuery);
```

Listing 5.13: Query-Übertragung zu Google

Über das so erhaltene Ergebnis können nun mittels des Punktoperators Entries iteriert werden, um alle Ergebnisse auszugeben.

```
1 foreach (EventEntry singleEvent in myResultsFeed.Entries)
2 {
3     //singleEvent enth"alt die Daten von einem Ereignis in dem Kalender
4 }
```

Listing 5.14: Über EventFeed iterieren

**Version** Die Version, die in dieser Arbeit benutzt wurde ist die Laufzeitversion 2.0.50727. Dieses zum Entwicklungszeitpunkt (15. Mai 2012) die aktuelle Version ([Mark Stahl](#)).

### 5.2.5 Day-View-Kalender

Der Day-View-Kalender ist eine grafische Unterstützung, die dem Benutzer die Termine der nächsten Woche ab dem aktuellen Datum anzeigt. Sie bietet das gewohnte „look&feel“, das der Benutzer aus Outlook gewöhnt ist. Es wird ausschließlich zur Visualisierung von Terminen genutzt und bietet keine Möglichkeit zur Bearbeitung.

**Benötigte Methoden** Der Kalender ist als Komponente verfügbar und kann über einfaches Klicken zur Form hinzugefügt werden.

Um eine Übersicht der nächsten sieben Tage darzustellen, wird die Startzeit auf die aktuelle Zeit gesetzt und der Punktoperator `DaysToShow` auf sieben gesetzt.

Zusätzlich müssen noch zwei Methoden überladen, die aufgerufen werden, wenn ein neues Ereignis eingetragen wird. Der fertige Quellcodeabschnitt sieht folgendermaßen aus:

```
1 dayView1.StartDate = DateTime.Now;
2 dayView1.NewAppointment +=
3     new NewAppointmentEventHandler(dayView1_NewAppointment);
4 dayView1.ResolveAppointments += new Calendar.
5     ResolveAppointmentsEventHandler(
6         this.dayView1_ResolveAppointments);
7 dayView1.DaysToShow = 7;
```

Listing 5.15: Initialisieren des DayView-Kalenders

Die beiden Methoden, die zur Überladung dienen, sehen folgendermaßen aus:

```
1 private void dayView1_NewAppointment(object sender,
2     NewAppointmentEventArgs args)
3 {
4     Appointment m_Appointment = new Appointment();
5
6     m_Appointment.StartDate = args.StartDate;
7     m_Appointment.EndDate = args.EndDate;
8     m_Appointment.Title = args.Title;
9
10    m_Appointments.Add(m_Appointment);
11 }
12
13 private void dayView1_ResolveAppointments(object sender,
14     ResolveAppointmentsEventArgs args)
15 {
```

```
16     List<Appointment> m_Apps = new List<Appointment>();
17
18     foreach (Appointment m_App in m_Appointments)
19         if ((m_App.StartDate >= args.StartDate) &&
20             (m_App.StartDate <= args.EndDate))
21             m_Apps.Add(m_App);
22
23     args.Appointments = m_Apps;
24 }
```

Listing 5.16: Über EventFeed iterieren

Mittels dieser Methoden ist es nun möglich, neue Einträge im DayView-Kalender vorzunehmen.

Kombiniert man nun die Methoden mit den Google-Funktionen, erhält man eine Kalenderansicht wie die in [Abbildung 5.14](#) gezeigte Ansicht.

**Version** Die hier verwendete Version ist die Version 1.0, die seit 2011 die aktuellste Version darstellt. Die Version ist unter der CPOL-Lizenzierung im Internet verfügbar ([Tike](#)).

### 5.3 Schlafanalyse

Bei dem hier gezeigten Ansatz handelt es sich um einen Prototypen, um die Funktion zu verdeutlichen.

Das folgende Schlafprofil in [Abbildung 5.18](#) zeigt den Schlaf einer Nacht.

Das Schlafprofil wurde mit der entwickelten Anwendung unter Verwendung der entwickelten Hardware im privaten Bereich getestet. Start des Schlafprofils liegt bei ca. 0 Uhr, in dieser Zeit liegt auch die geschätzte Einschlafzeit. Um eine möglichst lange Aufzeichnung zu bekommen, liegt die Aufwachzeit erst bei 10 Uhr. So beträgt die Schlafdauer ca. zehn Stunden. Die senkrechten, großen Striche zeigen starke, ruckartige Bewegungen. Dazwischen sind deutlich zu erkennen lange Zeiten ohne starke Bewegungen. Diese Zeit wird wie in [Abbildung 2.4](#) definiert als Schlafphase bezeichnet. Um dies tatsächlich abschließend als Schlafphase bezeichnen zu können, sind weitere Untersuchungen nötig. Je höher der angezeigte Messwert ist, desto mehr ist die Latte, auf der der Dehnungsmessstreifen verklebt ist, belastet. Somit wird die Latte durch ein größeres Gewicht mittig belastet.

## 5 Realisierung und Schlafanalyse

---

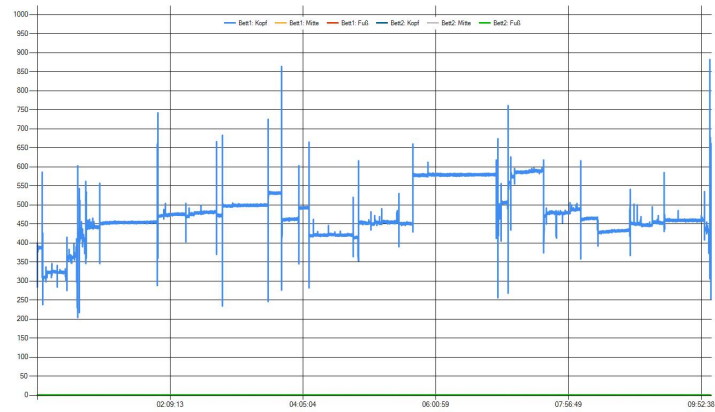


Abbildung 5.18: Komplettes Schlafprofil einer Nacht

In der Grafik (Abbildung 5.19) ist die komplette Nacht als Bewegungsstärkeprofil gezeigt. Hier ist die Erkennung der Phasen, die als Wachphase definiert wurden, besser zu erkennen.

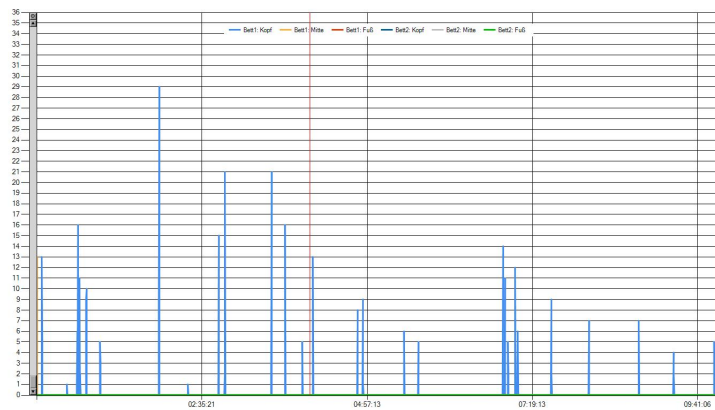


Abbildung 5.19: Stärke der Bewegungen des Schlafenden einer Nacht

## 5 Realisierung und Schlafanalyse

So ist wie in der Grafik ([Abbildung 5.20](#)) in der Zeit um 7 Uhr eine deutliche Häufung an Bewegungen zu erkennen. Diese Zeit wird präferiert, um erleichternd aufstehen zu können, da hier eine Wachphase vermutet wird.

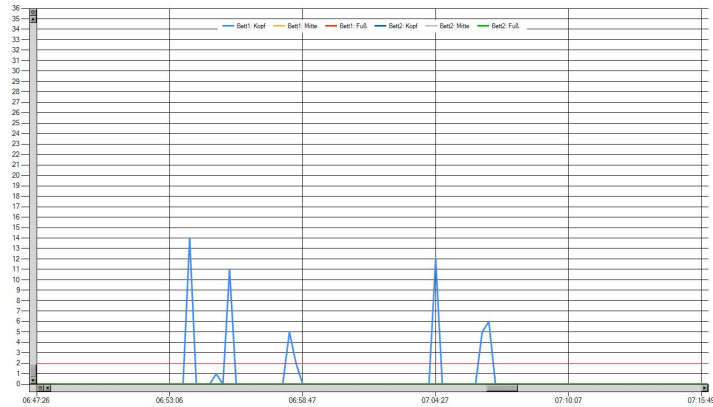


Abbildung 5.20: Stärke der Bewegungen in Wachphase morgens als Ausschnitt

Je höher der Wert in dieser Grafik ist, desto stärker war die Bewegung und umso mehr ist der Wert vom Durchschnittswert abgewichen. Derselbe Zeitraum ist nochmals in der Messwertdarstellung in der Grafik in [Abbildung 5.21](#) gezeigt.

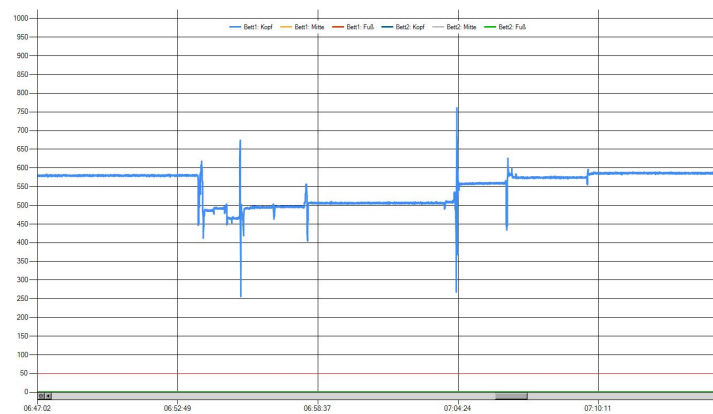


Abbildung 5.21: Bewegungsprofilausschnitt, in Wachphase morgens

Die Messung bietet die Möglichkeit, durch die Matratze die Atmung einer Person zu erkennen. So besteht die Möglichkeit, Atemaussetzer, wie in **Abbildung 5.22** gezeigt, zu erkennen.

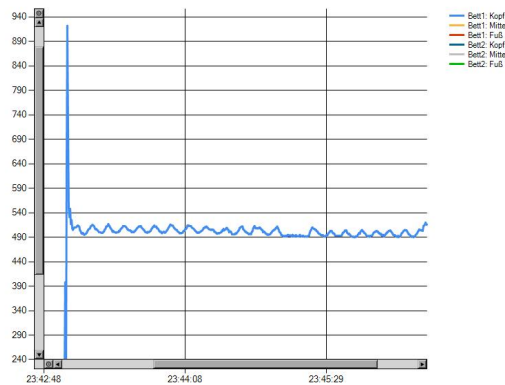


Abbildung 5.22: Atmung mit Aussetzer

## 5.4 Auswertung und Probleme

Während der Ausarbeitung sind Fehler aufgetreten, die in diesem Abschnitt beschrieben und zusammen mit den gefundenen Lösungen aufgezeigt werden, die beim Herausarbeiten entwickelt wurden.

### 5.4.1 Störung durch Funk

Im Living Place sind auf Grund verschiedener Funksender starke Einstrahlungen festzustellen. Die Dehnungsmessstreifen sind offen auf den Latten aufgebracht und die Verbindung zur Verstärkerschaltung und zur Wheatstone'schen Brückenschaltung sind ungeschirmt. So bietet die Installation viele Möglichkeiten, durch Strahlung Störungen zu beeinflussen.

### Problemlösung

Um Störungen auf dem Weg zur Brückenschaltung zu beseitigen, wurde die Wheatstone'sche Brückenschaltung direkt neben den Dehnungsmessstreifen auf einer kleinen Platine an den Latten befestigt. Die Störungen, die auf der Strecke bis zur Verstärkerschaltung entstehen, sind durch ein doppelt geschirmtes Aktorenkabel abgeschwächt und auf ein herausfilterbares Maß reduziert worden, sodass eine Mittelwertberechnung aus wenigen Werten ausreicht, um ein

sauberes Signal zu erzeugen.

#### **5.4.2 Mechanischer Verschleiß**

Auf Grund der mechanischen Beanspruchung, die durch das deformieren der Latten des Lattenrostes entstehen, kam es zu Brüchen an der Verbindungsstelle zwischen Dehnungsmessstreifen und dem Kabel, das zur Wheatstone'schen Brückenschaltung führt. Da diese Stelle verlötet werden muss, verformt sich der Lötspunkt nicht stark genug und führt zum Abreißen der Lötkontakte und des Lötspads vom Dehnungsmessstreifen.

#### **Problemlösung**

Durch das aufbringen von selbstklebenden Stützpunkten konnte diesem Problem entgegen gewirkt werden. So ist die mechanische Beanspruchung nicht direkt auf dem Dehnungsmessstreifen. Zudem wurde der Anschluss über einen sehr dünnen Draht realisiert, der sich unter dem zur Befestigung angebrachten Klebeband leicht verschieben lässt. Dies führt in Verbindung mit einer gelegten Schlaufe zu einem dynamischen Ausgleich der Lattendeformation. Ein ähnliches Prinzip wird von Spinnen beim Bau von Spinnennetzen verwendet, um diese besonders strapazierfähig zu machen.

#### **5.4.3 Nicht abgegichene Brückenschaltung**

Auf Grund der Formgebung und der Deformation der Latten wurden die Dehnungsmessstreifen bereits im Ruhezustand gedehnt, was dazu führte, dass die Dehnungsmessstreifen nicht abgeglichen waren.

#### **Problemlösung**

Als Problemlösung wurde die zweistufige Brückenschaltung eingeführt, die nach der ersten Verstärkung eine Offsetschaltung beinhaltet, um den zuvor erwähnten Fehler auszubessern.

#### **5.4.4 Probleme beim Abgleich der Brückenschaltung**

Der Aufbau führte dazu, dass die Schaltung nicht erfolgreich abgeglichen werden konnte. Die Entfernung vom Dehnungsmessstreifen zur Brückenschaltung war zu groß und die Kabellänge führte zum fehlerhaften Abgleich der Brückenschaltung.



### **Problemlösung**

Die Messbrücke wurde direkt neben die Dehnungsmessstreifen platziert und die Kabel zu dem Dehnungsmessstreifen auf der Ober- und Unterseite wurden exakt aufeinander abgestimmt.

### **5.4.5 Overhead an Daten**

Da die Clientsoftware einen Graphen beinhaltet, der den Verlauf der letzten 24 Stunden beinhaltet, werden die Messwerte aus diesem Zeitraum benötigt. Dies sind bei ca. einer Messung pro Sekunde pro Bettseite ca. 173000 Messungen. Dies würde im ActiveMQ des Living Places zu einem zu hohen Datenaufkommen bei einem Dauerbetrieb führen.

### **Problemlösung**

Dieses Problem ist wie hier beschrieben gelöst ([Abschnitt 5.2.2](#)). Dieses Problem wurde durch eine Datenbank gelöst, die in regelmäßigen Abständen bereinigt wird. Alte Nachrichten, die sich länger als 24 Stunden in der Datenbank befinden, werden hierbei gelöscht.

So bleiben die Messungen für jeden Client erhalten, ohne zu einem Overhead an Daten zu führen, die von keiner anderen Anwendung im Living Place genutzt werden.

Nachdem alle Probleme, die bei der Erstellung dieser Arbeit aufgetreten sind, erörtert wurden, wird jetzt aufgezeigt, wie sich die Arbeit in Zukunft entwickeln könnte.

# 6 Fazit

## Zusammenfassung der Arbeit

In dieser Arbeit wurde die Möglichkeit zur Analyse von Bewegungen im Bett realisiert. Hierfür wurden DMS auf einem Bett im Living Place appliziert. Die Signale dieser DMS wurden durch eine Wheatstone'sche Brückenschaltung messbar gemacht und mittels einer entworfenen Verstärkerschaltung verstärkt. Die so verstärkten Signale wurden in einen Arduino eingespeist, damit der Arduino die Signale digitalisieren und an einen Server übermitteln konnte.

Es wurde eine Clientanwendung entwickelt, die die Messwerte aufbereitet und dem Benutzer grafisch darstellt. Diese Clientanwendung besitzt zudem die Anbindung an einen Google Kalender. Mittels dieses Kalenders und den zur Verfügung stehenden Messwerten wurde der Prototyp eines Weckers entworfen.

## Analyse der Arbeit und Vergleich mit den Anforderungen

Die **Anforderungen an die Grundfunktion** wurden im Laufe dieser Arbeit gewährleistet. Über den Aufbau mit Dehnungsmessstreifen wurden die **Anforderungen an die Bedienungs-freundlichkeit** erfüllt. Der Anwender muss keine speziellen Vorkehrungen treffen, um dieses Bett nutzen zu können.

Mit der Verwendung eines Ethernet-Shields auf dem Arduino konnte die Anforderung unter **Anforderungen an die Strahlungsreduzierung** erfüllt werden. Die Verwendung des Ethernet-Shields verursacht keine zusätzlich Strahlung und kann somit bedenkenlos eingesetzt werden. Durch die Verwendung eines verteilten Systems ist die Anforderung der Ressourcenschonung im Abschnitt **Anforderungen an die Ressourcenschonung** eingehalten worden. Hierdurch werden nur Ressourcen verwendet, die auch benötigt werden.

Da die DMS in dem verwendeten Aufbau thermische Einflüsse eliminieren, ist die Anforderung unter **Anforderungen an die Umwelteinflüsse** erfüllt. Ebenso ist die Schlafhygiene nicht durch den Aufbau gefährdet, da der Arduino keine Geräusche verursacht und durch die Verwendung von Dehnungsmessstreifen keine Strahlung verursacht wird. Somit ist auch die Anforderung unter **Anforderungen an die Schlafhygiene** erfüllt.

Während der Bearbeitung eröffnete sich ein breites Spektrum an Erweiterungen und Anpassungen, die das hier vorgestellte Bett zu einer innovativen Lösung machen können. Unter der Annahme, dass die zu testende Person sich passend zu den Schlafphasen bewegt, ist dieser Ansatz vollkommen ausreichend, um eine Unterscheidung zwischen Wach- und Schlafphase zu treffen. Dies erfordert weitere Untersuchungen, um einen generellen Erfolg zu beweisen.

### **Ausarbeitung**

Während der Ausarbeitung wurde herausgestellt, dass eine genaue Unterteilung in Schlafphasen und die Schlaftiefe durch einen berührungslosen Aufbau nicht realisierbar sind. Einzig die Bewegungen können ermittelt werden und so eine Annahme getroffen werden, dass sich die zu weckende Person in einer geringen Schlaftiefe befindet, wenn sich die zu überwachende Person bewegt. Ob dieses bei allen Personen zutrifft, unterliegt weiteren Untersuchungen.

### **Bewegungserkennung**

Die Detektion funktioniert präzise und erkennt zusätzlich zu den Bewegungsmustern des Körpers auch kleinere Bewegungen wie zum Beispiel die Atmung. Dies könnte wie im Ausblick beschrieben zu weiteren medizinischen Untersuchungen genutzt werden.

### **Schlafphasenerkennung**

Zu diesem Zeitpunkt ist keine genaue Unterteilung der Schlafphasen möglich, da diese nur präzise durch einen Laboraufbau wie im Schlaflabor festzustellen sind. Hierfür werden sowohl die Muskelbewegungen des Gesichtes als auch die Hirnströme und deren Veränderungen benötigt. Dies ist für den alltäglichen Gebrauch nicht praktikabel. Daher ist das hier erzielte Ergebnis unter Anbetracht der gestellten Anforderungen ein zufriedenstellendes Ergebnis.

## 7 Ausblick

Im Laufe weiterer Arbeiten können die durch den Aufbau erhobenen Daten analysiert werden. Durch die Unterstützung medizinisch geschulten Personals können nach Untersuchungen die Schlafphasen erkannt und in weiteren Projekten zum Wecker 2.0 ausgebaut werden.

Als mögliche Verbesserung wird die zusätzliche Überwachung der Herzfrequenz mittels Brustgurt gesehen. Hier wird Abstand von der berührungslosen Schlafphasenerkennung genommen, aber da dieser kaum spürbar getragen werden kann und mittels Bluetooth, also kabellos, seine Messwerte übermittelt, wäre dies eine denkbare Weiterentwicklung, um ein präziseres Schlafbild zu erhalten.

### **Vorteile dieses Ansatzes:**

Auf Grund der Veränderung wäre vermutlich eine frühzeitige Erkennung der Schlafphase möglich und könnte so besser der Interaktion mit dem Licht dienen, sodass das Licht bereits frühzeitig langsam hochgedimmt werden kann, um den Aufwachprozess zum richtigen Zeitpunkt besser zu unterstützen. Ob und inwieweit dies tatsächlich ein besseres Ergebnis liefern kann, bleibt unter Vorbehalt weiterer Untersuchungen.

Eine weitere Verbesserung, die mit dem zuvor genannten Schritt einhergehen könnte, wäre die zusätzliche Überwachung der Körpertemperatur. Dies würde den obigen Ansatz weiter unterstützen, da die Körpertemperatur ebenso der Schlafphase gleichzusetzen ist wie die Bewegungen. Jedoch kann es vorkommen, dass die zu überwachende Person sich in einer Wachphase befindet, ohne sich zu bewegen. Die Körpertemperatur ist hingegen Schwankungen unterlegen und könnte der Schlafphase zugeschrieben werden (vgl. [Leifeld](#)).

Die vorhandene Software könnte zudem so angepasst werden, dass sie Bewegungsmuster wie die Atmung der zu testenden Person ermittelt, um gegebenenfalls Atemaussetzer (Schlafapnoe) zu erkennen, um eine Warnung auszusprechen. Ein Beispiel für die Atmung ist in [Abbildung 5.22](#) zu sehen.

Als mögliche Softwareerweiterung könnte eine Datenbank, in der Orte und deren Entfernung und zeitliche Dauer zum Erreichen des Ziels vermerkt sind, sein. Hierüber könnte anhand des Termins, dessen Ort und anhand von Personenprofilen, ein idealer Weckzeitpunkt bestimmt werden. So kann zum Beispiel eine Person, die eine sehr kurze morgendliche Routine hat, automatisch später geweckt werden. Diese morgendliche Routine kann zum Beispiel anhand von Bewegungsprofilen im Living Place durch die Positionsbestimmung realisiert werden, sodass ein Grundprofil mit einer Varianz zu bestimmten Ereignissen zu einem Termin zugeordnet wird und so der perfekte Morgen realisiert wird.

Eine weitere Softwareerweiterung betrifft die Kommunikation. Hier wäre es von Vorteil, die Kommunikation vollständig mittels JSON-Nachrichten zu bewerkstelligen, um im Living Place mit anderen Anwendungen kompatibel zu bleiben und zudem die Kommunikation zu erleichtern und weniger rechenintensiv zu gestalten. Dies ist bis jetzt noch nicht implementiert, da keine zuverlässig funktionierende JSON-Library für den Arduino existiert. Dies wäre im Rahmen einer weiteren Arbeit weiter ausbaubar.

Als weitere Option kann zudem eine VLC-Steuerung integriert werden, die zusätzlich zum Weckton morgens das Fernsehbild einschaltet. Ein ähnliches Prinzip wird vom UbiSense-Projekt im Living Place verwendet.

Ein weiterer Schritt zur Integration wäre das Ausstellen des Wecktons mittels des Cubicals, der im Living Place entwickelt wurde und dort zum Einsatz kommt.

# Literaturverzeichnis

- [sleepsmart ] : *SleepSmart*. – URL <http://www.sleepsmart.com.au/index.php>. – Zugriffsdatum: 2011-12-05
- [afp ] AFP: *Gerät wählt passende Schlafphase für Alar*. – URL <http://www.rp-online.de/wissen/forschung/neuer-wecker-fuer-morgenmuffel-1.1604892>. – Zugriffsdatum: 2012-07-31
- [werner eis company austria ] AUSTRIA werner eis company: *axbo*. – URL [http://www.axbo.com/axbo/cms/leichter\\_aufwachen.aspx](http://www.axbo.com/axbo/cms/leichter_aufwachen.aspx). – Zugriffsdatum: 2012-07-31
- [Baumgart ] BAUMGART, Silvia: *Der gesunde Schlaf*. – URL <http://www.medizininfo.de/kopfundseele/schlafen/schgesund.htm>. – Zugriffsdatum: 2012-08-25
- [Busch und Busch 2011] BUSCH, Rudolf ; BUSCH, Rudolf: Das elektrische Feld. In: *Elektrotechnik und Elektronik*. Vieweg+Teubner, 2011, S. 2–25. – URL [http://dx.doi.org/10.1007/978-3-8348-8170-0\\_1](http://dx.doi.org/10.1007/978-3-8348-8170-0_1). – ISBN 978-3-8348-8170-0
- [Company ] COMPANY, Arduino: *Ethernet-Shield*. – URL <http://arduino.cc/en/Reference/ClientConstructor>. – Zugriffsdatum: 2012-07-31
- [disynet Data Interface Systems und GmbH ] DATA INTERFACE SYSTEMS disynet ; GMBH, Networks: *DMS, Dehnungsmessstreifen*. – URL <http://www.esensors.net/DMS.png>. – Zugriffsdatum: 2012-07-31
- [Dvorak 2012] DVORAK, Dipl.-Ing. W.: *Wheatstonesche Messbrücke*. 7 2012. – URL [http://de.wikipedia.org/wiki/Wheatstonesche\\_Messbr%C3%BCcke](http://de.wikipedia.org/wiki/Wheatstonesche_Messbr%C3%BCcke). – Zugriffsdatum: 18.7.2012
- [Elektronik-Kompendium ] ELEKTRONIK-KOMPENDIUM: *Differenzverstärker (Transistor)*. – URL <http://www.elektronik-kompendium.de/sites/slt/0209091.htm>. – Zugriffsdatum: 2012-07-31
- [Ellenberg ] ELLENBERG, Jens: *Wecker 2.0*. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/Ellenberg/bericht.pdf>. – Zugriffsdatum: 2012-08-30

- [Fachverband ] FACHVERBAND, Wasserbetten: *Auflagedruckmessung gibt Klarheit über Liegekomfort.* – URL [http://www.fachverband-wasserbett.de/pdf/presse/aufgedruckmessung\\_gibt\\_klarheit\\_ueber\\_liegekomfort.pdf](http://www.fachverband-wasserbett.de/pdf/presse/aufgedruckmessung_gibt_klarheit_ueber_liegekomfort.pdf). – Zugriffsdatum: 2012-08-30
- [Fischer 2012] FISCHER, Georg: *Die Schlaf- und Traumphasen.* 07 2012. – URL <http://www.schlafen-cocoon.de/bausteine-des-erholsamen-schlafes-wie-bettklima-schlaflage-usw/schlafphasen>. – Zugriffsdatum: 2012-07-20
- [Hardenack ] HARDENACK, Frank: *Das intelligente Bett - Sensorbasierte Detektion von Schlafphasen.* – URL <http://users.informatik.haw-hamburg.de/~ubicomp/arbeiten/master/hardenack.pdf>. – Zugriffsdatum: 2012-07-31
- [JEPICO ] JEPICO, JEPICO C.: *Air mattress type bio-signal and sleep monitoring system.* – URL [http://www2.nict.go.jp/wireless/spacelab/universalca\\_ets8/SJR/English/e-45/capitalproducte.pdf](http://www2.nict.go.jp/wireless/spacelab/universalca_ets8/SJR/English/e-45/capitalproducte.pdf). – Zugriffsdatum: 2012-08-30
- [Klaus Tkotz 2005] KLAUS TKOTZ, Günter Springer Ulrich W.: *Formeln für Elektrotechniker.* Bd. 12. Europa Lehrmittel Verlag, 2005. – ISBN 978-3808533628
- [Klettke ] KLETTKE, Sascha: *Sleep Cycle: Per Zufall geweckt.* – URL <http://www.spiegel.de/video/sleep-cycle-per-zufall-geweckt-video-1044062.html>. – Zugriffsdatum: 2012-07-31
- [Kuonen ] KUONEN, Jan: *Was sind Alphawellen?.* – URL <http://www.lerntipp.com/alphawellen/>. – Zugriffsdatum: 2012-08-25
- [Leifeld ] LEIFELD, Dr. U.: *Der Rhythmus, bei dem man mit muss....* – URL <http://www.schlafauskunft.de/Schlafauskunft/Bio+Rhythmus/>. – Zugriffsdatum: 2012-07-31
- [Mark Stahl ] MARK STAHL, Anirudh D.: *google-gdata.* – URL <http://code.google.com/p/google-gdata/>. – Zugriffsdatum: 2012-05-15
- [MattressLab ] MATTRESSLAB, Mattress L.: *The Mattress Lab Process.* – URL <http://www.mattresslab.com/our-process.html>. – Zugriffsdatum: 2012-08-30
- [Messtechnik ] MESSTECHNIK, Preusser: *Dehnungsmeßstreifen - technische Begriffe.* – URL <http://www.preusser-messtechnik.de/dms/dms/terms.htm>. – Zugriffsdatum: 2012-07-31

- [Neuendorf ] NEUENDORF, Andreas: *DMS Verstärker*. – URL [http://www.telemotorix.de/html/dms\\_verstarker.html](http://www.telemotorix.de/html/dms_verstarker.html). – Zugriffsdatum: 2012-07-20
- [Nowotny ] NOWOTNY, Marcus: *Arduino JSON Library*. – URL <http://interactive-matter.eu/how-to/ajson-arduino-json-library/>. – Zugriffsdatum: 2012-08-30
- [Penzel a] PENZEL, Prof. T.: *Die Untersuchung im Schlaflabor*. – URL <http://www.charite.de/dgsm/rat/labor.html>. – Zugriffsdatum: 2012-08-30
- [Penzel b] PENZEL, Prof. T.: *Verhaltensregeln zur Förderung eines gesunden Schlafs*. – URL <http://www.charite.de/dgsm/rat/hygiene.html>. – Zugriffsdatum: 2012-07-31
- [Prochanska 2012] PROCHANSKA, Prof. E.: *Halbbrücken Simulation in Flash*. 07 2012. – URL [http://www.bipede.de/Downloads/DMS\\_Halb.swf](http://www.bipede.de/Downloads/DMS_Halb.swf). – Zugriffsdatum: 2012-07-20
- [SleepCycle ] SLEEP CYCLE: *SleepCycle*. – URL <http://www.sleepcycle.com/>. – Zugriffsdatum: 2012-07-31
- [TI ] TI, Texas I.: *Datasheet PGA204/205*. – URL <http://www.ti.com/lit/ds/symlink/pga205.pdf>. – Zugriffsdatum: 2012-07-31
- [Tike ] TIKE, Ertan: *Calendar DayView Control*. – URL <http://www.codeproject.com/Articles/12201/Calendar-DayView-Control>. – Zugriffsdatum: 2012-05-15
- [UbiSense ] UBISENSE: *UbiSense*. – URL <http://www.ubisense.net/en/>. – Zugriffsdatum: 2012-08-30
- [Walter ] WALTER, Regina: *rückengerecht angepasste Betten durch Computer-Liegetest*. – URL <http://www.rueckencenter.ch/shop/info/Computer-Liegetest.html>. – Zugriffsdatum: 2012-07-31
- [Watterott ] WATTEROTT, Andreas: *RedFly-Shield*. – URL [https://github.com/watterott/RedFly-Shield/blob/master/src/libraries/RedFly/examples/WebClient\\_Socket/WebClient\\_Socket.ino](https://github.com/watterott/RedFly-Shield/blob/master/src/libraries/RedFly/examples/WebClient_Socket/WebClient_Socket.ino). – Zugriffsdatum: 2012-07-31



# Glossar

- $\rho$   $\rho$  ist der spezifische Widerstand eines Materials oder auch  $\frac{1}{\gamma}$  die spezifische Leitfähigkeit, Seite 4
- A/D-Wandler Gerät, das Analoge Spannungen in digitale Signale umwandelt, Seite 46
- App Kurzform für engl. Application zu Deutsch Programm, Seite 18
- DMS Dehnungsmessstreifen, Seite 9
- Fast-Wach-Moment Begriff, der vom aXbo-Entwickler definiert wurde und den Moment beschreibt, in dem eine Person fast wach ist, Seite 16
- Firmware Software, die in Geräte eingebettet ist und die eigentliche Funktion des Geräts beschreibt, Seite 46
- Lötunkt Anschlussstelle zum Anlöten eines Drahtes/Kabels, Seite 5
- Poissonzahl Querdehnungszahl - Wert, der die Ausdehnung beschreibt, Seite 7
- Raumtemperatur Die Raumtemperatur wird in der Physik mit  $20^{\circ}C$  angegeben, Seite 6
- Remoteing Methodenaufruf über das Netzwerk in verteilten Anwendungen, Seite 51
- SNR signal-to-noise-ratio - Verhältnis von Signalstärke zu Stärke des Rauschens, Seite 34
- TCP Transmission Control Protocol zu deutsch Übertragungs Steuerungs-Protokoll, Seite 24
- VLC VideoLanClient ist eine OpenSource-Software zur Wiedergabe von multimedialen Inhalten, Seite 75
- wav Sounddatei auf Basis eines Standards von Microsoft, Seite 56

*Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.*

Hamburg, 31. August 2012 

---

 Oliver Quast