



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht Masterprojekt 2

Oliver Dreschke

Umsetzung eines „intelligenten Stuhls“ als
Beispiel für das Computational Furniture

Inhaltsverzeichnis

Abbildungsverzeichnis	3
1 Ziele für das Masterprojekt 2	4
2 Überlegungen zur Umsetzung	5
2.1 Schnittstelle Mensch - Computational Furniture	5
2.1.1 Geeignete Form des Stuhls	6
2.1.2 Einsatz eines Kompasses	6
2.1.3 Einsatz eines Schwerpunktsensors	7
2.1.4 Rückmeldungen an den Benutzer	8
2.2 Eingesetzter Controller	9
2.3 Schnittstelle Computational Furniture - Computational World	9
2.3.1 Nutzung der seriellen Schnittstelle	10
2.3.2 Implementieren der Schnittstelle als Human Interface Device/Physical Interface Device	10
3 Stationen der Projektumsetzung	12
3.1 Schwierigkeiten und Lösungen zur Arduino Entwicklungsumgebung	12
3.2 Erweiterung Arduino als Human Interface Device	14
3.3 Nutzung eines Kompassmodules zur Erfassung der Orientierung des Stuhls .	15
3.4 Erkennung des Schwerpunktes und der Bewegungen einer sitzenden Person	17
3.5 Aufbau der Software	18
4 Fazit	21
4.1 Derzeitiger Stand	21
4.2 Perspektive	22
Literaturverzeichnis	23

Abbildungsverzeichnis

3.1	Arduino Duemilanove, FTDI FT232RL und Anschluß X3 markiert	14
3.2	Arduino mit USB-Shield und fertigen Anschluß für ein Kompassmodul	15
3.3	Kompassmodul HMC6343	16
3.4	Kraftsensor aus Personenwaage, links Aufnahmepunkt markiert, rechts Dehnungsmessstreifen mit spezialkleber an Metallkonstruktion befestigt	17
3.5	Grundaufbau der Wheatstone-Brücke (Quelle:Wikipedia)	18
3.6	Layout für Instrumentenverstärker für vier Kraftsensoren auf Punktrasterplatine	19
3.7	Programmfluss zum Lesen von Sensordaten und versenden über USB	20

1 Ziele für das Masterprojekt 2

Diese Arbeit baut nicht auf dem Masterprojekt 1 auf, sondern steht im Rahmen des in der AW2 Arbeit „Computational Furniture“ [Dreschke \(2009\)](#) behandelten Themas, welches die AW1 Arbeit „Der intelligente Stuhl“ [Dreschke \(2008\)](#) einschließt. Diese Arbeiten bewegen sich im Bereich des Living Place Hamburg, einer „intelligenten“ Umgebung, welche sich auf den Bewohner eines Wohnraumes einstellt und auf Basis verschiedener Daten Aktionen automatisch ausführt, um den Menschen von Routineaufgaben zu befreien und ihn zu unterstützen. Aus dem Masterprojekt 1 werden lediglich die Erfahrungen mit der Arduino Hardwareplattform weiter übernommen.

Ziel des Projektes ist es einen ersten Prototypen für mitdenkende Möbel zu konstruieren. Dieser Prototyp bildet die Grundlage, um erste Erfahrungen und Eindrücke darüber zu gewinnen, was für Daten mitdenkende Möbel sammeln und wie diese Daten sinnvoll genutzt werden können. Zusätzlich ermöglicht es ein funktionierender Prototyp ein solches Gerät in einer lebenden Umgebung zu sehen und seine Integration in den Alltag und damit die Akzeptanz der neuen Technik zu erproben.

Da auf diesem Gebiet noch keine Erfahrungen vorliegen, ist als erster Ansatz der Stuhl als Ausgangspunkt gewählt worden. Dieser wurde bereits in der Arbeit „Der intelligente Stuhl“ [Dreschke \(2008\)](#) thematisiert. Erkannt werden soll die Bewegung des Menschen auf dem Stuhl, sowie die Ausrichtung des Stuhls. Somit steht am Ende des Masterprojektes 2 eine Apparatur mit der die Ausrichtung einer Person, sowie einfache Gesten erkannt werden können. Aus den gewonnenen Daten können dann Modelle entwickelt werden, die möglicherweise Rückschlüsse über das Vorhaben oder das derzeitige Empfinden von Benutzern zulassen. Aus diesen Rückschlüssen soll eine intelligente Umgebung Aktionen herleiten können, die den Menschen von Routineaufgaben befreien.

2 Überlegungen zur Umsetzung

Dieses Kapitel beschäftigt sich mit der Frage welche Mittel zu einem „Intelligenten Stuhl“ führen können, welche Hardware Sinn macht und wie Schnittstellen aussehen müssen.

Begonnen wird hierbei mit dem Kapitel [2.1](#). Hier wird auf die Wahl eines geeigneten Möbelstückes, sowie auf die Wahl der Sensoren eingegangen, die in diesem Projekt verwendet werden sollen. Anschließend beschäftigt sich das Kapitel [2.2](#) mit der Wahl des eingesetzten Controllers. Der darauf folgende Teil in Kapitel [2.3](#) beschäftigt sich mit den Schnittstellen von und zum Stuhl.

2.1 Schnittstelle Mensch - Computational Furniture

Beim Computational Furniture ist ein wichtiger Bestandteil der Funktionalität das Sammeln von Daten über den Nutzer eines Möbelstückes. Die Medizin kennt viele Sensorarten um Daten über Menschen zu sammeln. Auch im Bereich des Sports ist das sogenannte Bodymonitoring heute üblich. Informationen zum Thema Bodymonitoring gibt es unter anderem in der Masterarbeit von Olaf Tetzlaff [Tetzlaff \(2008\)](#).

Als Beispiel und erster Ansatzpunkt soll in diesem Projekt eine erste Ausprägung eines möglichen „intelligenten Stuhls“ entstehen, wie er in der AW1 Arbeit „Der intelligente Stuhl“ [Dreschke \(2008\)](#) behandelt wird. Einige Erfahrungen aus dieser Arbeit sollten dann auf weitere Möbelstücke, wie Betten, übertragen werden können.

Im Bereich der Möbel gilt die Besonderheit, dass eingesetzte Sensoren nur sehr ungenau an den Menschen platziert werden können und auch dann noch zuverlässig funktionieren müssen, wenn der Nutzer Kleidung trägt. Im Alltag ist es nicht praktikabel eine besondere Bedienung oder gar das Anlegen eines Sensors für die Nutzung eines Gegenstandes wie eines Stuhls zu verlangen.

Aus diesem Grund müssen für den „Intelligenten Stuhl“ externe Sensoren genutzt werden, die ohne besonderes Zutun des Nutzers funktionieren. Das schränkt die Auswahl an Sensoren stark ein. Beispiele für Sensoren zeigt das Kapitel 3 der AW1 Arbeit „Der intelligente Stuhl“ [Dreschke \(2008\)](#).

In diesem Masterprojekt 2 soll eine möglichst einfache, aber potenziell aussagekräftige Apparatur entstehen. Darum wurde die Auswahl für die Schnittstelle zwischen dem Menschen und dem zu entwickelnden intelligenten Stuhl auf zwei Sensortypen reduziert. Diese bestehen aus einem Kompass und einem Schwerpunktssensor.

Die näheren Überlegungen zur Form des Stuhls, geeigneten Sensoren und Möglichkeiten dem Nutzer ein Feedback zu geben, finden sich in den nächsten Unterkapiteln.

2.1.1 Geeignete Form des Stuhls

Das Einsatzgebiet des fertigen Stuhls ist, das in Kapitel 1 erwähnte Living Place Hamburg. Hier wäre der Aufbau eines intelligenten Sessels als Ausprägung des intelligenten Stuhls sinnvoll. Dieser Sessel ließe sich sehr gut in die Umgebung integrieren. Aus Gründen der Flexibilität wird bei diesem Projekt jedoch zunächst auf einen Bürostuhl zurückgegriffen.

Bürostühle stellen eine sehr komplizierte Art eines Stuhls dar. Sie sind drehbar, in der Höhe verstellbar und bieten oft diverse Mechaniken zur Neigung. Damit bieten sie eine deutlich flexiblere Grundlage zur Erprobung von Sensoren. Zusätzlich sind sie bedeutend leichter zu transportieren und nehmen weniger Stauraum in Anspruch. Hierin liegt gleichzeitig der Nachteil. Bürostühle haben ein deutlich geringeres Volumen als Sessel und bieten keine besonders gute Möglichkeit kameragestützte Sensoren zu verwenden. Kameragestützte Sensoren benötigen einen bestimmten Mindestabstand zum „Motiv“. Bei Bürostühlen ist dieser nicht vorhanden. Sitzfläche und Lehne sind sehr dünn und meist von stabilen Materialien gestützt.

Die Bewegungsmöglichkeiten eines Sessels können jedoch sehr gut mit einem Bürostuhl simuliert werden. Da es in dieser Stufe des Projektes zunächst darauf ankommt ein Gefühl für die Güte der Informationen aus möglichst einfachen Sensoren zu erforschen, ist der Ansatz mit dem Bürostuhl der bessere und flexiblere. In einer nächsten Stufe macht es Sinn die Rückmeldung auf Bewegungen durch kompliziertere Sensoren zu erproben. So kann gleichzeitig eine Aussage über das Kosten/Nutzen Verhältnis der Sensorklassen getroffen werden.

2.1.2 Einsatz eines Kompasses

Der Kompass gibt die Ausrichtung des Stuhls wieder. Daraus kann geschlossen werden, in welche Richtung der Nutzer gerade orientiert ist. Zusätzlich kann die Drehgeschwindigkeit des Stuhls erfasst werden, woraus sich möglicherweise weitere Dinge ableiten lassen.

Um möglichst flexibel zu bleiben, sollte der Kompass störunempfindlich gegenüber Störquellen sein, neigungsunempfindlich und über drei Achsen funktionieren.

Mit der Störunempfindlichkeit gegenüber Störquellen ist gemeint, dass der Kompass Mechanismen haben sollte, die ihm erlauben die korrekte Himmelsrichtung auch dann wiederzugeben, wenn er neben einem massiven Eisenblock angebracht ist oder verschiedene elektrische Störquellen in seiner Umgebung liegen. Dieses ist notwendig, da es gut denkbar ist in einer weiteren Ausbaustufe auf einen elektrisch verstellbaren Sessel umzusteigen. Hier wäre der Kompass potenziell von verschiedenen Motoren und ihren Störfeldern umgeben und müsste weiterhin funktionieren.

Eine Neigungskompensation und Funktionieren über drei Achsen gehen in die gleiche Richtung. Bereits Bürostühle neigen sich mit ihrem Benutzer nach hinten. Auch in diesem Zustand muss der Kompass zuverlässige Richtungsinformationen liefern, um Sinn zu machen. Hierbei ist momentan weniger interessant, um wie viel Grad sich der Stuhl neigt. Wichtiger ist die Sicherheit, nicht darauf angewiesen zu sein, eine Halterung produzieren zu müssen, die den Kompass genau waagerecht hält. Dennoch ist die zusätzliche Information über die Neigung des Stuhls sehr willkommen.

2.1.3 Einsatz eines Schwerpunktsensors

Im Gegensatz zum Kompass des vorherigen Unterkapitels [2.1.2](#), erkennt der Schwerpunktsensor nicht die indirekte Bewegung des Stuhls, also eine mögliche Bewegung eines Menschen. Bei der Erkennung des Schwerpunktes und der Bewegungen einer sitzenden Person wird direkt am Menschen beziehungsweise dem Objekt auf dem Stuhl gemessen. Es ist nicht notwendig, dass der Mensch eine besondere Drehbewegung durchführt.

Der Schwerpunktsensor reduziert die komplexe Sitzhaltung des Nutzers auf einen einzelnen Punkt. Hierdurch können Unruhe, Neigungen des Körpers und Sitzposition erkannt werden. Anbieten tut sich ein Aufbau ähnlich des „Wii-Balanceboards“ der Firma Nintendo. Dadurch kann zusätzlich zu den oben genannten Sitzpositionen auch das Gewicht des Nutzers gemessen werden. Hieraus lassen sich Rückschlüsse auf den Nutzer ziehen. Zum Beispiel wird ein Hund sich bewegen, jedoch vergleichsweise leicht sein. Eine Tasche hingegen kann das gleiche Gewicht wie ein Hund haben, wird sich jedoch nicht bewegen.

Das Balanceboard steht wie eine Personenwaage auf vier Füßen. In jedem Fuß befindet sich ein Kraftsensor. Diese Kraftsensoren geben die Informationen an einen Mikrocontroller weiter. Bei der Personenwaage wird die Größe dieser Werte genutzt um das Gewicht der Person möglichst genau zu ermitteln. Beim Balanceboard werden zusätzlich die Abweichungen verwendet, um daraus den Belastungsschwerpunkt der Person zu errechnen, welche

auf dem Sensor steht. Dafür werden die einzelnen Werte voneinander abgezogen. Heraus kommt eine XY-Position.

Da die Grundelemente des Balanceboards mit der Personenwaage übereinstimmen, ist es möglich aus einer gewöhnlichen Personenwaage ein Balanceboard zu bauen. Hieraus entstehen Vor- und Nachteile. Der Hauptvorteil ist die gewonnene Flexibilität. Das Balanceboard ist aufgrund seiner Ausmaße nicht in einem Stuhl unterzubringen. Zusätzlich steigt beim Selbstbau die Kontrolle über das Gerät. Schnittstellen und Hardware können so geschaffen werden, wie sie für den speziellen Einsatzzweck benötigt werden.

Der Nachteil des Eigenbaus besteht darin, dass mit dem Balanceboard ein sehr gut funktionierendes System existiert. Für die Auswertung der Daten gibt es sowohl fertige Software, als auch Erfahrung an der HAW. Beides ist sofort verfügbar und stellt eine große Zeitersparnis dar. Ob der Eigenbau diese Funktionalität und die Präzision in so kurzer Zeit erreicht ist sehr fraglich.

Dennoch macht es Sinn die Frage zu beantworten, welche Qualität ein Eigenbau erreicht. Die gewonnenen Informationen dienen nicht nur dem Zweck ein Balanceboard zu bauen, sondern zeigen gleichzeitig den Weg auf, wie relativ preiswerte Kraftsensoren verwendet werden können und was von ihnen zu erwarten ist. Damit eröffnen sich neue Felder, wie zum Beispiel der Einsatz in einem Sensornetz, wie es in Kapitel 3.3ff der AW1 Ausarbeitung „Der intelligente Stuhl“ [Dreschke \(2008\)](#) dargestellt wird.

Aus diesem Grund wird eine Eigenkonstruktion des Schwerpunktsensors Teil dieses Masterprojektes.

2.1.4 Rückmeldungen an den Benutzer

Neben der Erfassung von Informationen über den Nutzer, kann es für intelligente Möbel sinnvoll sein, einem Nutzer Rückmeldungen oder Informationen zu liefern. Diese Rückmeldungen sollen in diesem Projekt durch den Einsatz von Vibrationsmotoren ermöglicht werden. So könnte ein Sitzmöbel seinen Nutzer wecken, ihm auf Wunsch mitteilen, dass er eine ungesunde Sitzhaltung hält oder den Hund aus dem obigen Beispiel verscheuchen.

Um die Sensordaten zu erfassen, Vibrationsmotoren zu steuern und Informationen auszutauschen wird ein Controller benötigt. Hiermit beschäftigt sich das nächste Kapitel.

2.2 Eingesetzter Controller

Aus Kapitel 2.1 geht hervor, dass für dieses Projekt sehr einfache Sensoren verwendet werden. Benötigt werden die Möglichkeit analoge Signale zu messen und einen Kompasssensor über den I^2C Bus anzusprechen. Für diese Anforderungen genügt wie beim Masterprojekt 1 ein einfacher 8-Bit-Mikrocontroller. Daher macht es Sinn erneut auf die Arduino Plattform zurückzugreifen. Dieses bringt verschiedene Vorteile mit sich.

1. Die Plattform ist bekannt und verfügbar.
2. Die Plattform gibt es in verschiedenen Bauformen.
3. Im Masterprojekt 1 wurden bereits Erfahrungen zu der Plattform gesammelt.
4. Mit den Mikrocontrollern der Firma Atmel werden gut dokumentierte und weit verbreitete Controller verwendet.
5. Wird mit dem allseits verfügbaren USB-Anschluß verwendet, es gibt zusätzlich sehr viele fertige Erweiterungsmöglichkeiten für Zigbee, LAN, etc.

Sollte es in einer späteren Ausbaustufe notwendig sein, kompliziertere Berechnungen auszuführen, kann das System um leistungstärkere Komponenten ergänzt werden.

2.3 Schnittstelle Computational Furniture - Computational World

Das Computational Furniture, zu dem der in diesem Projekt entstehende „Intelligente Stuhl“ gehört, bewegt sich, wie in Kapitel beschrieben, im Kontext des „Living Place Hamburg“. Hier werden die Daten aller verwendeten Sensoren zentral auf einem so genannten „Event-Heap“ gesammelt und weiteren Systemen zur Verfügung gestellt. Bei dem in diesem Projekt entstehenden „Intelligenten Stuhl“ handelt es sich aus Systemsicht um einen komplexen Sensor, welcher Daten liefert. Aus diesem Grund sollte die Schnittstelle zum Event-Heap des Living-Place-Hamburg implementiert werden.

Es gibt zwei Gründe zunächst eine andere, universelle Schnittstelle zum Sammeln der gewonnenen Daten zu wählen. Zum einen befindet sich der Event-Heap zu Beginn dieses Projektes noch in der Entwicklung, weshalb eine Änderung der Schnittstelle noch möglich ist und die Infrastruktur noch nicht jederzeit zur Verfügung steht. Zum Anderen macht es Sinn eine Schnittstelle zu schaffen auf der unabhängig, mit wenig Aufwand auf einer breiten Basis von Systemen, Versuche mit dem „Intelligenten Stuhl“ durchgeführt werden können.

Da der fertige „Intelligente Stuhl“ seine Daten auf irgendeinem Weg in einen Host-Computer zur Verarbeitung liefern wird, benötigt er eine Schnittstelle zu diesem Computer. Die Wahl des Arduino-Boards, als Basis für das Erfassen von Sensordaten und die Steuerung, bringt zwei Schnittstellen in die engere Auswahl. Die Nutzung der virtuellen seriellen Schnittstelle des Arduino, welche den Standard für den Datenaustausch zwischen Arduino und Computer darstellt, oder das Implementieren des Systems als „Human Interface Device“ (HID). Die Vor- und Nachteile werden in den folgenden Unterkapiteln betrachtet.

2.3.1 Nutzung der seriellen Schnittstelle

Die einfachste Möglichkeit einen Datenaustausch zwischen einem Computer und einem Mikrocontrollerboard nach dem Arduino Standard, besteht darin die virtuelle, serielle Schnittstelle des Arduino zu verwenden. Bei diesem Szenario wird ein Protokoll erarbeitet, welches die Daten des Stuhls über die serielle Schnittstelle an einen Hostcomputer sendet. Diese Daten können mit jedem Terminalprogramm empfangen und gelesen werden. Um die Daten sinnvoll weiter zu verwenden, müsste auf dem Hostcomputer ein Programm laufen, welches die Daten empfängt und auswertet. Für die Arduino Seite existiert hierbei bereits die Implementierung eines erprobten Protokolls. Für die Computerseite existiert ein Javaprogramm zum Empfangen der Daten, welches allerdings das Protokoll nur sehr rudimentär implementiert. Diese Programmstücke sind im Laufe des Masterprojektes 1 entstanden und könnten so verändert werden, das sie auf den „Intelligenten Stuhl“ passen.

Nachteil dieser Methode ist die beschränkte Nutzbarkeit der Daten. Die Daten werden zwar zuverlässig empfangen, haben jedoch ihren praktischen Nutzen fast ausschließlich in Spezialsoftware, welche für die Nutzung des „Intelligenten Stuhls“ geschrieben wird. Zu Vorführungszwecken müsste daher ein Demoprogramm geschrieben werden, welches die Fähigkeiten des Stuhls zeigt. Das bedeutet zusätzlichen Aufwand, welcher aufgrund des geringen Zeitbudgets zu einem eher unspektakulären Ergebnis führen würde.

Diesem Problem wird mit dem Ansatz aus Kapitel [2.3.2](#) begegnet.

2.3.2 Implementieren der Schnittstelle als Human Interface Device/Physical Interface Device

Im vorherigen Unterkapitel [2.3.1](#) wurde eine sehr einfach zu implementierende Schnittstelle benannt, die jedoch den Nachteil aufweist, dass jede Nutzung der Daten mit Implementierungsaufwand auf der Computerseite verbunden ist. Hier setzt der Ansatz an die Schnittstelle zum Host-Computer als „Human Interface Device“ (HID) zu implementieren. Human

Interface Devices sind als Geräteklasse im USB-Standard beschrieben und definieren Geräte, mit denen ein Benutzer interagiert. In den meisten Fällen sind dies Mäuse, Tastaturen oder Joysticks.

Wird der im Rahmen des Projektes zu entwickelnde „Intelligente Stuhl“ als Joystick betrachtet, ergeben sich hieraus einige Vorteile. Treiber für Human Interface Devices stehen unter jedem Desktop Betriebssystem zur Verfügung. Gelingt es also die Schnittstelle so zu gestalten, dass diese Standardtreiber verwendet werden können, könnten Entwickler an beliebigen Plattformen Versuche mit dem Gerät durchführen. Da es zahlreiche Spiele gibt, die mit Joysticks arbeiten, kann die Entwicklung einer Demoapplikation zum einfachen Aufzeigen der Funktionen des „Intelligenten Stuhls“ gespart werden. Wird zum Beispiel ein Motorrad mit Hilfe von Gewichtsverlagerung und Motorradjoystick, aber ohne unnatürliche andere Eingabemöglichkeiten gesteuert, erschließt sich der Nutzen und die Bedienweise sofort.

Um an die Rohdaten des „Intelligenten Stuhl“ zu gelangen, müssen lediglich die Schnittstellen genutzt werden, die jedes Desktopbetriebssystem bereitstellt. Diese Rohdaten stehen so für die Entwicklung von neuen Applikationen zur Verfügung. Damit ist der „Intelligente Stuhl“ nicht nur ein weiterer Joystick für Spiele, sondern steht zusätzlich für seine eigentliche Aufgabe, als Sensor einer intelligenten Umgebung, zur Verfügung.

Der Nachteil besteht in der Notwendigkeit weitere Hardware zu schaffen und dieses deutlich kompliziertere Protokoll zu implementieren.

Funktioniert die Implementation als Human Interface Device, macht es Sinn über die Erweiterung als Physical Interface Device (PID) nachzudenken. Aus Computersicht würde das Gerät wie ein „Force Feedback Joystick“ behandelt werden. Somit könnten dem Nutzer des Stuhls Impulse übermittelt werden, was eine sinnvolle Ergänzung darstellt. Da es sich bei PID um eine Erweiterung der HID handelt, sollten diese Funktionen nachträglich ergänzt werden.

3 Stationen der Projektumsetzung

Dieses Kapitel beschäftigt sich mit der konkreten Umsetzung der unterschiedlichen Teile, die zusammen den „Intelligenten Stuhl“ ergeben. Hierbei wird auf den Überlegungen der einzelnen Unterkapitel von Kapitel 2 aufgebaut.

3.1 Schwierigkeiten und Lösungen zur Arduino Entwicklungsumgebung

Der Name Arduino steht nicht nur für verschiedene, zueinander compatible, Mikrocontroller-boards, sondern gleichzeitig für eine komplette Entwicklungsumgebung mit einfacher Bibliothek für alle wichtigen Funktionen des Mikrocontrollers. Bei der Nutzung des Arduino stehen alle Werkzeuge bereit, die für einen kompletten Entwicklungsweg von der Programmierung bis zum laufenden Programm auf dem fertigen Board benötigt werden, was im Folgenden als „Toolchain“ bezeichnet wird. Dieses ist eines der wichtigen Grundlagen des Erfolges dieser Plattform und gilt für einfache Projekte.

Eine deutliche Limitierung der Arduino-IDE bestand zum Zeitpunkt dieses Projektes darin, dass ein modularer Aufbau von Programmen nur sehr rudimentär möglich war. Eigene Bibliotheken mussten in ein globales Bibliotheksverzeichnis kopiert werden, welches nicht mit den übrigen Programmdateien kompiliert wurde. Das erschwerte eine saubere Aufteilung des Programmcodes und machte die Arduino-IDE für das Projekt unbrauchbar. Diese Limitierung entfällt laut den „Release Notes“ mit der Version 0017 der Arduino-IDE, die auf der Projektseite¹ seit dem 12.08.2009 zum Download bereitsteht.

Auf Arduinobords werden verschiedene Mikrocontroller der Firma Atmel verwendet. Diese Mikrocontroller sind wohl dokumentiert und weit verbreitet, sodass für diese verschiedene Toolchains existieren. Ein genauerer Blick auf die Arduinosoftware enthüllt, dass diese verschiedene Werkzeuge einer der Atmel Toolchains nutzt. Das bedeutet, dass die Arduino-plattform auch außerhalb ihrer Entwicklungswerkzeuge verwendet werden kann.

¹<http://arduino.cc>

Die oben beschriebene Einschränkung der Arduino-IDE führte zu der Entscheidung auf die Atmel Toolchain auszuweichen. Auf der Arduino Projektseite finden sich zwei Anleitungen zur Nutzung der Eclipse IDE zur Entwicklung für den Arduino. Hierbei wird auf ein bestehendes AVR Plugin aufgebaut, welches für die Programmierung von Atmel Mikrocontrollern existiert. Der Weg ist für die Nutzung unter MacOSX beschrieben, funktionierte bis einschließlich der Version 0014 allerdings mit leichten Anpassungen auch auf einer Windows Plattform. Unter Linux wurde es bei diesem Projekt nicht getestet, sollte jedoch ebenfalls funktionieren.

Der beschriebene Weg funktionierte mit Einführung der Version 0015 nicht mehr. Hier wurde eine Veränderung an einer grundlegenden Bibliotheksdatei vorgenommen, woraus Inkompatibilitäten entstanden und ein Kompilieren des Codes auf dem oben beschriebenen Weg unmöglich machten. Ein Umstieg auf die aktuelle Version war aus zwei Gründen notwendig. Zum einen würde ein Verweilen mit der alten Version vor dem eigentlichen Entwicklungsbeginn zu einem unnötigen Ausschluss späterer Entwicklungen und Bugfixes aus dem Projekt führen. Zum Anderen wurde mit der Version 0015 die Unterstützung des Arduino Mega eingeführt, welcher aufgrund seiner zahlreichen Ports für das Projekt besonders gut geeignet ist.

Die Problematik ist im Support Forum erschienen in dem nach einer Lösung gesucht wurde². Eine funktionierende Lösung hat ein Entwickler unter dem Pseudonym lz3060 in einem Blog veröffentlicht³. Diese wird für das Projekt erweitert und genutzt.

Ein weiteres Problem beim Verzicht auf die Verwendung der Arduino-IDE ist das Flaschen des Compilats im .hex Format auf den Mikrocontroller. Die Arduino Plattform nutzt hierfür einen Bootloader, um auf zusätzliche Hardware zum Flashen zu verzichten. Um einen neuen Programmcode auf das Arduinoboard zu bekommen, ist es daher notwendig den Bootloader zu starten, was einen Reset notwendig macht. Dieses kann manuell durchgeführt werden, was besonders bei der Verwendung von Shields, also aufgesteckten Erweiterungen, unpraktikabel ist. Daher musste eine Lösung gefunden werden diesen Reset automatisch durch zu führen. Hierzu wurde keine funktionierende Lösung im Internet gefunden. Die eingesetzte Lösung findet sich im Code der Arduino Software, welche einen Reset ausführt. Mit Unterstützung von Christian Stachow wurde dieser Programmteil aus dem Open Source Code der Arduino IDE extrahiert und als Komandozeilenprogramm ausführbar gemacht. Dieses Programm kann so aus der Eclipse-IDE ausgeführt werden und schreibt das Programm sehr bequem auf das Arduinoboard.

²<http://arduino.cc/cgi-bin/yabb2/YaBB.pl?num=1240829167>

³<http://lz3060.wordpress.com/2009/05/21/avr-makefile-first-attempt>

3.2 Erweiterung Arduino als Human Interface Device

In Kapitel 2.3.2 wurde über die Vorteile des Human Interface Devices (HID) gesprochen. In diesem Kapitel geht es um die praktische Umsetzung.

Beim verwendeten Arduino Duemilanove ist der vorhandene USB-Anschluß direkt mit dem FTDI FT232RL Chip verbunden zu dem auch die vorgesehenen Pins an X3 führen ⁴ (Abb. 3.1). Bei diesem Chip handelt es sich um ein USB zu Serial UART Interface, welches es erlaubt die serielle Schnittstelle des auf dem Controllerboard verwendeten Atmel Mikrocontroller ohne zusätzliche Hardware mit einer USB-Schnittstelle eines PC zu verbinden.

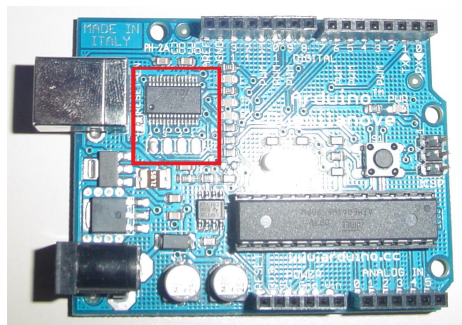


Abbildung 3.1: Arduino Duemilanove, FTDI FT232RL und Anschluß X3 markiert

Das führt dazu, dass weitere Hardware geschaffen werden muss, um das Arduino Board als HID mit dem PC zu verbinden. Zusätzlich muss das USB-Protokoll für Human Interface Devices implementiert werden. Hier hilft das Projekt V-USB der Firma Objective Development, welches eine Softwareimplementierung für low speed USB Geräte auf Atmel Mikrocontrollern zur Verfügung stellt. Die Bibliothek, die vorher unter dem Namen AVR-USB bekannt war, steht unter der GPLv2 oder einer alternativen kommerziellen Lizenz und ist für dieses Projekt geeignet. Auf der Projektseite⁵ finden sich auch ein Schaltplan zum Anschluss der USB-Buchse an einen Atmel Mikrocontroller sowie verschiedene Beispielprojekte.

Ein zusätzliches Beispielprojekt, welches die Bibliothek speziell auf der Arduino Plattform nutzt, findet sich im Internet in einem „Projekt Log“⁶ mit Aufbauanleitung, Schaltplänen, weiterführenden Links und Beispielprojekt für den Arduino. Dieses ist als Einstiegspunkt in die Thematik geeignet und führt zu einem schnell funktionierenden Ergebnis, wobei der Arduino als Tastatur erkannt wird. Mithilfe dieser Internetseite ist das USB-Shield für den „Intelligenten Stuhl“ entstanden, wie es in Abbildung 3.2 dargestellt ist. Die Nummer 3 zeigt hierbei auf

⁴Den Schaltplan des Arduino Duemilanove gibt es unter <http://arduino.cc/en/uploads/Main/arduino-duemilanove-schematic.pdf>

⁵V-USB Projektseite: <http://www.obdev.at/products/vusb/index.html>

⁶<http://code.rancidbacon.com/ProjectLogArduinoUSB> (Autor leider unbekannt)

die zweite USB-Buchse für den Anschluss an den Computer als HID. Zusätzlich wurde die im roten Kasten als eine gekennzeichnete Schnittstelle zu einem Kompassmodul und die mit grünen Kasten mit zwei gekennzeichnete Schnittstelle für den Anschluss an den AD-Wandler des Mikrocontrollers auf dem Arduino Board geschaffen.

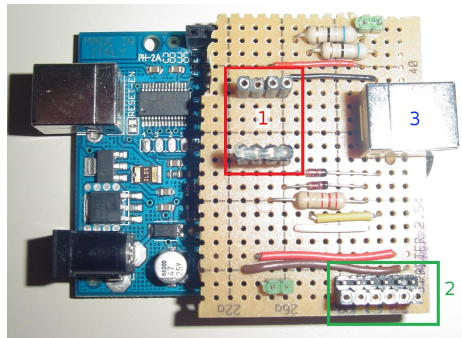


Abbildung 3.2: Arduino mit USB-Shield und fertigen Anschluß für ein Kompassmodul

Um von der Beispielimplementation zum speziellen Eingabegerät „Intelligenter Stuhl“ zu kommen, werden zusätzliche Informationen über Human Interface Devices benötigt. Diese finden sich umfassend auf den Seiten des „USB Implementers Forum, Inc.“ einer nicht kommerziellen Vereinigung, die von den Firmen gegründet wurde, welche die USB-Spezifikation entwickelt haben ⁷. Hier findet sich auch ein Programm, mit dessen Hilfe sich schnell HID Deskriptoren erstellen lassen.

Mit der V-USB Bibliothek und den Informationen aus den HID-Seiten des USB Implementers Forum, konnte die einfache HID-Schnittstelle für dieses Projekt implementiert werden. Die Erweiterung zum Physical Interface Device (PID) wurde aus Zeitgründen noch nicht angegangen.

3.3 Nutzung eines Kompassmodules zur Erfassung der Orientierung des Stuhls

Nachdem im letzten Unterkapitel 2.3.2 die Schaffung der Schnittstelle zwischen dem Stuhl und einem Hostcomputer beschrieben wurde, geht es in diesem Unterkapitel um die Technik zur Erfassung der Orientierung des Stuhls.

Ein möglicher Kompass IC, der die in Kapitel 2.1.2 geforderten Eigenschaften erfüllt, ist der HMC6343 der Firma Honeywell. Diese Komponenten bieten zusätzlich eine Genauigkeit von

⁷Informationen zu Human Interface Devices auf den Seiten des USB Implementers Forum: <http://www.usb.org/developers/hidpage>

zwei Grad und ein I^2C Interface für eine sehr einfache Nutzung. Zusätzlich ist der HMC6343 als Kompassmodul auf einem DIP kompatiblen Breakout Board verfügbar, wie es Abbildung 3.3 zeigt. Darum bietet er sich für dieses Projekt sehr gut an.



Abbildung 3.3: Kompassmodul HMC6343

Da das Modul mit seiner Baugröße von ca. zwei Quadratzentimetern sehr klein ausfällt, passt es bequem auf das in Kapitel 3.2 gezeigte USB-Shield. Das hat zusätzlich den Vorteil, dass es damit automatisch in einer sehr stabilen Halterung steht und die zeitaufwendige Halterungskonstruktion entfallen kann.

Die vom Kompass gelieferten Daten zeigen immer die Richtung in Grad an. Hierbei stehen null Grad für Norden, 90° für Osten, 180° für Süden und 270° für Westen. Da das Einsatzgebiet für den Stuhl ein Raum ist und die Himmelsrichtungen dort meistens eine untergeordnete Rolle spielen. Macht es Sinn diese Werte nur als Option direkt weiter zu leiten. Viel mehr Sinn macht es dem Stuhl eine bevorzugte Ausrichtung zu geben und diese als 0° zu definieren.

Der einfachste Weg das zu erreichen ist ein Taster zur Kalibrierung der Ausrichtung. Eine möglicherweise bessere Lösung ist die bevorzugte Ausrichtung aus der Kombination der Belastung und der Verweildauer in einer Position zu ermitteln und ohne zusätzliche Benutzereingaben fest zu legen. Dieses ist möglich, da Stühle nur selten die Richtung wechseln. Im Regelfall setzt sich ein Nutzer auf einen Stuhl, richtet ihn im Falle eines Bürostuhles aus und sitzt dann längere Zeit in dieser Richtung.

Neben der Ausrichtung kann es für einige Anwendungen von Vorteil sein, wenn die Ausgabe der Richtung nicht wie beim Kompass ausgegeben wird. Soll nur erkannt werden, ob der Nutzer sich nach links oder rechts gedreht hat, kann es sinnvoll sein, die Gradzahlen in einem Bereich von -179° bis $+179^\circ$ oder weniger auszugeben. Diese Möglichkeit wurde ebenfalls implementiert.

Nach dem Erkennen von Bewegungen des Stuhls geht es im nächsten Kapitel um die Erkennung von Bewegungen des Nutzers.

3.4 Erkennung des Schwerpunktes und der Bewegungen einer sitzenden Person

Wie in Kapitel 2.1.3 beschrieben, soll hier das Prinzip des „Wii-Balanceboards“ der Firma Nintendo genutzt werden, wobei eine gewöhnliche Personenwaage als Basis dient.

Die Herausforderung besteht hierbei in der Verstärkung der Spannungsveränderungen in den Kraftsensoren der Waage durch Belastung. Jeder Kraftsensor besteht hierbei aus zwei Teilen, einen mechanischen Aufbau aus Metall und einem Dehnungsmessstreifen (DMS). Abbildung 3.4 zeigt einen Kraftsensor aus der verwendeten Personenwaage. Der mechanische Teil ist so aufgebaut, dass ein kleiner Punkt die Kraft aufnimmt und an den weiteren Körper weiterleitet. Dieser verformt sich durch die Krafteinwirkung. Die Verformung ist mit dem bloßen Auge nicht wahrnehmbar. Um sie zu messen, wird der zweite Teil, der DMS mit Spezialkleber sehr fest mit der Metallkonstruktion verbunden.

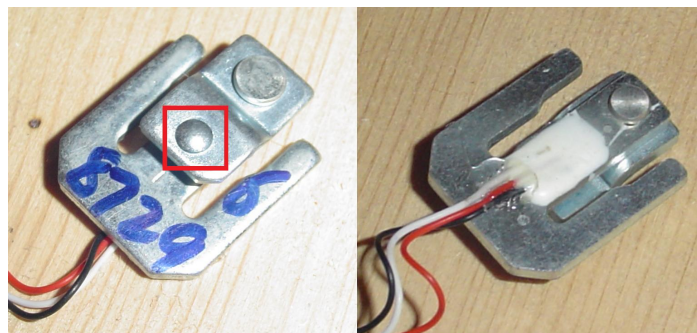


Abbildung 3.4: Kraftsensor aus Personenwaage, links Aufnahmepunkt markiert, rechts Dehnungsmessstreifen mit Spezialkleber an Metallkonstruktion befestigt

Die Dehnungsmessstreifen dieser Personenwaage bilden jeweils eine Halbbrücke mit dem Widerstand $1K\Omega$. Wird diese Halbbrücke mit zwei Widerständen der gleichen Größe ergänzt, entsteht eine Wheatstonesche Messbrücke, wie sie Abbildung 3.5 zeigt. U Stellt hierbei den Spannungsabgriff der Verstimmung der Brücke dar. Sind alle vier Widerstände gleich groß, liegt an U eine Spannung von null Volt an. Dieser Aufbau erlaubt das Messen sehr geringer Spannungsunterschiede.

Diese Spannungsunterschiede zeigen die Biegung des Dehnungsmessstreifens, dessen Widerstand sich aufgrund der Verformung verändert. Da sich die gemessenen Spannungen im Millivolt Bereich bewegen, für eine möglichst präzise Messung am Mikrocontroller des Arduino Boards der gesamte Bereich von null bis fünf Volt ausgenutzt werden sollte, ist eine Verstärkung der Spannung an dieser Stelle notwendig.

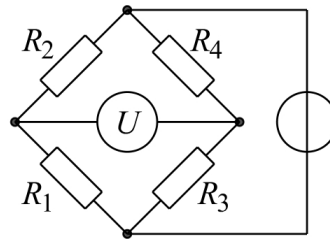


Abbildung 3.5: Grundaufbau der Wheatstone-Brücke (Quelle:Wikipedia)

Für die Verstärkung der Spannung werden Operationsverstärker benötigt. Diese gibt es in verschiedenen Bauformen. Gewöhnliche Operationsverstärker benötigen eine negative und eine positive Betriebsspannung zwischen der verstärkt wird. Da die bisherige Elektronik des „intelligenten Stuhls“ lediglich Spannungen zwischen null und fünf Volt liefert und die vorteilhafte Versorgung über den USB-Port bestehen bleiben sollte, wurden bei diesem Projekt „Single-Supply“ Operationsverstärker genutzt. Diese haben neben der positiven Betriebsspannung einen Masseeingang.

Als brauchbar stellte sich der Operationsverstärker LM324 heraus. Er kann Eingangsspannungen von null bis $V^+ - 1,5V$ verstärken. Für dieses Projekt bedeutet das eine Verstärkung zwischen 0 und 3,5V. Dabei kann der Messbereich des Mikrocontrollers nicht ausgenutzt werden, dafür wird der Schaltungsaufbau einfacher. Es muss sich zeigen, ob die Messauflösung reicht.

Um die geringen Spannungsunterschiede zu verstärken ohne die Messschaltung zu beeinflussen müssen die Operationsverstärker als „Instrumentenverstärker“ verschaltet werden. Das bedeutet, dass für je einen Kraftsensor drei Operationsverstärker benötigt werden. Eine Übersicht über den Aufbau der Schaltung für eine Punktrasterplatine zeigt [Abbildung 3.6](#).

3.5 Aufbau der Software

Nachdem in den vorherigen Unterkapiteln auf die im Laufe des Projektes entstandene Hardware und die Probleme beim Einrichten einer Entwicklungsumgebung eingegangen wurde, beschäftigt sich dieses Kapitel mit der Software.

Bei der Erstellung der Software ist darauf geachtet worden, dass sie möglichst flexibel gehalten ist. Sie ist in C++, mit den bekannten Einschränkungen des für Mikrocontroller beschränkten Sprachumfanges, geschrieben. Die Verarbeitung jedes Sensors ist in eine eigene Klasse in einer eigenen Datei ausgelagert und bietet Funktionen zum Triggern des Messvorganges und zum Auslesen der Messwerte.

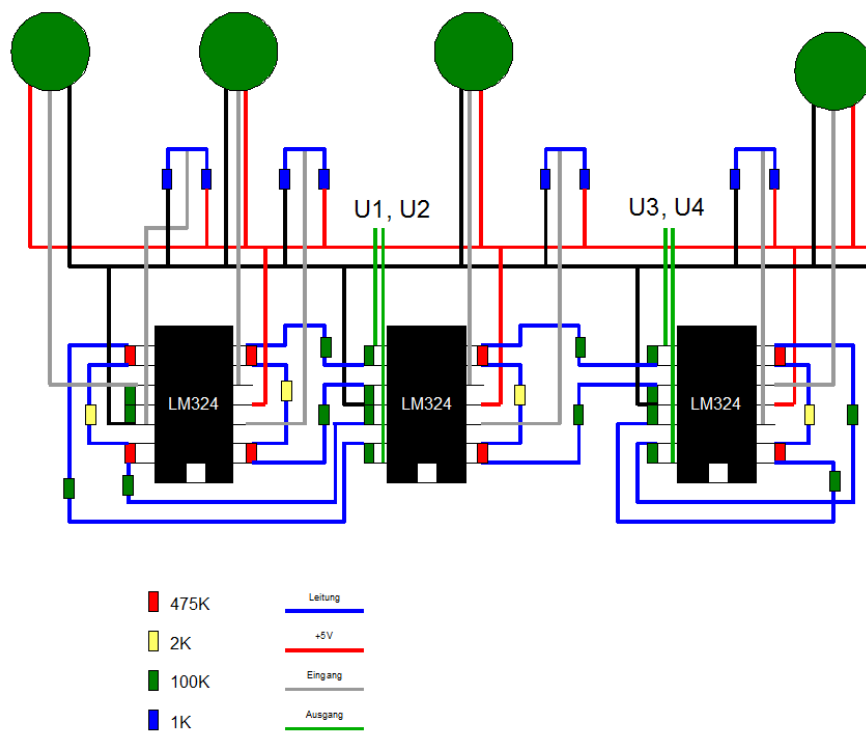


Abbildung 3.6: Layout für Instrumentenverstärker für vier Kraftsensoren auf Punktrasterplatine

Bei jeder Sensorklasse musste darauf geachtet werden, dass keine Wartezeiten enthalten sind, da durch das Warten das für die USB-Kommunikation einzuhaltende Polling-Intervall von 50 ms gestört werden könnte. Zusätzlich muss auf die „delay()“ Funktionen der Arduino Bibliotheken verzichtet werden.

Eine Übersicht der wichtigsten Komponenten und des Programmflusses zeigt Abbildung 3.7. Der dicke Pfeil symbolisiert hierbei die Hauptprogrammschleife. Die grauen Kästen zeigen die Komponenten, die zu der neuen Joystickfunktionalität gehören und von dort aufgerufen werden. Die orangen Felder zeigen verwendete Teile der V-USB-Bibliothek, welche in Kapitel 3.2 vorgestellt wurde.

Die gepunkteten Pfeile stellen die Sensordaten dar, welche in einer Datenstruktur gesammelt werden und zum Erstellen des USB-Report von dort ausgelesen werden. Hierbei sollte die gleiche Reihenfolge eingehalten werden, die auch im USB-Descriptor definiert wurde.

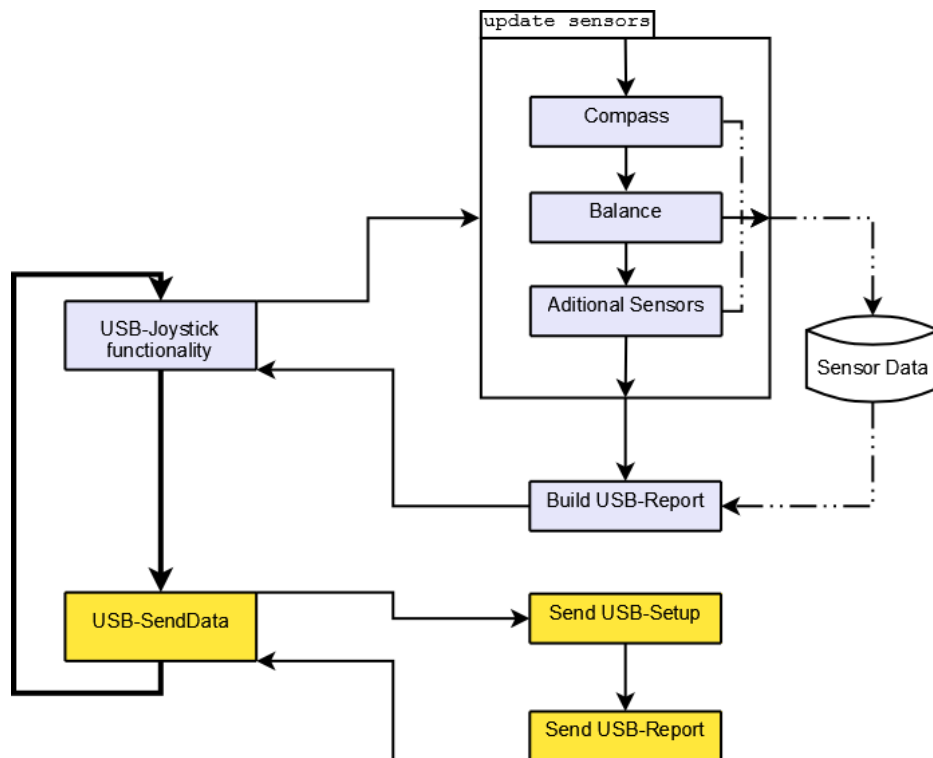


Abbildung 3.7: Programmfluss zum Lesen von Sensordaten und versenden über USB

4 Fazit

4.1 Derzeitiger Stand

In diesem Masterprojekt 2 wurde eine erste Grundlage für einen „intelligenten Stuhl“ umgesetzt. Dafür sind verschiedene Teile fertig gestellt worden.

Es ist eine Entwicklungsumgebung geschaffen worden, mit deren Hilfe komplexere Projekte auf Arduinobasis realisiert werden können.

Die Hard- und Software für die Erkennung der Ausrichtung des Stuhls ist fertig, der Arduino wird unter Windows Vista als Human Interface Device erkannt. Andere Systeme standen für die Erprobung noch nicht bereit. Die Ausrichtung des Arduino Boards mit Kompasssensor bestimmt die Ausrichtung des „Hat Sensors“ des erkannten Joysticks.

Beim Schwerpunktsensor ist der größte Teil der Arbeit fertig. Die Elektronik ist gelötet und beinhaltet noch kleine Fehler, die unschön sind, aber in Software ausgeglichen werden können. Die hierfür benötigte Kalibrierungssoftware ist noch nicht geschrieben, ist vom Aufwand jedoch überschaubar. Der bessere Ansatz ist den Fehler in der Hardware zu eliminieren. Die Software für den Schwerpunktsensor ist geschrieben, aber noch ungetestet. Hier werden erfahrungsgemäß noch kleinere Anpassungen notwendig sein.

Auf die Rückmeldung durch Vibrationsmotoren und den Umbau des Systems zum PID musste aus Zeitgründen verzichtet werden. Für diesen Schritt ist es sinnvoll das USB-Shield neu zu schaffen und auf den Arduino Mega umzusteigen. In dem Zuge macht es unter Umständen Sinn die Elektronik für den Schwerpunktsensor mit auf das Shield zu bringen. Hierfür sollte überprüft werden, ob auf fertige Funktionsverstärker ICs zurückgegriffen werden kann.

Die Nutzung des Stuhls als PID und damit als „Force Feedback Joystick“ bedeutet eine weitere Auseinandersetzung mit dem USB-Teil. Hier könnte eine längere Einarbeitungszeit notwendig werden. Das ist der eigentliche Aufwand.

Als Resultat des Projektes steht ein kompaktes Gerät, welches die Form einer Personenwaage hat. Es kann unter das Sitzkissen eines Bürostuhls montiert werden und gibt dann die geforderten Informationen über USB als HID an den Computer weiter. Da der Stuhl somit

als Joystick erkannt wird, kann seine Funktionalität zu Demonstrationszwecken mit Spielen vorgeführt werden. Am Besten eignen sich hierbei Motorradsimulationen, in denen der Kopf bewegt werden und die Neigung des Motorrads durch Gewichtsverlagerung beeinflusst werden kann.

Der im Moment noch nicht fertige Schwerpunktsensor wird in den nächsten Wochen fertig integriert werden.

4.2 Perspektive

Da der Sensor für den „Intelligenten Stuhl“ noch nicht abschließend erprobt werden kann, muss zunächst herausgefunden werden, wie viel die Daten aussagen. Da die integrierten Sensoren sehr wenig Platz benötigen, stören sie nicht und können als Ergänzung weiter verwendet werden.

Über den HID-Deskriptor können weitere Funktionen einfach in das bestehende System integriert werden, die auf der Computerseite im Joystick sofort zur Verfügung stehen. Alte Funktionalität wird dadurch nicht beeinflusst.

Das Konzept des „Intelligenten Stuhls“ ist in einer sehr rudimentären Basisaustattung umgesetzt. Sinnvolle Erweiterungen gibt es in drei Richtungen.

Die Anzahl der Sensoren innerhalb der Sitzfläche könnte erhöht werden. Hierzu gibt es drucksensitive Auflagen für Stühle aus der Medizin oder dem Automobilbau. Eine solche Matte wurde in der AW1 Arbeit „Der intelligente Stuhl“ vorgestellt [Dreschke \(2008\)](#).

Sehr interessant wäre der Einbau einer Wärmebildkamera. Dieses hätte die Nachteile einer Kamera, die besonders für den Stuhl bedeuten, dass er eine größere Tiefe bekommen müsste. Dafür könnte eine Wärmebildkamera durch leichte Stoffe hindurch die warmen Bereiche des Stuhls und damit des darauf sitzenden Menschen erkennen. Wie bei der oben beschriebenen drucksensitiven Matte dürften dadurch gute Rückschlüsse auf die Sitzhaltung gewonnen werden können. Zusätzlich könnten über Temperaturveränderungen möglicherweise zuverlässig Schlafphasen erkannt werden. Einfache Dinge wie ein Anpassen der Heizung bei zu niedriger Temperatur der gemessenen Person fallen nebenbei ab.

Als Drittes sollte unbedingt über Rückmeldungsmechanismen nachgedacht werden. Dabei könnten neben einfachen Vibrationsmotoren, wie in Kapitel [2.1.4](#) beschrieben, Mechanismen zur Unterstützung der Sitzposition entwickelt werden.

Literaturverzeichnis

[Dreschke 2008] DRESCHKE, Oliver: *Der intelligente Stuhl*. 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomprojekte/master08-09-aw1/dreschke/bericht.pdf>

[Dreschke 2009] DRESCHKE, Oliver: *Computational Furniture*. 2009

[Tetzlaff 2008] TETZLAFF, Olaf: *Bodymonitoring: Entwicklung eines Prototypen für intelligente Kleidung*. 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicomarbeiten/master/tetzlaff.pdf>