



Hochschule für Angewandte Wissenschaften Hamburg

Hamburg University of Applied Sciences

Projektbericht

Stefan Gehn

Evaluation einer infrarotbasierten Multitouchhardware

*Fakultät Technik und Informatik
Studiendepartment Informatik*

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

Inhaltsverzeichnis

1	Einleitung	3
1.1	Projektziel	3
2	Projekt	4
2.1	Verwendete Software	4
2.2	Verwendete Hardware	4
2.2.1	Vorbereitung der Hardware	5
2.3	Testapplikation	6
2.3.1	Hardwareeigenschaften	6
2.3.2	Softwareansätze	9
3	Fazit	13
3.1	Bewertung	13
	Literaturverzeichnis	14
A	Abbildungen	15

1 Einleitung

Das nachfolgend dokumentierte Projekt dient primär als Vorbereitung für eine anstehende Masterarbeit, welche die Usability von Gesten auf Basis von Multitoucheingabe als Schwerpunkt haben soll. Um diese durchzuführen, muss zuvor eine geeignete Hardware angeschafft, getestet sowie eine Ansteuerung der Hardware erfolgen. Die letzteren beiden Punkte werden dabei im Folgenden beschrieben.

1.1 Projektziel

Ziel des Projekts war es, die Möglichkeiten und Grenzen der nun zur Verfügung stehenden Multitouchhardware auszuloten. Hierbei sollten vor allem hardwarebedingte Probleme und Risiken bereits im Vorfeld erkannt und, sofern möglich, ausgeräumt werden.

Es sollte weiterhin geprüft werden, ob und welche der in [Gehn \[2007\]](#) vorgestellten Gesten auf der zur Verfügung stehenden Hardware umsetzbar sind, da diese im Rahmen einer Masterarbeit genauer betrachtet werden sollen. Da innerhalb der anstehenden Masterarbeit auch eine Usability-Studie mit Hilfe der Hardware durchgeführt werden soll, wird im Rahmen dieses Projekts insbesondere die die Qualität der Eingabedaten betrachtet.

2 Projekt

Nach Vorstellung der eingesetzten Hard- und Software findet eine Betrachtung verschiedener Hardwareeigenschaften statt, die während der Arbeit an einer Testapplikation sichtbar wurden oder explizit abgeprüft wurden. Anschließend werden verschiedene Lösungsansätze vorgestellt, um die von der Hardware gelieferten Daten aufzubereiten und zu interpretieren sowie zuvor festgestellte Unzulänglichkeiten der Hardware soweit möglich auszubessern.

Im Anschluss wird in der Auswertung dargelegt, ob sich verwertbare Ansätze für die weitere Arbeit mit der Hardware finden ließen und welche offenen Fragen noch bestehen.

2.1 Verwendete Software

Zur Entwicklung der Testapplikation wurde zu Anfang die Programmiersprache Java ausgewählt, da diese die Grundlage für das DiamondSpin-Toolkit (siehe [Vernier u. a.](#)) bildet. Vor Beginn des Projekts war eine Nutzung des DiamondSpin-Toolkits geplant, was jedoch später verworfen wurde. Im Projektverlauf wurde daraufhin die zuerst in Java erstellte Testapplikation auf C++ portiert. Weiterhin wurde die Qt-Klassenbibliothek der Firma Trolltech Inc. verwendet. Die Wahl für C++ sowie Qt lag unter anderem darin begründet, dass der Autor zuvor schon umfangreichere Erfahrungen in diesem Bereich sammeln konnte und somit eine Einarbeitungszeit in oben genannte Techniken vermieden werden konnte.

Für die Ansteuerung der seriellen Schnittstelle wurde die rxtx-Bibliothek unter Java sowie QExtSerial unter C++ eingesetzt. Bei der Wahl der einzusetzenden Software wurde darauf geachtet, dass diese möglichst plattformunabhängig ist, damit der während des Projekts entstandene Sourcecode möglichst einfach wiederverwendet werden kann.

2.2 Verwendete Hardware

Als Hardware wurde ein infrarotbasierter Multitouchaufsatz der Firma IRTOUCHSYSTEMS (siehe [Irtouch](#)) verwendet. Dieser stand in einer 17"- sowie einer 50"- Variante zur Verfügung.

Der Aufsatz besteht aus einem Rahmen, der vor einer bereits vorhandenen Anzeige angebracht wird. Innerhalb des Rahmens sind Infrarotdioden sowie auf der jeweils gegenüberliegenden Seite Infrarotempfänger integriert, welche ein engmaschiges Infrarotgitter oberhalb der Anzeige erzeugen (vgl. Abbildung 2.1). Die Erkennungsgenauigkeit, die sich aus dem Abstand der Infrarotdioden ergibt, liegt laut Hersteller bei zirka 4mm.

Wird das Infrarotgitter unterbrochen, so wird die dazugehörige Position in X- und Y-Richtung sowie die Breite und Höhe der Unterbrechung per RS232-Schnittstelle übermittelt. Pro Achse können dabei maximal sieben Unterbrechungen zur gleichen Zeit festgestellt und übermittelt werden. Die Position sowie die Ausdehnung einer Unterbrechung wird auf beiden Achsen in Werten von 0 bis 4095 angegeben.

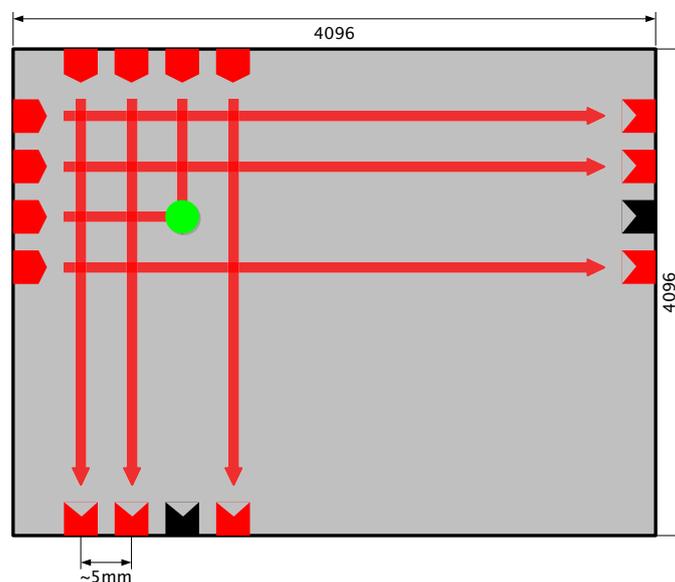


Abbildung 2.1: Schema: Infrarottechnik

2.2.1 Vorbereitung der Hardware

Zu Anfang des Projekts musste zuerst die hardwareseitige Verbindung zum Multitouchaufsatz geklärt werden. Da der Aufsatz nur über einen seriellen Anschluss verfügt und die Stromversorgung nur über einen beiliegenden PS/2-Adapter vorgesehen ist, ist der Betrieb an Laptops sowie Apple-Computern nicht ohne weiteres möglich. Als Lösung wurde ein Adapterkabel (Schaltplan siehe Abb. A.1 auf Seite 15) erstellt, welches den Strom aus einem USB-Anschluss bezieht, da dieser ebenfalls die erforderliche Spannung von 5V bietet. Weiterhin kam ein handelsüblicher USB-to-Serial-Adapter zum Einsatz (Prolific 2303), für den für alle gängigen Betriebssysteme Treiber zur Verfügung stehen.

2.3 Testapplikation

Im Laufe des Projekts wurde eine Testapplikation erarbeitet, um die von der Hardware gelieferten Daten zu visualisieren und erste Verarbeitungsmöglichkeiten für die Daten zu überprüfen. Während des Aufbaus der Testapplikation wurden dabei Eigenschaften der Hardware sichtbar, die an dieser Stelle näher erläutert werden sollen. Im Anschluss an die Betrachtung der Hardwareeigenschaften werden einige Ansätze zur Datenaufbereitung genauer erläutert sowie deren Ergebnisse betrachtet.

Die zuerst angedachte Nutzung des DiamondSpin-Toolkits wurde frühzeitig aufgegeben, da dieses einen Schwerpunkt auf den Mehrbenutzerbetrieb sowie die Entwicklung spezieller GUI-Elemente hatte. Es stellte sich zudem heraus, dass bisher keine weitergehende Interpretation der Touchdaten im Sinne einer Gestenerkennung innerhalb des Toolkits erfolgt, die von Vorteil hätte sein können.

2.3.1 Hardwareeigenschaften

Reaktionsgeschwindigkeit des Aufsatzes

Um die Reaktionsgeschwindigkeit der Hardware beurteilen zu können, wurde sowohl die Anzahl der Datenpakete pro Sekunde als auch der zeitliche Abstand zwischen einzelnen Paketen betrachtet. Getestet wurde mit 1, 5 sowie 10 Fingern, die gleichzeitig den Aufsatz berührten, wobei Überlappungen der X- und Y- Werte vermieden wurden.

Bei Auswertung der in 2.1 auf der nächsten Seite zusammengefassten Ergebnisse stellt sich heraus, dass sich die beiden zur Verfügung stehenden Aufsatzgrößen signifikant in ihrer Abtastrate unterscheiden. Dies ist vermutlich durch die unterschiedliche Anzahl abzufragender IR-Dioden bei identischer Abtastgeschwindigkeit begründet.

Weiterhin nimmt die Anzahl der Pakete/s bei steigender Anzahl von Werten ab. Warum die Paketverzögerung erst bei der maximalen Anzahl von 7 Werten je Achse ansteigt und nicht linear mit der Anzahl zurückzuliefernder Werte zunimmt, ist bislang unklar. Ebenfalls unklar ist, warum dieser Effekt nur beim kleineren Multitouchaufsatz zu beobachten ist.

Wenn man die Anzahl der Pakete pro Sekunde mit der Abtastrate einer Maus vergleicht, so fällt auf, dass die Paketrate des 50"-Rahmen mit 25 Paketen/s unterhalb der Rate einer PS/2- (100 Samples/s) oder USB-Maus (125 Samples/s) liegt (vgl. [Chapweske \[2003\]](#)). Multitouchlösungen auf Basis von FTIR sowie einer herkömmlichen Webcam besitzen üblicherweise ebenfalls eine Abtastrate von 25 - 30 Hz. Hier ist die Webcam der limitierende Faktor. Verglichen mit anderen Multitouchlösungen ist die Abtastrate daher nicht als problematisch anzusehen. Für schnelle Bewegungen wie sie mit einer Maus ausgeführt werden wird die Abtastrate jedoch nicht ausreichen.

Tabelle 2.1: Reaktionsgeschwindigkeit

Aufsatzgröße	Anz. Werte	Pakete/s	Pause zwischen Paketen (ms)
17"	2	76	10 - 20
50"	2	26	30 - 40
17"	10	76	10 - 20
50"	10	26	30 - 40
17"	14	38	20 - 30
50"	14	26	30 - 40

Format der übermittelten Daten

Zu Anfang des Projekts wurden die Daten gemäß serielltem Protokoll eingelesen und die gelieferten Werte unverändert dargestellt. Hierzu ist anzumerken, dass die festgestellten Unterbrechungen in vertikaler sowie horizontaler Richtung in zwei Arrays, jeweils aus Position sowie Ausdehnung der Unterbrechung bestehend, übermittelt werden. Es existieren an dieser Stelle noch keine Wertepaare die den realen Druckpunkten entsprechen würden. Aufgrund dieser Tatsache werden die Werte für beide Achsen in Form von Rechtecken visualisiert, die jeweils die komplette Breite bzw. Höhe der jeweils anderen Achse einnehmen.

Es stellte sich heraus, dass die Position der Werte innerhalb der Pakete ausschließlich an ihre Position auf dem Display gebunden ist. Die Arrays für die X- und Y-Achse sind dabei in aufsteigender Reihenfolge sortiert. Aufgrund dieses Verhaltens ist eine reine Betrachtung von Wertepaaren anhand ihrer Indizes in den Arrays nicht möglich.

Mehrdeutigkeit bei mehr als einem Druckpunkt

Durch die fehlende Zuordnung von Werten beider Achsen, ergeben sich bei mehr als einem Druckpunkt Mehrdeutigkeiten. Abbildung 2.2 auf der nächsten Seite verdeutlicht dies anhand von zwei Druckpunkten, deren Werte sich an insgesamt vier Punkten kreuzen. Die Anzahl der möglichen Druckpunkte steigt quadratisch mit der Anzahl an Unterbrechungen an.

Diese Eigenschaft ist nicht problematisch, solange die exakten Druckpunktdaten für die auszuführende Aktion unerheblich sind. Bei einer Objektdrehung um den Objektmittelpunkt wäre dies z.B. der Fall. Würde einer der Punkte jedoch den Drehmittelpunkt angeben, so müssten beide Punkte während der Drehung eindeutig identifizierbar bleiben. Insbesondere bei der Nutzung durch mehrere Personen ist die Kenntnis über die Positi-

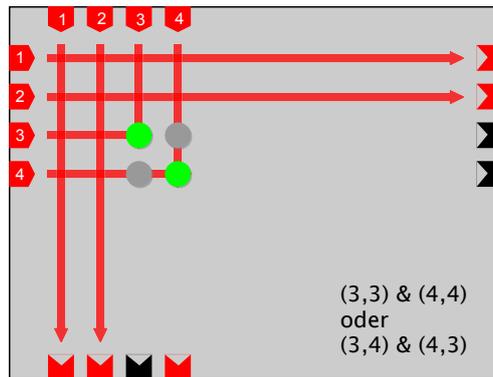


Abbildung 2.2: Ambiguität zweier Druckpunkte

on der realen Druckpunkte zwingend erforderlich, da sonst Aktionen mit Fingern unterschiedlicher Personen nicht parallel ausgeführt werden können, da sich diese gegenseitig beeinflussen würden.

Abschattung von Druckpunkten auf einer der Achsen

Sobald sich zwei Druckpunkte auf der gleichen Höhe auf einer Achse befinden, legt der Punkt mit der größten Ausdehnung die Gesamtausdehnung an dieser Stelle fest (siehe Abbildung 2.3). Die Ausdehnung der restlichen Druckpunkte ist an dieser Stelle nicht feststellbar.

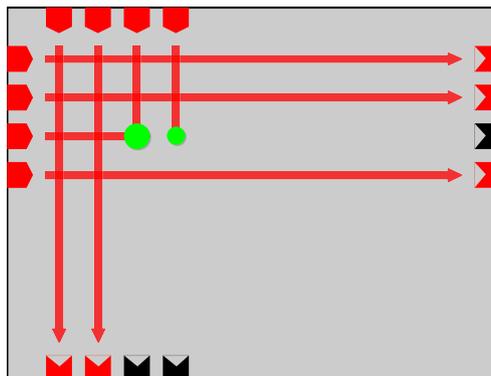


Abbildung 2.3: Abschattung von Druckpunkten auf der gleichen Achse

Dieses Verhalten ist insbesondere für sich bewegende Druckpunkte problematisch, da während der Bewegung einer der zugehörigen Achsenwerte verschwinden zu scheint,

wenn er sich mit einem anderen Wert auf der gleichen Achse überschneidet. Um eine sichere Druckpunkterkennung zu erlauben, muss dieses Verhalten bei der Erzeugung von eindeutigen Druckpunkten berücksichtigt werden.

Eine ähnliche Situation ergibt sich bei zeitlich sehr kurzen Unterbrechungen oder falls eine Unterbrechung nicht vollständig ist (z.B. Finger nicht auf Glasfront aufgesetzt). In diesem Fall kann es insbesondere beim 50"-Aufsatz dazu kommen, dass kurzzeitig nur ein Achsenwert für den Druckpunkt übertragen wird. Der zweite Achsenwert folgt entweder in einem nachfolgenden Datenpaket oder fehlt komplett.

Weitere Eigenschaften

Werte auf einer Achse, deren Abstand nahe an oder unterhalb der Genauigkeit der Hardware liegen, neigen dazu, zu einem einzelnen Wert zusammengefasst zu werden. Da die Genauigkeit bei wenigen Millimetern liegt, sind vom Benutzer sichtbare Effekte die durch dieses Verhalten entstehen eher unwahrscheinlich, solange die Anzahl der Werte eines Datenpakets nicht als einziger Faktor für eine Unterscheidung von Zuständen oder Objekten verwendet wird.

Es wurde weiterhin festgestellt, dass die X-Achse der 50"-Aufsätze ihren Nullpunkt nicht auf der linken Rahmenseite sondern auf der rechten Rahmenseite hat. Der Nullpunkt der Y-Achse befindet sich bei allen getesteten Rahmen auf der oberen Rahmenseite und ist damit identisch zum Nullpunkt der Y-Achse der Bildschirmdarstellung.

2.3.2 Softwareansätze

An dieser Stelle werden die während des Projekts getesteten Ansätze vorgestellt, welche zum Einen der Datentransformation in ein für Applikationen hilfreiches Format dienen sollen, zum Anderen die zuvor genannten Eigenschaften der Hardware verbessern bzw. Unzulänglichkeiten ausgleichen sollen. Insbesondere die Abschattung von Druckpunkten auf gleicher Achsenhöhe ([2.3.1](#) auf der vorherigen Seite) sowie die Mehrdeutigkeit bei mehr als einem Druckpunkt ([2.3.1](#) auf Seite 7) werden dabei im Folgenden betrachtet. Desweiteren wird ein Ansatz für die Unterscheidung von Objekten und Handstellungen vorgestellt.

Eindeutige Druckpunkterkennung

Um eine eindeutige Druckpunkterkennung zu ermöglichen muss zuerst eine paarweise Zuordnung von Werten aus den Arrays beider Achsen (siehe [2.3.1](#) auf Seite 7) erfolgen.

Die erzeugten Wertepaare werden daraufhin als Druckpunkte angesehen. Der aktuelle Ansatz erzeugt Wertepaare durch eine Iteration über beide Arrays, was in der Praxis eine sichere Zuordnung zweier Werte nur zulässt, wenn innerhalb eines Datenpakets maximal ein Wert je Achse hinzukommt. Für den Anwender bedeutet dies konkret, dass Finger nacheinander und niemals gleichzeitig auf der Oberfläche des Bildschirms auftreffen dürfen. Es ist zu vermuten, dass diese Einschränkung vom Benutzer wahrgenommen und nicht ohne explizite Erklärung verständlich ist.

Um die erzeugten Druckpunkte über die Zeit verfolgen zu können, wird in nachfolgenden Datenpaketen eine Suche nach Wertepaaren durchgeführt, welche eine Ähnlichkeit mit zuvor erkannten Druckpunkten aufweisen. Die dabei erhaltenen Wertepaare werden daraufhin als Nachfolger der alten Druckpunkte angesehen. Als Ähnlichkeitskriterium wird dabei die Position sowie die Größe im Vergleich zum alten Druckpunkt verwendet. Die danach verbliebenen Werte werden wie anfangs beschrieben zu neuen Wertepaaren kombiniert.

Optimierungsmöglichkeiten Eine mögliche Verbesserung des Ähnlichkeitskriteriums wäre die Beachtung der bisherigen Bewegungsrichtung des Druckpunktes um den Suchradius entsprechend zu optimieren (Suche nur in Bewegungsrichtung). Eine Gefahr bei Einbeziehung der Bewegungsrichtung wäre jedoch eine zu geringe Abtastrate auf Seiten der Hardware, da diese eine Richtungsänderung nicht mitbekommen würde und der neue Druckpunkt außerhalb des gesuchten Bereichs liegen könnte.

Schwierigkeiten Ein grundlegendes Problem der obigen Vorgehensweise ergibt sich bei einer ungeraden Anzahl von Werten innerhalb eines Datenpakets, da dann ein Wert am Ende der Wertzuordnung ungenutzt bleibt. Für genau drei verbliebene Werte (zwei Werte auf der Einen und ein Wert auf der anderen Achse) wurde die Annahme getroffen, dass sich zwei Druckpunkte auf gleicher Höhe auf einer der Achsen befinden (siehe [2.3.1](#) auf Seite [8](#)). Der "fehlende" Achsenwert wird durch Duplizierung des einzelnen Achsenwerts erzeugt.

Ein weiteres Problem ergibt sich, wenn ein erkannter Druckpunkt einen Achsenwert eines anderen Druckpunkts kreuzt. Dieser Übergang von vier Werten (zwei pro Achse) auf nur drei Werte sowie der umgekehrte Fall werden momentan noch nicht gesondert betrachtet. Ein Ansatz um die Druckpunkte an dieser Stelle weiterhin sicher zu erkennen steht noch aus. Ein möglicher Ansatz wäre die Fortführung der Bewegung der beiden sich auf einer Achse überlappenden Druckpunkte unter Berücksichtigung ihrer bisherigen Bewegungsrichtung. Es würden für diesen Fall für einen bestimmten Zeitbereich fiktive Druckpunkte erzeugt werden, unter der Annahme, dass sich beide Druckpunkte in nachfolgenden Datenpaketen wiederfinden lassen.

Wertglättung

Da sich die übertragenen Werte auch bei stillstehenden Objekten verändern, wurde versucht, eine Wertglättung über zwei Datenpakete durchzuführen. Diese basierte, ähnlich wie die Druckpunkterkennung auf der Wiedererkennung von Werten über zwei Datenpakete. Als Ähnlichkeitsmerkmal wurde ebenfalls die Position sowie Ausdehnung des Wertes verwendet. Die zueinander zugeordneten Werte wurde daraufhin testweise über zwei verschiedene Mittelwertformeln miteinander vereinigt:

$$\overline{wert} = \frac{wert_t + wert_{t+1}}{2}$$
$$\overline{wert} = \frac{(wert_t * 9) + wert_{t+1}}{10}$$

Bei nachfolgenden Tests mit beiden Mittelwertberechnungen stellte sich jedoch heraus, dass diese die Fluktuationen in den Koordinaten stillstehender Objekte nur wenig milderten und zudem die in [2.3.2](#) auf Seite 9 vorgestellte Druckpunkterkennung merklich verschlechterte. Bei Anwendung der Wertglättung verlor die Druckpunkterkennung wesentlich öfter den Druckpunkt bei schnellen Bewegungen als dies bei ungefilterten Werten der Fall gewesen war. Ein Grund für dieses Verhalten ließ sich nach einigen Tests nicht endgültig finden. Die Wertglättung wurde daraufhin bis auf Weiteres deaktiviert, da sie mehr Vor- als Nachteile brachte.

Objekterkennung

Zusätzlich zur Erkennung einzelner Druckpunkte und ihrer Bewegung über die Zeit ist auch von Interesse, durch welches Objekt Druckpunkte erzeugt werden. So kann z.B. bei Erkennung einer Handkante auf dem Display eine Aktion ähnlich eines Wegwischens umgesetzt werden. Ziel der Objekterkennung ist primär die Unterscheidung mehrerer Eingabemodi.

Der aktuelle Ansatz für die Objekterkennung basiert auf der Verknüpfung mehrerer Eigenschaften, die sich aus den Eingabedaten ableiten lassen

- Breite und Höhe eines Druckpunktes
- Breite und Höhe des Rechtecks über alle Druckpunkte
- Verhältnis von Breite zu Höhe
- Anzahl von Werten oder Wertepaaren

Ein Objekt wird dabei ausschließlich durch Einschränkung einer oder mehrerer der oben genannten Parameter definiert. Um eine sichere Unterscheidung von Objekten zu er-

möglichen, ist es daher von Vorteil, möglichst viele Constraints für ein Objekt zu definieren.

Verwendete Einheiten Da alle Touchaufsätze bei unterschiedlicher physikalischer Größe das gleiche Koordinatensystem verwenden (vgl. 2.2 auf Seite 4), sind die o.g. Eigenschaften bei Beibehaltung des Koordinatensystems nicht zwischen unterschiedlichen Aufsatzgrößen vergleichbar. Um die Objekterkennung von der Aufsatzgröße zu entkoppeln, wurde eine Umrechnung der Touchkoordinaten in die Einheit Millimeter vorgenommen. Ein Nachteil dieses Vorgehens ist, dass die physikalische Größe des Touchaufsatzes zuvor bekannt sein muss, da diese Werte nicht vom Aufsatz über die serielle Schnittstelle abgefragt werden können.

Qualität der Objekterkennung Bei kurzen Tests mit unterschiedlichen Personen ergaben sich bisweilen Probleme mit der sicheren Erkennung von Objekten, die sich jedoch größtenteils auf unpassend gewählte Constraints zurückführen lassen. Eine genauere Festlegung der Constraints steht an dieser Stelle noch aus.

3 Fazit

Im Rahmen des Projekts wurde die vorhandene Multitouchhardware in Betrieb genommen und auf ihre Eignung für weitere Arbeiten überprüft. Dabei wurden zum einen Fähigkeiten als auch Einschränkungen der Hardware festgestellt sowie mögliche Verfahren für eine Datenaufbereitung evaluiert.

3.1 Bewertung

Das in [Gehn \[2007\]](#) vorgestellte Gestenrepertoire lässt sich mit der vorhandenen Hardware grundsätzlich umsetzen, da alle dafür benötigten Daten (Abstände, Winkel) zur Verfügung stehen. Insbesondere für die in [2.3.1](#) auf Seite [7](#) und [2.3.1](#) auf Seite [8](#) dokumentierten Probleme müssen jedoch noch andere Verfahren gefunden werden, um diese ausgleichen zu können.

Eine Mehrbenutzerfähigkeit ist aufgrund der zuvor genannten Probleme unwahrscheinlich, da nicht davon auszugehen ist, dass Benutzer die aus diesen Hardwareeigenschaften entstandenen Phänomene tolerieren werden. Die Eingaben mehrerer Nutzer werden sich unweigerlich gegenseitig beeinflussen.

Literaturverzeichnis

- [Irtouch] : *IRTOUCHSYSTEMS Co. Ltd.* – URL <http://irtouch.com>
- [Chapweske 2003] CHAPWESKE, Adam: *The PS/2 Mouse Interface*. April 2003. – URL <http://www.computer-engineering.org/ps2mouse/>
- [Gehn 2007] GEHN, Stefan: *Eingabepformance von Multitouch-Displays am Beispiel von Spielen*. July 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2007/gehn/bericht.pdf>
- [Shen u. a. 2004] SHEN, Chia ; VERNIER, Frédéric D. ; FORLINES, Clifton ; RINGEL, Meredith: *DiamondSpin: an extensible toolkit for around-the-table interaction*. In: *CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems*, ACM Press, 2004, S. 167–174. – ISBN 1-58113-702-8
- [Vernier u. a.] VERNIER, Fred ; SHEN, Chia ; BESACIER, Guillaume ; WIGDOR, Daniel ; FORLINES, Cliff: *DiamondSpin Tabletop Toolkit Project*. – URL <http://diamondspin.free.fr>

A Abbildungen



Abbildung A.1: Schaltplan für Adapter zur Stromversorgung per USB