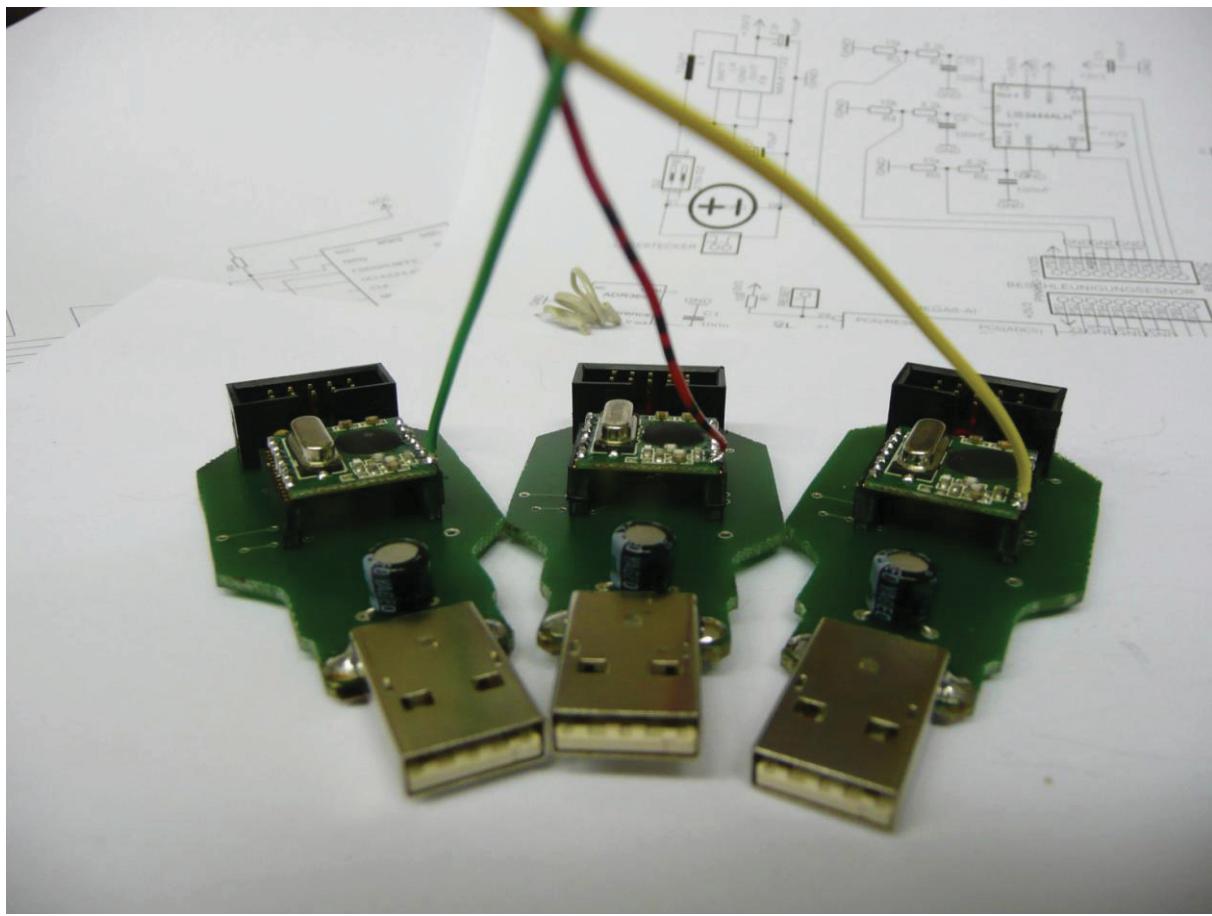


C-A-T Steuerinterface



Jugend Forscht Projekt von:

Phil Stelzer
Ole Stecker-Schürmann
Tobias Markus

Projektbetreuer: Berthold Sommer
Berufskolleg der Stadt Rheine

Inhaltsverzeichnis

1.	Die Aufgabenstellung	3
2.	Die Umsetzung	3
2.1.	Mehrdimensionale 3D Eingabesysteme	3
2.2.	Klein, Handlich, Flexibel	3
3.	Die Anforderungen	3
3.1.	Anforderungen an das Gesamtsystem	3
3.2.	Anforderungen an die Hardware:	4
3.3.	Anforderungen an die Software:	4
4.	Kurzbeschreibung	4
5.	Hardware	5
5.1.	Handinterface	5
5.1.1.	Spannungsversorgung	5
5.1.2.	Controllereinheit	5
5.1.3.	Sensorik	5
5.1.4.	Funkmodul	6
5.2.	Empfänger	6
5.2.1.	Funkmodul	6
5.2.2.	Controllereinheit	6
5.2.3.	USB – Schnittstelle	6
6.	Schaltplan und Layout	6
6.1.	Handsteuerung	6
6.2.	Empfänger	8
6.3.	Bilder der fertigen Platinen	8
7.	Physikalischer Hintergrund	9
7.1.	Winkelbestimmung	9
7.2.	Positionsbestimmung	9
8.	Software	9
8.1.	Handinterface und Empfänger	10
8.1.1.	Programmablaufplan	10
8.1.2.	Beschreibung	10
8.2.	Anwendung	11
8.2.1.	UML – Diagramm	11
8.2.2.	Beschreibung	11
8.2.3.	Testprogramme	12
9.	Entwicklungen des Handinterface	13
10.	Abgrenzungen zu ähnlichen Systemen	14
11.	Resümee	14
12.	Quellen	15
13.	Danksagungen	15

1. Die Aufgabenstellung

Die Computermaus hat sich seit ihrer Vorstellung im Jahre 1968, als einfaches, universelles und intuitives Eingabemedium etabliert.

Sie ist aber grundsätzlich nur als Eingabegerät für zweidimensionale Systeme konzipiert. Zur Steuerung von mehr als zweidimensionalen Systemen ist sie daher nur bedingt geeignet.

Für die Steuerung komplexer Anwendungen im dreidimensionalen Raum wird ein Eingabemedium benötigt, welches die Fähigkeit besitzt auch mehrdimensionale Eingaben zu erfassen und weiter zu leiten. Mit dieser Arbeit stellen wir ein universelles Computereingabesystem vor, mit dem intuitiv komplexe Objekte im dreidimensionalen Raum gesteuert werden können.

Der "Mauszeiger" wird ähnlich wie bei einer klassischen Computermaus intuitiv über die Handposition navigiert. Zusätzlich zur Position ist auch die Lage im Raum beeinflussbar.

Dieses ermöglicht eine einfache, und flexibel einsetzbare Steuerung von komplexen dreidimensionalen Systemen.

2. Die Umsetzung

2.1. Mehrdimensionale 3D Eingabesysteme

Entweder werden drei oder bis zu sechs Freiheitsgrade gesteuert. Die Position und/oder die Lage im Raum. Besonders im CAD-Bereich haben sich die 3D Mäuse von Logitech etabliert.

Diese Systeme sind Kabelgebunden und können somit nur an Tischen genutzt werden. Für Präsentationen und den mobilen Einsatz sind sie weniger geeignet. Andere Systeme können nur drei Freiheitsgrade erfassen und benötigen einen Aufwändigen Sensoraufbau (zum Beispiel: <http://www.wpi.edu/news/20067/popsciaward.html>).



2.2. Klein, Handlich, Flexibel

Mit dem hier vorgestellten System steuern wir sechs Freiheitsgrade und sind weitgehend unabhängig von örtlichen Gegebenheiten.

Unser Eingabemodul beschränkt sich auf ein kleines handliches Modul, welches einfach auf die Hand gesteckt wird. Die komplette Sensorik befindet sich in diesem Handinterface, ein zusätzlicher Sensoraufbau für die Positionserfassung im Raum ist nicht erforderlich. Die Sensordaten überträgt das System per Funk an den Rechner. Dort werden die Daten mit einem speziellen Programm ausgewertet und an das zu steuernde System übergeben.

Weitere Funktionen können über Tasten und Schieber angesteuert werden.

3. Die Anforderungen

Für die Umsetzung der Aufgabenstellung, haben wir Anforderungen definiert. So konnten wir zielgerichtet arbeiten.

3.1. Anforderungen an das Gesamtsystem

Folgende Funktionen sollen realisiert werden:

- Die Steuerung der Position von Objekten in 3- Dimensionen
- Die Beeinflussung der Lage eines mehrdimensionalen Objektes (Rotation um die X/Y/Z Achse)
- Zusatzfunktionen können durch Knöpfe und Schieber aktiviert werden

3.2. Anforderungen an die Hardware:

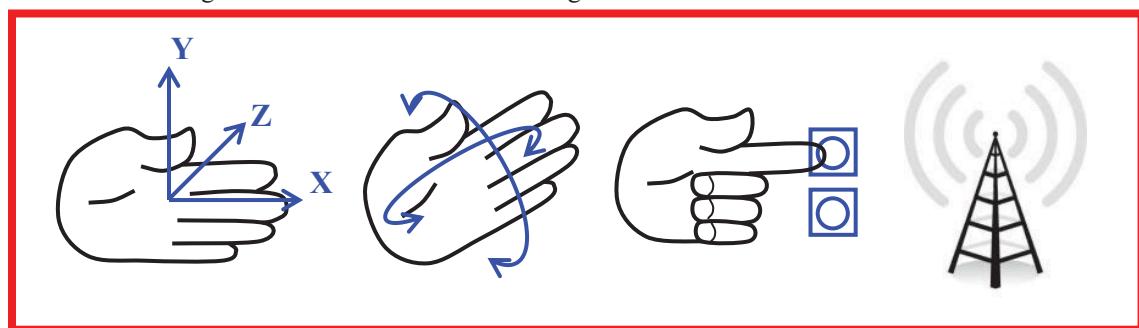
- Die X/Y/Z Position und die Drehung der Hand wird detektiert und an den Rechner übertragen.
- Fingereingaben sind zu verarbeiten
- Um Flexibilität und Bewegungsfreiheit zu gewährleisten werden alle Sensordaten per Funk übertragen
- Eine sehr kleine Bauform gewährleistet Komfort und offeriert einfache Möglichkeiten zur Befestigung
- Um eine maximale Laufzeit bei geringer Baugröße zu erreichen muss die Hardware energieeffizient arbeiten

3.3. Anforderungen an die Software:

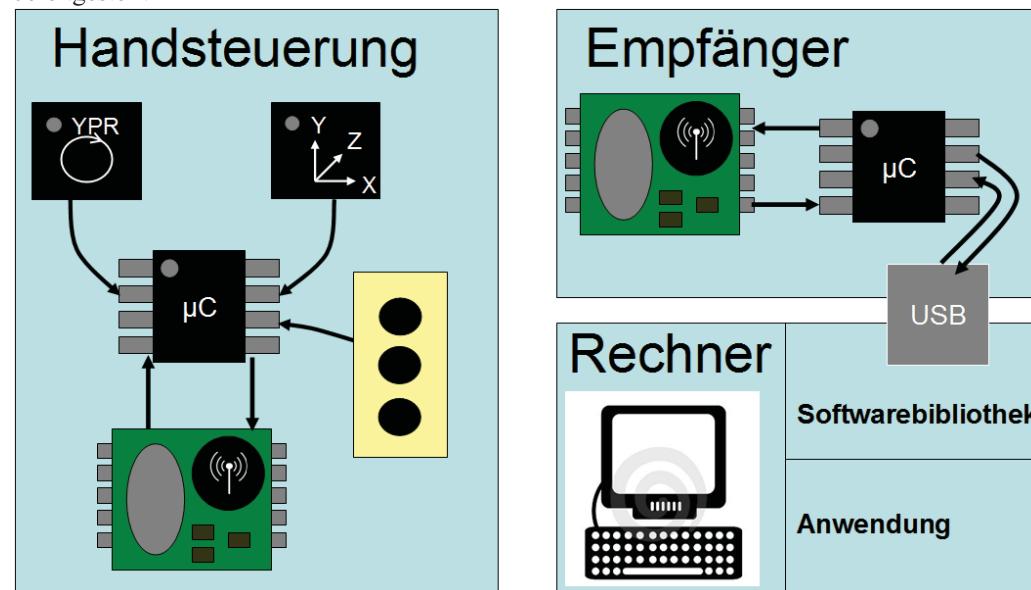
- Einfache Handhabung des Programms (selbsterklärend)
- Übergeben der Handposition an den Mauszeiger
- Auslesen der USB Schnittstelle

4. Kurzbeschreibung

Die Sensoren der Handelektronik erfassen Beschleunigung und die Winkelgeschwindigkeit der Hand. Zusätzlich können durch Eingabeelemente weitere Funktionen gesteuert werden.



Alle Funktionen werden von einem Mikrocontrollersystem koordiniert. Zur Erfassung der Positionsveränderung wird ein 3-Achsen Beschleunigungssensor verwendet. Die Winkelgeschwindigkeit wird von einem Gyrosensor gemessen. Alle diese Sensoren werden über den A/D Wandler des Mikrocontrollers ausgelesen und per Funk an den Empfänger in einem Computersystem übermittelt. Die Spannungsversorgung wird über einen kleinen Akku bereitgestellt



Funktionsdiagramm: Grundlegende Funktionen wurden in Blöcke aufgeteilt

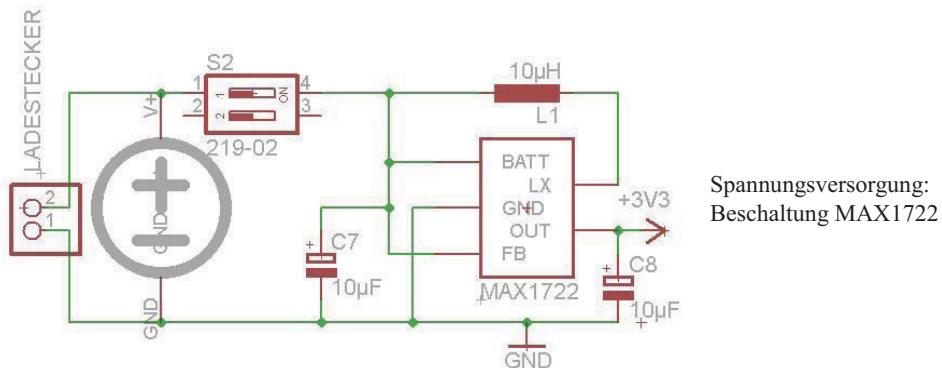
Der Empfänger übergibt die Daten über den USB- Anschluss an den Rechner, wo sie dann weiterverarbeitet werden.

5. Hardware

5.1. Handinterface

5.1.1. Spannungsversorgung

Da das Handinterface möglichst klein werden soll haben wir uns für einen kleinen Lithiumakku, als Spannungsquelle entschieden. Um eine konstante Versorgungsspannung sicher zu stellen, setzen wir einen Step Up/Down Wandler von Maxim (MAX1722) ein. Hierdurch ist es uns möglich die Versorgungsspannung der Schaltung vom Ladezustand des Akkus unabhängig zu gewährleisten, da der Step Up Wandler aus einer Spannung zwischen 0,8-5,5V geregelte 3,3V erzeugt. Der MAX1722 liefert des Weiteren maximal 150mA was für unsere Anwendung genügt. Er hat außerdem einen äußerst hohen Wirkungsgrad von über 90% was ihn für Handgeräte prädestiniert. Über den Schalter S2 lässt sich die komplette Schaltung abschalten.



5.1.2. Controllereinheit

Als Controller für unser Handinterface haben wir uns für den Mikrocontroller ATMega8 von Atmel entschieden, da wir bereits gute Erfahrungen mit diesem Controller haben. Dieser kleine Controller bietet eine ausreichende Funktionalität, um das System zu managen. Wir lassen den Controller mit der internen Taktfrequenz von 8 MHz laufen. Da die meisten Befehle des Mega8 1-2 Takte benötigen haben wir somit eine Rechenleistung von bis zu 8MIPS. Der Mikrocontroller wurde zusätzlich mit einer Referenzspannungsquelle, dem ADR360 beschaltet um so eine genaue Spannungsreferenz für den A/D Wandler (ADC) zu gewährleisten.

Der ADR360 liefert eine Referenzspannung von 2,048V. Die Auflösung des A/D Wandler beträgt 10bit also 1024 Schritte. Eine aufwändige Rechnung mit Kommazahlen wird so vermieden. Analog Devices, der Hersteller der Referenzspannungsquelle, gibt eine Abweichung von maximal $\pm 3\text{mV}$ an. Dieser Fehler ist statisch und somit einfach zurückzurechnen. Die Welligkeit der Referenzspannung gibt der Hersteller mit $6,8\mu\text{V}$, also 3,32ppm an. Dieser statistische Fehler liegt weit unter dem statistischen Fehler des A/D Wandlers so, dass wir ihn für unsere Betrachtungen nicht zu berücksichtigen brauchen. Der statistische Fehler des A/D Wandlers wird von Atmel mit $\pm 2\text{LSB}$ (0,195%) angegeben.

Für unsere Anwendung (Handsteuerung) ist dies ein akzeptabler Wert, welcher ein Nutzer nicht spüren wird.

5.1.3. Sensorik

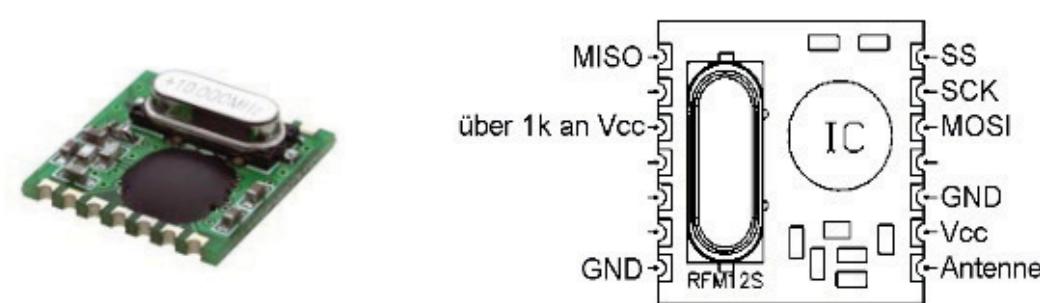
Für die Erfassung der Position wird ein 3-Achsen Beschleunigungssensor der LIS344ALH von ST genutzt. Die Winkelgeschwindigkeit wird mit einem Gyrosensor dem LPR5150AL gemessen.

Über die Pins ST (Self Test), PD (Power Down) und FS (Full Scale) können durch das jeweilige Logic Level, welche vom Mikrocontroller angesteuert werden können, verschiedenste Einstellungen vorgenommen werden. So kann über die ST Pins ein Selbsttest zur Funktionskontrolle der Sensoren aktiviert werden. Über PD können die Sensor in einen Stromsparmodus gefahren werden und über FS kann der Messbereich gewählt werden, $\pm 2g$

oder $\pm 6g$. Der LIS344ALH gibt eine Ausgangsspannung von $V_{out} = \frac{V_{dd}}{2} + \frac{V_{dd}}{5 \cdot g}$ oder $V_{out} = \frac{V_{dd}}{2} + \frac{V_{dd}}{15 \cdot g}$.

Dieser Ausgangspegel wird durch einen Spannungsteiler auf einen Bereich von 0 - ca. 2,048V angepasst. Der LPR550AL ist nach Datenblatt beschaltet, wir verwenden die verstärkten Spannungsausgänge. So können wir einen Wertebereich zwischen $\pm 1500^{\circ}\text{s}$ erfassen. Der Sensor gibt eine Ausgangsspannung von $\frac{1,23V + 2mV}{\circ \cdot s}$. Auch der LPR550AL lässt sich über einige Pins einstellen wobei die Pins ST und PD die gleiche Funktion wie beim LIS344ALH haben

5.1.4. Funkmodul



Als Funkboard benutzen wir das RFM12 Modul, da es sehr klein und preiswert ist. Außerdem haben wir bereits gute Erfahrungen mit dieser Baugruppe gesammelt. Die Anbindung an dem Mikrocontroller erfolgt über die SPI Schnittstelle. Das RFM12 fungiert als Slave und der Mikrocontroller als Master.

5.2. Empfänger

5.2.1. Funkmodul

Das Funkmodul des Empfängers ist ebenfalls das RFM12. Hier wird das Funkmodul im Empfangsmodus betrieben und gibt die Daten weiter an den Controller.

5.2.2. Controllereinheit

Beim Empfänger ist ebenfalls ein ATMega8 das zentrale System, welcher die Daten vom Funkmodul empfängt und zur USB Schnittstelle weiter leitet.

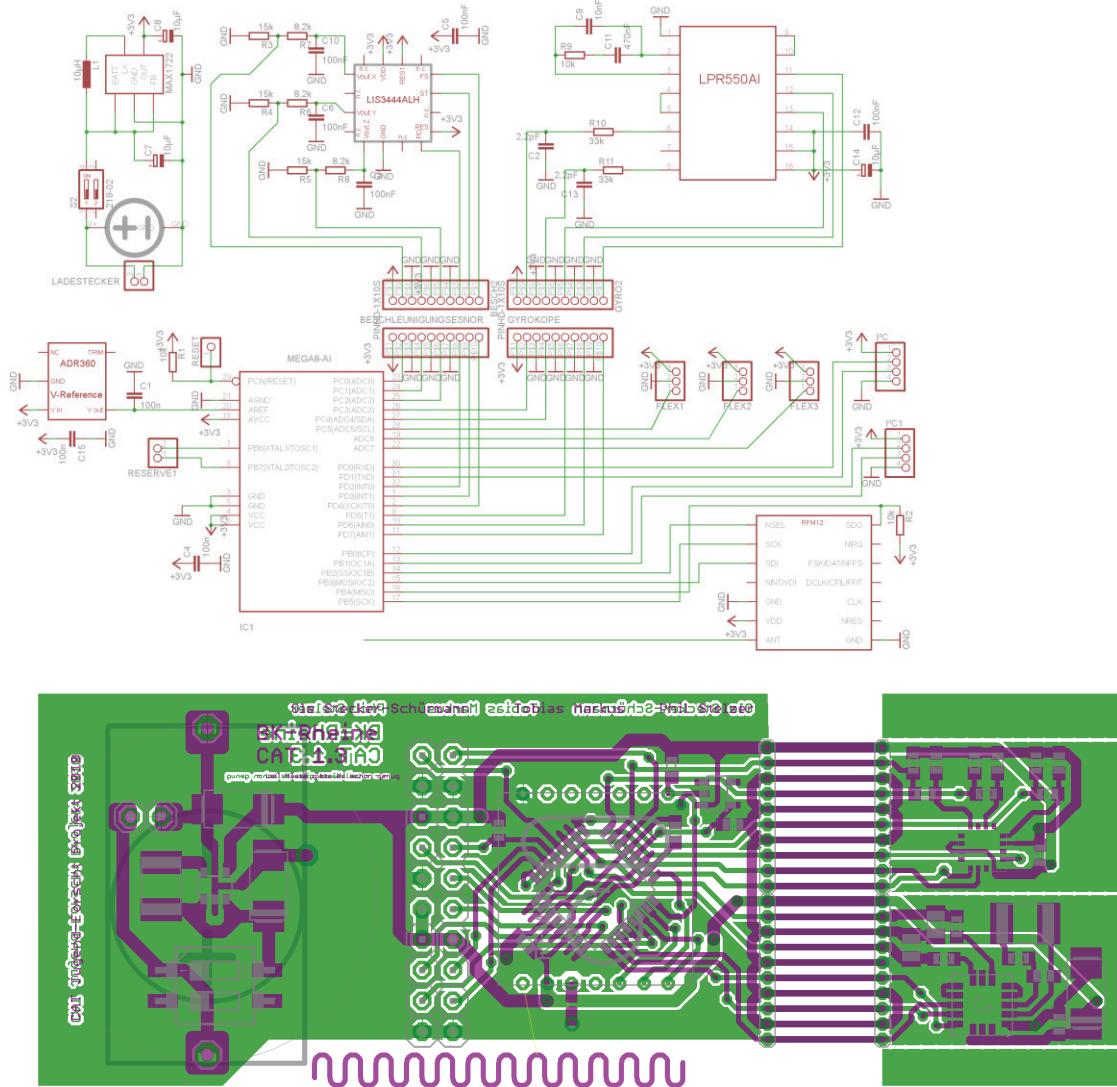
5.2.3. USB –Schnittstelle

Die USB- Kommunikation wird durch den FT232RL von FTDI Chips realisiert. Der IC simuliert am Computer eine virtuelle Serielle Schnittstelle. Diese Schnittstelle lässt sich nun ohne Probleme in C# oder mit anderen Programmiersprachen auslesen. Die Spannungsversorgung der Schaltung wird über den USB- Port bereitgestellt.

6. Schaltplan und Layout

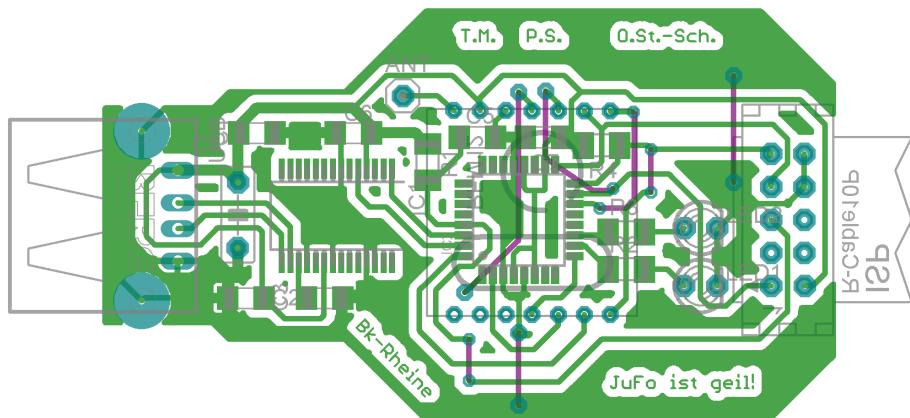
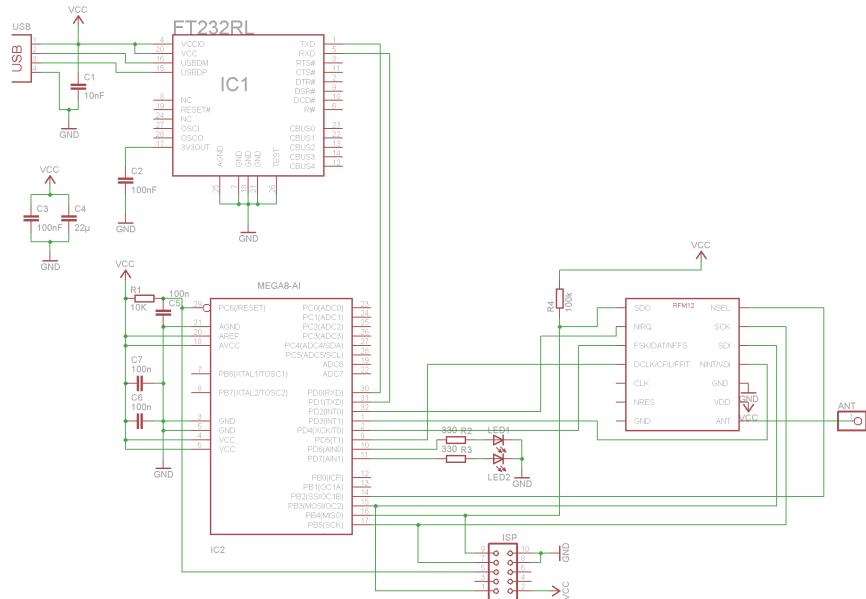
Die Schaltungen und Layouts wurden von uns mit der CAD Software Eagle (www.cadsoft.com) erstellt:

6.1. Handsteuerung



Schaltplan und Layout/Bestückung Handinterface

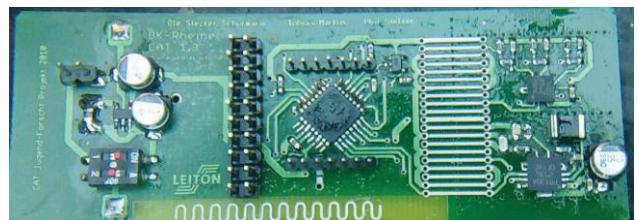
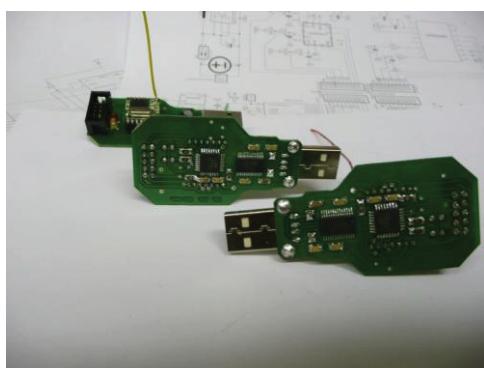
6.2. Empfänger



Schaltplan und Layout/Bestückung Empfänger Platine

6.3. Bilder der fertigen Platinen

Da wir die Grenzen des möglichen unserer Ätzanlage in der Schule mit diesem Projekt erreicht haben wurden die Platinen zum Fertigen an die Firma LeitOn rausgegeben und anschließend bestückt.



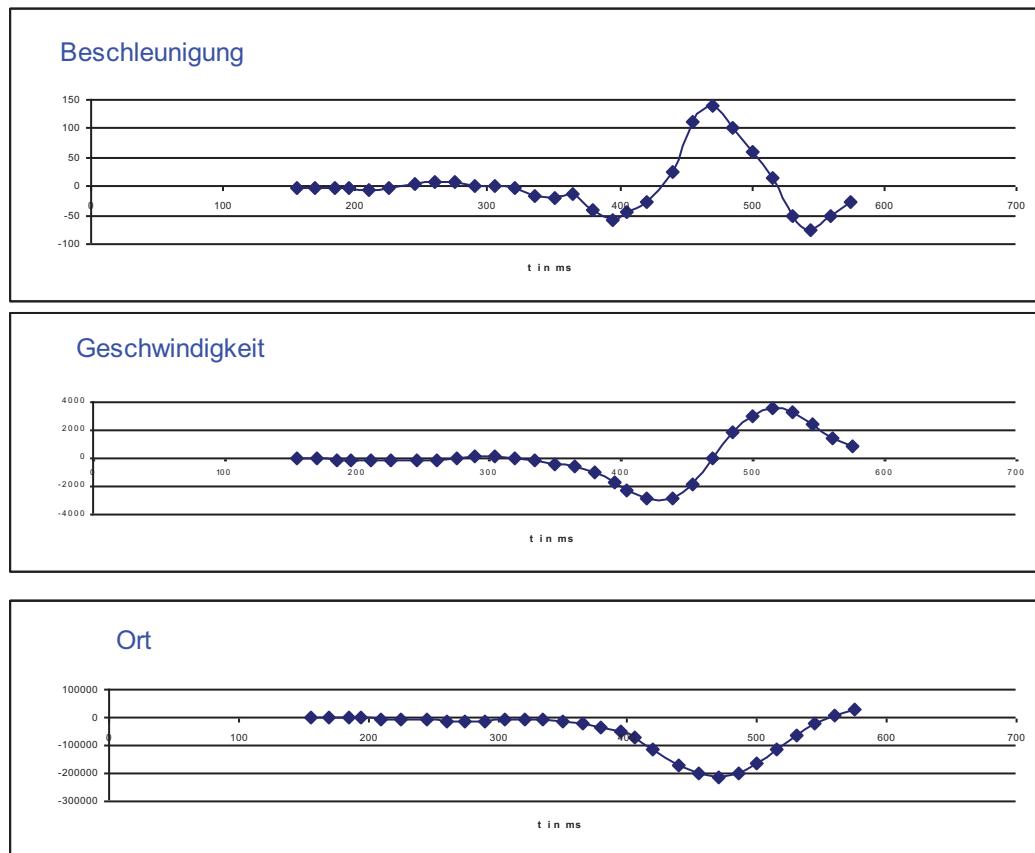
7. Physikalischer Hintergrund

7.1. Winkelbestimmung

Um den aktuellen Winkel des Handinterface zu bestimmen wird 3 Achsen Beschleunigungssensor genutzt. Durch das Verhältnis der Erdbeschleunigung auf den Achsen des Sensors lassen sich Roll und Nick Winkel des Handinterface bestimmen. Die Winkelgeschwindigkeiten werden anschließend genutzt um den Winkel von den Beschleunigungen entlang der Achsen zu isolieren. Zum fusionieren beider Sensoren und zum Filtern der Beschleunigungswerte aus Winkelveränderungen haben wir einen Kalmanfilter implementiert. Der Kalmanfilter gehört zu den Schätzwertfiltern und errechnet durch stochastische und statistische Mittel zukünftige Werte und korrigiert so die gemessenen Werte. Eine Winkelbestimmung nur mit den Gyroskopen durch einfaches Integrieren der Winkelgeschwindigkeiten ist aufgrund des Temperaturdrifts der Sensoren nicht möglich.

7.2. Positionsbestimmung

Die Positionsbestimmung durch die Beschleunigungssensoren erfolgt durch zweimaliges Integrieren der Beschleunigungswerte.



Zweimalige Integration einer Messreihe unserer Sensoren die Werte sind nicht skaliert und proportional zur eigentlichen physikalischen Größe

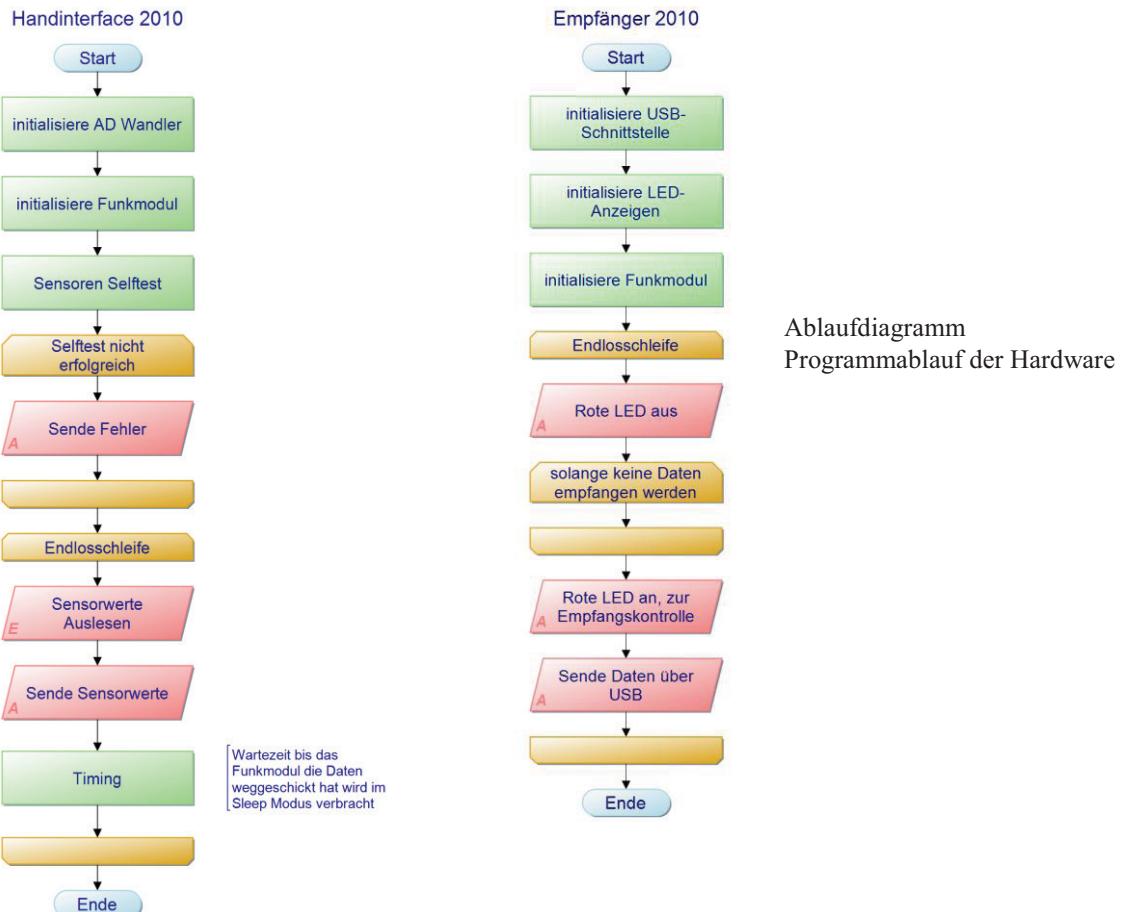
$$\frac{a_{t+1} + a_t}{2} \cdot \Delta t + V_{t-1} = V_t \quad \frac{V_{t+1} + V_t}{2} \cdot \Delta t + s_{t-1} = s_t$$

Die Annäherung des Integrals wurde mit dem Trapez Verfahren durchgeführt.

8.1. Handinterface und Empfänger

Die Mikrocontroller der Hardware wurden von uns in C programmiert.
Als Entwicklungsumgebung wurde AVR- Studio von Atmel verwendet.

8.1.1. Programmablaufplan



8.1.2. Beschreibung

Beim Einschalten der Handsteuerung werden die AD Wandler und das Funkmodul konfiguriert. Anschließend erfolgt die Einstellung der Sensoren und ein kleiner Selbsttest zur Funktionskontrolle. Bei Fehlern wird ein kurzer Bericht an den Empfänger gesendet.

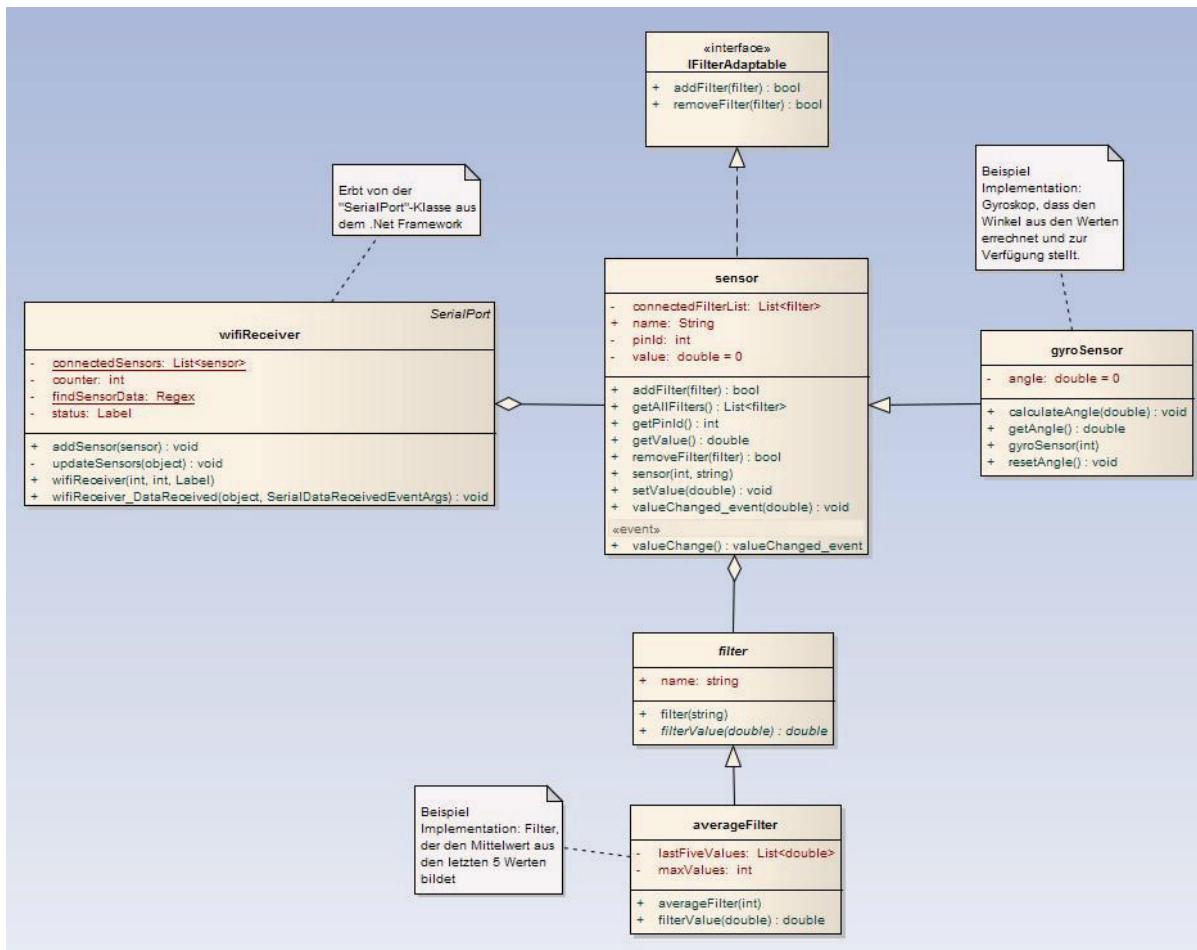
Nun beginnt sich das Programm in eine Endlosschleife, in welcher zuerst die Sensorwerte ausgelesen werden und an das Funkmodul weitergereicht werden.

Bis das Funkmodul die Werte aus dem Buffer gesendet hat, muss der Controller eine gewisse Zeit warten. In diesem Zeitraum, in dem der Controller nicht arbeiten muss, wird er von uns in einen Sleep Modus geschickt. Hierdurch kann der Stromverbrauch des Controllers, während dieser Zeit, auf 1/3 gesenkt werden.

8.2. Anwendung

Alle Anwendungen werden von uns in C# programmiert. Als Programmierumgebung nutzen wir Visual Studio 2008.

8.2.1. UML – Diagramm



UML- Klassendiagramm
Zusammenhang und Kommunikation der Klassen

8.2.2. Beschreibung

Die Anwenderseite unserer CAT-Handsteuerung wird durch Klassen bereitgestellt, die den einfachen Zugriff auf die verschiedenen einsteckbaren Module der Handsteuerung zulassen.

Die Klasse "wifiReceiver" hat die Hauptaufgabe den an kommenden Datastream des Serial Ports zu entschlüsseln und die geschickten Daten an die Sensoren zu übergeben, die vorher als Objekte einer Liste im "wifiReceiver" hinzugefügt wurden.

Die vorher erwähnten einsteckbaren Module werden auch als Klassen abstrahiert. Das "concreteHandModule" holt sich die benutzten Ports vom "wifiReceiver" und interpretiert die Werte, die an den Sensoren anliegen.

Dadurch ist es möglich z.B. ein Event bei einem Buttonklick auszulösen oder die aktuelle Position der Hand zu bestimmen.

Ein Programmierer, der Anwendungen für das Handinterface schreibt, muss sich durch diese Modellierung nicht um die Entschlüsselung der ankommenen Daten kümmern sondern nur wissen, welches Handmodul eingesteckt ist und die entsprechende Klasse instanzieren.

8.2.3. Testprogramme

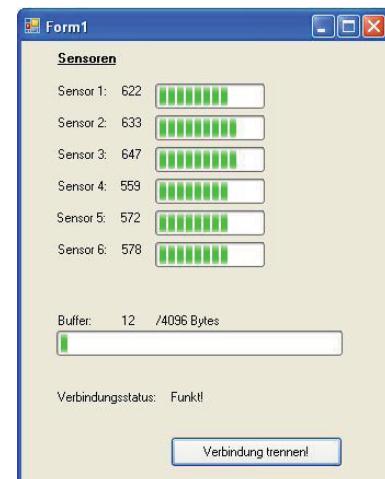
Um die Grundfunktionen des Systems auszuprobieren haben wir ein erstes Testprogramm erstellt.

Das Programm liest Sensorwerte aus und stellt die Verbindung zwischen Empfänger und Rechner her.

Die aktuellen Sensorwerte werden in einem Balken und als Wert dargestellt, so kann einfach abgeschätzt werden, ob die Aktualisierungsrate hinreichend hoch ist und ob die Sensoren richtig arbeiten. Hier ging es uns im Wesentlichen um die „gefühlte“ Mensch-Maschine-Interaktion. So können wir beurteilen, ob das System schnell genug interagiert.

Außerdem wird der Buffer der seriellen Schnittstelle angezeigt um zu überprüfen ob schnell genug gelesen wird.

In unserer zweiten Version des Testprogramms werden bereits alle Werte ausgelesen und die Winkel und Tastaturwerte visualisiert.

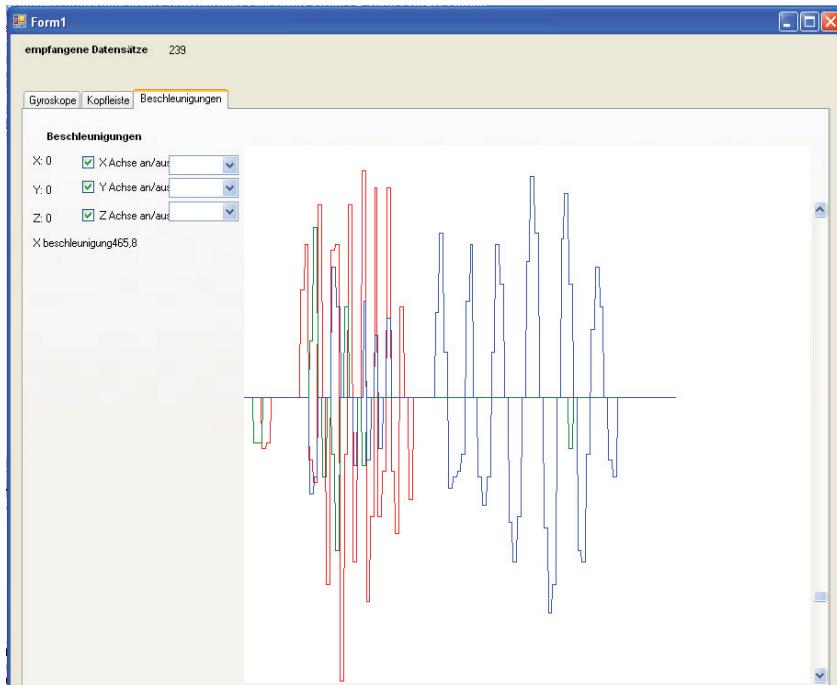


Testprogramm V0.1
Anzeige der Sensorwerte und Buffer

Testprogramm V1.0
Anzeige der Sensorwerte und Visualisierung der Gyroskope und der Tastatur

Um besser mit den Beschleunigungswerten arbeiten zu können, haben wir unser Testprogramm ein weiteres Mal umgeschrieben und diese in Form von Graphen visualisiert.

Mit der direkten Anzeige der Werte konnten wir die Aufbereitung der Sensorwerte weiter optimieren. Probleme waren Rauschen, die Erdbeschleunigung oder zu kleine Beschleunigungswerte.



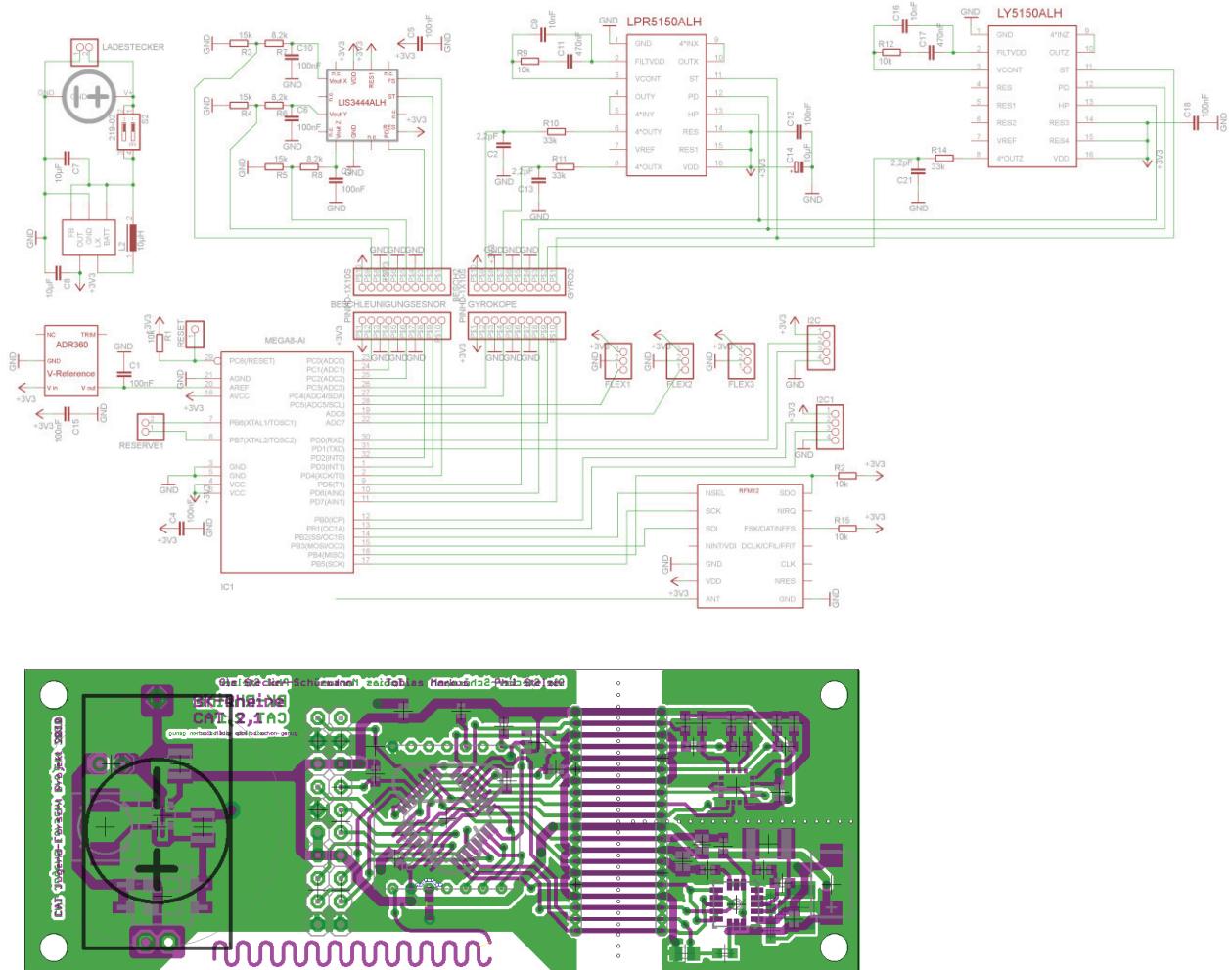
Testprogramm V1.1
Anzeige der Sensorwerte und
Visualisierung aller Funktionen
Hier gefilterte Beschleunigungen
im Diagramm

Da wir für unsere Anwendung keine absoluten Beschleunigungswerte benötigen, ermitteln wir nur relativ, auswertbare Daten. In diesen Daten stecken nur noch Informationen über die Beschleunigungsrichtung und die Beschleunigung in Relation zum Maximalwert (siehe Bild: Testprogramm V1.1).

9. Entwicklungen des Handinterface

In unserer ersten Version der Steuerplatine, konnten wir die wichtigsten Grundfunktionen in Betrieb nehmen, Testen, Optimieren und Zeigen. In den ersten Test wurde klar, dass Ergänzungen vorgenommen werden mussten. Folgende Punkte wurden in der zweiten Version ergänzt und optimiert:

- Die Gyroskope wurden durch Module (LPR5150AL) mit einem größeren Messbereich ersetzt
- Die fehlende Drehachse wird durch einen weiteren Sensor (LY5150AL) erfasst
- Das Knopfzellenpackage wurde angepasst.
- Die Schaltung des Spannungsreglers wurde verbessert
- Ein Pull-Up Widerstand am Funkmodul wurde hinzugefügt
- Bohrlöcher für Befestigungen sind hinzugekommen
- Ein Antennenanschluss wurde direkt an das Funkmodul geführt
- Kondensatoren und Widerstandspackages wurden angeglichen
- Die Integration durch das Trapezverfahren wurde durch das Runge Kutta Verfahren abgelöst



Neue Board und Schaltplan Version

10. Abgrenzungen zu ähnlichen Systemen

Mehrfach wurden wir darauf angesprochen das eine Ähnlichkeit zu dem „WiiController“ von Nintendo zu unserem System besteht. Daher scheint es uns wichtig eine Abgrenzung zu diesem System vorzunehmen. Ein wesentlicher Unterschied ist die grundsätzliche Anwendung des WiiControllers. Dieser wurde entwickelt als Eingabegerät für Spielumgebungen und ist ein vom Hersteller geschlossenes System. Unser System ist als universelles 3D Eingabegerät für verschiedene Anwendungen mit offenem Quellcode konzipiert.

Ein weiterer Unterschied ist die Sensorik, im WiiController arbeitet eine Infrarotkamera, welche die X/Y Koordinaten der vier wärmsten Punkte (Infrarot) im Sichtfeld ausgibt. Um die Zuverlässigkeit des Systems zu erhöhen werden zusätzliche Infrarotquellen benötigt.

Das CAT System arbeitet gänzlich ohne externen Orientierungspunkte.

Ein weiterer Unterschied ist die Handhabung der Geräte. Während der WiiController in die Hand genommen werden muss, wird das CAT System an der Hand getragen, die Finger bleiben weitestgehend frei.

11. Resümee

Alle Teilfunktionen wurden erfolgreich getestet. Für den Einsatz als 3D Eingabegerät sind grundsätzliche Steuerungsfunktionen funktionsfähig. Die Interpretation der Sensordaten muss weiter verfeinert werden. Testpersonen die das grundsätzliche Prinzip des intuitiven Eingabegerätes, eine Aktion des Benutzers verursacht eine direkte Reaktion des zu steuernden Systems, äußerten sich begeistert.

Der bisherige Entwicklungsstand zeigt, dass mit unserer Idee und deren Umsetzung eine Realisierung für ein universelles 3D Eingabegerät möglich ist.

12. Quellen

<http://de.wikipedia.org/wiki/Computermaus>
http://www.maxim-ic.com/quick_view2.cfm/qv_pk/3024
http://www.atmel.com/dyn/resources/prod_documents/doc2486.pdf
http://www.analog.com/static/imported-files/data_sheets/ADR360_361_363_364_365_366.pdf
<http://www.st.com/stonline/products/literature/ds/14337/lis344alh.pdf>
<http://www.st.com/stonline/products/families/sensors/datasheets/lpr550al.pdf>
<http://www.hoperf.com/pdf/rfm12.pdf>
<http://www.mikrocontroller.net/articles/RFM12>
<http://www.3dconnexion.de/>

13. Danksagungen

Wir danken der Firma LeitOn für die freundliche Unterstützung bei der Herstellung der Leiterplatten und Elektron Spelle für die Unterstützung bei der Bestückung der Platinen.
Außerdem Herrn Glaßmeyer, Herrn Petz und den Förderverein der Schule für die ideelle und finanzielle Unterstützung, sowie für die Bereitstellung verschiedener Ressourcen.
Herrn Sommer danken wir für die engagierte Projektbetreuung.

Für weitere Informationen:

<http://www.cat-projekt.de>