



Hochschule für Angewandte
Wissenschaften Hamburg
Hamburg University of Applied Sciences

HAW Hamburg
Fakultät Life Sciences
Studiengang Medizintechnik

Bachelor-Arbeit

Möglichkeiten der Lokalisation von Menschen
mittels einfacher Inertial-Sensorik

Dräger

Durchführung der Abschlussarbeit in der
Grundlagenentwicklung
Drägerwerk AG
Moislinger-Allee 53-55
23542 Lübeck

Dirk Dreffkorn

30. Juli 2012

Erstgutachter: Prof. Dr. B. Kellner
Zweitgutachter: Dr. S. Morley

Danksagung

Zunächst möchte ich mich an dieser Stelle bei Prof. Dr. Bernd Kellner und der Träger AG für die Ermöglichung und Betreuung meiner Bachelorarbeit bedanken.

Ganz besonderer Dank gilt Herrn Dr. Stefan Morley, der viel Zeit und Mühe investiert hat. Mit seinem Fachwissen und konstruktiven Antworten auf meine Fragen hat er zum Gelingen dieser Arbeit beigetragen.

Auch Robert Pilz, David Laule und Simon Ebert möchte ich danken für die aufmunternden und motivierenden Gespräche.

Ein besonderer Dank gilt meinem Vater und Birger Timm, die zahlreiche Stunden Korrektur gelesen haben.

Ein ganz herzlicher Dank geht an meine Frau, die mich in meiner geistigen Abwesenheit ertragen hat.

Nicht zuletzt gilt meiner Familie und meinen Freunden ein Dank für die emotionale Unterstützung während des gesamten Studiums.

Inhaltsverzeichnis

| | |
|---|-------------|
| Abbildungsverzeichnis | VIII |
| Tabellenverzeichnis | IX |
| Abkürzungsverzeichnis | XI |
| Symbolverzeichnis | XIII |
| 1. Einleitung | 1 |
| 2. Einführung in das Themengebiet | 3 |
| 2.1. Historische Hintergründe zur inertialen Navigation | 3 |
| 2.2. Heutiger Stand der Technik | 4 |
| 3. Grundlagen der inertialen Navigation | 7 |
| 3.1. Begriffserklärung Navigation und Lokalisation | 7 |
| 3.2. Bewegung im dreidimensionalen-Raum | 8 |
| 3.2.1. Translation | 9 |
| 3.2.2. Rotation | 15 |
| 3.3. Zentrifugalbeschleunigung | 16 |
| 3.4. Corioliskraft | 17 |
| 3.5. Gravitation | 18 |
| 3.6. Trägheit | 18 |
| 3.7. Inertiale Navigation | 18 |
| 3.8. Die Koordinatensysteme-Erdmodell des WGS 84 | 19 |
| 4. Mikro-Elektronische-Mechanische Systeme | 23 |
| 4.1. Beschleunigungssensoren | 25 |
| 4.2. Drehratensensoren | 30 |
| 5. Das Arduino-Mikrocontrollerboard | 35 |
| 5.1. Das Entwicklungsboard | 36 |
| 5.2. Die Entwicklungsumgebung | 40 |

| | |
|---|-----------|
| 6. Entwicklung eines Demonstrators zur Lokalisation mit Inertialsensoren | 45 |
| 6.1. Systemarchitektur | 46 |
| 6.2. Arduino Mikrocontrollerboard | 47 |
| 6.3. Datenübertragung mittels Bluetooth | 47 |
| 6.4. Beschleunigungssensor ADXL335 | 48 |
| 6.5. Inertial Measurement Unit MPU6050 | 51 |
| 7. Berechnungsalgorithmus zur Lokalisation mittels Euler-Winkel | 57 |
| 7.1. Lagebestimmung | 58 |
| 7.2. Ausgleich der Gravitationswerte | 62 |
| 7.3. Berechnung des zurückgelegten Weges | 63 |
| 8. Testaufbau zur Bestimmung der Position | 67 |
| 8.1. Entwicklung des Testaufbaus | 67 |
| 8.2. Testdurchführung mit dem Sensorsystem MPU6050 | 72 |
| 9. Ergebnisse | 73 |
| 10. Zusammenfassung | 81 |
| Literaturverzeichnis | 83 |
| Anhang | 89 |
| A. World Geodetic System 1984 Manuel | 89 |
| B. Schematisches Layout Arduino Duemilanove | 90 |
| C. Schematisches Layout Arduino Uno Rev3 | 91 |
| D. Datenblatt ATmega328P | 92 |
| E. Datenblatt ATmega8U2 | 93 |
| F. Datenblatt FTDI FT232RL | 94 |
| G. Schematisches Layout Bluetooth-Modul BlueSMIRF | 95 |
| H. Datenblatt Bluetooth-Modul BlueSMIRF | 96 |
| I. Datenblatt Beschleunigungssensor ADXL335 | 97 |
| J. Datenblatt Beschleunigungs-Drehratensensor MPU6050 | 98 |

| | |
|--|-----|
| K. Quelltext Beispielprogramm MPU6050-Rohdaten | 99 |
| L. Datenblatt Labornetzgerät Toellner Toe 8733 | 100 |
| M. Datenblatt Oszilloskop Yokogawa DL 1540 | 101 |
| N. Datenblatt Lichtschranke Panasonic PM-L24 | 102 |
| O. Datenblatt Gleichstrommotor MFA-Como-Drills-918D30112 | 103 |

Abbildungsverzeichnis

| | |
|--|----|
| 3.1. Bewegung im dreidimensionalen Raum [4] | 8 |
| 3.2. Richtungswinkel eines Vektors[5] | 9 |
| 3.3. Ort-Zeit-Diagramm mit unterschiedlichen Verschiebungen | 10 |
| 3.4. Beschleunigung-Geschwindigkeit-Weg-Diagramm | 12 |
| 3.5. Beschleunigung-Diagramm mit linearer Änderung von $a(t)$ | 14 |
| 3.6. Schematische Darstellung der Zentrifugalbeschleunigung | 16 |
| 3.7. Grafische Darstellung zur Coriolis-Kraft[12] | 17 |
| 3.8. Koordinatensystem für die erdnahe Navigation[7] | 21 |
| | |
| 4.1. Signalverarbeitungsrichtung im Sensorsystem | 24 |
| 4.2. Physikalisches Prinzip des MEMS-Beschleunigungssensor [12] | 26 |
| 4.3. Elektrisches Prinzip des MEMS-Beschleunigungssensor [12] | 28 |
| 4.4. Funktionsprinzip der Corioliskraft im Inertialsystem[12] | 31 |
| 4.5. Funktionsprinzip eines MEMS-Drehratensensor[17] | 32 |
| | |
| 5.1. Der Arduino Uno [32] | 36 |
| 5.2. Die Arduino Entwicklungsumgebung mit dem Beispiel <i>Blink</i> | 40 |
| 5.3. Die Schnellstartleiste Arduino | 41 |
| 5.4. Das Arduino Symbol- <i>Verify</i> | 41 |
| 5.5. Das Arduino Symbol- <i>Upload</i> | 42 |
| 5.6. Das Arduino Symbol- <i>New</i> | 42 |
| 5.7. Das Arduino Symbol- <i>Open</i> | 42 |
| 5.8. Das Arduino Symbol- <i>Save</i> | 42 |
| 5.9. Das Arduino Symbol- <i>Serial Monitor</i> | 43 |
| 5.10. Die Arduino-Informationsleiste | 43 |
| | |
| 6.1. Schema des Demonstrators mit den Systemkomponenten | 46 |
| 6.2. Das verwendete Bluetooth Modul BlueSMIRF [35] | 47 |
| 6.3. Funktionales Blockschaltbild des Beschleunigungssensors ADXL335 [38] | 49 |
| 6.4. Schaltplan des Demonstrator ADXL335-Arduino-Bluetooth | 49 |
| 6.5. Testprogramm des Beschleunigungssensor ADXLTEST1 | 50 |
| 6.6. Funktionales Blockschaltbild des Sensors MPU6050 [39] | 51 |
| 6.7. Bestückung der Leiterplatte des Sensors MPU6050 [39] | 52 |

| | |
|--|----|
| 6.8. Schema zur Pinbelegung und Bauteilbestückung des Sensors MPU6050 [39] | 52 |
| 6.9. Der fertige Demonstrator Arduino MPU6050 Bluetooth | 53 |
| 6.10. Der Demonstratorschaltplan: Arduino-MPU6050-Bluetooth | 54 |
| 6.11. Schema des Demonstrators in der Schiene | 54 |
| 7.1. Grafische Darstellung der Messkette | 57 |
| 7.2. Prozessschema des Berechnungsalgorithmus [7] | 58 |
| 8.1. Schematische Darstellung des Testaufbaus | 68 |
| 8.2. Diagramm zur Umdrehungsfrequenz des Laufrades | 70 |
| 8.3. Die Lichtschranke im Testaufbau | 71 |
| 8.4. Der komplette Testaufbau | 71 |
| 8.5. Der Demonstrator in der Schiene | 72 |
| 9.1. Diagramm ADXL335 Beschleunigung x-Achse | 74 |
| 9.2. Diagramm MPU6050 Beschleunigung y-Achse | 76 |
| 9.3. Diagramm Detektion der Erdbeschleunigung mit MPU6050 | 77 |
| 9.4. Diagramm Simulation Test des Berechnungsalgorithmus | 78 |
| 9.5. Diagramm Zentrifugalbeschleunigung auf y-Achse MPU6050 | 78 |
| 9.6. Diagramm Drehraten-Achsen x, y und z MPU6050 | 79 |

Tabellenverzeichnis

| | |
|---|----|
| 3.1. Parameter des Erdmodells WGS 84 [7] | 19 |
| 5.1. Arduino Optionen | 37 |
| 6.1. Die Pinbelegung des Sensors MPU6050 | 53 |
| 8.1. Hauptkomponenten des Testaufbau | 68 |
| 8.2. Umdrehungsfrequenz des Laufrades im Testaufbau | 69 |
| 9.1. Vergleichsparameter ADXL335 MPU6050 [38][39] | 73 |

Abkürzungsverzeichnis

| | |
|------------------------|--|
| AREF | Analog Referenz |
| C;C++ | Programmiersprache |
| CC | Creative Commons |
| COM-Port | Serielle Schnittstelle |
| ECEF | Earth Centred, Earth Fixed |
| EGM 96 | Earth Gravity Field Modell |
| FTDI | Future Technology Device International |
| GPS | Global Positioning System |
| I ² C | Inter-Integrated Circuit |
| ICSP | In-Circuit Serial Programming |
| IDE | Integrated Development Environment |
| MEMS | Mikro Elektronische Mechanische Systeme |
| MISO | Master in, Slave out |
| MIT MediaLab | Massachusetts Institute of Technology Media Laboratory |
| MOSI | Master out, Slave in |
| OFDM | Orthogonal Frequency Division Multiplexing |
| PPL | Precision Personnel Locator |
| PWM | Pulsweitenmodulation |
| RX | Receive |
| SCK | Serial Clock |
| SPI | Serial Peripheral Interface |
| SS | Slave Select |
| TWI | Two-Wire-Interface |
| TX | Transmit |
| WGS 84 | World Geodetic System |
| Wiring | Programmiersprache |
| WPI | Worcester Polytechnik Institute |

Symbolverzeichnis

| | |
|---------------------------------------|--|
| | |
| α | Richtungswinkel der Achse y |
| β | Richtungswinkel der Achse x |
| Δv | Geschwindigkeitsänderung |
| Δs | Ortsveränderung-Weg |
| Δt | Zeitänderung |
| γ | Richtungswinkel der Achse z |
| λ | Dämpfungskonstante |
| $\lim_{\Delta t \rightarrow 0}$ | Grenzwert-Zeitänderung geht gegen Null |
| Ω | Drehrate |
| ω | Winkelgeschwindigkeit, bzw. Drehrate |
| $\Omega \vec{e}_y$ | Winkelgeschwindigkeit um die y-Achse mit Einheitsvektor |
| ω_{r0} | Resonanzfrequenz |
| Ω_{z1} | Winkelgeschwindigkeit um die z-Achse im Bezugssystem I_1 |
| ϕ | Roll-Winkel Drehung um die x-Achse |
| ψ | Yaw-Winkel Drehung um die z-Achse |
| \sum | Summe |
| θ | Pitch-Winkel Drehung um die y-Achse |
| ε | Dielektrizitätskonstante |
| ε_0 | absolute Dielektrizitätskonstante |
| ε_r | relative Dielektrizitätskonstante |
| φ | Winkel (allgemein) |
| $\vec{0}$ | Nullvektor |
| \vec{a} | Vektor mit den Komponenten x, y und z |
| \vec{v} | Geschwindigkeitsvektor |
| \vec{e}_x | Einheitsvektor der Achse x |
| \vec{e}_y | Einheitsvektor der Achse y |
| \vec{e}_z | Einheitsvektor der Achse z |
| $\vec{\omega}$ | Winkelgeschwindigkeitsvektor |
| \vec{a}_c | Coriolis-Beschleunigung |
| \vec{a}_m | gemessener Beschleunigungsvektor mit den Komponenten a_{mx} , a_{my} und a_{mz} |
| \vec{a}_{FC} | Vektor der Coriolis-Beschleunigung |
| \vec{e} | Einheitsvektor |

Symbolverzeichnis

| | | |
|-------------|-------|--|
| \vec{F}_D | | Vektor der Dämpfungskraft |
| \vec{F}_f | | Vektor der Federkraft |
| \vec{F}_G | | Vektor der Gravitationskraft |
| \vec{g} | | Erdbeschleunigungsvektor |
| \vec{g}_m | | gemessener Erdbeschleunigungsvektor |
| \vec{l} | | Vektor der Auslenkung einer Prüfmass |
| \vec{s} | | Vektor des Weges mit den Komponenten s_x , s_y und s_z |
| \vec{v} | | Vektor der Geschwindigkeit mit den Komponenten v_x , v_y und v_z |
| A | | Plattenfläche des Kondensators |
| $a(t)$ | | Funktion-Beschleunigung in Abhängigkeit der Zeit |
| a_G | | Gravitationsbeschleunigung |
| a_x | | x-Komponente des Beschleunigungswertes |
| a_y | | y-Komponente des Beschleunigungswertes |
| a_z | | z-Komponente des Beschleunigungswertes |
| a_{mx} | | x-Komponente des gemessenen Beschleunigungswertes |
| a_{my} | | y-Komponente des gemessenen Beschleunigungswertes |
| a_{mz} | | z-Komponente des gemessenen Beschleunigungswertes |
| a_{rot} | | Winkelbeschleunigung |
| a_{trans} | | lineare Beschleunigung |
| a_{zf} | | Zentrifugalbeschleunigung |
| C | | Kapazität eines Kondensators |
| C_1 | | Kapazität des 1.Kondensators |
| C_2 | | Kapazität des 2.Kondensators |
| d | | Plattenabstand des Kondensators |
| D_G | | Gesamtdrehmatrix um die Achsen z, y und x |
| D_x | | Drehmatrix um die x-Achse |
| D_y | | Drehmatrix um die y-Achse |
| D_z | | Drehmatrix um die z-Achse |
| F_a | | Beschleunigungskraft |
| F_C | | Corioliskraft |
| f_U | | Umdrehungsfrequenz |
| F_Z | | Zentrifugalkraft |
| F_{ZP} | | Zentripetalkraft |
| g_x | | x-Komponente des Erdbeschleunigungsvektor |
| g_y | | y-Komponente des Erdbeschleunigungsvektor |
| g_z | | z-Komponente des Erdbeschleunigungsvektor |
| g_{mx} | | gemessene x-Komponente des Erdbeschleunigungsvektor |
| g_{my} | | gemessene y-Komponente des Erdbeschleunigungsvektor |
| g_{mz} | | gemessene z-Komponente des Erdbeschleunigungsvektor |
| I_1 | | Bezugssystem I_1 |

| | |
|-----------------|---|
| K_B | Faktor für die Umrechnung der Beschleunigung-Rohdaten |
| K_D | Faktor für die Umrechnung der Drehraten-Rohdaten |
| k_f | Federkonstante |
| l | Auslenkung einer Prüfmasse |
| m_{pk} | Masse des Prüfkörpers |
| P_A | Startpunkt-Ort s_A zur Zeit t_A |
| P_E | Endpunkt-Ort s_E zur Zeit t_E |
| Q | Elektrische Ladung |
| r | Radius eines Kreises (allgemein) |
| r_{Ab} | Radius vom Abtrieb |
| r_{An} | Radius vom Antrieb |
| $s(t)$ | Funktion-Weg in Abhängigkeit der Zeit |
| s_0 | Anfangsweg |
| s_A | Anfangsort-Weg |
| s_E | Endpunkt-Weg |
| s_i | Einzelverschiebung-Weg |
| S_{Abw} | Standardabweichung |
| t_A | Anfangszeit |
| t_E | Endzeit |
| T_U | Umdrehungszeit |
| U | Elektrische Spannung |
| U_A | Ausgangsspannung |
| U_B | Betriebsspannung |
| U_S | Speisespannung |
| $v(t)$ | Funktion-Geschwindigkeit in Abhängigkeit der Zeit |
| v_0 | Anfangsgeschwindigkeit |
| v_{x1} | Geschwindigkeit in Richtung der x-Achse im Bezugssystem I_1 |
| x_1 | Koordinatenachse x im Bezugssystem I_1 |
| x_i | Messwert einer Einzelmessung |
| x_m | Mittelwert der Einzelmessungen |
| y_1 | Koordinatenachse y im Bezugssystem I_1 |
| z_1 | Koordinatenachse z im Bezugssystem I_1 |
| b | Hoch-Index für das körperfeste Koordinatensystem (body) |
| e | Hoch-Index für das erdfeste Koordinatensystem |
| i | Einzelmessung |
| i | Hoch-Index für das inertielle Koordinatensystem |
| m | Masse |
| n | Hoch-Index für das Navigationskoordinatensystem |
| q | Anzahl der Einzelmessungen |
| u | Übersetzungsverhältnis |
| x | Koordinatenachse |

Symbolverzeichnis

| | |
|-------------|---|
| x' | Koordinatenachse x nach einer Translation |
| x'' | Koordinatenachse x' nach einer Rotation |
| y | Koordinatenachse |
| y' | Koordinatenachse y nach einer Translation |
| y'' | Koordinatenachse y' nach einer Rotation |
| z | Koordinatenachse |
| z' | Koordinatenachse z nach einer Translation |
| z'' | Koordinatenachse z' nach einer Rotation |

1. Einleitung

Die Lokalisation von Menschen gewinnt immer mehr an Bedeutung. Wachsendes Interesse aus den Bereichen Sicherheitstechnik und Medizintechnik führt zunehmend zur Forschung und Entwicklung an geeigneten Systemen, um eine Position eines Menschen mit hoher Genauigkeit bestimmen zu können. Außerdem wird es als sinnvoll erachtet, zusätzlich eine Aussage über die Art der Bewegung treffen zu können, um Rückschlüsse auf den Zustand der Person zu ziehen.

Die Grundlagenentwicklung der Dräger AG, in der diese Arbeit geschrieben wurde, forscht im Bereich der Sensorik, um neue, verbesserte und weiterentwickelte Lösungen hervorzubringen. So entstand die Idee, Möglichkeiten der Lokalisation von Menschen mittels einfacher Inertial-Sensoren zu untersuchen. Mit dem Hintergrund der Sicherheitstechnik, können beispielsweise Feuerwehr-Rettungskräfte im Einsatz überwacht werden, um im Falle auftretender Schwierigkeiten zu erkennen, wo sich die Einsatzkraft befindet. Damit im Vorfeld eine Abschätzung über den Zustand der Person getroffen werden kann, ist eine Bewegungsmusteranalyse vorteilhaft. Hinsichtlich der Faktoren Lokalisation und Bewegungsmustererkennung lassen sich weitere Einsatzmöglichkeiten ableiten. So können unter dem Aspekt der Medizintechnik auch Menschen, beispielsweise Patienten mit Orientierungsstörungen aufgefunden werden und eine Abschätzung ihres körperlichen Zustandes anhand der Bewegungsmustererkennung erfolgen.

Bislang auf dem Markt befindliche Systeme, wie GPS (Global Positioning System) weisen in vorgesehenen Einsatzgebieten Einschränkungen auf. So haben diese besonders unter Tage, in geschlossenen Gebäuden und eng bebauten Gebieten Signalausfälle. Da aber gerade in diesen Bereichen die Lokalisation von Menschen in Hinblick auf die eben genannten Einsatzmöglichkeiten wichtig ist, könnte mithilfe eines Inertial-Navigationssystems diese Nische geschlossen werden.

Das Ziel der vorliegenden Arbeit ist, die bei der Entwicklung auftretenden Probleme und Schwierigkeiten und die Einsatzmöglichkeiten eines Inertial-Navigationssystems zu untersuchen. So soll mit möglichst einfachen Mitteln ein System entwickelt werden, mit dem eine Positionsbestimmung und eine Bewegungsmustererkennung einer Person mit geringst möglichen Abweichungen durchgeführt werden kann. Dabei zählt zu den Hauptaufgaben ein mit Inertial-Sensoren bestücktes Steu-

erelement, einen Demonstrator aufzubauen, in Betrieb zu nehmen und zu testen. Für die Testung soll ein Teststand entwickelt werden, mit dem einzelne Parameter der verwendeten Sensoren und eine Positionsbestimmung überprüft werden kann. Die Entwicklung des Demonstrators beinhaltet die Auswahl geeigneter Software- und Hardwarekomponenten. Der Demonstrator soll die Möglichkeiten einer Positionsbestimmung, als auch einer Auswertung von Bewegungsmustern bieten. Das Gesamtkonzept soll dazu befähigt sein, unterschiedliche Sensoren anzuschließen. Ebenso soll durch die Auswahl von Hilfsmitteln eine Implementierung verschiedener Algorithmen zur Berechnung der Position erlaubt sein.

Die vorliegende Arbeit gliedert sich in zehn Kapitel. Dabei soll im folgenden Kapitel 2 eine Einführung in das Themengebiet gegeben werden. Im Kapitel 3 werden die für diese Arbeit grundlegenden Bedingungen erläutert. Das Kapitel 4 beinhaltet Erläuterungen zur Funktionsweise der verwendeten Sensoren, während das Kapitel 5 das ausgewählte Mikrocontroller-Board beschreibt. In Kapitel 6 ist die Entwicklung des Demonstrators aufgeführt. Für die Berechnung der Position ist in Kapitel 7 ein Algorithmus hergeleitet worden. Im achten Kapitel ist die Entwicklung des Teststandes und die erdachte Vorgehensweise für eine Testung beschrieben. Die abschließenden zwei Kapitel beinhalten die Ergebnisse und eine Zusammenfassung dieser Arbeit.

2. Einführung in das Themengebiet

Die ersten Anwendungen von Navigation waren bereits in frühesten Zeiten in der Seefahrt zu verzeichnen. So haben Menschen versucht sich durch Maßnahmen zur Bestimmung der Position, zur Berechnung zurückgelegter Wegstrecken und zur Richtungsbestimmung auf einem Kurs auf den Meeren zurecht zu finden. Damalige Systeme bestanden aus der Kombination mehrerer Verfahren miteinander. Diese Kombination wird auch unter dem Begriff Koppelnavigation geführt. Damals wurde die zurückgelegte Strecke mit einem Log und die Richtung mit einem Kompass festgelegt. Zusätzlich wurde aufgrund der sich aufsummierenden Fehler anhand dem Sonnenstand mittels einem Sextant der Breitengrad bestimmt. Heutige Systeme bestehen auch aus Koppelnaviationsverfahren. So wird beispielsweise bei der Inertialen Navigation eine Kombination aus Lagebestimmung, vergangener Zeit und zurückgelegter Wegstrecke in dieser Zeit vorgenommen. [7]

2.1. Historische Hintergründe zur inertialen Navigation

Erstmals wurde Anfang des zwanzigsten Jahrhunderts versucht anhand dem Trägheitsgesetz zu navigieren. Das erste Patent zum Prinzip der inertialen Navigation wurde 1910 beschrieben. So waren die Deutschen zu Zeiten des zweiten Weltkrieges mit die Ersten, die ein funktionierendes Navigationssystem entwickelten, welches einem heutigen Inertialnavigationssystem nahekommt. Dieses System wurde in der V2-Rakete zur Kurskorrektur eingesetzt. Die Raketen waren damals nicht in der Lage die Beschleunigung zu bestimmen, aber mittels Gyroskopen hatten sie die Möglichkeit Kursveränderungen zu detektieren und mit Servomotoren, die die Ruder steuerten den Kurs zu korrigieren.

Mitte des zwanzigsten Jahrhunderts gewann dieses Verfahren zur Navigation kontinuierlich an Bedeutung. Es sollte in der Verkehrstechnik zur Navigation von Flugzeugen, Schiffen und Automobilen zum Einsatz kommen. Anfangs waren es noch große kardanisch gelagerte Plattform-Systeme, auf denen die Beschleunigungssensoren montiert wurden. Ihre sensitiven Achsen wurden in die Richtungen Norden, Osten und unten ausgerichtet. Die verbauten Drehratensensoren detektierten eine Lageänderung, welche mittels Elektromotoren ausgeglichen wurde. So war die Plattform immer in der Nord-Ost-Unten-Ausrichtung und die Werte der Beschleunigungssensoren konnten durch Integration in Geschwindigkeitswerte und nochmalige Integration in Wegstrecken umgerechnet werden. Die Systeme wie sie heute

Anwendung finden, werden auch als *Strapdown-Systeme* bezeichnet, da sie fest mit dem zu lokalisierenden Objekt verbunden sind. Sie wurden erstmalig in den sechziger Jahren durch die Entwicklung von Ringlaserkreiseln möglich, bei ihnen geht die Lageänderung durch Integration der Drehrate in die Berechnung ein. Was eine kompliziertere Umrechnung in ein Koordinatensystem mit raumfesten Koordinatenachsen mit sich bringt. Mit der Weiterentwicklung der Mikrosystemtechnik entstanden am Ende des zwanzigsten Jahrhunderts neue Anwendungsgebiete. Mit den heutigen Sensoren aus Komponenten mit Abmessungen im Mikrometerbereich lässt sich die inertielle Navigation in ganz neue Anwendungsfelder, wie Medizintechnik, Sicherheitstechnik und der Computerindustrie erweitern. [7]

2.2. Heutiger Stand der Technik

Die Bestimmung der Position einer Person hat besonders im Bereich der Sicherheitstechnik in den letzten Jahren zu erhöhtem Forschungsaufwand geführt. Es wird gesucht, nach geeigneten Systemen, welche eine Positionsbestimmung in Gebäuden mit geringer Abweichung von der tatsächlichen Position möglich macht. In den USA im Jahr 1999 war ein Brand in einem Warenhaus in Worcester, bei dem sechs Feuerwehrrettungskräfte ihr Leben ließen, weil sie den Ausgang nicht fanden, der Auslöser für Forschungsaufrufe in diesem Bereich. So beschäftigt sich beispielsweise das WPI (Worcester Polytechnic Institute) seit einigen Jahren mit Forschungen zu dem Thema Positionsbestimmung von Einsatzkräften. Das WPI veranstaltet regelmäßig Workshops und Kongresse. Auf diesen Veranstaltungen stellen sie ihre Projektarbeiten und Forschungen vor, bieten aber auch externen Einrichtungen die Möglichkeit sich mit Beiträgen zu beteiligen. Mittlerweile existieren einige ernstzunehmende Projekte, welche sich aber für den Einsatzbereich der Feuerwehr spezialisiert haben. Beispielhaft ist das Projekt *PPL (Precision Personnel Locator)* [1] vom WPI. Dieses System basierend auf dem Prinzip der OFDM (Orthogonal Frequency Division Multiplexing) kommt dem Radarprinzip sehr nahe. Dabei müssen mindestens drei mit Sendern und Empfängern ausgestattete Fahrzeuge sich um ein Haus positionieren, um die Signalinformation aufzunehmen. Die mit Sendern bestückten Einsatzkräfte senden fortlaufend, während sie das Haus betreten Signale aus, welche von der Kommandozentrale aufgezeichnet und auf einem Bildschirm dargestellt werden. Die ausgesendeten Signale enthalten zusätzlich physiologische Informationen zur einzelnen Einsatzkraft.

Weitere Techniken basieren auf Ultraschallverfahren, bei denen die an Hauswänden reflektierenden Signale ausgewertet werden. Oder es wird ein elektromagnetisches Nahfeld erzeugt, bei dem die von einem Transmitter erzeugten Signale vermessen werden. [2]

Bei der inertialen Navigation sind die Einsatzgebiete überwiegend in der Verkehrstechnik, der Verfahrenstechnik, im Maschinenbau aber auch im Modellbau angesiedelt. Frühere Systeme, basierend auf Faserkreisel und Ringlaserkreisel bei den Drehratensensoren werden zunehmend von MEMS (Mikro-Elektronische-Mechanische Systeme) abgelöst, wobei die Genauigkeit bei den MEMS-Sensoren im Gegensatz zu anderen Techniken noch verbesserungswürdig ist. Die MEMS zeichnen sich aber durch ihre besonders kleinen Abmessungen, geringen Material- und Energieverbrauch und ihre durch Massenproduktion ständig fallenden Kosten aus. So wird bei Systemen, welche MEMS-Sensoren verwenden auch von *Low Cost*-Inertial-Navigationssystemen gesprochen. Bei der derzeitigen Entwicklung ist davon auszugehen, dass die Mikrosystemtechnik die Wandlerprinzipien erweitern und stark verbessern wird, so werden die Sensoren stetig genauer, kleiner, billiger und erhalten zusätzliche Komponenten und Funktionen. In der vorliegenden Arbeit wurden diese MEMS-Sensoren verwendet, welche in Kapitel 4 näher erläutert werden.

3. Grundlagen der inertialen Navigation

In diesem Kapitel sollen die Grundlagen für die im Verlauf der Arbeit dargestellten Bedingungen geschaffen werden. In späteren Kapiteln wird Bezug auf die hier dargelegten Grundlagen genommen.

3.1. Begriffserklärung Navigation und Lokalisation

- Navigation

In der vorliegenden Arbeit wird die Zustandsbestimmung durch ein inertiales Navigationssystem von Personen behandelt. Die inertielle Navigation hat ihre Wurzeln in der Fahrzeugüberwachung, laut der DIN 13312 ist die *Navigation* definiert als:

Maßnahme - Beobachtungen, Messung und Auswertungsmethoden - zur Fahrzeugführung, mit deren Hilfe ermittelt wird,

- 1. wo sich das Fahrzeug befindet,*
- 2. wohin das Fahrzeug gelangen würde, wenn keine seine Bewegung verändernden Maßnahmen ergriffen werden und*
- 3. was zu tun ist, um ein gewünschtes Ziel sicher zu erreichen, gegebenenfalls auf einem vorgegebenen Weg*

Diese Definition lässt sich hier fast uneingeschränkt auf die Navigation einer Person übertragen, wobei sich die folgenden Untersuchungen auf das Ermitteln der Bewegung beschränken. Das Sensorsystem soll die Lage, die Geschwindigkeit und die Position einer Person erfassen. Diese Informationen bestimmt der erste Punkt in der Definition und wird auch als *Navigationslösung* bezeichnet. Die Punkte 2 und 3 werden nicht betrachtet, es geht in der vorliegenden Arbeit um die Ortung der Person. Die Behandlung dieser unter dem 1. Punkt gestellten Frage erfordert eine Fusion mehrerer Informationen, so wird nicht auf dem direkten Weg die Position erkannt. Über ein inertiales Navigationssystem kann nicht der zurückgelegte Weg gemessen werden, sondern er wird über Gleichungssysteme anhand dem Trägheitsgesetz berechnet.

- Lokalisation

Die Lokalisation (lat. locus Ort) bedeutet ursprünglich, den Ort eines Geschehens oder eines Objektes festzustellen oder einzugrenzen. Der Begriff wird vor allem in der Technik und der Medizin verwendet. [3] In der vorliegenden Arbeit wird unter dem Begriff *Lokalisation* sinngemäß der Sachverhalt in Bezug zur *Navigationslösung* verstanden. Dabei geht es um die Bestimmung des Ortes der Person.

3.2. Bewegung im dreidimensionalen-Raum

Die Bewegung im Raum lässt sich durch zwei Mechanismen darstellen, zum einen die Translation und zum anderen die Rotation. Die Abbildung 3.1 zeigt dies.

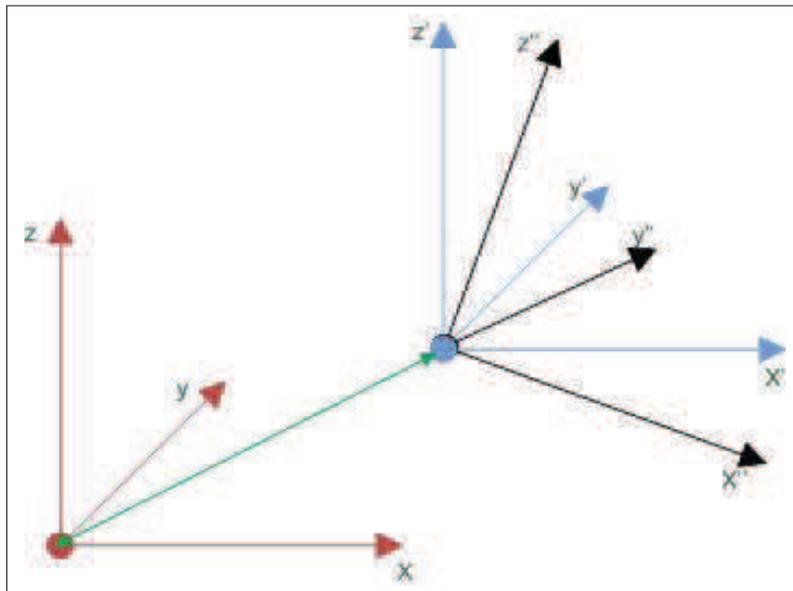


Abbildung 3.1.: Bewegung im dreidimensionalen Raum [4]

Es resultieren aus einer Bewegung insgesamt 6 Freiheitsgrade, also jeweils zwei um die drei Achsen x , y und z . Das rote Koordinatensystem ist der Ursprung, bevor eine Bewegung ausgelöst wurde. Der grüne Pfeil zeigt eine Translation, in das neu entstandenen blaue Koordinatensystem mit den Achsen x' , y' und z' . Das blaue Koordinatensystem wurde gedreht, und daraus ist das schwarze Koordinatensystem mit den Achsen x'' , y'' und z'' entstanden. Die Orientierung im Raum durch kartesische Koordinaten lässt sich mit dem Vektor \vec{a} anhand der Formel 3.1 darstellen, wobei die Einheitsvektoren \vec{e}_x , \vec{e}_y und \vec{e}_z als Basisvektoren dienen.

Komponentendarstellung des Vektors \vec{a} [5]:

$$\vec{a} = \vec{a}_x + \vec{a}_y + \vec{a}_z = a_x \cdot \vec{e}_x + a_y \cdot \vec{e}_y + a_z \cdot \vec{e}_z = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} \quad (3.1)$$

Die Abbildung 3.2 stellt die Richtungswinkel des Vektors \vec{a} grafisch dar.

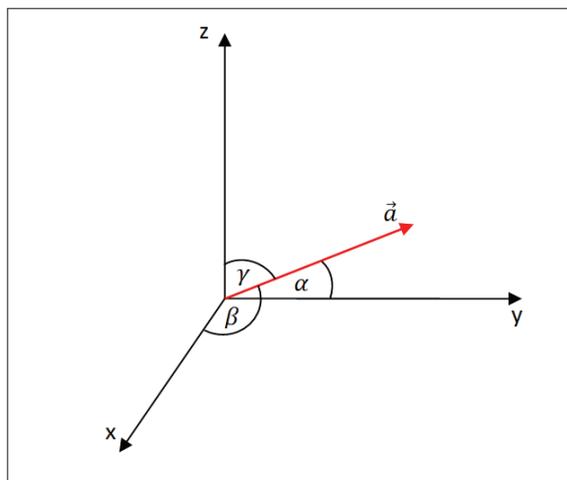


Abbildung 3.2.: Richtungswinkel eines Vektors[5]

Für die Richtungswinkel α , β und γ des Vektors $\vec{a} \neq 0$, auch Richtungskosinus genannt, gelten folgende, in den Gleichungen 3.2 und 3.3 dargestellten Beziehungen [5]:

$$\cos \beta = \frac{a_x}{|\vec{a}|}, \cos \alpha = \frac{a_y}{|\vec{a}|}, \cos \gamma = \frac{a_z}{|\vec{a}|} \quad (3.2)$$

Für den *Nullvektor* $\vec{0}$ lassen sich keine Richtungswinkel angeben. Aus dem Richtungskosinus und dem Betrag des Vektors lassen sich mit der Gleichung 3.2 auch umgekehrt die Vektorkoordinaten ausrechnen.

3.2.1. Translation

In diesem Abschnitt wird die Translation beschrieben, sie stellt die Grundlage für die Berechnungen in den folgenden Kapiteln dar. Die Translation, auch Verschiebung genannt, bezeichnet eine lineare Bewegung. Die Bewegung eines Gegenstandes führt vom Ort s_A zur Zeit t_A hin zum neuen Ort s_E zur Zeit t_E . Diese

3. Grundlagen der inertialen Navigation

Ortsveränderung Δs lässt sich durch Differenz aus Endwert und Anfangswert in einem bestimmten Zeitintervall Δt berechnen.

Ortsveränderung[6]:

$$\Delta s = s_E - s_A \quad (3.3)$$

Mit einer Zeitdauer[6]:

$$\Delta t = t_E - t_A \quad (3.4)$$

In Abhängigkeit vom Bezugssystem kann eine Bewegung in positiver oder negativer Richtung erfolgen. Die Verschiebung geschieht mit einer bestimmten Geschwindigkeit, beschrieben durch die Rate, mit der sich der Ort verändert. Das heißt aus dem Quotienten von Ortsverschiebung Δs und der Zeitdifferenz Δt . Diese bestimmte Geschwindigkeit beschreibt aber nur die mittlere Geschwindigkeit, mit der sich das Objekt vom Anfangspunkt s_A zum Endpunkt s_E bewegt und nicht die momentane Geschwindigkeit, welche sich durchaus ständig ändern kann. So gibt es verschiedene Möglichkeiten, einen Ort zu erreichen. Die Abbildung 3.3 zeigt dies.

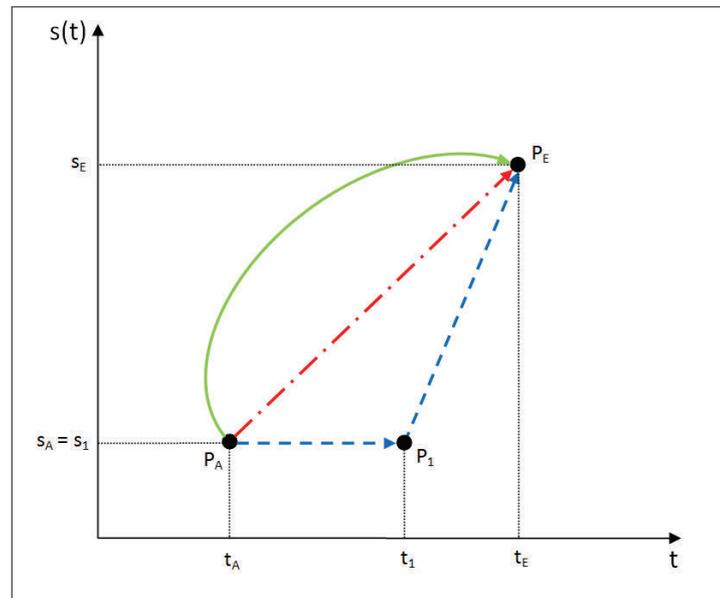


Abbildung 3.3.: Ort-Zeit-Diagramm mit unterschiedlichen Verschiebungen

Der Punkt P_E kann durch Bewegungen mit unterschiedlichen Geschwindigkeiten vom Punkt P_A aus erreicht werden. Gibt es mehrere Verschiebungen, dann setzt sich die Gesamtverschiebung aus der Summe \sum der Einzelverschiebungen s_i zusammen. So besteht zum Beispiel die Bewegung, welche durch den blauen Pfeil dargestellt ist, aus zwei einzelnen Verschiebungen.

Gesamtverschiebung aus einzelnen Teilverschiebungen[6]:

$$\Delta s = \sum_i \Delta s_i \quad (3.5)$$

Wie in Abbildung 3.3 verdeutlicht kann eine Ortsverschiebung unterschiedliche Geschwindigkeiten haben. Um auf die Momentan-Geschwindigkeit eines Objektes schließen zu können, müssen zwei Zeitpunkte bestimmt werden, deren Differenz so klein wie möglich ist. Es wird also der Grenzwert des Quotienten $\frac{\Delta s}{\Delta t}$ bestimmt. Dies entspricht dem Anstieg einer Tangente in dem Punkt, also der ersten Ableitung der Funktion $s(t)$.

Momentangeschwindigkeit $v(t)$ [6]:

$$v(t) = \lim_{\Delta t \rightarrow 0} \frac{\Delta s}{\Delta t} = \frac{\delta s}{\delta t} = \dot{s} \quad (3.6)$$

Nochmaliges Differenzieren beschreibt die Änderung der Geschwindigkeit Δv in Bezug zur Zeit, dies ist die Beschleunigung $a(t)$. Es wird ebenfalls zwischen mittlerer- und momentaner Beschleunigung unterschieden.

Mittlere Beschleunigung[6]:

$$a(t) = \frac{\Delta v}{\Delta t} \quad (3.7)$$

Die Beschleunigung hat die Dimension einer Länge, dividiert durch das Quadrat einer Zeit. Die Momentanbeschleunigung ist der Grenzwert $\lim_{\Delta t \rightarrow 0}$ des Quotienten aus der Geschwindigkeitsänderung und der Zeitänderung, wenn die Zeitänderung gegen *Null* geht. In einem Geschwindigkeit-Zeit-Diagramm ist die Momentan-Beschleunigung zum Zeitpunkt t gleich dem Anstieg der Tangente an der Kurve in diesem Punkt. Somit ist die Beschleunigung die Ableitung der Geschwindigkeit nach der Zeit.

Momentanbeschleunigung[6]:

$$a(t) = \lim_{\Delta t \rightarrow 0} \frac{\Delta v}{\Delta t} = \frac{\delta v}{\delta t} = \dot{v} = \frac{\delta^2 s}{\delta t^2} = \ddot{s} \quad (3.8)$$

Bleibt die Beschleunigung null, herrscht keine Geschwindigkeitsänderung während der Zeit vor, die Geschwindigkeit ist für das Intervall konstant. Die Abbildung 3.4 zeigt ein Beschleunigung-Geschwindigkeit-Weg-Diagramm mit einer konstanten Beschleunigung, daran ist gut die Auswirkung der Beschleunigung über die Zeit auf den zurückgelegten Weg ersichtlich.

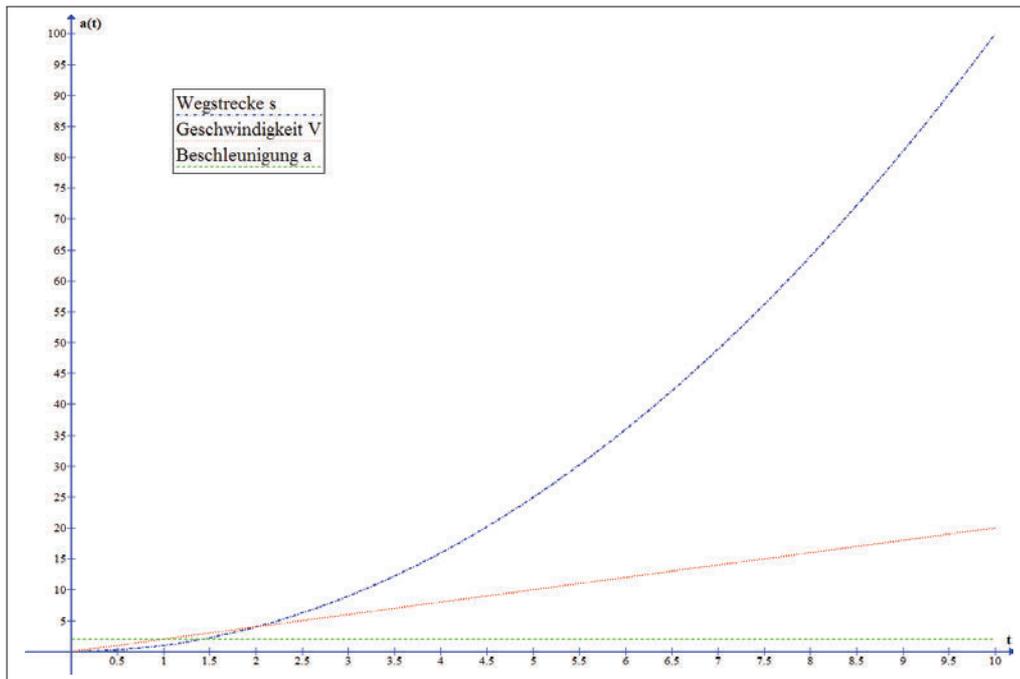


Abbildung 3.4.: Beschleunigung-Geschwindigkeit-Weg-Diagramm

Über die Beschleunigung kann wiederum der Weg errechnet werden. Über das Integrieren der Beschleunigung erhält man die Geschwindigkeit, und das nochmalige Integrieren den Weg. Allgemein gelten für die Berechnung des Weges, wenn $a(t)$ konstant ist, folgende Gleichungen. Dabei ist die Anfangsgeschwindigkeit v_0 zu berücksichtigen.

Für die Geschwindigkeit $v(t)$:

$$v(t) = \int a(t)dt + v_0 \quad (3.9)$$

Für die Wegstrecke $s(t)$ gilt dann:

$$s(t) = \int v(t)dt + s_0 \quad (3.10)$$

$$s(t) = \int (a_0 \cdot t + v_0)dt + s_0 \quad (3.11)$$

Daraus folgt für die Wegstrecke $s(t)$:

$$s(t) = \frac{1}{2} \cdot a_0 \cdot t^2 + v_0 \cdot t + s_0 \quad (3.12)$$

Für $v_0 = 0$ und $s_0 = 0$ folgt daraus:

$$s(t) = \frac{1}{2} \cdot a_0 \cdot t^2; \quad v_0 = 0; \quad s_0 = 0 \quad (3.13)$$

Ist die Beschleunigung nicht konstant ändert sich der Sachverhalt zu folgenden Gleichungen. Die Grafik in Abbildung 3.5 veranschaulicht die Darstellung der Gleichungen.

Nun geht die Änderung der Beschleunigung in die Berechnung mit ein.

Beschleunigungsänderung:

$$a(t) = a_0 + \frac{a_1 - a_0}{t_1 - t_0} \cdot (t - t_0) \quad (3.14)$$

Daraus resultiert für die Geschwindigkeit.

Geschwindigkeit bei linearer Beschleunigungsänderung:

$$v(t) = v_0 + \int_{t_0}^t a_0 + \frac{a_1 - a_0}{t_1 - t_0} \cdot (t - t_0)dt \quad (3.15)$$

Daraus folgt für die Geschwindigkeit:

$$v(t) = v_0 + a_0 \cdot t + \frac{1}{2} \cdot \frac{a_1 - a_0}{t_1} \cdot t^2 \quad (3.16)$$

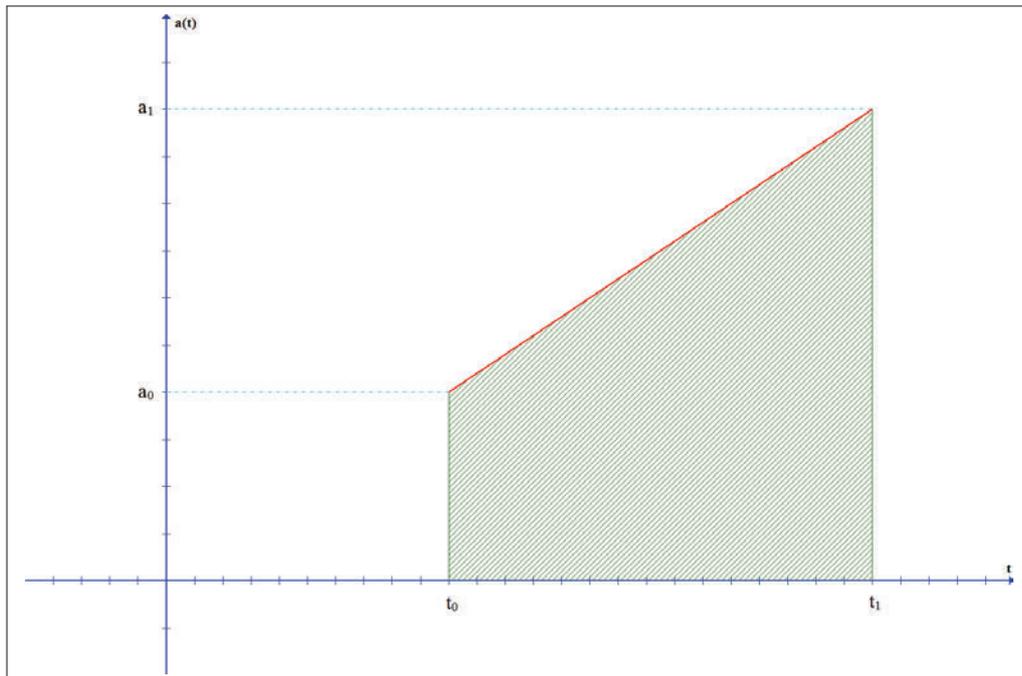


Abbildung 3.5.: Beschleunigung-Diagramm mit linearer Änderung von $a(t)$

Für die Wegstrecke folgt daraus.

Wegstrecke mit Beschleunigungsänderung:

$$s(t) = s_0 + \int_{t_0}^t a_0 \cdot t + \frac{1}{2} \cdot \frac{a_1 - a_0}{t_1} \cdot t^2 + v_0 dt \quad (3.17)$$

Das führt dann zu folgender Gleichung für die Wegstrecke.

Wegstrecke bei linearer Beschleunigungsänderung:

$$s(t) = s_0 + \frac{1}{2} a_0 \cdot t^2 + \frac{1}{6} \cdot \frac{(a_1 - a_0)}{t_1} \cdot t^3 + v_0 \cdot t \quad (3.18)$$

Die Gleichung 3.18 geht in folgenden Berechnungen in Kapitel 7 ein.

3.2.2. Rotation

Als Rotation wird eine Bewegung um eine Rotationsachse bezeichnet. Ein Objekt, was sich auf einer Kreisbahn mit dem Radius r vom Punkt P_A zum Punkt P_E eine Strecke $\delta s = s_E - s_A$ in der Zeit $\delta t = t_E - t_A$ bewegt, spannt während dieser Bewegung einen Winkel $\delta\varphi$ auf.

Für diesen Winkel gilt[6]:

$$\delta\varphi = \frac{\delta s}{r} \quad (3.19)$$

Dieser Winkel ist charakteristisch für den zurückgelegten Drehwinkel, welcher auch mit dem Bogenmaß bezeichnet wird. Eine vollständige Umdrehung, also 360° , entspricht einer zurückgelegten Wegstrecke:

$$s = 2 \cdot \pi \cdot r \quad (3.20)$$

Im Bogenmaß entspricht das:

$$360^\circ = \frac{2 \cdot \pi \cdot r}{r} = 2 \cdot \pi \quad (3.21)$$

Für die Periode p eines Umlaufs in der Zeit T folgt daraus[6]:

$$p = \frac{2 \cdot \pi r}{T} \quad (3.22)$$

Die Winkelgeschwindigkeit ω , bzw. Drehrate ist die Geschwindigkeit in der das Objekt in der Zeit $\delta t = t_E - t_A$ den Winkel $\delta\varphi$ durchlaufen hat.

Für die Winkelgeschwindigkeit gilt:

$$\omega = \frac{\delta\varphi}{\delta t} \quad (3.23)$$

Dabei ist zu beachten, dass dies die mittlere Winkelgeschwindigkeit darstellt. Wie auch bei der Translation muss bei der Art der Bewegung zwischen gleichförmiger Kreisbewegung mit konstanter Winkelgeschwindigkeit und beschleunigter Bewegung auf einer Umlaufbahn unterschieden werden.

Für die Winkelbeschleunigung a_{rot} gilt die allgemeine Gleichung[12]:

$$a_{rot} = \frac{\delta\omega}{\delta t} = \frac{\delta^2\varphi}{\delta t^2} = \ddot{\varphi} \quad (3.24)$$

3.3. Zentrifugalbeschleunigung

In einem späteren Kapitel wird die Zentrifugalbeschleunigung verwendet, um einen definierten Wert für die Überprüfung der Sensoren zu erhalten. Die Zentrifugalbeschleunigung a_{zf} bestimmt die Zentrifugalkraft F_Z , welche der Zentripetalkraft F_{ZP} entgegen gerichtet ist. Bewegt sich ein Objekt mit der Masse m auf einer Kreisbahn, so wirkt auf das Objekt nach außen die Zentrifugalkraft. Um durch sie nicht nach außen weggedrückt zu werden, muss eine vom Betrag her gleiche Kraft, die Zentripetalkraft, in entgegengesetzter Richtung das Objekt auf der Kreisbahn halten. Die Zentrifugalkraft ist proportional zum Radius r der Kreisbahn und zum Quadrat der Winkelgeschwindigkeit ω .

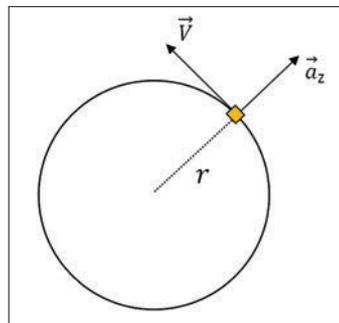


Abbildung 3.6.: Schematische Darstellung der Zentrifugalbeschleunigung

Zentripetalkraft[6]:

$$F_{ZP} = -m \cdot \omega^2 \cdot r \quad (3.25)$$

Nach dem dritten Newton'schen Axiom gilt[6]:

$$F_{ZP} = -F_Z \quad (3.26)$$

Zentrifugalkraft[6]:

$$F_Z = m \cdot \omega^2 \cdot r \quad (3.27)$$

Demnach gilt für die Beschleunigung:

$$a_{zf} = \omega^2 \cdot r \quad (3.28)$$

Die Abbildung 3.6 verdeutlicht die Richtung der Zentrifugalbeschleunigung. Diese Beschleunigung wird im späteren Verlauf in Kapitel 8 genutzt, um eine definierte Beschleunigung für die Tests zur Überprüfung der Sensoren als Eingangswert zu erhalten.

3.4. Corioliskraft

Um die Corioliskraft F_C zu beschreiben, muss ein Zustand definiert werden, welcher es erst möglich macht, diese *Scheinkraft* zu erklären. Die Abbildung 3.6 soll die Darstellung in Form eines bei Sensoren angewandten Prinzips vereinfachen. In diesem Zustand wird angenommen, das 2 Bezugssysteme existieren, ein in Ruhe befindliches System und ein um eine Achse rotierendes System. Bewegt sich ein Objekt mit der Masse m und der Geschwindigkeit \vec{v} (im bewegten Bezugssystem) in einem um die Achse w mit der Winkelgeschwindigkeit $\vec{\omega}$ rotierendes System, so wirkt die Coriolis-Kraft senkrecht zum Geschwindigkeitsvektor und entgegengesetzt zur Drehung des rotierenden Bezugssystems. In Bezug zum ruhenden System bewegt sich der Körper geradlinig.[6]

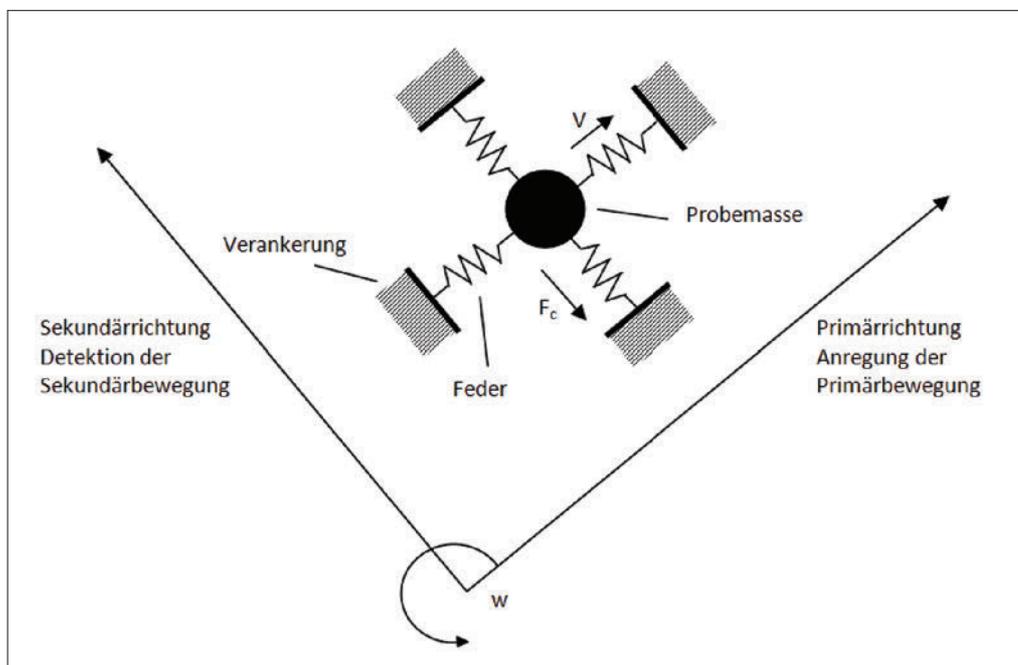


Abbildung 3.7.: Grafische Darstellung zur Coriolis-Kraft[12]

Dann ergibt sich für diesen Sachverhalt folgende Gleichung[12]:

$$\vec{F}_C = 2 \cdot m \cdot (\vec{v} \times \vec{\omega}) \quad (3.29)$$

Für die Beschleunigung \vec{a}_{F_C} ergibt sich daraus:

$$\vec{a}_{F_C} = 2 \cdot (\vec{v} \times \vec{\omega}) \quad (3.30)$$

3.5. Gravitation

Der Name Gravitation kommt aus dem Lateinischen und heißt Schwere. Als Gravitationskraft \vec{F}_G wird die Kraft bezeichnet, welche aus der gegenseitigen Anziehung zwischen den Massen von Körpern resultiert. Nach dem zweiten Newton'schen Axiom gilt für die Gravitations-, bzw. Gewichtskraft der folgende Sachverhalt.[6]

Gravitationskraft[6]:

$$\vec{F}_G = m \cdot \vec{a}_G \quad (3.31)$$

Mit der für die Nähe der Erdoberfläche geltenden Gravitationsbeschleunigung a_G [6]:

$$a_G = g \approx 9,81274 \cdot m \cdot s^{-2} \quad (3.32)$$

3.6. Trägheit

Die Trägheit ist die Eigenschaft von Körpern, die sie dazu veranlasst, solange in ihrem Zustand zu verbleiben, bis eine äußere Krafteinwirkung zu einer Änderung dieses Zustandes führt. Dieser Zustand kann eine Ruhelage oder eine Bewegung mit konstanter Geschwindigkeit sein. Newton hat dies in seinem Trägheitsgesetz dem ersten Newton'schen Axiom beschrieben. Als Nebenkriterium lässt sich über das erste Newton'sche Axiom ein Kriterium für ein Inertialsystem ableiten. So lässt sich ein Inertialsystem wie folgt definieren:

Jedes Bezugssystem, in dem sich ein kräftefreier Körper geradlinig gleichförmig bewegt, ist ein Inertialsystem[6]. Für die vorliegende Arbeit ist das Konzept des Inertialsystems von zentraler Bedeutung, da die in Inertialsystemen geltenden Newton'schen Axiome eins, zwei und drei hier zur Anwendung kommen.

3.7. Inertiale Navigation

Aus den vorherigen Erklärungen in diesem Kapitel geht hervor, dass die inertielle Navigation auf den Newton'schen Axiomen beruht. So wird in dieser Art der Navigation mittels Inertialsensoren, speziell Beschleunigungssensoren und Drehratensensoren, über einen sogenannten *Strapdown*-Algorithmus durch numerische Integration der Sensordaten die Lage, die Geschwindigkeit und schließlich die Position berechnet. *Strapdown* bedeutet festschnallen. In der vorliegenden Arbeit ist das Sensorsystem fest mit dem zu lokalisierendem Objekt verbunden.

3.8. Die Koordinatensysteme-Erdmodell des WGS 84

Für die Navigation im erdnahen Bereich wurden nach dem *World Geodetic System* (WGS 84) 1984 [8] ein Referenzsystem definiert, welches eine einheitliche globale Navigation ermöglichen soll. In der vorliegenden Arbeit dient dieses System als Orientierung. In diesem Modell sind folgende Punkte festgelegt:

- Die Koordinatensysteme (ECEF-*Earth Centred, Earth Fixed*-System)
- Ein Modell zur Erdbeschleunigung
- Ein Erd-Gravitationsmodell und ein dazugehöriges Geoidmodell (EGM 96 *Earth Gravity Field Modell*)
- Ein Modell zur Gestalt des Erdellipsoids

In der Tabelle 3.1 sind wichtige Parameter zum Erdmodell zusammengefasst.

| Bezeichnung | Wert |
|---------------------------------|--|
| Große Halbachse des Ellipsoids | 6378137,0m |
| Kleine Halbachse des Ellipsoids | 6356752,3142m |
| Abflachung des Ellipsoids | 0,0033528106647 |
| Exzentrizität des Ellipsoids | 0,0818191908426 |
| Erddrehrate | $7,292115 \cdot 10^{-5} \frac{rad}{s}$ |

Tabelle 3.1.: Parameter des Erdmodells WGS 84 [7]

Den Ursprung für die inertielle Navigation bildet die Umrechnung von einem Bezugs-Koordinatensystem in ein anderes. In diesem Abschnitt werden die vier nach WGS 84 definierten Koordinatensysteme beschrieben, wobei nicht alle in der vorliegenden Abhandlung verwendet werden. Alle vier Systeme sind rechtshändige, kartesische Koordinatensysteme.

1. Inertiales Koordinatensystem (Index-i)

Dieses Koordinatensystem dient als Grundlage für die Navigation, alle folgenden Systeme stehen in Bezug zu diesem. In ihm gelten die Newton'schen Axiome, es bewegt sich also gleichmäßig mit konstanter Geschwindigkeit und rotiert nicht. Es lässt sich durch folgende Punkte beschreiben.

- Der Ursprung liegt im Massenmittelpunkt der Erde
- Die Koordinatenachsen sind zum Fixstern-Himmel ausgerichtet

- Die z-Achse führt entlang der Rotationsachse der Erde
- Die x-Achse zeigt zum Frühlingspunkt
- Die y-Achse liegt senkrecht zur x-Achse in der Äquator-Ebene

2. Erdfestes Koordinatensystem (Index-e)

Dieses Koordinatensystem wird auch als *ECEF*-Koordinatensystem bezeichnet.

- Der Ursprung liegt ebenfalls im Massenmittelpunkt der Erde
- Die z-Achse führt entlang dem Rotationsvektor der Erde
- Die x-Achse führt entlang der Erdäquator-Ebene durch den Nullmeridian
- Die y-Achse liegt senkrecht zur x-Achse in der Äquator-Ebene

3. Körperfestes Koordinatensystem (Index-b)

Dieses Koordinatensystem ist fest mit dem Objekt verbunden, der Index-b kommt von *body*.

- Der Ursprung liegt im Massenmittelpunkt des Objekts
- Die z-Achse zeigt nach unten
- Die x-Achse in Objekt-Längsrichtung nach vorne
- Die y-Achse in Objekt-Querrichtung nach rechts

4. Navigationskoordinatensystem (Index-n)

Dieses Koordinatensystem dient als Referenzsystem und hat daher seinen Namen.

- Der Ursprung liegt im Massenmittelpunkt des Objekts
- Die z-Achse zeigt nach unten entlang des Erdbeschleunigungsvektors
- Die x-Achse zeigt zum Nordpol der Erde
- Die y-Achse zeigt zum geografischen Osten der Erde

In der Abbildung 3.8 sind die für die inertielle Navigation definierten Koordinatensysteme aufgeführt. Sie finden Anwendung bei der Navigation in Erdnähe, wobei eine Überführung nur in Bezug auf das erdfeste Koordinatensystem sinnvoll ist. Für eine Positionsbestimmung eines Objektes in der Nähe der Erdoberfläche ist weniger eine Ausrichtung zu den Fixsternen, als eine Bestimmung der Höhe, des Längen- und Breitengrades wichtig.

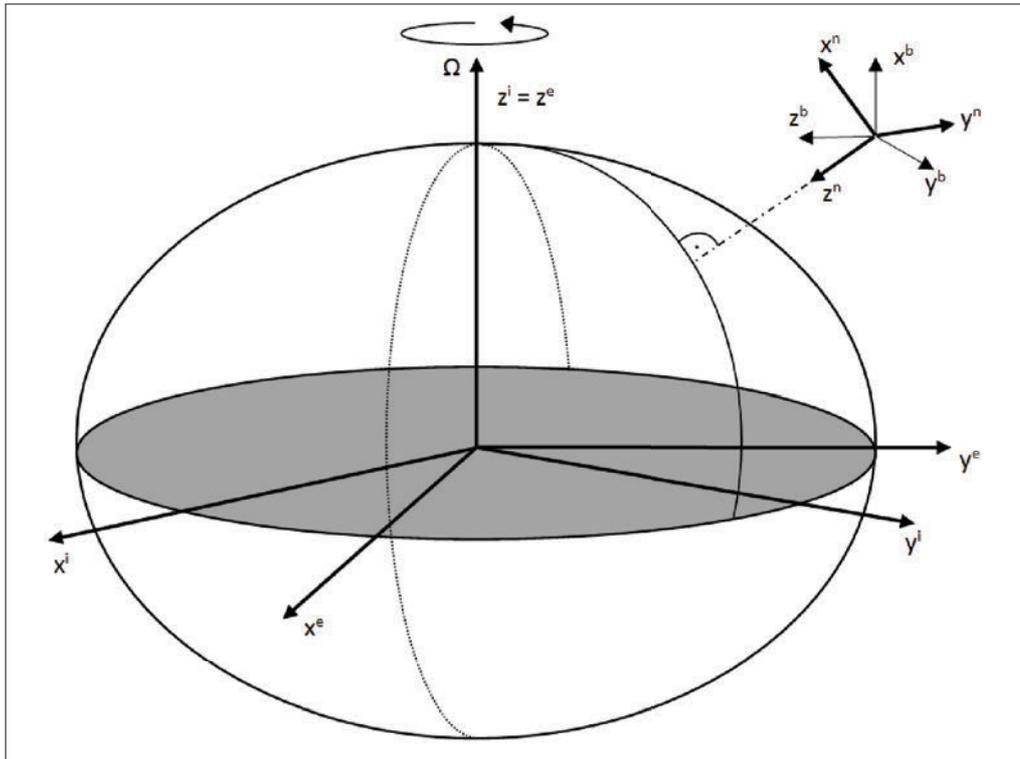


Abbildung 3.8.: Koordinatensystem für die erdnahe Navigation[7]

Im Verlauf der Arbeit kamen das Körperfeste- und das Navigationskoordinatensystem zum Tragen, dafür wurden einige Anpassungen vorgenommen, da es für die vorliegende Abhandlung um die Navigation in geringem Umfeld geht. Bei dem Körperfesten Koordinatensystem wurden die x- und y-Achsen vertauscht, da es in Bezug zum Sensor die Konstruktion des Demonstrators erleichterte. Ebenso wurde versucht ein Referenzkoordinatensystem zu bilden, welches dem Navigationskoordinatensystem zwar nahe kommt, aber dessen Ursprung sich unverändert am Startpunkt der Navigation befindet. Auf dieses Bezugskoordinatensystem wurden die Berechnungen zurückgeführt.

4. Mikro-Elektronische-Mechanische Systeme

In diesem Kapitel werden, nach einer Einführung in die Mikrosystemtechnik, die Funktionsprinzipien der verwendeten Sensoren vorgestellt.

Der Mensch orientiert sich gerne am Beispiel der Natur. Mikrobiologische Einheiten sind in der Lage Umweltreize aufzunehmen, zu verarbeiten und in geeigneter Weise durch Aktionen, wie Bewegungen, Geräusche oder Aussonderungen auf diese Reize zu reagieren. Der technische Fortschritt hat besonders in den letzten 30 Jahren Möglichkeiten eröffnet, miniaturisierte Systeme und Strukturen mit minimalen Abmessungen zu entwickeln. Einen guten Überblick über den historische Entwicklung der Mikrosystemtechnik gibt das Werk *Grundlagen der Mikrosystemtechnik* von G. Gerlach und W. Dötzel. [9]

Beispiele für Anwendungsgebiete sind:

- Verkehrstechnik
Piezo-Ventile, Beschleunigungs-, Drehraten- und Abstandssensoren
- Medizintechnik
Ultraschallwandler, Mikropumpen, Druck- und Kraftsensoren
- Umwelttechnik
Durchfluss-, Druck- und optische Sensoren
- Fertigungs- und Verfahrenstechnik
Kleinmotoren, Durchfluss- und Drucksensoren
- Sicherheitstechnik
Gassensoren
- Kommunikationstechnik
Mikrofone, Kopfhörer, Festplattenlaufwerke und Beamer
- Maschinenbau
Positioniersysteme, piezoelektrische Ventile und Schwingungsdämpfer

4. Mikro-Elektronische-Mechanische Systeme

Prägend ist dabei die Kombination der verschiedensten Mikrotechniken aus den Bereichen der Mikroelektronik, Mikromechanik, Mikrooptik und Mikrofluidik mit System- und Integrationstechniken. [10]

Laut Prof. Dr.-Ing. Manfred Kasper von der Technischen Universität Hamburg-Harburg ist die Mikrosystemtechnik wie folgt definiert:

Ein Mikrosystem ist die miniaturisierte Gesamtheit integrierter Senso-, Signalverarbeitungs- (oder Informationsverarbeitungs-) und Aktorkomponenten mit charakteristischen Abmessungen im Mikrometerbereich.

Die Mikrosystemtechnik ist die Gesamtheit von Verfahren zum Entwurf und zur Herstellung von Mikrosystemen. [11]

In den USA hat sich die Bezeichnung *Micro Electro Mechanical Systems* (MEMS) durchgesetzt. Dabei werden unter MEMS integrierte Bauelemente oder Systeme aus elektrischen und mechanischen Komponenten verstanden, dessen Gesamtgröße zwischen Mikrometern und Zentimetern variieren. In Japan werden miniaturisierte Systeme sehr allgemein als *Mikromaschinen* bezeichnet. Dabei besteht zu den europäischen und amerikanischen Systemen mit sensorisch geprägten Elementen ein wesentlicher Unterschied. Die japanischen Einheiten mit Anwendungsbereichen in der Medizin- und Industrietechnik sind eher auf Mikro- und Miniaktoren spezialisiert. [12]

Die Herstellungsverfahren von Mikrosystemen können verschiedenartig ausfallen. Für eine genauere Betrachtung sei an dieser Stelle auf einschlägige Literatur verwiesen, einen guten Überblick gibt das Buch *Grundlagen der Mikrosystemtechnik* [9]. Das Werk *Elektromechanische Systeme der Mikrotechnik Mechatronik-Dynamischer Entwurf-Grundlagen und Anwendungen* [13] gibt einen Einblick speziell für Elektromechanische Systeme.

Grundlegend geht es in der Mikrosystemtechnik um die Wandlung einer nicht elektrischen Größe in eine auswertbare elektrische Größe. Dafür wurden bereits diverse Möglichkeiten entdeckt, z.B. Wandlung für ionisierende Strahlung, optische-, thermische-, chemische-, biologische-, magnetische- und die, für die vorliegende Arbeit relevante mechanische Wandlung. Dabei sei erwähnt, dass die mechanische Wandlung in den meisten Fällen eine Umkehrbarkeit in der Signalverarbeitungsrichtung besitzt. Für diese Arbeit ist in Abbildung 4.1 veranschaulichte Verarbeitungsrichtung von Bedeutung.



Abbildung 4.1.: Signalverarbeitungsrichtung im Sensorsystem

Für die Beschreibung einer Bewegung sollen die schon in Kapitel 3 aufgezeigten translatorischen- und rotatorischen Bewegungen erfasst und in eine elektrische Größe gewandelt werden. Es ergibt sich folgendes Anforderungsprofil:

- Messung der Bewegung (Translation/ Rotation)
- Geeigneter Messbereich
- Geeignete Speisespannung
- Geeignete Auflösung und Empfindlichkeit
- Geringe Temperaturabhängigkeit
- Analoges oder digitales Signal zur Weiterverarbeitung
- Kurze Reaktionszeiten, Echtzeitmessung
- Kompakte Bauweise
- Geringe Komplexität und Abhängigkeit
- Geringer Leistungsaufnahme
- Kostengünstig

Anhand des Anforderungsprofils wurde eine Liste mit Sensoren erstellt. Dabei wurde festgestellt, dass momentan auf dem Markt diverse Produkte mit den unterschiedlichsten Eigenschaften existieren. So sind einige integrierte Sensorsysteme erhältlich, welche sowohl die Beschleunigungswerte, als auch die Drehraten der einzelnen Achsen zusammen in einer Einheit detektieren. Im Verlauf der Arbeit wurde entschieden, auf ein solches System zurückzugreifen, da dies eine notwendige Ausrichtung der sensitiven Achsen nicht erforderlich machte. Die Sensorachsen müssen für eine genaue Errechnung der Bewegungen orthogonal zueinander ausgerichtet sein, da sonst eine unvermeidliche Abweichung in die Berechnung einfließt. Auf mögliche Abweichungen und deren Ursachen wird im späteren Verlauf in Kapitel 9 der Arbeit noch näher eingegangen.

4.1. Beschleunigungssensoren

Die Beschleunigungssensoren, welche auch Accelerometer genannt werden, gehören derzeit zu den am häufigsten verbreiteten Sensorsystemen. In der vorliegenden Arbeit wurden kapazitive Sensoren verwendet, im folgenden wird eine Herleitung für das Funktionsprinzip erläutert. Mit den Sensoren lassen sich Beschleunigungen in translatorischen-, als auch in rotatorischen Systemen messen. Über die Ableitung

des Weges s nach der Zeit t und nochmaliges Ableiten der Geschwindigkeit v nach der Zeit t lässt sich die translatorische Beschleunigung a_{trans} berechnen.

Translatorische Systeme[12]:

$$a_{trans} = \frac{dv}{dt} = \frac{d^2s}{dt^2} = \ddot{s} \quad (4.1)$$

Über die Ableitung des Winkels φ nach der Zeit t lässt sich die Winkelgeschwindigkeit ω bestimmen und durch Ableitung der Winkelgeschwindigkeit ω nach der Zeit t lässt sich die rotatorische Beschleunigung a_{rot} berechnen.

Rotatorische Systeme[12]:

$$a_{rot} = \frac{d\omega}{dt} = \frac{d^2\varphi}{dt^2} = \ddot{\varphi} \quad (4.2)$$

Als Ansatz der Detektion der Beschleunigung dient das zweite Newton'sche Axiom. Demnach sind Kräfte die Ursachen für die Änderung im Bewegungszustand.

Dynamisches Grundgesetz[6]:

$$\vec{F}_a = m \cdot \vec{a} = m \cdot \frac{d^2l}{dt^2} \quad (4.3)$$

Als Basis für einen kapazitiven Beschleunigungssensor dient ein Feder-Masse-System. Die Abbildung 4.2 veranschaulicht das Funktionsprinzip.

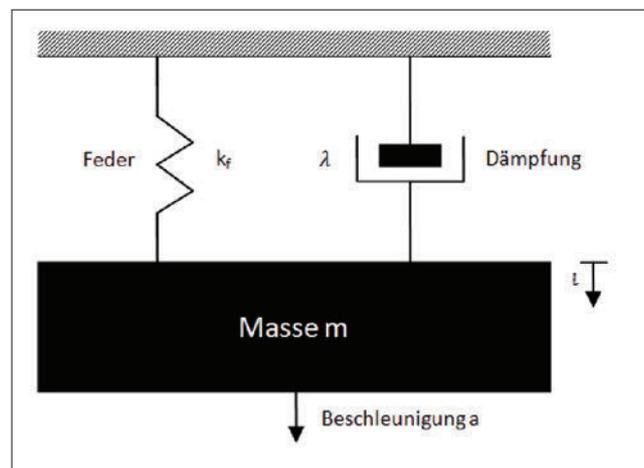


Abbildung 4.2.: Physikalisches Prinzip des MEMS-Beschleunigungssensor [12]

Eine durch Federn, mit der Federkonstante k_f , aufgehängte Prüfmasse, auch seismische Masse genannt, wird durch eine Beschleunigungskraft F_a um den Betrag l entlang der sensitiven Achse ausgelenkt. Die sensitive Achse verläuft entlang dem in der Abbildung 4.2 dargestellten Pfeil. Solange die Federkraft \vec{F}_f proportional zur Auslenkung \vec{l} ist gilt das Hooke'sche Gesetz.

Hooke'sches Gesetz[6]:

$$\vec{F}_f = -k_f \cdot \vec{l} \quad (4.4)$$

Dabei ist zu beachten, dass die Federkraft \vec{F}_f entgegengesetzt zur Auslenkung \vec{l} der Masse wirkt. In dem MEMS-Sensor umgibt Luft mit der Dämpfungskonstante λ das Innere des Sensors, diese Luft dämpft das System. Die Kraft \vec{F}_D ergibt sich aus der Dämpfungskonstante λ multipliziert mit der Geschwindigkeit \vec{v} . Auch hier sind der Geschwindigkeits- und Kraftvektor entgegengerichtet.

Dämpfungskraft[6]:

$$\vec{F}_D = -\lambda \cdot \vec{v} = -\lambda \cdot \frac{dl}{dt} \quad (4.5)$$

Aus der Summe der drei aufgeführten Kräfte ergibt sich die homogene lineare Differentialgleichung zweiter Ordnung mit konstanten Koeffizienten.

Differentialgleichung für einen gedämpften Oszillator[6]:

$$m \cdot \frac{d^2l}{dt^2} = -\lambda \cdot \frac{dl}{dt} - k_f \cdot \vec{l} \quad (4.6)$$

Aufgrund des Kräftegleichgewichts gilt für die Auslenkung der Masse durch eine externe Kraft in einem kapazitiven Beschleunigungssensor folgende Differentialgleichung.

Differentialgleichung für das Feder-Masse-Element [12]:

$$\vec{F}_a = m \cdot \vec{a} = m \cdot \frac{d^2l}{dt^2} + \lambda \cdot \frac{dl}{dt} + k_f \cdot \vec{l} \quad (4.7)$$

Nach der Division durch die Masse folgt daraus für die Beschleunigung[12]:

$$a = \frac{\delta^2 l}{\delta t^2} + \frac{\lambda}{m} \cdot \frac{\delta l}{\delta t} + \omega_{r0}^2 \cdot \vec{l} \quad (4.8)$$

Die Dämpfung durch die Luft ist in diesem System vernachlässigbar gering. Für

die Resonanzfrequenz ω_{r0} des Systems folgt dann [12]:

$$\omega_{r0} = \sqrt{\frac{k_f}{m}} \quad (4.9)$$

Für die Empfindlichkeit des Sensors gilt im stationären Zustand:

$$\frac{l}{a} = \frac{m}{k_f} \quad (4.10)$$

Die im Verlauf dargestellten Gleichungen erklären das allgemeine physikalische Prinzip eines kapazitiven Beschleunigungssensors. Die folgenden Gleichungen sollen die Wandlung auf eine elektrische und damit leichter auswertbare Größe beschreiben. Wie in Abbildung 4.3 ersichtlich wird besteht der Sensor aus einem Differentialkondensator mit den Einzelkapazitäten C_1 und C_2 .

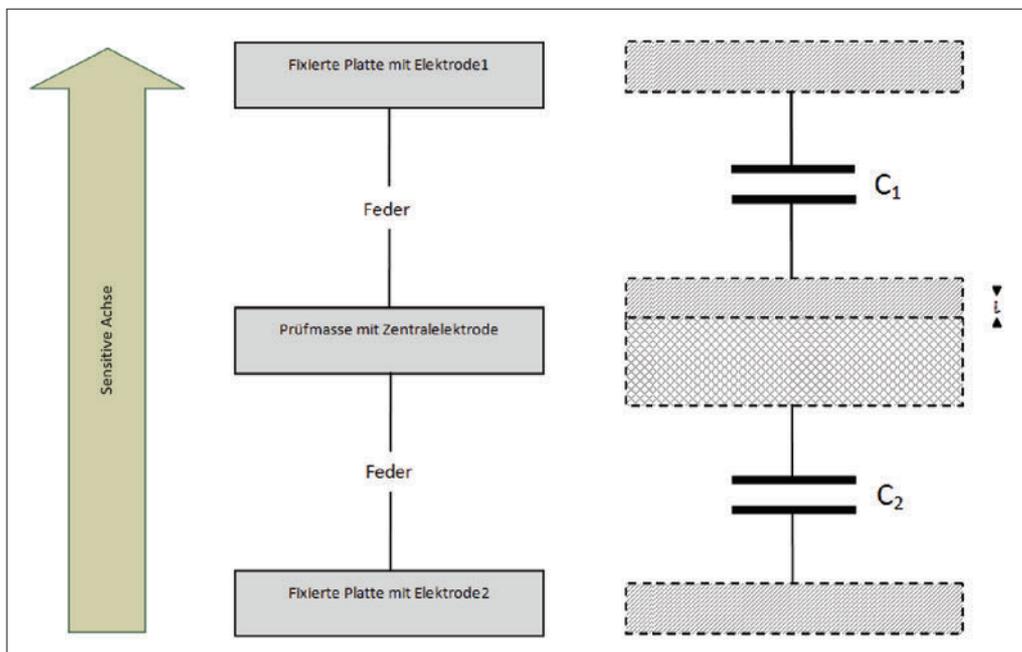


Abbildung 4.3.: Elektrisches Prinzip des MEMS-Beschleunigungssensor [12]

Dabei ist jeweils eine Platte der Kondensatoren mit der Prüfmasse verbunden und jeweils die zweite Platte befindet sich außen. Die äußeren Platten sind fest verbunden und die inneren Platten durch Verbund an der Prüfmasse variabel. Die Kondensatoren sind längs der sensitiven Achse ausgerichtet, dadurch detektieren sie eine Verschiebung der Prüfmasse durch die Änderung der Kapazität. In Ruhelage besitzen beide Ladungsspeicher die gleiche Kapazität C , sie setzt sich zusammen aus dem Quotienten von Ladung Q und Spannung U . Dies ist dem Produkt

aus der Dielektrizitätskonstante ε und dem Quotienten der Fläche A und dem Abstand d der Kondensator-Platten gleichgesetzt, siehe Gleichung 4.14. Die Dielektrizitätskonstante ε setzt sich aus dem Produkt der elektrischen Feldkonstante ε_0 (absolute Dielektrizitätskonstante) und der Permittivitätszahl ε_r (relative Dielektrizitätskonstante) zusammen.

Dielektrizitätskonstante[14]:

$$\varepsilon = \varepsilon_0 \cdot \varepsilon_r \quad (4.11)$$

Die elektrische Feldkonstante ε_0 hat den Wert[14]:

$$\varepsilon_0 = 8,85419 \cdot 10^{-12} \frac{As}{Vm} \quad (4.12)$$

Die Permittivitätszahl ε_r ist abhängig vom Material, in dem sich das Feld ausbreitet.

Permittivitätszahl ε_r für Luft[14]:

$$\varepsilon_r \approx 1 \quad (4.13)$$

Elektrische Kapazität[14]:

$$C = \frac{Q}{U} = \varepsilon \cdot \frac{A}{d} \quad (4.14)$$

Wird nun der Abstand d der Kondensator-Platten durch die Auslenkung l verschoben, ändert sich dadurch die Ladung Q , da die Speisespannung U_S konstant bleibt und demzufolge auch die Kapazität der einzelnen Kondensatoren. Für die Einzelkapazitäten in Abhängigkeit der Verschiebung der Prüfmasse mit konstantem A und ε erhält man.

Elektrische Kapazität mit Verschiebung:

$$C_1 = \varepsilon \cdot \frac{A}{d+l}; \quad C_2 = \varepsilon \cdot \frac{A}{d-l} \quad (4.15)$$

Durch den kapazitiven Spannungsteiler stellt sich am Ausgang des mit der konstanten Spannung U_S gespeisten Sensors folgende Spannung U_A ein.

Kapazitiver Spannungsteiler[14]:

$$U_A = U_S \cdot \frac{\frac{1}{C_2}}{\frac{1}{C_1} + \frac{1}{C_2}} = U_S \cdot \frac{1}{1 + \frac{C_2}{C_1}} \quad (4.16)$$

Die Gleichung 4.10 wird nach l aufgelöst, so erhält man:

$$l = \frac{m}{k_f} \cdot a \quad (4.17)$$

Wird nun der Quotient aus den Einzelkapazitäten C_2 und C_1 (Gleichung 4.15) gebildet und l eingesetzt, erhält man:

$$\frac{C_2}{C_1} = \frac{d + a \cdot \frac{m}{k_f}}{d - a \cdot \frac{m}{k_f}} \quad (4.18)$$

Wird $\frac{C_2}{C_1}$ in die Gleichung 4.16 eingesetzt, so ergibt sich daraus folgende Gleichung für die Ausgangsspannung U_A .

Ausgangsspannung des MEMS-Beschleunigungssensors[12]:

$$U_A = \frac{1}{2} \cdot U_S \cdot \left(1 - \frac{m}{k_f \cdot d} \cdot a\right) \quad (4.19)$$

Durch diese Herleitung [12] lässt sich der lineare Zusammenhang zwischen der Beschleunigung und der Ausgangsspannung herstellen. Die Ausgangsspannung ist also direkt proportional zur Beschleunigung.

Wird die Auslenkung direkt gemessen, bezeichnet man den Sensor mit diesem Verfahren als *open loop*-Beschleunigungssensor. Diese Sensoren eignen sich nur für sehr kleine Auslenkungen sehr gut, da der Abstand der Kondensatorplatten in Mikrosystemen sehr beschränkt ist. Nur so kann dann der Zusammenhang als linear angenommen werden.

Die sogenannten *closed loop*-Beschleunigungsmesser umgehen diese Problematik, indem sie die Auslenkung durch gezieltes Aufbringen einer Ladung elektrostatisch auf *Null* regeln. Die Masse wird im Gleichgewicht gehalten und die dafür erforderliche elektrostatische Kraft ist dann ein Maß für die Beschleunigung. Jan Wendel leitet die dazu notwendigen Beziehungen in [7] her. Die *closed loop*-Beschleunigungssensoren eignen sich deutlich besser für die Auswertung der Beschleunigungen, daher arbeiten die heutzutage verwendeten Sensoren überwiegend nach diesem Prinzip.

4.2. Drehratensensoren

Die Drehratensensoren, auch als Gyroskope oder Kreisel bezeichnet, detektieren die physikalische Größe Winkelgeschwindigkeit, bzw. Drehrate eines bewegten Objek-

tes. Für die Sensoren zur Messung der Winkelgeschwindigkeit sollte korrekterweise der Begriff Drehratensensor verwendet werden. Es gibt diverse Möglichkeiten zur Realisierung, dies können z.B. MEMS-Drehratensensoren, Faser-Kreisel und Ringlaser-Kreisel sein. D. H. Titterton beschreibt in [15] Anwendungen zur Detektion der Drehrate. Im folgenden wird vereinfacht das Messprinzip, welches auf dem Coriolis-Effekt beruht, betrachtet, dazu soll die Abbildung 4.4 behilflich sein. Die in der vorliegenden Arbeit verwendeten Sensoren arbeiten nach diesem Verfahren.

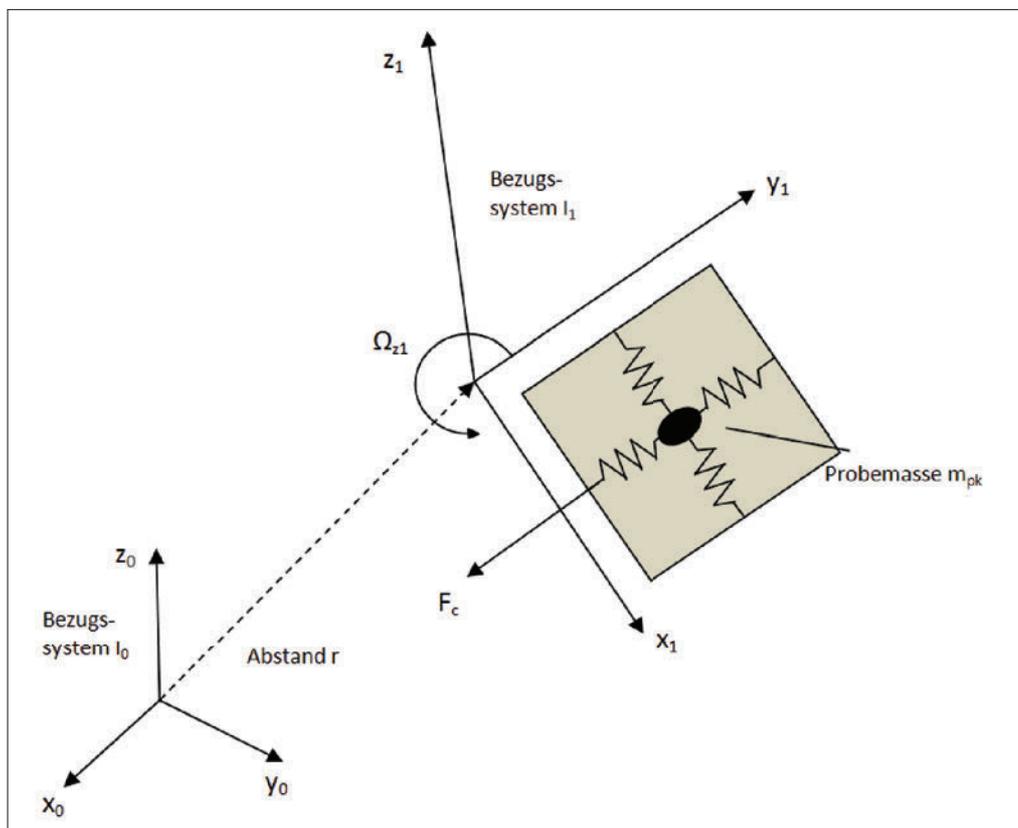


Abbildung 4.4.: Funktionsprinzip der Corioliskraft im Inertialsystem[12]

Der Drehratensensor beinhaltet einen Prüfkörper mit der Masse m_{pk} und befindet sich im Bezugssystem I_1 . Wird dieser Prüfkörper mit der Geschwindigkeit v_{x1} in x_1 -Richtung bewegt und rotiert dabei mit der Winkelgeschwindigkeit Ω_{z1} um die z_1 -Achse, so wirkt auf ihn die Corioliskraft in Richtung der y_1 -Achse.

Allgemeine Gleichung für die Corioliskraft[6]:

$$\vec{F}_c = -2m(\vec{v} \times \vec{\Omega}) \quad (4.20)$$

Dabei muss der Prüfkörper, um in eine Richtung die Drehrate detektieren zu können, zwei Bewegungsfreiheitsgrade besitzen, in x- und y-Richtung. Bei der Primär-Bewegung (Erregerschwingung) wird die Probemasse durch Anlegen einer Kraft aktiv zu einer Schwingung angeregt. Die Schwingbewegungen haben translatorische und rotatorische Anteile und können durch geeignete Aufhängungen der Probemasse mit Federn, Lagern und Führungen an das Bezugssystem (Chip) realisiert werden. Eine Winkelgeschwindigkeit ruft aufgrund der resultierenden Coriolis-Kraft eine Sekundärbewegung in translatorischer oder rotierender Form senkrecht zur primären Bewegung hervor. Diese Bewegung wird detektiert. Bei Systemen ohne Rückkopplung ist die Amplitude der Auslenkung ein Maß für die Winkelgeschwindigkeit. Bei rückgekoppelten Systemen bestimmt die durch Rückkoppelung auftretende Kraft als Maß die Winkelgeschwindigkeit. [12]

Für eine genaue Betrachtung kann die Arbeit *Robust Micromachined Vibratory Gyroskopes*[16] von C. Acar dienen.

Die MEMS-Sensoren haben in der Regel zwei flache Prüfkörper, wie in der Funktionsskizze in Abbildung 4.5 ersichtlich wird.

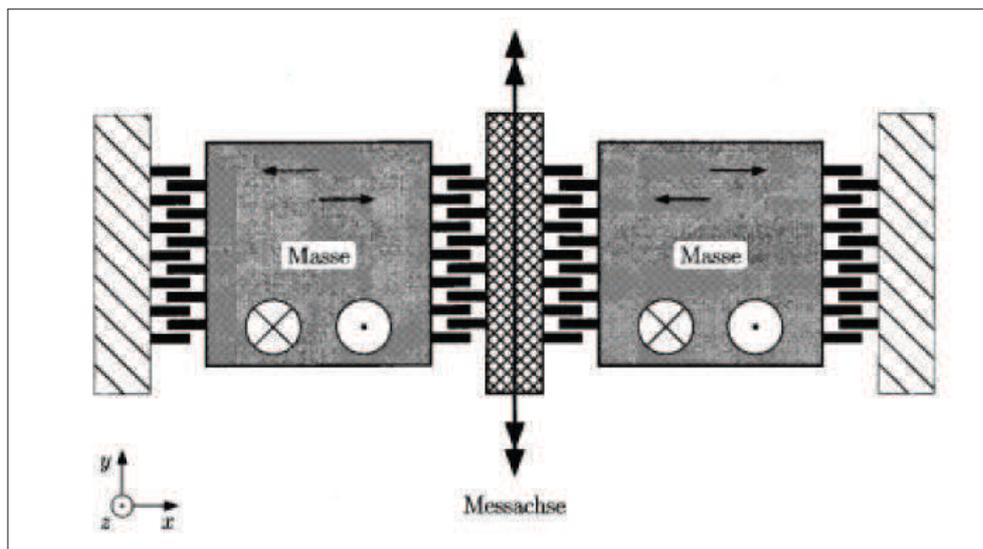


Abbildung 4.5.: Funktionsprinzip eines MEMS-Drehratensensor[17]

Die Prüfkörper werden durch elektrostatische Anregung zur gegenphasigen Schwingung in x-Richtung gebracht. Durch die Verwendung der Einheitsvektoren \vec{e} lassen

sich die Richtungen der Bewegungen erkennen. Liegt nun eine Drehrate $\Omega\vec{e}_y$ vor, so erhält man aufgrund der Corioliskraft eine Beschleunigung in z -Richtung.

Die Beschleunigung resultierend aus der Corioliskraft[7]:

$$\boxed{\vec{a}_c = 2\Omega\vec{e}_y \times v_a(t)\vec{e}_x = -2v_a(t)\Omega\vec{e}_z} \quad (4.21)$$

Die Beschleunigung \vec{a}_c verursacht eine Schwingung der Massen in Richtung der z -Achse. Die daraus resultierende Auslenkung stellt eine Verschiebung in der Kamm-Struktur des Sensors dar, was eine kapazitive Änderung hervorruft. Diese Änderung ist ein Maß für die Drehrate, bzw. Winkelgeschwindigkeit Ω .

5. Das Arduino-Mikrocontrollerboard

Bei Beginn der Arbeit stellte sich die Frage nach einem geeigneten Steuerelement im Sensorsystem zur Aufnahme, Übertragung und Auswertung der Sensordaten. Für dieses Steuerelement wurde folgendes Anforderungsprofil erstellt:

1. Geringe Komplexität (einfach in der Bedienung) und Abhängigkeit (von äußeren Einflüssen, wie Spannungsversorgung)
2. Einfache Handhabung für den Anschluss von Sensoren
3. Diverse Anschlussmöglichkeiten für Sensoren (Analog/Digital)
4. Präzise Signalverarbeitung (AD-Wandlung)
5. Bereitstellung einer Schnittstelle für eine kabellose Signalübertragung
6. Echtzeitmessung
7. Für den Einsatz gebräuchliche Eigenschaften (Abmessungen, Gewicht, robust)
8. Kostengünstig

Die Analyse und Auswertung dieser Anforderung führte zur Wahl des *Arduino*-Mikrocontrollerboards. In *Arduino-Physikal Computing für Bastler, Designer und Geeks* [19] ist die Namensgebung und Entstehung des *Arduino*-Projekts beschrieben. Die Programmiersprache für das Projekt wurde ausgehend von der Entwicklungsumgebung *Processing* geschrieben. *Processing* ist eine Grafikprogrammierplattform vom Massachusetts Institute of Technology Media Laboratory (MIT MediaLab).

Unter der *Creative-Commons*-Lizenz (CC-Lizenz) wurde die erste Serie von *Arduinos* produziert und im Internet veröffentlicht. Eine *Creative Commons* ist eine *Non-Profit*-Organisation, die in Form vorgefertigter Lizenzverträge eine Hilfestellung für die Veröffentlichung und Verbreitung digitaler Medieninhalte anbietet.[18] Die Rechte der Marke *Arduino* und an den Schaltplänen liegen bei der Firma *tin-ker.it*.

Arduino ist ein *Open-Source*-Projekt mit einer Entwicklungsumgebung, der Software und einem Entwicklungsboard, der Hardware aus dem Bereich Physical-Computing. Ursprünglich waren Designer und Künstler die Zielgruppe, mittlerweile bietet Arduino aber auch fortgeschrittenen Nutzern die Möglichkeit das Board als Schnittstelle für tiefer gehende Funktionen anzuwenden. Die hohe Beliebtheit der Plattform führt zu einer stetigen Weiterentwicklung und Verbesserung des Layouts des Boards, als auch der Entwicklungsumgebung. Im Internet sind diverse, gut dokumentierte Projekte zum Nachbauen und Ausprobieren zu finden. Zudem sind im Handel Bücher erhältlich, welche durch das Führen durch unterschiedliche Projekte die Welt des *Arduino* erklären [20], [21].

5.1. Das Entwicklungsboard

Arduino bietet mittlerweile eine Reihe von Entwicklungsboards an. Auf der *Arduino*-Website [26] sind alle Boards mit Informationen zu den Revisionen, *Shields* und Zubehörten aufgeführt. *Shields* sind Erweiterungen zum Aufstecken auf das *Arduino*-Board, sie ermöglichen eine leichte Schaltungs- und Funktionsergänzung. Durch die Open-Source-Vereinbarung wurden diverse Erneuerungen und Weiterentwicklungen vorangetrieben. Derzeit gibt es viele Unternehmen, die *Arduino*-Kompatible Produkte herstellen und anbieten. So sind besonders in den letzten Jahren eine Vielzahl an Zubehörteilen, Mikro-controller-Boards und *Shields* für die unterschiedlichsten Anwendungen und Funktionen entstanden.

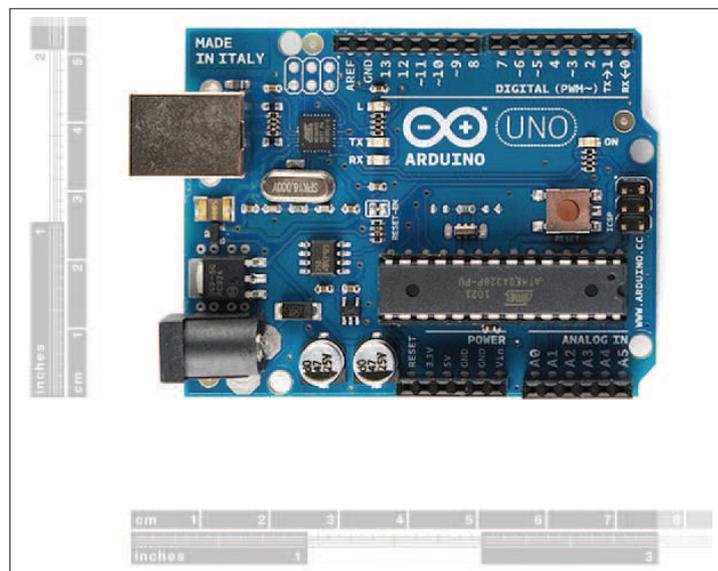


Abbildung 5.1.: Der Arduino Uno [32]

Im Verlauf der Arbeit kamen hauptsächlich der *Arduino-Duemilanove* und der *Arduino-Uno*, in Abbildung 5.1 aufgeführt, zur Anwendung. Im Anhang sind Datenblätter und Schematische Layouts zu den Boards und dem Mikrocontroller aufgeführt. Der Name *Duemilanove* kommt aus dem Italienischen und heißt 2009, er wurde nach seinem Erscheinungsjahr benannt. Beide *Arduino*-Typen sind funktionell sehr ähnlich und unterscheiden sich im Wesentlichen nur in einem Bauteil, dem Chip für die USB-Zu-Seriell-Konvertierung. Beim *Arduino-Uno* ist anstelle des FTDI(Future Technology Device International)FT232RL-Chips[30] der *ATmega8U2*[29] von der Firma *Atmel* verbaut, beide Datenblätter sind im Anhang aufgeführt. Zusätzlich kam es zu einer Designänderung, die die Identifikation der Ein- und Ausgänge erleichtern soll. Die Tabelle 5.1 zeigt eine Übersicht nützlicher Optionen des Mikrocontroller-Boards, welche im Verlauf der Arbeit zur Anwendung kamen und noch näher beschrieben werden.

| Bestandteil | Funktion/ Nutzung |
|-------------------|-----------------------------------|
| Microcontroller | ATmega328P |
| Flashspeicher | 32 KB (30 KB/ 2 KB Bootloader) |
| SRAM | 2 KB |
| EEPROM | 1 KB |
| Quarzoszillator | 16 MHz (Taktung) |
| Digital I/O Pins | 14 (davon 6 PWM) |
| DC-Strom I/O Pins | 40 mA |
| Analog Pins | 6 (Analoge Sensoren) |
| Analog Pin 4 | SDA (für I ² C) |
| Analog Pin 5 | SCL (für I ² C) |
| 3,3V Pin | 1 (50mA) (Energieversorg. Sensor) |
| 5V Pin | 1 |
| AREF Pin | Referenzspannung für ADC |
| TX Pin | Sender (seriell) |
| RX Pin | Empfänger (seriell) |
| USB Device | Programmierschnittstelle |
| <i>Power-Jack</i> | Externe Energieversorgung |

Tabelle 5.1.: Arduino Optionen

Beim *Duemilanove* wurde ab Mai 2009 der *ATmega168* durch den neueren und leistungsstärkeren *ATmega328P*[28] ersetzt, welcher auch im neueren Modell, dem *Arduino Uno* verwendet wird. Der *ATmega328P* wird durch einen 16MHz-Quarzoszillator getaktet und bietet eine für diese Arbeit gute zeitliche Auflösung. Das Board weist 6 analoge Eingänge auf, diese wurden mit den analogen Signalen der Sensoren ge-

speist. Die analogen Pins A0 bis A5 können konfiguriert werden, so sind sie als Energieversorgung (GND und VCC) und für eine I²C(Inter-Integrated Circuit)-Kommunikationsschnittstelle nutzbar. Dabei sind A4 als Daten-Pin(SDA) und A5 als Takt-Pin(SCL) vorbereitet. Bei dieser TWI(Two-Wire-Interface)-Schnittstelle handelt es sich um einen digitalen synchronen seriellen Datenbus, welcher eine spezielle *Library* erfordert. Die *Wire-Library* ist in der *Arduino*-Software bereits enthalten, muss aber in das entsprechende Programm eingebunden werden. Über einen Analog-Digital-Wandler mit einer Auflösung von 10 bit werden die Werte der analogen Pins in 1023 Schritten (1024 Werte) digitalisiert, dabei dient als Bezugsspannung standardmäßig 5V. Diese Referenzspannung kann aber über den *AREF*-Pin unter der Verwendung der Funktion *analogReference* auf einen anderen Wert gesetzt werden. Für eine genauere Betrachtung sei auf die späteren Abhandlungen in der vorliegenden Arbeit verwiesen, dabei wird sowohl das Einbinden einer Bibliothek(*Library*), als auch die Verwendung der I²C-Schnittstelle beschrieben[33]. Weiter stehen zusätzlich 14 digitale INPUT/OUTPUT Pins, von denen 6 mit einer Pulsweitenmodulation verwendet werden können, ein USB-Device, ein *Power-Jack*, ein ICSP(In-Circuit Serial Programming)-Anschluss und ein Reset-Schalter bereit. Die Energieversorgung kann zum einen über den USB-Port erfolgen, mit einem AC/DC-Adapter oder mit einer Batterie an den dafür vorgesehenen *Power-Jack*. Der *Power-Jack* ist eine zusätzliche Anschlussbuchse zur Spannungsversorgung über ein externes Netzteil, deren Pluspol mittig sitzt. Für die externe Versorgung muss keine Setup-Änderung vorgenommen werden, da die Energieversorgung automatisch umgestellt wird. Für den Betrieb mit einer Batterie kann außerdem ein GND-Pin und der V_{in} -Pin in der *Power*-Leiste genutzt werden. GND dient der Verbindung zum Bezugspotential. An V_{in} kann auch die Eingangsspannung mit einem Verlust von etwa zehn Prozent abgegriffen werden, da eine Durchschleifung zum *Power-Jack* existiert. Der 5V-Pin dient als regulierter Anschluss für den Mikrocontroller und andere Komponenten auf dem Board. Ein integrierter Regler nivelliert die Spannung bei einer externen Versorgung auf 5V. Liegt eine USB- oder eine andere geregelte 5V-Spannung vor, wird dieser Wert unverändert übernommen. Am 3V3 Pin wird mittels dem FTDI-Chip eine Spannung von 3,3V generiert, dabei steht ein maximaler Strom von 50mA zur Verfügung. Laut Angaben des Herstellers liegt der Arbeitsbereich bei 6-20V, es wird aber ein Betrieb bei 7-12V empfohlen. Bei einer Einspeisung von weniger als 7V ist es möglich, dass nicht ausreichend Spannung am 5V-Pin anliegt. Während die Benutzung mit mehr als 12V zu einer Überlastung des Spannungsreglers führt, und damit eine Beschädigung des Boards hervorgerufen werden kann. Der *ATmega328* ist zu den 32kB Flashspeicher mit 2kB SRAM und 1kB EEPROM ausgerüstet.

Von den 14 Digital-Pins des *Arduino*-Boards kann jeder als Ein- oder Ausgang verwendet werden, indem die Funktionen *pinMode()*, *digitalWrite()* und *digitalRead()* in das gewünschte Programm implementiert werden. Die Pins können maximal

40mA aufnehmen oder bereitstellen, wobei jeder einzelne einen internen *Pull-Up*-Widerstand mit 20-50kOhm besitzt. Die Standardeinstellung der Widerstände ist *deaktiviert*, sie können aber über die Eingabe von *digitalWrite(pin, HIGH)* aktiviert werden. Zusätzlich haben einige der Digital-Pins spezielle Funktionen:

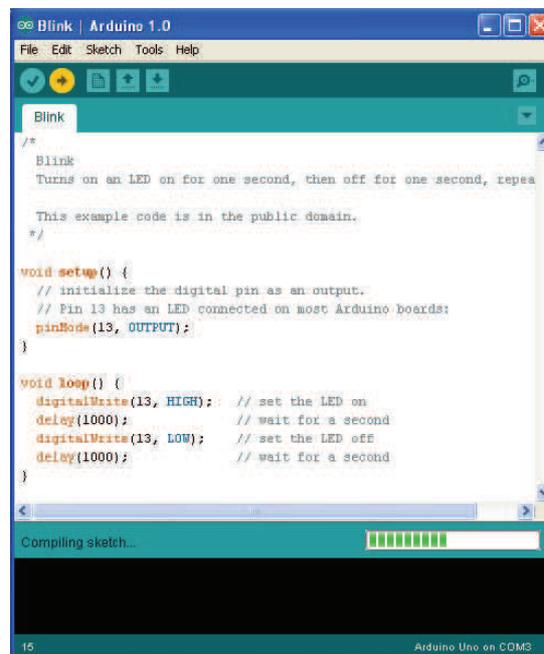
- Seriell: Pin0 (RX) und Pin1 (TX) sind mit den entsprechenden Pins des *ATmega8U2* USB-zu-TTL Chip verbunden. Sie werden für das Empfangen (RX-*Receive*) und das Übertragen (TX-*Transmit* der TTL-Daten verwendet.
- Externe Interrupts: Pin2 und Pin3 können so konfiguriert werden, dass bei niedrigen Pegeln, ansteigenden-, abfallenden Flanken oder bei einer Wertänderung Interrupts ausgelöst werden.
- PWM: Die Pins3, 5, 6, 9, 10 und 11 stehen in Verbindung mit der Funktion *analogWrite()* als Ausgang mit einer 8-bit Pulsweitenmodulation zur Verfügung.
- LED: Der Pin13 ist mit einer SMD Leuchtdiode verbunden. Bei *HIGH*-Pegel an Pin13 leuchtet die LED.
- RESET: Dieser Pin dient dem Zurücksetzen und Neustarten des Mikrocontrollers, dies erfolgt durch einen *LOW*-Pegel. Diese Funktion kann bei einer Blockierung des Boards beispielsweise durch ein *Shield* sehr nützlich sein.
- AREF: Bei Nutzung von analogen Eingängen kann dieser Pin zur Einspeisung der Referenzspannung dienen. Die Referenz liefert dann den Maximalwert für die 10 bit Analog-Digital-Wandlung. Die Funktion *analogReferenz()* muss dafür in das Programm implementiert werden.
- *SPI*: Die Pins10 bis 13 unterstützen unter Einbindung der *SPI-Library* die *SPI*-Kommunikation, wobei Pin10 SS (*Slave Select*), Pin11 MOSI (*Master out, Slave in*), Pin12 MISO (*Master in, Slave out*) und Pin13 SCK (*Serial Clock*) ist.

Die *SPI*-Kommunikation ist ein von *Motorola* entwickeltes und diversen Herstellern unterstütztes Bus-System zur Kommunikation von einem Hostprozessor mit Peripheriebausteinen über einen synchronen seriellen Datenbus. In einem Artikel von Martin Schwerdtfeger (06/2000) sind detaillierte Informationen aufgeführt.[34] Die Pinbelegung ist beim *Arduino Duemilanove* und dem *Arduino Uno* identisch, dadurch konnten im späteren Verlauf der Arbeit beide Boards für den entwickelten Prototypen verwendet werden. Um die volle Freiheit bei der Entwicklung nutzen zu können, wurde auf die Einbindung von *Shields* verzichtet. Zu Beachten ist allerdings, dass *Arduino* einen speziellen Abstand zwischen den einzelnen Pin-Leisten verwendet, was das eigene Konstruieren etwas erschwerte. Durch die einheitlichen

Abmessungen der meisten *Shields* lassen sich diverse Aufbauten vollziehen. Belegt ein *Shield* dabei eine Funktion nicht, werden die Pins nach oben weitergereicht. Somit hat das nächste *Shield* die identischen elektrischen Verbindungen zur Verfügung. Im Kapitel 6, Entwicklung eines Prototypen zur Lokalisation mit Inertial-Sensoren wird noch näher auf die verwendeten Hardwarebestandteile eingegangen.

5.2. Die Entwicklungsumgebung

Die Firma *Arduino* stellt die Entwicklungsumgebung, die *IDE* (Integrated Development Environment) [22] unter der aktuellen Version Arduino 1.0.1 frei zur Verfügung. Die *IDE* ist die gleiche wie die des *Processing*, und ist in der Programmiersprache Java geschrieben. Sie läuft auf handelsüblichen Computern mit den Betriebssystemen Windows, Mac OS X und Linux. Für die vorliegende Arbeit wurde die Version Arduino 1.0 verwendet, sie enthält einen schlicht gehaltenen und übersichtlichen Editor, eine Dokumentation sowie eine Vielzahl von Libraries und Beispielprogrammen. Die Programme werden *Sketches* genannt. Das Programmieren des Mikrocontrollers, also die Eingaben in den Editor, erfolgen in der Sprache *Wiring* [23] [31]. In Abbildung 5.2 ist ein Screenshot der Entwicklungsumgebung aufgeführt, zu sehen ist eines der Beispielprogramme, das *Blink*-Programm.

The image shows a screenshot of the Arduino IDE interface. The window title is "Blink | Arduino 1.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for opening files, saving, and running. The main text area contains the following code:

```
/*
 * Blink
 * Turns on an LED on for one second, then off for one second, repea
 *
 * This example code is in the public domain.
 */
void setup() {
  // initialize the digital pin as an output.
  // Pin 13 has an LED connected on most Arduino boards:
  pinMode(13, OUTPUT);
}

void loop() {
  digitalWrite(13, HIGH); // set the LED on
  delay(1000);           // wait for a second
  digitalWrite(13, LOW); // set the LED off
  delay(1000);           // wait for a second
}
```

At the bottom of the window, there is a status bar that says "Compiling sketch..." with a progress indicator and "15 Arduino Uno on COM3".

Abbildung 5.2.: Die Arduino Entwicklungsumgebung mit dem Beispiel *Blink*

Der Editor lässt sich, von oben nach unten, in drei wesentliche Bereiche unterteilen. In den ersten Bereich für die Funktionen und Steuerung, den zweiten Bereich für die Eingabe der *Sketches* und den dritten Bereich für Meldungen, die *Arduino* ausgibt. In der oberen Zeile des ersten Bereiches befindet sich eine Menüleiste, von der sich die einzelnen Optionen *File*, *Edit*, *Sketch*, *Tools* und *Help* aufrufen lassen. In den einzelnen Untermenüs der Optionen können diverse Funktionen ausgewählt werden. Unter *File* lassen sich alle dateibezogenen Funktionen aufrufen, wie beispielsweise Öffnen, Schließen, Speichern, Drucken, Hochladen der Programme und das Hochladen der Beispiel-*Sketches*. Mit den Funktionen unter *Edit* können die offenen Dateien bearbeitet werden. Beim Bedienen des *Sketch*-Untermenüs besteht die Möglichkeit die Programme zu kompilieren, sich den beinhalteten Ordner anzeigen zu lassen, oder Bibliotheken einzufügen. Unter *Tools* können die verwendeten Boards, *Serial*-Ports und *Programmer* ausgewählt werden. *Help* stellt die nötigen Informationen und Hilfestellungen bereit. In der darunter liegenden Zeile befindet sich die in Abbildung 5.3 zu sehende Schnellstartleiste mit den am häufigsten verwendeten Befehlen *Verify*, *Upload*, *New*, *Open*, *Save* und *Serial Monitor*.



Abbildung 5.3.: Die Schnellstartleiste Arduino

- *Verify*, das Symbol ist in Abbildung 5.4 zu sehen, bietet die Möglichkeit die geschriebenen *Sketches* auf Fehler zu überprüfen, dabei verifiziert der Compiler die Eingaben auf Richtigkeit, ohne das Programm auf den Programmspeicher des Mikrocontrollers zu spielen. Fehlerhafte Zeilen werden markiert und eine entsprechende Fehlermeldung wird ausgegeben.



Abbildung 5.4.: Das Arduino Symbol-*Verify*

- Mit dem *Upload*-Befehl, siehe Abbildung 5.5, wird der aktuelle *Sketch* kompiliert und auf das *Arduino*-Board hochgeladen. Im dritten Bereich für die Meldungen werden das Ergebnis des Kompiliervorgangs, sowie die Größe des erzeugten Programms angezeigt.



Abbildung 5.5.: Das Arduino Symbol-*Upload*

- Das in Abbildung 5.6 aufgeführte Symbol *New* öffnet einen neuen leeren *Sketch*. Es ist empfehlenswert, einen gerade bearbeiteten *Sketch* vorher abzuspeichern. Jedes Programm kann aus mehreren Dateien zusammengesetzt werden. So können einzelne Sequenzen von mehreren Programmen genutzt werden und es besteht die Möglichkeit, Funktionen für verschiedene Anwendungen zu verwenden.



Abbildung 5.6.: Das Arduino Symbol-*New*

- Mit dem Bedienen des Befehls *Open* dessen Symbol in Abbildung 5.7 zu sehen ist, lässt sich ein bereits abgespeicherter *Sketch* oder ein Beispielprogramm aus den *Examples* öffnen. Ebenfalls besteht damit die Möglichkeit, einen Blick in eine Bibliothek zu werfen und zu überprüfen, welche Anweisungen oder Deklarationen aus ihr hervorgehen.



Abbildung 5.7.: Das Arduino Symbol-*Open*

- Das Symbol *Save* (siehe Abbildung 5.8.) speichert den aktuellen *Sketch*.



Abbildung 5.8.: Das Arduino Symbol-*Save*

- In Abbildung 5.9 ist der in der Schnellstartleiste rechts angeordnete Schaltfläche *Serial Monitor* zu sehen. Durch bedienen der Schaltfläche wird der Serielle Monitor geöffnet, dieser bietet die Möglichkeit über die Befehlseingabe *Serial.print()* und *Serial.println()* Daten auf diesem zusätzlichen Fenster über die serielle Schnittstelle auszugeben.

Abbildung 5.9.: Das Arduino Symbol-*Serial Monitor*

Für die Ausgabe von Werten oder Daten des Mikrocontrollers können aber auch andere Software-Tools bzw. Terminal-Programme verwendet werden, auf die im Verlauf der Arbeit noch näher eingegangen wird. Unterhalb der Schnellstartleiste befinden sich die Reiter für die geöffneten *Sketches* und ein weiteres Schaltfläche zur Verwaltung der Dateien.

Der zweite Bereich ist das Texteingabefeld für den Programm-Code. Im dritten, schwarz hinterlegten Bereich, unterhalb des Texteingabefeldes, werden Benutzer- und Fehlermeldungen ausgegeben. Das Fenster schließt eine in Abbildung 5.10 aufgeführte Informationsleiste ab, in der unten links die Nummer der Zeile steht (hier im Beispiel 15), in der sich der Cursor im Textfeld befindet. Unten rechts befinden sich das ausgewählte *Arduino*-Board und der verwendete COM-Port. Der COM-Port ist hier die Bezeichnung der seriellen Schnittstelle, welche für die Verbindung von Mikrocontrollerboard und Computer angelegt wurde. Die Convertierung der Daten übernimmt beim *Arduino Uno* der schon angesprochene Chip *ATmega8U2* und beim Duemilanove der FTDI.



Abbildung 5.10.: Die Arduino-Informationsleiste

Wurde über die *Tools* im Menü der COM-Port und das Entwicklungsboard entsprechend ausgewählt, können über den Befehl *Upload* in der Schnellstartleiste oder im Menü unter *File* die gewünschten Programme aufgespielt werden. Es besteht die Möglichkeit, über den USB-Port Programme mit einer Größe von 30 kB auf den *Arduino* aufzuspielen. Zwar besitzen die beiden in der vorliegenden Arbeit verwendeten Entwicklungs-Boards einen Flashspeicher von 32 kB, aber 2 kB werden davon vom Bootloader benötigt. Bei Benutzung eines externen Hardware Programmiers [24] ständen diese 2 kB zusätzlich zur Verfügung.

Der integrierte Compiler der *Arduino*-Entwicklungsumgebung prüft, kompiliert und lädt die Programme auf den Mikrocontroller. Die Datenübertragung ermöglicht der FTDI Seriell zu USB Konvertierer, welcher eine entsprechende Treiber-Installation voraussetzt. Die *FTDIUSBSeriellDriver* sind in dem Unterordner *Driver* in dem von der *Arduino* Seite herunter geladenen komprimierten Archiv enthalten. Sie werden aber auch beim erstmaligen Anschließen des Boards an den Rechner mit einem Windows Betriebssystem, durch Auswahl des entsprechenden

Arduino-Ordners über den Windows Installationsassistenten installiert, falls sie nicht bereits vorhanden sind. In der *Arduino*-Umgebung erfolgen die Eingaben in der Sprache *Wiring* [31], sie ist eine Abwandlung von C und C++. Beim Kompilieren eines Quelltextes führt die *Arduino* Entwicklungsumgebung erst kleine Transformationen in einen echten C-, bzw. C++-Code durch. Mit dem Compiler *avr-gcc* wird der Sketch in Maschinencode übersetzt, welcher dann mit den Standard Libraries von *Arduino* verlinkt wird. Die daraus entstehende Hex-Datei wird dann in den Programmspeicher des Mikrocontrollers geladen.[25] Mit dem Programmer *avrdude* kann der Upload über eine serielle- oder USB-Verbindung durchgeführt werden.

Zum Schreiben eines lauffähigen Programms sind zwei Funktionen nötig. Die *setup*-Funktion und die *loop*-Funktion. Beide Funktionen umschließen Anweisungen und folgen dem aufgeführten Schema:

- void setup ()
 {
 Anweisungen;
 }
- void loop()
 {
 Anweisungen;
 }

Die *setup*-Funktion kommt im Programmablauf nach der Variablendefinition, sie wird nur einmal ausgeführt und dient Initialisierung (z.B. Hardware und Baudrate). Hier können beispielsweise Baudrate und Pins als Ein- oder Ausgang deklariert werden. Sie muss allerdings keine Anweisungen enthalten. Nach dem Durchlauf der *setup*-Funktion folgt der eigentliche Programmcode, die *loop*-Funktion. Sie beinhaltet die Hauptlogik und ist eine Endlosschleife. Über die Endlosschleife werden dann kontinuierlich die erfassten Sensordaten ausgegeben und der Berechnung zugeführt. Des weiteren können die Programme bis zur Grenze des Speichers mit Funktionen und Anweisungen verlängert werden. Für dieses Sensorsystem ist der Programmcode sehr gering gehalten, damit so viele Daten, bzw. Werte wie möglich, ohne zeitliche Verzögerung ausgegeben werden können. Für die Programmierung waren neben den schon aufgeführten Büchern, das *Arduino Programmier-Handbuch* von *Freeduino* [27] ein gute Hilfe, dort sind auch die Struktur, die Datentypen, die Konstanten, die Funktionen, die Arithmetik und die Ablaufsteuerung detailliert erklärt. Auf die einzelnen in den Programmen verwendeten Funktionen wird in dem Kapitel 6 näher eingegangen.

6. Entwicklung eines Demonstrators zur Lokalisation mit Inertialsensoren

Das Ziel dieser Arbeit war es, ein möglichst einfaches System zu entwickeln, um eine Lokalisation von Personen, welche sich in Häusern oder in eng bebauten Gebieten bewegen, zu untersuchen. Diese Personen können Feuerwehrleute, Rettungskräfte oder auch orientierungslose Patienten sein. Die verwendeten Hardware- und Softwarekomponenten sollten einfach in der Handhabung und vielfältig kompatibel sein. Das Gesamtsystem sollte die Möglichkeit bieten unterschiedliche Sensoren zu verwenden. Die verwendeten Bestandteile setzen sich überwiegend aus kommerziell erhältlichen Komponenten zusammen. Im Folgenden werden nach der Einführung in den Ablauf, die grundlegenden Entwürfe und die Systemarchitektur, die verwendeten Komponenten und die entwickelten Bestandteile vorgestellt. Die Entwicklung des Demonstrators lässt sich in die folgenden Abschnitte

- Anforderungsanalyse
- Anforderungsdefinition
- Entwurf/ Realisierung (Hardware und Software)
- Einsatz/ Prüfung

gliedern. Im Verlauf der Arbeit wurde die Anforderungsdefinition stetig überarbeitet und damit das Systemkonzept verbessert. So wurde beispielsweise entschieden, ein Sensorsystem zu verwenden, welches einen Beschleunigungssensor und einen Drehratensensor in einem Chip enthält. Diese Variante umgeht die sehr aufwendige Abstimmung, bzw. Ausrichtung (Orthogonalität der Achsen) der einzelnen Sensoren aufeinander. Der Vollständigkeit halber ist die Entwicklung mit dem Beschleunigungssensor als einzelnes Bauteil mit aufgeführt.

6.1. Systemarchitektur

Folgende grundlegende Anforderungen an das Konzept wurden gestellt:

1. Kompakte Bauweise (Größe, Gewicht, Robustheit)
2. Anschlussmöglichkeiten für unterschiedliche Sensoren
3. Anschlussmöglichkeit für ein Funkübertragungsmodul
4. Anschlussmöglichkeit für eine Spannungsversorgung

Da für die Tests mit dem Demonstrator eine drehbare Apparatur vorgesehen ist, war es zwingend notwendig eine Funkübertragung zu installieren. Eine Datenübertragung mit dem USB-Kabel würde bei längeren Tests zu Verwicklungen führen. Die Abbildung 6.1 zeigt die allgemeine Systemarchitektur des Demonstrators. Dabei bilden der Mikrocontroller, das Bluetooth Modul, der Sensor und die Spannungsversorgung eine Einheit als Sensorsystem und ein handelsüblicher Computer die Einheit der Auswertungsstelle. Der Computer muss dabei über eine Bluetooth Schnittstelle sowie über installierte Software und Treiber verfügen, damit er die vom Sensorsystem gesendeten Daten aufnehmen kann.

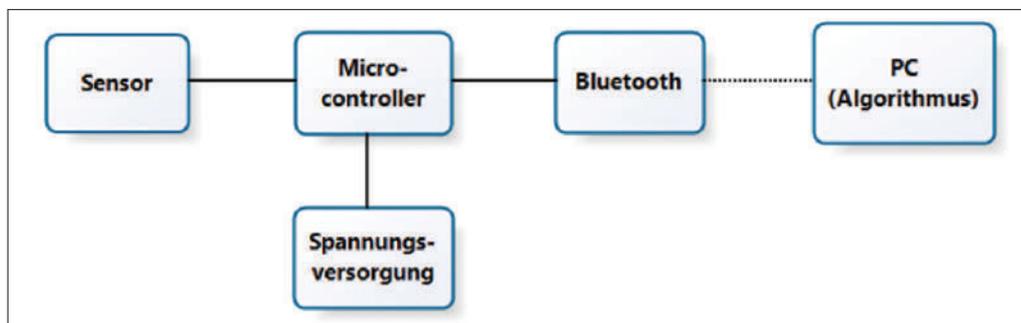


Abbildung 6.1.: Schema des Demonstrators mit den Systemkomponenten

Die Realisierung der kompakten Bauweise konnte wie folgt vorgenommen werden: Damit der Demonstrator tragbar ist und dabei nicht zu große Abmessungen oder ein zu massives Gewicht mit sich bringt wurden alle Komponenten, bis auf die Spannungsversorgung weitestgehend auf dem Mikrocontrollerboard integriert. So konnten der Sensor und das Funkmodul auf eine Laborkarte mit dem Rastermaß von 2,54 mm integriert werden. Unter Integration wird in diesem Kapitel die Einbindung in das Gesamtsystem verstanden. Dabei müssen die einzelnen Komponenten fest mit dem System verbunden sein, um nicht durch impulsartige Belastung gelöst zu werden.

6.2. Arduino Mikrocontrollerboard

Die in der vorliegenden Arbeit verwendeten Mikrocontrollerboards, der *Arduino Uno* und der *Arduino Duemilanove* haben exakt die gleichen Abmessungen. Die Abmessungen sind, wie in Abbildung 5.1 in Kapitel 5 dargestellt 75 mm in der Länge und 53 mm in der Breite. Die Pin-Leisten haben ebenfalls ein Rastermaß 2,54 mm, was den Einsatz der Laborkarten wesentlich beeinflusst hat. Insgesamt befinden sich vier Leisten für die Pins auf dem *Arduino* Board, wovon zwei auf jeder Seite angeordnet sind. Die in Abbildung 5.1 aufgeführte untere Pin-Leiste hat als freien Zwischenraum einen Abstand von 2 mm, dies entspricht einem freien Loch in der Laborkarte. Auf der gegenüberliegenden Seite ist der Abstand jedoch 1 mm, was den Einsatz der Laborkarte an dieser Stelle etwas erschwert. In der vorliegenden Arbeit wurden die linken oberen Pins, also Pin 8-15 nicht verwendet, dem zufolge konnte auf die Anschlüsse verzichtet werden. Um das Board vor Beschädigungen von unten zu schützen wurde eine etwa 3 mm dicke Plexiglasscheibe mit den Abmessungen 75 mm in der Länge und 55 mm in der Breite von unten an dem Board angebracht. Zur Befestigung wurden die im Board vorgestanzten Löcher verwendet. Für die externe Spannungsversorgung wurde ein im Elektronikhandel erhältlicher Batteriehalter für vier Mignon Batterien mit einem DC-Stecker 2,1·5,5 mm verbunden und an die dafür vorgesehene Anschlussbuchse des *Arduino* angeschlossen. Die vier Mignon Batterien lieferten insgesamt eine 6 V Spannung.

6.3. Datenübertragung mittels Bluetooth

Für die Funkübertragung standen mehrere Möglichkeiten zur Auswahl. Für die vorliegende Arbeit wurde entschieden, vorläufig mit der recht einfach installierbaren Bluetooth Übertragung zu arbeiten. Das verwendete Bluetooth Modul hat laut Datenblatt [35] eine Reichweite von maximal 100 m.



Abbildung 6.2.: Das verwendete Bluetooth Modul BlueSMIRF [35]

Es ist davon auszugehen, dass durch die Nutzung im *Indoor* Bereich die Reichweite auf bis zu 10 m sinkt, diese ist aber für die vorgesehenen Versuche ausreichend. In Abbildung 6.2 ist das verwendete Bluetooth Modul aufgeführt, im Anhang befinden sich ein Schema zum Layout und die erste Seite vom Datenblatt [35]. Zur Integration auf der Laborkarte wurde das Bluetooth-Modul mit einer Pin-Leiste und die Laborkarte mit einer entsprechenden Pin-Buchse bestückt, wobei nur vier der sechs Pins benötigt wurden. Der Pin 2 ist für den Anschluss der Versorgungsspannung von 3,3 V vorgesehen, der Pin 3 für Masse, der Pin 4 für TX und der Pin 5 für RX. Die Verbindungen zum RX-Pin und TX-Pin wurden zusätzlich mit einem Schalter versehen, um für das Hochladen des Programms die serielle Schnittstelle für die USB-Verbindung freizuschalten, anderenfalls würde die serielle Schnittstelle von der Bluetooth-Verbindung blockiert werden. Die Anschlüsse sind in den Abbildungen 6.4 (Kapitel 6.4) und 6.10 (Kapitel 6.5) dargestellt.

Das Bluetooth Modul unterstützt Datenraten von 1,2 Kbps bis 921 Kbps. Für das vorliegende System wurde eine Datenrate von 115,2 Kbps gewählt. Es nimmt bei einer Versorgungsspannung von 3,3 V, die vom Mikrocontrollerboard zur Verfügung gestellt werden maximal 100 mA auf. Da der *Arduino* auf dem 3,3 V-Pin nur 50 mA bereitstellt ist zu empfehlen, das Bluetooth Modul mit dem integrierten *Level Shifter* an den 5 V-Pin anzuschließen. Der Pegelwandler ist auf dem Schema im Anhang dargestellt und wandelt die Spannung von 5 V auf 3,3 V. Weitere Informationen können dem Datenblatt entnommen werden. Für die Implementierung in das Programm wurde die *Library-Wire.h* durch den Befehl `include <Wire.h>` am Anfang des Programms eingebunden, ein `char inCommand = ' ';` deklariert und in der `void loop()`-Schleife eine `if (Serial.available() > 0)` Abfrage eingefügt. In Kapitel 6.5 wird der vollständige Programmcode für ein Testprogramm dargestellt. Für Anwendungen, die über die Reichweite von Bluetooth hinaus gehen, sollte eventuell auf eine Funk-Technik, wie Zigbee zurückgegriffen werden. Zigbee ist allerdings deutlich komplizierter bei der Implementierung in Drahtlosnetzwerken [36].

6.4. Beschleunigungssensor ADXL335

In der vorliegenden Arbeit wurde der Beschleunigungssensor fertig aufgebaut auf einem bei *Farnell* [37] erhältlichen *Evaluation* Board verwendet. Auf dem *Eval* Board waren bereits alle nötigen Bauteile bestückt. Die Abbildung 6.3 zeigt das Blockschaltbild des Sensors. Der ADXL335 detektiert auf allen drei Achsen (x, y, z) die Beschleunigungen von ± 3 g mit einer Genauigkeit von etwa einem Hundertstel der Erdbeschleunigung. Es handelt sich um einen analogen Sensor, dies bedeutet, dass er die Speisespannung entsprechend der anfallenden Geschwindigkeitsänderung auf jeder Achse wandelt und als ein analoges Signal ausgibt. Daraus folgt also wieder ein Spannungssignal für den Eingang am Mikrocontrollerboard.

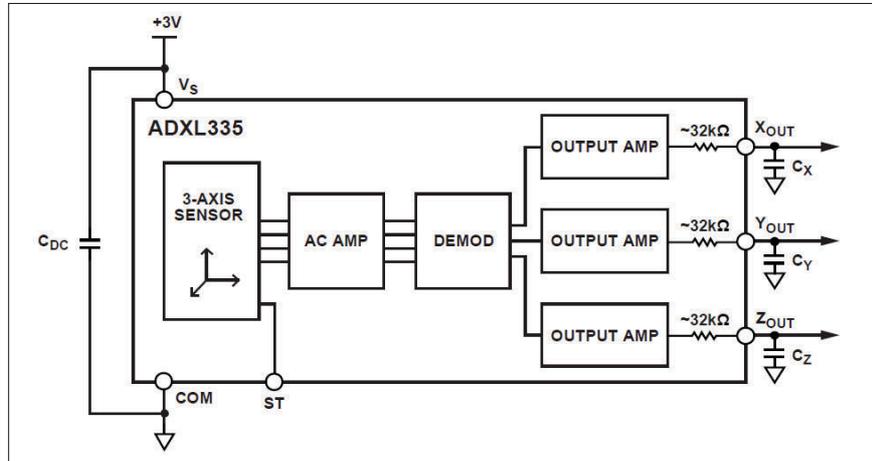


Abbildung 6.3.: Funktionales Blockschaltbild des Beschleunigungssensors ADXL335 [38]

Da der Sensor keine negativen Spannungen ausgeben kann wird die eingespeiste Spannung halbiert, dies entspricht dann einer Beschleunigung von ± 0 g. Alle Werte die unter dieser Spannung liegen sind dann negative Geschwindigkeitsänderungen und alle Werte darüber entsprechend positiv. Der Sensor wird mit einer Speisepannung von 3,3 V, die der *Arduino* bereitstellt betrieben, dabei verbraucht er $350 \mu A$. In Abbildung 6.4 ist die Pin-Belegung des gesamten Sensorsystems dargestellt.

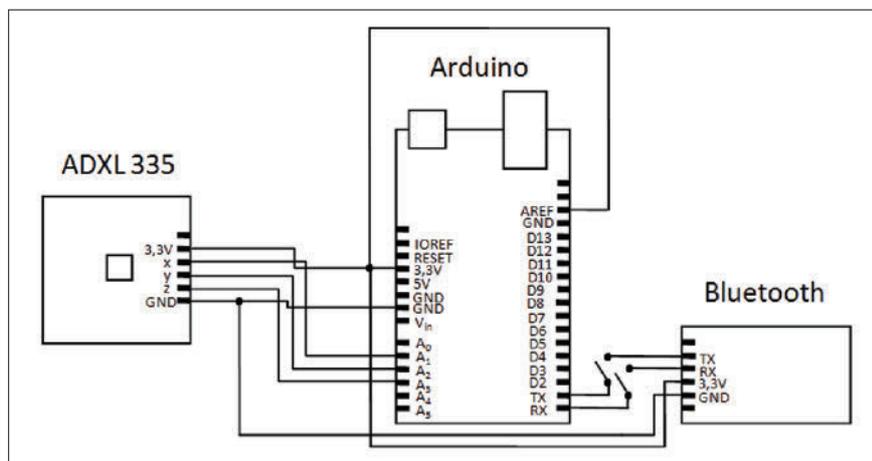


Abbildung 6.4.: Schaltplan des Demonstrator ADXL335-Arduino-Bluetooth

Die Pins für die Achsen x, y und z sind mit den analogen Pins des *Arduinos* verbunden. Die Pins GND und VCC am Sensor werden mit den entsprechenden Pins am *Arduino* verbunden, für die Versorgungsspannung wurde der 3,3 V-Pin gewählt. Der 3,3 V-Pin wird ebenfalls mit dem Pin AREF (Analoge Referenz) am *Arduino* verbunden, da dieser als externe Referenz für die AD-Wandlung dient. Der 10 bit Analog-Digital-Wandler würde sonst als Referenz die Speisespannung des Mikrocontrollers verwenden und die 1024 Schritte auf 5 V beziehen. Der Hersteller hat den Sensor mit einer Empfindlichkeit von $300 \frac{mV}{g}$ spezifiziert. Die ersten Tests wurden mit dem Beschleunigungssensor in Verbindung mit dem USB-Anschlusskabel durchgeführt. Dazu wurde ein Programm geschrieben, welches als Grundlage die Beispielprogramme der *Arduino* Entwicklungsumgebung hatte. Das Programm wurde so einfach wie möglich gehalten, um einen ersten Einblick in die Funktionsweise des Systems zu bekommen. In Abbildung 6.5 ist das Programm ADXLTEST1 aufgeführt.

```
const int xpin = A3; // X-Achse
const int ypin = A2; // Y-Achse
const int zpin = A1; // Z-Achse

void setup()
{
  Serial.begin(9600); // Initialisierung Baudrate
}

void loop()
{
  Serial.print(analogRead(xpin)); // Liest die Sensorwerte aus
  Serial.print("\t"); // Macht einen TAB
  Serial.print(analogRead(ypin));
  Serial.print("\t");
  Serial.print(analogRead(zpin));
  Serial.println(); // Springt in nächste Zeile
  delay(100); // Wartet 100ms
}
```

Abbildung 6.5.: Testprogramm des Beschleunigungssensor ADXLTEST1

Die ersten *Gehversuche* wurden in Form von einfachen Verschiebungen auf den Achsen unternommen. Daraufhin wurde der Sensor gedreht, um eine Detektion der Erdbeschleunigung auf den jeweiligen Achsen zu erkennen. Im weiteren Verlauf wurde die Standardabweichung in Ruhelage auf den einzelnen Achsen bestimmt. Im Verlauf der Arbeit wurden Modifikationen am Programm vorgenommen, die die Ausgabe der Daten verbesserten. Beispielsweise wurde die Zeit gemessen, die

zwischen den einzelnen Messungen vergeht und die Sensordaten wurden gemittelt.

6.5. Inertial Measurement Unit MPU6050

Der MPU6050 ist ein Sensorsystem bestehend aus einem Drei-Achsen Beschleunigungssensor, einem Drei-Achsen Drehratensensor und einem integrierten Motion Prozessor. Der Sensor liefert ein digitales Signal und unterstützt die Kommunikation über einen I^2C -Datenbus mit maximal 400 KHz. Die digitalen Sensorwerte können als Rohdaten oder über den Motion Prozessor als vorverarbeitete Daten ausgegeben werden. Der Drehratensensor kann mit einer Empfindlichkeit von bis zu $131/(^\circ/s)$ Drehraten in den Messbereichen von $\pm 250^\circ/s$, $\pm 500^\circ/s$, $\pm 1000^\circ/s$ und $\pm 2000^\circ/s$ je nach Einstellung detektieren. Der Beschleunigungssensor kann auf Messbereiche von $\pm 2g$, $\pm 4g$, $\pm 8g$ und $\pm 16g$ eingestellt werden, er hat bei der Auswahl des $\pm 2g$ -Messbereichs eine Empfindlichkeit von 16384 LSB/g. Die Speisespannung wird über einen 16 Bit-Analog-Digital-Wandler konvertiert. Bei einer Versorgung mit 3,3 V benötigt der Sensor maximal 3,9 mA, weitere Spezifikationen sind dem Datenblatt [39] zu entnehmen. In Abbildung 6.6 ist ein Blockschaltbild des Sensors aufgeführt.

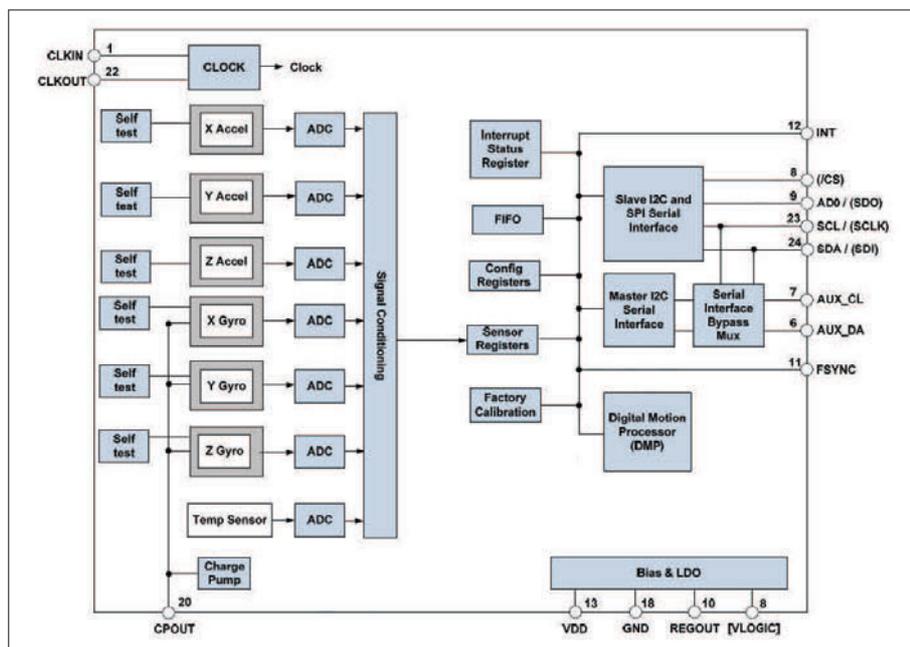


Abbildung 6.6.: Funktionales Blockschaltbild des Sensors MPU6050 [39]

6. Entwicklung eines Demonstrators zur Lokalisation mit Inertialsensoren

Zu dem Zeitpunkt, an dem der Sensor benötigt wurde, war er nicht als fertig aufgebautes Evaluation Board erhältlich. Er wurde als einzelner Chip mit den quadratischen Abmessungen von 4 mm, einer Höhe von 0,9 mm und einem QFN-24-Pin-Format bestellt. Für dieses Format musste eine Leiterplatte, in Abbildung 6.7 ersichtlich, bei einer externen Firma hergestellt werden.

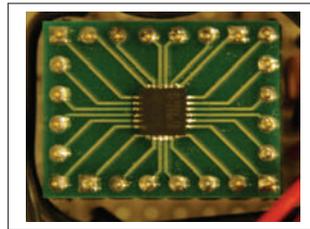


Abbildung 6.7.: Bestückung der Leiterplatte des Sensors MPU6050 [39]

Für die Auswertung wurde ein Evaluation Board nach der Empfehlung aus dem Datenblatt [39] nachgebaut. Eine Pinbelegung und Bestückung mit den nötigen Bauteilen ist in dem Schema der Abbildung 6.8 aufgeführt.

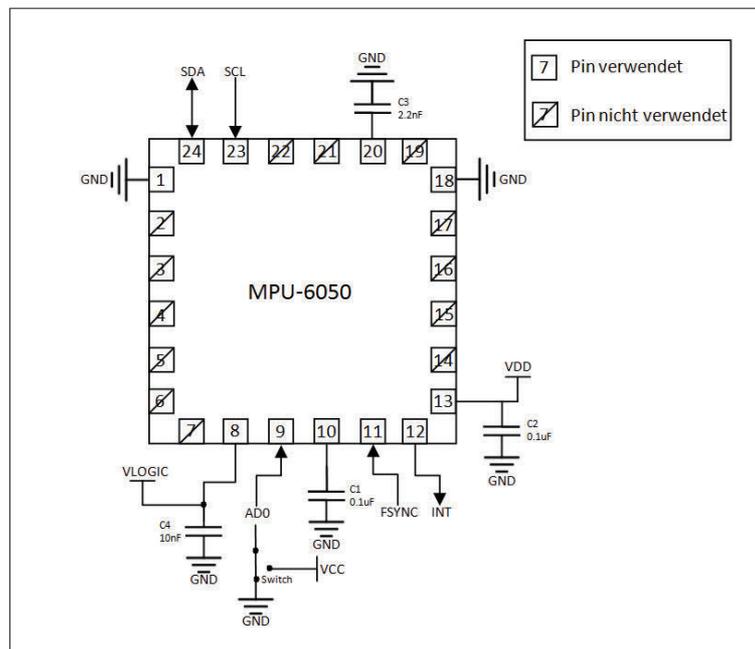


Abbildung 6.8.: Schema zur Pinbelegung und Bauteilbestückung des Sensors MPU6050 [39]

Die nicht benötigten Pins sind, wie in der Abbildung 6.8 ersichtlich durch streichen kenntlich gemacht. In der Tabelle 6.1 sind die verwendeten Pins, mit kurzen Erläuterungen aufgeführt.

| Pin-Nummer | Pin-Name | Beschreibung/Verwendung im Arduino |
|------------|----------|--|
| 1 | CLKIN | Clock-extern, wird nicht verwendet, auf GND |
| 8 | VLOGIC | Festlegung Spannungspegel, auf VDD(3,3V) |
| 9 | ADO | I^2C -Adressierung(AD0), auf GND |
| 10 | REGOUT | Regulator Filter, über einen 0,1 μ F auf GND |
| 11 | FSYNC | Frame Synchronisation dig. Input, auf GND |
| 12 | INT | Interrupt digital Output, Arduino Pin D2 |
| 13 | VDD | Versorgungsspannung, auf VDD(3,3V) |
| 18 | GND | Potential(Masse), auf GND |
| 20 | CPOUT | Charge pump, über 2,2 nF auf GND |
| 23 | SCL | I^2C -Clock, Arduino Pin A5 |
| 24 | SDA | I^2C -Daten, Arduino Pin A4 |

Tabelle 6.1.: Die Pinbelegung des Sensors MPU6050

Der MPU6050 wurde mit Pin-Buchsen auf einer auf 55×55 mm zurecht geschnittenen Laborkarte befestigt. Die Bauteile, wie in Abbildung 6.8 zu sehen, wurden ebenfalls auf die Laborkarte gelötet. Im Anschluss daran wurde die Karte mit den Buchsen für das Bluetooth Modul bestückt und die Kabel und Schalter dafür mit den entsprechenden Pins verbunden. Dabei wurde darauf geachtet, dass der Sensor genau mittig auf dem Demonstrator sitzt. Darüber ist ebenfalls mittig das Bluetooth Modul montiert. Die Abbildung 6.9 zeigt den fertigen Demonstrator ohne die Spannungsversorgung.

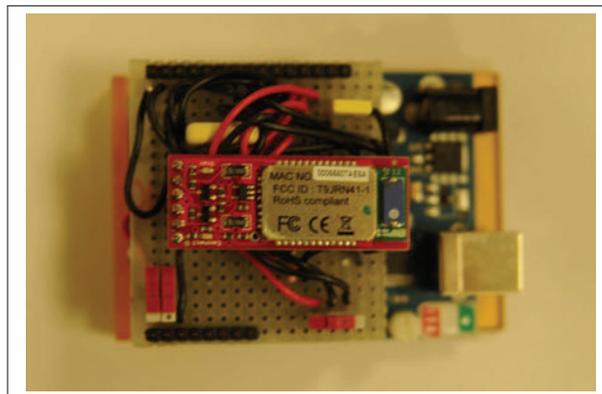


Abbildung 6.9.: Der fertige Demonstrator Arduino MPU6050 Bluetooth

6. Entwicklung eines Demonstrators zur Lokalisation mit Inertialsensoren

Die Abbildung 6.10 zeigt den Schaltplan des kompletten Demonstrators.

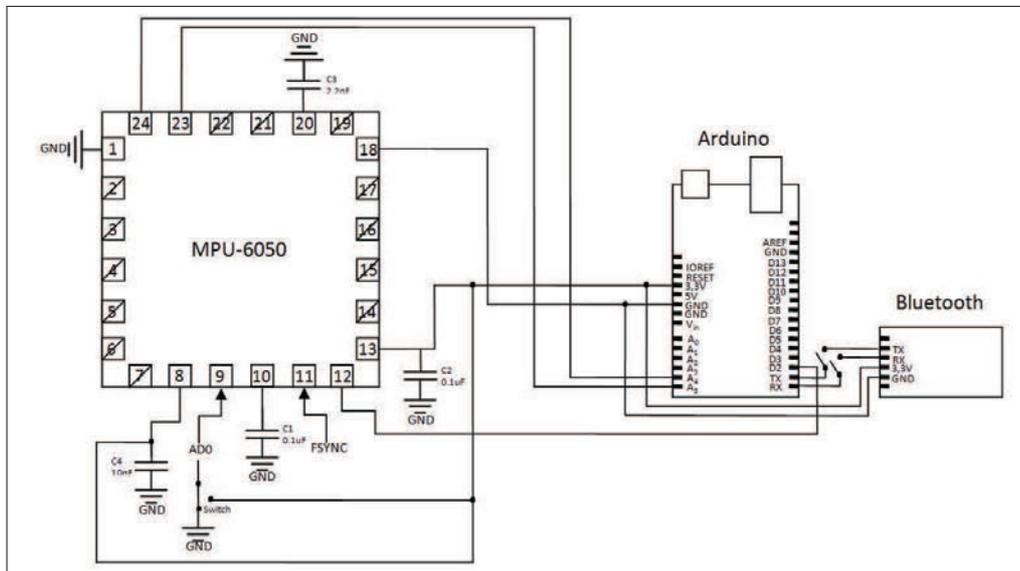


Abbildung 6.10.: Der Demonstratorschaltplan: Arduino-MPU6050-Bluetooth

In der Abbildung 6.11 ist der Demonstrator mit den Abmessungen, wie er in dem Testaufbau in Kapitel 8 durch Montage in eine Schiene integriert werden soll, aufgeführt.

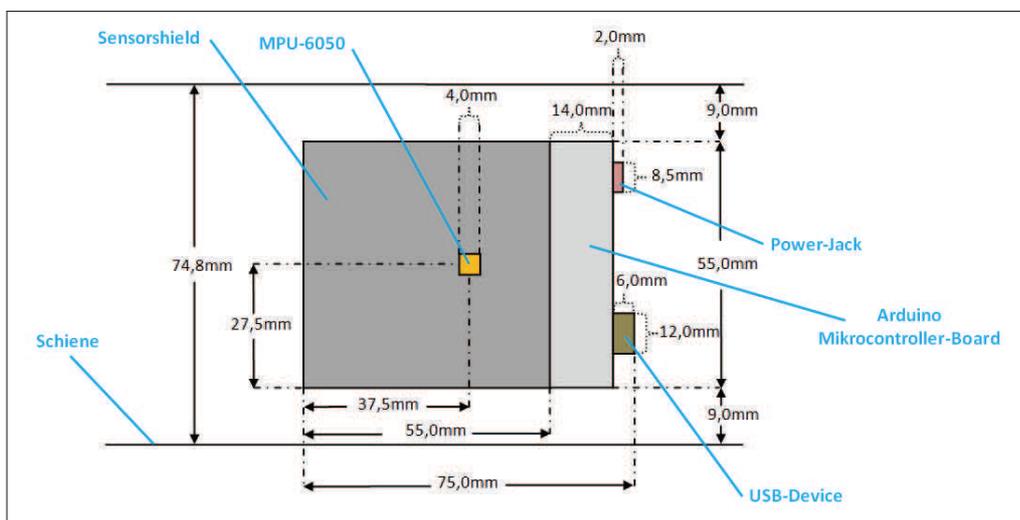


Abbildung 6.11.: Schema des Demonstrators in der Schiene

Für den *MPU6050* mussten zwei *Libraries* in die Software implementiert werden. Die eine *Library* bezieht sich auf den *I²C*-Datenbus, diese *Library* kann von der Seite [40] bezogen werden. Auf dieser Seite sind zusätzlich einige andere *Libraries* zu Bauteilen mit denen der *Arduino* arbeiten kann und zugehörige Beispiel-Programme zu finden. Die *MPU6050-Library* [41] ist dort auch aufgeführt. Um die *libraries* in die Entwicklungsumgebung des *Arduino* zu implementieren, werden die Dateordner der *libraries* in den Ordner *arduino-1.0/libraries* kopiert. Wenn die Ordner kopiert wurden, muss die *Arduino IDE* neu gestartet werden, um sicher zu gehen, dass die Dateien eingescannt wurden und verfügbar sind. Danach können die *Libraries* über die *Arduino IDE* unter *Sketch* und *Import Library...* ausgewählt und eingefügt werden. Die IDE bindet sie automatisch am Beginn des Quelltextes ein, dieses Verfahren ist zu empfehlen, da einige *Libraries* in mehrere kleinere Dateien unterteilt sind.

In dem *MPU6050*-Dateiordner [41] sind Beispielprogramme enthalten, eines dieser Beispiele, das *MPU6050 raw* wurde etwas modifiziert, bezüglich der Einbindung des Bluetooth Moduls und der Zeitabfrage zwischen den einzelnen Datenerfassungen des Sensors. Das Programm ist dabei so konzipiert, dass die erfassten Sensordaten als Rohdaten über die Bluetooth Verbindung an den Hauptrechner weitergeleitet werden. Im Anhang ist der Programmtext der Beispieldatei aufgeführt.

7. Berechnungsalgorithmus zur Lokalisation mittels Euler-Winkel

Der Berechnungsalgorithmus dient der Umsetzung der vom Sensorsystem aufgenommenen Daten. Um eine Berechnung mittels Euler-Winkeln vornehmen zu können, benötigen wir Daten aus der translatorischen- und der rotatorischen Bewegung. Diese Daten stellt das Sensorsystem MPU6050 in Form von linearen Beschleunigungswerten auf den Achsen x, y und z und von Drehraten auf den Achsen x, y und z bereit. Die Messkette in Abbildung 7.1 zeigt die einzelnen Schritte auf, wie die Daten vom Sensorsystem vorverarbeitet werden, damit der Hauptrechner mit ihnen besser umgehen kann.

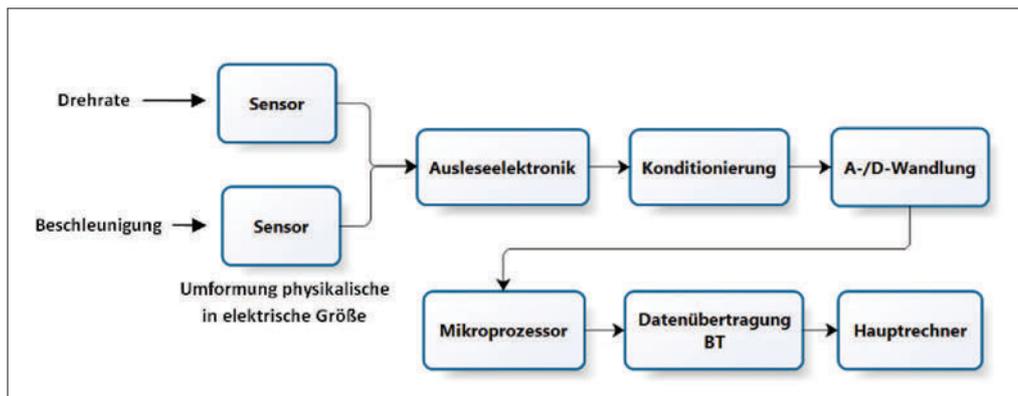


Abbildung 7.1.: Grafische Darstellung der Messkette

Das Sensorsystem MPU6050 wandelt über die in Kapitel 4 dargestellten Prinzipien die physikalische in eine elektrische Größe um. Die ausgelesenen elektrischen Größen werden noch im Sensor konditioniert (z.B. verstärkt), von einem integrierten 16 Bit Analog-Digital-Wandler konvertiert und danach nochmals konditioniert, um sie über die I^2C -Schnittstelle an den Mikrocontroller zu übertragen. Über den Mikrocontroller werden die aufgenommenen Daten mit Hilfe des Bluetooth Moduls an den Hauptrechner weitergeleitet. Am Hauptrechner stehen nun digitale *Rohdaten* in Form von ganzzahligen Werten (Integer) zwischen -32768 und

+32768 zur Verfügung. Diese Daten müssen, um sie mit dem Berechnungsalgorithmus verarbeiten zu können, noch in echte Beschleunigungen, bzw. Drehraten überführt werden. Im Schema der Abbildung 7.2 sind die einzelnen in den Berechnungsalgorithmus implementierten Schritte aufgeführt. Grundlegend lässt sich der

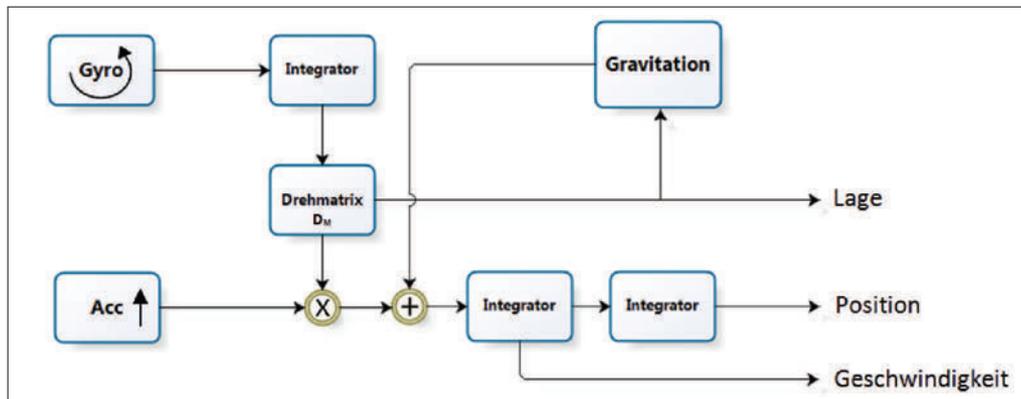


Abbildung 7.2.: Prozessschema des Berechnungsalgorithmus [7]

Algorithmus in drei wesentliche Abschnitte unterteilen. Das ist zum einen die Bestimmung der Lage. Damit wird die Ausgangslage von der die Lokalisierung startet und die Lage-Umrechnung in das gewählte Bezugssystem verstanden. Der nächste Schritt besteht aus der Eliminierung der mit detektierten Gravitation. Der Sensor kann nicht zwischen einer Beschleunigung resultierend aus einer Bewegung und einer Beschleunigung hervorgegangen aus der Erdmassenanziehung unterscheiden. Demzufolge muss die Gravitation, welche die Lokalisation maßgeblich verfälscht, von den detektierten Werten subtrahiert werden. Der letzte und dritte Abschnitt macht die eigentliche Umrechnung von der Beschleunigung in den zurückgelegten Weg aus.

7.1. Lagebestimmung

Um die Lage genau definieren zu können, muss das Bezugssystem für die Rückrechnung feststehen. In der vorliegenden Arbeit wurde als Bezugssystem, ein am Startpunkt ausgerichtetes Koordinatensystem, welches dem Navigationskoordinatensystem aus Kapitel 3.8 nahe kommt, ausgewählt. Dieses Bezugssystem hat folgende Eigenschaften:

- Der Ursprung liegt im Massenmittelpunkt des Objektes.
- Die z-Achse verläuft entlang des Erdbeschleunigungsvektors und zeigt nach oben.

- Die y-Achse zeigt in Richtung des voraussichtlichen Bewegungsraumes. Wird davon ausgegangen, dass die zu lokalisierende Person in ein Haus gehen will, dann ist der Startpunkt vor dem Haus und die y-Achse zeigt auf die Haus-Eingangstür.
- Die x-Achse steht senkrecht zur y-Achse und zeigt nach rechts.

Dabei spielt in der vorliegenden Arbeit die Nord-Ost-Ausrichtung keine Rolle. Es ist davon auszugehen, dass das körperfeste Koordinatensystem am Startpunkt nicht genau auf das Bezugssystem ausgerichtet werden kann, aber über die Messung der mit dem Beschleunigungssensor erfassten Erdbeschleunigung, kann eine Rückrechnung erfolgen. Dabei muss der wirksame Anteil der Erdbeschleunigung zu vollem Umfang auf der z-Achse liegen, allerdings mit dem für das in dieser Arbeit festgelegte System negativen Vorzeichen. Daraus ergibt sich für die gemessenen Werte folgender Vektor \vec{g}_m mit den Komponenten g_{mx} , g_{my} und g_{mz} .

Gemessener Vektor \vec{g}_m :

$$\vec{g}_m = \begin{pmatrix} g_{mx} \\ g_{my} \\ g_{mz} \end{pmatrix} \quad (7.1)$$

Wird nun am Startpunkt gemessen, werden auch auf der x- und y-Achse Anteile detektiert, also müsste das Koordinatensystem für eine Ausrichtung gedreht werden. Diese Ausrichtung kann durch zwei Drehungen realisiert werden.

Zunächst muss allerdings die Drehmatrix anhand der Euler-Winkel aufgestellt werden. Die Drehmatrix stellt die Grundlage für die Transformation von dem körperfesten Koordinatensystem in das Bezugssystem dar. Über sie lässt sich mit drei Drehungen um die Winkel der Achsen x, y und z eine Überführung realisieren. Grundsätzlich gibt es mehrere Möglichkeiten bezüglich der Reihenfolge der Drehungen. In der vorliegenden Arbeit wird mit der Drehung um die z-Achse begonnen, da davon auszugehen ist, dass bei der Fortbewegung einer Person die Drehungen um die z-Achse überwiegen. Der Winkel um diese z-Achse, auch als *yaw*-Winkel bezeichnet, wurde mit ψ benannt. Der Winkel um die y-Achse, der *pitch*-Winkel, hat die Bezeichnung θ erhalten und der *roll*-Winkel um die x-Achse wurde mit ϕ bestimmt.

- Drehmatrix D_z um die z-Achse

$$D_z = \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (7.2)$$

- Drehmatrix D_y um die y-Achse

$$D_y = \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \quad (7.3)$$

- Drehmatrix D_x um die x-Achse

$$D_x = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \quad (7.4)$$

Diese drei Drehungen lassen sich zu folgender Gesamtdrehmatrix D_G zusammenfassen.

Gesamtdrehmatrix D_G [7]:

$$D_G = \begin{pmatrix} \cos\theta\cos\psi & -\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi & \sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi \\ \cos\theta\sin\psi & \cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi & -\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta \end{pmatrix} \quad (7.5)$$

Der Vektor für die Erdbeschleunigung \vec{g} besteht aus den Einzelkomponenten g_x , g_y und g_z .

Tatsächlicher Erdbeschleunigungsvektor \vec{g} :

$$\vec{g} = \begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ -1 \end{pmatrix} \quad (7.6)$$

Die Überführung des gemessenen Vektor \vec{g}_m aus Gleichung 7.1 in den tatsächlichen Vektor \vec{g} mit den zwei oben aufgeführten Drehungen (x- und y-Achse) entspricht einer Drehung mit der Matrix D_G mit $\psi = 0$.

Wird in die Gleichung 7.5 $\psi = 0$ eingesetzt und aus-multipliziert, so erhält man folgendes Gleichungssystem [7].

gemessener Erdbeschleunigungsvektor \vec{g}_m :

$$\vec{g}_m = \begin{pmatrix} g_{mx} \\ g_{my} \\ g_{mz} \end{pmatrix} = \begin{pmatrix} -\sin\theta \cdot g_z \\ \sin\phi\cos\theta \cdot g_z \\ \cos\phi\cos\theta \cdot g_z \end{pmatrix} \quad (7.7)$$

Daraus ergeben sich für die Winkel θ und ϕ für die Drehung folgende Gleichungen [7].

Pitch-Winkel θ :

$$\theta = \arcsin\left(\frac{g_{mx}}{g_z}\right) \quad (7.8)$$

Roll-Winkel ϕ :

$$\phi = \arctan\left(\frac{g_{my}}{g_{mz}}\right) \quad (7.9)$$

Die als Rohdaten vorliegenden Beschleunigungswerte und Drehraten müssen für eine Berechnung modifiziert werden. Die am Startpunkt erfassten Werte können über einen Faktor K_D für die Drehrate angepasst werden.

Drehratenfaktor K_D :

$$K_D = \frac{\pm 250^\circ/s}{15bit - 1} = 7.6296274 \cdot 10^{-3} / s \quad (7.10)$$

Nach der Modifizierung der Werte liegen echte Drehraten, bzw. Winkelgeschwindigkeiten vor. Die Drehrate ω muss integriert werden um einen Winkel φ für die Berechnung zu erhalten.

Allgemeines Integral für den Winkel φ :

$$\varphi = \int \omega dt \quad (7.11)$$

Daraus folgt für die Berechnung die Gleichung:

$$\varphi = \int_{t_0}^t \frac{\omega_1 - \omega_0}{t_1 - t_0} \cdot (t - t_0) + \omega_0 dt + \varphi_0 \quad (7.12)$$

Zusammenfassung-Lagebestimmung

Zur besseren Übersicht sind die unter der Lagebestimmung aufgeführten Herleitungen hier zusammengefasst.

- Die Rohdaten der Drehrate werden mit dem Faktor K_D aus Gleichung 7.10 in echte Drehraten modifiziert.

- Die Drehraten werden durch die Gleichung 7.12 in die Winkel integriert.
- Die Anfangslage wird durch die Gleichungen 7.8 und 7.9 bestimmt.
- Die Drehmatrix D_G aus Gleichung 7.5 wird durch das Einsetzen der Anfangslage für den Start bestimmt.

7.2. Ausgleich der Gravitationswerte

Der folgende Abschnitt besteht im Wesentlichen aus der Eliminierung der Gravitation für die Berechnung. Dafür müssen die vom Sensor gelieferten Beschleunigungsrohdaten noch mit dem Faktor K_B angepasst werden. Der Faktor K_B berechnet sich wie folgt.

Beschleunigungsfaktor K_B :

$$K_B = \frac{\pm 2g}{15bit - 1} = 6.1037019 \cdot 10^{-5}g \quad (7.13)$$

Der Faktor K_B kann für negative, wie auch für positive Werte genutzt werden. Die Beschleunigungswerte liegen nun als Vielfaches der Erdbeschleunigung vor, für die Subtraktion der Gravitation werden sie vorerst so belassen.

Um die Gravitation, bzw. Erdbeschleunigung zu eliminieren gibt es mehrere Möglichkeiten. Dies ist zum einen die Subtraktion sofort nach Eingang der Werte in den Berechnungsalgorithmus, diese Varianten beinhaltet aber die ständige Umrechnung des Vektors \vec{g} aus dem Bezugssystem in das körperfeste Koordinatensystem mit der Rücktransformierten der Drehmatrix D_G . Die andere Möglichkeit, welche für den Algorithmus in der vorliegenden Arbeit gewählt wurde, besteht darin, die gemessenen Beschleunigungsdaten \vec{a}_m erst in das Bezugssystem zu dem Vektor \vec{a} zu transformieren und danach den bekannten Wert für \vec{g} zu addieren. Dabei ist zu beachten das der Vektor \vec{g} der z-Achse im Bezugssystem entgegen gerichtet ist, der resultierende Wert ist also $g_z = -1$.

Eliminierung des Erdbeschleunigungs-Vektors \vec{g} :

$$\vec{a} = \begin{pmatrix} a_x \\ a_y \\ a_z \end{pmatrix} = D_G \times \begin{pmatrix} a_{mx} \\ a_{my} \\ a_{mz} \end{pmatrix} + \begin{pmatrix} g_x \\ g_y \\ g_z \end{pmatrix} \quad (7.14)$$

Anschließend wurde der Vektor auf die SI-Einheiten umgerechnet. Dies erfolgte über die Multiplikation mit dem Faktor für die Erdbeschleunigung $9.81274 \frac{m}{s^2}$.

Zusammenfassung-Ausgleich der Gravitationswerte

Zur besseren Übersicht sind die unter dem Abschnitt Ausgleich der Gravitationswerte aufgeführten Herleitungen hier zusammengefasst.

- Die Beschleunigungs-Rohdaten werden mit dem Faktor K_B aus Gleichung 7.13 in echte Beschleunigungen modifiziert.
- Die echten Beschleunigungen \vec{a}_m werden einschließlich der Anfangsbedingungen in der Drehmatrix D_G aus Gleichung 7.5 in den Vektor \vec{a} in das Bezugssystem überführt.
- Auf den Beschleunigungsvektor $D_G \times \vec{a}_m$ wird der Erdbeschleunigungsvektor \vec{g} aufaddiert.
- Der Beschleunigungsvektor wurde mit dem Faktor für die Erdbeschleunigung multipliziert.

7.3. Berechnung des zurückgelegten Weges

Der aus den Abschnitten 7.1 und 7.2 entstandene Beschleunigungsvektor geht nun in die Berechnung des Weges \vec{s} ein. Dabei wird davon ausgegangen, dass die Zeitdauern der einzelnen Messschritte so klein sind, dass die Beschleunigungsänderung in dem Zeitintervall als linear angenommen werden kann. Die aus den Einzelberechnungen der Messschritte entstandenen Streckenteile werden zu dem gesamten Weg aufaddiert. Für die Berechnung des Weges aus der Beschleunigung wird über das erste Integral die Geschwindigkeit \vec{v} bestimmt. Die in den Grundlagen in Kapitel 3 hergeleitete Gleichung 3.17 wird dazu genutzt.

Daraus folgt für die Geschwindigkeit:

$$v(t) = v_0 + a_0 \cdot t + \frac{1}{2} \cdot \frac{a_1 - a_0}{t_1} \cdot t^2 \quad (7.15)$$

Wegstrecke bei linearer Beschleunigungsänderung:

$$s(t) = s_0 + \frac{1}{2} a_0 \cdot t^2 + \frac{1}{6} \cdot \frac{(a_1 - a_0)}{t_1} \cdot t^3 + v_0 \cdot t \quad (7.16)$$

Zusammenfassung-Berechnung des zurückgelegten Weges

- Die Geschwindigkeiten werden aus den in Kapitel 7.2 bestimmten Beschleunigungswerten berechnet.

- Die Wegstrecken werden aus den in Abschnitt 7.2 bestimmten Beschleunigungswerten und den in Abschnitt 7.3 bestimmten Geschwindigkeiten berechnet.
- Die Einzelwegstrecken werden nach jedem Messpunkt auf den Gesamtweg addiert.

Zusammenfassung-Gesamt-Berechnung

In diesem Abschnitt wird der Algorithmus bestehend aus den in den Kapiteln 7.1, 7.2 und 7.3 aufgeführten Berechnungen zusammengefasst.

- Die Drehraten-Rohdaten werden mit dem Faktor K_D aus Gleichung 7.10 in echte Drehraten modifiziert.
- Die Drehraten werden durch die Gleichung 7.12 in die Winkel integriert.
- Die Anfangslage wird durch die Gleichungen 7.8 und 7.9 bestimmt.
- Die Drehmatrix D_G aus Gleichung 7.5 wird durch das Einsetzen der Anfangslage für den Start bestimmt.
- Die Beschleunigungs-Rohdaten werden mit dem Faktor K_B aus Gleichung 7.13 in echte Beschleunigungen modifiziert.
- Die echten Beschleunigungen \vec{a}_m werden einschließlich der Anfangsbedingungen in der Drehmatrix D_G aus Gleichung 7.5 in den Vektor \vec{a} in das Bezugssystem überführt.
- Auf den Beschleunigungsvektor $D_G \times \vec{a}_m$ wird der Erdbeschleunigungsvektor \vec{g} aufaddiert.
- Der Beschleunigungsvektor wird mit dem Faktor für die Erdbeschleunigung multipliziert.
- Die Geschwindigkeiten werden aus den in Kapitel 7.2 bestimmten Beschleunigungswerten berechnet.
- Die Wegstrecken werden aus den in Abschnitt 7.2 bestimmten Beschleunigungswerten und den in Abschnitt 7.3 bestimmten Geschwindigkeiten berechnet.
- Die Einzelwegstrecken werden nach jedem Messpunkt auf den Gesamtweg addiert.

Der hier aufgeführte Algorithmus wurde mit simulierten Daten wie folgt getestet. Es sollte eine Kreisbahn simuliert werden, mit den dafür errechneten Werten für die Drehung um die z-Achse und dem entsprechenden linearen Werten für die Achsen x und y. Ausgehend von der Drehrate um die z-Achse, wurden alle Berechnungen ausgeführt. Bei dem Test sollte die Drehung bei den Anfangsbedingungen $t = 0$, $s = 0$, $v = 0$, $a = 0$ und $\omega = 0$ für alle drei Achsen x,y und z starten. Die Berechnung wurde nach folgendem Ablauf durchgeführt.

1. Bestimmung der auf die z-Achse wirkenden Erdbeschleunigung
2. Bestimmung des Zeitintervalls für die einzelnen Messschritte
3. Bestimmung der Winkelgeschwindigkeit auf der Z-Achse
4. Bestimmung des Startzeitpunktes der Winkelgeschwindigkeit, wann die Winkelgeschwindigkeit erreicht sein sollte
5. Bestimmung der Zeit, in der die Winkelgeschwindigkeit erreicht sein soll
6. Bestimmung der Winkelbeschleunigung, die zum Erreichen der Winkelgeschwindigkeit führt
7. Bestimmung der auf die x-Achse wirkenden Beschleunigung, ausgehend von Winkelgeschwindigkeit um die z-Achse
8. Bestimmung der auf die y-Achse wirkenden Beschleunigung, ausgehend von Winkelgeschwindigkeit um die z-Achse

Die Ergebnisse der Simulation sind in Kapitel 9 aufgeführt.

8. Testaufbau zur Bestimmung der Position

Der Versuchsaufbau dient in erster Linie der Analyse des Systems auf richtige Funktionsweise in Bezug auf die Positionsbestimmung. So sollen damit definierte Werte für die Beschleunigungen und die Drehraten auf den einzelnen Achsen zur Verfügung stehen. Im Gesamten soll überprüft werden, ob das System die dynamischen Veränderungen korrekt verarbeitet und ob mittels des Berechnungsalgorithmus die Daten richtig umgesetzt werden. Aus der Anforderung bezüglich eines definierten Beschleunigungswertes heraus, entstand die Idee, die Zentrifugalbeschleunigung zu nutzen. Daraus folgte die Idee der Verwendung eines Rades, welches über einen Motor angetrieben wird, zudem wird gleichzeitig eine auswertbare Drehrate zur Verfügung gestellt. Das drehende Rad sollte dabei die Möglichkeit bieten veränderliche Beschleunigungen einzustellen. Wie die in den Grundlagen Kapitel 3.3 dargestellte Gleichung 3.29 zeigt, ist die Beschleunigung proportional zum Radius und zum Quadrat der Winkelgeschwindigkeit, also sollten für den Versuchsaufbau eine oder bestenfalls beide Größen variabel einstellbar sein.

Die Veränderung der Winkelgeschwindigkeit lässt sich durch eine wandelbare Drehzahl realisieren, dies führt zu der Auswahl eines Gleichstrommotors, der über z.B. ein Labornetzgerät mit änderbaren Betriebsspannungen versorgt werden kann. Der Radius kann durch das Verschieben des Sensors auf einer Schiene abwechslungsreich gestaltet werden. Auch aus diesem Grund wurde ein Hinterrad von einem Fahrrad als Drehscheibe gewählt. Das Rad hat eine entsprechende Größe, ein integriertes Lager und bietet die Möglichkeit es über einen in der Felge laufenden Riemen anzutreiben.

8.1. Entwicklung des Testaufbaus

Für die Befestigung der Versuchsausrüstung wurde eine 40 mm dicke, 800 mm breite und 1200 mm lange Multiplex-Sperrholzplatte gefertigt, darauf wurden alle beweglichen Einzelkomponenten befestigt. Die Holzplatte selber konnte mit Schraubzwingen an einem handelsüblichen Bürotisch stabil aufgebracht werden. Die Tischplatte wurde in zwei Bereiche unterteilt, einen für den eigentlichen Teststand und einen für die notwendigen Elektronikgeräte für die Steuerung und Auswertung, dabei nimmt der Bereich der Geräte etwa ein Drittel des Platzes ein die restlichen zwei

8. Testaufbau zur Bestimmung der Position

Dritteln stehen für die Dreh-Apparatur zur Verfügung. In der Tabelle 8.1 sind die wichtigsten Komponenten und Materialien des Testaufbaus aufgeführt.

| Gerätetyp/ Materialtyp | Gerätbezeichnung/Werkstoff |
|--------------------------|------------------------------------|
| Oszilloskop (4-Kanal) | Yokogawa DL 1540 |
| Labornetzgerät (3-Kanal) | Toellner TOE 8733 |
| Gleichstrommotor | MFA/ COMO DRILLS 918D30112 |
| Lichtschranke | Panasonic EW PM-L24 |
| Rad (Fahrrad-Hinterrad) | BOC 26 Zoll |
| Riemenscheibe | Mercateo ALU 165 mm |
| Keilriemen | Mercateo 2200 mm |
| Aluminium-Profile | ALU 30 mm x 30 mm (diverse Längen) |
| Tischplatte | Holz 800 mm x 1200 mm x 40 mm |
| Blechprofil(1mm stark) | 75 mm x 20 mm x 100 mm |
| Wellenstahl | 12 mm x 120 mm |
| Kugellager | Stahl 10 mm x 28 mm, Bohrung 12 mm |
| Schraubzwingen | Mercateo 300 mm x 80 mm |
| Prüf- und Messleitungen | diverse Microklebs |

Tabelle 8.1.: Hauptkomponenten des Testaufbau

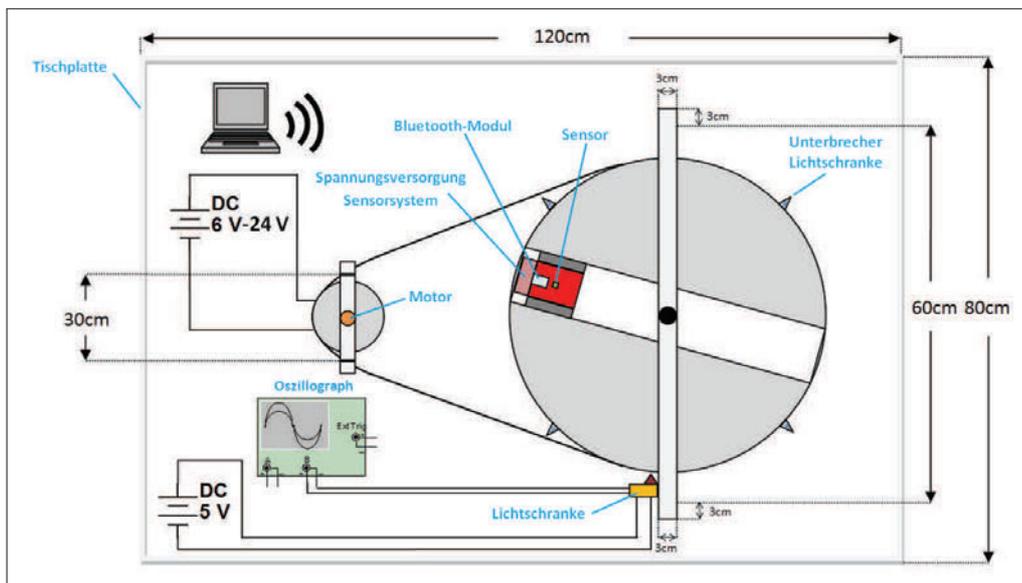


Abbildung 8.1.: Schematische Darstellung des Testaufbaus

Die Abbildung 8.1 zeigt eine grafische Darstellung zum Testaufbau. Die Darstel-

lung ist nicht maßstabsgerecht, zeigt aber schematisch die wichtigsten funktionellen Bestandteile. Für das Laufrad wurde mit den 30x30 mm breiten Aluprofilen eine 600 mm lange und 200 mm hohe Halterung konstruiert. Dafür wurden für die Aufhängung des Rades zwei Aussparungen in die Profile gefräst. Mit zwei auf die Nabe passenden Muttern konnte das Laufrad stabil auf dem Gestell befestigt werden. Das Gestell wurde durch drei in die Tischplatte geschraubten Gewindestifte verankert. Das Antriebsgestell, also die Aufhängung für die Riemenscheibe und den Antrieb wurde nach dem gleichen Verfahren angebracht.

Als Antrieb wurde ein Gleichstrommotor [42] verwendet. Die Motorwelle wurde mit einem 120 mm langen Wellenstahl zur Aufnahme der Riemenscheibe verlängert. Um die Welle drehbar am Gestell zu befestigen, wurde in das auf dem Tisch verankerte 300 mm lange Alu-Profil mit der Fräse eine Aussparung für ein Kugellager geschaffen. Die Riemenscheibe wurde mit zwei Gewindestiften auf der Welle positioniert. Der Gleichstrommotor konnte oberhalb der Riemenscheibe am Gestell verbaut werden. Entsprechend der im Laufrad resultierenden Zentrifugalbeschleunigung wurde der Gleichstrommotor ausgewählt, dabei sollten Beschleunigungswerte von bis zu 3 g zur Testung zu Verfügung stehen.

Das Übersetzungsverhältnis u der riemengetriebenen Vorrichtung entspricht dem

| U_B in V | T_U in s | a_{ZF} in m/s^2 | f_U in 1/s |
|------------|------------|----------------------------|--------------|
| 9,0 | 1,625 | 3,431 | 0,615 |
| 10,0 | 1,440 | 4,369 | 0,694 |
| 11,0 | 1,258 | 5,725 | 0,795 |
| 12,0 | 1,130 | 7,096 | 0,885 |
| 13,0 | 1,048 | 8,249 | 0,954 |
| 14,0 | 0,962 | 9,790 | 1,040 |
| 15,0 | 0,880 | 11,700 | 1,136 |
| 16,0 | 0,816 | 13,607 | 1,225 |
| 17,0 | 0,756 | 15,853 | 1,323 |
| 18,0 | 0,710 | 17,973 | 1,408 |
| 19,0 | 0,666 | 20,427 | 1,502 |
| 20,0 | 0,630 | 22,828 | 1,587 |
| 21,0 | 0,602 | 25,001 | 1,661 |
| 22,0 | 0,568 | 28,083 | 1,761 |
| 23,0 | 0,544 | 30,616 | 1,838 |
| 24,0 | 0,512 | 34,562 | 1,953 |

Tabelle 8.2.: Umdrehungsfrequenz des Laufrades im Testaufbau

Verhältnis der Radien von Abtrieb r_{Ab} zu Antrieb r_{An} .

Übersetzungsverhältnis u :

$$u = \frac{r_{Ab}}{r_{An}} \approx 3,4 \quad (8.1)$$

Mit der Lichtschranke PM-L24 [43] und am Laufrad angebrachten Unterbrechern wurde die Zeit T_U für eine Umdrehung entsprechend der am Motor über das Labornetzgerät Toe 8733 [44] eingestellten Betriebsspannungen U_B gemessen. Daraus ergaben sich die in der Tabelle 8.2 angegebenen Werte für die Zentrifugalbeschleunigung a_{ZF} und die Frequenz f_U .

Die Abbildung 8.2 zeigt die Gegenüberstellung der am Motor eingestellten Betriebsspannung und der Umdrehungsfrequenz am Laufrad.

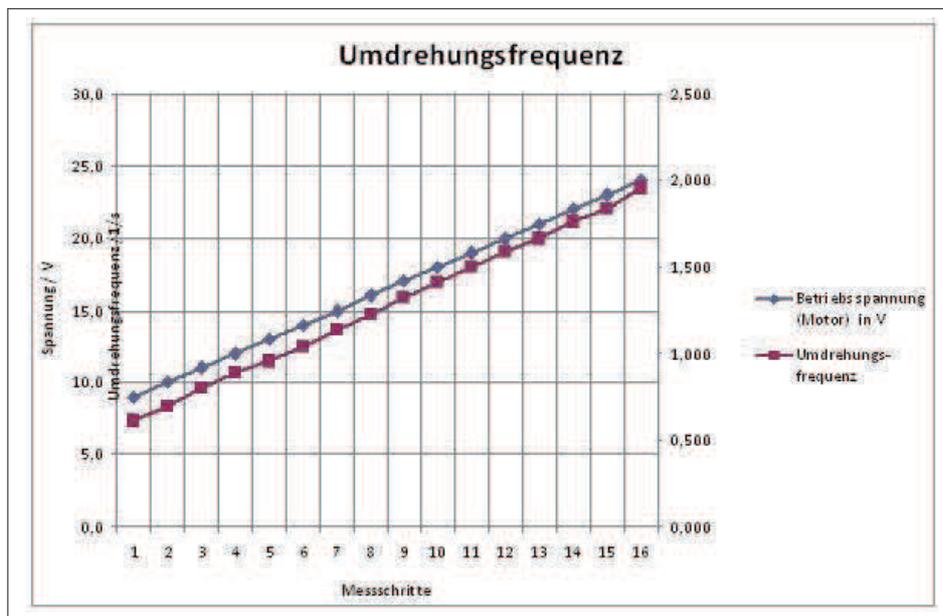


Abbildung 8.2.: Diagramm zur Umdrehungsfrequenz des Laufrades

Bei der Auswahl des Motors war die wichtigste Dimension die über die Betriebsspannung einstellbare Drehzahl. Dabei sollte der Motor über genügend Leistung verfügen, um das Laufrad anzutreiben. Der Motor wird mit Betriebsspannungen von 6 V bis 24 V betrieben. Das aufgesetzte Getriebe hat eine Abtriebsdrehzahl von 560 Umdrehungen pro Minute bei einem Übersetzungsverhältnis von 30:1. Zur Auswertung des Messsignals der Lichtschranke (siehe Abb. 8.3), welche mit einer Spannung von 5 V über das Labornetzgerät Toe 8733 [44] betrieben wurde, stand das Oszilloskop Yokogawa DL1540 [45] zur Verfügung. Mit dem Oszilloskop wurden die durch die am Rad befestigten Unterbrecher ausgelösten abfallenden Flanken des Lichtschranken-Signals detektiert und vermessen. Am Rad sind insgesamt vier

Unterbrecher angebracht, sodass jede fünfte abgefallene Flanke für eine Umdrehung steht. Um den Demonstrator für die Testreihen auf das Laufrad anzubringen, wur-



Abbildung 8.3.: Die Lichtschranke im Testaufbau

de eine Blechschiene mit Abmessungen 75 mm in der Breite, 550 mm in der Länge und 20 mm in der Höhe mit Schrauben auf der Felge des Rades befestigt. Dabei ermöglicht die Breite der Schiene eine Drehung des Demonstrators, sodass durch einfaches Umrüsten des Systems sowohl über die x-Achse-, als auch die y-Achse des Sensorsystems die Beschleunigung aus der Zentrifugalkraft detektiert werden kann. Um den Sensor mit der y-Achse längs in der Schiene auszurichten, wurden Abstandshalter aus Aluminium U-Profilen hergestellt. Die Abstandshalter haben eine Breite von 9 mm, eine Länge von 75 mm und eine Höhe von 20 mm. Mit ihnen konnte das Sensorsystem mittig in der Schiene positioniert werden. In der Schiene nach außen versetzt wurde die Spannungsversorgung für das Sensorsystem montiert. Die Abbildung 8.4 stellt den kompletten Testaufbau dar.

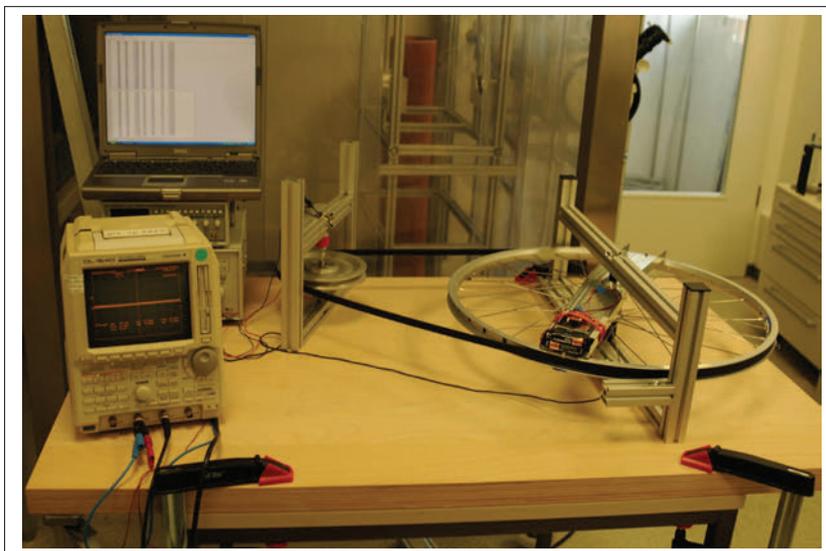


Abbildung 8.4.: Der komplette Testaufbau

8.2. Testdurchführung mit dem Sensorsystem MPU6050

Der Demonstrator wurde mit den dafür vorgesehenen Abstandshalter in die Schiene verbaut und zusätzlich mit zwei Gummibändern in der Schiene fixiert. Die Verbindungskabel für die Spannungsversorgung wurden ebenfalls mit den Gummibändern, und zusätzlich mit Klebestreifen an der Schiene befestigt. Die Abbildung 8.5 zeigt den verbauten Demonstrator.

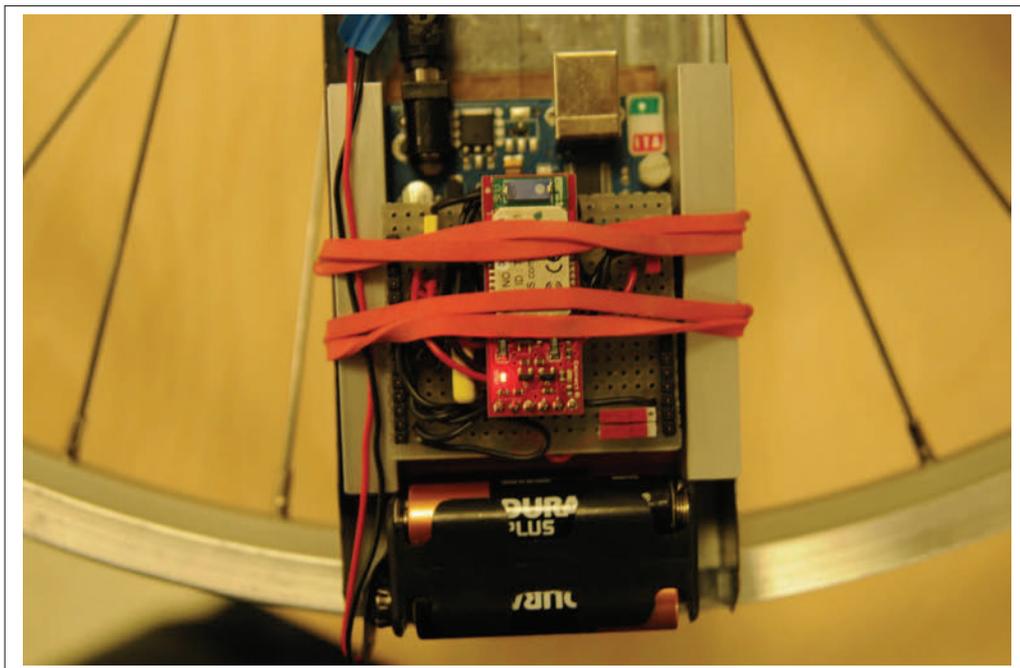


Abbildung 8.5.: Der Demonstrator in der Schiene

Mit dem Versuchsaufbau, wie er in Abbildung 8.4 dargestellt ist, wurden die ersten Tests bezüglich der Detektion der Zentrifugalbeschleunigung durchgeführt. Darüber hinaus konnte durch Messung der Erdbeschleunigung bestimmt werden, ob der Sensor in der Schiene richtig ausgerichtet ist. Dann wurden Testläufe mit unterschiedlichen Umdrehungen des Laufrades gestartet. Einer dieser Testläufe diente zur Überprüfung des Berechnungsalgorithmus. Die Ergebnisse der Testläufe sind im folgenden Kapitel 9 dokumentiert und entsprechend ausgewertet.

9. Ergebnisse

In vorherigen Kapiteln wurden Tests mit zwei verschiedenen Sensortypen durchgeführt. Die Tests fanden ausschließlich unter Laborbedingungen statt, sodass äußere Einflüsse und Einwirkungen wie Temperaturschwankungen auf die Sensoren vorläufig nicht überprüft wurden. Beide Sensortypen haben voneinander abweichende Parameter, welche in der Tabelle 9.1 zusammenfassend aufgeführt sind.

| | Hersteller | Analog Devices | InvenSense |
|---------------------------|------------|------------------|--------------------|
| | Produkte | ADXL 335 | MPU 6050 |
| | | Acc | Acc/Gyro |
| Versorgungsspannung [V] | | 1,8 V - 3.6 V | 2,375V - 3,46V |
| Versorgungsstrom [mA] | | 0,35 | 3,9 |
| Anschluss zur Auswertung | | analog | digital |
| Baugröße(LxBxH) [mm] | | 4,0 x 4,0 x 1,45 | 4,0 x 4,0 x 0.9 |
| Nichtlinearität [%] | | ±0,3 | ±0,5/±0,2 |
| Temperaturspanne [°C] | | -40 bis +85 | -40 to +85 |
| Messspanne [g] | | ±3 | ±2,4,8,16 |
| Empfindlichkeit (ACC) | | 300 mV/g | 16384 LSB/g |
| Messspanne [°/s] | | - | ±250,500,1000,2000 |
| Empfindlichkeit (GYRO) | | | 131 LSB/(°/s) |
| Empfindl.-Änderung [%/°C] | | ±0,01 | ±0,02 |
| Resonanz Frequenz [kHz] | | 5,5 | - |
| Empfindl.-Drehachse [%] | | pm1 | pm2 |
| Datenrate Ausgang [Hz] | | 550 | 1000 |
| Auflösung [mg] | | 5,8594 | 0,0610 |
| Auflösung [circ] | | - | 0,0076 |

Tabelle 9.1.: Vergleichsparameter ADXL335 MPU6050 [38][39]

Der ADXL335-Beschleunigungssensor wurde ausgewählt, weil er derzeit weit verbreitet genutzt wird. Er ist beispielsweise in der *WiiRemote* verbaut. Die ersten *Gehversuche* mit dem ADXL335 bestanden aus den Betrachtungen der Sensorwerte bei einfachen linearen Verschiebungen und der Detektion der Erdbeschleunigung auf den einzelnen Achsen x, y und z durch Drehung des Sensors. Dabei zeigte sich ein gutes Ansprechverhalten, eine Zeitmessung zwischen Ausrichtung des Sensors

und der Ausgabe des detektierten Wertes war mit herkömmlichen Messmethoden nicht möglich. Die Verarbeitungszeiten lagen im Bereich weniger Millisekunden. Diese Werte begünstigen nahezu eine Echtzeitmessung. Im weiteren Verlauf wurden die Daten über ein Terminal-Programm, das *Hyper Terminal* ausgelesen und in *Excel* konvertiert, sodass die Messwerte visualisiert werden konnten. Es wurden mit dem sich in Ruhelage befindlichen ADXL335 50000 Messwerte aufgenommen, dies entsprach bei einer Abtastrate von 200 Hz der Gesamtzeit von etwas über vier Minuten. Im Folgenden sind in der Abbildung 9.1 exemplarisch die für die x-Achse aufgenommenen Daten dargestellt. Anhand der aufgenommenen Daten

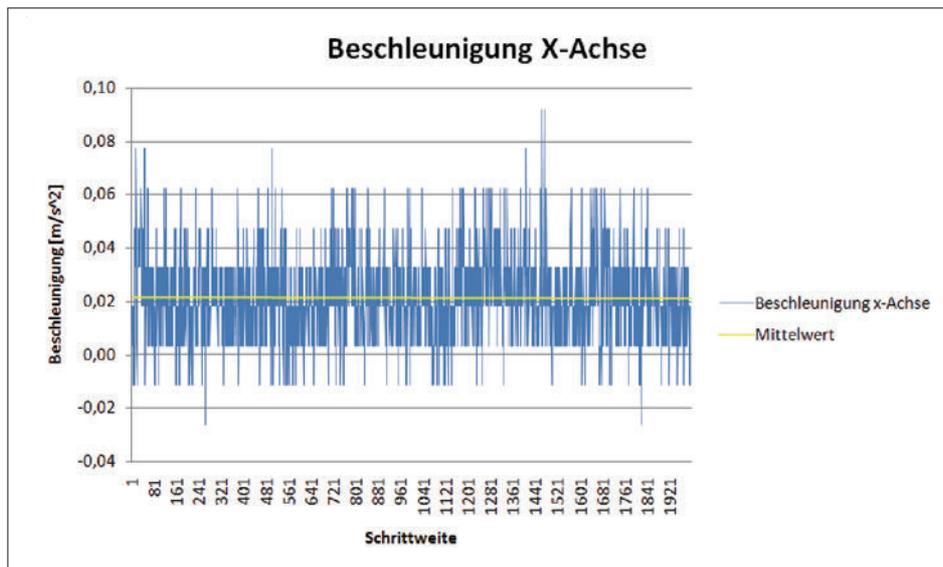


Abbildung 9.1.: Diagramm ADXL335 Beschleunigung x-Achse

lässt sich gut erkennen, dass der Sensor in Ruhelage ein Rauschen aufweist. Ebenso ist Offset zu verzeichnen. Diese Nullpunkts-Abweichung kann nicht als typischer Bias, auch Nullpunktfehler bezeichnet werden, da die Wahrscheinlichkeit, dass der Sensor Anteile aus der Erdbeschleunigung detektiert sehr hoch ist. Das Rauschen in den aufgenommenen Messwerten kann unterschiedliche Ursachen haben, zum einen durch externe Vibrationen, welche bei einer Bewegung noch deutlich höher ausfallen können und zum anderen auch durch eine nicht konstante Versorgungsspannung entstehen. Die Ursachen von Abweichungen wurden in dem vorliegenden Fall nicht näher betrachtet. Von der aufgenommenen Messreihe wurde die Standardabweichung S_{Abw} der Einzelmessungen bestimmt nach folgender Gleichung mit der Anzahl q der Einzelmessungen i , den Messwerten x_i und dem Mittelwert x_m der Einzelmessungen.

Standardabweichung der Einzelmessungen [5]:

$$S_{Abw} = \sqrt{\frac{\sum_{i=1}^q \cdot (x_i - x_m)^2}{q - 1}} = 0,01936 \frac{m}{s^2} \quad (9.1)$$

Die Standardabweichung ist Maß für Streuung der Messwerte um ihren Mittelwert. Die Messwerte der Sensoren können mit einer Reihe von Abweichungen behaftet sein. Für Inertialsensoren typische Abweichungen sind [17]:

- Der *Bias-Nullpunktsfehler* ist die Abweichung von Null des gemessenen Wertes, wenn der Sensoreingang Null ist.
- Die *Nichtlinearität* ist die Abweichung, bei einem linearen Eingang zu einem nicht linearen Ausgangssignal. Sie ist bei den verwendeten Sensoren mit Werten zwischen $\pm 0,2\%$ und $\pm 0,5\%$ angegeben. In der Regel fällt sie bei Drehratensensoren etwas höher aus als bei Beschleunigungssensoren.
- Der *Quantisierungsfehler* ist die Abweichung die bei der Digitalisierung des Messsignals auftreten. Den Fehler beeinflusst maßgeblich die Auflösung des AD-Wandlers, das Sensorsignal des ADXL335 wurde mit einem 10 bit AD-Wandler des *ATmega328P* konvertiert und der MPU6050 hat einen internen Wandler mit 16 bit, wobei die daraus resultierende Abweichung deutlich geringer ist.
- Der *Skalenfaktorfehler/ Skalenfaktorasymmetrie* ist die Abweichung im Skalenfaktor des Sensors, bedingt durch Alterung und Herstellungstoleranzen des Sensors.
- Die *Totzone* ist die Abweichung, eines sehr kleinen Eingangswertes, bei dem noch kein Ausgangswert entsteht.

Die meisten Abweichungen sind stark temperaturabhängig. Der MPU6050 beinhaltet beispielsweise einen Temperatursensor, dessen Messwerte bei einer Kalibration berücksichtigt werden können. [17]

Nach den ersten Arbeiten mit dem ADXL335 wurde über die Einbindung eines Drehratensensors in das System mit einem ADXL335-Beschleunigungssensor geforscht. Dabei traten einige Problemstellungen auf, welche sich mit den im Labor vorhandenen Möglichkeiten nur schwer realisieren lassen. Um für eine inertielle Messeinheit unter Verwendung von Beschleunigungs- und Drehratensensoren genaue Werte für die Berechnung zu erhalten, setzt dies eine sehr genaue Ausrichtung beider Sensoren voraus. Dabei müssen die sensitiven Achsen der beiden Sensoren aufeinander liegen und die Achsen x, y und z orthogonal zueinander ausgerichtet sein. Diese Ausrichtung lässt sich nur mit einem hohen technischen Aufwand

realisieren. Diese Erkenntnis führte zu der Entscheidung, nach einem bereits ausgerichteten System mit beiden Sensoren zu suchen. Auf dem Markt sind derzeit viele Komplettsysteme erhältlich. Insbesondere fiel aber unter diesen Systemen das Sensorsystem von InvenSense der MPU6050 auf. In diesem System sind beide Sensortypen integriert, er bringt gute Voraussetzungen zur Einbindung in das Gesamtsystem mit, wie z.B. eine kompakte Baugröße und eine gut dimensionierte Versorgungsspannung, welche problemlos vom Mikrocontrollerboard bereitgestellt werden kann. Er stellt ein digitales Ausgangssignal mit einer hohen Auflösung bereit, welches über einen I^2C -Datenbus an den Mikrocontroller übertragen werden kann. Des Weiteren weist er für die Anwendung in der vorliegenden Arbeit passende einstellbare Messbereiche auf, bietet die Möglichkeit durch Anschlüsse für externe Sensoren z.B. einen Kompass anzuschließen und er beinhaltet einen *Digital Motion Processor* mit integrierten Algorithmen, welcher die erfassten Daten als Rotationsmatrix, Quaternion, Eulerwinkel oder im Rohdatenformat bereitstellt. In der vorliegenden Arbeit wurden die Daten bislang nur im Rohdatenformat verarbeitet und über den in *Excel* implementierten Algorithmus ausgewertet. Ein Nachteil ergab sich für diese Abhandlung jedoch aus dem Problem der Verfügbarkeit eines Evaluation-Boards, sodass etwas Zeit für die eigenständige Entwicklung investiert werden musste, um mit den Tests beginnen zu können. In Abbildung 9.2 sind die Rohdaten der y-Achse exemplarisch aufgeführt.

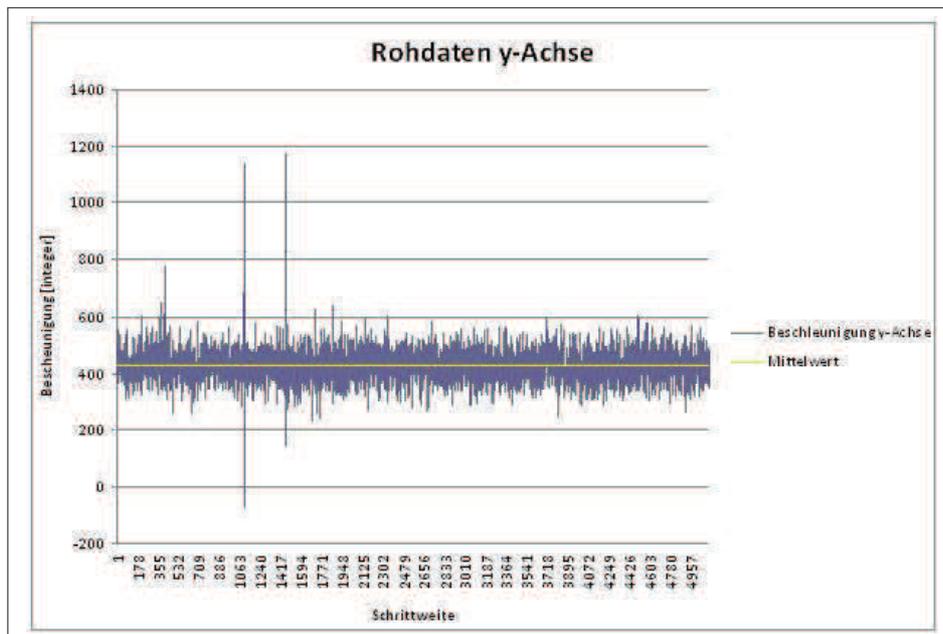


Abbildung 9.2.: Diagramm MPU6050 Beschleunigung y-Achse

Bei den Daten sind deutlich einige Werte zu erkennen, die nicht zum restlichen Teil der Messreihe passen. Derartige Werte können im Verlauf immer wieder auftreten, sie sind charakteristisch für MEMS-Sensoren. Es gilt, diese über einen Filter, beispielsweise über einen *Kalman*-Filter zu eliminieren, die Implementierung eines Filters ist für folgende Arbeiten zu empfehlen. Anbei tritt auch bei dieser Messreihe ein gewisses Rauschen auf, auch das ist typisch für diese Sensor-Arten.

Im folgenden wurde versucht die Erdbeschleunigung auf einer Achse zu detektieren, dies ist in der Abbildung 9.3 aufgeführt. Dabei ist die x-Achse nahezu korrekt auf der Senkrechten zum Erdbeschleunigungsvektor. Die y-Achse ist um etwa $2,54^\circ$ verdreht und die z-Achse demzufolge auch. An der Stelle würde es Sinn machen mit einem Referenzsystem, in Form einer zweiten Messeinheit diese Werte zu überprüfen, da eine fehlerhafte Ausrichtung bei der Herstellung der Sensorachsen untereinander auch zu zusätzlichen Abweichungen führen kann.

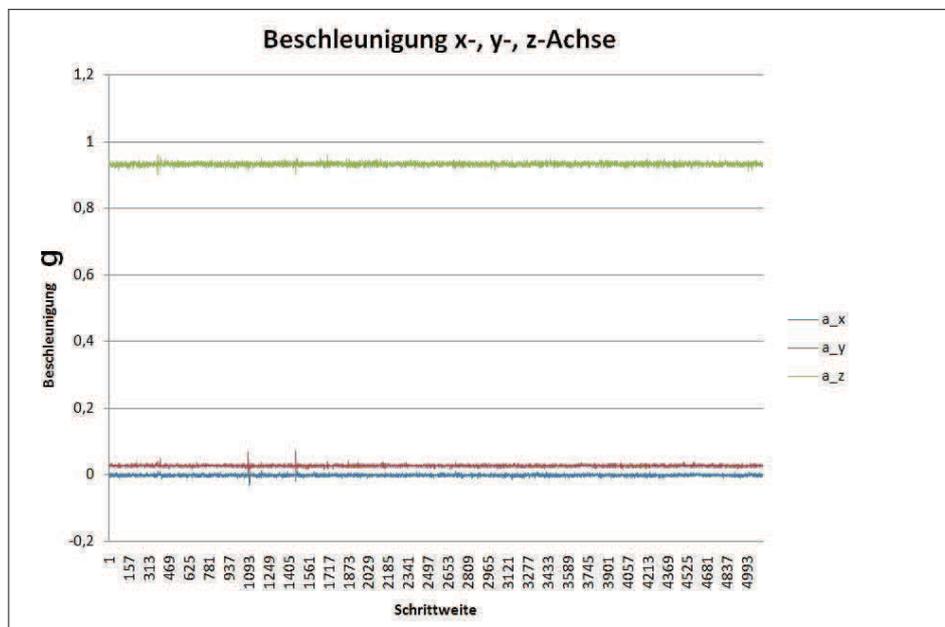


Abbildung 9.3.: Diagramm Detektion der Erdbeschleunigung mit MPU6050

In der Abbildung 9.3 ist zu erkennen, dass die in Kapitel 7.1 angesprochene manuelle Ausrichtung der z-Achse auf den Erdbeschleunigungsvektor nicht zu vollen Teilen gelingt, anhand der Gleichungen 7.8 und 7.9 kann aber eine Berechnung der Winkel erfolgen. Diese Winkel gehen dann als Ausgangslage in den Berechnungsalgorithmus ein. Der Berechnungsalgorithmus wurde mit simulierten Daten getestet. Dabei sollte eine Kreisbahn durchlaufen werden, die Schritte der Berechnung sind in Kapitel 7.3 aufgeführt. Das in Abbildung 9.4 folgende Diagramm verdeutlicht

das Ergebnis der Simulation. Darauf sind die vom Hauptrechner über den Algorithmus berechneten Wegstrecken s_x und s_y jeweils auf einer Achse dargestellt.

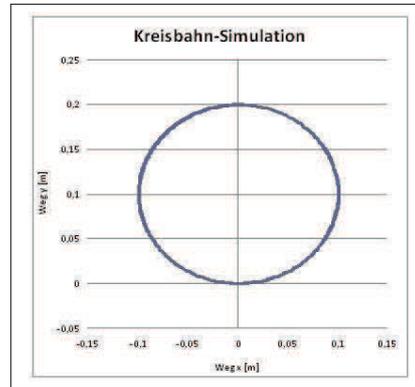


Abbildung 9.4.: Diagramm Simulation Test des Berechnungsalgorithmus

Für den ersten Testdatensatz mit dem Teststand wurde der Aufbau aus der Ruhelage heraus auf eine Winkelgeschwindigkeit von $225 \frac{\circ}{s}$ beschleunigt. Die Winkelgeschwindigkeit wurde für etwa 45 s detektiert, wobei ein Messschritt etwa 5 ms dauert. Wie in Abbildung 9.5 ersichtlich wurde eine Zentrifugalbeschleunigung auf der y-Achse hervorgerufen.

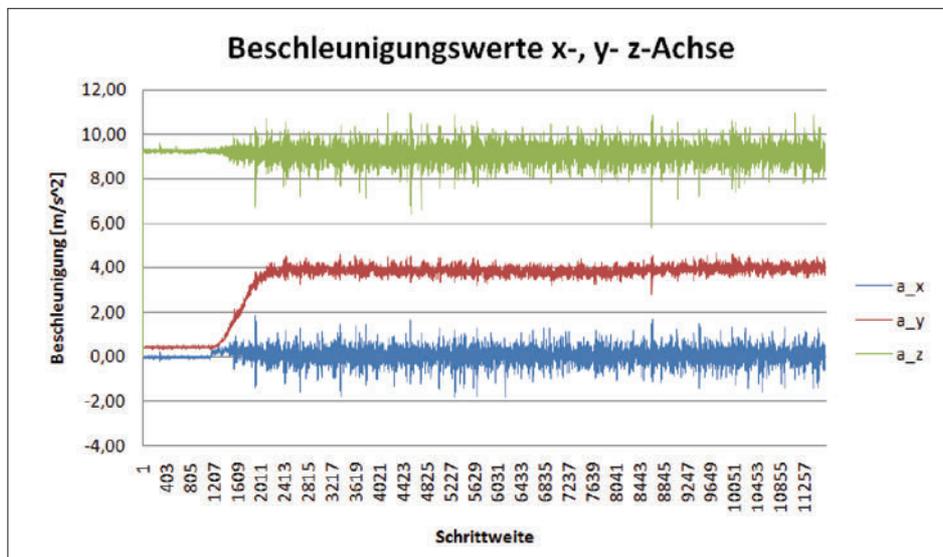


Abbildung 9.5.: Diagramm Zentrifugalbeschleunigung auf y-Achse MPU6050

Die Beschleunigungswerte weisen ein hohes Rauschen auf, dies ist zum größten Teil bedingt durch die vom Testaufbau bei der Rotation hervorgerufenen Vibrationen. Die Vibrationen fallen auf der x- und z-Achse höher aus, was durch den stabileren Lauf in y-Richtung, durch gute Befestigung des Laufrades bedingt ist. Die Vibrationen auf der x-Achse resultieren im Wesentlichen aus dem Antrieb über den Riemen und die Vibrationen auf der y-Achse entstehen aus der Unwucht der Felge heraus.

In der Abbildung 9.6 sind die Drehraten der einzelnen Achsen x, y und z dargestellt.

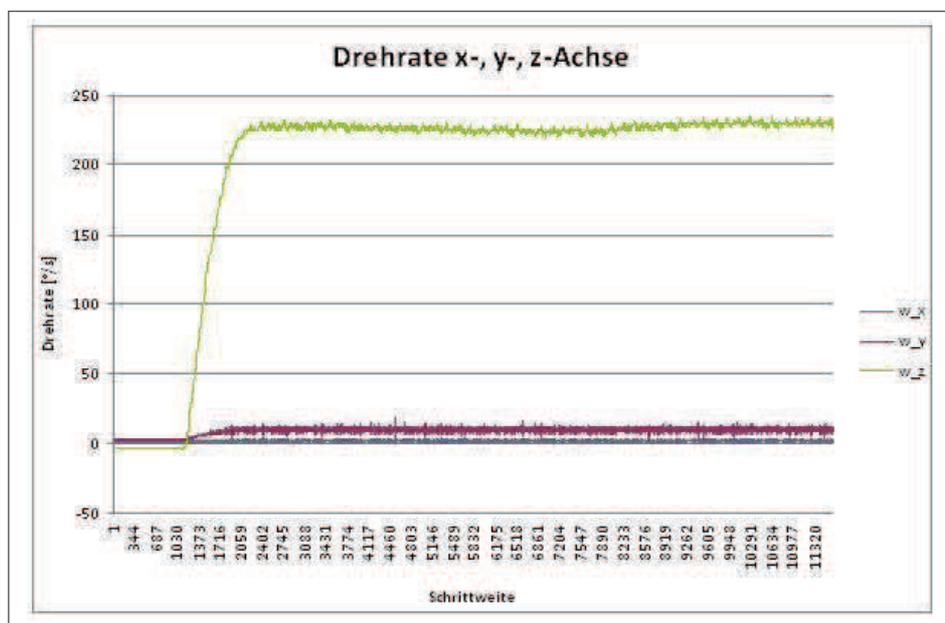


Abbildung 9.6.: Diagramm Drehraten-Achsen x, y und z MPU6050

In dem Diagramm ist ersichtlich, dass eine Drehung mit einer Drehrate von etwa $10 \frac{\circ}{s}$ abläuft. Diese Drehung resultiert aus der Nicht korrekt ausgerichteten Anfangslage. In die Berechnung der Position geht diese Drehung ein. Über den Berechnungsalgorithmus sollte diese Anfangslage korrigiert werden, was bis zum Ende der Arbeit nicht gelungen ist.

10. Zusammenfassung

In der vorliegenden Arbeit wurde ein Demonstrator zur Positionsbestimmung mittels Inertial-Gesetze entwickelt. Ein *Arduino*-Mikrocontrollerboard dient dabei als Basis für den Demonstrator, in welchen ein austauschbares Sensormodul und ein Bluetooth Funkübertragungsmodul implementiert wurden. Dabei ist der Demonstrator so konzipiert, dass er kabellos und tragbar verwendet werden kann. Die ausgewählte Sensoreinheit, bestehend aus einem Drei-Achsen-Beschleunigungssensor und einem Drei-Achsen-Drehratensensor, kommuniziert über einen I^2C -Datenbus mit dem Mikrocontroller-Board. Das Bluetooth Modul leitet die Sensordaten an einen tragbaren Rechner weiter.

Der Rechner dient dabei der Programmierung des Mikrocontrollers und der Auswertung der Sensorsignale. Er enthält die dafür relevanten Softwaretools, wie die *Arduino*-Entwicklungsumgebung, für die Programmierung des Mikrocontrollers, und ein Terminal-Programm, das *Hyperterminal* für die Anzeige der Sensorsignale. Es wurde versucht die Sensordaten mit *Processing* darzustellen. Dies führte zu einer nicht annehmbaren zeitlichen Verzögerung, welche eine Echtzeitmessung nicht möglich machte. So wurde der Berechnungsalgorithmus in *Excel* implementiert, darüber die Rohdaten umgerechnet und die ersten Tests grafisch dargestellt. Darüber hinaus wurde ein Teststand entwickelt mit dem der Demonstrator und der auf dem Rechner implementierte Algorithmus überprüft wurde. Dabei kann der Demonstrator durch Einsetzen in eine Schiene auf einem Laufrad Kreisbahnen durchlaufen mit variabel einstellbaren Winkelgeschwindigkeiten und Kreisradien. Die Einstellung der Winkelgeschwindigkeit lässt sich durch das Verstellen der Betriebsspannung des Antriebsmotors realisieren, wobei die Umdrehung des Laufrades über eine Lichtschranke kontrolliert wird. Die Variabilität im Kreisradius entsteht durch das Verschieben des Demonstrators in einer auf dem Laufrad befestigten Schiene.

Nach der Fertigstellung der einzelnen Elemente wurde der Demonstrator ersten Tests unterzogen. Die Tests zeigten eine funktionierende Sensoreinheit, allerdings mit einem abweichungsverursachenden Berechnungsalgorithmus, welche bis zum Ende der Arbeit nicht endgültig analysiert und behoben werden konnten. In der aktuell entwickelten Version verfügt das System über keine Bewegungsmustererkennung. In der Zukunft kann eine Erweiterung und Verbesserung im Algorithmus am System vorgenommen werden. Langfristig ist dann eine Analyse der Bewegung möglich.

Literaturverzeichnis

- [1] Worcester Polytechnic Institute: Precision Personnel Locator Project (aktualisiert 26.07.2012) Internet: <http://www.ece.wpi.edu./Research/PPL/> (Zugriff: 27.07.2012)
- [2] Schantz, Hans G. (Q-TRACK®): Near-Field Electromagnetic Ranging (NFER®) Real-Time Location System. (2011) Internet: <http://www.q-track.com/index.php/location-innovation/technical-papers> (Zugriff: 27.07.2012)
- [3] Bauer, Martin uni-protokolle.de V.i.S.d.P.: Lokalisation Lexikon. Internet: <http://www.uni-protokolle.de/Lexikon/Lokalisation.html> (Zugriff: 20.07.2012)
- [4] Lüthi, P.; Moser, T.; ETH Zürich: Low Cost Inertial Navigation System Studien-Arbeit (2000)
- [5] Papula, Lothar: Mathematische Formelsammlung, 10.Auflage ©Vieweg+Teubner (ISBN 978-3-8348-0757-1) (2009)
- [6] Tipler, Paul A.; Mosca, Gene: Physik-Für Wissenschaftler und Ingenieure, 2.Auflage ©Elsevier GmbH, Spektrum Akademischer Verlag (ISBN 3-8274-1164-5) (2006)
- [7] Wendel, Jan: Integrierte Navigationssysteme-Sensordatenfusion, GPS und Inertiale Navigation, 2.Auflage ©Oldenbourg Wissenschaftsverlag GmbH (ISBN 978-3-486-70439-6) (2011)
- [8] EUROCONTROL-European Organization for Safety of Air Navigation: WGS 84 Implementation Manuel, Version 2.4, February 12, 1998 Internet: <http://www.eurocontrol.int> (Zugriff: 19.06.2012)
- [9] Gerlach, Gerald; Dötzel, Wolfgang; u.a.: Grundlagen der Mikrosystemtechnik, ©Carl Hanser Verlag München (ISBN 3-446-18395-7) (1997)
- [10] Botthof, A.; u.a.: Innovation durch Mikointegration: Intelligente Produkte über Systemintegration von Mikro-, Bio- und Nanotechniken, VDI/VDE-Technologiezentrum Informationstechnik GmbH (ISBN 978-3897500907) (1998)

- [11] Kasper, M.: Mikrosystementwurf-Entwurf und Simulation von Mikrosystemen, ©Springer-Verlag (ISBN 3-540-66497-1) (2000)
- [12] Fischer, Wolf-Joachim: Mikrosystemtechnik, 1.Auflage ©Vogel Verlag (ISBN 3-8023-1772-6) (2000)
- [13] Ballas, R.G.; u.a.: Elektromechanische Systeme in der Mikrotechnik und Mechatronik-Dynamischer Entwurf-Grundlagen und Anwendungen, 2.Auflage ©Springer Verlag (ISBN 978-3-540-89317-2) (2009)
- [14] Kories, Ralf; Schmidt-Walter, Heinz: Taschenbuch der Elektrotechnik, 8.Auflage ©Wissenschaftlicher Verlag Harri Deutsch GmbH (ISBN 978-3-8171-1830-4) (2008)
- [15] D.H. Titterton, D.H.; Weston, J.L.: Strapdown Inertial Technology-Second Edition, ©The Institution of Electrical Engineers Reprinted (ISBN 978-1-56347-693-8) (2009)
- [16] Acar, C.; University of California: Robust Micromachined Vibratory Gyroscopes, PhD thesis (Irvine 2004)
- [17] Winkler, Stefan: Zur Sensordatenfusion für integrierte Navigationssysteme unbemannter Kleinflugzeuge, ©Shaker Verlag Aachen (ISBN 978-3-8322-6060-6) (2007)
- [18] Creative Commons Deutschland: Was ist CC? Internet: <http://de.creativecommons.org/was-ist-cc/> (Zugriff: 11.06.2012)
- [19] Odendahl, Manuel; u.a.: Arduino-Physical Computing für Bastler, Designer und Geeks, 1.Auflage ©O'Reilly Verlag (ISBN:978-3-89721-893-2) (2009)
- [20] Schmidt, Maik: Arduino-Ein schneller Einstieg in die Microcontroller-Entwicklung, 1.Auflage ©dpunkt.verlag GmbH (ISBN:978-3-89864-764-9) (2012)
- [21] Margolis, Michael: Arduino Cookbook-Second Edition 2.Auflage ©O'Reilly Verlag (ISBN:978-1-449-31387-6) (2011)
- [22] Arduino Homepage: Software Internet: <http://www.arduino.cc/en/Main/Software> (Zugriff: 16.06.2012)
- [23] Arduino Homepage Internet: <http://www.arduino.cc> (Zugriff: 17.06.2012)
- [24] Arduino Homepage: Programmer Internet: <http://www.arduino.cc/en/Hacking/Programmer> (Zugriff: 17.06.2012)

- [25] Arduino Homepage: BuildProcess Internet:
<http://www.arduino.cc/en/Hacking/BuildProcess> (Zugriff: 17.06.2012)
- [26] Arduino Homepage: Hardware Internet:
<http://www.arduino.cc/en/Main/Hardware> (Zugriff: 19.06.2012)
- [27] Freeduino: Arduino-Programmier-Handbuch Internet:
<http://www.freeduino.de/books/arduino-programmier-handbuch> (Zugriff: 06.07.2012)
- [28] Atmel: Datenblatt-ATmega328P Internet:
<http://www.atmel.com/Images/doc8161.pdf>- Vereinigte Staaten (Zugriff: 15.07.2012)
- [29] Atmel: Datenblatt-ATmega8U2. Internet:
<http://www.atmel.com/Images/doc7799.pdf>- Vereinigte Staaten (Zugriff: 15.07.2012)
- [30] Future Technology Device International Ltd.: Datenblatt-FTDI232R Internet: <http://www.ftdichip.com/Products/ICs/FTDI232R.htm> (Zugriff: 15.07.2012)
- [31] Wiring Internet: <http://www.wiring.org.co> (Zugriff: 17.06.2012)
- [32] Sparkfun: Arduino Uno Internet: <http://www.sparkfun.com/products/9950> (Zugriff: 20.06.2012)
- [33] IIC: I²C-Inter-Integrated Circuit Internet: <http://www.i2c-bus.org> (Zugriff: 05.07.2012)
- [34] MCT Paul und Scherer: SPI-Serial Peripheral Interface Internet: <http://www.mct.de/faq/spi.html> (Zugriff: 05.07.2012)
- [35] Sparkfun: Datenblatt-Bluetooth Modul BlueSMIRF. Internet: <http://www.sparkfun.com/products/10268> (Zugriff: 17.07.2012)
- [36] Faludi, Robert: Building Wireless Sensor Networks ©O'Reilly Verlag (ISBN 978-0-596-80773-3) (2011)
- [37] Farnell: Analog Devices ADXL335-Eval-Board Internet: <http://www.de.farnell.com/analog-devices/eval-adxl335z/kit-eval-f-adxl335-accelerometer/dp/1699046> (Zugriff: 17.07.2012)
- [38] Analog Devices: Datenblatt-Analog Devices ADXL335 Internet: <http://www.analog.com/en/mems-sensors/mems-inertial-sensors/adxl335/products/produkt.html> (Zugriff: 17.07.2012)

- [39] InvenSense: Datenblatt-InvenSense MPU6050 Internet:
<http://www.invensense.com/mems/gyro/mpu6050.html> (Zugriff:
17.07.2012)
- [40] Jeff Rowberg; I^2C devlib: I^2C -Library. Internet:
<http://i2cdevlib.com/usage> (Zugriff: 18.07.2012)
- [41] Jeff Rowberg; I^2C devlib: MPU6050-Library. Internet:
<http://i2cdevlib.com/usage> (Zugriff: 18.07.2012)
- [42] MFA COMO DRILLS: Datenblatt DC-
Motor MFA/COMO-Drills 918D30112 Internet:
<http://www.mfacomodrills.com/gearboxes/918dseries.html> (Zugriff:
20.07.2012)
- [43] Farnell: Datenblatt-Lichtschanke Panasonic EW PM L24 Inter-
net: <http://uk.farnell.com/panasonic-ew/pm-l24/optoswitch/dp/4461344>
(Zugriff: 20.07.2012)
- [44] Toellner: Datenblatt-Labornetzgerät Toellner Toe8733 Internet:
[http://www.toellner.de/html/pages/de-startseite-produkt-netzgeraete-
toe8732-8735.htm](http://www.toellner.de/html/pages/de-startseite-produkt-netzgeraete-toe8732-8735.htm) (Zugriff: 20.07.2012)
- [45] Singer, Helmut: Datenblatt Oszilloskop Yokogawa DL1540 Inter-
net: <http://www.helmut-singer.de/stock/-1035136129.html> (Zugriff:
20.07.2012)

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und ohne fremde Hilfe angefertigt zu haben. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quelle kenntlich gemacht.

Dirk Dreffkorn

Version 2.4
February 12, 1998

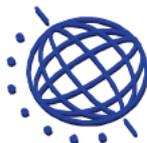
WGS 84 IMPLEMENTATION MANUAL

Prepared by



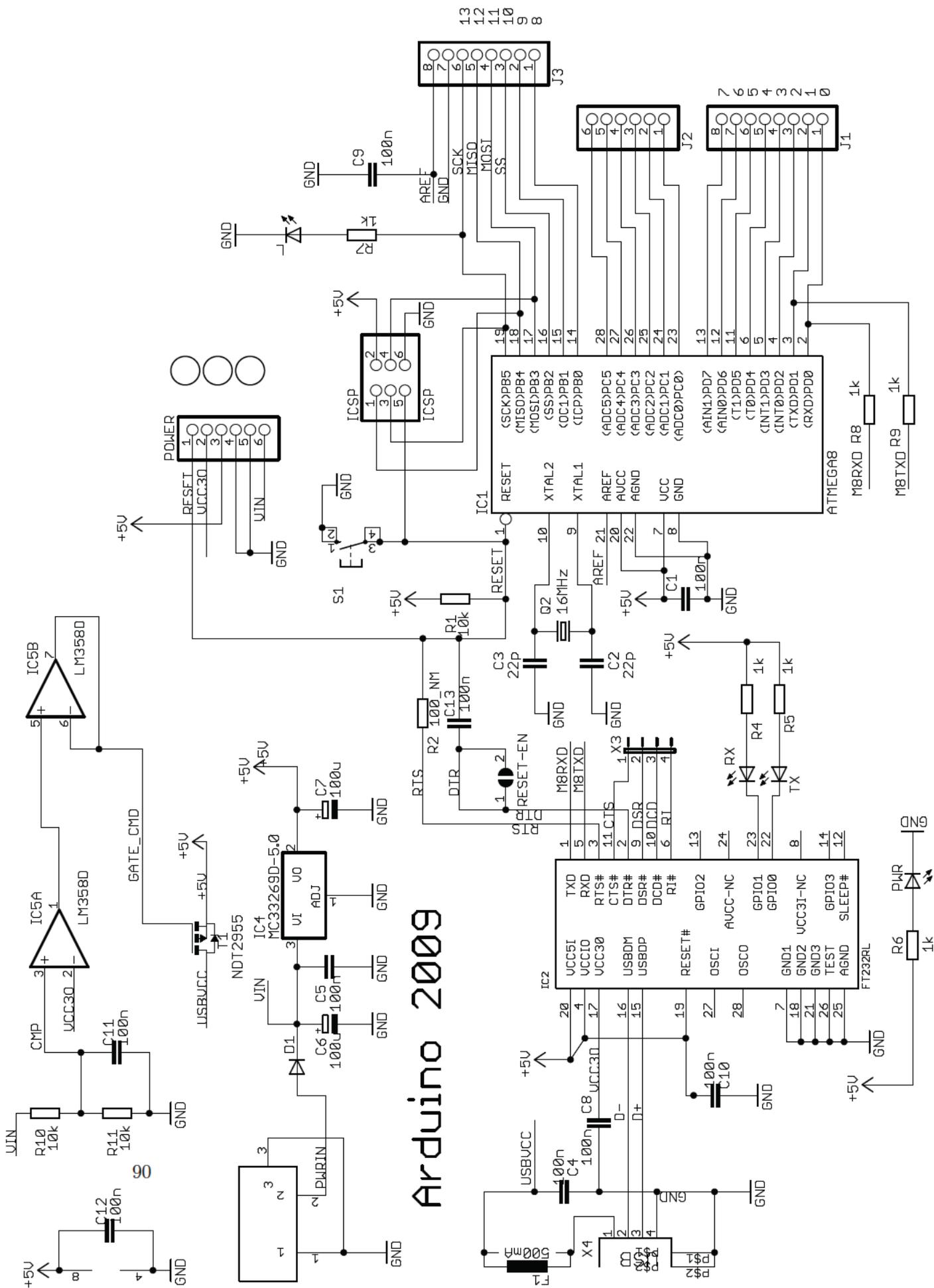
EUROCONTROL
European Organization for the Safety of
Air Navigation
Brussels, Belgium

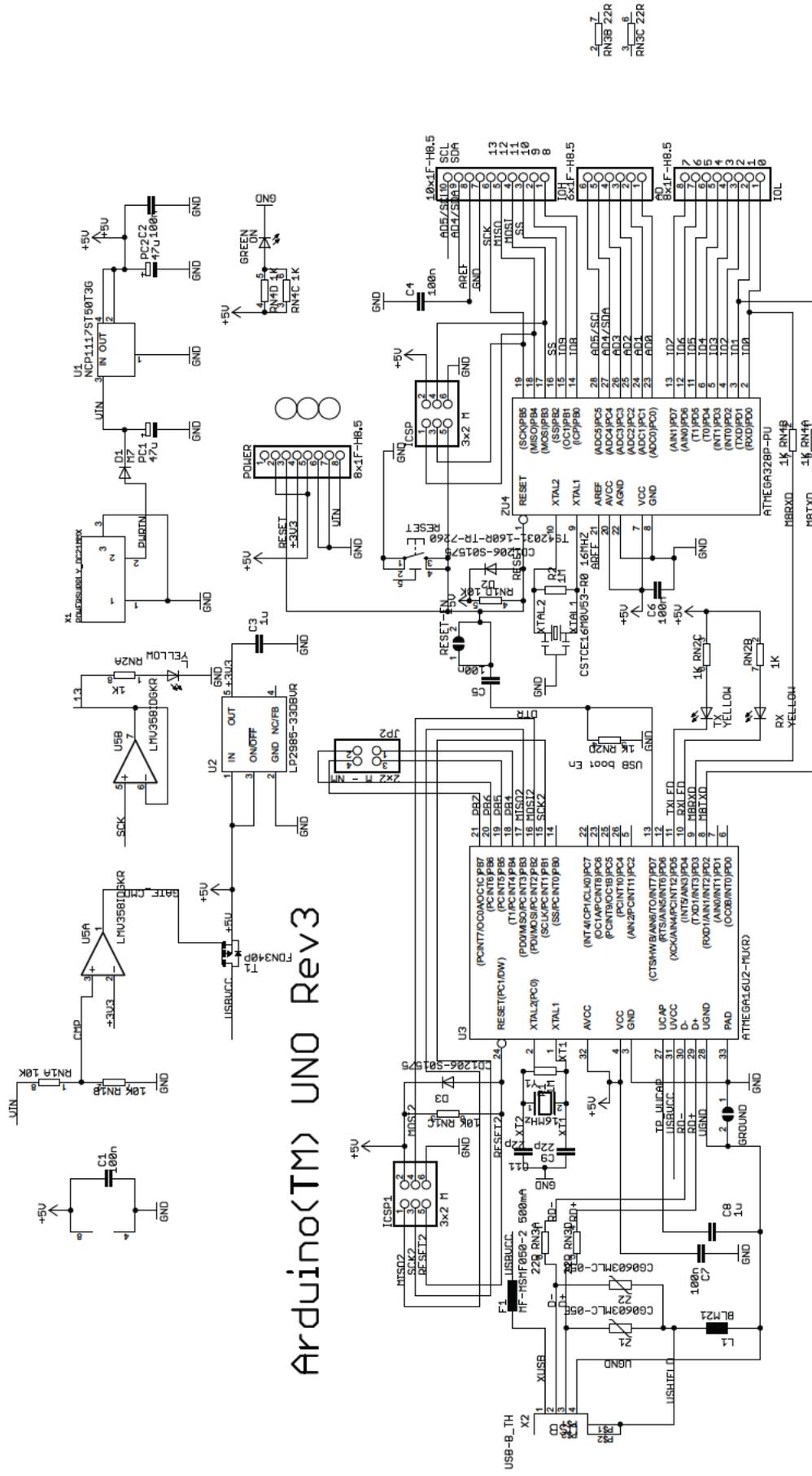
and



IfEN
Institute of Geodesy and Navigation (IfEN)
University FAF Munich, Germany

Arduino 2009





Arduino(TM) UNO Rev3

Reference Designs ARE PROVIDED "AS IS" AND "WITH ALL FAULTS. Arduino DISCLAIMS ALL OTHER WARRANTIES, EXPRESS OR IMPLIED, REGARDING PRODUCTS, INCLUDING BUT NOT LIMITED TO, ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Arduino may make changes to specifications and product descriptions at any time, without notice. The Customer must not rely on the absence or characteristics of any features or instructions marked "reserved" or "undefined." Arduino reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them. The product information on the Web Site or Materials is subject to change without notice. Do not finalize a design with this information. ARDUINO is a registered trademark.

Use of the ARDUINO name must be compliant with <http://www.arduino.cc/en/Main/Policy>

Features

- High Performance, Low Power AVR[®] 8-Bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 20 MIPS Throughput at 20 MHz
 - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
 - 4/8/16/32K Bytes of In-System Self-Programmable Flash program memory (ATmega48PA/88PA/168PA/328P)
 - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
 - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
 - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
 - Data retention: 20 years at 85°C/100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Six PWM Channels
 - 8-channel 10-bit ADC in TQFP and QFN/MLF package
 - Temperature Measurement
 - 6-channel 10-bit ADC in PDIP Package
 - Temperature Measurement
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Byte-oriented 2-wire Serial Interface (Philips I²C compatible)
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 23 Programmable I/O Lines
 - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
 - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
 - 40°C to 85°C
- Speed Grade:
 - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
 - Active Mode: 0.2 mA
 - Power-down Mode: 0.1 µA
 - Power-save Mode: 0.75 µA (Including 32 kHz RTC)



8-bit AVR[®]
Microcontroller
with 4/8/16/32K
Bytes In-System
Programmable
Flash

ATmega48PA
ATmega88PA
ATmega168PA
ATmega328P

Rev. 8161D-AVR-10/09



Features

- High Performance, Low Power AVR[®] 8-Bit Microcontroller
- Advanced RISC Architecture
 - 125 Powerful Instructions – Most Single Clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
- Non-volatile Program and Data Memories
 - 8K/16K/32K Bytes of In-System Self-Programmable Flash
 - 512/512/1024 EEPROM
 - 512/512/1024 Internal SRAM
 - Write/Erase Cycles: 10,000 Flash/ 100,000 EEPROM
 - Data retention: 20 years at 85°C/ 100 years at 25°C⁽¹⁾
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by on-chip Boot Program hardware-activated after reset
 - True Read-While-Write Operation
 - Programming Lock for Software Security
- USB 2.0 Full-speed Device Module with Interrupt on Transfer Completion
 - Complies fully with Universal Serial Bus Specification REV 2.0
 - 48 MHz PLL for Full-speed Bus Operation : data transfer rates at 12 Mbit/s
 - Fully independent 176 bytes USB DPRAM for endpoint memory allocation
 - Endpoint 0 for Control Transfers: from 8 up to 64-bytes
 - 4 Programmable Endpoints:
 - IN or Out Directions
 - Bulk, Interrupt and Isochronous Transfers
 - Programmable maximum packet size from 8 to 64 bytes
 - Programmable single or double buffer
 - Suspend/Resume Interrupts
 - Microcontroller reset on USB Bus Reset without detach
 - USB Bus Disconnection on Microcontroller Request
- Peripheral Features
 - One 8-bit Timer/Counters with Separate Prescaler and Compare Mode (two 8-bit PWM channels)
 - One 16-bit Timer/Counter with Separate Prescaler, Compare and Capture Mode (three 8-bit PWM channels)
 - USART with SPI master only mode and hardware flow control (RTS/CTS)
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
 - Interrupt and Wake-up on Pin Change
- On Chip Debug Interface (debugWIRE)
- Special Microcontroller Features
 - Power-On Reset and Programmable Brown-out Detection
 - Internal Calibrated Oscillator
 - External and Internal Interrupt Sources
 - Five Sleep Modes: Idle, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
 - 22 Programmable I/O Lines
 - QFN32 (5x5mm) / TQFP32 packages
- Operating Voltages
 - 2.7 - 5.5V
- Operating temperature
 - Industrial (-40°C to +85°C)
- Maximum Frequency
 - 8 MHz at 2.7V - Industrial range
 - 16 MHz at 4.5V - Industrial range

Note: 1. See "Data Retention" on page 6 for details.



8-bit **AVR[®]**
Microcontroller
with
8/16/32K Bytes
of ISP Flash
and USB
Controller

ATmega8U2

ATmega16U2

ATmega32U2

7799D-AVR-11/10



Future Technology Devices International Ltd.

FT232R USB UART IC



The FT232R is a USB to serial UART interface with the following advanced features:

- Single chip USB to asynchronous serial data transfer interface.
- Entire USB protocol handled on the chip. No USB specific firmware programming required.
- Fully integrated 1024 bit EEPROM storing device descriptors and CBUS I/O configuration.
- Fully integrated USB termination resistors.
- Fully integrated clock generation with no external crystal required plus optional clock output selection enabling a glue-less interface to external MCU or FPGA.
- Data transfer rates from 300 baud to 3 Mbaud (RS422, RS485, RS232) at TTL levels.
- 128 byte receive buffer and 256 byte transmit buffer utilising buffer smoothing technology to allow for high data throughput.
- FTDI's royalty-free Virtual Com Port (VCP) and Direct (D2XX) drivers eliminate the requirement for USB driver development in most cases.
- Unique USB FTDIChip-ID™ feature.
- Configurable CBUS I/O pins.
- Transmit and receive LED drive signals.
- UART interface support for 7 or 8 data bits, 1 or 2 stop bits and odd / even / mark / space / no parity
- FIFO receive and transmit buffers for high data throughput.
- Synchronous and asynchronous bit bang interface options with RD# and WR# strobes.
- Device supplied pre-programmed with unique USB serial number.
- Supports bus powered, self powered and high-power bus powered USB configurations.
- Integrated +3.3V level converter for USB I/O.
- Integrated level converter on UART and CBUS for interfacing to between +1.8V and +5V logic.
- True 5V/3.3V/2.8V/1.8V CMOS drive output and TTL input.
- Configurable I/O pin output drive strength.
- Integrated power-on-reset circuit.
- Fully integrated AVCC supply filtering - no external filtering required.
- UART signal inversion option.
- +3.3V (using external oscillator) to +5.25V (internal oscillator) Single Supply Operation.
- Low operating and USB suspend current.
- Low USB bandwidth consumption.
- UHCI/OHCI/EHCI host controller compatible.
- USB 2.0 Full Speed compatible.
- -40°C to 85°C extended operating temperature range.
- Available in compact Pb-free 28 Pin SSOP and QFN-32 packages (both RoHS compliant).

Neither the whole nor any part of the information contained in, or the product described in this manual, may be adapted or reproduced in any material or electronic form without the prior written consent of the copyright holder. This product and its documentation are supplied on an as is basis and no warranty as to their suitability for any particular purpose is either made or implied. Future Technology Devices International Ltd will not accept any claim for damages howsoever arising as a result of use or failure of this product. Your statutory rights are not affected. This product or any variant of it is not intended for use in any medical appliance, device or system in which the failure of the product might reasonably be expected to result in personal injury. This document provides preliminary information that may be subject to change without notice. No freedom to use patents or other intellectual property rights is implied by the publication of this document. Future Technology Devices International Ltd, Unit 1, 2 Seaward Place, Centurion Business Park, Glasgow G41 1HH United Kingdom. Scotland Registered Company Number: SC136640

Class 1 Bluetooth® Module



Features

- Fully qualified Bluetooth 2.1/2.0/1.2/1.1 module
- Bluetooth v2.0+EDR support
- Postage stamp sized form factor, 13.4mm x 25.8 mm x 2mm
- Low power (30mA connected,, <10mA sniff mode)
- UART (SPP or HCI) and USB (HCI only) data connection interfaces.
- Sustained SPP data rates - 240Kbps (slave), 300Kbps (master)
- HCI data rates - 1.5Mbps sustained, 3.0Mbps burst in HCI mode
- Embedded Bluetooth stack profiles included (requires no host stack): GAP, SDP, RFCOMM and L2CAP protocols, with SPP and DUN profile support.
- Bluetooth SIG Qualified, End Product Listing
- Castellated SMT pads for easy and reliable PCB mounting
- Class 1 high power amplifier with on board ceramic RF chip antenna.
 - Certifications: FCC, ICS, CE
 - Environmentally friendly, RoHS compliant

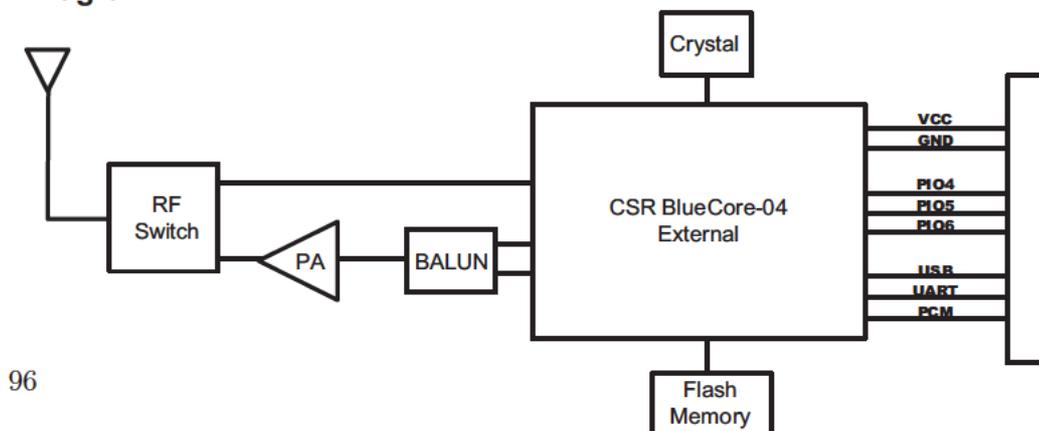
Applications

- Cable replacement
- Barcode scanners
- Measurement and monitoring systems
- Industrial sensors and controls
- Medical devices
- Asset tacking

Description

The RN41 is a small form factor, low power, highly economic Bluetooth radio for OEM's adding wireless capability to their products. The RN41 supports multiple interface protocols, is simple to design in and fully certified, making it a complete embedded Bluetooth solution. With its high performance on chip antenna and support for Bluetooth® Enhanced Data Rate (EDR), the RN41 delivers up to 3 Mbps data rate for distances to 100M.. The RN41 is the perfect product for engineers wanting to add wireless capability to their product but don't want to spend significant time and money developing Bluetooth specific hardware and software.

Block Diagram



96



Small, Low Power, 3-Axis $\pm 3 g$ Accelerometer

ADXL335

FEATURES

- 3-axis sensing
- Small, low profile package
 - 4 mm \times 4 mm \times 1.45 mm LFCSP
- Low power : 350 μ A (typical)
- Single-supply operation: 1.8 V to 3.6 V
- 10,000 g shock survival
- Excellent temperature stability
- BW adjustment with a single capacitor per axis
- RoHS/WEEE lead-free compliant

APPLICATIONS

- Cost sensitive, low power, motion- and tilt-sensing applications
- Mobile devices
- Gaming systems
- Disk drive protection
- Image stabilization
- Sports and health devices

GENERAL DESCRIPTION

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of $\pm 3 g$. It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration.

The user selects the bandwidth of the accelerometer using the C_x , C_y , and C_z capacitors at the X_{OUT} , Y_{OUT} , and Z_{OUT} pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for the X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.

The ADXL335 is available in a small, low profile, 4 mm \times 4 mm \times 1.45 mm, 16-lead, plastic lead frame chip scale package (LFCSP_LQ).

FUNCTIONAL BLOCK DIAGRAM

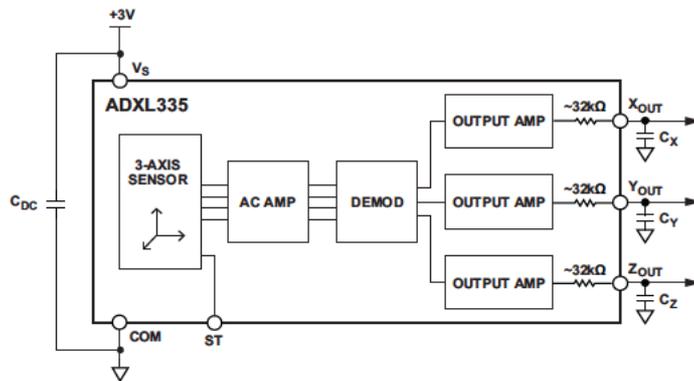


Figure 1.

Rev. B

Information furnished by Analog Devices is believed to be accurate and reliable. However, no responsibility is assumed by Analog Devices for its use, nor for any infringements of patents or other rights of third parties that may result from its use. Specifications subject to change without notice. No license is granted by implication or otherwise under any patent or patent rights of Analog Devices. Trademarks and registered trademarks are the property of their respective owners.

One Technology Way, P.O. Box 9106, Norwood, MA 02062-9106, U.S.A.
Tel: 781.329.4700 www.analog.com
Fax: 781.461.3113 ©2009–2010 Analog Devices, Inc. All rights reserved.



InvenSense Inc.
1197 Borregas Ave, Sunnyvale, CA 94089 U.S.A.
Tel: +1 (408) 988 7339 Fax: +1 (408) 988 8104
Website: www.invensense.com

Document Number: PS MPU 6000A 00
Revision: 3.2
Release Date: 11/16/2011

MPU-6000 and MPU-6050 Product Specification Revision 3.2

ADVANCE INFORMATION

Quelle: <https://github.com/jrowberg/i2cdevlib>

```
#include "Wire.h"
#include "I2Cdev.h"
#include "MPU6050.h"

MPU6050 accelgyro;
int16_t ax, ay, az;
int16_t gx, gy, gz;
#define LED_PIN 13
bool blinkState =false;

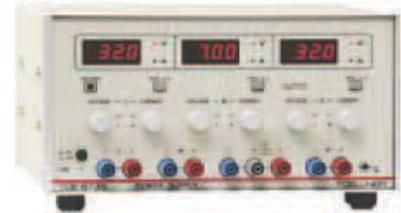
void setup() {
    Wire.begin();
    Serial.begin(38400);
    Serial.println("Initializing I2C devices...");
    accelgyro.initialize();
    Serial.println("Testing device connections...");
    Serial.println(accelgyro.testConnection() ?
        "MPU6050 connection successful" : "MPU6050 connection failed");
    pinMode(LED_PIN, OUTPUT);
}

void loop() {
    accelgyro.getMotion6(&ax, &ay, &az, &gx, &gy, &gz);
    Serial.print("a/g:\t");
    Serial.print(ax); Serial.print("\t");
    Serial.print(ay); Serial.print("\t");
    Serial.print(az); Serial.print("\t");
    Serial.print(gx); Serial.print("\t");
    Serial.print(gy); Serial.print("\t");
    Serial.println(gz);

    blinkState = !blinkState;
    digitalWrite(LED_PIN, blinkState);
}
```

Multiple-output power supplies

TOE 8732
TOE 8733
TOE 8735



TOE 8735

The dual-output power supplies

of the 8732 series offer two absolutely identical supply units in a single housing. The outputs are electrically isolated from each other and floating. They can be easily connected either in parallel or in series, and are additionally provided with a tracking facility that allows output II to be synchronized by output I. The two power supplies then provide a positive voltage and a negative voltage compared to the common reference point. Current limits for the two outputs can be set independently of each other.

In addition, the power supplies are equipped with three digital displays, and allow external voltage measurement in the ranges 0 to 20 V and 0 to 200 V. The voltages at outputs I and II are measured with autoranging. Up to 19.99 V, the resolution is 10 mV; above 20.0 V, the resolution is 100 mV. The outputs I and II can be optionally adjusted by means of an external control voltage (0 to 10 V) via an analog control input.

The triple-output power supplies

of the 8733 series have three independent floating outputs. The 16 V, 32 V and 48 V outputs are mainly intended to supply linear circuits. The output voltage and output current are adjustable using ten-turn potentiometers. A tracking facility is provided. The 0 to 7 V output has a current rating of 3 A or 5 A as standard and is chiefly intended for supplying digital circuits.

The power supplies also have three digital displays as standard. The voltages at outputs I and II are automatically displayed with a resolution of 10 mV up to 19.99 V and a resolution of 100 mV above 20.0 V. The current and voltage of outputs I, II and III can be optionally set by an external control voltage (0 to 10 V) via an analog control input.

The quintuple-output power supplies

The TOE 8735 power supplies have five independent floating outputs. The 16 V and 32 V outputs are mainly used to supply linear circuits, the 7 V output chiefly serves to supply digital systems. The symmetrical fixed voltage output* can be used to supply bipolar or CMOS components. The 16 V and 32 V outputs are provided with a tracking facility. The output values are displayed on three selectable 3½-digit LCDs; the max. resolution is 10 mV or 1 mA. The current and voltage of outputs I, II and III can be optionally set by an external control voltage (0 to 10 V) via an analog control input. (Option: TOE 8730/253)

* The fixed voltage output ± 15 V can be switched internally to ± 12 V.

Special features

- Extremely low residual ripple < 50 μ V
- Electrically decoupled outputs
- Precise digital displays

DIGITAL OSCILLOSCOPES & DIGITAL SCOPES

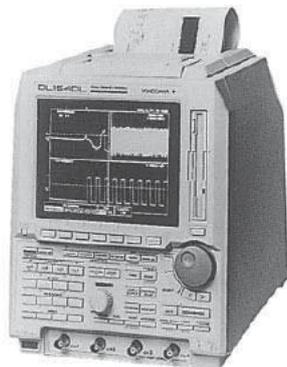


DL1540/DL1540L/DL1520/DL1520L

701510/701520/701505/701515 Digital Oscilloscopes DL1540/DL1540L/DL1520/DL1520L



DL1540(701510)
215 × 268 × 295mm 5kg
(8-1/2 × 10-9/16 × 11-5/8" 11.0 lbs)



DL1540L(701520)
215 × 268 × 295mm 6kg
(8-1/2 × 10-9/16 × 11-5/8" 13.2 lbs)



DL1520(701505)
215 × 268 × 295mm 6kg
(8-1/2 × 10-9/16 × 11-5/8" 13.2 lbs)



DL1520L(701515)
215 × 268 × 295mm 6kg
(8-1/2 × 10-9/16 × 11-5/8" 13.2 lbs)

NEW

NEW



★ Safety Standards; EN61010
EMI Standard; EN55011 Group 1 Class A
Immunity Standard; EN50082-2: 1995

The DL1500 Series Digital Oscilloscopes are high performance personal digital oscilloscope in a compact and light body.

FEATURES

- Full range 4 channels (DL1540/DL1540L)
- Max. 200 MS/s(8 bits)
- 150 MHz analog bandwidth

- Max. 2M words (DL1540L)
1M words (DL1520L)
120k words (DL1540)
20k words (DL1520)
- Approx. 6 kg (DL1540L/DL1520L/ DL1520)
Approx. 5 kg (DL1540)
- Footprint smaller than A4
- Built-in 3.5-inch FDD
- Built-in printer (option)

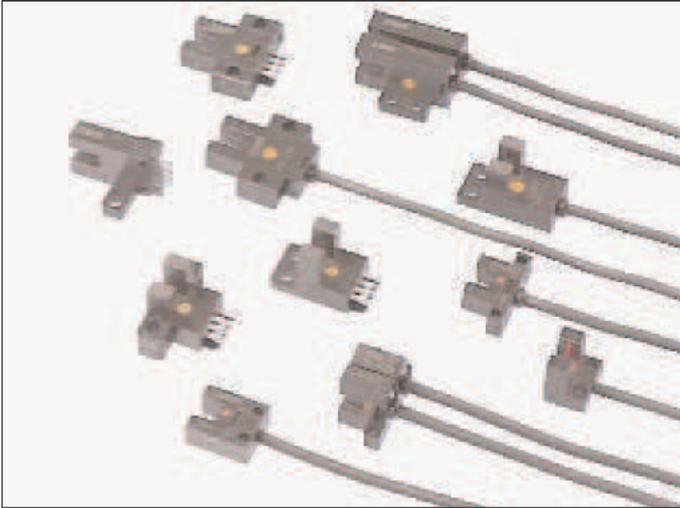
DL1500 SERIES

| | | DL1540L | DL1540 | DL1520L | DL1520 |
|----------------------|------------------------------|-----------|-----------|-----------|---------------|
| Basic Specifications | Number of input channels | 4 | 4 | 2 | 2 |
| | Max. Sampling rate | 200 MS/s | 200 MS/s | 200 MS/s | 200 MS/s |
| | Max. record length | 2M word | 120k word | 1M word | 20k word*1 |
| | Frequency bandwidth | 150MHz | 150MHz | 150MHz | 150MHz |
| Interface | Advanced trigger | Option | Option | NA | NA |
| | 3.5 inch FDD | Std. | Std. | Std. | Std. |
| | Option box control connector | Std. | Std. | NA | NA |
| | Centronics | NA | NA | Std.*2 | Std.*2 |
| | RS-232-C | Option | Option | Std.*2 | Std.*2 |
| | GP-IB | Std. | Std. | Std.*2 | Std.*2 |
| Functions | Built-in printer | Option | Option | Option | Option |
| | Snap shot | Available | Available | Available | 101 Available |
| | History memory | Available | Available | Available | NA |
| | Sequential store | Available | Available | Available | NA |

*1: In Roll mode. *2: Varies depending on the model.

PM SERIES

U-shaped Micro Photoelectric Sensor

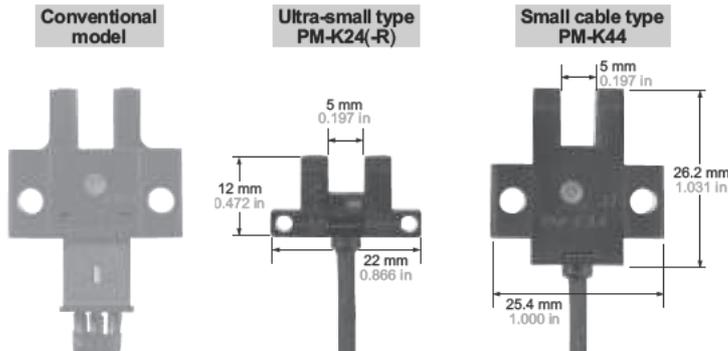


Extremely small size enables space saving and quick installation!



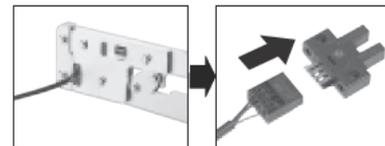
Extremely small

Ultra-small type PM-□24(-R) contributes to the miniaturization of your equipment. Even the small cable type has become very compact.



Quick fitting hook-up connector

Easy to maintain connector type models are available. Its exclusive connector is the industry's first hook up connector. Since only crimping with exclusive pliers is to be done, cumbersome soldering or insulation is absolutely not required. Further, connector attached cable is also available.



Crimp the connector on the cable.

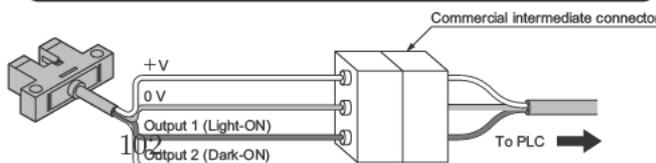
Quick connection to the sensor.

Equipped with two independent outputs

All models are equipped with two independent outputs –Light ON and Dark ON. Hence, one model suffices even if the output is to be used differently, depending upon the location of use.

Also, since two independent outputs have been provided, cumbersome handling of the output conversion control input, or fear of logic inversion due to a cable break, is eliminated. The sensor can be connected to the existing wiring as it is.

Example of connection with a commercial intermediate connector



Just connect the cable of the used output (either Light-ON or Dark-ON).

Connected device side can be left as it is.

Note: Ensure to insulate the unused output wire.

Wide model variety

A wide variety of 17 shapes and 34 models is available. You may select from this wide range to suit the mounting conditions.

Meets global requirements

Conforms to Europe's EMC Directive and obtains UL Recognition. Both, NPN and PNP output models are available.

CY

PX-2

Amplifier Built-in

RT-610

MS-AJ

Sensor Mounting Stand

PM

Micro

PM2

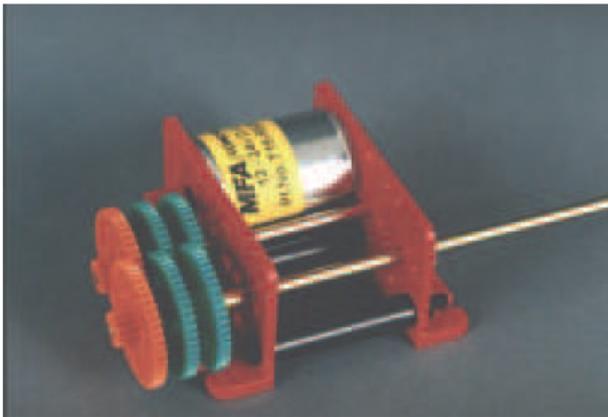
NX5

VF

Multi-voltage

EQ-500

LOW VOLTAGE D.C. MOTORS & GEARBOX UNITS



Printed on 100% recycled paper.

