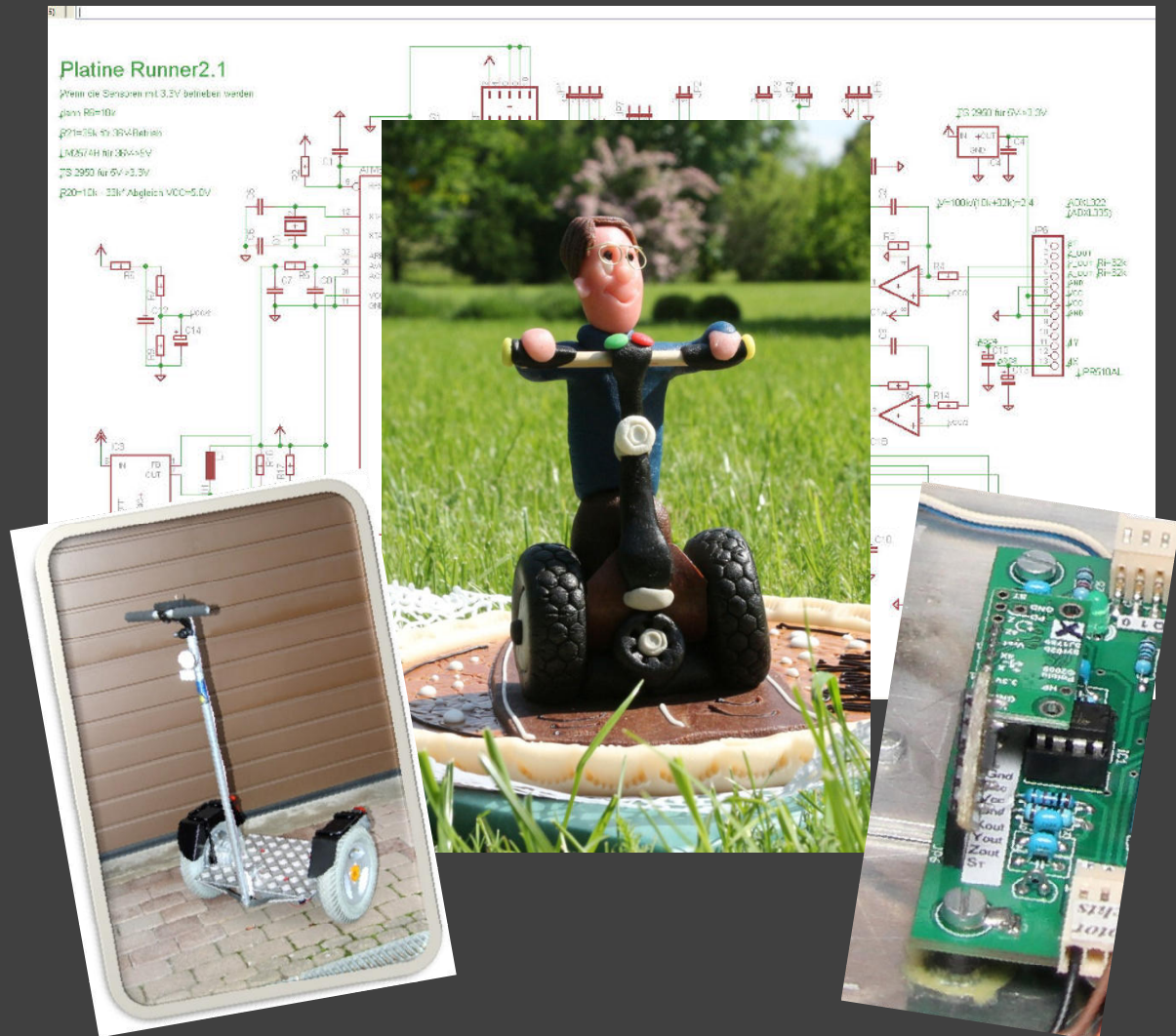


Wolfgang Schmidt

Runner Wheelie Segway & Co

Selbstbalancierende Fahrzeuge im Selbstbau



Dokumentation für einen DIY Segway Clone
Theorie, Software, Schaltungen, Platinen
Ausgabe 011116

Selbstbalancierende Fahrzeuge im Selbstbau



1.	Historisches.....	11
1.1.	Segway.....	11
1.2.	Selbstbauer	13
2.	Grundlagen	15
2.1.	Anschauliche gemacht.....	15
2.2.	Das inverse Pendel.....	15
2.3.	Theoretischer Ansatz	15
3.	Die Mechanik	18
3.1.	Mechanisches Konzept.....	18
3.2.	Details eines mechanischen Aufbaus	18
3.3.	Motoren und ihre Leistung	19
3.4.	Räder und Antriebe.....	22
3.4.1.	Übersetzungen	22
3.4.2.	Variante 1	22
3.4.3.	Variante 2	23
3.4.4.	Variante 3	24
3.4.5.	Variante 4	25
3.4.6.	Variante 5	25
4.	Strom muss fließen	26
4.1.	Batterien / Akkus.....	26
4.2.	Laden der Akkus.....	28
4.3.	Der Einschaltvorgang.....	29
5.	Elektronisches Konzept.....	30
5.1.	Sensoren	30
5.1.1.	Der Beschleunigungssensor	30
5.1.2.	Das Gyroskop.....	30
5.1.3.	Die Gierrotation.....	31
5.1.4.	Mikrocontroller als zentrale Steuerung.....	31
5.2.	Der Mikrocontroller	33
5.2.1.	Auslastung des Mikrocontrollers	34
5.3.	Schaltung der Hauptplatine.....	37
5.3.1.	Funktionen des Port A	37

5.3.2.	Funktionen des Port B	37
5.3.3.	Funktionen des Port C	38
5.3.4.	Funktionen des Port D	38
5.3.5.	Schaltplan der Hauptplatine 2.6	39
5.4.	! – Fast ein Fehler – in Schaltung und Layout	41
5.5.	Ansteuerung der Motoren	41
5.6.	Die H-Brücke	42
6.	Was ist eine PWM-Steuerung?.....	43
6.1.	PWM-Modi.....	43
6.1.1.	PWM-Modus 1	44
6.1.2.	PWM-Test.....	46
6.2.	Schaltplan der H-Brücke	49
6.3.	Platine der H-Brücke.....	50
6.3.1.	H-Brücke Version 2.8.....	50
6.3.2.	Kühlung der H-Brücke.....	53
7.	Gesamtplan der Elektronik	56
7.1.	Gesamtplan Runner2+.....	57
7.2.	Hochstromverkabelung	58
7.3.	Anschlussklemmen und Kabel	59
7.4.	Signalstromverkabelung	60
7.5.	Wichtig: Der gemeinsame Massepunkt.....	60
8.	Software	61
8.1.	Programmkonzept	61
8.2.	Flussdiagramme	62
8.2.1.	Startablauf des Runner Start.....	63
8.2.2.	Mainloop des Runner.....	64
8.2.3.	Balance-Interrupt	65
8.3.	Regelroutine	65
8.3.1.	Der Balancealgorithmus.....	65
8.3.2.	Dynamische Parameteranpassung-Balanceregung	68
8.3.3.	Die Lenkung	68
8.3.4.	Der Balance-Lenk-Algorithmus	69

8.3.5.	Mein Dank gilt Trevor Blackwell	70
8.4.	Messgrößen und Parameter	71
8.4.1.	Der digitale Sensor MPU-6050	71
8.4.2.	Auswertung des Sensors MPU-6050	72
8.4.3.	Auswertung des Gyroskops im MPU-6050	73
8.4.4.	Parameter	73
8.4.5.	Anordnung des Sensors MPU-6050	74
8.5.	Digitale Filter	75
8.5.1.	Gegen das Rauschen	75
8.5.2.	Ein einfacher digitaler Filter – Der Simple_Filter	75
8.5.3.	Doppelt gefiltert – ein Komplementärfilter	79
8.5.4.	Kalman-Filter „mal ganz einfach“	81
8.6.	Geschwindigkeitsabhängige Lenkung	82
8.6.1.	Zusammenhänge	82
8.6.2.	Realisierung der Kurvenfahrt	83
8.6.3.	Vereinfachung	84
8.6.4.	Praktische Umsetzung 1. Art	85
8.6.5.	Praktische Umsetzung 2. Art	85
9.	Sicherheit	87
9.1.	Technische Sicherheit	87
9.1.1.	Aufbau	87
9.1.2.	Der Notausschalter	88
9.1.3.	Elektronik	88
9.2.	Sichere Software	89
9.3.	Verkehrssicherheit	89
10.	Der Peukert-Effekt	91
10.1.	Was ist der Peukert – Effekt?	91
10.2.	Die Peukert – Funktion	91
10.3.	Berechnung der entnommenen Kapazität	93
10.3.1.	Beispiel 1	93
10.3.2.	Beispiel 2	93
10.3.3.	Variable Kapazitätsentnahme	93

10.3.4.	Der Algorithmus	94
10.3.5.	Praktische Umsetzung	95
10.4.	Berechnung der Peukertzahl aus Batteriedaten	97
10.5.	Realisierung der Strommessungen	98
10.6.	Kontrolle der Strommessung	99
11.	Bauanleitung	100
11.1.	Motoren und Räder	100
11.2.	Plattform-Gehäuse	102
11.3.	Lenkung	104
11.4.	Elektronik	106
11.5.	Modifikation der BTS-H-Brücken	108
11.5.1.	Leistungsoptimierung	108
11.5.2.	Temperaturoptimierung	109
11.5.3.	Einschaltverhalten und Fehlerbeseitigung	110
11.5.4.	Vergleich mit der eigenen H-Brücke	111
11.5.5.	Fazit:	111
11.6.	Anschluss der Tastatur	112
11.7.	Anschluss der Lichtschranke	113
11.8.	Software	114
11.8.1.	Download	114
11.8.2.	Kompilieren	114
11.8.3.	Atmega programmieren	114
11.8.4.	Updates	114
11.8.5.	Hinweise zur Inbetriebnahme der Hauptplatine	114
11.9.	Hinweise zur Montage	117
11.10.	Funktionsprüfung	118
11.10.1.	Normalmodus	118
11.10.2.	Fußschalter	118
11.10.3.	Tastatur - Parametermodus	118
11.10.4.	Offsetwerte	118
11.10.5.	Testmodus	119
11.11.	Empfehlung	120

11.11.1.	Akkus	120
11.11.2.	Mainboard	120
11.11.3.	H-Brücken	121
11.11.4.	Chassis	121
11.11.5.	Räder und Übersetzung	121
11.11.6.	Lenkerbefestigung.....	122
11.11.7.	Sicherheit.....	123
12.	Betriebsanleitung.....	124
12.1.	Das Fahren	124
12.2.	Die Einschaltprozedur im Detail.....	125
12.2.1.	Sicherheitsprüfungen	125
12.2.2.	Fehlernummern	126
12.2.3.	Automatische Initialisierung	126
12.2.4.	Manuelle Initialisierung	126
12.2.5.	Erste Initialisierung	126
12.3.	Der Parametermodus	127
12.3.1.	Bedeutung der Parameter.....	127
12.3.2.	Übersicht der veränderbaren Parameter ab V7.35:.....	132
12.3.3.	Ändern der Parameter	134
12.4.	Testmodus	135
12.4.1.	Aktivierung des Testmodus.....	135
12.4.2.	Automatischer Testablauf	135
12.4.3.	Beenden des Testmodus	137
13.	Rechtliches in Deutschland	138
13.1.	Straßenfahrten und Zulassung	138
13.1.1.	Die Mobilitätshilfeverordnung.....	138
13.1.2.	Allgemeine Betriebserlaubnis.....	141
13.1.3.	Haftpflichtversicherung	143
14.	Updates.....	144
14.1.	Hauptplatine	144
14.1.1.	Runner_AT32_36V_2.5 digital	144
14.1.2.	Runner_AT32_36V_2.6 digital	146

14.1.3.	Runner_M_2.7	(ab Ende 2016).....	148
14.2.	H-Brücken		151
14.2.1.	Runner_Mot_H_2.8		151
14.2.2.	BTN / BTS7960B / 7970 - Module.....		151
14.3.	Software.....		153
14.3.1.	Update V7.30.....		153
14.3.2.	Update V7.33.....		154
14.3.3.	Update V7.35.....		154
14.3.4.	Update V7.36.....		155
14.3.5.	Update V7.37.....		155
Index.....			156

Hinweis:

In jeder neuen Ausgabe dieser Dokumentation erscheinen **neue und geänderte** Beschreibungen. Änderungen im Dokument werden zukünftig in **rot** geschrieben. Bei der nachfolgenden Ausgabe erscheinen diese wieder in schwarzer Schrift und aktuelle Neuigkeiten in **rot**, u.s.w..

Die Versionen der Hauptplatine werden mit dem Buchstaben **M** benannt und erhalten z.B. die Bezeichnung M_2.6.

Die Versionen der H-Brücken werden mit dem Buchstaben **H** benannt und erhalten z.B. die Bezeichnung H_2.8.

Die Versionen der Software werden weiterhin mit dem Buchstaben **V** benannt und erhalten wie bisher z.B. die Bezeichnung V_7.35

Der Leser kann unter der Adresse:

www.ups.bplaced.de/Dokumentation

die neusten und auch früheren Veröffentlichungen finden.

Vorwort

Seit einigen Jahren ist das unter dem Namen Segway bekannte selbstbalancierende Fahrzeug immer häufiger auf öffentlichen Straßen zu sehen.

Das als „Human Transporter“ oder auch als „*self-balancing personal transportation device with two wheels*“ bezeichnete Fahrzeug lässt Beobachter erstaunt aufsehen. Die Frage, warum fällt das Fahrzeug nicht um, schließt sich an die Frage nach der Funktion des menschlichen Gleichgewichtssinns an.

Die Entwicklung der Elektronik in den letzten Jahrzehnten hat auch hier zu einer Lösung geführt.

Theoretische Überlegungen sind in der universitären Physik schon längere Zeit üblich. Sie fallen unter den Begriff des „inversen Pendels“. Das inverse Pendel ist dem Leser wahrscheinlich seit seiner Kindheit bekannt. So hat fast ein jeder das Balancieren eines Besenstiels auf einer Hand mit mehr oder weniger Erfolg ausprobiert.

Die Entwicklung einer geeigneten Elektronik wird heute durch die Verfügbarkeit von Mikrokontrollern unterstützt. Diese kleinen elektronischen Bausteine findet man in fast allen elektronischen Geräten des Haushalts bis hin zu den modernen PKWs, in denen z.B. schon jeder elektrische Fensterheber oder Scheibenwischermotor seinen eigenen Mikrokontroller besitzt.

Mikrokontroller sind programmierbar und somit jeder äußeren Situation durch eine flexible und bei Bedarf änderbare Software anpassbar.

Nicht nur der Preis eines industriell gefertigten selbstbalancierenden Fahrzeuges sondern auch das Interesse an Mathematik, Physik, Elektronik und Informatik kann einem zum Selbstbau eines im weiteren Verlauf als **Runner** bezeichneten Fahrzeuges motivieren.

Geboten wird eine Vielzahl von theoretischen als auch sehr praxisorientierten Informationen, die ein Verständnis der Softwarefunktionen sowie den Selbstbau eines solchen Fahrzeuges ermöglichen.

Diese Beschreibung erhebt keinen Anspruch auf Vollständigkeit. Ich bitte den Leser den Inhalt als Anregungen einzuordnen. Dabei bin ich für jede Rückmeldung dankbar und werde diese ggf. hier berücksichtigen.

Weitere Informationen findet man auf der Seite: www.ups.bplaced.de

W. Schmidt
18.04.2014

Vorwort 2

Wie der geneigte Leser feststellen wird, erscheinen in dieser Dokumentation, die nach Updates der Software oder Hardware jeweils ergänzt wird, immer mehr Informationen, an Hand derer der gesamte Entwicklungsprozess nachvollzogen werden kann.

Diese Dokumentation wurde wahrscheinlich über meine Internetseite heruntergeladen. Auf dieser Seite wird zukünftig auch die jeweils neuste Dokumentation angekündigt.

Der Umfang der Dokumentation mit mehr als 150 Seiten beinhaltet zwischenzeitlich viele Informationen, welche durch Weiterentwicklungen überholt sind.

Analoge Sensoren werden von mir nicht mehr verwendet, H-Brücken wurden weiterentwickelt.

Um diesen Umständen gerecht zu werden, verschwinden ältere Abschnitte und werden durch den aktuellen Stand der Entwicklung ersetzt. Dies ist z.B. die Hauptplatine 2.5, die H-Brücke 2.8 und die Software V7.30.

Das Format dieser Dokumentation ergibt sich aus der Tatsache, dass technische Publikationen vorzugsweise diese Abmessung besitzen.

Der **beschnittene** Ausdruck passt dann in mein entsprechendes Bücherregal.

Beim Ausdruck auf DIN A4 Blättern bleiben unten einige cm frei. Diese bieten sich für Notizen oder auch eigene Fragestellungen an.

Der Fehlerteufel steckt auch in diesem Buch.

Nicht in dem Sinne, dass fehlerhafte Angaben leichtsinnig niedergeschrieben wurden, sondern dass durch Tippfehler oder Satzumstellungen, Einfügungen oder Streichungen sich im Text ein Fehler eingeschlichen hat.

Dies fällt dem Leser insbesondere beim ersten Lesen auf.

Auch sachliche Fehler kann ich nicht ausschließen.

Ich würde mich über eine Rückmeldung freuen!

Einfach die fehlerhafte Passage kopieren oder sie Seite/Zeile angeben und an mich schicken. Diese Zeilen dürfen bei mir „umgetauscht“ werden.

1. Historisches

1.1. Segway

Dieses Kapitel besteht aus Auszügen einer Veröffentlichung in den Solarmobil Mitteilungen Nr.62 vom September 2006.

SEGWAY Meilensteine

27. Juli 1999; Der renommierte Erfinder Dean Kamen gründet ein neues Unternehmen mit der Vision effiziente emissionsfreie Transportmittel auf der technologischen Basis der "Dynamischen Stabilisierung" zu entwickeln. Man konzentrierte sich dabei auf die Entwicklung von Geräten, die minimalen Platz benötigen, leicht manövrierbar sind und somit auf Gehsteigen und Wegen zum Einsatz kommen. Die ersten Mitarbeiter von SEGWAY LCC kommen von Dean Kamens bestehender Firma, DEKA Research and Development Corp.

1. Dezember 2001; Die Firma bekommt den Namen SEGWAY LCC (früher Acros LLC)

3. Dezember 2001; SEGWAY LCC präsentiert der Öffentlichkeit den ersten Segway® Human Transporter.

21. März 2002; Das erste produzierte Serien Model I-167 mit der Nummer: 022111 000 001 wird von Dean Kamen, dem Gründer und Vorsitzenden von SEGWAY LCC, feierlich präsentiert.

29. März 2002; SEGWAY LLC versteigert eine limitierte Edition von 3 SEGWAYS bei amazon.com. Über 500 Kaufangebote aus dem ganzen Land treffen ein, wobei ein Erlös von US\$ 364.800 erzielt werden konnte, der der Wohltätigkeitsorganisation FIRST gespendet wird. SEGWAY HT Nummer 1 wird um US\$ 160.100,- versteigert, SEGWAY Nr. 2 um \$104.100,- und SEGWAY Nr. 3 bringt US\$ 100.600,- ein.

28. August 2002; SEGWAY Fans gründen das Forum www.segwaychat.com, um eine Plattform zu schaffen, auf welcher SEGWAY Enthusiasten und zukünftige SEGWAY Besitzer Informationen, Geschichten und Bilder tauschen können. Innerhalb eines Jahres treten mehr als 1400 User dem Forum bei.

18. November 2002; SEGWAYS werden erstmals über amazon.com angeboten. Das Unternehmen veranstaltet einen Wettbewerb bei dem 30 Personen einen SEGWAY HT, eine Tour durch das SEGWAY LCC Hauptquartier und ein Dinner mit Dean Kamen, gewannen.

28. Februar 2003; Segway HTs werden an amazon.com Kunden versandt.

29. August 2003; 200 Segway Besitzer und Liebhaber versammeln sich in Chicago, Illinois zum ersten Segway Fest, das von Segway Besitzern organisiert wurde, um das SEGWAY Erlebnis zu feiern, Ideen zu fördern und die Technology besser kennen zu lernen.

15. Oktober 2003; Die Segway p-Serie wird vorgestellt. Mit optimierter Portabilität, noch kompakter und beweglicher auf engem Raum.

April 2004: Vorstellung des Segway auf der „multi-modal-mobil“ Konferenz im Bundeshaus in Berlin. Segway an einer Park&Charge Tankstelle bei einer Konferenz im April 2004 im Bundeshaus in Berlin.

16. November, 2004; Das erste Fahrerteam von „Amerika mit 10 m.p.h.“ beendet seine transkontinentale Reise quer durch Amerika auf dem Segway HT.

Links:

<http://www.solarmobil.net/download/sm62-segway.pdf>

<http://www.thomasknauf.de/ginger/ginger.htm>

http://www.segway-pt.eu/index.php/milestones_segway_de_de.html

1.2. Selbstbauer

Bei den Selbstbauern gibt es eine Veröffentlichung von Trevor Blackwell auf <http://www.trevorblackwell.com/#scooter> die einen grundlegenden Regelalgorithmus vorstellt.

Roberto schrieb im Elektor-Forum zur Historie:
„Am Anfang war das Projekt AT3329 (AVR Design Contest 2006)

Nach einiger Zeit kam das Zzaag-Projekt zum Vorschein!
Aufbauend auf AT3329 entstand ein Fahrzeug, welches vom Erbauer über den Elektor-Verlag veröffentlicht und in modifizierter Form vertrieben wurde

Im Jahr 2009 stellt der ELEKTOR-Verlag in seiner Zeitschrift dieses Elektor Wheelie vor. Das Fahrzeug kostete als Bausatz rund 1600.-€.

Im Forum von <http://www.elektor.de> beschreibt Günter Gerold unter Elektor Wheelie/ Mein Wheelie seinen eigenen Nachbau des Elektor Wheelie, verbunden mit wesentlichen Verbesserungen.

Eine Projektbeschreibung meines „**Runner**“ steht auf der Seite www.ups.bplaced.de und als Dokument im Downloadbereich zur Verfügung.

Fast alle DIY-Segway-Projekte verwenden mehr oder weniger den Code, der im Projekt AT3329 erstmalig veröffentlicht wurde. Auch ich habe mich daran orientiert und nach dem Verständnis einen eigenen Code entwickelt.

Diese Ausführungen werden von Thomas Scherer im Elektor-Forum ergänzt:
„So ist grundlegende Erfindung der Selbstbalancierung eines derartigen Gefährts wohl schon 1987 durch den japanischen Robotik-Forscher Yamafuji erfolgt. Sie ist in Japan 1996 patentiert worden. Darauf basieren dann die Teile, die Sony gebaut hat. Insofern war wohl Yamafuji der Erste, der auf die Idee kam. Und eine andere sehr bekannte Firma nutzt dieses Prinzip genau so gut wie die ganzen Projekte im Internet oder Chris Krohne oder eben das Elektor Wheelie. Wenn hier also einem Ehre gebührt, dass Yamafuji für den Geistesblitz (er ist immer noch Uni-Professor) und die bekannte Firma für die Demonstration, dass sich dies wirklich in der Praxis umsetzen lässt und bewährt, denn ohne deren Anregung gäbe es die ganzen Projekte nicht, weil wir wohl alle nichts von diesem japanischen Patent wüssten.“

Entscheidend war wohl das im o.g. *AVR Design Contest 2006* vorgestellte Projekt *AT3329*. Es zeigt den ersten Selbstbau mit Hilfe zweier Sensoren und eines Mikrocontrollers.

An diesem Konzept hat sich zwischenzeitlich nicht viel geändert.

Eine Dokumentation kann auch unter <http://en.pudn.com> und dem Suchwort *AT3329* (etwas umständlich) herunter geladen werden.

Weiterentwicklungen, wie das *zzaag*-Projekt, das *Elektor Wheelie* und die Verbesserungen von Günter Gerold findet man im Netz.

Eine Vielzahl von Videos ist auf der Plattform YouTube gespeichert. Suchen mit Kombinationen aus den Begriffen *Segway*, *Clone*, *homemade*, *Runner*, *Rutscherle* und *Wheelie* fördern über 100 kleiner Videos zutage, die den weltweiten Stand der Selbstbauerszene dokumentieren.

2. Grundlagen

2.1. Anschauliche gemacht

Das im Vorwort erwähnte Spiel mit dem Besenstiel liefert den Schlüssel zum Verständnis des Balancierens.

Jede geringe Neigung wird intuitiv durch eine nachfolgende Bewegung der Hand ausgeglichen, um das Herabstürzen des Besenstiels zu verhindern. Hat der im Idealfall senkrecht stehende Stiel jedoch eine gewisse Neigung überschritten, so folgt der unweigerliche Absturz.

In dieser Situation ist erkennbar, dass eine größere Neigung des Besenstiels ein umso kräftigeres und schnelleres Nachführen der Hand erforderlich macht. Dieser Bewegung sind natürliche Grenzen gesetzt, so dass beim Erreichen oder Überschreiten dieser Grenze, die sich in der Kraft zum Nachführen der Hand widerspiegelt, der Besenstiel überkippt.

Der Leser möge diesen Balanceakt nochmals ausprobieren.

2.2. Das inverse Pendel

Unter diesem Begriff erscheint die Problematik des Balancierens in der Physik. Der theoretische Lösungsansatz führt zu gekoppelten, nichtlinearen Differentialgleichungen, deren Nichtlinearität auf die Funktionen $\sin()$ und $\cos()$ zurückzuführen ist, die zur Beschreibung der Pendelbewegung erforderlich sind. Eine geschlossene Lösung ist kaum zu finden. Falls doch, so ist sie extrem kompliziert.

2.3. Theoretischer Ansatz

Für kleine Pendelbewegungen, die im Fall eines selbstbalancierenden Fahrzeuges vorliegen sollten, ist eine Näherung durch linearisierte Gleichungen möglich.

Eine Lösung soll hier nicht vorgestellt werden. Die variablen Größen, die sich aus den resultierenden Differentialgleichungen ergeben, führen zum Verständnis des später erläuterten Regelalgorithmus.

Das bereits beschriebene Balanceexperiment führt zu der Erkenntnis, dass sowohl der Neigungswinkel θ als auch die Geschwindigkeit des Kippvorgangs benannt als

Änderungsgeschwindigkeit des Neigungswinkels oder Neigungsgeschwindigkeit $\dot{\theta}$ von Bedeutung sind.

Die aktuelle Bewegungsgeschwindigkeit $\mathbf{v}$ der stützenden Hand muss neigungsbedingt verändert werden und führt somit zur physikalische Größe der Beschleunigung $\mathbf{a}$, die durch die Wirkung einer Kraft auf die Hand realisiert wird.

Diese Überlegungen zeigen, dass die Messung und Auswertung des Neigungswinkels θ und die Neigungsgeschwindigkeit $\dot{\theta}$ für den Regelalgorithmus herangezogen werden müssen.

Aus diesen genannten Größen werden die Geschwindigkeit und ihre Änderung, die Beschleunigung (der Hand) abgeleitet.

In welchem Maße der Neigungswinkel θ und die Neigungsgeschwindigkeit $\dot{\theta}$ zur Steuerung herangezogen werden, lässt sich unter Berücksichtigung der zu bewegend Masse, der Lage des Schwerpunktes und der mechanischen Beschaffenheit der Elektronik und des Antriebes sicher berechnen.

In der hier vereinfachten Form werden diese Parameter empirisch bestimmt.

Eine weitere Veranschaulichung ist folgende:

Ich möchte dabei in einfachen Worten auch einem Laien die Notwendigkeit und Funktion der Sensoren zu erklären.

Man kann auf einem solchen Fahrzeug stehen, ohne dass ein Antrieb vorhanden ist. Aber dieser theoretische "Gleichgewichtszustand" ist auf Dauer nicht haltbar. Man spricht von einem labilen Gleichgewichtszustand. Anschaulich liegt eine Kugel auf der Spitze einer Stecknadel.

Misst man den Neigungswinkel des Fahrzeuges, so kann man proportional zu Neigung die Motoren ansteuern.

Anschaulich hat sich die Spitze der Stecknadel zu einem nach oben spitz zulaufenden Dreieck geweitet. Die Geschwindigkeit der einmal herabrollenden Kugel wird dabei ständig zunehmen.

Nun sagt die Physik, dass aus der vektoriellen Komposition der Kraftvektoren folgt, dass die Kraft des horizontalen Antriebes, durch den eine Balance erreicht werden soll, proportional zum Tangens des Winkels ist und somit im ungünstigsten Fall gegen unendlich geht. Dabei wurde die vorliegende Geschwindigkeit noch nicht berücksichtigt.

Wenn man dies geeignet darstellt und dabei die Werte der Tangensfunktion nach unten, in Richtung der wirkenden Erdbeschleunigung in einem Koordinatensystem

aufträgt, dann entsteht das Bild einer näherungsweise nach unten geöffneten Parabel, auf deren Oberseite sich die bereits genannte Kugel stabil halten soll.

Der obere Punkt ermöglicht ebenfalls ein labiles Gleichgewicht, womit diese alleinige Lösung unter Berücksichtigung des Neigungswinkels zur Berechnung der notwendigen Antriebskräfte nicht ausreicht.

Faktoren können dies "Parabel" flacher werden lassen, das Gleichgewicht wird jedoch nie stabil.

Man stelle sich nun eine Neigung von 5° vor. Gewinnt man daraus eine Antriebsleistung von z.B. 50 Watt, so ist bei einem Neigungswinkel von 10° wegen der immer steiler werdenden Parabel eine Leistung von z.B. 150 Watt erforderlich. Bedenkt man weiter, dass eine Neigung von 5° auch mit einer Kippgeschwindigkeit von $1^\circ/\text{s}$ bis $10^\circ/\text{s}$ verbunden sein kann, so ist auch dieser Kipp- bzw. Fallprozess aufzufangen.

Der letzte Gedanke lässt erkennen, dass insbesondere diese Kippgeschwindigkeit mit in die Regelung einbezogen werden muss.

Die mathematische Beschreibung führt hier zur Ableitung des Neigungswinkels, was der Kippgeschwindigkeit entspricht.

Der Beschleunigungssensor erfasst durch die Messung der richtungsabhängigen Erdbeschleunigung und deren Auswertung lediglich den Neigungswinkel.

Der Gyrosensor erfasst die Rotations- bzw. Kippgeschwindigkeit für eine (bis drei) Achse(n) und schiebt (nicht biegt) somit die äußeren Parabeläste nach oben, so dass an der vorherigen Spitze eine Delle entsteht.

Die Messwerte ergeben zusammen ein die Grundlage für die, den Motoren zugeführte Leistung, was letztendlich der Kugel den Bereich eines stabilen Gleichgewichtes zur Verfügung stellt, in der die Kugel immer wieder zum stabilen Punkt zurück läuft.

Wird der immer noch vorhandene Rand überschritten, so verlässt auch die Kugel den zulässigen selbstbalancierenden Bereich und stürzt ab, was ein Segway oder auch mein Runner ebenfalls erkennen lässt.

3. Die Mechanik

3.1. Mechanisches Konzept

Die Grundelemente, zwei Räder, eine Plattform und eine Lenker sind die notwendige Voraussetzung für das elegant wirkende Fortbewegungsmittel.

Geschwungene Lenker, Schutzbleche und Gepäckträger sind optische Accessoires, sie sind für die Funktion nicht erforderlich. Frontstrahler, Rückleuchte, Klingel usw. sind durchaus sinnvoll und ggf. notwendig.

Vorerst beschränkt sich die Beschreibung auf notwendige Elemente.

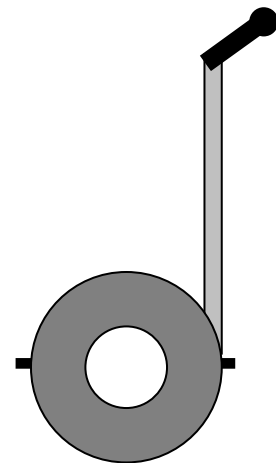


Abb. 3.1

3.2. Details eines mechanischen Aufbaus

Lenker und Stange aus Aluminium
Drehlager aus Fahrradpedal (rot)
Plattform Aluminium 5mm (schwarz)
Stahlachse d=20mm (dunkelgrau)
2-3 Blei-Gel-Akkus (grün)
Motoren mit Ritzel (dunkelrot)
Zahnräder mit Kette (blau)
Räder D=38cm (hellgrau)

Breite gesamt 69,5cm
Höhe gesamt 125cm

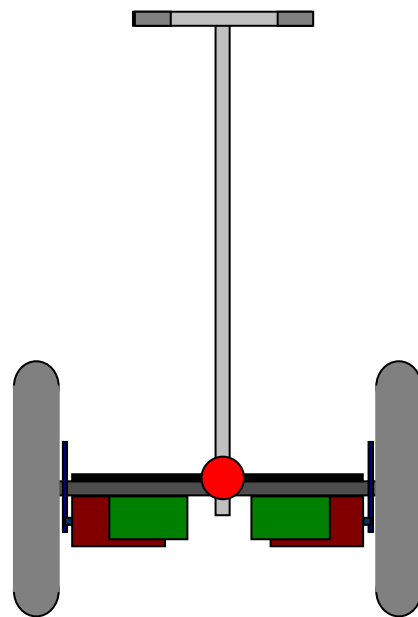


Abb. 3.2

3.3. Motoren und ihre Leistung

Bei der Auswahl der Motoren gilt es zu bedenken, dass der Motor zwar die Höchstgeschwindigkeit mitbestimmt, die jedoch nicht das Kriterium für seine Leistung ist.

Die Beschleunigung des Fahrzeugs erfolgt durch das Neigen des Lenkers nach vorne, wobei die Beschleunigung der Motoren und der Antriebe ein Überkippen nach vorne verhindert. Aus dieser Grundüberlegung ist ersichtlich, dass die höhere Motorleistung auch ein höheres Maß an Sicherheit bietet und zwar gerade dann, wenn in einer Notsituation das Ausbalancieren eine größere Motorleistung erforderlich macht. So liest man in Foren, dass Versuche mit Motorleistungen von 2 x 200 Watt durchaus erfolgreich sind, das Beschleunigungsverhalten auch ohne Notsituationen jedoch als schwach empfunden wird. Viele Versuche des Autors haben gezeigt, dass bei einer bewusst starken Neigung nach vorne auch die beiden 500 Watt Motoren überfordert sein können. Dieser Fall ist nur ein einziges Mal in einer Extremsituation aufgetreten. Eine Abschätzung der Motorleistung ergibt sich aus nachfolgenden Überlegungen:

Besitzt das selbstbalancierende Fahrzeug eine Neigung mit dem Winkel φ gegenüber der Senkrechten, so muss, statisch betrachtet, mindestens der Kraftanteil $F_P = m \cdot 9,81 \text{ m/s}^2 \cdot \tan(\varphi)$ in Fahrtrichtung aufgebracht werden, um ein Kippen zu verhindern. Liegt im dynamischen Fall zusätzlich eine Kippbewegung vor, so ist eine größere Kraft erforderlich. Die Betrachtung des statischen Falls führt zu einer unteren Leistungsgrenze der Motoren, die bei der Dimensionierung der Antriebe hilfreich ist.

Neigungswinkel φ

Senkrechte Kraft F_S

Kraft parallel zur Fahrtrichtung F_P

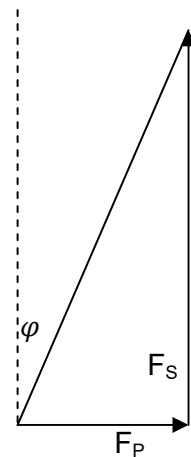


Abb. 3.3

Bei einer Fahrzeugmasse von 35kg und einem Fahrergewicht von max. 115kg ergibt sich eine Gesamtmasse $m=150\text{kg}$. Soll eine Neigung $\varphi = 10^\circ$, die beim Befahren eines Hindernisses auftreten kann, durch eine Beschleunigung aufgefangen werden, so liefern physikalische Grundlagen eine erforderliche Mindestleistung für diesen Zustand, während das Fahrzeug mit der Geschwindigkeit v fährt.

Dabei gilt für die Leistung: $P = F_p \cdot v$. Für die relativ kleine Geschwindigkeit $v = 1\text{ m/s} = 3,6\text{ km/h}$ ergibt sich im statischen Fall, d.h. wenn eine Neigung aber keine weitere Kippbewegung vorhanden ist, die Mindestleistung, die ein weiteres Kippen verhindert, wie im folgenden Beispiel mit der Annahme einer Gesamtmasse von 150kg gezeigt wird:

Es gilt: $F_s = F_g = m \cdot g = 150\text{ kg} \cdot 9,81\text{ m/s}^2$ und $F_p = F_s \cdot \tan(\varphi)$

→ $P_{\min} = 150\text{ kg} \cdot 9,81\text{ m/s}^2 \cdot \tan(10^\circ) \cdot 1\text{ m/s} = 255\text{ Watt}$

$v / (\text{m/s})$	$v / (\text{km/h})$	$P_{\min} / \text{Watt}$
1	3,6	255
3	10,8	765
5	18	1275

Tabelle 3.1 für $\varphi = 10^\circ$

Bei konstanter und unbehinderter Fahrt treten solche Neigungen eher nicht auf. Legt man einen geringeren max. Neigungswinkel von z.B. $\varphi = 5^\circ$ zugrunde, so erhält man nachfolgende Tabelle:

$v / (\text{m/s})$	$v / (\text{km/h})$	$P_{\min} / \text{Watt}$
1	3,6	128
3	10,8	384
5	18	640

Tabelle 3.2 für $\varphi = 5^\circ$

Diese zweite Rechnung mit $\varphi = 5^\circ$ zeigt, dass eine Gesamtleistung von 1000 Watt bei zwei Motoren auch bei höherer Geschwindigkeit das Fahrzeug stabil halten kann, wenn keine größeren Neigungen auftreten.

Eine stärkere Neigung ($\varphi = 10^\circ$) kann bei höherer Geschwindigkeit nicht ausgeglichen werden und führt unweigerlich zum Sturz.

Wichtig: Bei dieser ersten Abschätzung wurde eine möglicherweise vorhandene Kippbewegung **nicht** berücksichtigt. Durch diese Bewegung werden die hier angegebenen Minimalwerte noch größer.

Eine stark vereinfachte Abschätzung der Leistung, die zum Auffangen einer Kippbewegung erforderlich ist, ergibt sich aus nachfolgenden Überlegungen.

Vereinfacht wird dabei angenommen, dass sich die gesamte drehbare Masse in einer Höhe von $r=1\text{m}$ über dem Drehpunkt der Räder befindet. Liegt vor einem Sturz eine Kippbewegung, also eine Drehung um die Radachse mit einer Neigungsgeschwindigkeit von $d\varphi = 10^\circ \text{ in } dt = 0.1\text{s}$ vor, so ergibt sich daraus eine Winkelgeschwindigkeit $\omega = \frac{d\varphi}{dt} = 100^\circ/\text{s} = 1,74 \text{ rad/s}$.

Die aus der kinetischen Energie abgeleitete Rotationsenergie $W = \frac{1}{2}mv^2 = \frac{1}{2}m(r \cdot \omega)^2$ führt zu $W = \frac{1}{2} 150\text{kg} (1\text{m} \cdot 1,74\text{rad/s})^2 = 225 \text{ J}$.

Soll diese Rotationsbewegung in einer Sekunde ausgeglichen werden, so ist eine zusätzliche Motorleistung von rund 225 Watt erforderlich. Legt man eine halbe Sekunde zugrunde, so resultiert daraus ein Leistungswert von 550 Watt.

Aus diesen Überlegungen ist erkennbar, dass die erforderliche Motorleistung wesentlich von der Fahrsituation abhängt. Ein Selbstbauer beschrieb sein Fahrzeug, in welchem zwei Motoren mit jeweils 250Watt eingebaut waren, mit den Worten „geradeso ausreichend“. Für sein nächstes Fahrzeug hatte er leistungsstärkere Motoren vorgesehen.

Bei vielen Selbstbauern finden sich häufig Motoren mit einer Leistung von 2×500 Watt, sie werden im Allgemeinen als ausreichend empfunden.

Diese Motoren werden aus zwei bis vier 12Volt Bleigel-Batterien gespeist. Eine Reihen- und Parallelschaltung führt zu einem Batteriepaket mit 24Volt und einer Kapazität von 14Ah.

Bei der Verwendung von Motoren für eine Betriebsspannung von 24V hat sich auch der Betrieb mit 36V bewährt. Der Betrieb der Motoren mit einem PWM-Signal reduziert die zugeführte Leistung auf das jeweils notwendige Maß, welches in der Regel nicht über 500 Watt pro Motor liegt, so dass die Motoren keinen Schaden nehmen und trotzdem eine hohe Leistungsreserve zur Verfügung steht.

Wichtig: Diese Rechnungen stellen lediglich eine Abschätzung dar. Diese Rechnung ist nicht vollständig und zeigt nur exemplarisch, welche Leistungen erforderlich werden können. Für das bewältigen kritischer Situation ist die Summe der genannten Leistungen nicht ausreichend, insbesondere dann, wenn das Fahrzeug sich mit größerer Geschwindigkeit bewegt.

Alle Risiken hängen letztendlich von der selbst gewählten Fahrweise ab.

3.4. Räder und Antriebe

Der Kontakt zum Boden ist wichtig, jedoch nicht problematisch. Die auftretenden Kräfte und Geschwindigkeiten führen in der Regel nicht zum Rutschen auf trockener Straße. Geeignet sind somit fast alle Bereifungstypen, die verfügbar sind.

Bei einem geeigneten Durchmesser zwischen 36cm und 40cm bieten sich Räder von sog. Pocket Bikes, Dirt Bikes, Schubkarrenräder, Räder von elektrischen Rollstühlen oder Motorrollern an.

Auch kleinere Räder mit einem Durchmesser von ca. 20cm findet man bei einigen Selbstbauern. Ein besonderes Augenmerk ist dabei auf die verbleibende Bodenfreiheit zu richten. Ca. 5cm sollten mindestens bleiben, um kleine Hindernisse unbeschadet überrollen zu können. Dabei ergibt sich die konstruktive Herausforderung Motoren und Batterien über der Grundplatte anzuordnen. Dabei werden Motoren und Batterien mit einer zweiten Platte abgedeckt.

3.4.1. Übersetzungen

Das Übersetzungsverhältnis bestimmt die Geschwindigkeit aber auch die Kraft, mit der das Fahrzeug beschleunigt werden kann. In der Praxis ist eine hohe Geschwindigkeit zweitrangig. Wichtig ist, dass zur Beschleunigung eine ausreichende Kraft zur Verfügung steht, um in jeder Situation das Fahrzeug ausbalancieren zu können.

Mehrere Beispiele sollen dies verdeutlichen.

Betrachtet man den Motortyp, der unter der Bezeichnung MY1020 zu finden ist, so stößt man auf ein breites Spektrum mit verschiedenen Leistungen und Ausführungen. Ein Standardmotor für Selbstbauer mit $P=500$ Watt bei 24Volt gibt diese Leistung bei 2500 Umdrehungen pro Minute ab.

3.4.2. Variante 1

Diese Motorausführung besitzt ein angebautes Getriebe mit einem Übersetzungsverhältnis 6,7:1. Man erhält rund 400 Umdrehungen pro Minute bei dem direkt befestigten Rad. Ein Raddurchmesser von 36cm ermöglicht theoretisch eine Höchstgeschwindigkeit von

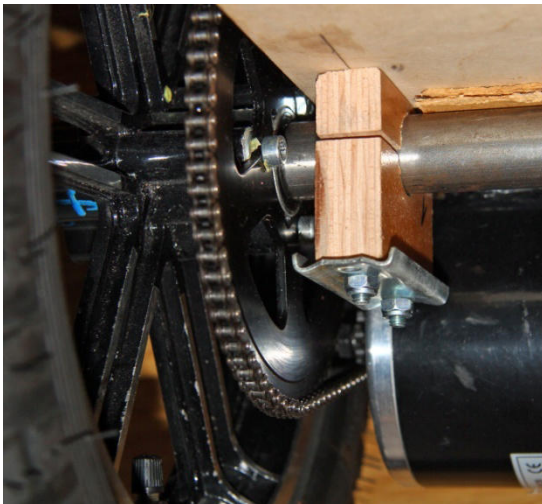


27km/h, die u.a. aus Gründen der zu geringen Motorleistung nicht praktikabel ist. Beim Befestigen des Rades an dem herausstehenden Achsenstück sind weitere Spezialteile erforderlich. Die Belastung der kurzen Achsstummel durch das Gesamtgewicht von Fahrzeug und Fahrer ist übermäßig hoch und hat in vielen Fällen zur Beschädigung der Lager und dem notwendigen Austausch der Motoren geführt.

3.4.3. Variante 2

Eine Variante unter Verwendung des gleichen Motortyps besteht im Verzicht auf das Getriebe. In diesem Fall wird der Motor MY1020 mit einer Leistung von z.B. 500 Watt in der Regel mit einem Ritzel mit 11 Zähnen ausgeliefert. Große Kettenblätter sind mit bis zu 80 Zähnen in Zubehörshops für Pocket Bikes oder E-Scooter wie www.ritzelladen.de zu finden. Passende Ketten werden dort ebenfalls angeboten. Die daraus resultierende Übersetzung ist vergleichbar.

Im Gegensatz zur Befestigung der Räder am Motor ist nun eine durchgehende Achse erforderlich. Derartige Achsen mit einem Durchmesser von 20mm sind relativ einfach im Auktionshaus zu finden. Die Befestigung an der Grundplatte erfolgt mit Achsböckchen die ggf. auch in anderer Form selbst hergestellt werden können, wie das nachfolgende Bild zeigt.



Nebenstehende Abbildung zeigt den Motor, die Stahlachse mit Achshalter aus Holz sowie das große Kettenblatt, welches mit drei Schrauben an bereits vorhandenen Löchern an der Felge befestigt wurde. Diese Bauart ist wegen der durchgehenden Stahlachse hoch belastbar.

3.4.4. Variante 3

Völlig anders gestaltet sich die Verwendung spezieller Rollstuhlmotoren. Diese Ausführung besteht aus einem Motor, einem angebauten, geschlossenen Getriebe mit herausgeführter Achse. Zusätzlich ist das Antriebsrad fest auf der Achse aufgesetzt. Diese Bauart ist hoch belastbar. Bedenkt man, dass das zulässige Gesamtgewicht dieser Fahrzeuge über 250kg liegt, so erscheint diese Antriebsart auch für ein selbst balancierendes Fahrzeug geeignet.

Zwei Eigenschaften sind zu berücksichtigen.

Elektrische Rollstühle werden in der Regel mit Höchstgeschwindigkeiten von 6km/h und 10km/h hergestellt. Die zweite Bauart erfordert eine Betriebserlaubnis und ein Haftpflichtkennzeichen und ist an diesem zu erkennen. In diesem Fall werden Motoren mit einer Leistung von 350 Watt verwendet, eine Leistung, die durchaus erhöht werden kann.

Versorgt werden dies Motoren mit einer Betriebsspannung von 24 Volt. Im Zusammenhang mit einer PWM-Steuerung ist jedoch auch ein Betrieb mit einer Versorgungsspannung von 36 Volt möglich, ohne dass diese Motoren Schaden nehmen. Die Höchstleistung, die in einem Laborversuch mit 36 Volt und einer Stromaufnahme von 20A gemessen wurde, liegt kurzfristig deutlich über 500 Watt und ist auch für eine Notsituation ausreichend. Die Drehzahl und die damit verbundene Höchstgeschwindigkeit bei einer Versorgungsspannung von 36V liegen je nach Ausführung bei 20km/h bis 25km/h.

Die Raddurchmesser variiert bei Rollstühlen dieser Art von 20cm bis 40cm. Bei der Konstruktion des Fahrzeuges sollte eine ausreichende Bodenfreiheit von mindestens 5cm bedacht werden.

Die gewinkelte Gehäuseform dieser Antriebseinheit erfordert eine geschickte Verbindung mit der gewählten Bodenplatte. Wegen der fehlenden durchgehenden Achse ist eine Versteifung der Bodenplatte erforderlich.

Offizielle Bezugsquellen gibt es für Rollstuhlantriebe nicht. Hier ist man auf ein Auktionshaus angewiesen. Auf der Suche nach ausgesonderten Rollstühlen wird man ggf. auch bei Sanitätshäusern fündig, über die diese Fahrzeuge vertrieben und zurückgeholt werden. Auch Krankenkassen, die formal Eigentümer dieser Rollstühle sind, können bei der Suche behilflich sein.

3.4.5. Variante 4

Neben elektrischen Rollstühlen gibt es auch sogenannte Elektromobile. Diese Fahrzeuge mit einer Geschwindigkeit bis 15km/h besitzen ebenso geeignete Motor-Getriebe-Rad Kombinationen mit vergleichbaren Leistungen und der Möglichkeit einer erhöhten 36 Volt Versorgung.

Hier sind die Radgrößen recht klein, so dass eine geeignete Befestigung an der Bodenplatte erforderlich wird.

3.4.6. Variante 5

Kollektorlose Motoren, wie sie immer mehr als Nabenmotoren in Fahrrädern zu finden sind, haben zumindest in einem Fall zu einem erfolgreichen Antrieb geführt.

Die Ansteuerung erfordert eine spezielle Elektronik verbunden mit einer ausgeklügelten Software. Diese Motoren sind geräuschlos und ermöglichen ein absolut lautloses Gleiten. Auf der Videoplattform YouTube ist ein solches Fahrzeug unter Verwendung des Suchwortes „Rutscherle“ zu finden. Der Bezug der Motoren höherer Leistung, ggf. aus China, gestaltet sich schwierig. Ihr Einbau in eine Felge geeigneter Größe ist problemlos möglich. Speichensätze werden in Fahrradgeschäften millimetergenau angefertigt.

4. Strom muss fließen

4.1. Batterien / Akkus

Bereits erwähnt wurde eine Zusammenstellung der Akkus aus zwei 12V Bleigel-Akkus mit der Gesamtspannung von 24 Volt. Die Spannungsangaben sind sog.

Nennspannungen. Günstig zu erwerben sind geschlossene Batterien mit einer Kapazität von 7 Amperestunden (Ah), die zu einem Block mit den Daten 24 Volt / 7 Ah und 168 Wattstunden (Wh) zusammengefügt werden. Die Gesamtkapazität einer solchen Batterie kann jedoch nicht genutzt werden, wofür auch der an anderer Stelle beschriebene Peukert-Effekt verantwortlich ist.

Bei einem angenommenen Entladestrom von 9 Ampere (A) liegt die nutzbare Kapazität bei ca. 3,5 Ah liegen. Wegen des während der Entladung zunehmenden Innenwiderstandes wird zusätzlich die Hochstrombelastbarkeit reduziert, was bei einer erhöhten Stromentnahme zu einem Spannungseinbruch und Leistungsverlusten an den Motoren führt. Eine Verdopplung der Batteriekapazität durch das Parallelschalten von je zwei Akkus führt zu einem Energieblock mit 24 Volt / 14 Ah mit 336Wh. Die höhere Batteriekapazität reduziert den Peukert-Effekt und vermindert den Innenwiderstand, was sich insbesondere bei hohen Strömen als vorteilhaft erweist. Bei einer angenommenen Entladung des Akkus mit 9 Ampere stehen nun maximal 9 Ah zur Verfügung.

Wegen Strömen über 30A sollte die letztgenannte Kapazität der Akkus gewählt werden.

Eine weitere Verbesserung in mehrfacher Weise erreicht man durch die Verwendung eines 36 Volt Batteriepakets. Drei Bleigel-Akkus mit je 12 Volt bieten einen Energiespeicher mit 36 Volt / 14 Ah und 504 Wattstunden, wofür ein Gewicht von insgesamt 12,6 kg einzuplanen ist.

Die Energiedichte liegt in diesem Fall bei 40 Wh/kg.

Der Strom, der zum Erbringen einer bestimmten Motorleistung erforderlich ist, reduziert sich um ein Drittel, was auch den nachteiligen Peukert-Effekt weiter reduziert.

Die Alternative zu den beschriebenen, geschlossenen Bleigel-Akkus sind Lithium-Ionen-Akkus. Sie bieten gegenüber den beschriebenen Standard-Akkus deutliche Vorteile, aber auch Nachteile.

Lithium-Ionen-Akkus (Li-Ion) zeichnen sich durch ihre hohe Energiedichte aus. Der Zellentyp 18650, der mit einer Kapazität von rund 2 Ah angeboten wird, speichert bei einem Gewicht von ca. 40g rund 7,2 Wh. Die Energiedichte erreicht mit 180 Wh/kg mehr als das Vierfache der herkömmlichen Bleigel-Batterie. Ein Lithium-Ionen-Akku mit

einer Kapazität von 504 Wh wiegt somit rund 2,8kg verbunden mit einer Gewichtseinsparnis von ca. 10kg. Neuere Entwicklungen reduzieren das Gewicht zusätzlich.

Darüber hinaus ist der Li-Ion-Akku thermisch stabil, hat eine fast konstante Ausgangsspannung von 3,6 Volt pro Zelle über den gesamten Entladezeitraum und praktisch keine Selbstentladung. Die Entladeschlussspannung liegt bei 3 Volt. Die Ladeschlussspannung beträgt 4,2 Volt. Diese darf beim Ladevorgang auf keinen Fall überschritten werden. Die Ladezeit kann in der Regel bis auf eine Stunde reduziert werden.

Um einen 24 Volt Akku zu realisieren sind sechs bis 7 Zellen in Reihe zu schalten. Zehn Zellen ergeben einen 36 Volt-Akku. Eine gleichzeitige Parallelschaltung mehrerer Zellen ist zum Erreichen der gewünschten Kapazität erforderlich. Der Peukert-Effekt ist bei diesem Batterietyp bedeutungslos.

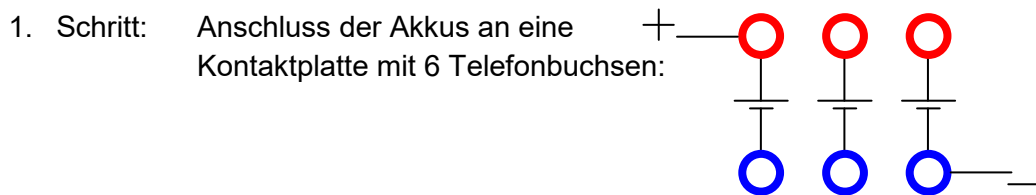
Die Kapazitätsentnahme ist somit auch bei hohen Strömen ohne Einschränkung möglich. Der Ladebetrieb erfordert ein spezielles Ladegerät, in dem ein sogenannter Balancer integriert ist, der das Übersteigen der maximalen Ladespannung von 4,2 Volt pro Zelle bei den in Reihe geschalteten Zellen verhindert.

Der hohe Preis dieser Zellen sowie die Explosionsgefahr im Falle eines Kurzschlusses, auch bei einzelnen Zellen, sind weitere Nachteile.

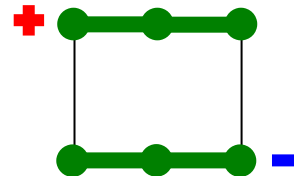
4.2. Laden der Akkus

Die serielle Anordnung der Akkus, die zu einer Gesamtspannung von $2 \times 14,4\text{V} = 28,8\text{V}$ bzw. $3 \times 14,4\text{V} = 43,2\text{V}$ führt, macht ein spezielles Ladegerät erforderlich. Die Verwendung eines Standard-Ladegerätes für einen 12V-Akku vereinfacht den Ladevorgang, macht aber ein Parallelschalten der Akkus notwendig. Ein geregeltes Ladegerät vermeidet dabei eine Überladung und Beschädigung der Akkus.

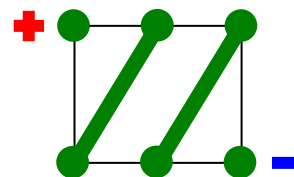
Die geschickte Anordnung der Akkuanschlüsse und deren Verbindung mit einem speziell ausgelegten Stecker erlaubt das parallele Laden und serielle Entladen der Akkumulatoren. Dieser Stecker mit 6 Bananensteckern muss dabei selbst angefertigt werden.



2. Schritt: Der Stecker Typ A erlaubt das parallele Laden mit 12V,
- Typ A



3. Schritt: Der Stecker Typ B ermöglicht das serielle Verbinden der Akkus zu 36V und den Anschluss der gesamten Elektronik.
- Typ B



4.3. Der Einschaltvorgang

Insbesondere die Verwendung eines 36V-Batteriepakets aber auch schon 24V führen beim Verbinden der Batterien mit der Elektronik zu einem kräftigen Funkenschlag. Die Kondensatoren der H-Brücken, einige tausend Mikrofarad, werden schlagartig geladen und verursachen im ersten Moment einen enorm hohen Ladestrom. Erfolgt das Einschalten über einen manuellen Schalter oder ein Relais, so kommt es zu einer starken Kontaktbelastung, die durchaus zum Verschmelzen der Schalt- bzw. Relaiskontakte führen kann. Ein gekapseltes Relais ist dann unbrauchbar. Das langsame Laden der Kondensatoren mit einem geringen Strom beseitigt dieses Problem. Vor dem eigentlichen Schließen der notwendigerweise hochstromfesten Kontakte im Relais oder Schalter ist es sinnvoll, die Kondensatoren über einen Widerstand fast vollständig zu vorzuladen. Im vorgestellten Konzept wird während dieser Ladephase der Mikrocontroller aktiviert. Er misst die aktuelle Spannung, die während des Ladevorgangs vorliegt und schaltet das Relais erst ein, wenn die Kondensatoren fast vollständig geladen sind. Das Schalten der Relais ist deutlich zu hören. Danach ist das Fahrzeug betriebsbereit.

5. Elektronisches Konzept

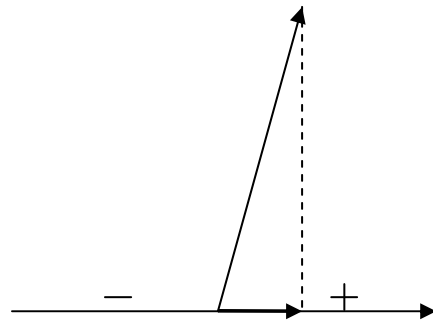
Selbstbalancierende Fahrzeuge benötigen ein Organ, vergleichbar dem Gleichgewichtsorgan im menschlichen Innenohr. Dieses Organ registriert zum einen die Abweichung von der senkrechten Stellung und zu zweitem eine Neigungsgeschwindigkeit, wenn der Mensch nach vorne oder hinten kippen sollte. Beide Sensorwerte, die im menschlichen Gehirn verarbeitet werden und für das aufrechte Stehen und Gehen verantwortlich sind, müssen auch bei der elektronischen Gleichgewichtssteuerung erfasst und ausgewertet werden.

5.1. Sensoren

5.1.1. Der Beschleunigungssensor

Ein z.Z. häufig verwendeter Beschleunigungssensor mit der engl. Bezeichnung „Accelerometer“ besteht aus drei Sensoren, welche Neigungen bezüglich der Richtung von drei zueinander orthogonalen Achsen messen. Nebenstehendes Beispiel zeigt die Auswirkung der Erdbeschleunigung auf einen geneigten Sensor. Je nach Neigungsrichtung entsteht (hier horizontal dargestellt) ein Messwert, der um einen mittleren Wert schwankt. Dieser Messwert ist ein Maß für die Neigung des Sensors und letztendlich auch für die Neigung des Fahrzeuges.

Abb. 5.1



5.1.2. Das Gyroskop

Der verarbeitende Mikrocontroller erzeugt aus dem Ausgangssignal des Neigungssensors ein Steuersignal für die Motoren. Genügt bei einer kleinen Neigung ein schwacher Antrieb für die Motoren so kann der Leser leicht nachvollziehen, dass der notwendige Antrieb deutlich stärker sein muss, wenn diese kleine Neigung kein statischer Zustand ist, sondern gleichzeitig eine Kippbewegung stattfindet. Dabei gilt es auch die Kippgeschwindigkeit durch eine verstärkte Motorleistung aufzufangen. Ein zweites Set von Sensoren, Gyroskope, registrieren diese Kipp- bzw. Neigungsgeschwindigkeit.

5.1.3. Die Gierrotation

Die Fahrt auf dem zweirädrigen Gefährt gleicht einem Schweben, im Idealfall ohne fühlbaren Bodenkontakt. Trifft jedoch ein Rad auf einen kleinen Stein oder ein anderes Hindernis, so wird dieser Idealzustand gestört, das behinderte Rad verlangsamt seine Fahrt und das Fahrzeug führt eine mehr oder weniger starke Drehbewegung um die z-Achse eines virtuellen Koordinatensystems aus. Dieses Drehen trotz einer gewünschten Geradeausfahrt bezeichnet man als Gieren.

Das vorhandene Gyroskop kann auch zur Erkennung dieser Rotation um die z-Achse genutzt werden.

Das winzige elektronische Gyroskop besitzen drei Sensoren, die jeweils die Rotation um eine von drei möglichen Achsen messen. Die geschickte Anordnung eines solchen Bausteins ermöglicht das Erfassen der Kipprotation und der Gierrotation.

5.1.4. Mikrocontroller als zentrale Steuerung

Kennen Sie **EVA**?

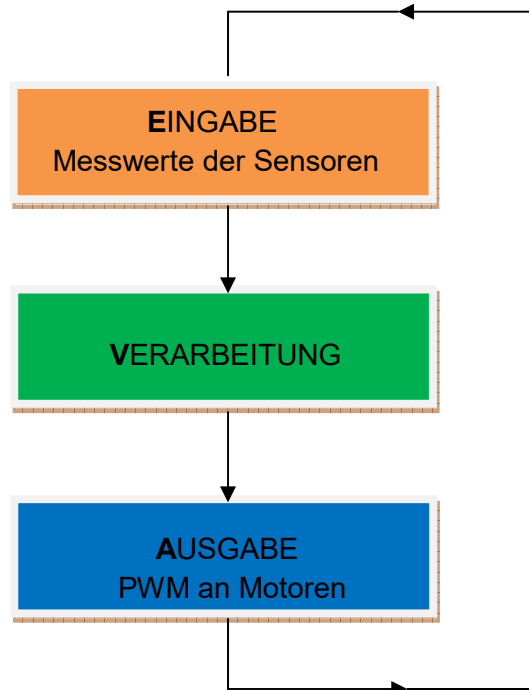


Abbildung 5.2

In diesem Fall als permanenter Kreislauf, um auf jede Störung der Gleichgewichtslage unmittelbar zu reagieren.

Aus der bisherigen Beschreibung ergeben sich folgende Aufgaben im Programmteil **EINGABE**:

Messung der Batteriespannung	(nicht unbedingt erforderlich)
Messung der Stromstärke	(nicht unbedingt erforderlich)
Erfassung der Neigungsdaten	(notwendig)
Erfassung der Kipprotation	(notwendig)
Erfassung der Gierrotation	(empfehlenswert)
Erfassung der Lenkerstellung	(notwendig)

Im Programmteil **VERARBEITUNG** erfolgt nun eine mathematische Bearbeitung und Auswertung der erfassten Daten.

Der wesentliche Teil der **AUSGABE** besteht im Erzeugen eines PWM-Signals zu Steuerung der Motoren. Ebenso wird die Drehrichtung der Motoren festgelegt. Betriebswerte können auf einem LC-Display ausgegeben werden.

5.2. Der Mikrocontroller

Um die genannten Aufgaben zu erledigen, wurde der Mikrocontroller ATMEGA32 ausgewählt.

Dieser arbeitet mit einer Taktfrequenz von 16MHz. Viele der insgesamt 131 elementaren CPU-Befehle werden in nur einem Zyklus ausgeführt, so dass eine schnelle Verarbeitung möglich ist.

Es stehen 32Kbytes Flash Speicher als Programmspeicher zur Verfügung.

Ein EEPROM mit 1024 Byte dient der Ablage von Parametern und Offsetwerten, die bei jedem Programmstart zur Verfügung stehen müssen.

Das 2048 Byte große interne SRAM nimmt die für Berechnungen erforderlichen Variablen des Programmes auf.

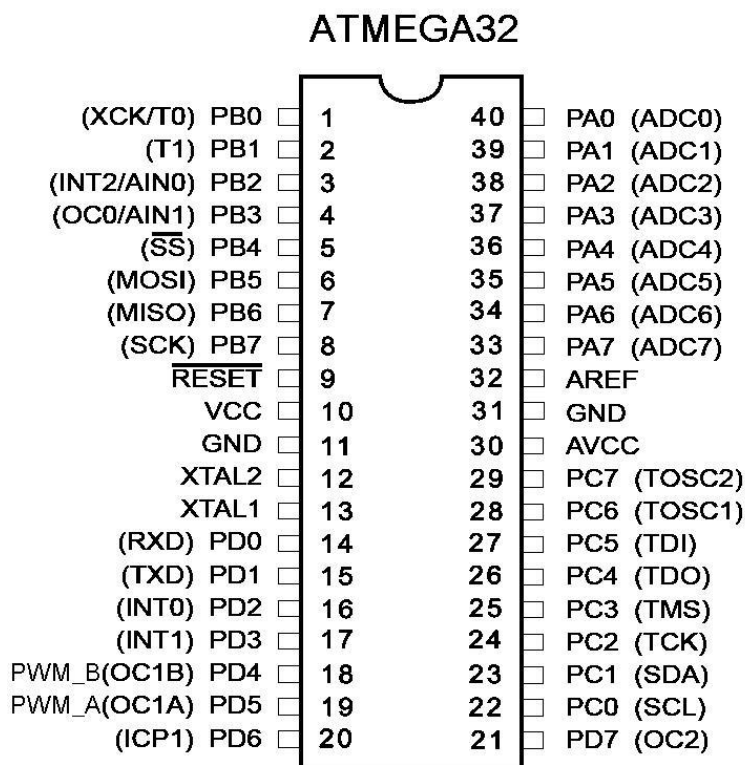


Abbildung 5.3

Wesentliche Bestandteile des Controllers sind vier Ports (A, B, C, D) mit jeweils 8 Bit. Jedes Bit kann als digitaler Eingang, digitaler Ausgang bzw. in einer Sonderfunktion genutzt werden. So kommen alle Anschlüsse des Port A mit ihrer Sonderfunktion als

Analog-Digital-Wandler zum Einsatz. Die Anschlüsse PD4 und PD5 des Port D werden in ihrer Funktion als PWM-Ausgänge genutzt. Die Funktion der Anschlüsse wird in Initialisierungsteil des Programmes festgelegt.

PIN-kompatible Mikrokontroller sind Atmega324 mit einer zulässigen Taktfrequenz von 20 MHz für den Fall, dass der Rechenaufwand umfangreicher wird. Einen zusätzlich größeren Speicher bietet der Typ Atmega642.

5.2.1. Auslastung des Mikrocontrollers

Aus Sicherheitsgründen gab es mehrere Versuche die Auslastung der CPU zu bestimmen.

So wurde eine erste Messung so durchgeführt, indem beim Eintritt in den Interrupt ein Portbit gesetzt wurde, welches vor dem Verlassen wieder zurückgesetzt wurde. Mit einem angeschlossenen Oszillographen wurde die Zeitdauer mit ca. 2,5ms ermittelt. Dieser Interrupt wird 100 Mal in einer Sekunde aufgerufen, so dass rund 250ms hierfür anzusetzen sind.

In gleicher Weise wurde ein Portbit beim Einstieg in die LCD-Ausgabe gesetzt und nach der Abarbeitung zurückgesetzt, wobei ca. 300ms für diese Aufgaben ermittelt wurden. Da in diesen 300ms rund 30 Interrupts der Regelroutine à 2.5ms enthalten sind, kann für die LCD-Ausgabe eine Nettozeit von 225ms angenommen werden. Im Ergebnis entstehen rund 525ms für CPU-Zeiten innerhalb einer Sekunde. Diese grobe Abschätzung führt zu einer 55%igen Auslastung.

Um derartige Messungen nach größeren Änderungen zu vermeiden, wäre eine Softwarelösung eleganter und weniger aufwändig.

In diesem Fall führte eine weitere Optimierung zu einer verminderten Auslastung der CPU, so dass diese probeweise bei einem Betrieb mit einem 8 MHz Takt auch keine Überlastung erfahren hat.

Eine interne softwarebasierende Auslastungsbestimmung ließ sich nach einigen Überlegungen recht einfach realisieren.

Die folgende Laufschleife wird, wenn keine LCD-Ausgabe vorliegt und kein Interrupt erfolgt, mit hoher Häufigkeit durchlaufen. Die Durchläufe werden durch die Variable *Freilauf* gezählt.


```

Do
  Incr Freilauf

  If Loop_count = 100 Then      'LCD –Ausgabe bei Loop_count = 100
    Loop_count=0

    .
    Freilauf% = Freilauf / 4000  '%-Berechnung aus den Freiläufen
    LCD 100 - Freilauf%         'Ausgabe der Auslastung in %
    .
  End If
  .
  .
  .
Loop
.
.
.
Calculate_balance:              'Interruptroutine zur Balanceregulierung
.
Return

```

Für die rot gekennzeichneten Zeilen – ohne Aufruf eines Interrupts - gibt der BASCOM-Simulator 29 CPU-Takten. Eine andere Methode liefert als Ergebnis 39 CPU-Takte, in der Praxis wurden 40 Takte festgestellt, die auch bei den weiteren Überlegungen Verwendung finden. Geht man von einer Taktfrequenz von 16MHz aus, so sind in einer Sekunde $16\,000\,000 / 40 = 400\,000$ Durchläufe möglich. Dies setzt voraus, dass keine Interrupts und keine LCD-Ausgabe auftreten. Um später direkt eine %-Angabe zu erhalten wird mit einem entsprechend kleineren Wert 4000 weitergerechnet.

Jeder Interrupt, der CPU-Takte für seine Abarbeitung benötigt sowie jede LCD-Ausgabe reduzieren die Anzahl der der Durchläufe dieser *Do ... Loop* –Schleife.

Erreicht die Variable *Freilauf* den Wert 300000, für nur 300000 statt 400000 Durchläufe der *Do ... Loop* –Schleife, so ergeben $300\,000/400\,000 = 0.75$ bzw. $300\,000/4000 = 75.0$, welches dem Freilauf-Prozentwert entspricht.

Die Auslastung beträgt somit $100\% - 75.0\% = 25.0\%$.

Nach diesem Prinzip berechnet der Mikrocontroller seine Auslastung selbst und zeigt ab der Softwareversion V7.09 einen Wert von ca. 25% an.

Ein beruhigendes Gefühl, da jede Softwareerweiterung automatisch erfasst wird.

Wegen der unterschiedlichen Ergebnisse für die CPU-Takte, 29, 39 bzw. 40, wurde deren Gültigkeit durch das Einfügen eines *Waitms 200* – Befehles in die LCD-Ausgabe geprüft.

Diese Pause erhöht die CPU-Auslastung pro Sekunde um 200ms, also um 20%.

Genau dieses Ergebnis wird durch die entsprechende erhöhte Anzeige bei der Verwendung von 40 CPU-Takten bestätigt.

Ein Atmega324 mit einem 20MHz-Quarz wurde zu Vergleichszwecken genutzt.

Theoretisch benötigt der Atmega32 mit 16MHz und einer Auslastung von 25% in einer Sekunde 4.000.000 Takte, um seine Aufgaben zu erledigen.

Diese 4.000.000 Takte führen beim Atmega324 mit 20MHz zu einer Auslastung von 20%. Die Anzeige im Display bestätigte diese Überlegungen.

5.3. Schaltung der Hauptplatine

Vor dem Erstellen einer Schaltung ist eine Zusammenstellung aller notwendigen Funktionen erforderlich.

Die nachfolgende Auflistung orientiert sich an der Reihenfolge der Ports und berücksichtigt auch bisher nicht genannte Funktionen.

5.3.1. Funktionen des Port A

Wie bereits erwähnt, werden die Anschlüsse dieses Ports in ihrer Funktion als analoge Eingänge benutzt.

Ab Software Version **V7.xx** mit digitalem Sensor:

PA0	Eingang für die Messung der Batteriespannung
PA1	SCL für I2C-Anschluss
PA2	SDA für I2C-Anschluss
PA3	frei (Steckerleiste)
PA4	frei (Steckerleiste)
PA5	Eingang für Lenkungssignal
PA6	Reserve (JP1.1)
PA7	Strom-Sensor (JP1.2)

5.3.2. Funktionen des Port B

Die Anschlüsse des Portes B erfüllen z.T. zwei Funktionen, sie werden zur Programmübertragung und für das LC-Display benutzt.

PB0	LCD-Datenbit 4	
PB1	LCD-Datenbit 5	
PB2	LCD-Datenbit 6	
PB3	LCD-Datenbit 7	
PB4	LCD-RS	
PB5	LCD-E	Progr. MOSI
PB6		Progr. MISO
PB7		Progr. SCK

5.3.3. Funktionen des Port C

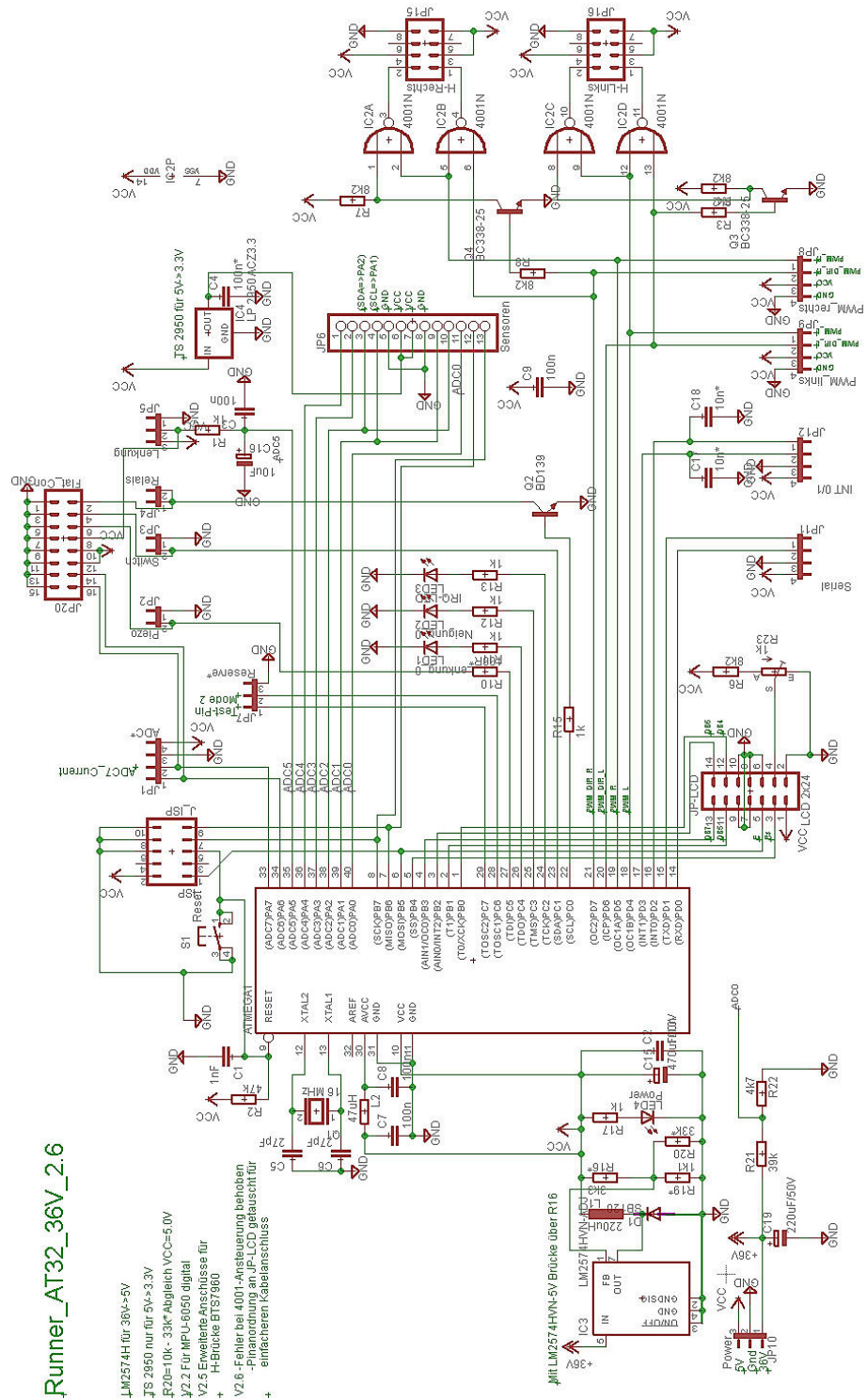
PC0	Ausgang Schalttransistor-Relais
PC1	Eingang Fußschalter Switch
PC2	Ausgang LED-IRQ
PC3	Ausgang LED Neigung
PC4	Ausgang LED Lenkung
PC5	Ausgang LED Piezo
PC6	Eingang (JP7.2) siehe BTS-Modus
PC7	Eingang (JP7.1) siehe Testmodus

5.3.4. Funktionen des Port D

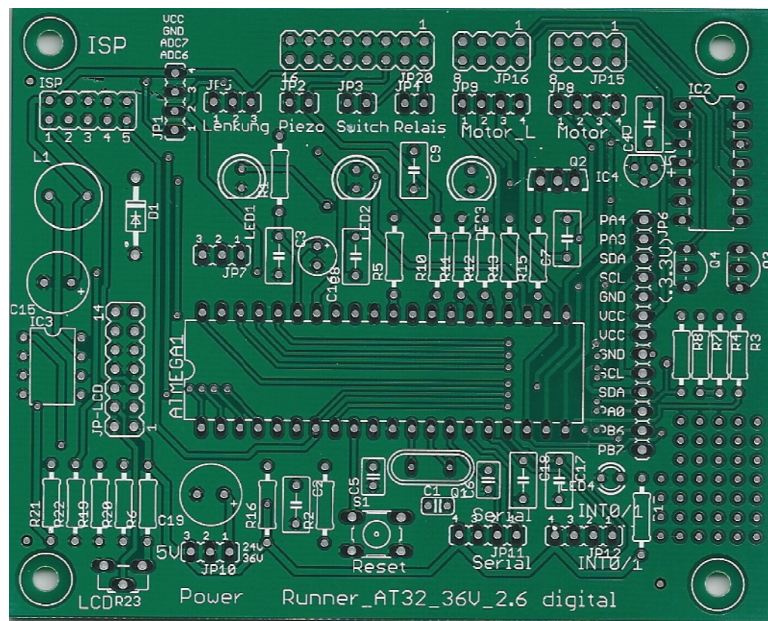
PD0	Serieller Eingang RXD (auch für PC-Tastatur)
PD1	Serieller Ausgang TXD (auch für PC-Tastatur)
PD2	Interrupteingang INT0 (Geschwindigkeitsmessung)
PD3	Interrupteingang INT1 (frei)
PD4	Ausgang PWM-linker Motor
PD5	Ausgang PWM-rechter Motor
PD6	Ausgang Drehrichtung linker Motor
PD7	Ausgang Drehrichtung rechter Motor

Die Funktion der hier nicht genannten Anschlüsse ergibt sich aus dem Schaltplan.

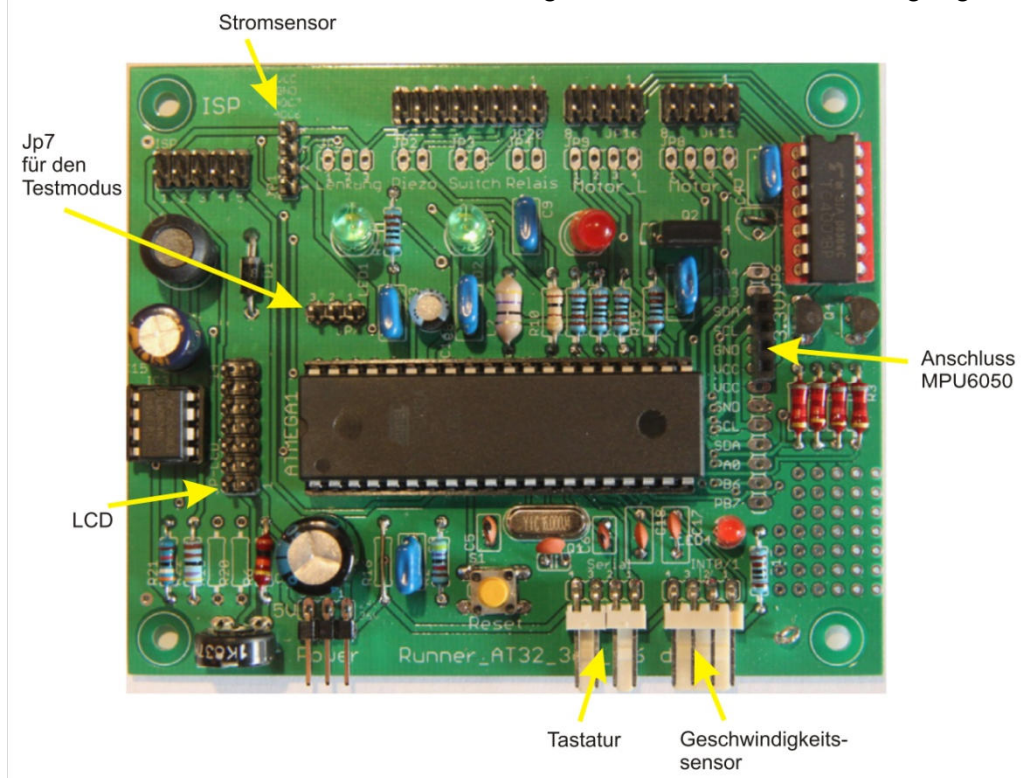
5.3.5. Schaltplan der Hauptplatine 2.6



... und die Platine M_2.6



Ein kleines Lochrasterfeld steht für Erweiterungen der Elektronik zur Verfügung.



Die fertige bestückte Platine verwendet hier einen Spannungsregler LM2475N -5V, der sich durch eine maximal zulässige Betriebsspannung von 45V auszeichnet.

Die Widerstände (R16-Brücke) des vorgesehenen Spannungsteilers entfallen (R19, R20).

Im Layout wurde eine weitere Anpassung durchgeführt, so dass ein aktuell verfügbares 2 x 24 - LCD ohne paarweise getauschte Verbindungsdrähte angeschlossen werden kann.

5.4. ! – Fast ein Fehler – in Schaltung und Layout

Das schwaches „Zucken“ eines Motors beim Einschalten der Betriebsspannung regte zum Nachdenken an.

Es zeigte sich, dass beim Startbeginn des Atmega alle Anschlüsse als Eingänge definiert sind. So treffen die Eingänge 5/6 und 9/12 des Baustein C4001 auf die noch nicht als Ausgänge definierten PINs 18 und 19 des Atmega32. Der für Bruchteile einer Sekunde zufällige Zustand „0“ ist für die Reaktion des Motors verantwortlich. Danach wird im Programm der notwendige Zustand „1“ festgelegt.

Die Abhilfe ist einfach. Wie bereits auf der Platine der früheren Version der H-Brücke verhindert ein Widerstand (10k – 100k gegen VCC) an den Eingängen 5/6 und 9/12 des Baustein C4001 den Zustand „0“, so dass die Motoren nicht angetrieben werden. Diese Widerstände sind nachträglich einzufügen. Eine geeignete Positionen für diese Widerstände sind JP8/1-3 und JP9/1-3.

5.5. Ansteuerung der Motoren

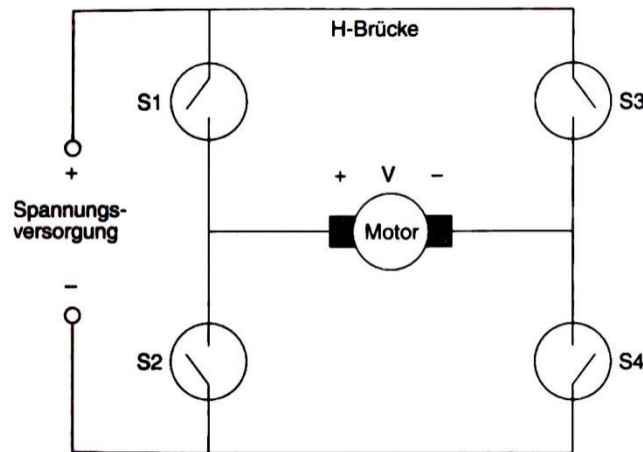
Der Motor wird aus einer Reihenschaltung von drei Batterien gespeist. Die aufgenommene Leistung erreicht dabei mehr als 1000 Watt bzw. mehr als 1,36PS. Kommt es zu einem unkontrollierten Einschalten des Motors, so besteht eine nicht zu unterschätzende Verletzungsgefahr. Ein Selbstbauer berichtet: „Ich habe in meiner ganzen Zeit als Fußballer nicht so häufig blaue Schienbeine gehabt wie zu der Zeit, als ich mein Fahrzeug gebaut habe.“

Der Mikrocontroller schaltet die Motoren ein und aus, bestimmt die Drehrichtung und die Geschwindigkeit.

Der wesentliche Bestandteil der Motorsteuerung sind die nun folgenden H-Brücken. Diese MOSFET-Schaltung übernimmt die Ausführung der genannten Aufgaben.

5.6. Die H-Brücke

Mit dieser Anordnung der Schalter und des Motors erreicht man folgende Funktionen:

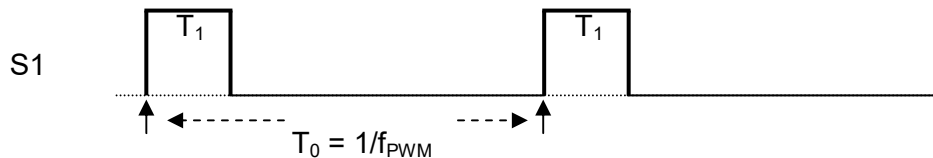


Alle Schalter offen	->	Stillstand
S1 und S4 geschlossen	->	Motor rechtsdrehend
S2 und S3 geschlossen	->	Motor linksdrehend
S2 und S4 geschlossen	->	Motorbremse

Im dritten Fall wird der Motor kurzgeschlossen. Da der Motor gleichzeitig einen Generator darstellt, verursacht die induzierte Spannung einen Stromfluss, welcher sich hemmend auf die Drehbewegung des Motorankers auswirkt. Diese Bremswirkung kann unterschiedlich stark sein.

6. Was ist eine PWM-Steuerung?

Wird in der vorausgegangenen Abbildung bei geschlossenem Schalter S4 der Schalter S1 im Laufe einer Sekunde jeweils für zwei Zehntel Sekunden geschlossen und bleibt er in den restlichen acht Zehntel Sekunden offen, so wird dem Motor eine auf 20% reduzierte Leistung zugeführt.



6.1. PWM-Modi

Mit dem oben gezeigten Bild ist es möglich, einen Motor in jede Drehrichtung laufen zu lassen, je nach dem welcher Anschluss des Motors über einen Schalter mit dem Minuspol der Spannungsquelle verbunden ist.

Da eine zweite Drehrichtung unbedingt erforderlich ist, erzeugt der Mikrocontroller auch ein Richtungssignal, welches in der H-Brücke verarbeitet werden muss.

Ab der Softwareversion V7.36 wird der PWM-Modus, der speziell für BTS-Module enthalten war, nicht mehr unterstützt, da beide Versionen der H-Brücke in gleicher Weise angesteuert werden.

Gleichzeitig werden andere PWM-Modi eingeführt, die im Folgenden beschrieben werden.

Bei der Weiterentwicklung der Hauptplatine wurde der Baustein CD4001 von der Platine der H-Brücke auf die Hauptplatine übernommen.

Dieser Baustein ist bei den nachfolgenden H-Brücken nicht mehr vorgesehen.

Der Anschluss der H-Brücken ab H_2.8 erfolgt wie auch der Anschluss der BTS-Module über das Flachbandkabel.

Die Hauptplatinen werden zukünftig z.B. mit M_2.6 bezeichnet.

Während bei der Hauptplatine M_2.6 das Flachbandkabel an JP15, 16 und 20 angeschlossen werden konnte und alternativ auch die JP 8 und JP9 zur Verfügung standen, werden die letztgenannten Anschlüsse bei Platinen ab M_2.7 entfallen.

6.1.1. PWM-Modus 1

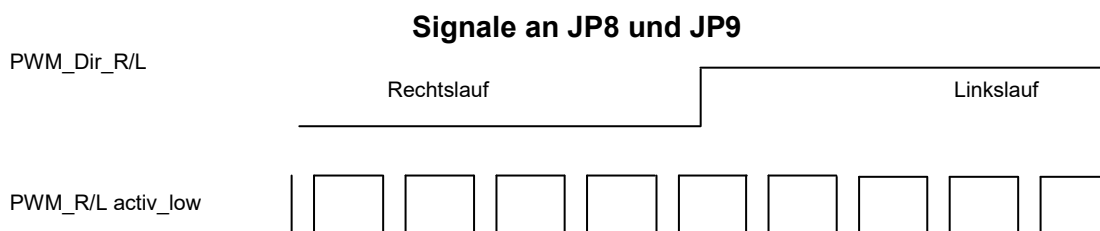
Anwendung für die H-Brücken bis Version H_2.7 an JP8 und JP9 sowie H-Brücken ab Version H_2.8 ohne CD4001 an JP15 und JP16 und für H-Brücken Module mit BTS7960 an JP15 und JP16.

Im Modus 1 stehen am Mikrocontroller die Signale für PWM und Drehrichtung zur Verfügung.

Die hier gezeigte Signalform wird an den Steckverbindungen JP8 und JP9 ausgegeben.

Die beiden Signale beinhalten die Drehrichtung und das eigentliche PWM-Signal, welches im Zustand „low“ einen Stromfluss durch den angeschlossenen Motor zur Folge hat. Das Ein- und Ausschalten der einzelnen Halbbrücken erfolgt durch die Bausteine CD4001N und IR2184 auf der Platine der H-Brücke.

Dies trifft für alle die von mir geplanten H-Brücken H_2.x zu.



Dieses PWM-Signal erzeugt der Atmega32x intern durch das entsprechende Programmieren geeigneter Register.

Bei der Hauptplatine M_2.5 wurden einige Veränderungen vorgenommen. Für die Erzeugung des PWM-Signales wurde der Baustein CD4001N auf die Platine übernommen. Je zwei NOR-Gatter und ein Inverter, die pro H-Brücke erforderlich sind, sind somit auf der Platine der H-Brücke nicht mehr erforderlich.

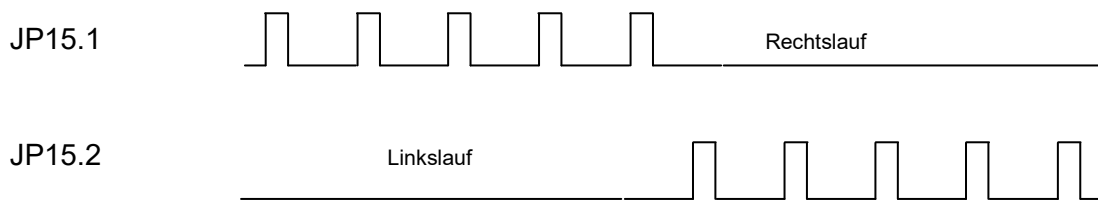
Die daraus gewonnenen PWM-Signale werden direkt dem Halbbrückentreiber IR2184 an PIN 1 über die 4k7 Widerstände zugeführt. Auf der Platine der H-Brücke entfallen entsprechende Bauteile incl. 5V- Spannungsregler

Signale an JP15 und JP16

Die durch den Baustein CD4001N erzeugten PWM-Signale vom Typ „activ-high“ stehen an den Jumpers JP15 und JP16 zur Verfügung und erscheinen in gleicher Form auch an den Ausgängen der H-Brücke.

Jeweils ein Ausgang ist dabei mit GND verbunden, der andere Ausgang liegt auf dem Potential der Versorgungsspannung mit 24V bzw. 36V.

Der Signalverlauf an JP15.1 und JP15.2 zeigt das kurzzeitige Schalten der aktiven Ausgänge.



Neben der Ansteuerung der H-Brücken ab H_2.8 (ohne den Baustein CD4001) können auch H-Brücken Module mit den ICs BTS7960 direkt angesteuert werden.

6.1.2. PWM-Test

Vor dem Anschluss der Motoren muss die korrekte Erzeugung des PWM-Signales geprüft werden.

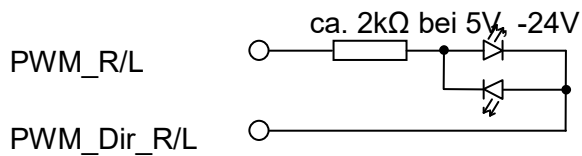
Dies geschieht in zwei Stufen:

6.1.2.1. Test im Modus 1

Wichtig: Der Modus 1 ist ausgewählt, wenn Jumper JP7/2-3 nicht eingesteckt ist.

In diesem Fall kann der entsprechende Signalverlauf mit einem Oszilloskop an den PINs PWM_Dir_R/L und PWM_R/L direkt geprüft werden.

Ein kleiner PWM-Tester bestehend aus zwei LEDs und einem Widerstand ist sehr hilfreich.



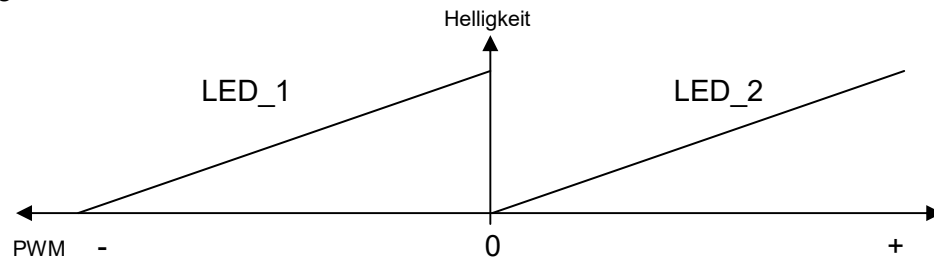
Im Modus 1 leuchtet beim Rechtslauf und sehr kleinen PWM-Werten eine LED hell, da am PWM_R/L-Ausgang fast immer 5V anliegen und an PWM_Dir_R/L 0 Volt ausgegeben werden.

Bei zunehmender Neigung nimmt die Helligkeit ab.

Erfolgt eine Neigung der Sensoren in die andere Richtung so springt die Spannung an PWM_Dir_R/L auf 5V und stimmt mit der Spannung an PWM_R/L überein. Bei zunehmender Neigung und somit ansteigenden PWM-Werten wird die andere LED immer heller werden.

Statt die Hauptplatine zu neigen, kann auch die vorherige Mittelstellung des Lenker-Potentiometers verändert werden.

Helligkeitsdiagramm Nr. 1 der LEDs



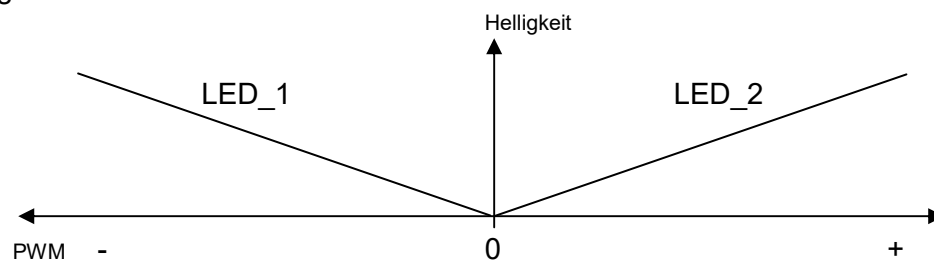
Nun kann die H-Brücke H_2.x angeschlossen werden.

Bevor die Motoren angeschlossen werden kann auch hier ein Test mit dem PWM-Tester durchgeführt werden. Somit besteht keine Gefahr ruckartig loslaufender Motoren.

Der Widerstand sollte für den Test der H-Brücken ca. $4.7\text{k}\Omega$ betragen.

Das Testergebnis sieht nun anders aus.
In der Ruhestellung leuchtet keine LED.

Helligkeitsdiagramm Nr. 2 der LEDs



Nach diesem Test sollte noch ein Test mit Autolampen erfolgen.

Bei der Verwendung einer 12V / 55 Watt Lampe ist es erforderlich den PWM-Wert zu beschränken. Dazu wird $\text{PWM_max} = 100$ gesetzt. Im Testmodus ist $\text{PWM_max}=100$ festgelegt.

Verwendet man 2 x 12V / 55 Watt Lampen in einer Reihenschaltung, so ist bei einer Versorgungsspannung von 24V die Beschränkung des PWM-Wertes nicht erforderlich.

6.1.2.2. Test im Modus 2

Der Modus 2 wurde entfernt, da die Hauptplatine, die neuen H-Brücken H_2.8+ und die Software so angepasst wurden, dass die BTS-H-Brücken in gleicher Weise wie H_2.8+ angeschlossen werden.

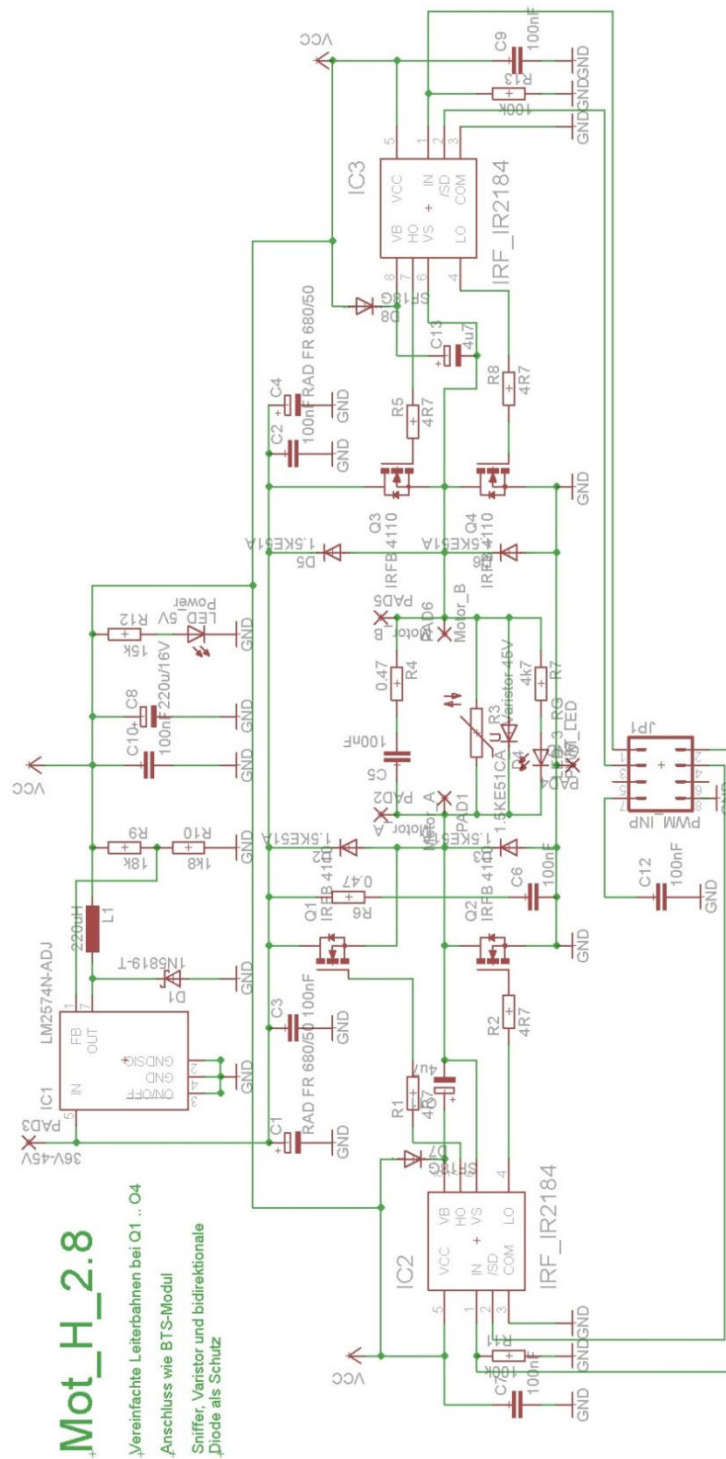
Auch die weitere Beschreibung wurde entfernt.

6.2. Schaltplan der H-Brücke

Die Ausführung der H-Brücke mit elektronischen Schaltern erlaubt die vollständige und sehr schnelle Steuerung mit Hilfe des Mikrocontrollers. Auch im Schaltplan sind die Elektronischen Schalter deutlich zu erkennen. Diese MOSFETs benötigen Treiberbausteine, hier mit der Bezeichnung IR2184.

Die abgebildete H-Brücke ist für den Betrieb mit 36 Volt geeignet. Dies wird einzig durch die verwendeten Überspannungsschutzdioden und die Spannungsfestigkeit der Kondensatoren und Spannungsregler bestimmt.

Bei der Verwendung einer Hauptplatine ab M_2.5 ist ein früher enthaltener Baustein CD4001 auf dieser Platine untergebracht.



6.3. Platine der H-Brücke

Die zu erwartenden hohen Ströme bis 50A machen eine Verstärkung der z.T. recht schwachen Leiterbahnen erforderlich.

Die MOSFETs vom Typ IRF4110 sind elektronische Schalter für Ströme bis über 100 Ampere. Der Einbau dieser Halbleiter kann stehend bzw. liegend erfolgen, wobei die Anordnung der Kühlelemente berücksichtigt werden sollte.

Ein liegender Einbau vereinfacht den Anschluss der elektrischen Verbindungen zur Batterie und zu den Motoren!

Flach liegende MOSFETs ermöglichen das Befestigen der gesamten Platine durch die Befestigungslöcher der MOSFETs auf einer Metall- bzw. Aluminiumplatte, die dann die Funktion eines Kühlkörpers übernimmt. Isolationspads, z.B. **IS 6018**, sorgen dabei für die elektrische Trennung der MOSFETs. Diese Position der MOSFETs ermöglicht einen problemlosen Zugang zu den Kabelklemmen.

Die verwendeten Kabelklemmen müssen eine hohe Strombelastbarkeit aufweisen. Die durchschnittlichen Ströme liegen im Bereich von 5A bis 15A.

Induzierte Spannungsspitzen aus der Motorwicklung können MOSFETs zerstören. Als Schutz dienen die auf der Platine befindlichen Schutzdioden. Ein zusätzlich an den Motorklemmen angeschlossener Varistor (ca.50V-60V), verdrehte Kabel zu den Motoren und Ferritringe aus alten PC-Netzteilen reduzieren die Zerstörungsgefahr.

6.3.1. H-Brücke Version 2.8

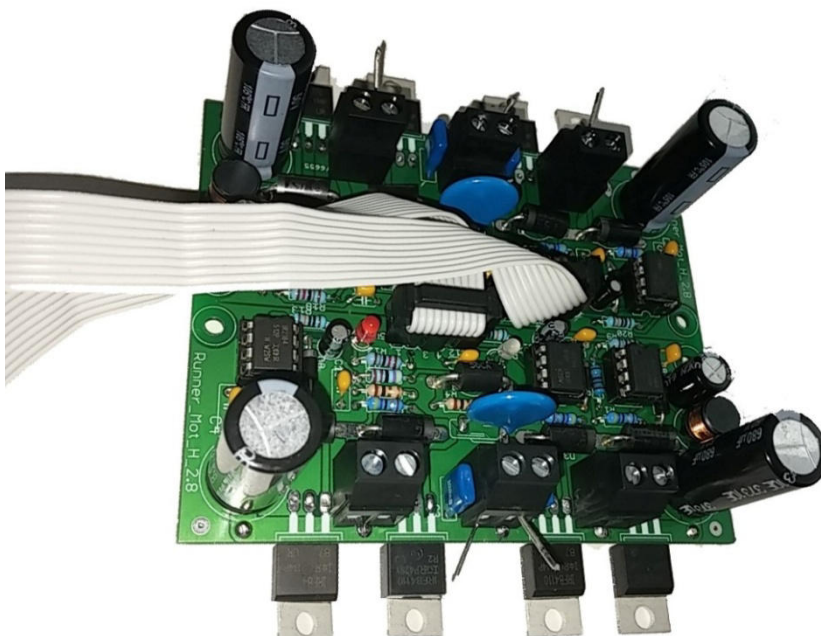
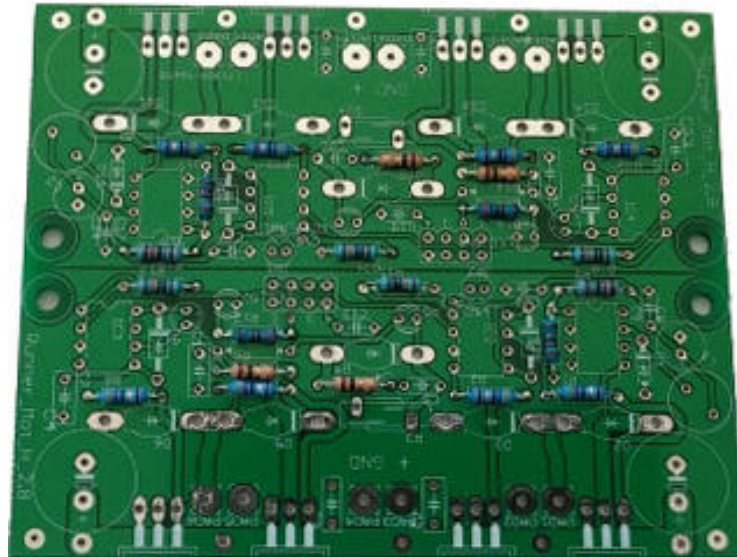
Diese überarbeitete Version der H-Brücke verwendet das grundsätzlich bewährte Prinzip der früheren H-Brücken.

Vereinzelte MOSFET-Schäden führten zu weiteren Schutzmaßnahmen gegen Spannungsspitzen durch einen Varistor und eine bidirektionale Schutzdiode am Motoranschluss.

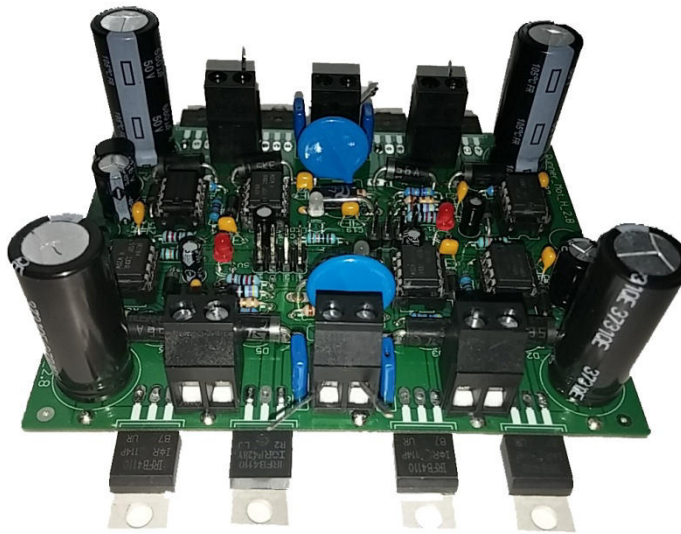
Neu ist der eingebaute PWM-Tester. Eine rot-grüne LED liegt mit einem Widerstand zwischen den Motoranschlüssen und leuchtet je nach Polung und Stärke des PWM-Signals in unterschiedlichen Farben verschieden hell.

Einige Veränderungen führen zum erleichterten Aufbau und einfacheren Verbindung zur Hauptplatine.

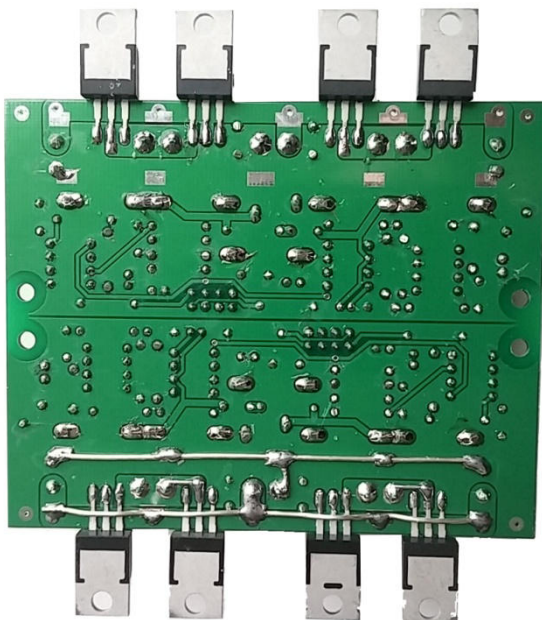
Das nachfolgende Bild zeigt eine teilweise bestückte Platine für die H-Brücke der Version 2.8.



Im Bild werden jeweils zwei nicht getrennte H-Brücken gezeigt. Das Flachbandkabel führt weiter zur Hauptplatine.



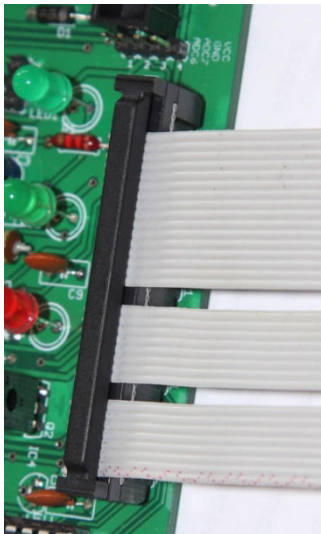
Die MOSFETs werden seitlich unter der Platine angelötet. In dieser Form sind die Lötflächen ausreichend groß und die Transistoren können direkt auf einen Kühlkörper oder das Aluminiumgehäuse des Runner aufgeschraubt werden. Isolierungen darf man



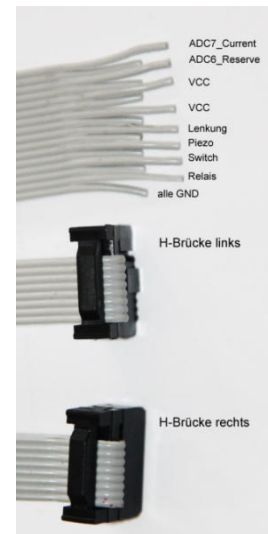
natürlich nicht vergessen (z.B. Reichelt **IS 6018**).

Wie auch bei früheren Versionen müssen wegen der hohen zu erwartenden Ströme Leiterbahnen verstärkt werden. Der hier verwendete versilberte Kupferdraht ist nun wesentlich einfacher zu verlöten und wird als Brücke über die Anschlussbeine der MOSFETs geführt.

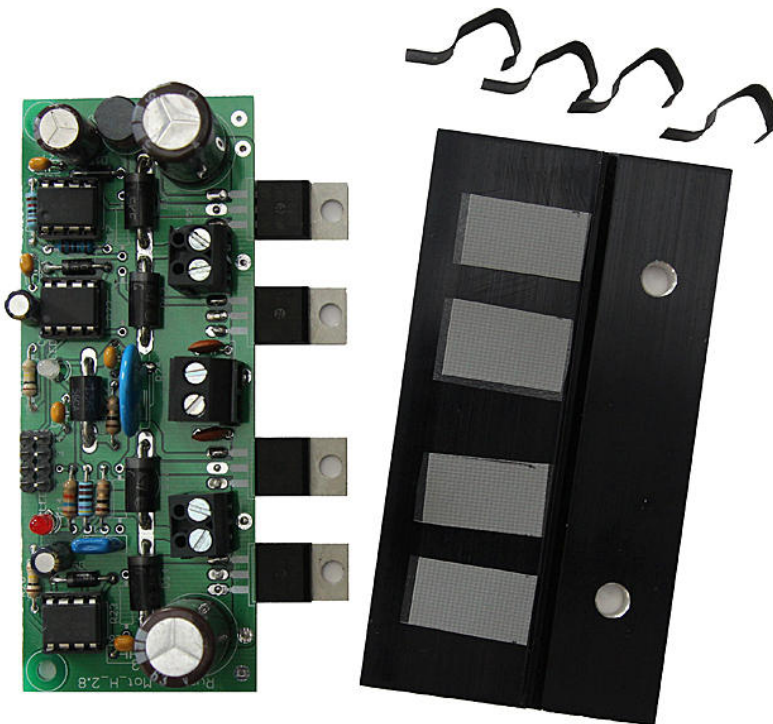
Die Verbindung zwischen Hauptplatine und H-Brücken wird durch ein Flachbandkabel vereinfacht.



Erforderlich ist ein mindestens 36-poliger Stecker (hier 40-pol.).
Überschüssige Drähte wurden entfernt.
Das ca. 28cm lange Flachbandkabel dient dem Anschluss der H-Brücken und anderer Komponenten.
Der Anschluss des Geschwindigkeitssensors erfolgt anderer Stelle.

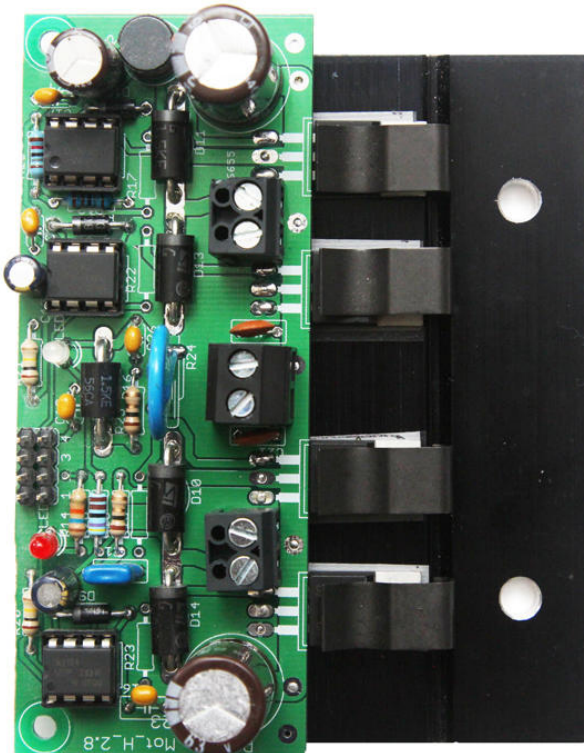


6.3.2. Kühlung der H-Brücke



Wird der H-Brücke eine hohe Leistung abverlangt, so wird eine Kühlung der MOSFETs erforderlich. Einfache Fahrversuche führen in der Regel nicht zu einer Erwärmung.

Eine Spezielle Aluminiumplatte, die als Kühlkörper so vorbereitet ist, hält die MOSFETs durch Klammern, die in eine spezielle Kerbe eingesetzt werden.



Zwischen die selbstklebenden Isolationspads und die Metallseite der TO220-Gehäuse der MOSFETs wird zusätzlich ein wenig **Wärmeleitpaste** aufgetragen. Schrauben halten den Kühlkörper sowie die Platine. Auf der linken Seite ist die Platine zusätzlich durch Abstandshalter zu unterstützen.



Eine Alternative ist im folgenden Bild zu sehen.

Preiswerter und mit weniger Aufwand verbunden ist die Befestigung der Platine am Gehäuse der Runners, sofern dieses aus Metall besteht.

Wie vorher, sind auch in diesem Fall Isolationspads (IS 6108) erforderlich.

Wärmeleitpaste verbessert die Wärmeleitfähigkeit. Die MOSFETs werden durch eine kleine Aluminiumplatte überdeckt, mit Schrauben gehalten und auf die Isolationspads gedrückt. Auf der anderen Seite wird die Platine unter Verwendung von

Distanzstücken ebenfalls

durch Schrauben im

Aluminiumblech des

Gehäuses gehalten.

Natürlich kann auf diese

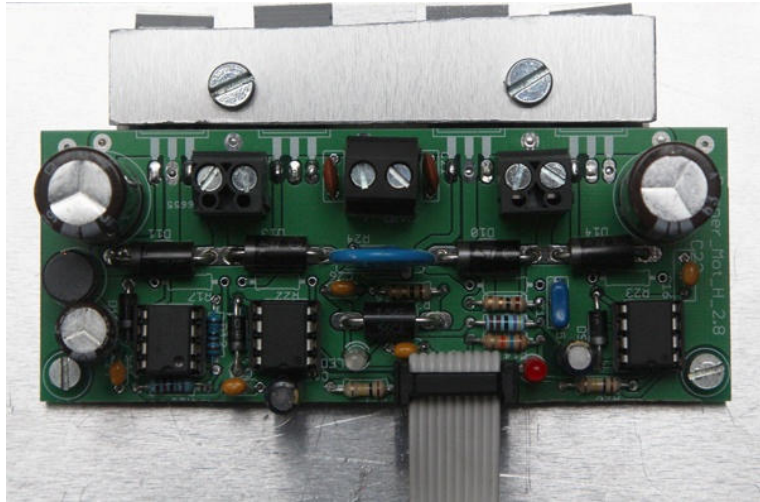
Weise auch ein Kühlblech

hergestellt werden,

welches in einem

Holzgehäuse eingebaut

wird.



Bei verschiedenen Tests zeigte der bisherige Aufbau ein Problem, welches zum Ausfall von zwei MOSFETs führte. Ein Transistortester offenbarte deutlich unterschiedliche

Gate-Kapazitäten, welche die Schaltzeiten

wesentlich beeinflussen.

Um die Schaltzeiten der MOSFETs zu verkürzen

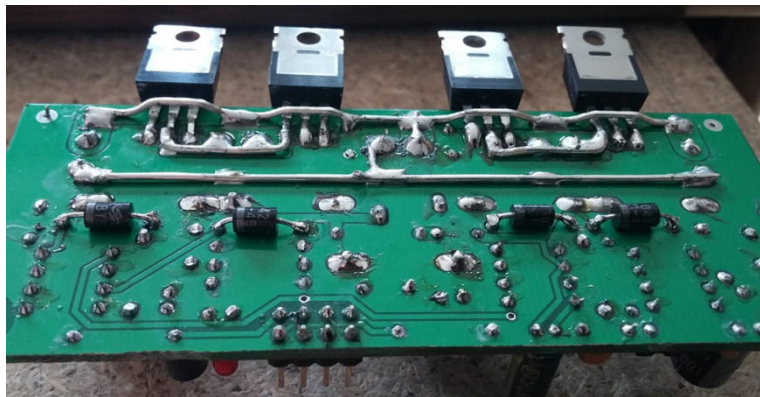
wurden auf der Unterseite

der Platine vier zusätzliche

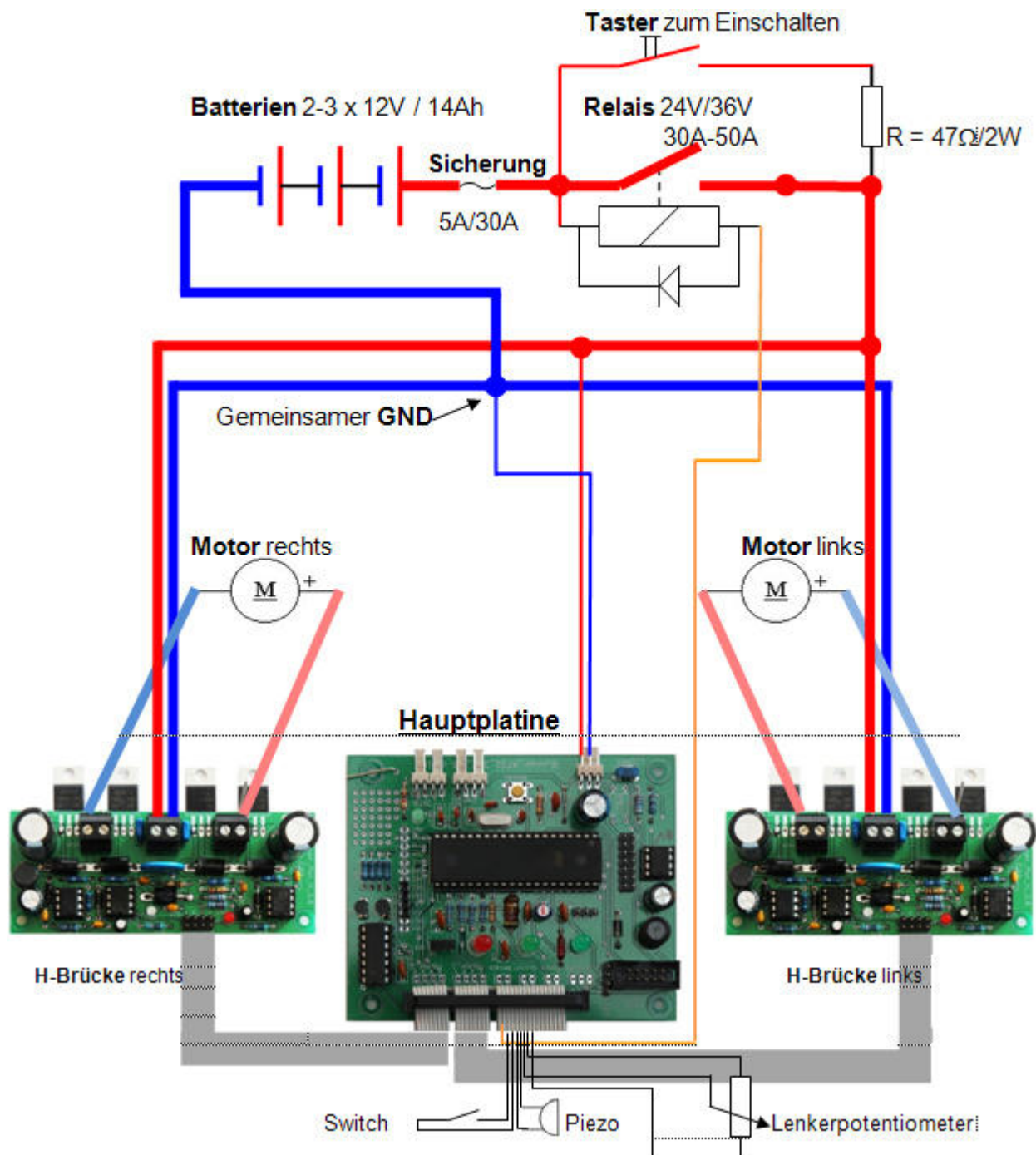
Dioden eingelötet. Diese

Dioden von Typ SB120

oder 1N5819 liegen parallel zu den Gate-Vorwiderständen.



7. Gesamtplan der Elektronik

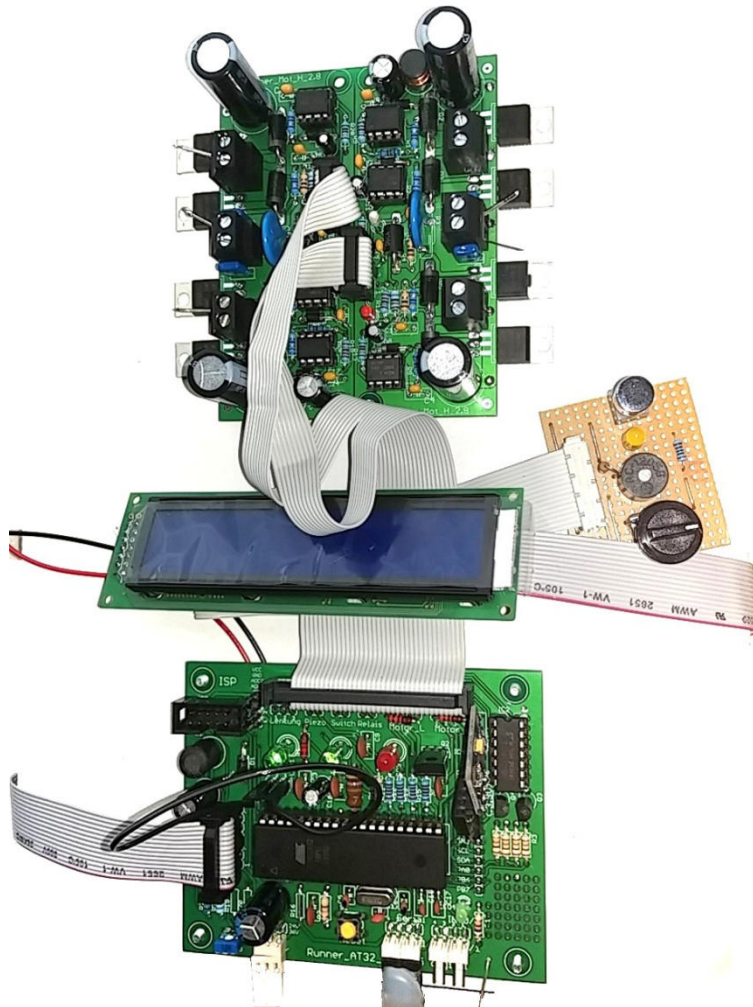


7.1. Gesamtplan Runner2+

Nach der Weiterentwicklung der Hauptplatine zur Version 2.5 / 2.6 und der H-Brücke zur Version 2.8, welche beide für eine Verbindung mit Hilfe eines Flachbandkabels vorbereitet sind, ergibt sich nun folgender Aufbau:

Das 36-polige Kabel führt, aufgeteilt in 10+10+16 Drähte zu den beiden Pinheads der H-Brücken und hier zu einer kleinen Hilfsplatine, die eine LED an Stelle eines Relais, einen Summer, ein (Lenker-) Potentiometer und einen (Fuß-)Schalter. Diese Elemente dienen dem Test der Komponenten, ohne dass diese in mein Fahrzeug eingebaut werden müssen. Das LC-Display wurde links an der Hauptplatine angeschlossen.

Der Anschluss der Versorgungsspannung erfolgt weiterhin entsprechend dem vorherigen Gesamtplan.



7.2. Hochstromverkabelung

Die Verkabelung der einzelnen Komponenten ist im Gesamtplan zu erkennen. Dick gezeichnete Kabel werden durch hohe Ströme belastet und sollten einen Querschnitt von mindestens $2,5\text{mm}^2$ besitzen. Alle anderen Verbindungen haben im Wesentlichen eine reine Signalfunktion, so dass ein Querschnitt von $0,14\text{mm}^2$ für einfache Signalleitungen genügt.

Wesentlich ist die Funktion und Verkabelung des Relais. Dieses Relais schaltet die Hauptstromversorgung, also auch Ströme bis max. 50A. Im geschlossenen Zustand oder auch bei induktiver Belastung ist dies kein Problem. Wird dieses Relais geschlossen, so werden auch die großen Kondensatoren, die sich auf den Platinen der H-Brücken befinden, schlagartig aufgeladen. Der Ladestrom erreicht beim Verbinden der Kondensatoren mit der Batterie weit über 100A, was zu einem Verschmelzen der Relaiskontakte führt und diese dauerhaft geschlossen bleiben. Im ungünstigen Fall wird das Relais dabei unbrauchbar und muss ausgetauscht werden.

Sollte der Fall eintreten, so hilft häufig ein festes Klopfen mit einem Schraubenziehergriff auf das Relaisgehäuse.

Um diesem Problem vorzubeugen, werden die Kondensatoren über einen Widerstand, z.B. **470 Ω** und einen kleinen Zusatztaster aufgeladen. Hat die ansteigende Versorgungsspannung an der Hauptplatine ca. 22V /33V erreicht, so wird das Relais kontrolliert geschlossen, ohne dass das beschriebene Problem entsteht.

Das Ausschalten der Elektronik erfolgt ebenfalls durch das gesteuerte Schalten des Relais durch die Software.

Die Ausführung des Relais kann sehr vielfältig sein. Im 24V-Betrieb kann auf Relais aus dem KFZ-Bereich (LKW) zurück gegriffen werden. Alternativ können auch die Spulen zweier 12V-Relais in Reihe geschaltet werden. Diese Variante ermöglicht weiter das Parallelschalten der Schaltkontakte, so dass eine höhere Strombelastbarkeit erreicht wird. Bei einer Versorgungsspannung von 36V konnten im Runner2 zwei Relais mit einer Betriebsspannung von je 18V in Reihe geschaltet werden. Die parallel geschalteten Kontakte, von denen jeder mit 30A belastbar ist, sind für den Betrieb mehr als ausreichend.

Hinweis:

Ist das passende Relais nicht zu finden, so kann z.B. im 24V-Betrieb ein 12V-Relais benutzt werden, indem dessen Spule mit 12V von der ersten Batterie betrieben wird. Diese Anschlussvariante ist auch beim 36V-Betrieb möglich, wobei 12V- und 24V-Relais verwendet werden können.

Wichtig: Freilaufdiode bei jeder Spulenwicklung nicht vergessen!

7.3. Anschlussklemmen und Kabel

Der oben genannte Querschnitt von 2,5mm² hat sich bewährt. Probleme zeigten sich beim mehrfachen Lösen und einstecken in die Anschlussklemmen.

Auch wenn die Anschlussklemmen für diesen Querschnitt ausgelegt sind, wird das Einstecken mit der Zeit schwieriger.

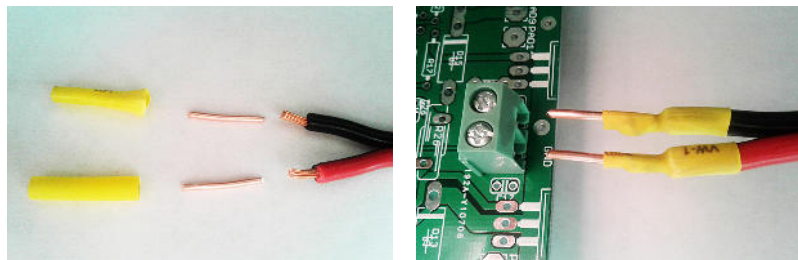
Einfache Kupferlitze spleißt auf, so dass die Öffnung der Anschlussklemme zu klein wird.

Wird das Kupferende verzinnt, so entsteht das gleiche Problem. Das mit Lötzinn getränkte Kabelende wird beim Zuschrauben der Klemme breit gedrückt und passt danach nicht mehr problemlos in diese Öffnung.

Weiter ist das verzinnte Leitungsende weich! Dies bedeutet, dass der anfangs starke Halt durch die Schraube der Anschlussklemme im Laufe der Zeit schwächer wird und sogar eine lose Verbindung mit Übergangswiderständen entstehen kann.

Ideal ist eine Verwendung von Aderendhülsen, die im Elektrohandwerk Verwendung finden. Diese und das Spezialwerkzeug stehen einem „Bastler“ jedoch selten zur Verfügung.

Bewährt hat sich das Anlöten von Kupferdrähten und das anschließende Isolieren mit Schrumpfschlauch.



Nun ist dauerhaft ein einfaches Einstecken des Drahtes in die Anschlussklemme möglich und ein sicherer elektrischer Kontakt bleibt erhalten.

7.4. Signalstromverkabelung

Ein Querschnitt von $0,14\text{mm}^2$ ist für einfache Signalleitungen ausreichend. Kabel mit Steckern, die an der Hauptplatine aufgesteckt werden, werden sinnvollerweise an den H-Brücken angelötet.

Die weitere Verkabelung zu Schaltern kann ebenfalls dünn gehalten werden.

Der Anschluss einer PC-Tastatur erfolgt am Steckanschluss der seriellen Schnittstelle. Die Lichtschranke für die Geschwindigkeitsmessung wird mit dem Stecker der Interruptanschlüsse verbunden.

7.5. Wichtig: Der gemeinsame Massepunkt.

Dies geht aus dem Gesamtplan der Elektronik nicht in der Deutlichkeit hervor, die zur Vermeidung von Störungen erforderlich ist.

Hohe Ströme verursachen Spannungsabfälle auf den Zuleitungen. Diese Potentialverschiebungen zwischen Signalausgängen der Hauptplatine und Eingängen der H-Brücken können zu Fehlfunktionen führen. Deshalb verbindet man alle GND-Zuleitungen der Platinen mit einem gemeinsamen GND-Punkt, wodurch das genannte Problem deutlich reduziert wird.

Sollte es möglich sein, alle Platinen auf einer Metallplatte unterzubringen, z.B. einem Aluminiumgehäuse des Fahrzeuges, so hat sich eine leitende Verbindung der Platinen an der Stellen der Verschraubungen mit dem Fahrzeuggehäuse als weitere Schutzmaßnahme bewährt.

8. Software

8.1. Programmkonzept

Ein Mikrocontroller kann viel, wenn man ihn lässt.

Wesentlich ist dabei wertvolle Rechenzeit nicht mit Wartezyklen zu füllen.

Mit diesem Grundgedanken wurden das gesamte Programm und insbesondere der Steuer- und Regelalgorithmus konzipiert.

Die wesentlichen Aufgaben sind dabei:

- a) Initialisierung aller Einheiten
- b) Ausgabe von Messwerte auf dem Display
- c) Balancealgorithmus und Funktionsüberwachungen.

Zu a) Dieser Punkt ist zeitunkritisch. In diesem Teil der Software werden Grundeinstellungen vorgenommen und elementare Anzeigen auf das Display gebracht. Hier sind Wartezyklen bewusst vorgesehen.

Zu b) Das Display zeigt im Sekundenrhythmus verschiedene Messwerte an. Diese Displayausgabe benötigt, je nach Umfang der Ausgabe, eine Zeit von bis zu 100ms. In der Software V7.05 wurden 84ms bei Verwendung des Atmega234PA mit 20MHz gemessen. An dieser Stelle erhält der Punkt c) eine wesentliche Bedeutung.

Zu c) Der Balancealgorithmus wird in zeitlichen Abständen von 10ms aufgerufen. Die in b) genannte Zeit führt dabei unweigerlich zu Unregelmäßigkeiten, die es auf alle Fälle zu vermeiden gilt. Ein Aufruf des Regelinterrupts benötigt 1,7ms (Atmega324PA/20MHz)

Eine Lösung wurde durch folgende Verschachtelung erreicht:

Der Balancealgorithmus wird durch einen zeitgesteuerten Interrupt alle 10ms mit höchster Priorität aufgerufen und ausgeführt.

Innerhalb dieses Algorithmus werden die Aufrufe in einer globalen Variablen mitgezählt.

Nach 100 Aufrufen wird eine Displayausgabe aktiviert, die jedoch durch nachfolgende Interrupts zur Ausführung des Balancealgorithmus unterbrochen werden darf. Diese Unterbrechungen verlängern die Displayausgabe um die Zeit der in die

Ausgabeperiode fallenden Interrupts, so dass die Gesamtausgabe z.B. durchaus erst nach 500ms abgeschlossen ist.

Diese Zeitverteilung kann durchaus kritisch werden, so dass bei der Programmkonzeption und Änderungen eine Zeitüberwachung mit dem Oszillographen durchgeführt wurde.

Im Rahmen der weiteren Entwicklung wurde ein Algorithmus entwickelt, der eine eigene Auslastungsmessung des Mikrocontrollers zulässt.

Die Gesamtauslastung lag bei der Softwareversion V7.36 bei ca. 25%.

Eine Fehlfunktion durch Überlastung des Controllers ist daher nicht zu befürchten.

8.2. Flussdiagramme

Diese Diagramme zeigen den Ablauf des Programmes.

Insbesondere lässt der Ablauf der Startprozedur erkennen, in welcher Weise Eingriffe des Benutzers durch eine Veränderung von Parametern oder die Verwendung des Testmodus herbei geführt werden können.

Die nachfolgenden Diagramme basieren auf der Softwareversion V7.36. Änderungen werden nach und nach ergänzt, das Grundprinzip ändert sich jedoch nicht.

Um einen Vergleich mit dem Programm zu ermöglichen, wurden in den Diagrammen Zeilennummern eingefügt.

Die drei Diagramme geben nicht jedes Detail wieder, nur wesentliche Schritte sind darin enthalten.

8.2.1. Startablauf des Runner

Zeile

326

361

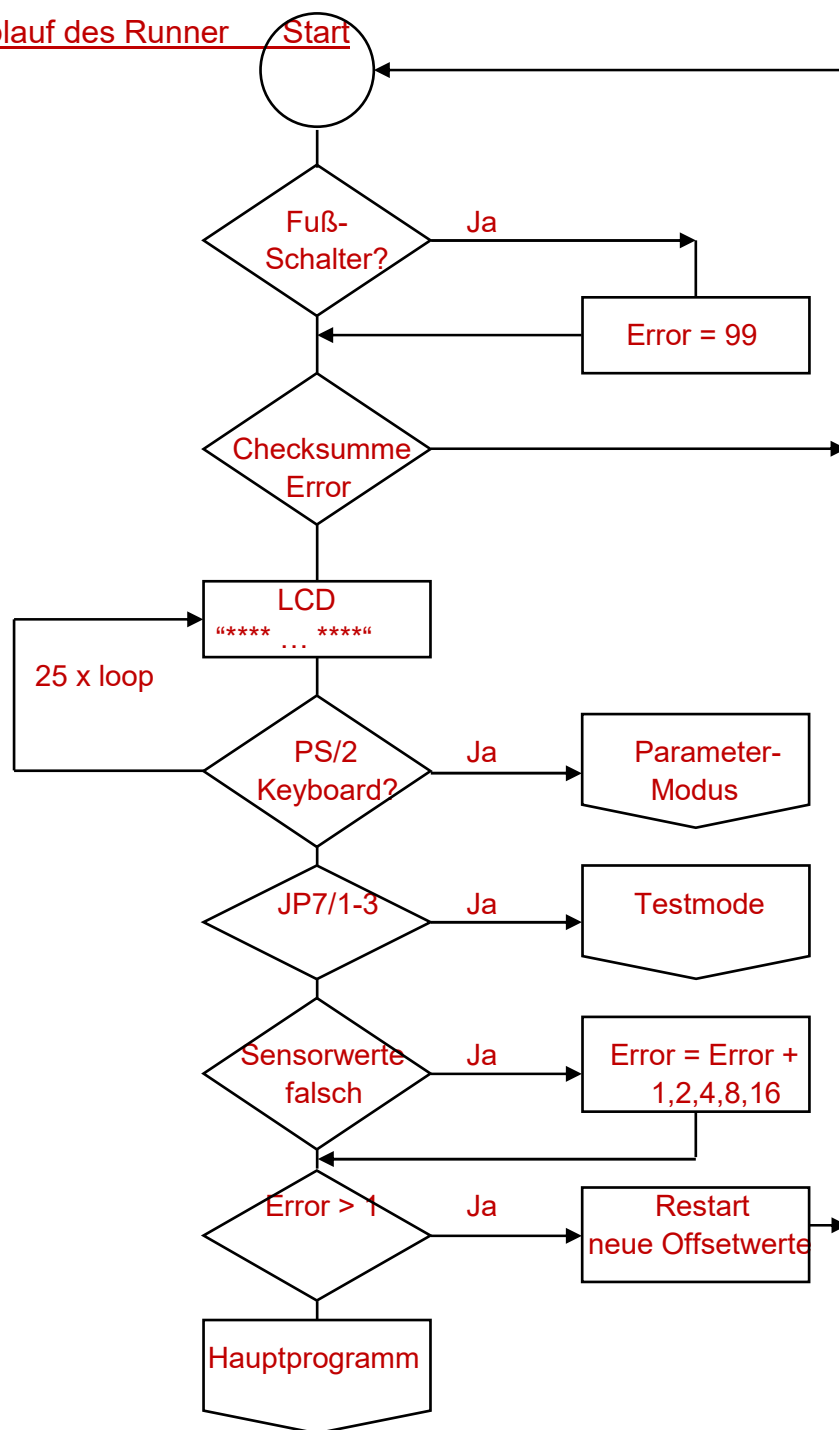
443

520

599

641

688



8.2.2. Mainloop des Runner

703

733

740

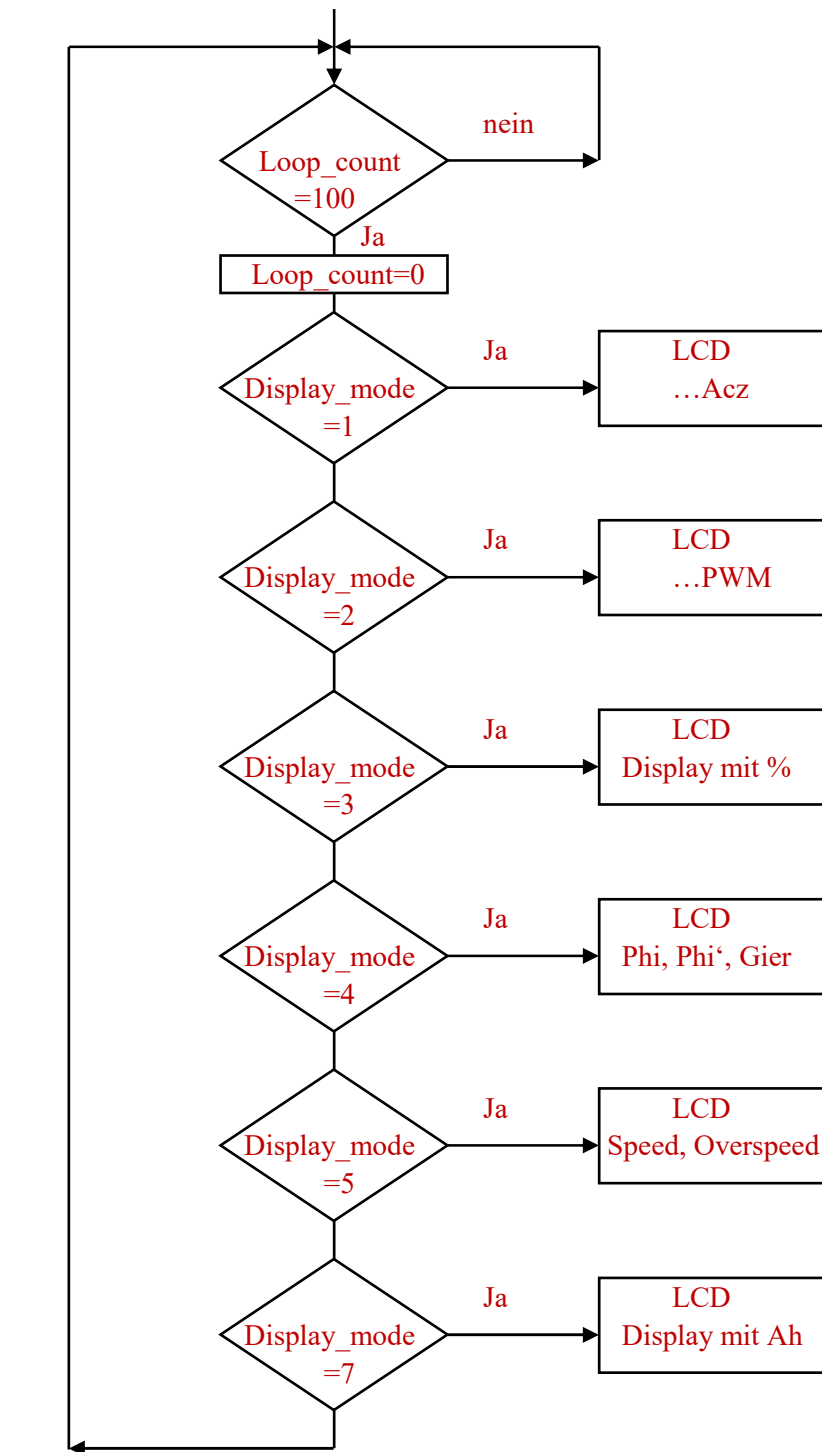
747

811

825

843

907



8.2.3. Balance-Interrupt

Strommessung

8.3. Regelroutine

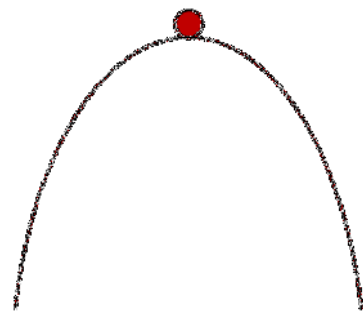
8.3.1. Der Balancealgorithmus

Ausgelöst durch einen Interrupt werden nach jeweils 10ms, d.h. 100 Mal pro Sekunde, die Zustandswerte des Fahrzeuges erfasst.

Die wesentlichen Werte sind dabei der Neigungswinkel φ des Fahrzeuges gegenüber der Ruhelage mit einer waagerechten Plattform, der mit Hilfe eines Beschleunigungssensors erfasst wird, sowie die Neigungsgeschwindigkeit $\rho = \frac{d\varphi}{dt}$, über die ein Gyroskop Auskunft gibt.

Überlegungen, die zur charakteristischen Regelgleichungen führen, sollen nun ohne aufwändigen Hintergrund dargestellt werden.

Es gibt sicher genau einen Punkt, in dem das Fahrzeug ohne äußere Einwirkung im Gleichgewicht ist. Dieser Punkt des labilen Gleichgewichtes lässt sich als oberer Punkt in nebenstehender Grafik darstellen. Eine geringe Abweichung von der Gleichgewichtslage führt zu einer mit zunehmenden Kräften angetriebenen Neigungsbewegung und zu einer ansteigenden Neigungsgeschwindigkeit. Letzteres lässt erkennen,



dass die Neigungsgeschwindigkeit ebenfalls in den Balance-Algorithmus mit einbezogen werden muss.

Die Regelung der Motoren erfolgt über ein PWM-Signal, welches in seiner Größe nun durch die oben genannten Messwerte bestimmt wird. Im Fall eines theoretisch negativen PWM-Wertes wird das Vorzeichen abgetrennt und zur Festlegung der Drehrichtung des jeweiligen Motors benutzt. Der Betrag des PWM-Wertes bleibt unverändert.

Im einem ersten Schritt ist nachvollziehbar, dass das PWM-Signal für den Antrieb der Motoren mit zunehmendem Neigungswinkel φ ebenfalls zunehmen muss, um ein Umkippen zu verhindern. Für den Fall, dass nur kleine Abweichungen von der Gleichgewichtslage auftreten, wovon man ausgehen kann, da der Fahrer stets fast senkrecht auf seinem Fahrzeug stehen wird, kann man vereinfachend annehmen, dass der PWM-Wert proportional zu φ sein muss.

Mit einem Proportionalitätsfaktor a folgt:

$$PWM = a \cdot \varphi \quad (1)$$

Der Einfluss der Neigungsgeschwindigkeit kann für kleine Abweichungen aus der Gleichgewichtslage ebenfalls als proportional zur Neigungsgeschwindigkeit ρ angenommen werden. Mit einem zweiten Proportionalitätsfaktor b ergibt sich:

$$PWM = a \cdot \varphi + b \cdot \rho \quad (2)$$

In dieser Form steht (2) für eine Beschleunigung des Fahrzeuges, die durch die Motoren aufgebracht werden muss.

Im Punkt des labilen Gleichgewichtes gilt: $\varphi = 0$ und $\rho = 0$.

Unabhängig von den Überlegungen zum Ausbalancieren bewegt sich das Fahrzeug mit einer Geschwindigkeit v . Um diese Geschwindigkeit zu erhalten ist ein PWM-Grundwert erforderlich, der wiederum von dieser Geschwindigkeit abhängt und zur Vereinfachung als proportional zur Geschwindigkeit v angenommen wird.

Zusammengefasst findet man nun:

$$PWM = a \cdot \varphi + b \cdot \rho + c \cdot v \quad (3)$$

Diese Gleichung (3) steht nun für ein ausbalanciertes Fahren mit konstanter Geschwindigkeit v .

Verursacht der Fahrer bewusst eine dauerhafte Neigung, so wird ein andauerndes Ausgleichen der Schiefelage provoziert, was eine Erhöhung der Geschwindigkeit zur Folge hat. Die vorherige Geschwindigkeit v wächst somit um das Produkt aus Beschleunigung und Beschleunigungszeit. Diese Beschleunigungszeit dt ist für jeden durchzuführenden Rechenschritt gleich der Zeit zwischen zwei Interruptereignissen, also $dt = 10ms$. Bei jedem Interrupt erfolgt vor der Ausführung der Gleichung (3) eine Neuberechnung der Geschwindigkeit v unter Berücksichtigung der Beschleunigung aus Gleichung (2) und der vergangenen Zeit dt in der Form:

$$v_{neu} = v_{alt} + (a \cdot \varphi + b \cdot \rho) \cdot dt \quad (4)$$

Mit $v_{alt} = v_{neu}$ wird nun die aktuelle Geschwindigkeit als Ausgangswert für die erneute Berechnung im nächsten 10ms-Interrupt gespeichert.

Dies ist nichts anderes als ein Integral der zum Winkel φ proportionalen Beschleunigung, wobei das Einbeziehen der Neigungsgeschwindigkeit ρ eine zusätzliche Stabilisierung bewirkt.

Mit der Geschwindigkeit v_{neu} wird nun das endgültige PWM-Signal berechnet, so dass sich aus Gleichung (3) ergibt:

$$PWM = a \cdot \varphi + b \cdot \rho + c \cdot v_{neu} \quad (5)$$

Aus dem Teil $c \cdot v_{neu}$ ergibt sich der neue PWM-Grundwert erhöht um die aktuell vorliegenden Beschleunigungswerte $(a \cdot \varphi + b \cdot \rho)$, welche durch eine temporäre Erhöhung des PWM-Wertes zu einem Auffangen der Kippbewegung führen.

Die algorithmische Umsetzung dieser Überlegungen wird in einem späteren Beispiel gezeigt.

8.3.2. Dynamische Parameteranpassung-Balanceregung

Bei höheren Geschwindigkeiten > 12km/h war die bisherige Balanceregung etwas „weich“. Um ein Kippen zu verhindern war ein stärkeres werdendes Neigen des Lenkers erforderlich, je größer die Geschwindigkeit war.

Der Grund ist physikalischer Natur. Um die Geschwindigkeit zum Erhalt des ausbalancierten Zustandes zu erhöhen, muss dem Fahrzeug zusätzlich kinetische Energie zugeführt werden. Die dazu notwendige Energie, die durch die Motoren aufgebracht werden muss, ist proportional zur aktuellen Geschwindigkeit, was bedeutet, dass die, den Motoren zugeführte Leistung ebenfalls proportional zur Geschwindigkeit geregelt werden muss.

Die in diesem Sinne durchgeführte Verbesserung der Software führte zu einem besseren Balanceverhalten bei höheren Geschwindigkeiten.

Auch hierfür sind zwei Parameter, Ref_speed_a_e und Ref_speed_b_e vorgesehen.

Sie wirken auf die Parameter A und B. Durch die Reduzierung des jeweils vorgegebenen Wertes 20 erfolgt eine stärkere werdende Balanceregung.

Sollten nach einer Verminderung der Parameter Ref_speed_a_e und Ref_speed_b_e während der Fahrt Vibrationen oder Schwingungen auftreten, so sind die eingestellten Werte zu klein.

Veränderungen sind stets mit Bedacht durchzuführen, sie können auch Unfälle zur Folge haben.

8.3.3. Die Lenkung

Der Lenkvorgang erfolgt durch die unterschiedliche Ansteuerung der beiden Räder. In einer Rechtskurve wird der PWM-Wert des linken Rades erhöht und der des rechten Rades reduziert. Die Linkskurve erfordert eine umgekehrte Veränderung der Werte. Die Stellung des Lenkers wird über eine Spannung am Abgriff eines Potentiometers erfasst. Der Analog-Digital-Converter (ADC) generiert einen entsprechenden digitalen Wert, der ebenfalls in der Interrupt-Routine verarbeitet wird.

8.3.4. Der Balance-Lenk-Algorithmus

Diese 100 Mal pro Sekunde durchgeführte Balanceregung besteht nun im Wesentlichen aus folgenden Zeilen:

Interrupt:

$$Phi^* = \varphi = \varphi(ACC_Sensor) - ACC_Ruhewert$$

/ für Phi* ist ggf. eine Skalierung

erforderlich

$$Rho^* = \rho = \rho(Gyro_Sensor) - Gyro_Ruhewert$$

/ für Rho* ist ggf. eine Skalierung

erforderlich

$$v = v + (a \cdot Phi + b \cdot Rho) \cdot dt$$

/ Berechnung der neuen Geschwindigkeit

$$PWM = c \cdot v + a \cdot Phi + b \cdot Rho$$

/ Berechnung des PWM-Wertes

$$PWM = PWM \cdot PWM_Faktor$$

/ Skalierung

$$Lenkwert^* = ADC(Lenk_Potentiometer) - Lenk_Ruhewert$$

/ für Lenkwert* ist ggf. eine Skalierung

erforderlich

$$PWM_{rechts} = PWM + Lenkwert$$

$$PWM_{links} = PWM - Lenkwert$$

/ Es folgt die Ausgabe von PWM_{rechts} und PWM_{links} an die entsprechende PWM-Erzeugung (z.B. Timer, IRQ, ...)

Return

Die Beschleunigungs- und Rotationswerte werden ab Softwareversion 7.xx aus dem Sensor MPU-6050 mit digitalem Ausgang und einer Datenübertragung nach dem I2C-Protokoll ausgelesen.

8.3.5. Mein Dank gilt Trevor Blackwell

Zu dieser Thematik schreibt **Trevor Blackwell**, ein Pionier im Bereich der selbstgebauten selbstbalancierenden Fahrzeuge, unter Berücksichtigung einer Geschwindigkeitsbegrenzung auf seiner Internetseite: <http://www.tlb.org/#scooter>.

Putting it all together

What takes many paragraphs to explain is surprisingly simple to code.

Here is the basic pseudocode of the balance algorithm, complete with the numbers which made my scooter feel stable and responsive.

Inputs

angle, angle_rate: the tilt angle of the scooter in radians and its derivative in radians/sec

steer_knob: the reading from the steering knob, between -1 and +1.

Balance

*balance_torque = 5.0 * (angle - rest_angle) + 0.4 * angle_rate*

Limit top speed by tilting back

overspeed = max(0, cur_speed - 0.5)

if (overspeed > 0) {

*overspeed_integral = min(0.4, overspeed_integral + min(0.2, overspeed+0.05) * dt) }*

*else {overspeed_integral = max(0, overspeed_integral - 0.04*dt)}*

*rest_angle = 0.4*overspeed + 0.7*overspeed_integral*

Steer. Decrease steering rate at high speed

*steer_cmd = 0.07/(0.3+abs(cur_speed)) * steer_knob*

Track current speed

*cur_speed = cur_speed + 1.2 * balance_torque * dt*

Differential steering

left_motor_pwm = balance_torque + cur_speed + steer_cmd

right_motor_pwm = balance_torque + cur_speed - steer_cmd

Outputs

left_motor_pwm and right_motor_pwm directly set the duty cycle of the pulse width modulator for the wheel controller, and range from -1 to +1 (+1 is 100% forward, -1 is 100% reverse.)

Diese Zeilen ließen mich erkennen, dass die Programmierung doch recht einfach ist. Mein Code umfasst zwischenzeitlich rund 2000 Zeilen und weicht wesentlich von Trevor Blackwells Zeilen ab.

8.4. Messgrößen und Parameter

Der Balance-Lenk-Algorithmus weist einige noch unbekannte Größen aus.

Alle Winkelgrößen werden im Bogenmaß gemessen, eine Winkereinheit, bei der 360° dem Wert $2 \cdot \pi$ entspricht.

Der aktuelle Sensor liefert im beschriebenen Fall digitale Ausgangswerte, deren Größe und Messbereich sich aus dem Datenblatt ergibt. Durch die bereits erwähnte Skalierung der Messwerte werden diese so aufbereitet, dass sie im Bogenmaß vorliegen.

Das Unterprogramm:

```
Read_6050(acx , Acy , Acz , Gyx , Gyy , Gyz , Sens_temp)
```

liest alle Messwerte aus den Registern des Sensors.

8.4.1. Der digitale Sensor MPU-6050

Nachdem die Produktion von Sensoren mit digitalem Ausgang, d.h. einer sogenannten I2C-Schnittstelle immer weiter zugenommen hat und Breakout-Boards mit analogen Sensoren bei den Anbietern fast ausverkauft sind, fiel die Entscheidung die Software für Nachbauer auf die Auswertung der neuen Sensoren umzustellen.

Für die ersten Experimente wurde der Sensor MPU-6050 gewählt, vor allem, weil er besonders preiswert zu beziehen ist.

Weiter enthält dieser Baustein einen Beschleunigungssensor und ein Gyroskope. Qualitätsmerkmale dieser Sensoren sind das Rauschen der Messwerte und die Nullpunkt- bzw. Temperaturstabilität, die sich als ausreichend erwiesen haben.

Bei der hohen Auflösung des Sensors sind Rauschwerte von bis zu 100 Bit im ersten Moment erschreckend. Bedenkt man die hohe Auflösung, so wird dies jedoch relativiert.

Des Weiteren kommt hier ein anderer Vorteil des Sensors zum Tragen. Der interne Tiefpass mit wählbaren Grenzfrequenzen führt zu einem erfreulich ruhigen Ausgangssignal, so dass die Verwendung des MPU-6050 für den Runner problemlos erscheint.

Die weiteren Filterungen durch bereits beschriebene Filter in der Software des Runners verbessern dieses Ergebnis.

Der Datenaustausch nach dem I2C-Protokoll kann in einer relativ kurzen Zeit durchgeführt werden.

8.4.2. Auswertung des Sensors MPU-6050

Dieser Sensor dient der Bestimmung des Neigungswinkels über die Messung des Beschleunigungsanteiles der Erdbeschleunigung in eine Vorzugsrichtung. Von den drei enthaltenen Beschleunigungssensoren ist im Prinzip einer der Sensoren ausreichend, um die Neigung des Fahrzeuges zu erfassen.

Die Sensoren mit der Bezeichnung X , Y und Z sind rechtwinklig zueinander angeordnet. Die Ausrichtung Z-Achse erfolgt dabei **parallel** zur Fahrzeugachse.

Die Praxis zeigt bei der Winkelbestimmung mit Hilfe eines Sensors, z.B. X, einen Nachteil. Fährt das Fahrzeug durch eine Vertiefung bzw. über eine Erhöhung, so beeinflusst die dabei auftretende vertikale Beschleunigung auch den Messwert. Die Folge ist eine positive oder negative Beschleunigung des Fahrzeuges, da die Messwertänderung als Neigung interpretiert wird.

Eine Winkelberechnung unter Einbeziehung des Y-Sensors führt zu einer fast vollständigen Beseitigung des Problems.

Wählt man als Ruhestellung des Sensors eine um 45° geneigte Position gegenüber der X- bzw. Y-Achse, so befinden sich die beiden inneren Sensoren X und Y bezüglich einer vertikalen Bezugsrichtung in einer symmetrischen Position. Jede Form der vertikalen Bewegung wirkt sich auf beide Messwerte gleich aus.

Die erhaltenen Messwerte der X- und Y- Sensoren sind nun proportional zu $\sin(\varphi)$ und $\cos(\varphi)$ Weiter gilt:

$$\tan(\varphi) = \frac{\sin(\varphi)}{\cos(\varphi)}$$

Im Ergebnis: $\tan(\varphi) = \frac{X_{out}^*}{Y_{out}^*}$ bzw. , $\varphi = \arctan\left(\frac{X_{out}^*}{Y_{out}^*}\right)$

wobei im Zähler und Nenner die um den Ruhewert (Offset) reduzierten Messwerte des Sensors Verwendung finden.

In dieser Schreibweise wird der eigentliche Vorteil erneut deutlich. Jede vertikale Erschütterung hat auf den Zähler und Nenner des Bruches die gleiche Auswirkung, so dass der Quotient konstant bleibt und der berechnete Winkel keiner Veränderung unterliegt. Der Vollständigkeit halber sei erwähnt, dass die nur im Idealfall der senkrechten Stellung des Fahrzeuges gilt. Sollte das Fahrzeug im Fall einer vertikalen

Beschleunigung eine geringe Neigung besitzen, so tritt theoretisch der eingangs erwähnte Effekt in minimaler Stärke auf, ohne im Geringsten spürbar zu werden. Das Ergebnis für den Neigungswinkel φ liegt nach der Verwendung des $\arctan()$ im Bogenmaß vor, somit ist keine Skalierung erforderlich.

8.4.3. Auswertung des Gyroskops im MPU-6050

Die Neigungsgeschwindigkeiten, die vom Gyroskop des integrierten Sensors erfasst werden, beziehen sich auf die Rotation um die bereits genannte X-, Y- und Z-Achse. Die Anordnung der Z-Achse des Sensors parallel zur Radachse des Runners hat zur Folge, dass dieser Messwert des Sensors für die Balanceberechnung Verwendung finden muss. Nach der Berücksichtigung eines Offsetwertes und einer Skalierung liegt die Rotationsgeschwindigkeit in der Einheit rad/s vor.

8.4.4. Parameter

Die im Kapitel Balancealgorithmus erläuterten Zusammenhänge können nun mit Hilfe dieser Messwerte umgesetzt werden.

Die dort genannten Parameter a , b und c bestimmen nun, in welchem Maß der Neigungswinkel Φ und die Neigungsgeschwindigkeit ρ bei der Berechnung des PWM-Wertes berücksichtigt werden.

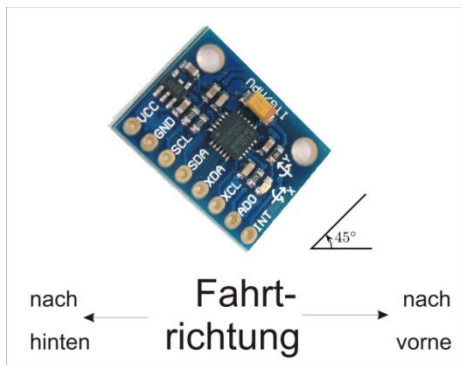
Dabei haben sich $a = 0,2$, $b = 0,2$ und $c = 15$ für anfängliche Versuche bewährt.

Das entstandene Ergebnis der Beschleunigungsberechnung liegt in einem Wertebereich von -1 bis +1 und wird für zur Anpassung an die notwendigen PWM-Werte mit einem weiteren Faktor 350 multipliziert.

Bei der Verwendung anderer PWM-Auflösung, z.B. von 8 Bit mit 256 möglichen Werten ist der Faktor 175 sinnvoll. Die Anpassung an eine 10 Bit Auflösung erfolgt durch einen Faktor 700.

Verschiedene PWM-Modi werden an anderer Stelle beschrieben.

8.4.5. Anordnung des Sensors MPU-6050



Die kleine Platine muss in der Stellung eingebaut werden, wie sie hier zu sehen ist.

Dabei ist auf eine Neigung von 45° zu achten.

Geringe Abweichungen sind nicht problematisch.

Das rechts erkennbare Loch **muss** in Fahrtrichtung zeigen. Die kleine Markierung auf dem IC zeigt dabei nach hinten! Die korrekte

Position ist, Zitat: **„extrem WICHTIG“**.

Die Sensorplatine kann auf die Hauptplatine aufgesteckt werden, wenn die Art der Steckverbindung eine Schrägstellung von 45° ermöglicht und die Hauptplatine eine geeignete Position besitzt.

Einfacher ist es, den Sensor über ein 4-poliges Kabel mit einem angelöteten Stecker, den man aus den beiliegenden Steckverbindern herstellen kann, mit der Platine zu verbinden.

Dieser kleine Stecker wird auf der Hauptplatine eingesteckt.

Bei neueren Platinen sind die Anschlüsse auf der Platine entsprechend benannt.

8.5. Digitale Filter

Beschrieben wird ein alternativer Filter zum Kalman-Filter für selbstbalancierende Fahrzeuge (SBF)

Digitale Filter ermöglichen die Verbesserung von Messwerten, sofern diese z.B. Rauschanteile oder Störungen enthalten. Diese Anteile liegen in der Regel in einem Frequenzbereich, der über dem Frequenzbereich des eigentlichen zu messenden Signals liegt. Im analogen Fall würde man auf einen Tief pass zurückgreifen, der sich auch digital realisieren lässt.

In diesem Zusammenhang kommt auch dem Kalman-Filter eine wesentliche Bedeutung zu.

Hier sollen jedoch nur Alternativen zu dem schwer durchschaubaren Algorithmus von Kalman besprochen werden. Nachfolgend wird die Funktionsweise einfacher Filter erläutert ergänzt um Erfahrungen in Zusammenhang mit dem Selbstbau eines selbstbalancierenden Fahrzeugs.

8.5.1. Gegen das Rauschen

Das Rauschen, als Grundübel in der Messtechnik, führt bei präzisen Messungen zu dem Problem, dass das nachfolgende System auf scheinbare Änderungen des Messwertes reagiert, obwohl im System keine Zustandsänderung vorliegt.

Die relevanten Sensoren sind bauartbedingt als analoge sowie als digitale Sensoren mit einem hohen Rauschanteil behaftet, so dass diesem Problem schon früh entgegengewirkt werden sollte.

Die analoge Elektronik greift in diesem Fall in der Regel auf einen Tiefpass zurück, welcher sich durch eine R-C-Kombination realisieren lässt.

Dieser Tiefpass entfällt bei der Verwendung des digitalen Sensors MPU-6050 ab der Software V7.xx.

In dem verwendeten Sensor ist je ein einstellbarer Tiefpass für die Messwerte des Beschleunigungs- und Gyrosensors enthalten. Die jeweilige Grenzfrequenz wird bei der Initialisierung des Sensors beim Programmstart festgelegt.

8.5.2. Ein einfacher digitaler Filter – Der Simple Filter

Anwendung eines digitalen Tiefpasses (1. Ordnung) als teilweiser Ersatz für ein Kalman-Filter. Die Messwerte des Gyrosensors bleiben dabei noch unberücksichtigt.

Einen einfachen digitalen Filter erhält man, wenn vor der Weiterverarbeitung eines Messwertes dieser mit einem oder mehreren vorherigen Werten zu einem mittleren Wert verrechnet. Ein Filter dieser Art stellt einen Tiefpass dar. Er wird nachfolgend genauer beschrieben.

a) Der Grundgedanke

Bsp.: y neuer Messwert (1)
 y_{alt} vorheriger Messwert (Mittelwert)

```

Do
:
y = ACC_Wert            ,ACC-Wert auslesen
y = 0,9*  $y_{alt}$  + 0,1*y    'neuen gewichteten Mittelwert bilden
 $y_{alt} = y$                 'Y-Wert für den nächsten Durchgang merken.
:
Loop
  
```

b) Verallgemeinerung mit einer variablen Dämpfungszahl k:

$a = 1/k$ 'k = Dämpfungszahl, $k > 1$ (2)

```

Do
:
y = ...                    'aktueller Messwert
y = (1-a)* $y_{alt}$  + a*y    'neuen gewichteten Mittelwert bilden
 $y_{alt} = y$                 'Y-Wert für den nächsten Durchgang merken.
:
Loop
  
```

c) Bestimmung der Zeitkonstanten/Grenzfrequenz dieses Tiefpassfilters:

Ein Regelalgorithmus wird üblicherweise in bestimmten Zeitintervallen dt ausgeführt. Dieses Zeitintervall dt bestimmt auch die Reaktion des Ausgangssignals auf eine Veränderung des Eingangssignals. Unter Berücksichtigung einer geeigneten

Anfangsbedingung lässt sich die Grenzfrequenz eines solchen digitalen Filters bestimmen.

Besitzt y seit längerer Zeit den Wert 1, so ist die Zeit gefragt, nach der beim Wechsel des aktuellen Messwertes von 1 auf 0 der gefilterte Ausgangswert z.B. um 3db abgesunken ist.

Dabei ist jeder nachfolgende Messwert $y = 0$ und es ergibt sich: $y = (1-a)^n * y_{\text{alt}}$.

Nach n Durchläufen ergibt sich: $(1-a)^n = 1 / 1,41$ (bei -3db)

$$\text{und } n = \frac{\log(1/1,41)}{\log(1-a)} \quad (3)$$

n Durchläufe entsprechen einer Zeit $T = n \cdot dt$.

Beispiel 1: mit $dt = 0,01s = 10ms$, $k = 10 \rightarrow a = 0,1$

Nach Gleichung (3) sind $n = \frac{\log(1/1,41)}{\log(1-0,1)} = 3,4 \rightarrow n=3,4$ Durchläufe erforderlich.

Daraus ergibt sich eine durchschnittliche Zeit $T = 34ms$ und eine 3db-Grenzfrequenz $f_g = 1/T = 30Hz$.

Für den Fall eines SBF sind Alternativen zur 3db-Grenze denkbar. Erfahrungsgemäß sollten Rauscheffekte deutlich stärker unterdrückt werden, so dass sich eine Amplitudenreduzierung von 6db anbietet.

Analog zu (3) ergibt sich: $(1-a)^n = 0,5$ (bei -6db)

$$\text{und } n = \frac{\log(0,5)}{\log(1-a)} \quad (4)$$

Nach Gleichung (3) sind $n = \frac{\log(0,5)}{\log(1-0,1)} = 6,58 \rightarrow n=6,6$ Durchläufe erforderlich.

Es folgen eine Zeit $T = 66ms$ und eine 6db-Grenzfrequenz $f_g = 1/T = 15Hz$.

Beispiel 2: mit $dt = 0,01s = 10ms$, $k = 20 \rightarrow a = 0,05$

Nach Gleichung (3) sind $n = \frac{\log(1/1,41)}{\log(1-0,05)} = 6,75 \rightarrow n=7$ Durchläufe erforderlich.

Daraus ergibt sich eine Zeit $T = 70ms$ und eine 3db-Grenzfrequenz $f_g = 1/T = 15Hz$.
Für den Fall eines SBF sind Alternativen zur 3db-Grenze denkbar. Erfahrungsgemäß sollten Rauscheffekte deutlich stärker unterdrückt werden, so dass sich eine Amplitudenreduzierung von 6db sinnvoll erscheint.

Analog zu (3) ergibt sich: $(1-a)^n = 0,5$ (bei -6db)

$$\text{und } n = \frac{\log(0,5)}{\log(1-a)} \quad (4)$$

Nach Gleichung (3) sind $n = \frac{\log(1/2)}{\log(1-0,05)} = 13,5 \rightarrow n=14$ Durchläufe erforderlich.

Es folgen eine Zeit $T = 140ms$ und eine 6db-Grenzfrequenz $f_g = 1/T = 7Hz$.

Eine Möglichkeit diese Grenzfrequenz zu verändern ergibt sich durch die Veränderung der Dämpfungszahl k .

Beispiel 3: mit $dt = 0,01s = 10ms$, $k = 50 \rightarrow a = 0,02$

Nach Gleichung (3) sind $n = \frac{\log(1/1,41)}{\log(1-0,02)} = 17 \rightarrow n=17$ Durchläufe erforderlich.

Daraus ergibt sich eine Zeit $T = 170ms$ und eine 3db-Grenzfrequenz $f_g = 1/T = 5,9Hz$.

Bei einer Amplitudenreduzierung von 6db, d.h. auf 50%, so erhält man:

$$(1-a)^n = 1/2 \text{ (bei -6db) und } n = \frac{\log(1/2)}{\log(1-0,02)} = 34 \rightarrow n=34 \text{ Durchläufe erforderlich.}$$

Daraus ergibt sich eine Zeit $T = 340ms$ und eine 6db-Grenzfrequenz $f_g = 1/T = 3Hz$.

Wählt man bei gleichem $k = 50$ als Maß für die Reduzierung der Amplitude den Faktor

$1/e = 0,3678$, so findet man $n = \frac{\log(1/e)}{\log(1-0,02)} = 49$ und somit eine Zeit von 490ms, was einer Grenzfrequenz von 2Hz entspricht.

Ergebnis:

In der Praxis wurden mit $dt = 0.01s$ die Dämpfungszahlen $k = 20$ und $k = 50$ im Vergleich mit einem Kalman-Filter mit der Dämpfungszahl $R=40$ erprobt. Beide digitalen Filter dieser Art wurden dabei auf das nach einem Algorithmus erhaltene Ergebnis für die Variable *Balance_torque* angewendet. Mit $k=10$ reagiert das SBF ohne erkennbare Verzögerung. Eine zusätzlich zum analogen Filter entstandene Rauschunterdrückung hat keine negativen Auswirkungen auf die Reaktionsgeschwindigkeit.

Mit $k=20$ erhält man bei wesentlich kürzerer Rechenzeit ein fast vergleichbares Ergebnis. Das Balance-Verhalten fühlt sich geringfügig verzögert an. Die zur Beschleunigung notwendige Neigung wird dabei als etwas größer empfunden.

Mit $k=50$ ergibt sich ein vollständig anderes Ergebnis. Die Balance-Regelung reagiert so langsam, dass weite Neigungen erforderlich sind, bis die angesteuerten Motoren reagieren. Dieses Verhalten führt zu einer Pendelbewegung des Fahrzeuges mit zunehmender Amplitude mit einer Frequenz von gut 1Hz und dem dann unvermeidlichen Sturz.

8.5.3. Doppelt gefiltert – ein Komplementärfilter

Google findet eine anschauliche Veröffentlichung mit den Suchbegriffen „shane colton balance filter“.

Bei diesem Filter werden Messwerte des Accelerometer und des Gyroskops zusammengefasst. Dabei wird, wie auch beim **Simple_Filter**, durch eine gewichtete Mittelwertbildung eine Tiefpassfilterung durchgeführt und Rauscheffekte unterdrückt. Bedauerlicherweise wird das System dabei etwas träge und reagiert langsamer auf Veränderungen der Neigung.

Das Integral über die Winkelgeschwindigkeit, die Messwerte des Gyroskops, erlaubt ebenfalls eine Berechnung des aktuellen Winkels. Eine systembedingte Drift des Nullpunktes eines jeden Gyroskops führt zu einer deutlichen Verfälschung des Ergebnisses, so dass der relativ träge Messwert des gefilterten Accelerometerwertes durch die über einen Hochpass gewonnenen Änderungen des Gyroskope-Integrals ergänzt werden können. Die Anwendung eines Hochpasses, der auch mit der Software realisiert wird, unterdrückt diese Drift des Gyroskops vollständig.

Eine Umsetzung dieser theoretischen Überlegungen liefert ein Algorithmus, der als Komplementärfilter bezeichnet wird.

Der gewonnene Winkel φ besitzt nach dieser Filterung die Eigenschaft, fast rauschfrei und trotzdem nicht träge zu sein.

Aus den genannten theoretischen Überlegungen ergibt sich für den gefilterten Winkel:

$$\Phi = a * \Phi_{alt} + (1 - a) * \Phi_{Acc} + a * dt * Gyro$$

Die Messwerte des Accelerometers und des Gyrosensors und müssen in der Winkereinheit „rad“ vorliegen. Eine korrekte Umrechnung der Sensorwerte unter Berücksichtigung der Offsetwerte der Sensoren unumgänglich. Offsetwerte werden automatisch oder auf Tastendruck durch die Software bestimmt.

Deutlich zu erkennen ist der enthaltene Tiefpass, der bereits im Simple-Filter Verwendung findet und den verrauschten Messwert des Neigungswinkels glättet.

Der Faktor a berechnet sich in der Form: $a = \frac{\tau}{1 + \tau}$, aus $\tau = \frac{1}{f_{grenz}}$, der Filterkonstanten (dem Kehrwert der Grenzfrequenz) und $dt = 0,01$, die Interruptzeit des Atmega32.

Nebenstehende Tabelle zeigt den Zusammenhang zwischen dem Faktor a und der Grenzfrequenz des Filters. Der Faktor a kann ab Software-Version V7.26 als Parameter unter Verwendung der der Taste F5 eingetragen werden.

f_{grenz}/Hz	a
20	0,83
10	0,91
5	0,95
2	0,98

Die Verwendung dieses digitalen Filters ab der V7.26 führt zu einem weiter verbesserten und ausgeglichenerem Fahrgefühl.

In der Praxis wurden Werte $0,90 < a < 0,99$ erprobt.

Hinweis:

In der oben angegebenen Schreibweise findet man $a * \Phi_{alt}$ an der Stelle von man $(1 - a) * y_{alt}$ mit gleicher Bedeutung für Simple-Filter. Die relevanten Werte sind entsprechend zu tauschen.

8.5.4. Kalman-Filter „mal ganz einfach“.

Der digitale Filter von Rudolf E. Kálmán dient der Beseitigung von Störungen, die sich in einer Folge von Messwerten befinden.

Weiter berücksichtigt dieser digitale Filter die aktuelle Veränderung eines Messwertes und liefert unter Verwendung des Gyro-Wertes letztendlich ein vorausschauendes Ergebnis.

Die zugehörige Theorie ist sehr komplex, so dass ein Umsetzen des Algorithmus durchaus Schwierigkeiten bereiten kann.

Der Kalman-Filter stellt weiter einen Tiefpass dar, der durch Parameter beeinflusst wird. Diese Parameter, welche auch bezüglich der Rauscheigenschaften des verwendeten Sensors auszuwählen sind, sind statistische Werte und somit schwierig zu bestimmen.

Versuche mit einem Filter dieser Art haben wegen einer ungeeigneten Parameterwahl kein überzeugendes Ergebnis gezeigt.

Auf eine weitere Beschreibung und die zugehörigen Programmzeilen wird aus diesem Grund verzichtet.

So einfach war es dann doch nicht.

8.6. Geschwindigkeitsabhängige Lenkung

8.6.1. Zusammenhänge

Bei der Kurvenfahrt zeigt sich, dass die Reaktion des Fahrzeuges auf eine seitliche Neigung des Lenkers deutlich von der Geschwindigkeit v abhängt.

Geht man davon aus, dass der Abstand der Räder $D = 2 \cdot d$ bekannt ist und der seitliche Neigungswinkel α des Lenkers ggf. mit Hilfe eines Potentiometers gemessen werden kann, so muss zusätzlich eine zuverlässige Angabe über die Geschwindigkeit vorliegen.

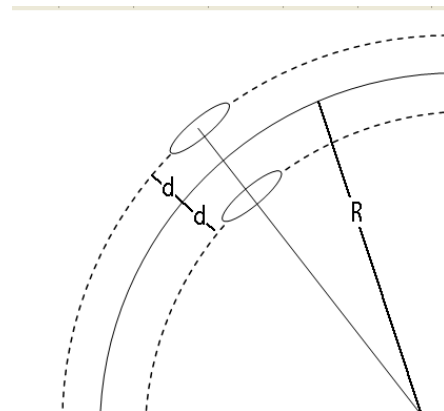
Der Abstand von Radmitte bis Radmitte beträgt häufig rund 0,6m so dass sich daraus $d=0,3m$ ergibt. Hierbei und bei allen anderen Größen werden elementare Einheiten verwendet.

Der Kurvenradius R ergibt sich aus den nachfolgenden Überlegungen und wird ebenfalls in der Einheit Meter (**m**) angegeben.

Die Bestimmung der Geschwindigkeit ist ggf. etwas problematisch. Die beste Methode ist zweifelsfrei das direkte Messen von Radumdrehungen pro Zeiteinheit, die Zeitmessung für eine Umdrehung oder die Zeitmessung zwischen zwei Zähnen eines Zahnrades im Antriebsbereich. Die Geschwindigkeit v muss mit der Einheit (**m/s**) vorliegen.

Eine Abschätzung der Geschwindigkeit aus der relativen Größe des PWM-Signals kann nur ein grober Ersatz für die tatsächliche Geschwindigkeit sein.

Nachdem die maßgeblichen Größen α , d und v vorliegen ist zu beachten, dass der stehende Fahrer in der Kurvenfahrt eine gewisse seitliche Neigung einnehmen muss, um nicht von der Zentrifugalkraft nach außen gedrückt zu werden. Der Lenker wird bei der entsprechenden seitlichen Neigung mitgeführt, so dass letztendlich Fahrer und Lenker die gleiche Neigung aufweisen.



Berücksichtigt man die dabei auftretenden Kräfte, die Erdanziehungskraft $F_g = mg$, $g=9,81\text{m/s}^2$ und die Zentrifugalkraft $F_z = mv^2/R$ so ergibt sich unter Berücksichtigung der zugehörigen Richtungen

$$\tan(\alpha) = mv^2/(mR \cdot g) \quad (1)$$

Die Gleichung (1) zeigt den Zusammenhang zwischen den o.g. Größen und erlaubt den zu jeder beliebigen Lenkerneigung α gehörenden Bahnradius R .

Für R ergibt sich aus (1) $R = v^2/(\tan(\alpha) \cdot g)$ (2)

Die Gleichungen (1) und (2) dürfen bei Geradeausfahrt wegen $\alpha=0$ nicht angewendet werden.

Nachfolgende Realisierung ist somit nur für $\alpha \neq 0$ anzuwenden.

8.6.2. Realisierung der Kurvenfahrt

Wie kann man nun erreichen, dass das Fahrzeug eine Kurve mit dem zutreffenden Radius fährt. Nur in diesem Fall fühlt sich der Fahrer auf seinem Fahrzeug wohl, weil der resultierende Kraftvektor vom Kopf exakt zwischen die beiden Füße zeigt und diese Situation als **senkrecht stehen** empfunden wird.

Dieses Problem ist relativ einfach zu lösen.

Für den Bahnradius des äußeren Rades gilt: $R_1 = R + d$.

Für den Bahnradius des inneren Rades gilt: $R_2 = R - d$.

Der Umfang des äußeren (inneren) Bogenstücks sowie die Geschwindigkeit des äußeren (inneren) Rades verändern sich im gleichen Verhältnis.

Relativ zum Radius R der Bahnkurve betragen die Änderungen

für das äußere Rad $R_1/R = 1 + d/R$ (3)

sowie für das innere Rad $R_2/R = 1 - d/R$ (4)

Da die Umfänge und Geschwindigkeiten proportional zu den Radien $R_{1/2}$ sind ergibt sich auch: $v_1/v = (1 + d/R)$ und $v_2/v = (1 - d/R)$.

Die eigentliche Kurvenfahrt erreicht man nun, indem man die gewünschte Geschwindigkeit für das äußere Rad mit $v_1 = v^*(1 + d/R)$ und das innere Rad mit $v_2 = v^*(1 - d/R)$ festlegt.

Falls eine exakte Geschwindigkeitsmessung vorliegt, so können die PWM-Werte für v_1 und v_2 entsprechend nachgeregelt werden bis die jeweilige Geschwindigkeit vorliegt.

In einer einfacheren Form, in der keine exakte Geschwindigkeitsmessung vorliegt, genügt es in erster Näherung, die PWM-Werte der beiden Antriebsmotoren mit den Faktoren (3) und (4) zu multiplizieren.

Die letztgenannte einfache Form wurde mit zufrieden stellendem Ergebnis realisiert. Wegen der hier vorliegenden geringen Untersezung des Kettenantriebes sind die Antriebsmomente an den Rädern leider zu klein, um Einflüsse von Hindernissen, z.B. Unebenheiten des Untergrundes, ausreichend zu unterdrücken. Bei schnellerer Fahrt wurde ein gutes Ergebnis erzielt.

Der Winkel α muss im Bogenmaß (rad) zur Verfügung stehen. Dazu genügt es, den Wert des A/D-Wandlers nach Abzug eines Ruhewertes mit einem passenden Faktor zu multiplizieren. Bei relativ kleinen Winkeln, hier bis ca. $15^\circ = 0,26\text{rad}$ kann eine Proportionalität mit noch vernachlässigbaren Fehlern angenommen werden.

8.6.3. Vereinfachung

Winkelfunktionen wie $\tan(\alpha)$ in (2) sind für Mikrocontroller eine zeitraubende Aufgabe.

Eine Reduzierung des Rechenaufwandes erreicht man, wenn die Eigenschaft $\tan(\alpha) \cong \alpha$ für kleine Winkel berücksichtigt wird. Auch bei Neigungswinkeln von ca. 30° verursacht diese Vereinfachung keine Probleme. Es entstand keine spürbare Veränderung der Kurvenfahrt.

Die Gleichung (2) reduziert sich zu: $R = v^2/(\alpha \cdot g)$ (2*)

Die Faktoren (3) und (4) werden zu: $\dots = 1 \pm d \cdot \alpha \cdot g / v^2 = 1 \pm c \cdot \alpha / v^2$ (3/4*)

mit der Konstanten $c = d \cdot g$, in der auch Anpassungen an die ADC-Werte enthalten sein können. Ohne Anpassung an ADC-Werte ergibt sich mit den vorgenannten Werten $c=3$.

8.6.4. Praktische Umsetzung 1. Art

In meinem Projekt ändert sich der ADC-Wert bei einer Winkeländerung von $\alpha = 20^\circ = 0,351 \text{ (rad)}$ um 80 Einheiten.

Somit ergibt sich ein Faktor $c_1 = 0,351/80 = 0,0044$ um den ADC-Wert in eine Winkelgröße im Bogenmaß (rad) umzurechnen.

Mit dem o.g. Wert für c findet man eine neue Konstante $c = 3 \cdot 0,0044 = 0,0132$

Ein PWM-Wert von 300 entspricht einer Geschwindigkeit $v = 18 \text{ km/h} = 5 \text{ m/s}$. Daraus ergibt sich $v = \text{PWM}/60$ bzw. $v^2 = \text{PWM}^2/3600$. Auch dieser Faktor kann in die Konstante c mit einfließen, so dass letztendlich $c = 3 \cdot 0,0044 \cdot 3600 = 47,5$ in dem aus (3/4) entstandenen Termen zu finden ist.

Aus (3/4*) entstehen somit ... $= 1 \pm d \cdot \alpha \cdot g / v^2 = 1 \pm 47,5 \cdot \text{ADC} / \text{PWM}^2$
(5)

Wählt man zur Realisierung die Multiplikation des PWM-Wertes mit dem Term (5) so ergibt sich der neue PWM-Wert aus $\text{PWM} \cdot (1 \pm 47,5 \cdot \text{ADC} / \text{PWM}^2)$
(6)

und nach dem Ausmultiplizieren: $\text{PWM} \pm 47,5 \cdot \text{ADC} / \text{PWM}$.

Das Ergebnis von $47,5 \cdot \text{ADC} / \text{PWM}$ kann nach der Berechnung jeweils mit dem geeigneten Vorzeichen zum PWM-Wert des jeweiligen Motors hinzugefügt werden.

Dieser letzte Term wurde im Programm umgesetzt.

8.6.5. Praktische Umsetzung 2. Art

Zu einem späteren Zeitpunkt stand bei meinem selbstbalancierenden Fahrzeug auch die Geschwindigkeit, gemessen in Metern pro Sekunde (m/s) zur Verfügung.

Gleichung (6) ergibt nach dem Ausmultiplizieren: $\text{PWM} \pm 47,5 \cdot \text{ADC} / \text{PWM}$, wobei der PWM-Wert statt der Geschwindigkeit benutzt wurde. Der ADC-Wert der Lenkung wurde durch den PWM-Wert statt durch die Geschwindigkeit dividiert.

Geht man zurück zur Geschwindigkeit in m/s und betrachtet Gleichung (3/4*) so ist erkennbar, dass der Radius sowie die Geschwindigkeit v für die Kurvenfahrt mit $1 \pm c \cdot \alpha / v^2$ multipliziert werden müssen.

D.h. $v_{1/2} = v \cdot (1 \pm c \cdot \alpha / v^2)$ bzw. $v_{1/2} = v \pm c \cdot \alpha / v$ mit dem oben bereits genannten Faktor $c=3$.

Das besondere Augenmerk gilt nun dem addierten Teil $c \cdot \alpha / v$.

Für $v=0$ ist dieser Term nicht definiert, was zu einem unkontrollierten Verhalten des Fahrzeuges führt. So wurde auch in einer frühen Entwicklungsstufe des Segways die Möglichkeit einer Kurvenfahrt bei $v=0$, d.h. eine Drehung auf der Stelle, ausgeschlossen.

Eine kleine Änderung behebt diesen Mangel und behält trotzdem die geschwindigkeitsabhängige Regelung der Lenkung fast unverändert bei.

Ändert man $c \cdot \alpha / v$ in $c \cdot \alpha / (0.1 + \text{ABS}(v))$ (7) ab, so wird eine Division durch 0 vermieden und die Drehung auf der Stelle möglich.

Der Wert 0,1 wurde willkürlich gewählt. Gleichung (7) liefert für kleine Geschwindigkeiten einen großen Wert, der bei der weiteren Steuerung durch das Maximum $\max(c \cdot \alpha / (0.1 + \text{ABS}(v)), 2 \cdot c \cdot \alpha)$ begrenzt wird.

Beide Formen der Umsetzung wurden ausführlich getestet.

In beiden Fällen war eine geringfügige Verbesserung der Kurvenfahrt festzustellen, die aber auch vor den beschriebenen Korrekturen unproblematisch war.

Da zum frühen Zeitpunkt dieser Versuche in der Softwareversion V5.xx noch keine genauere Erkenntnis der CPU-Auslastung des Mikrocontrollers vorlag, wurde diese Korrektur ab der Version V6.xx nicht mehr verwendet, was keinen wesentlichen Nachteil zur Folge hat.

9. Sicherheit

9.1. Technische Sicherheit

9.1.1. Aufbau

Der rein mechanische Aufbau muss im Wesentlichen die Bedingung der ausreichenden Belastbarkeit erfüllen.

Beim Gehäuse bestehen die tragenden Teile aus Aluminiumblech. Für das Trittbrett wurde (5,5 + 1,5)mm starkes Riffelblech verwendet. Unter der Platte befinden sich senkrecht zur Platte angeordnete gekantete Aluminiumbleche mit einer Stärke von 3 mm. Bei einem Belastungstest der 40cm x 50cm großen Bodenplatte des Fahrzeuges mit 120kg konnte kein Durchbiegen des Trittbrettes festgestellt werden.

Die Befestigung des Lenkers, der keine Belastung tragen muss, erfolgt mit dem Drehlager eines Fahrradpedals. Die durch das Pedal laufende Achse aus hochfestem Stahl erlaubt Belastungen von weit über 100kg.

Die Motoren, Getriebe und Räder sind besonderen Belastungen ausgesetzt. In der Ausführung des Runner2 wurde der Belastbarkeit der Teile in besonderer Weise Rechnung getragen. Die Motoren, Getriebe und Halbachsen wurden für die Verwendung in einem elektrischen Rollstuhl konzipiert. Dieser trug sein Eigengewicht von 130kg und zusätzlich das Gewicht des Fahrers bis 120kg. Die Antriebe, die für diese Belastung entwickelt wurden, dienen nun als sehr belastbare Grundlage für den Runner2. Auch die Bereifung wurde original übernommen, so dass diese Einheiten den weiteren Aufbau vereinfachten.

Beim vorangegangenen Bau des Runner1 folgten die Antriebe einem anderen Konzept. Zwei Motoren vom Typ MY1020 bildeten die Grundlage für das Vorwärtskommen. Neben den Motoren, die ihre Kraft über Zahnräder und Ketten auf die Räder übertrugen, lag die 18mm dicke, aus hochfestem Sperrholz gefertigte Grundplatte des Fahrzeuges auf einer 70cm langen und 25mm dicken Stahlachse, an deren Enden Räder befestigt waren, wie sie für Pocket Bikes oder Dirt Bikes zu finden sind. Die Belastbarkeit dieser Räder wurde vom Händler mit 100kg pro Rad angegeben.

Der Belastungstest durch einen 120kg wiegenden Fahrer zeigte keine Probleme.

9.1.2. Der Notausschalter

Warum ein Notausschalter?

Das Fahrzeug, welches nur senkrecht stehend seinen Bewegungszustand der Ruhe oder der gleichförmigen Bewegung nicht ändert, besitzt ein gewisses Eigenleben. Verlässt der Fahrer sein Gefährt freiwillig oder unfreiwillig, so könnte das Fahrzeug unkontrolliert weiterfahren. Um dies zu verhindern muss festgestellt werden, ob der Fahrer an Bord ist. Ein permanent zu drückender Handschalter, ein Stiftschalter, dessen Stift über eine am Fahrer befestigte Schnur beim Verlassen des Fahrzeuges herausgezogen wird oder Fußschalter sind mögliche Kontrollelemente. Das Ausschalten darf auf keinen Fall während der Fahrt erfolgen. Das Bewegen einer Hand, eine Körperbewegung oder das Entlasten eines Fußes könnten zu dem unweigerlich folgenden Sturz führen.

Vier parallele Fußschalter (Schließer) bieten hier ein hohes Maß an Sicherheit. Die Anordnung der Schalter unter dem Vorderfuß und der Ferse beider Füße führt dazu, dass ein unbeabsichtigtes Öffnen aller Schalter sehr unwahrscheinlich ist. Weiter wirkt sich eine Unterbrechung aller vier Sicherheitsschalter nur dann aus, wenn dieser Zustand länger als eine Sekunde anhält.

9.1.3. Elektronik

Ein wesentlicher Bestandteil der Elektronik sind die sogenannten H-Brücken, über die der Strom der Motoren gesteuert wird. Bei einer Leistungsaufnahme von rund 1000 Watt bei einer Versorgungsspannung von 36 Volt ist mit einem Gesamtstrom von rund 30A zu rechnen. Jede der beiden H-Brücken muss somit regelmäßig bis zu 15A ein und aus schalten. Die als elektronische Schalter verwendeten Leistungs-MOSFETs IRFB4110 erlauben einen Strom von 120A im Dauerbetrieb.

Schutzmaßnahmen gegen induktionsbedingte Spannungsspitzen verhindern sehr früh das Überschreiten der zulässigen Höchstspannung von 100V.

Alle anderen elektronischen Bauteile unterliegen keiner nennenswerten Belastung. Ein betriebsabhängiger Defekt ist somit auszuschließen.

9.2. Sichere Software

„Keine Software ist fehlerfrei.“

Dieser Satz ist rund 29.000 Mal im Internet zu finden, was sicher seine Berechtigung hat.

Viele Tests waren notwendig, um zu prüfen, ob das Programm wie erwartet auf unterschiedliche Ausgangssituation reagiert.

So beschrieb ein Selbstbauer seine Erfahrungen mit dem Satz: "Ich habe während meiner aktiven Zeit als Fußballer nie so blaue Schienbeine gehabt, wie während der Entwicklungsphase meines Fahrzeuges."

Schon beim Einschalten sind folgende Tests sinnvoll:

- Prüfung der Batteriespannung
- Anzeige der restlichen Batteriekapazität
- Messen alle Sensoren die erwarteten Werte
- Manuelle Prüfung der Funktion der Notausschalter

Auch während des Betriebes sind Kontrollfunktionen aktiv.

- Prüfung der Batteriespannung
- Anzeige der restlichen Batteriekapazität,
- Anzeige des aktiven Regelalgorithmus, der das Fahrzeug senkrecht hält
- Anzeige einer Überlastung der Motoren, Gesamtstrom > 36A

9.3. Verkehrssicherheit

Die Teilnahme am Straßenverkehr erfordert weitere Sicherheitsfaktoren, die auch bei Fahrten auf nicht öffentlichem Gelände sinnvoll erscheinen.

Auskunft hierrüber gibt die Mobilitätshilfenverordnung.

So verlangt:

§ 1 Anwendungsbereich u.a.

„Eine Prüfung der elektromagnetische Verträglichkeit“

Es soll sichergestellt werden, dass keine aktiven und passiven Störungen der eigenen bzw. fremden Elektronik auftreten können.

§ 4 Anforderung an die Verzögerungseinrichtung u.a.

Eine Verzögerungseinrichtung ..., die

1. das Fahrzeug bis zum Stillstand abbremsen kann und
2. mindestens einen Verzögerungswert von $3,5 \text{ m/s}^2$ erreicht.

Diese Bedingung muss ebenfalls erfüllt werden.

Auch ohne Betriebserlaubnis erscheint das Einhalten dieser Verzögerung aus Sicherheitsgründen sinnvoll. Eine eigene Prüfung ist recht einfach. Da es sich beim Bremsvorgang um eine (negative) beschleunigte Bewegung handelt, gelten hier die entsprechenden physikalischen Gesetzmäßigkeiten wie: $s = 0,5 \cdot a \cdot t^2$ und $v = a \cdot t$, wobei $a = 3,5 \text{ m/s}^2$ der oben genannte Verzögerungswert ist.

An einem einfachen Beispiel soll nun die Prüfungsmöglichkeit dargestellt werden.

Eine Geschwindigkeit von $12,6 \text{ km/h}$ entspricht $v = 3,5 \text{ m/s}$. Zum Erreichen dieser Geschwindigkeit muss das Fahrzeug genau eine Sekunde beschleunigen (oder bremsen).

In einer Sekunde legt das Fahrzeug einen Weg $s = 0,5 \cdot 3,5 \text{ m/s}^2 \cdot (1\text{s})^2 = 1,75 \text{ m}$ zurück. Somit muss das Fahrzeug nach dieser Strecke zum Stillstand kommen.

§ 5 Anforderung an die lichttechnischen Einrichtungen

In diesem Fall können die lichttechnischen Teile aus der Fahrradtechnik verwendet werden. Auf eine entsprechende Zulassungsnummer ist zu achten.

§ 6 Anforderung an die Schalleinrichtung

„Eine Mobilitätshilfe darf nur in Betrieb gesetzt werden, wenn sie mit einer Glocke ausgerüstet ist.“

10. Der Peukert-Effekt

10.1. Was ist der Peukert – Effekt?

Unter dem Peukert-Effekt bei Batterien versteht man die Tatsache, dass die entnehmbare Ladung in Abhängigkeit von der Entladestromstärke bei höheren Stromstärken z.T. deutlich unter die vom Hersteller angegebene Nennkapazität fällt. So findet man bei einem Bleiakku mit einer angegebenen Kapazität von 14Ah Kapazitätswerte wie z.B.

Kapazität (bei 25°C) - 20h Rate (0,7 A)	14 Ah
Kapazität (bei 25°C) - 10h Rate (1,31 A)	13 Ah
Kapazität (bei 25°C) - 5h Rate (2,24 A)	11,2 Ah
Kapazität (bei 25°C) - 1h Rate (9,04 A)	9 Ah
Kapazität (bei 25°C) - 15min Rate (25,9 A)	6,5 Ah

Tabelle 7.1

Die deutlich reduzierte Kapazität bei höheren Entladeströmen ist u.a. auf Verluste am Innenwiderstand der Batterie sowie auf die Tatsache, dass die bei der Stromabgabe auftretenden chemischen Reaktionen verzögert ablaufen, zurückzuführen.

Je nach Batterietyp (Alkalien, NiCd, ... , Blei-Säure) und deren Ausführung sind diese internen Verluste unterschiedlich hoch. Auch die Gesamtkapazität der betroffenen Batterie ist entscheidend für die relativen Verluste.

Der aus Böhmen stammende Ingenieur Wilhelm Peukert beschrieb bereits 1897 die Kapazität eines Akkumulators in Abhängigkeit von Entladestrom. Je höher der Entladestrom ist, desto weniger Energie kann einem Akkumulator entnommen werden.

10.2. Die Peukert – Funktion

Für nachfolgende Überlegungen wird die Peukertsche Gleichung in der Form:

$$C' = I' \cdot t = C \left(\frac{C}{I \cdot H} \right)^{k-1} \quad \text{verwendet.} \quad 9.1$$

(Quelle: http://en.wikipedia.org/wiki/Peukert's_law)

Das Ergebnis C' ist dabei die dem Akkumulator effektiv entnommene Ladung bei einem Entladestrom I .

Dabei sind:

- H die Zeitdauer der Entladung der Nennkapazität
- I der effektive Entladestrom
- C die Nennkapazität
- k Peukertzahl für den jeweiligen Akkutyp

Verschiedene Peukertzahlen:

Blei-Säure Akku	1,1 – 1,3
Blei-AGM Akku	1,05 – 1,15
Blei-Gel	1,1 – 1,25
LiPo Akku	1,05

Tabelle 7.2

Bsp.: Wird ein Blei-Säure Akku mit $k=1.2$, $C=100\text{Ah}$ bei $H=20\text{h}$ angegeben, so können

bei einer Stromstärke $I = 10\text{A}$: $C' = C \cdot \left(\frac{C}{I \cdot H} \right)^{k-1} = 100 \left(\frac{100}{10 \cdot 20} \right)^{0,2} = 87 \text{ Ah}$

entnommen werden.

Berücksichtigt man statt eines Entladestromes von $I = 10\text{A}$ nun einen „scheinbaren“

Entladestrom I' : in der Form: $I' = 10\text{A} \cdot (100\text{Ah} / 87\text{Ah}) = 11,5\text{A}$

so ergibt sich, dass der Blei-Akku nach $T = 100\text{Ah} / 11,5\text{A} = 8,7\text{h}$ entladen ist.

Dies ist ein gleichbedeutendes Ergebnis bezüglich des realen Entladestromes und der Entladezeit.

Für diesen *scheinbaren Entladestrom* I' gilt allgemein:

$$I' = I \cdot \frac{C}{C'} = I \cdot \frac{C}{C \cdot \left(\frac{C}{I \cdot H} \right)^{k-1}} = I \cdot \left(\frac{I \cdot H}{C} \right)^{k-1} \quad 9.2$$

Dieser so definierte Strom I' erlaubt nun eine einfache Berechnung der entnommenen Kapazität.

10.3. Berechnung der entnommenen Kapazität

Die Frage nach der Größe der in einer Batterie noch verfügbare Restkapazität lässt sich durch das Bestimmen der bereits entnommenen Kapazität beantworten.

10.3.1. Beispiel 1

Wird die Eingangs genannte 14Ah-Batterie 10 Minuten mit $I_1 = 15A (=2,5Ah)$ und 5 Minuten mit $I_2 = 4A (=0,33Ah)$ belastet, so ergibt sich die entnommene Kapazität im ersten Fall nach Peukert aus:

$$C_1 = I'_1 \cdot dt = \left(\frac{I_1 \cdot H}{C} \right)^{k-1} \cdot dt = 15 \cdot \left(\frac{15 \cdot 20}{14} \right)^{0.2} \cdot \frac{10}{60} = 27.7 \cdot \frac{10}{60} = 4.6Ah$$

Der *scheinbare Entladestrom* liegt hier bei 27,7A. Er fließt für 10/60 Stunden.

10.3.2. Beispiel 2

Im zweiten Falle gilt:

$$C_2 = I'_2 \cdot dt = \left(\frac{I_2 \cdot H}{C} \right)^{k-1} \cdot dt = 4 \cdot \left(\frac{4 \cdot 20}{14} \right)^{0.2} \cdot \frac{5}{60} = 5.67 \cdot \frac{5}{60} = 0,47Ah$$

Hier beträgt der scheinbare Entladestrom $I'_2 = 5,67A$.

Insgesamt wurde der Batterie die Ladung $C_1 + C_2 = 5,07Ah$ entnommen. Die Restkapazität beträgt demzufolge rund 9Ah.

10.3.3. Variable Kapazitätsentnahme

Bezogen auf das Thema dieses Buches muss in Abhängigkeit von der Belastung der Motoren mit einer stets schwankenden Stromstärke gerechnet werden. In diesem Fall ist eine kontinuierliche Strommessung erforderlich.

Für den Fall einer Steuerung durch einen Mikrocontroller kann die Strommessung in einer Laufschleife oder auch bei der Abarbeitung eines Interrupts erfolgen. Liegt zwischen den Messungen eine konstante Zeitdifferenz dt vor, z.B.

$$dt = 10ms = \frac{0.01s}{3600s/h} = \frac{1}{360000}h$$

, oder

$$dt = 1s = \frac{1s}{3600s/h} = \frac{1}{3600}h$$

so ergibt sich nachfolgender Algorithmus für die Erfassung innerhalb einer Interruptsequenz.

10.3.4. Der Algorithmus

Festlegung der Batteriedaten:

H = 20 (h); k = 1,2; C = 14 (Ah) Anfangskapazität

C₂ = 14 (Ah) Variable für die Restkapazität

Interrupt:

I = Erfassung der Stromstärke über AD-Wandler

$I' = I \cdot \left(\frac{I \cdot H}{C} \right)^{k-1}$ Berechnung der scheinbaren Stromstärke I'

$C_2 = C_2 - I' \cdot dt$ Berechnung der Restkapazität

Return Ende des Interrupts

Das Speichern der Restkapazität vor dem Ausschalten in einem EERAM und das Erkennen eines vollgeladenen Akkus beim Einschalten sind weitere Schritte, die im übergeordneten Programm erfolgen sollten.

Warnung: EERAMs erlauben nur eine begrenzte Anzahl von Schreibzyklen. Somit führt das unmittelbare Speichern der Restkapazität nach jeder Berechnung zum schnellen Ausfall dieses permanenten Speichers im Mikrokontroller.

10.3.5. Praktische Umsetzung

In der Praxis verfügt nicht jede Programmiersprache über die Möglichkeit die angegebene Exponentialfunktion zur Basis a zu berechnen. Eine Exponentialfunktion zur Basis e=2,718... und der zugehörige Logarithmus sind in der Regel vorhanden.

Im vorliegenden Fall ist ein Wert der Form

$$I' = a^x \quad \text{mit } a = \frac{I \cdot H}{C} \text{ und } x = k - 1 = 0,2 \text{ zu berechnen.}$$

Nach der beidseitigen Anwendung des Logarithmus ergibt sich

$$\log(I') = x \cdot \log(a)$$

Die rechte Seite dieser Gleichung wird nun $\log(a) = \log\left(\frac{I \cdot H}{C}\right)$ ggf. mit der Vereinfachung

$$x \cdot \log\left(\frac{I \cdot H}{C}\right) = x \cdot \left(\log(I) + \log\left(\frac{H}{C}\right)\right) \text{ berechnet.}$$

Das Anwenden der Exponentialfunktion liefert nun

$$\left(\frac{I \cdot H}{C}\right)^x = \exp^{x \cdot \log\left(\frac{I \cdot H}{C}\right)} = \exp^{x \cdot (\log(I) + \log\left(\frac{H}{C}\right))} \text{ berechnet.}$$

Nachfolgende Zeilen zeigen die in Einzelschritte zerlegte Berechnung des relevanten Stromes mit dem BASCOM-Compiler. Im vorliegenden Fall wird mit Log() der natürliche Logarithmus ln() benutzt.

Interrupt:

```
.
Temp = Log(I)           ' I nach dem Auslesen des AD-Wandlers
Temp = Temp + 0.356675  ' +log(H/C) = log(20/14) = 0.356675
Temp = Temp * 0.2       ' Multiplikation mit x = k-1
I2 = I * Exp(temp)      ' Anwendung der e-Funktion

I2 = I2 / 3600          ' bei Messungen im Sekundenabstand
C2 = C2 - I2
.
Return
```

Die Variable I_2 beinhaltet nun den für die Kapazitätsberechnung relevanten Strom.

Die Berechnung der verbrauchten Kapazität ist sinnvoll.

Viel wichtiger ist sorgfältige Kontrolle des Ladezustandes vor jeder Fahrt, um einem Versagen des Fahrzeuges vorzubeugen.

Die Selbstentladung der Blei-Akkumulatoren bleibt dabei natürlich unberücksichtigt, so dass nach einer längeren Standzeit stets eine Nachladung erforderlich ist.

10.4. Berechnung der Peukertzahl aus Batteriedaten

In den vorangegangenen Beispielen wurde die Peukertzahl $k = 1,2$ benutzt. Dass diese Zahl, je nach Batterietyp bzw. –ausführung unterschiedlich sein kann, wurde bereits erwähnt.

Das Bestimmen dieser Zahl einer unbekannten Batterie ist mit zwei bekannten Messwerten möglich, wie im folgenden Teil gezeigt wird.

Tabelle 7.1 weist bei 20-stündiger Entladung mit $I = 0,7A$ eine Kapazität $C = 14 Ah$ aus. Bei 5-stündiger Entladung mit $I = 2,24A$ können der Batterie $C = 11,2 Ah$ entnommen werden.

In diesem Fall wäre die Nennkapazität von $14 Ah$ nach 5-stündiger Entladung mit einem *scheinbaren Strom* $I' = 14Ah / 5h = 2,8A$ ebenfalls entnommen.

Ausgehend von Gleichung 7.2, sie zeigt den Zusammenhang zwischen dem tatsächlichen Strom I und dem *scheinbaren Strom* I' , ergibt sich für:

$$I' = I \cdot \left(\frac{I \cdot H}{C} \right)^{k-1} = 2,24 \cdot \left(\frac{2,24 \cdot 20}{14} \right)^{k-1} = 2,8A$$

Diese Gleichung ist nach k aufzulösen. Eine erste Vereinfachung liefert: $1,25 = (3,2)^{k-1}$

Es folgt nach der Anwendung eines Logarithmus: $k - 1 = \frac{\log(1,25)}{\log(3,2)} = 0,1918$ bzw.

$$k = 1,1918 .$$

Dies zeigt, dass der Wert $k = 1,2$ auch in diesem Fall zutreffend ist.

Führt man die gleiche Rechnung für den Fall einer einstündigen Entladung mit $I = 9,04A$ durch und berücksichtigt, dass eine Kapazität $C = 9,04Ah$ entnommen werden kann, so ergibt sich aus der Zeit und der Gesamtkapazität ein *scheinbarer Entladestrom* $I' = 14A$.

$$\text{Es folgt: } I' = 14 = 9,04 \cdot \left(\frac{9,04 \cdot 20}{14} \right)^{k-1} = 9,04 \cdot 12,91^{k-1}$$

$$\text{und weiter } k - 1 = \frac{\log(14/9,04)}{\log(12,91)} = 0,171 \quad \text{bzw. } k = 1,171$$

Dieser geringfügige Unterschied ist kein Widerspruch zum vorherigen Wert. Bei stärkerer Belastung erscheint diese Batterie relativ besser als bei schwächerer

Belastung. Bei der Angabe des Herstellers wurden die angegebenen Tabellenwerte möglicherweise auch gerundet, was zum besseren Ergebnis führt.
Bei einer sich anschließenden Anwendung erscheint ein mittleres $k = 1,15$ als geeigneter Wert. Eine größere Sicherheit bietet weiterhin der etwas schlechtere Wert mit $k = 1,2$.

10.5. Realisierung der Strommessungen

Voraussetzung für die Messung der Kapazitätsentnahme ist eine Messung des Entladestromes. Strommessungen werden in der Regel mit Hilfe eines Shunts durchgeführt. Gemessen wird der an diesem Widerstand auftretende Spannungsabfall, aus dem sich die Stromstärke berechnen lässt.

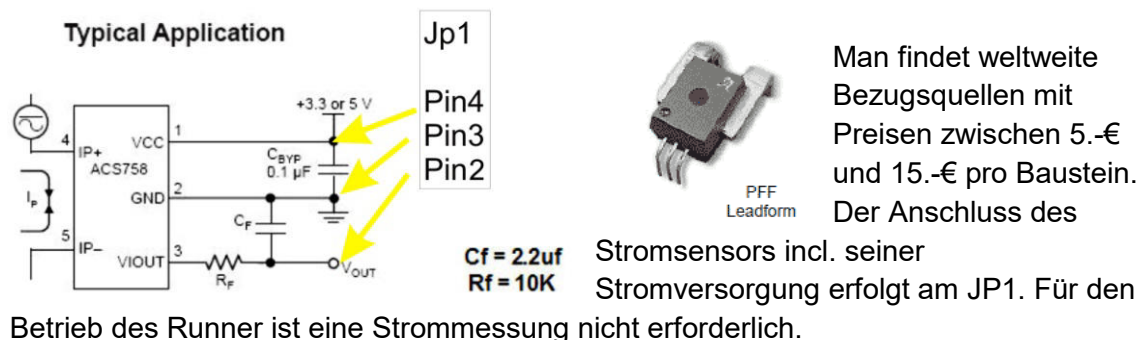
Wesentliche Nachteile dieser Methode sind der genannte Spannungsabfall am Shunt, eine Spannung, die dem angeschlossenen System verloren geht. Der größeren Erwärmung dieses Widerstands muss Rechnung getragen werden.

Die gemessene Spannung ist Teil des gesamten Stromkreises und sie kann nur mit zusätzlichem Aufwand und einer geeigneten Potentialverschiebung einem AD-Wandler des Mikrocontrollers zugeführt werden kann.

Die moderne Elektronik greift hier auf einen physikalischen Effekt zurück.

Schon 1879 entdeckte der Physiker Edwin Hall den später nach ihm benannten Hall-Effekt. Die Verschiebung bewegter Elektronen in einem Magnetfeld führt zu einer messbaren Spannung. Derartige Sensoren erlauben das Messen von Wechselströmen und liefern ein Ausgangssignal, welches von dem zu messenden Stromkreis galvanisch getrennt ist.

Der hier vorgestellte Baustein ACS758-050U bietet eine vollständige galvanische Trennung. Er erlaubt Strommessungen bis 50A und schwache Ströme in den umgekehrten Richtungen und bietet ein direkt von den gängigen Mikrocontrollern auswertbares analoges Ausgangssignal.



10.6. Kontrolle der Strommessung

Der Sensor ACS758-050U liefert an seinem Ausgang eine Offsetspannung von 600mV, die dem Mikrocontroller über JP1-PIN2 zugeführt wird. Dieser Wert kann im stromlosen Zustand geprüft werden.

Um die korrekte Funktion des Atmega zu prüfen, kann an PIN2 eine geeignete Spannung angelegt werden. Besonders einfach ist ein Spannungsteiler mit den Widerständen 1kOhm an GND und 6,8kOhm an VCC zu realisieren, welcher an seinem Abgriff eine Spannung von rund 645mV liefert. Die Software errechnet daraus einen Strom von 0,7A und zeigt diesen im Display an.

Verkleinert man den Widerstand von 6,8kOhm durch Parallelschaltung eines zweiten auf 3,4kOhm, so muss im Display ein Strom von 9A angezeigt werden.

Kleine Abweichungen sind wegen unvermeidbarer Widerstandstoleranzen möglich.

11. Bauanleitung

Dies ist keine Bauanleitung mit vorgegebenen Abmessungen und Teilelisten. Die Beschaffung vieler Teile ergibt sich aus Gelegenheitskäufen, da vorbereitete Teile für den Bau eines selbstbalancierenden Fahrzeuges nicht angeboten werden. Vielmehr sollen Hinweise zur Ausführung der eigenen Planung gegeben werden.

11.1. Motoren und Räder

Verschiedene Motorvarianten wurden bereits beschrieben. Beim Runner1 wurden zwei der Motor MY1020 mit einer Leistung von je 500 Watt bei 24V mit einem Ritzel mit 11 Zähnen verwendet. Montagewinkel am Motorgehäuse sind sehr zu empfehlen. Große Kettenblätter mit rund 80 Zähnen oder auch mehr sind von Vorteil, da durch diese Untersetzung eine kräftigere Beschleunigung erreicht wird.

Motoren dieser Art findet man mit Hilfe von eBay oder auch bei <http://www.ritzelladen.de/>.

Dieser Versender liefert auch passende Ritzel, Kettenräder und Ketten, die in der Länge anzupassen sind.

Schubkarrenräder mit einem Durchmesser von 40cm sind die preiswerteste Lösung. In diesem Fall müssen Kettenräder unter Verwendung von Abstandsbolzen mittig montiert werden. Gleiches lässt sich mit Sackkarrenrädern mit einem Durchmesser von 26cm erreichen. Diese Räder benötigen eine Achse, die sinnvollerweise mit einer Länge von maximal 69cm bemessen wird. Geeignete Stahllachsen findet man in Baumärkten. Ein Durchmesser von 20mm ist ausreichend und passt in der Regel zu den Achslöchern der Räder. Stellringe aus dem Baumarkt sorgen für den Halt der Räder.



Auch Stützräder für Schubkarren, wie im nebenstehenden Bild mit Achsstummel, können an einem stabilen Gehäuse befestigt werden.

Weitere Räder findet man mit dem Suchwort „Pocketbike Hinter Vorder Rad“ ebenfalls bei eBay. Bei der Bestellung von zwei Hinterrädern sind die notwendigen Kettenräder bereits montiert. Ggf. müssen zur Anpassung an die vorhandene Achsdicke die eingebauten Rillenkugellager ausgetauscht werden.



Bei einem Raddurchmesser von 40cm sind Kettenräder mit mindestens 80 Zähnen sinnvoll.

Liegt der Durchmesser bei 26cm, so sollten mindestens 52 Zähne vorhanden sein.

Eine weitere Möglichkeit sind zwei Motoren aus einem elektrischen Rollstuhl, die mit Getriebe und Rädern eine kompakte Einheit bilden. Während Zahnräder und Ketten ein variables Übersetzungsverhältnis ermöglichen, ist dies bei den Rollstuhlantrieben fest vorgegeben. Häufig findet man Motoren mit Getriebe aber ohne Räder. Und schon beginnt die Qual der Wahl.

Rollstuhlantriebe mit 250 Watt bzw. 350 Watt sind für Geschwindigkeiten von 6km/h bzw. 10km/h vorgesehen.

Beides ist für einen Runner zu langsam. Der Versuch einen 250 Watt-Motor mit 36 Volt zu betreiben, führte zu einer Geschwindigkeit von knapp 11km/h. Der leistungsstärkere Motor drehte erfreulich schnell und erreichte 22km/h im Leerlauf. Das war ideal. Nach diesen Vorüberlegungen war dies die Wahl für meinen Runner2.

Elektroscooter, die auch als Seniorenmobile angeboten werden, sind bis 15km/h schnell.

Auch deren Antriebe mit Getriebe und Rädern sind für das eigene Fahrzeug geeignet. Ausgesonderte Teile bzw. Scooter findet man über die Vertreiber und deren Reparaturwerkstätten.

Weitere Quellen sind Sanitätshäuser, die Elektrorollstühle vertreiben, diese warten und nach Gebrauch zurück nehmen. Hier empfiehlt sich eine deutschlandweite Suche.

11.2. Plattform-Gehäuse

Mein **Runner1** bestand aus einer Sperrholzplatte mit den Abmessungen 46cm x 38cm. Eine durchgehende Stahlachse mit einer Länge von 69cm stützte die Plattform und erlaubte das Befestigen der Pocketbike-Räder, bei denen dank eines hervorragenden Mechanikers die überbreiten Radlagerstummel abgefräst und Kugellager für eine Achse mit 20mm Durchmesser eingesetzt wurden. Batterien und Motoren und H-Brücken wurden unter der Holzplatte befestigt, die weitere Elektronik fand auf der Plattform ihren Platz.

Die starre Achse bot eine hervorragende Stabilität und Belastbarkeit. Die stabile Befestigung der Motoren erforderte größere Anstrengungen.

Der häufige Zugang zur Elektronik war relativ einfach, da diese z.T. auf der Oberseite der Plattform angebracht war. Das teilweise Umdrehen der Plattform war ebenfalls möglich, da ihre Breite mit 38cm kleiner war als der Reifendurchmesser mit 40cm.

Es folgte der **Runner2** mit einem vollständig anderen Konzept.

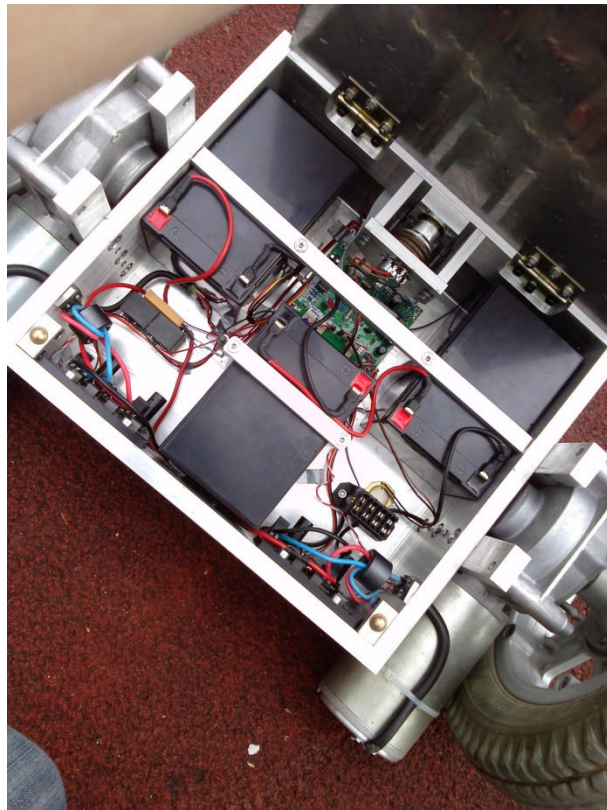
Rollstuhlantriebe mit 350 Watt bei 24 Volt und einer Geschwindigkeit von 10km/h erreichen bei 36 Volt erfreuliche 22km/h im Leerlauf. So entstand eine Plattform, unter der diese Motoren montiert waren. Das 6mm dicke Aluminium Riffelblech mit einer Abmessung von 51cm x 40cm wurde auf der Unterseite durch abgewinkelte Verstrebungen verstärkt, um ein Durchbiegen der Plattform zu verhindern. Alle Batterien, 3 x 12Volt/14Ah sind unter der Plattform angebracht. Auch die Hauptplatine und die H-Brücken mit ihrer Verkabelung liegen unter Aluminiumplatte. Bei Ausfahrten werden alle Teile durch ein Aluminiumblech abgedeckt.

Dieses Konzept bietet eine saubere Standfläche, ist aber mit einem wesentlichen Nachteil verbunden. Für alle Arbeiten muss das Gerät umgedreht werden. Ein spezieller Ständer erleichterte die Arbeiten auf der Unterseite, erforderte aber das Heben von fast 40kg bis auf Tischhöhe.

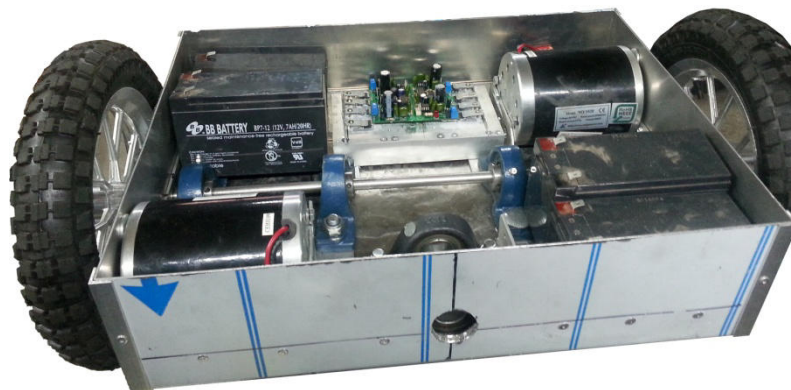
Meine Bandscheiben haben sich nach einigen Wochen über diese Prozedur beklagt. Daraus ergab sich ein ca. 2,30m hohes dreibeiniges Gestell, welches mit einem Flaschenzug ausgestattet wurden, um nun das Heben und Umdrehen meines Runner2 zu erleichtern.

Das Konzept des **Runner3**, dessen Mechanik von Rainer und Heiner gebaut wurde, berücksichtigte diesen Umstand. Ein Gehäuse mit einem verschraubten 16mm starken Aluminiumrahmen und außen liegenden Rollstuhlantrieben sowie einem nach oben zu öffnendem Deckel nimmt drei Akkus und die gesamte Elektronik auf. Diese Konstruktion erwies sich als außerordentlich servicefreundlich.

Bei der Planung des Gehäuses und der Anordnung der Akkus muss auch schon die Befestigung des Lenkers berücksichtigt werden. Diese besitzt eine besondere Bedeutung.



Andreas hat seinen Runner auch von oben zugänglich geplant. Hier finden die zwei Motoren MY1020 ihren Platz so, dass diese mit außenliegenden Zahnrädern betrieben werden. Die ebenfalls außen, eng am Gehäuse liegenden Kettenräder und die Ketten sind noch zu montieren. Eine durchgehende, innenliegende Achse verleiht den Rädern eine ausreichende Stabilität.



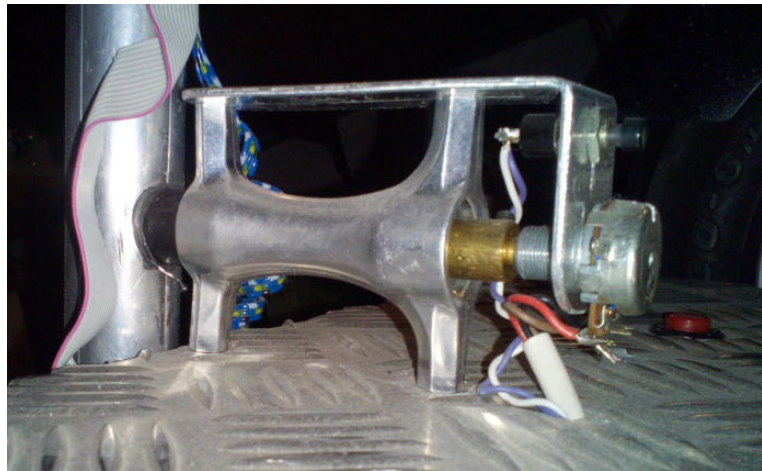
11.3. Lenkung

Neben der Richtungsänderung hat der Lenker auch die Aufgabe, eine Kippbewegung der Plattform zu verhindern. Die Kraft eines nach vorne beschleunigenden Rades kippt gleichzeitig die Plattform nach hinten. Die Sensoren erhalten somit eine falsche Information über den Bewegungszustand des Fahrzeuges, womit das wesentliche Problem genannt wurde.

Aus dem genannten Grund ergibt sich, dass die Verbindung der Plattform oder des Gehäuses mit dem Lenker in Fahrtrichtung gesehen starr sein muss.

Aus diesem Grund ist eine durchgehende Lenkerstange, die ggf. unten rund um 90° gebogen wurde, häufig Anwendung. Der untere Teil endet in zwei Drehlagern um eine starre Verbindung zu garantieren.

Eine recht zierliche Befestigung war mit Hilfe eines alten Fahrradpedals möglich. Beim Fahrradhändler fand sich ein Pedal aus Aluminium in der Abfalltonne. Die schon vorhandenen Gewinde ermöglichten eine problemlose Befestigung auf der Unterlage und das Anschrauben der



Abdeckung. Die Pedalachse aus hochfestem Stahl ist in der Regel für eine Belastung von mehr als 100kg vorgesehen, so dass man ein Verbiegen ausschließen kann. Diese Pedalachse wurde bei nebenstehender Konstruktion wie üblich in die Tretkurbel eingeschraubt, die sich mit einer Länge von fast 15cm im Inneren des Lenkerrohres befindet und, hier nicht sichtbar, mit einer zusätzlichen Schraube an einer tiefer liegenden Stelle mit dem Lenkerrohr verschraubt wurde. An der durchgehenden Achse konnte rechts gleichzeitig das Lenkerpotentiometer befestigt werden.

Weitere Teile aus dem Fahrradladen ergänzen den Lenker, dessen Griffe mit einer Höhe von 105cm über der Plattform und einem Vorbau von 10cm eine bequeme Auflage für die Hände bietet.

Das Display mit 2 x 24 Zeichen zeigt Betriebsdaten, auch während der Fahrt.

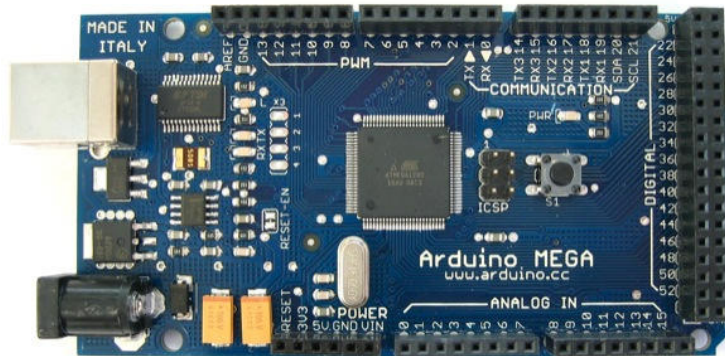
Der „Zündschlüssel“ verhindert die unerlaubte Verwendung.



11.4. Elektronik

Die elektronische Hardware wurde bereits beschrieben. Trotzdem werden hier auch andere Konzepte vorgestellt.

Arduino-Module mit einer ausreichenden Anzahl von PINs und einem Mikrocontroller mit mindestens 32kB Speicher und 16MHz Taktfrequenz sind als Basis geeignet.

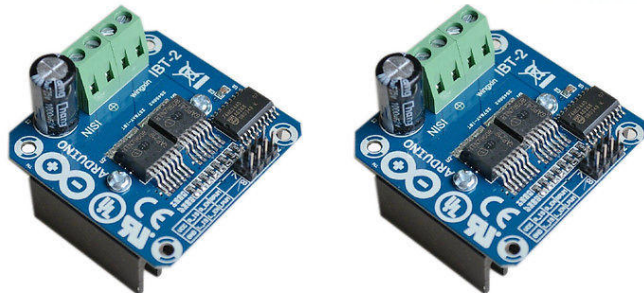


Für dieses Modell kann ich keine Software zur Verfügung stellen.

Ein bereits an anderer Stelle beschriebener Beschleunigungs- und Gyrosensor MPU6050 und



zwei H-Brücken, hier zwei Module mit je zwei BTS7960B Halb-Brücken, ergänzen die notwendige Elektronik.



BTS-Modifikation beachten!

Mindestens zwei Akkus mit einer Spannung von je 12 Volt und einer Kapazität von 12-14 Ah sind notwendig. Sie bringen einige Kilogramm auf die Waage. Alternativ sind auch vier Akkus á 7Ah möglich.

Li-Ionen Akkus sind ideal und sehr teuer.

Nun fehlen noch einige Verbindungskabel und dann kann es losgehen.

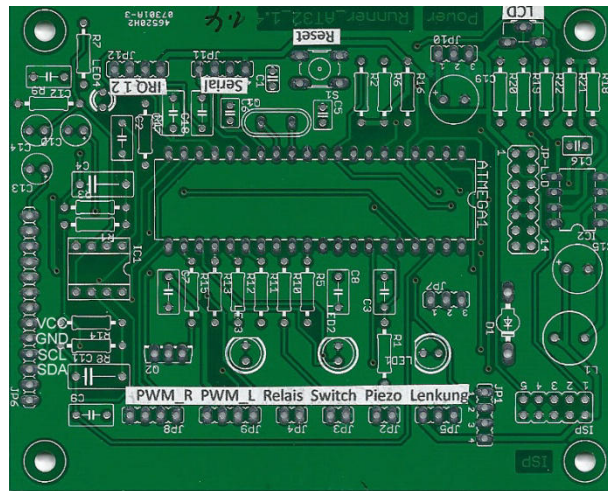


Meine Alternative zu den gezeigten Bausteinen ist eine passend entwickelte Hauptplatine, die in mehreren Entwicklungsstufen alle speziellen Bedürfnisse für die Steuerung eines selbstbalancierenden Fahrzeuges erfüllt.

Eine Hauptplatine, hier die Version 1.4 für die Verwendung mit 24Volt und den links bereits angegebenen Anschlüssen für den digitalen Sensor MPU6050.

Natürlich kann diese Platine auch bei der Verwendung analoger Sensoren eingesetzt werden.

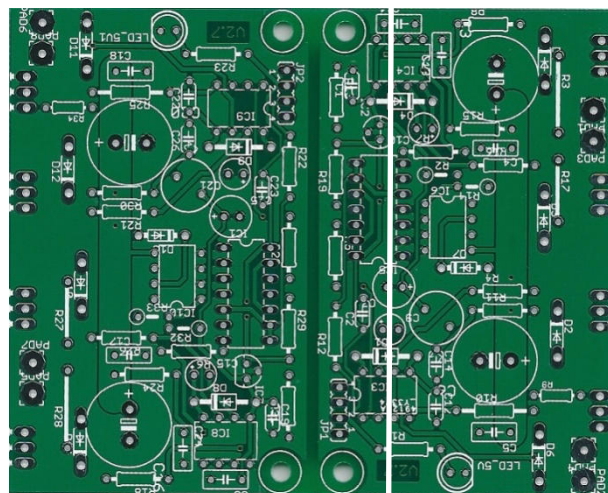
Die Platinenversion 2.1 besitzt im Unterschied zur vorherigen Version 1.4 einen anderen Schaltregler zur Erzeugung der internen Betriebsspannung von 5 Volt. Dieser etwas teurere Regler erlaubt bei Bedarf theoretisch Eingangsspannungen bis 60 Volt. Verwendung findet natürlich auch der oben beschriebene Beschleunigungs- und Gyrosensor MPU6050.



Zwei H-Brücken, die je nach Wahl des Spannungsreglers für 24 Volt und 36 Volt verwendet werden können.

Als Grundlage für die Teilebestellung dienen die in den EAGLE .brd-Dateien enthaltenen Bauteillisten.

Auch die Blei- oder Li-Ionen-Akkumulatoren sind unerlässlich. Vervollständigt werden die Platinen des letztgenannten Konzeptes durch eine speziell für diese Kombination erstellte Software.



11.5. Modifikation der BTS-H-Brücken

Die in der Regel aus China stammenden BTS7960B-H-Brücken sind verlockend preiswert.

Daraus ergab sich der Versuch, diese für den 24V-Betrieb geeigneten H-Brücken für mein Projekt zu verwenden.

Erste Laborversuche zeigten, dass sich die angeschlossenen Motoren bei kleinen PWM-Werten kaum drehen.

Der Blick ins Datenblatt führte zur Erklärung dieses Verhaltens. Die verwendeten H-Brücken besitzen eine einstellbare Ein- und Ausschaltverzögerung, welche die aktive Zeit eines PWM-Signals am Ausgang verkürzt.

Messungen ergaben, dass bei einem PWM-Wert 50 (von 512) gerade mal ein Strom von 340mA - 370 mA fließt.

11.5.1. Leistungsoptimierung

Eine Abhilfe ist einfach möglich. Die genannte Verzögerungszeit wird durch einen externen Widerstand eingestellt. Es empfiehlt sich, hierfür einen Widerstand von 0 Ohm (Drahtbrücke) zu wählen, um die kürzeste Totzeit der Halbbrücken festzulegen. Pro H-Brücke müssen zwei SMD-Widerstände durch Drahtbrücken ersetzt oder überbrückt werden.

(Ggf. sind zwei Lötkolben zum einfachen Auslöten des SMD-Widerstandes erforderlich.)

Eine Vergleichsmessung zeigt, dass beim gleichen PWM-Wert 50 der Strom auf rund 780mA gestiegen ist, ein brauchbares Ergebnis.

Vergleicht man das Ergebnis mit der Stromaufnahme der selbst entwickelten H-Brücke, so misst man 820mA.

Dieser Wert ist auf die noch kürzeren Verzögerungen beim Ein- und Ausschalten zurückzuführen.

11.5.2. Temperaturoptimierung

Die BTS-H-Brücke besitzt einen "Pseudo-Kühlkörper". Dieser sitzt getrennt vom IC BTS7960B auf der anderen Platinenseite, was seine Kühlwirkung praktisch unmöglich macht. Auffallend ist somit die starke Erwärmung der Bausteine, wobei der Kühlkörper sich fast nicht erwärmt.



Nach dem Abschrauben des Kühlkörpers und dem Auftragen von Wärmeleitpaste auf der Platinenrückseite hat sich die Wärmeabfuhr deutlich verbessert.

Weiter besteht die Möglichkeit über den ICs unter Verwendung von Wärmeleitpaste eine Aluminiumplatte oder einen zweiten passenden Kühlkörper zu befestigen, der mit längeren Kühlkörperschrauben angepresst wird.

Vergleich der Wärmeentwicklung:

Die nicht modifizierten BTS-Module wurde schon bei einer Belastung von 50W (KFZ-Lampe, Stromaufnahme der Brücke 25V / 2A) richtig heiß, so dass die interne Temperatursicherung die H-Brücke nach ca. 20 Sekunden ausgeschaltet hat. Die o.g. Modifikation der Widerstände reduziert diese Erwärmung deutlich. Nach 40 Sekunden ging jedoch auch hier die angeschlossene KFZ-Lampe aus. Wärmeleitpaste zwischen Platine und Kühlkörper führte dazu, dass in dem nachfolgenden 3-Minuten Test keine Überhitzung und thermische Abschaltung auftrat.

11.5.3. Einschaltverhalten und Fehlerbeseitigung

Während in der ersten halben Sekunde nach dem Einschalten die Spannung an den Kondensatoren noch langsam ansteigt, kann es vorkommen, dass eine H-Brücke bereits den Motorstrom einschaltet, bevor die Software einen definierten Zustand an den PWM-Eingängen der BTS-Module hergestellt hat.

Update:

Die Ursache für das unerwartete Einschalten der H-Brücke wurde gefunden.

Während der ersten Taktzyklen des Mikrocontrollers erfolgt die Initialisierung aller Ein- und Ausgänge des Atmega32.

Beim Einschalten des Atmega32 befinden sich alle Ports als Eingang definiert, so dass ein notwendiges Ausgangssignal "1" noch nicht verfügbar ist. Der Eingang des nachfolgenden Bausteines C4001 erhält somit einen falschen Wert "0", damit wird ein angeschlossener Motor für den Bruchteil einer Sekunde eingeschaltet.

Ergänzung: Auf den Platinen M_2.5+ ist der Baustein CD4001 verbaut. An diesem Baustein müssen die Pins 5 und 9 mit je einem Widerstand von ca. 10kOhm -100kOhm mit +5V verbunden werden, um einen logischen Wert "1" zu erzeugen.

Einfacher ist das Einlöten der Widerstände in freie Löcher von JP8/1-3 sowie JP9/1-3.

11.5.4. Vergleich mit der eigenen H-Brücke

Meine H-Brücke wurde in diesen Test mit einbezogen.

Nach einem 3-Minuten Test hatte sich ein MOSFET ohne Verwendung eines Kühlkörpers kaum spürbar erwärmt.

Die Gründe liegen in wesentlich kürzeren Schaltzeiten der MOSFETs IRFB4110 sowie in einem deutliche geringeren $R_{DS(on)}=3.8m\Omega$ gegenüber $7m\Omega - 9m\Omega$ bei den BTS-Halbbrücken.

11.5.5. Fazit:

Bei der Verwendung der BTS-H-Brücken in einem selbstbalancierenden Fahrzeug anderer Art mit leistungsschwächeren Motoren (ca. 250W) und kleiner Belastung gab es keine Ausfälle. Die mit Wärmeleitpaste verbesserten Kühlkörper wurden rund 70° warm.

Die von mir im Runner1 verwendete Eigenbau-H-Brücke führte bei der Verwendung von 24V/500 Watt Motoren und einer leichten Bergfahrt zu handwarmen Temperaturen der montierten Kühlkörper.

Im Runner2, der mit 36V betrieben wird, konnte ich keine Erwärmung feststellen, da die MOSFETs mit Isolierpads direkt auf das Alu-Chassis montiert sind.

Im Ergebnis kann ich die BTS-H-Brücken für einen normalen Runner nur dann empfehlen, wenn den Motoren keine große Leistung abverlangt wird.

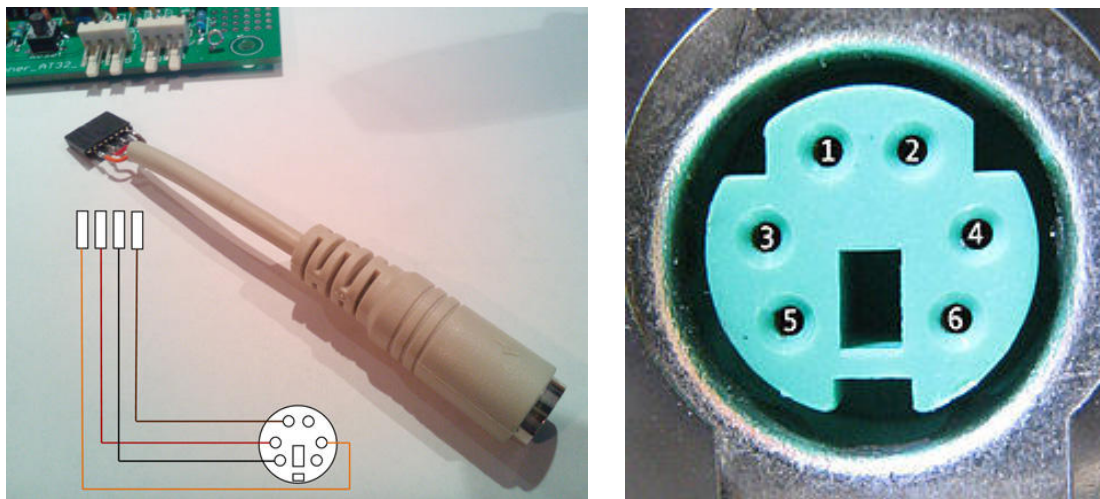
11.6. Anschluss der Tastatur

Die Software beinhaltet verschiedene Parameter, deren Veränderung das Fahrverhalten beeinflusst. Zu Beginn waren es verschiedene analoge Sensoren, deren Empfindlichkeiten verschieden waren. Auch unterschiedliche Motoren erforderten angepasste Parameter.

Nicht zuletzt erfordert auch die Testphase eine Reduzierung der Stärke des Motorantriebes, um wilde Reaktionen des Fahrzeuges zu vermeiden.

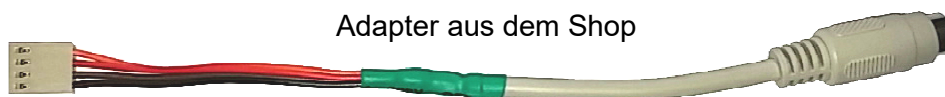
An dieser Stelle möchte ich aus Sicherheitsgründen darauf hinweisen, bei ersten Tests die Parameter PWM_max = 100 und PWM_faktor_max = 50 einzustellen.

Das Taktsignal für eine alte PS/2-Tastatur wird an PortD.0 ausgegeben und Daten werden an PortD.1 gelesen. Diese Anschlüsse entsprechen der Standardschnittstelle des Atmega32 für die serielle Kommunikation.



Ein Adapter und die Anschlussbelegung der entsprechenden Buchse:

Atmega	JP11 serial	PS/2	Bild
TXD(PD.1-15)	1	Data	1
GND	3	GND	3
VCC	4	+5V	4
RXD(PD.0-14)	2	CLK	5

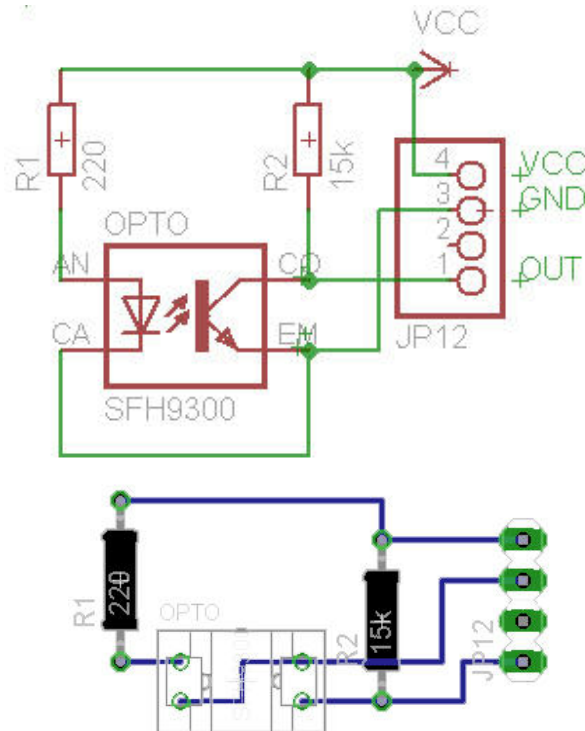


11.7. Anschluss der Lichtschranke

Zur Bestimmung der Geschwindigkeit, die auch der dynamischen Balanceregung dient und somit eine Verbesserung der Balance bei höheren Geschwindigkeiten bewirkt, dient eine Gabellichtschranke, welche die Umdrehungszahl des Motors erfasst.



Zu beachten ist dabei, dass die Anschlüsse CA und EM in der Regel **diagonal** angeordnet sind. Vor dem Einbau der Lichtschranke sollte nun die Funktion geprüft werden. Das Unterbrechen des Lichtstrahles zwischen der Leuchtdiode und dem Fototransistor muss ein Ansteigen der Spannung am Anschluss C bis fast 5V zur Folge haben.



Beim Runner1 wurde diese Lichtschranke neben dem Ritzel am Motor befestigt und der Lichtstrahl durch die einzelnen Zähne unterbrochen.

Beim Runner2 wurde auf der Motorachse, die geringfügig aus dem Motor herausragte, eine vierblättrige Scheibe aufgeklebt, neben welcher die Lichtschranke ebenfalls angeklebt wurde.

Neben GND und VCC vom JP12 wurde der Kollektorausgang CO dieser Lichtschranke mit PIN1 von JP12, dem Eingang des Interrupt INT0 verbunden.

Beim Runner2 mit einem Radumfang von 1,13m und einer Getriebeübersetzung von 25:1 sind für eine Radumdrehung 25 Motorumdrehungen erforderlich. Mit dem 4-flügligen Unterbrecher auf der Motorachse entstehen $4 \cdot 25 = 100$ Interrupts. Die Wegstrecke pro Interrupt beträgt somit $1,13\text{m} / (4 \cdot 25) = 1,13\text{cm}$.

Diese Wegstrecke pro Interrupt ist unter Verwendung der Taste W als Parameter einzutragen bzw. abzuändern.

11.8. Software

Alle Software bis zur Version 6.42 wurde von mir für analoge Sensoren erstellt. Diese Ausführung wird nicht weiter gepflegt, wie bereits an anderer Stelle begründet wurde. Für den digitalen Sensor steht Software mit der Versionsnummer V7.xx zur Verfügung. Verschiedene Parameter werden beim ersten Start und nach ihrer Änderung im EERAM des Atmega gespeichert.

Beim Softwareupdate bleiben diese Parameter unverändert.

11.8.1. Download

Downloads stehen auf der Webseite www.ups.bplaced.de zur Verfügung.

11.8.2. Kompilieren

Der BASCOM-Code, der auf der o.g. Internetseite herunter geladen werden kann, muss noch kompiliert werden. An dieser Stelle streikt die Freeware des BASCOM Compilers, der jedoch nach dem Laden die Ansicht des recht großen Programmes erlaubt. In Einzelfällen kann ich das Kompilieren meines Programmes auch für verschiedene Varianten von geeigneten Atmegas durchführen.

11.8.3. Atmega programmieren

Hier kann problemlos auch auf die Freeware des BASCOM Compilers zurück gegriffen werden.

Ein sehr preiswerter USB-Programmer mit der Bezeichnung USBASP ist für weniger als 5.-€ im Netz zu finden. Treiber für diesen Programmer habe ich auf meiner Internetseite zur Verfügung gestellt. Diese werden beim Anschließen der USBASP automatisch gesucht. Die Installation klappt bei Windows 7 -10 jedoch nur, wenn vorher die Installation nicht signierter Treiber erlaubt wird.

11.8.4. Updates

Diese werden grundsätzlich über den Downloadbereich meiner Homepage zur Verfügung gestellt.

Eine individuelle Anpassung des Programmes ist von meiner Seite nicht vorgesehen. Die zugänglichen Parameter bieten dazu ausreichend Gelegenheit.

11.8.5. Hinweise zur Inbetriebnahme der Hauptplatine

Nach dem Bestücken der Hauptplatine den Mikrocontroller **nicht sofort** einstecken.

Die drei PINs am Anschluss der Stromversorgung sind: 5V GND 24V-36V
Zuerst die Platine an eine Spannung von ca. 12V zwischen GND und dem rechten PIN-1 des JP10 anschließen.
Nun muss von GND zum linken PIN-3 eine Spannung von knapp 5V zu messen sein.

Erst jetzt den Widerstand R20 prüfen und einlöten.
Dieser Widerstand zwischen 10kOhm und 47kOhm muss so ermittelt werden, dass die Spannung am o.g. linken PIN relativ genau 5V beträgt.
Ein loses Einsetzen geeigneter Widerstände in die Löcher von R20 sorgt schon für einen ausreichenden Kontakt und erlaubt das Messen der Spannung.
Nach dem Einlöten des ausgesuchten Widerstandes bitte die geplante Versorgungsspannung von 24V bis 36V an den rechten PIN-1 anlegen und am linken PIN erneut 5V überprüfen.

Wichtig: Beim Wechsel von Komponenten stets die Versorgungsspannung entfernen.

Danach das Display anschließen und das Kontrast-Potentiometer ganz nach links drehen.
Erneut 12V-36V am rechten PIN-1 anlegen. Das Display zeigt nun viele dunkle bzw. helle Felder in der ganzen Anzeigefläche. Dies ist normal.

Nun kann der Mikrocontroller in die Fassung eingesteckt werden.
Erneut die Versorgungsspannung anschließen.
Im Display erscheint nun die Versionsnummer der Software.
Der Kontrast kann am Kontrast-Potentiometer weiter optimiert werden.

Es folgt der Anschluss des Tongebers (Lautsprecher, Piezo) und der MPU-6050 Platine. Beim erneuten Einschalten hört man nun eine kleine Melodie.

Für weitere Schritte sind nun ein Lenker-Potentiometer und ein Fußschalter erforderlich.

Der Jumper JP7/2-3 muss bei der Verwendung der BTS-H-Brücken nur gesteckt sein, wenn diese JP8 und JP9 angeschlossen werden. (Siehe BTS im Display)

Zur Durchführung weiterer Tests wird PIN JP7/1 mit GND z.B. JP1/3 verbunden.
Die dann ablaufenden Tests werden an einer anderen Stelle der Dokumentation ausführlich beschrieben.

Bitte unbedingt alle Tests durchführen!!

Bei diesen Tests sollen in der letzten Phase auch die H-Brücken, zuerst mit Lampen und später mit Motoren, angeschlossen sein.

Diese Tests sollten auch nach dem Einbau der Platine in das Fahrzeug durchgeführt werden, bevor die Software im Normal-Betrieb verwendet wird.

Wenn der Sensor korrekt montiert ist und das Fahrzeug nach vorne gekippt wird, werden Winkel negativ angezeigt.

Drehen sich die Motoren bei einer Neigung des Fahrzeuges in die falsche Richtung, so ist die Polarität der Anschlüsse zu tauschen.

Wird beim Anschluss einer falschen Versorgungsspannung eine Mitteilung im Display ausgegeben, so wechselt die Software in den **Parameter-Modus** und erwartet eine Eingabe über eine angeschlossene PS/2-Tastatur.

11.9. Hinweise zur Montage

Eine Lieferung umfasst ggf. die Hauptplatine, das Display (schon fertig verkabelt) und ein breites Flachbandkabel für den Anschluss der BTS-Module.

Vorbereitungen:

1. Für den Anschluss einer alten Tastatur benötigt man einen PS/2 Adapter. Dieser ist, wie in der Dokumentation gezeigt, anzufertigen.
2. Eine Verbindung an JP7/1-3 ist für den Testmodus erforderlich. Ohne das Kabel startet die Software im Normalmodus. **Unbedingt den Testmodus so lange benutzen, bis auch das fertige Fahrzeug damit schon einmal korrekt bewegt wurde. Ein Balacieren ist nicht möglich!**
3. Der Sensor muss nicht auf der Platine befestigt werden. Er muss jedoch um 45° nach vorne geneigt sein. Dies ist in der Dokumentation beschrieben. Später sollte der Sensor in der Nähe der Fahrzeugachse befestigt werden. Die Verbindung zur Hauptplatine erfolgt dabei über ein vieradriges Kabel.
4. Fragen auf dem Display werden durch das Drücken des Switch = Fußschalter mit „ja“ beantwortet.
5. Die Stromversorgung besitzt drei Anschlüsse in der Reihenfolge:
(+5V) (GND) (24V/36V)
6. LCD und das breite Flachbandkabel sind bei betriebsbereiten Platinen bereits angeschlossen. Eingebaute Metallbrücken verhindern einen falschen Anschluss.
7. Die Modifikation der BTS-H-Brücken, die ich auf meiner Internetseite beschrieben habe, sollte beachtet werden.
8. Das Bild mit BTS-H-Brücken im Shop lässt auch die Verkabelung erkennen. Beachte bitte auch beim späteren Aufbau, dass es je einen gemeinsamen Punkt für den Anschluss der Batterie (VCC=24V) und (GND= -) gibt!
9. Ich empfehle sehr, die Hauptplatine mit Sensor, H-Brücken, LCD, Akkus und Motoren zuerst einmal auf einem Tisch aufzubauen, damit alle Verkabelungen verständlich werden. Auch die Sicherung, das Relais, den Taster und den Piezo-Schallwandler anschließen. Die Sicherung sollte am Anfang bei ca. 5A, später bei ca. 30A -50A liegen.
10. Zu Beginn ist es empfehlenswert den Aufbau mit 12V steigend bis max. 27V(BTS) bzw. 39V aus einem Netzteil zu versorgen.
11. Bitte die H-Brücken zuerst mit einem kleinen PWM-Tester prüfen. Dieser ist in der Dokumentation beschrieben.

12. Die Gabellichtschranke kann später nachgerüstet werden. Die Geschwindigkeit muss gemessen werden, damit eine bessere Stabilität erreicht wird.
13. Der km-Stand kann bei der Änderung der Parameter auf „0“ gesetzt werden.
14. Bitte stets die aktuelle Dokumentation beachten.

11.10. Funktionsprüfung

11.10.1. Normalmodus

Dies bedeutet einfach fahren, ohne Einstellungen vorzunehmen.

Der Runner startet im Normalmodus, prüft, ob eine Tastatur angeschlossen ist und fährt dann in diesem Modus fort.

11.10.2. Fußschalter

Zur Aktivierung der Motoren müssen die Fußschalter 4x gedrückt werden. Dies ist eine Funktionsprüfung der Schalter.

Während der Fahrt muss stets mindestens ein Schalter geschlossen sein, sonst werden die Motoren deaktiviert.

11.10.3. Tastatur - Parametermodus

Mit einer PS/2 Tastatur ist im Parametermodus die Änderung aller Parameter möglich. Ob eine Tastatur angeschlossen ist, wird direkt nach Ablauf der Startmelodie geprüft. Während dieser Prüfung läuft eine Sternreihe ***** ... durch das Display. Das Drücken einer Taste in diesem Zeitfenster aktiviert den Parametermodus.

11.10.4. Offsetwerte

Wird beim Einschalten des Runners ein Fußschalter gedrückt gehalten, so wird ein scheinbarer Fehler ERROR=99 angezeigt. Im Anschluss werden die im Ruhezustand vorhandenen Offsetwerte der Sensoren neu eingelesen, um z.B. die horizontale Lage der Plattform und die Mittelstellung des Lenkers neu festzulegen.

Zu den Offsetwerten zählen z.B. die horizontale Plattformstellung und die Lenkermittelstellung.

11.10.5. Testmodus

Eine Verbindung der PINs an JP7/1-3 startet beim Einschalten des Runners den Testmodus.

Die damit verbundenen Prüfungen sind beim Aufbau des Runners unbedingt erforderlich.

Einzelne Prüfungen können bei einer entsprechenden Frage im Display für spätere Starts deaktiviert werden. Zu Beginn des Testmodus können alle Tests durch das Drücken eines Fußschalters wieder aktiviert werden.

In einem nachfolgenden Kapitel wird der Testmodus genauer beschrieben.

11.11. Empfehlung

Verschiedene Motorvarianten wurden bereits vorgestellt.

Meine erste Empfehlung gilt der Kombination von Motoren und Antriebe aus einem elektrischen Rollstuhl, die für eine Geschwindigkeit von 10km/h gebaut sind, bei einem Betrieb mit 36V jedoch gut 20km/h erreichen.

An zweiter Stelle folgen MY1020-Motoren (ohne Getriebe) mit 500Watt bei 24Volt. Der Betrieb mit 36Volt ist problemlos möglich, da ein 100%iger PWM-Wert nie erreicht wird und die Motoren in ihrer Leistung nicht überfordert werden. Der Vorteil liegt in einer höheren Leistung für den Fall, dass ein Kippen des Fahrzeuges aufzufangen ist.

Nicht zu empfehlen sind MY1020-Motoren mit integriertem Getriebe, da das Getriebe nicht für eine Belastung durch ein direkt befestigtes Rad konzipiert ist und bereits mehrfach von Lagerschäden berichtet wurde.

11.11.1. Akkus

An erster Stelle stehen hier eindeutig Li-Ionen Akkus, die bei einer Serienschaltung von 10 x 4.2V eine Anfangsspannung von 42V aufweisen. Mit einer Kapazität von ca. 10Ah entsteht ein sehr leichtes aber auch sehr teures Akkupaket.

An zweiter Stelle folgt ein Akku-Paket bestehend aus 3 x 12V Bleiakkus und einer Kapazität von 14Ah. Diese Batterien sind z.T. recht preiswert aber auch sehr schwer.

Nicht zu empfehlen ist eine Batteriekapazität von nur 7Ah, da dies neben einer geringeren Betriebsdauer auch den Nachteil eines höheren Innenwiderstandes mit sich bringt. Im Fall einer stärkeren Belastung bricht die Spannung deutlich ein.

11.11.2. Mainboard

Die Hauptplatine unterliegt einer ständigen Weiterentwicklung.

Nach mehreren Entwicklungsstufen steht nun eine Platine in der Version M_2.6 zur Verfügung, die speziell für eine digitale Datenübertragung nach dem I2C-Protokoll ausgelegt wurde. Der passende Sensor MPU-6050 beinhaltet Gyroskop und einen Beschleunigungssensor. Er ist zudem sehr preiswert.

Ergänzt wurde ein Steckanschluss für ein Flachbandkabel, so dass die entsprechenden Komponenten mit Hilfe nur einer Steckverbindung angeschlossen werden können und Verwechslungen ausgeschlossen sind.

Ein weiterer Baustein zur Ansteuerung der H-Brücken wurde auf diese Platine übernommen. Dieser und ein Spannungsregler werden auf jeder H-Brücke eingespart.

Der Anschluss der mehrfach genannten BTS-H-Brücken ist ebenfalls vereinfacht möglich.

11.11.3. H-Brücken

Hier gebe ich der von mir aufgebauten H-Brücke den eindeutigen Vorzug. Diese Brücke, basierend auf dem Hochleistungs-MOSFET IRFB4110 besitzt eine sehr hohe Strom- und Spannungsfestigkeit und ist für die auftretenden Belastungen mehr als ausreichend.

Beim Aufbau ist zu beachten, dass die Leiterbahnen, welche durch hohe Ströme belastet werden, durch Kupferdrähte oder versilbertem Schaltaht mit einem Durchmesser von 1mm verstärkt werden.

Eine Alternative findet man in den H-Brücken BTS7960 (43A) bzw. BTS7970 (68A), die verlockend preiswert sind. Einschränkend ist hier der Betrieb mit nur 24V (max. 28V). Bei einer zu hohen Betriebsspannung bzw. zu starker Erwärmung schalten die BTS-Brücken einfach ab. Letzteres kann im Betrieb zu unangenehmen Ausfällen führen. Die bei fertigen Modulen vorhandenen Kühlkörper werden unverständlicherweise ohne Wärmeleitpaste montiert und sind somit wirkungslos. Modifikationen werden an anderer Stelle beschrieben.

In der Praxis hat sich gezeigt, dass es auch qualitative Mängel bei den verwendeten ICs gibt, die letztendlich zur Auslieferung defekter Module geführt haben.

Eine Empfehlung für diese H-Brücken kann ich daher nicht ohne Einschränkung aussprechen.

11.11.4. Chassis

Hier sind der eigenen Fantasie keine Grenzen gesetzt.

Sehr sinnvoll und ausreichend ist eine Fahrzeugbreite von 69cm incl. der Räder. Eine Tiefe der Trittfläche von 40cm ist ebenfalls ausreichend. Die Achse der Räder sollte ca. 2cm hinter der Mitte verlaufen. Mehrfach gezeigt wurden Konstruktionen die den Zugang zu Batterien und Elektronik von oben ermöglichen. Bei meinem Runner2 habe ich dies leider nicht bedacht, so dass ein häufiges Heben und Drehen des recht schweren Fahrzeuges notwendig wurde.

11.11.5. Räder und Übersetzung

Hier muss gerechnet werden. Eine Konzeption des Fahrzeuges für eine maximale Geschwindigkeit von 20km/h bestimmt diese Komponenten. Ausgehend von der

Nennndrehzahl des Motors führen eher kleinere Räder mit einem Durchmesser von 30cm bis 36cm und eine Übersetzung durch ein Getriebe oder Kettenräder und –ritzel im Verhältnis 8:1 bis 10:1 zu einer geeigneten Anpassung.
Dem Aufstecken der Räder auf eine durchgehende 20mm dicke Achse ist gegenüber der Einzelbefestigung der Räder am Chassis der Vorzug zu geben.

Rechenbeispiel:

Ein Raddurchmesser von 40cm führt zu einem Umfang von 1,25m.
20km/h entsprechen 20000m/h bzw. 333m/min.
Somit sind $333\text{m}/1,25\text{m} = 266$ Umdrehungen pro min erforderlich.

Bei einer Höchstdrehzahl des Motors von 2500/min ergibt sich ein Übersetzungsverhältnis $266 : 2500 \approx 1 : (2500/266) = 1 : 9,4$

In der Praxis sollten Werte von 1:8 bis 1:9 gewählt werden.

11.11.6. Lenkerbefestigung

Ein häufig diskutierter Punkt.

Wesentlich ist eine kräftige Verbindung zwischen Lenker und Gehäuse, da alle Beschleunigungskräfte der Räder über den Lenker ausgeglichen werden müssen.
Bei zu loser Befestigung können zusätzlich Vibrationen entstehen.

Varianten zur Lenkerbefestigung sind:

Zwei Kugellager mit einer anschraubbaren Kugellagerhalterung

Ein Fahrradpedal (wie an anderer Stelle abgebildet)

Die Lenkerdurchführung eines Kinderrades

Gummi-Metall-Puffer mit beidseitigen Schraubgewinden

Alle Varianten können unter, in bzw. auf dem Gehäuse befestigt werden. Ggf. ist eine Versteifung des Lenkers erforderlich, um Biegeschwingungen zu vermeiden.

Die Befestigung eines Lenkerpotentiometers ist mit einzuplanen.

Die Mittelstellung kann durch zusätzliche Federn oder Stoßdämpfer erreicht werden.

Gummi-Metall-Puffer führen zu einer automatischen Mittelstellung.

11.11.7. Sicherheit

Ein wesentlicher Punkt ist ein Aktivierungsschalter, welcher im ausgeschalteten Zustand den elektrischen Antrieb der Motoren verhindert. Nur im gedrückten Zustand funktionieren die Motoren.

Das unfreiwillige Öffnen des Schalters während der Fahrt hat entsprechende Folgen. Um Stürze zu vermeiden wurde dieser Schalter 4-fach parallel ausgelegt. Das Öffnen aller Schalter, die sinnvollerweise unter den Vorderfüßen und Fersen angeordnet werden, ist dann während der Fahrt extrem unwahrscheinlich.

Wichtig: Keine Software ist fehlerfrei. Diese Fehlerfreiheit kann auch ich nicht garantieren.

Trotzdem stelle ich mich vertrauensvoll auf mein Fahrzeug und habe Spaß.

12. Betriebsanleitung

12.1. Das Fahren

Das Fahren ist denkbar einfach. Aufsteigen und losfahren.

Dies ist genau so einfach, wie es sich anhört, wobei trotzdem einiges falsch gemacht werden kann.

Was man wissen sollte ist, dass das Fahrzeug sich selbst ausbalanciert. Jede Abweichung der Plattform von der Waagerechten hat eine Fahrbewegung des aktivierten Systems zur Folge.

Nach dem Einschalten vergehen einige Sekunden, bis alle Funktionen des Systems initialisiert und geprüft sind und das Fahrzeug fahrbereit ist. Dies wird durch das Blinken der roten Kontrolllampe signalisiert.

Die erste Hürde ist das Aufsteigen auf das Gerät. Hält man sich am Lenker fest und zieht diesen zu sich, so reagiert die eingebaute Elektronik und die Plattform kommt ebenfalls auf den Fahrer zugefahren, was auch mit einer schmerzhaften Kollision am Schienbein enden kann. Richtiges Aufsteigen erfolgt, indem man den Lenker nicht zu sich heran zieht sondern einen Fuß auf die Mitte der Plattform setzt, seinen Körper nach vorne verlagert und nun sein Körpergewicht wie beim Besteigen einer Treppenstufe nach oben bewegt.

Die wichtigste Bedingung ist, den Lenker nicht zu sich heranzuziehen.

Beim ersten Versuch kann ein Helfer, der vor dem Fahrzeug steht, den Lenker festhalten. Dies wird auch bei kommerziellen Fahrzeugen bei der Einweisung praktiziert.

Der Fahrer sollte für den Fall, dass ein leichtes Schaukeln entsteht, auf keinen Fall durch eine eigene Balanceversuche eine Stabilisierung versuchen. Dies verursacht ein sich aufschaukelndes Schwingen des Fahrzeuges, was zu einem schmerzhaften Sturz führen kann.

Es hat sich als hilfreich erwiesen, den Abstand des Lenkers zum Bauch des Fahrers konstant zu halten.

Hat man das Fahrzeug ausbalanciert, so setzt sich das Fahrzeug durch eine leichte Neigung der Plattform bzw. des Lenkers nach vorne in Bewegung. Eine rückwärtige Neigung bremst das Fahrzeug ab bzw. erzeugt eine Rückwärtsfahrt.

Eine seitliche Neigung des Lenkers ermöglicht das Fahren von Kurven sowie das Drehen des Fahrzeuges auf der Stelle.

12.2. Die Einschaltprozedur im Detail

Das Einschalten erfolgt durch das Drehen eines Schlüssels oder durch die längere Betätigung des Einschaltknopfes. Dies muss bis zum hörbaren Einschaltklick des Hauptrelais erfolgen.

12.2.1. Sicherheitsprüfungen

Im Normalfall erscheint nach einer kurzen Melodie und dem Einschalten des Hauptrelais im Display eine Aufforderung zum Test der vier Fußschalter, die in beliebiger Reihenfolge gedrückt werden sollten.

Nach der Melodie läuft eine Reihe von Sternen * durch die erste Zeile des Displays. Wird während dieser Zeit eine Taste einer angeschlossenen PS/2-Tastatur gedrückt, so führt dies in den **Parametermodus**, der an späterer Stelle beschrieben wird.

Die Betätigung der vier Fußschalter soll deren Funktion sicher stellen indem geprüft wird, ob die Kontakte in den Schaltern schließen und sich auch wieder zuverlässig öffnen.

Werden die Fußschalter nicht gedrückt, so schaltet sich der Runner nach ca. einer Minute selbständig aus.

Es folgt eine interne Prüfung der Ruhewerte aller Sensoren. Für diese Prüfung ist es notwendig, dass das Fahrzeug korrekt, d. h. in keiner Weise schief, steht und sich auch nicht bewegt. Auch der Lenker muss senkrecht stehen. Liegt einer dieser Ruhewerte nicht im Normalbereich, so wird eine Fehlermeldung „**ERROR=xx**“ ausgegeben. Das Fahrzeug schaltet sich aus und muss neu eingeschaltet werden.

12.2.2. Fehlernummern

Die angezeigte Fehlernummer gibt Auskunft über den aufgetretenen Offset-Fehler.

Fehlernummern:

- | | |
|----|---|
| 1 | Neigungssensor (Phi) |
| 2 | Neigungsgeschwindigkeit Gyro (Phi_rate) |
| 4 | Giersensor |
| 8 | Lenker oder Lenkerpoti |
| 16 | Seitliche Neigung zu groß |
| 99 | manuell ausgelöste Initialisierungen der Sensoren |

Treten mehrere Fehler auf, so wird die Summe der Fehlernummern angezeigt.

12.2.3. Automatische Initialisierung

Beim ersten bzw. erneuten Einschalten wird eine Fehlernummer angezeigt. Es folgt die Aufforderung, das Fahrzeug waagrecht und ruhig aufzustellen. Nach dem erneuten Betätigen des Fußschalters wird die Initialisierung durchgeführt und das Fahrzeug automatisch ausgeschaltet.

Das Fahrzeug kann nun zur Benutzung neu eingeschaltet werden.

12.2.4. Manuelle Initialisierung

Der Initialisierungsprozess kann auch manuell gestartet werden.

Ausgelöst wird diese Initialisierung, indem beim Einschalten ein Fußschalter betätigt wird. Intern wird dieses Ereignis als Fehler mit der Nummer 99 gespeichert und das Fahrzeug automatisch ausgeschaltet.

12.2.5. Erste Initialisierung

Wird ein neuer Chip eines Mikrocontrollers zum ersten Mal benutzt, so werden Sensoren geprüft und als fehlerhaft im Vergleich mit EERAM-Werten erkannt. Es folgt nun die Prozedur der Automatischen Initialisierung.

12.3. Der Parametermodus

Der Parametermodus ermöglicht das Ändern interner Parameter, die das Verhalten des Fahrzeuges wesentlich beeinflussen.

Diese Änderungen sind mit Vorsicht durchzuführen, da sie zu unkontrolliertem Verhalten des Runners führen können.

12.3.1. Bedeutung der Parameter

Jede Änderung der in der nachfolgenden Tabelle angegebenen Parameter hat Auswirkungen auf das Fahrverhalten des Fahrzeuges.

Um Änderungen nur sinnvoll durchzuführen, ist sollte sich aus der Beobachtung des eigenen Fahrzeuges entsprechende Notwendigkeiten ergeben.

Die nun folgende Beschreibung ist dabei hilfreich.

F1 A

A regelt, wie stark die Motoren auf die Neigung des Runners reagieren.

Eine zu starke Reaktion führt ggf. zu einem heftigen hin- und her bewegen des Runners.

Die dabei entstehende Schwingbewegung beim Schließen der Fußschalter kann zur Kollision mit den Schienbeinen führen.

F2 B

B regelt, wie stark die Motoren auf die Neigungsgeschwindigkeit oder Kippgeschwindigkeit des Runners reagieren.

Eine zu starke Reaktion führt auch hier zu einem heftigen Hin- und Herbewegen des Runners und Gefahren wie bei A.

Ein hoher Faktor B verursacht eine schnellere Reaktion des Runners auf das Neigen der Plattform.

Die Faktoren A und B beeinflussen sich gegenseitig, so dass ein Schwingen durch das Erhöhen von A durch eine Reduzierung von B verhindert werden kann. Die Summe der Werte A und B sollte 1 nicht überschreiten.

Die Wahl der Parameter hängt auch von der Leistung der Motoren sowie der Übersetzung der Zahnräder/Getriebe ab.

- F3 C**
C regelt die Übernahme der Neigung und der damit verbundenen Beschleunigung zu einer Geschwindigkeit. Werte zwischen 10 und 20 sind brauchbar. C hat auch Einfluss auf die Wirkung von A und B und umgekehrt.
- F4 D**
Der Faktor D bestimmt den Integralanteil des PID-Filters.
- F5 Ref_Speed A und B**
Die Theorie und auch die Fahrpraxis zeigen, dass die Stabilisierung bei höheren Geschwindigkeiten nicht mehr so direkt reagiert wie bei langsamer Fahrt. In diesem Fall würden größere Werte für A und B zu einer besseren Stabilisierung führen.
Daraus ergibt sich, dass die Parameter A und B in Abhängigkeit von der Geschwindigkeit vergrößert werden. Die vorgegebenen Referenzwerte sind 20 mit der Bedeutung, dass bei einer Geschwindigkeit von 20km/h eine interne Verdopplung von A und B erreicht werden. Es ist ggf. sinnvoll hier experimentell kleinere geeignete Werte zu bestimmen, wenn die Balanceregung bei höheren Geschwindigkeiten zu „weich“ wird.
- F6 Filter_komp**
Dies ist der Parameter für den digitalen Komplementärfilter. Dieser Filter unterdrückt das Rauschen des Beschleunigungssensors und fügt die Änderungen des Gyrosensors-Integrals zur Winkelbestimmung hinzu.
- F7 Filter_MPU6050**
Parameter für den internen Tiefpass des Sensors MPU6050 ab Software V7.29
Die Parameter sind ganzzahlig mit Grenzfrequenzen laut Datenblatt:
1~185Hz, 2~85Hz, 3~43Hz, 4~21Hz, 5~10Hz, 6~5Hz
- Beide Filter reduzieren die durch A und B eingestellte Empfindlichkeit der Regelung und reduzieren somit auch eine Schwingneigung des Fahrzeuges.
- F8 Lenk_faktor**
Dieser Faktor bestimmt, in welchem Maß die Lenkerneigung zu einer Kurvenfahrt führt.

Filter_lenkung

Dieser Filterwert dämpft die Messwerte des Lenkerpotentiometers, um kleine Abweichungen/Erschütterungen zu reduzieren.

F9 Gier_faktor

Dieser Faktor bestimmt, in welchem Maß die eine Geradeausfahrt durch den z-Gyrosensor, der eine ungewollte Drehbewegung um die z-Achse registriert, stabilisiert wird. Dieser Faktor wirkt auch der Lenkbewegung entgegen.

Filter_gier

Dieser Filterwert dämpft die Messwerte des z-Gyrosensors bei Drehungen um die z-Achse, um kleine Abweichungen/Erschütterungen zu reduzieren.

F10 PWM_Mode

Charakteristisch für eine PWM-Ansteuerung ist die PWM-Frequenz, mit der die Motoren ein- bzw. ausgeschaltet werden. Diese beträgt hier 15.625kHz und ist nicht hörbar. Eine PWM-H-Brücke ist in der Regel nur bis zu einer bestimmten Frequenz problemlos brauchbar. Jedes Umschalten der Leistungs-MOSFETs führt zu einer Wärmeentwicklung, welche durch die Wahl einer niedrigeren PWM-Frequenz reduziert werden kann. Die Modi 2 und 4 arbeiten mit 10,4kHz, die Modi 3 und 5 mit 7,8kHz. Letztere sind für **BTS-Module** zu empfehlen.

F11 Pwm_max

Die Erzeugung des PWM-Signals mit 9 Bit erlaubt einen Maximalwert 511, der auf **keinen Fall** überschritten werden darf. Dies bedingt im Normalbetrieb eine Begrenzung auf 500. Bei Tests kann durch die Reduzierung dieses Wertes auf z.B. 100 die Motorleistung und somit auch die Verletzungsgefahr beschränkt werden. Im Testmodus ist dieser Wert auf 150 festgelegt.

F12 Pwm_faktor_max

Damit das Ein- und Ausschalten der Motoren nicht ruckartig geschieht, wird ein PWM_Faktor in ca. 2 Sekunden bis zum Wert von Pwm_faktor_max erhöht bzw. auf 0 reduziert.

Dieser Faktor wirkt auf das Ergebnis von A, B und C.

Änderungen sind bei diesem Faktor nicht erforderlich.

B U_batt_faktor

Der über einen Spannungsteiler und AD-Wandler gewonnene Wert der Batteriespannung führt bei Widerstandstoleranzen zu einer geringfügig

fehlerhaften Spannungsanzeige. Dieser Faktor kann entsprechend geändert werden.

D Display_mode

Das Display kann für Entwicklungszwecke verschiedene interne Werte anzeigen, die mit dem Display_mode 1 bis 8 ausgewählt werden. Die angezeigten internen Werte können dem Programmcode entnommen werden. Sie werden ggf. verändert und aus diesem Grund hier nicht beschrieben.

K Eingabe der Batteriekapazität

Die korrekte Anzeige der Restkapazität erfolgt nur bei Verwendung eines Stromsensors.

O Set all offset values

Die im Ruhezustand vorhandenen Offsetwerte der Sensoren (horizontale Position, Gyrooffset, Lenkermitte) werden neu eingelesen und für die spätere Verwendung im EERAM abgespeichert.

R Runner_typ

Der aktuelle Runner trägt die Nummer 23. Ab der Softwareversion V7.xx hat diese Nummer keine Auswirkungen auf interne Abläufe.

T Direkter Zugang zum Testmodus

U U_min1

Nach dem Betätigen des Hauptschalters werden verschiedene Kondensatoren über einen Widerstand geladen. Erst beim Erreichen der Spannung U_min1 wird das Hauptrelais verzögert eingeschaltet, um übermäßige Kontaktbelastungen zu vermeiden.

Dieser Wert ist beim 24V-Betrieb auf ca. **25V** einzustellen.

beträgt die Versorgungsspannung 36V, so sollten hier **37,5V** eingetragen werden.

Ab der Software V7.30 werden aus diesen Spannungen (25V bzw. 37,5V) andere Grenzwerte intern berechnet.

W INT0_way

Die Weg- und Geschwindigkeitsmessung erfolgt über das Auslösen des Interrupt INT0 durch eine Unterbrechung des Strahlengangs in der

Lichtschränke. Der hier angegebene Wert gibt die Auflösung des Weges pro INT0-Ereignis an. In diesem Fall wird ein Interrupt nach jeweils 1,13cm ausgelöst.

Im zweiten Schritt kann der km-Zähler zurückgesetzt oder auf einen wählbaren Wert eingestellt werden.

P Restore all Parameters

Alle Parameter, die hier aufgelistet werden, werden auf voreingestellte Werte zurückgesetzt. Alle eigenen Veränderungen müssen erneut durchgeführt werden.

ESC Exit Parameter Mode

Der Eingabemodus muss mit der ESC-Taste beendet werden.

12.3.2. Übersicht der veränderbaren Parameter ab V7.35:

Preset Parameters for Runner V7.29 – V7.xx Update V7.37				
Change Parameters by using a PS/2 keyboard				
Key	Parameter	preset	new	
F1	A_e	15		Faktor für Neigungssensor (phi)
F2	B_e	15		Faktor für Gyrosensor (phi_rate)
F3	C_e	20		Faktor für Geschwindigkeitszuwachs bei Neigung
F4	D_e	10		Faktor für PID-Integralanteil
F5	Ref_speed_a_e Ref_speed_b_e	20 20		Referenzgeschwindigkeiten für dynamische Balanceregulierung
F6	Filter_komp_e	0.98		Parameter für Komplementärfilter 0,99~1Hz, 0,95~5Hz
F7	Filter_MPU6050_e	5		Filterparameter für MPU6050 nur 1~180Hz, ... 6~5Hz
F8	Lenk_faktor_e Filter_lenkung_e	5 0.05		-Empfindlichkeit des Lenkers (r - l) -Tiefpassfilter für Lenkung (r - l) 0.1 ~ 10Hz, 0,05 ~ 5Hz
F9	Gier_faktor_e Filter_gier_e	5 0.1		Empfindlichkeit des Gyro_sensors zur Gierstabilisierung Tiefpassfilter für Gier um z-Achse 0.1 ~ 10Hz, 0,05 ~ 5Hz
F10	PWM_mode_e	1		1 2 und 4 3 und 5 15.6kHz 10,4kHz 7,8kHz
F11	Pwm_max_e	500		Maximaler zulässiger PWM-Wert
F12	Pwm_faktor_max_e	350		Faktor für Sensorwerte →PWM- Wert
B	U_batt_faktor_e	0.454		Spannungsteiler für 36V-Betrieb, Umrechnungsfaktor

D	Display_mode_e	7		Auswahl verschiedener Modi für das Display
K	Batteriekapazität	14		Eingabe der Batteriekapazität
O	Set all offset values			Runner ruhig stellen, dann Offsetwerte aus EERAM einlesen
R	Runner_typ_e	25		Runner_typ (unbedeutend)
T	Runnerboard_Test	100		-Start des Testmodus ohne Einschalten des Relais. -Eingabe des PWM_max für Tests
U	U_Batt0_e	37.5 / 25		Batterie-Normalspannung
W	INT0_way	1.13		-Wegstrecke in cm für einen Interrupt an INT0 für die Geschwindigkeitsmessung -km-Zähler einstellen (z.B. =0.0)
P	Restore all Parameters ==>			My_parameters intern ab V7.08
ESC	Exit keyboard routine			Parametermode beenden
How to adapt your Runner.				
1.	Use a PS/2 adapter to connect a PS/2 keyboard to the serial connector			
2.	Press "U" to set voltage to switch the relais			
3.	Press "D" to set Display_mode (LCD_mode)			
	Press „P“ to reset all parameters			

⇒ **Use ESC to exit. If you do not use ESC, all Parameters are restored.**

12.3.3. Ändern der Parameter

Wichtig:

Bei der Eingabe falscher Werte kann das Fahrzeug unvorhersehbar reagieren.

Mit einer PS/2 Tastatur ist im Parametermodus die Änderung aller Parameter möglich. Ob eine Tastatur angeschlossen ist, wird direkt nach Ablauf der Startmelodie geprüft. Während dieser Prüfung läuft eine Sternreihe ***** ... durch das Display. Das Drücken einer Taste in diesem Zeitfenster aktiviert den Parametermodus.

Wurde die Tastatur erkannt, so dienen die Funktionstasten **F1 – F12** und einige Buchstaben gemäß der vorhergehenden Tabelle der Auswahl der zu ändernden Parameter. Nach der Anzeige des aktuellen, alten Wertes, kann ein neuer Wert eingegeben werden. Die Eingabe wird mit Return beendet und nur dann übernommen, wenn sie mit „Y“ bestätigt wird.

Zum erfolgreichen Verlassen dieser Prozedur muss die ESC-Taste benutzt werden. Wird der Parametermodus durch das Ausschalten des Gerätes oder einen Reset unvorhergesehen beendet, so erfolgt beim nächsten Start ein zurücksetzen aller Parameter auf die Voreinstellungen.

12.4. Testmodus

Ab Softwareversion 7.20 ist im Programm ein umfangreicher Testmodus enthalten. Dieser dient dazu elementare Funktionen vor dem späteren Betrieb zu prüfen. Dieser Testmodus sollte nach dem Bestücken der Platine, oder auch bei späteren Problemen zur Diagnose benutzt werden.

Der Anschluss einer Tastatur ist erforderlich, falls beim Betrieb mit 24V der Parameter U auf 25V oder 37.5V eingestellt werden muss.

12.4.1. Aktivierung des Testmodus

JP7/1 bedeutet Pin1 am Jumper 7 auf der Platine.

Der Testmodus wird aktiviert, indem vor dem Einschalten der Stromversorgung auf der Platine **JP7/1 mit GND (JP7.3)** verbunden wird. JP7/2-3 bleibt davon unberührt. **GND** steht auch an anderen Jumpers, z.B. JP1/3 oder auch JP_ISP/10 zur Verfügung. In diesem Fall erfolgt der Start des Testmodus mit dem Einschalten des Relais.

Der Testmodus kann auch aufgerufen werden, wenn im Parametermodus die Taste „**T**“ gedrückt wird. In diesem Fall erfolgt der Start des Testmodus ohne das Einschalten des Relais.

Wurden einzelne Tests ausgeblendet, so können diese wieder aktiviert werden, wenn beim Start des Testmodus die entsprechende Frage durch das Drücken des Fußschalters (Switch) bis zum Erscheinen des entsprechenden Textes im Display gedrückt wird.

Fragen im Display werden grundsätzlich durch das Drücken eines Switch mit „Ja“ beantwortet.

12.4.2. Automatischer Testablauf

Die Tests starten nach der Aktivierung automatisch und können einzeln ausgeblendet werden.

- a) **LED-Test** Prüft die Funktion der LEDs

Im LED_Test blinken die LEDs der Hauptplatine im Wechsel für die Dauer von fünf Sekunden.

Bei allen Tests erscheint im Display die Frage: „**Test ausblenden?**“

Das Drücken der Fußtaste für ca. eine Sekunde blendet diesen Test bei erneuten Testläufen aus. Dies kann wieder rückgängig gemacht werden.

Wird die erste Frage im Display durch das Drücken eines Fußschalters bestätigt, so werden alle Tests erneut aktiviert.

b) **ADC-U-Batterietest** Anzeigen der aktuellen Versorgungsspannung

c) **MPU-6050 Sensor-Test** Prüfung der MPU-6050 Sensors

Als Ergebnis erscheint die I2C-Adresse des Sensors (208).

d) **Lenkerpoti-Test** Funktion des Lenkungs-Potentiometers

Der Offset-Wert für die Lenkerstellung sollte zwischen 462 und 562 liegen.

Größere Abweichungen werden angezeigt.

Dann den Lenkers nach rechts und anschließend nach links bewegen. Im Fehlerfall sind die Kabel am Poti zu tauschen. Abschließend muss der Lenker wieder in die Offset-Stellung gebracht werden, um den Test zu beenden.

e) **Gyro_Funktionstest** Anzeige von Gyro-Offset und Gyro_Wert

Kippt man das Fahrzeug nach vorne, so muss der Gyro_Wert **negativ** werden!

f) **Phi_Funktionstest** Anzeige von Phi_Offset und Phi_Wert

Phi_Offset, die montierte Sensorneigung, muss bei $45^\circ \pm 3^\circ$ liegen.

Ist dies nicht der Fall, so wartet das Programm bis dieser Winkel eingestellt ist.

Kippt man das Fahrzeug nach vorne, so muss der Phi_Wert **negativ** werden!

Wird ein Winkel von ± 20 Grad überschritten, so schaltet der Runner sich aus.

g) **Phi-PWM-Motor-Test** Funktionstest der PWM-Steuerung

Angezeigt werden der Neigungswinkel Phi, der Lenkwert und die PWM-Werte für den rechten und linken Motor. Das Neigen des Runners führt zu steigenden PWM-Werten und einem schwachen Antrieb durch die Motoren in die

entsprechende Richtung. Der PWM-Wert wird sicherheitshalber begrenzt. Drehen sich die Motoren in die falsche Richtung, so müssen die Anschlüsse umgepolt werden.

12.4.3. Beenden des Testmodus

Mein Runner besitzt keinen Ausschalter.

Möchte man einen solchen aber vorsehen, dann kann man einen Batterietrennschalter [auch Natoknochen genannt] einbauen.

Ein Schalter, der die Stromzufuhr zur Hauptplatine unterbricht, erfüllt ebenfalls diesen Zweck.

Wird im **Phi_Funktionstest** ein Winkel von ± 20 Grad überschritten, so schaltet der Runner sich aus.

Der Runner wird auch automatisch ausgeschaltet, wenn im **Phi-PWM-Motor-Test** eine der folgenden Bedingungen eintritt:

- a) Der Runner wird um mehr als 20° nach vorne oder hinten geneigt. Dies führt auch zum Ausschalten, wenn der Motor wegen eines Fehlers ungewollt aktiv wird.
- b) Wenn der Fußschalter länger als fünf Minuten nicht betätigt wird.

13. Rechtliches in Deutschland

13.1. Straßenfahrten und Zulassung

Aus dem fast alltäglichen Straßenbild kennen wir mittlerweile den großen Bruder, welcher mit einer Betriebserlaubnis und einer Haftpflichtversicherung ausgestattet am Straßenverkehr teilnehmen kann.

Auf der Suche nach den notwendigen Voraussetzungen stößt man an erster Stelle auf die Mobilitätshilfeverordnung, in deren Regelwerk die selbstbalancierenden Fahrzeuge eingeordnet wurden.

13.1.1. Die Mobilitätshilfeverordnung

Hier die wesentlichen Inhalte mit anschließenden Kommentaren.

§ 1 Anwendungsbereich, Grundsatz der Verwendung

0. Bauartbedingte Höchstgeschwindigkeit von nicht mehr als 20 km/h,

1. zweispuriges Kraftfahrzeug mit zwei parallel angeordneten Rädern mit integrierter elektronischer Balance- Antriebs-, Lenk- und Verzögerungstechnik,

2. eine Gesamtbreite von nicht mehr als 0,7 m,

3. eine Plattform als Standfläche für einen Fahrer,

4. eine lenkerähnliche Haltestange, über die der Fahrer durch Schwerpunktverlagerung die Beschleunigung oder Abbremsung sowie die Lenkung beeinflusst,

5. entspricht den Anforderungen der Richtlinie 72/245/EWG des Rates vom 20. Juni 1972 zur Angleichung der Rechtsvorschriften der Mitgliedstaaten über von Fahrzeugen verursachte Funkstörungen (elektromagnetische Verträglichkeit) (ABl. L 152 vom 6.7.1972, S. 15), die zuletzt durch die Richtlinie 2006/96/EG (ABl. L 363 vom 20.12.2006, S. 81) geändert worden ist, in der jeweils geltenden Fassung,

6. eine Anzeige für den Energievorrat.

§ 2 Anforderungen an das Inbetriebsetzen

1. einem genehmigten Typ entspricht oder eine Einzelgenehmigung besitzt sowie

2. ein gültiges Versicherungskennzeichen nach § 26 in Verbindung mit § 27 der Fahrzeug-Zulassungsverordnung führt

§ 4 Anforderung an die Verzögerungseinrichtung

1. das Fahrzeug bis zum Stillstand abbremsen kann und

2. mindestens einen Verzögerungswert von $3,5 \text{ m/s}^2$ erreicht.

§ 5 Anforderung an die lichttechnischen Einrichtungen

1. nach vorne wirkendem Scheinwerfer für weißes Licht (TA 23),

2. nach vorne wirkendem weißem Rückstrahler (TA 18),
3. an der Rückseite mit einer Schlussleuchte für rotes Licht (TA 14b),
4. an der Rückseite mit einem roten Rückstrahler (TA 18),
5. mit gelben Rückstrahlern nach beiden Seiten wirkend (TA 18).

§ 6 Anforderung an die Schalleinrichtung

Eine Mobilitätshilfe darf nur in Betrieb gesetzt werden, wenn sie mit einer Glocke ausgerüstet ist.

Kommentare:

§1.5. ... entspricht den Anforderungen der Richtlinie 72/245/EWG des Rates vom 20. Juni 1972 zur Angleichung der Rechtsvorschriften der Mitgliedstaaten über von Fahrzeugen verursachte Funkstörungen (elektromagnetische Verträglichkeit) (ABl. L 152 vom 6.7.1972, S. 15), die zuletzt durch die Richtlinie 2006/96/EG (ABl. L 363 vom 20.12.2006, S. 81) geändert worden ist, in der jeweils geltenden Fassung, Der Gesichtspunkt der elektromagnetischen Verträglichkeit (EMV) ist wohl der schwierigste und ggf. teuerste Aspekt in dieser Prozedur. Die aktive und passive EMV ist sicher sinnvoll, um die eigene Sicherheit die anderer Verkehrsteilnehmer zu gewährleisten.

Nach Aussage eines maßgeblichen Mitarbeiters des TÜV Rheinland (zuständig für die Prüfung des originalen Segway, eine Aussage, von der sich ein Mitarbeiter später distanziert hat) erfolgt die Prüfung in einem technischen Labor, welches vom TÜV benannt werden kann. Die Prüfung unter höchster Belastung der Elektronik und der Motoren in einer eigens für diesen Zweck zu erstellenden mechanischen Messeinrichtung kostet mindestens 1000.-€. Eine sehr wahrscheinliche Nachprüfung aufgrund der notwendigen Nachbesserungen kostet etwa 500.-€. In Rechnung gestellt werden dabei Ingenieursstunden sowie die Nutzungszeit des Messlabors.

§2.1. ... einem genehmigten Typ entspricht oder eine Einzelgenehmigung besitzt Eine Typgenehmigung (für den Fall einer Serienproduktion) kommt in diesem Fall eher nicht in Frage. Sie ist mit weit höheren Gebühren verbunden. Eine Einzelgenehmigung durch den TÜV erfolgt nach dem Erfüllen aller in der MobHV genannten Bedingungen ist im Regelfall für die Erteilung eines Versicherungskennzeichens erforderlich.

Auch hier fallen Gebühren an.

2. ... ein gültiges Versicherungskennzeichen nach § 26 in Verbindung mit § 27 der Fahrzeug-Zulassungsverordnung führt

Das Versicherungskennzeichen ist beim Fahren mit einer Geschwindigkeit von mehr als 6km/h erforderlich. Dies kann problemlos erworben werden, wenn eine allgemeine

Betriebserlaubnis vorliegt.

Es gibt aber auch mindestens eine Ausnahme.

Die Haftpflichtversicherung kostet zwischen 15.-€ (Leichtmofa) und 75.-€ (Moped).

Dies hängt von der Einstufung des Fahrzeuges ab. Die Preise schwanken zusätzlich in Abhängigkeit von der gewählten Versicherungsgesellschaft.

§ 4 Verzögerungseinrichtung

2. ... mindestens einen Verzögerungswert von $3,5 \text{ m/s}^2$ erreicht.

Diese Bedingung muss ebenfalls erfüllt werden.

Auch ohne Betriebserlaubnis erscheint das Einhalten dieser Verzögerung aus Sicherheitsgründen sinnvoll.

Eine eigene Prüfung ist recht einfach. Da es sich beim Bremsvorgang um eine (negative) beschleunigte Bewegung handelt, gelten hier die entsprechenden physikalischen Gesetzmäßigkeiten wie: $s = 0,5 \cdot a \cdot t^2$ und $v = a \cdot t$, wobei $a = 3,5 \text{ m/s}^2$ der oben genannte Verzögerungswert ist.

An einem einfachen Beispiel soll nun die Prüfungsmöglichkeit dargestellt werden.

Eine Geschwindigkeit von $12,6 \text{ km/h}$ entspricht $v = 3,5 \text{ m/s}$. Zum Erreichen dieser Geschwindigkeit muss das Fahrzeug genau eine Sekunde beschleunigen (oder bremsen).

In einer Sekunde legt das Fahrzeug einen Weg $s = 0,5 \cdot 3,5 \text{ m/s}^2 \cdot (1\text{s})^2 = 1,75 \text{ m}$ zurück. Somit muss das Fahrzeug nach dieser Strecke zum Stillstand kommen.

§ 5 ... Anforderung an die lichttechnischen Einrichtungen

Diese Anforderungen werden erfüllt, wenn entsprechende Beleuchtungsteile von Fahrrädern oder auch elektrischen Rollstühlen verwendet werden. Bruchsicherheit, Blendungsfreiheit und Mindesthelligkeit sind erforderlich. Dies wird durch eine auf den Beleuchtungsteilen eingeprägte Prüf- und Zulassungsnummer sichergestellt.

13.1.2. Allgemeine Betriebserlaubnis

In der Regel wird beim Abschluss einer Haftpflichtversicherung die Vorlage einer Betriebserlaubnis verlangt. Im nächsten Kapitel gehe ich näher darauf ein.

Das Erlangen einer allgemeinen Betriebserlaubnis oder einer Einzelbetriebserlaubnis ist nur durch die Vorstellung des Fahrzeuges beim TÜV oder einer anderen vergleichbaren Prüforganisation möglich.

Eine Prüfung bei einer der vielen Außenstellen ist dabei nicht ausreichend. Somit wird ein Besuch einer überregionalen Niederlassung notwendig.

Meine Recherchen ergaben:

Geprüft wird gemäß den Angaben der Mobilitätshilfeverordnung.

-Prüfung der Elektromagnetischen Verträglichkeit

Diese Prüfung des Fahrzeuges auf aktive und passive Störfestigkeit in einem speziellen Strahlungslabor ist recht aufwändig und kostenintensiv. Es können 1000.-€ bis 1500.-€ veranschlagt werden.

-Ausstattungsprüfung

Dies erfolgt gemäß Mobilitätshilfeverordnung. Diese Bedingungen können problemlos erfüllt werden.

-Fahrtechnische Prüfung

Hier erfolgt eine technische Sichtprüfung bei der im Ermessen des Prüfers die Eignung und Sicherheit beurteilt wird. Belastbarkeit, Festigkeit bei Überlastung und Fahrfehlern, wie z.B. bei Hindernissen, sind Inhalt des ersten Teiles dieser Prüfung.

Die Fahrprüfung selbst muss beim Beschleunigungs- und Bremsvorgang nach der Beurteilung des Prüfers bestanden werden.

Das Einhalten der angegebenen Höchstgeschwindigkeit sowie der Bremsverzögerung von $2,5\text{m/s}^2$ werden geprüft. Zur genannten Bremsverzögerung erhielt ich allerdings auch die mündliche Auskunft, dass dies für mein Fahrzeug nicht unbedingt gelten müsse.

Für diese Prüfung mit dem Ziel der allgemeinen Betriebserlaubnis, welche dann auch für baugleiche Fahrzeuge gilt, wurden Kosten von mindestens 2000.-€ genannt.

Eine Einzelerlaubnis sei wohl etwas preiswerter.

Fazit: In Anbetracht der Kosten habe ich auf diese hier beschriebene Prozedur verzichtet.

Die Folgen sind überschaubar. Fehlt die Betriebserlaubnis, so liegt eine Ordnungswidrigkeit vor.

Hier kann eine Strafe bis 50.-€ und ggf. ein Punkt in Flensburg anfallen, alles Strafen, die ohne größere Belastung zu verkraften sind.

13.1.3. Haftpflichtversicherung

Warum überhaupt versichern?

Bei der Teilnahme am Straßenverkehr, was ich ja mit meinem Fahrzeug mit einer Geschwindigkeit bis 20km/h erwogen habe, muss gemäß der deutschen Gesetzeslage eine Haftpflichtversicherung bestehen.

Besteht keine derartige Versicherung so liegt ein Vergehen vor, welches mit empfindlichen Geldstrafen, der Beschlagnahme des Fahrzeuges und ggf. auch einer Gefängnisstrafe geahndet werden kann.

Bedenkt man auch das Haftungsrisiko des Fahrers, so erscheint mir diese Versicherung als unbedingt erforderlich.

„Keine Haftpflichtversicherung ohne gültige Betriebserlaubnis.“

Diesen Satz hörte ich von drei Versicherungen bei meinen ersten Nachfragen.

Danke Thomas Knauf und seinem Erfahrungsbericht konnte ich auf zwei Gerichtsurteile zurückgreifen, die sich auf diese Problematik beziehen.

Höchststrichterlich wurde darin festgestellt, dass die Versicherer auch ohne das Bestehen einer Betriebserlaubnis zum Abschluss einer Haftpflichtversicherung verpflichtet sind. Siehe:

<http://www.thomasknauf.de/ginger/ginger.htm>

So ging ich nach einigen Wochen dieses Problem erneut an. Entsprechend vorbereitet mit einer eigenen technischen Fahrzeugbeschreibung gab ich meinen Versicherungsantrag bei drei Gesellschaften ab.

Nach zwei Wochen erhielt ich ein Angebot mit einer Jahresprämie von 220.-€. Das war stattlich.

Die zweite Gesellschaft erörterte die Problematik telefonisch und gab mir drei Tage später grünes Licht. Am folgenden Tag flatterte die dritte Zusage ins Haus. Mit einer Jahresprämie von 63.-€ war ich höchst zufrieden.

14. Updates

14.1. Hauptplatine

14.1.1. Runner AT32 36V 2.5 digital

Diese Überarbeitung der Platine (ab März 2015) beinhaltet mehrere Veränderungen.

Alle Bauelemente, die zur Verwendung der analogen Sensoren notwendig waren, wurden entfernt.

Ein Baustein CD4001N und zwei Transistoren als Inverter wurden hinzugefügt. Diese Logic ersetzt entsprechende Bausteine auf den Platinen der H-Brücken. Der 5V-Spannungsregler sowie das IC CD4001N sind bei den H-Brücken nicht mehr erforderlich. Entsprechende Verbindungen sind bei älteren H-Brücken von den Anschlüssen der Platine zu den Eingängen der Halbbrückentreiber durch Drahtbrücken herzustellen. (Z.Z. noch nicht empfehlenswert)

Die Ausgänge des CD4001N-Bausteines wurden an zusätzlichen Steckanschlüssen JP15 und JP16 herausgeführt.

Die PIN-Belegung entspricht der Belegung bei den bereits genannten BTS7960B H-Brücken, so dass die Verbindungen mit Steckverbindern und Flachbandkabel sehr einfach hergestellt werden können.

JP20 wurde hinzugefügt. An dieser Steckverbindung stehen verschiedene Signale zur Verfügung.

In Verbindung mit den benachbarten JP15 und JP16 kann ein 36 bis 40-poliges Flachbandkabel mit Stecker, wie es früher bei IDE-Festplatten Verwendung fand, als Verbindung zu H-Brücken und sonstigen Elementen verwendet werden.

Pinbelegung Hauptplatine M_2.5+

JP1

1	○	ADC6
2	○	ADC7 Strom
3	○	GND
4	○	VCC-5V

JP5

1	○	GND
2	○	Lenkung
3	○	VCC

JP7

1	○	Testmodus
2	○	BTS-Modus
3	○	GND

JP2

1	○	GND
2	○	Piezo

JP3

1	○	GND
2	○	Switch

JP4

1	○	Relais
2	○	Relais

JP9

1	○	PWM_L
	○	PWM_DIR_L
	○	VCC
4	○	GND

JP8

1	○	PWM_R
	○	PWM_DIR_R
	○	VCC
4	○	GND

JP20

ADC_7	16	○	○	GND
ADC_6		○	○	GND
VCC		○	○	GND
VCC		○	○	GND
Lenkung		○	○	GND
Piezo		○	○	GND
Switch		○	○	GND
Relais		○	○	GND

JP16

GND	8	○	○	VCC
frei		○	○	frei
VCC		○	○	VCC
PWM/DIR		○	○	1 DIR/PWM

JP15

GND	8	○	○	VCC
frei		○	○	frei
VCC		○	○	VCC
PWM/DIR		○	○	1 DIR/PWM

Hinweis: Je nach verwendeter H-Brücke sollte entweder JP8 und JP9 oder JP15 und JP16 bestückt werden.

14.1.2. Runner AT32 36V 2.6 digital

Die Vorgängerversion der Hauptplatine wies leider einen kleinen Planungsfehler auf. Dieser Fehler wurde beseitigt und gleichzeitig die Anschlussbelegung des JP-LCD so geändert, dass eine Verbindung zum LCD über Flachbandkabel möglich ist, ohne einzelne Kabel paarweise zu tauschen. Dies war auf die frühere Verwendung eines anderen LCD zurück zu führen.

Die Beschreibung der PINs entspricht der Anordnung auf der Platine.

Nun kann eine Stiftleiste an der Unterseite des LCD eingelötet werden und die Verbindung durch ein 16-pol. Flachbandkabel direkt realisiert werden.

Die Anordnung im Schaltplan wurde gespiegelt, um eine einfachere Leitungsführung zu ermöglichen. Der LCD-Stecker wird normal auf der Oberseite der Hauptplatine befestigt.



Das Flachbandkabel kann in dieser Form am LCD befestigt werden.

JP-LCD

DB6 14	○	○	13 DB7
DB4 12	○	○	11 DB5
GND 10	○	○	9 GND
GND 8	○	○	7 GND
GND 6	○	○	5 E
LCD 4	○	○	3 RS
GND 2	○	○	1 VCC

JP10 (Power)

3	○	VCC - 5V
2	○	GND
1	○	24V – 44V

JP6 (Sensor)

1	○	ADC4=PA4
2	○	ADC3=PA3
3	○	ADC2=PA2=SDA
4	○	ADC1=PA1=SCL
5	○	GND
6	○	VCC
7	○	VCC
8	○	GND
9	○	ADC1=PA1=SCL
10	○	ADC2=PA2=SDA
11	○	ADC0=PA0
12	○	PB6
13	○	PB7

Die PIN-Belegung des Jumper JP6 ergibt sich noch aus der vorherigen Verwendung für analoge Sensoren.

Bei der Verwendung des digitalen Sensors MPU6050 oder anderer Sensoren mit dem I2C-Bus sind nur die PINs 3,4,5,6 bzw. die PINs 7,8,9,10 zu verwenden.

14.1.3. Runner M_2.7 (ab Ende 2016)

Die neue Bezeichnung für Hauptplatinen lautet z.B. M_2.7

Die JP8 und JP9 wurden entfernt, da die Anschlüsse für die H-Brücken am Flachbandkabelanschluss zur Verfügung stehen.

Der JP20 wurde gekürzt, da die Vielzahl an PINs nicht erforderlich ist. In dieser neuen Form können Stiftleisten oder ein 34-poliger Wannenstecker, bei dem überflüssige Stifte gezogen wurden, für die JP15, JP16 und JP20 eingelötet werden. In den 34-poligen Wannenstecker oder ersatzweise einfache Pinheads passt auch das Flachbandkabel, welches in alten PCs als Anschlusskabel für ein Diskettenlaufwerk genutzt wurde.

JP6, die Anschlussstifte für die Sensoren, wurde für den Anschluss des digitalen Sensors MPU-6050 auf vier Pole reduziert.

Für Q3 und Q4 sind Transistoren mit der Anschlussreihenfolge CBE (z.B. BC338) erforderlich. Durch das Hinzufügen eines vierten Anschlusses für den Kollektor können nun, leicht versetzt, auch Transistoren mit der Anschlussreihenfolge BEC verwendet werden.

Durch das Verbinden (verlöten) zweier Lötunkte neben dem JP6 wird die Betriebsspannung gewählt, die an JP6 für den Sensor zur Verfügung steht. Hier müssen aktuell 5V ausgewählt werden.

Das Lochraster für Zusatzteile wurde entfernt. In diesem Bereich wurde ein Funkmodul RFM69CW vorbereitend eingeplant. Eine Funktionsfähigkeit ist noch nicht gegeben.

Die vier Befestigungslöcher wurden mit einem blanken Masse-Anschluss ausgestattet. Dies erleichtert die Herstellung einer Masseverbindung zum Metall-Chassis, wodurch die Störfestigkeit und Abschirmung der Hauptplatine verbessert wird.

Einige Bauteile wurden verschoben. Am JP-LCD kann nun auch ein 14-poliger Wannenstecker eingelötet werden.

Pinbelegung Hauptplatine M_2.7

JP1

1	○	ADC6
2	○	ADC7 Strom
3	○	GND
4	○	VCC-5V

JP5

1	○	GND
2	○	Lenkung
3	○	VCC

JP7

1	○	Testmodus
2	○	BTS-Modus
3	○	GND

JP2

1	○	GND
2	○	Piezo

JP3

1	○	GND
2	○	Switch

JP4

1	○	Relais
2	○	Relais

JP20

ADC_7	14	○	○	GND
ADC_6		○	○	GND
VCC		○	○	GND
Lenkung		○	○	GND
Piezo		○	○	GND
Switch		○	○	GND
Relais		○	○	1 GND

JP10 (Power)

3	○	(VCC - 5V)
2	○	GND
1	○	24V – 42V

JP15 / JP16

GND	8	○	○	VCC
frei		○	○	frei
VCC		○	○	VCC
PWM/DIR		○	○	1 DIR/PWM

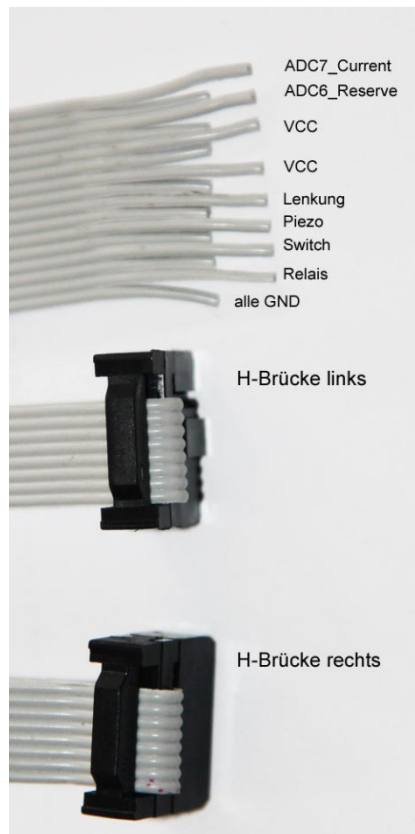
JP-LCD

DB6	14	○	○	13 DB7
DB4	12	○	○	11 DB5
GND	10	○	○	9 GND
GND	8	○	○	7 GND
GND	6	○	○	5 E
LCD	4	○	○	3 RS
GND	2	○	○	1 VCC

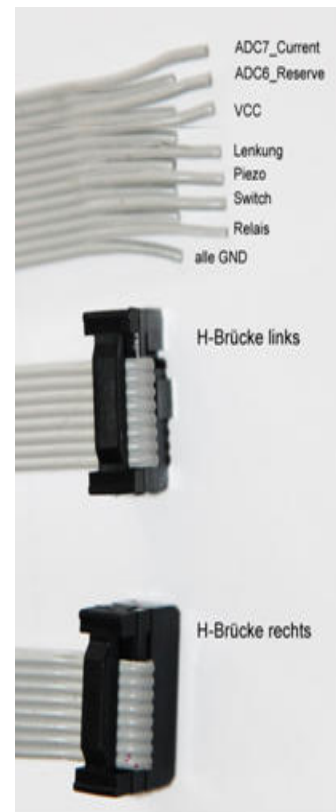
JP6 (Sensor)

1	○	ADC2=PA2=SDA
2	○	ADC1=PA1=SCL
3	○	GND
4	○	VCC

Die PIN-Belegung des Jumper JP6 wurde für digitale Sensoren reduziert. An einen I2C-Bus können mehrere Sensoren gleichzeitig angeschlossen werden.



FL_3 für – M_2.6



FL_3 für M_2.7+

14.2. H-Brücken

14.2.1. Runner Mot H 2.8

Die etwas geänderte Form der Platine erlaubt den leichteren Anschluss der Anschlusskabel, wenn die auf einer Platine enthaltenen H-Brücken getrennt angeordnet werden. Die Stromversorgung, 24V – 36V, wird seitlich in der Platinenmitte an Schraubklemmen angeschlossen. Die Klemmen für den Motoranschluss liegen ebenfalls seitlich und sind aus Stabilitätsgründen doppelt ausgeführt worden. Der Baustein CD4001 wurde entfernt, da die Aufbereitung der Signale auf der Hauptplatine erfolgt.

Die Verbindung mit der Hauptplatine würde für das bereits genannte Flachbandkabel optimiert und entspricht gleichzeitig der PIN-Belegung der BTS-Module.

Neu ist der eingebaute PWM-Tester. Eine rot-grüne LED liegt mit einem Widerstand zwischen den Motoranschlüssen und leuchtet je nach Polung und Stärke des PWM-Signals in unterschiedlichen Farben verschieden hell.

14.2.2. BTN / BTS7960B / 7970 - Module

In verschiedenen Online-Angeboten findet man H-Brücken als fertige Module mit den Halbbrücken BTS7960B und BTS7970B für eine Betriebsspannung bis max. 28 Volt. Diese Module können im normalen PWM-Modus 1 an JP15 und JP16 angeschlossen werden.

Wichtiger Hinweis für die Verwendung von BTS-H-Brücken:

BTS-Modifikation

Bei den Halbbrücken BTS7960, die bei dem genannten Modul zwei Mal verwendet werden, kann eine Verzögerungszeit für das Einschalten der Halbbrücken im PWM-Betrieb durch die Wahl eines Widerstandes bestimmt werden.

Diese Verzögerung des Einschaltzeitpunktes führt dazu, dass bereits kurze PWM-Impulse noch weiter verkürzt werden, so dass bereits im Testmodus bei einem PWM-Wert 50 (von max. 500) die KFZ-Lampen nicht leuchten.

Später angeschlossene Motoren reagieren je nach Typ auch sehr schwach.

Im Normalmodus setzen die Motoren erst bei etwas größerer Neigung des Fahrzeuges ein.

Notwendige Modifikationen werden an anderer Stelle beschrieben.
Abhilfe schafft das Austauschen entsprechender Widerstände. In der Reihe der Widerstände sind dies von links der erste und der fünfte SMD-Widerstand. Es hat sich als günstig erwiesen hier null Ohm zu verwenden, der kleinste Wert gemäß Datenblatt. Das Auslöten des bisherigen SMD-Widerstandes, ggf. mit zwei LötKolben, kann auch durch das Überbrücken mit einer Drahtbrücke mit einer Drahtbrücke ersetzt werden.

Im Testmodus wird dann das vorgesehene schwache Leuchten der zur Probe angeschlossenen KFZ-Lampen erreicht.

Der frühere PWM-Modus 2 für BTS-Module wird ab Software V7.36 nicht mehr unterstützt.

14.3. Software

Beginnend mit dem Update V7.xx wurde das System auf den digitalen Sensor MPU6050 umgestellt.

Dieser Sensor liefert Messwerte eines x-y-z-Beschleunigungssensors und eines x-y-z-Gyrosensors in einem einzigen Chip, häufig als 6 Freiheitsgrade bezeichnet.

Die Datenübertragung erfolgt in digitaler Form über einen I2C-Bus.

Neben der Tatsache, dass analoge Sensoren vom Markt verschwinden, war mit entscheidend, dass dieser Sensor MPU6050 sehr preiswert erworben werden kann.

Die notwendigen Änderungen der bisherigen Platine vereinfachen den Nachbau. Diese Veränderungen sind an anderer Stelle im Text beschrieben.

14.3.1. Update V7.30

Die Verwendung des optimierten Komplementärfilters ab V7.27 macht auch den Tiefpass für die Gyro-Werte überflüssig.

Mit der Taste **F6** kann nun eine Einstellung des Tiefpasses, der im Sensor MPU6050 integriert ist, vorgenommen werden. Die Grenzfrequenzen ergeben sich aus der Parametertabelle.

Werden alte Filtereinstellungen erkannt, so erfolgt in der Regel ein automatischer Aufruf des Parametermodus. Mit der Taste **P** müssen dann alle Parameter aktualisiert werden.

Eine Prüfung der Parameter A_e (**F1**) und B_e (**F2**) ist erforderlich (siehe Tabelle).

Beim Neustart des Runners werden vorherige **Maximalwerte** angezeigt.

Für die Reihenfolge in der 2. Zeile des Displays gilt:

PWM	Strom	Geschwindigkeit	Neigungswinkel
------------	--------------	------------------------	-----------------------

Für den Einsatz verschiedener Batteriekapazitäten ist nun eine Anpassung der Ausgangskapazität möglich.

Dazu wird im Parametermode die Taste **K** verwendet. Voreingestellt sind **14Ah**.

Zu beachten ist, dass grundsätzlich nicht die vollständige Batteriekapazität genutzt werden kann. Auch ein Kapazitätsverlust durch Alterung ist zu berücksichtigen.

Während vor diesem Update eine Mindestspannung einzugeben war, bei deren Erreichen während des Ladevorganges der internen Kondensatoren das Hauptrelais eingeschaltet wurde, wird diese Mindestspannung nun aus der eingestellten Batteriespannung errechnet.

Diese Batteriespannung wird weiterhin mit Hilfe der Taste **U** eingegeben.

Beim Betrieb mit 3 x 12V–Akkus sind **37.5V** einzugeben.

Beim Betrieb mit 2 x 12V–Akkus sind **25.0V** einzugeben.

Das Erkennen einer **vollgeladenen** Batterie erfolgt ebenfalls in Abhängigkeit von diesem Spannungswert. Die mit Hilfe der Taste **K** vorgegebene Batteriekapazität wird intern übernommen, wenn beim Einschalten des Runners die Batteriespannung über **41,25V (27,5V)** liegt. Dies ist in der Regel nur nach einem vollständigen Ladezyklus der Fall.

14.3.2. Update V7.33

Der Sensor muss mit einer Schrägstellung von **45°** montiert werden.

Um diesen Winkel ohne Winkelmesser einzustellen, wird dieser Winkel nun direkt durch den Sensor gemessen. Für diese absolute Winkelmessung im **Testmodus** ist es erforderlich, dass die Plattform waagrecht steht. Der Winkel mit der Bezeichnung Phi_0 wird im Display angezeigt. Bei einer Abweichung von mehr als 2° erscheint die Meldung * **Phi_0 Error!** *.

Nach einem Neustart wird der eingestellte Winkel angezeigt.

14.3.3. Update V7.35

Parameter D des Integralanteiles für PID-Filter in Parameterliste aufgenommen.

Menü und Tasten für Parameteränderungen neu geordnet.

Batteriefaktor angepasst für 3 Nachkommastellen.

Ausschalten bei Seitenlage >70°

14.3.4. Update V7.36

- PWM-Mode 2 für BTS7960 aus V7.09 wurde entfernt.
- Neuer Parameter PWM_mode (1 ... 3) hinzugefügt. Taste F10
Eine niedrigere PWM-Frequenz reduziert die Wärmeentwicklung der H-Brücken.
Dies ist insbesondere bei den BTS-H-Brücken zu empfehlen.
Leider sind die PWM-Frequenzen in Mode 2 und Mode 3 hörbar.

PWM_mode=1	→ $f_{\text{PWM}} = 15,6\text{kHz}$
PWM_mode=2,4	→ $f_{\text{PWM}} = 10,4\text{kHz}$
PWM_mode=3,5	→ $f_{\text{PWM}} = 7,8\text{kHz}$

- Offset-Toleranz für Acz, die teilweise nach dem Einschalten zum ERROR=16 geführt hat, wurde wegen des unterschiedlich starken Rauschens verschiedener Sensors erhöht.
- Eine frühere Erkennung der Seitenlage festgelegt.
- Wartezeit für U_min auf 10s festgelegt.
- Parameter U_batt0_e = 25 festgelegt. Ggf. mit Tastatur auf 37.5 ändern.
- Winkелеinstellung 45° im Testmode vereinfacht.

14.3.5. Update V7.37

- PWM-Mode = 1 automatisch für Testmode festgelegt,
 - Bug bei Pwm_max und Pwm_faktor_max beseitigt, EERAM-Werte intern nach
Keyboardeinstellung ebenfalls angepasst (2x erforderlich)
 - Fehlender Sensor oder Sensorerror wird beim Start angezeigt
 - Init I2C-Bus und Init MPU6050 zusammengefasst
 - 10ms-Timer2 auch für Test_modus verwendet. Wird im Testmode automatisch
angepasst.
 - Speichern und Anzeige von Maximalwerten beim Start entfernt
-
- ⇒ Zahleneingabe für A,B,D um den Faktor 100 vergrößert !
 - ⇒ Siehe Parametertabelle, wurde angepasst !

Index

Allgemeine Betriebserlaubnis	141	MY1020.....	22
Auslastung der CPU	34	Normalmodus.....	118
Ausschalter	137	Notausschalter	88
Balance-Lenk-Algorithmus.....	69	Offsetwerte.....	118
Bauanleitung.....	100	Parameter	73, 127, 132
Beschleunigungssensor.....	30	Parameter ändern	134
Betriebsanleitung	124	Parametermodus.....	118, 125, 127, 134, 135
Bleigel-Akku.....	26	Peukert-Effekt	91
Einschaltprozedur	125	Pinbelegung	145, 149
Elektronik	106	Plattform-Gehäuse	102
Fehlernummern	126	Ports.....	33
Funktionsprüfung und Modi	118	Programmkonzept.....	61
Fußschalter... 38, 88, 115, 117, 118, 125, 126, 127, 137		PS/2-Tastatur	112
Gesamtplan	56	PWM-Modi	43
Gierrotation	31	PWM-Steuerung.....	43
Gyroskop	30	PWM-Tester	46
Haftpflichtversicherung	143	Räder	100
Hauptplatine.....	144	Räder und Antriebe	22
H-Brücken.....	151	Rechtliches in Deutschland	138
Hinweise zur Montage	117	Regelroutine.....	65
Hochstromverkabelung.....	58	Rollstuhlmotoren	24
Inbetriebnahme.....	115	Runner1	102
Initialisierung.....	126	Runner2	102
Kalman-Filter	81	Runner3	103
km-Zähler einstellen.....	133	Schaltplan der H-Brücke	49
Kurvenfahrt	83	Schaltplan Hauptplatine 2.6	39
Ladegerät.....	28	Sicherheit	87
Leistung	19	Sicherheit Verkehr.....	89
Lenkung	82, 104	Sicherheitsprüfungen	125
Lichtschranke.....	113	Signalstromverkabelung.....	60
Li-Ion-Akku.....	27	Software	114, 153
Mikrocontroller	33	Tastatur.....	112
Mindestleistung.....	20	Testmodus	117, 135
Mobilitätshilfeverordnung	138	Updates.....	114
Modifikation der BTS-H-Brücken	108	Updates Hardware	144
Motoren.....	100	Vorbereitungen.....	117
MPU-6050.....	71		