



DEPARTMENT FÜR INFORMATIK  
SYSTEMSOFTWARE UND VERTEILTE SYSTEME

# **Konstruktion und Aufbau eines autonomen, fehlertoleranten Rasenmäherroboters**

**Diplomarbeit**

30. Mai 2014

Michael Lübke  
Carl-Heinrich-Slevogt Str. 1  
26133 Oldenburg  
m.luebke@me.com

**Erstgutachter**  
**Zweitgutachter**

Prof. Dr.-Ing. Oliver Theel  
Dipl.-Inform. Eike Möhlmann

---

# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>1</b>
<b>2. Bestehende Rasenmäherroboter</b>	<b>3</b>
2.1. Husqvarna Automower . . . . .	3
2.1.1. Navigation . . . . .	3
2.1.2. Mähwerk . . . . .	4
2.2. Robomow . . . . .	5
2.2.1. Navigation . . . . .	5
2.2.2. Mähwerk . . . . .	6
2.3. Zucchetti . . . . .	6
2.3.1. Navigation und Fahren . . . . .	7
2.3.2. Mähwerk . . . . .	7
2.4. Bosch Indego . . . . .	7
2.4.1. Navigation und Fahren . . . . .	7
2.4.2. Mähwerk . . . . .	8
2.5. weitere Hersteller . . . . .	8
2.6. Zusammenfassung . . . . .	8
<b>3. Grundlagen und verwendete Techniken</b>	<b>9</b>
3.1. Navigation . . . . .	9
3.1.1. Inertialnavigation . . . . .	9
3.1.2. Satellitennavigation . . . . .	10
3.1.3. Induktive Begrenzungsschleife . . . . .	10
3.1.3.1. Sender . . . . .	11
3.1.3.2. Empfänger oder Sensor . . . . .	11
3.1.4. Odometrie . . . . .	12
3.1.5. Ultraschallsensoren . . . . .	12

3.2. Aktorik . . . . .	13
3.2.1. Gleichstrommotor . . . . .	13
3.2.2. Schrittmotor . . . . .	14
3.3. Steuerung . . . . .	15
3.3.1. Raspberry Pi . . . . .	15
3.3.2. Arduino . . . . .	16
3.3.3. I <sup>2</sup> C-Bus . . . . .	17
3.4. Mähetechniken . . . . .	18
3.5. Sicherheit . . . . .	18
<b>4. Aufbau des Roboters</b>	<b>21</b>
4.1. Chassis . . . . .	21
4.2. Antrieb . . . . .	23
4.3. Mähwerk . . . . .	24
4.4. Schaltplan . . . . .	26
4.4.1. Hauptschalter . . . . .	26
4.4.2. Not-AUS . . . . .	27
4.4.3. Mähwerk . . . . .	27
4.4.4. Spannungswandler . . . . .	29
4.4.5. I <sup>2</sup> C-Bus . . . . .	29
4.4.6. Schrittmotorsteuerung . . . . .	32
4.4.7. Taster und Signalleuchten . . . . .	32
4.5. Steuerung . . . . .	34
4.5.1. Hardware . . . . .	34
4.5.2. Sensorik . . . . .	34
4.5.3. Software . . . . .	37
4.5.3.1. Grundbefehle . . . . .	38
4.5.3.2. Manöver . . . . .	41
4.5.3.3. main()-Methode . . . . .	44
4.6. Gesamtaufbau des Roboters . . . . .	46
<b>5. Stücklisten und Kostenberechnung</b>	<b>47</b>
5.1. Chassis und Gehäuse . . . . .	47
5.2. Antrieb . . . . .	47
5.3. Mähwerk . . . . .	49

5.4. Energieversorgung . . . . .	49
5.5. Sensorik . . . . .	49
5.6. Steuerungselektronik . . . . .	50
5.7. nicht benötigte Bauteile . . . . .	50
5.8. Gesamtkosten . . . . .	50
<b>6. Zusammenfassung und Ausblick</b>	<b>53</b>
<b>Literaturverzeichnis</b>	<b>55</b>
<b>A. Anhang</b>	<b>57</b>
A.1. Bedienungsanleitung . . . . .	57
A.1.1. Benutzung des Roboters . . . . .	57
A.1.2. Aufladen der Akkus . . . . .	57
A.1.3. Wechsel der Mähmesser . . . . .	59
A.1.4. SD-Karte des Roboters neu erstellen . . . . .	60
A.1.5. Kompilieren neuer Programme . . . . .	61
A.1.6. Netzwerkzugang . . . . .	61



# Abbildungsverzeichnis

2.1. Bewegungsmuster der Automower[hana] . . . . .	4
2.2. Montage der Mähmesser[hana] . . . . .	4
2.3. Mähwerk eines Robomow [hanb] . . . . .	6
3.1. Schaltplan der Sendeschaltung . . . . .	11
3.2. Schaltplan der Empfängerschaltung . . . . .	12
3.3. prinzipielle Funktionsweise eines Gleichstrommotors [ele] . . . . .	14
3.4. Prinzipskizze eines Schrittmotors - Halbschritt [sch] . . . . .	14
4.1. Aufbau des Chassis des Rasenmäherroboters . . . . .	22
4.2. Chassis mit Gehäuse des Rasenmäherroboters . . . . .	23
4.3. Aufbau des Antriebs . . . . .	24
4.4. Schutzabdeckung des Mähwerks . . . . .	25
4.5. Gesamtschaltplan des Rasenmäherroboters . . . . .	26
4.6. Schaltplan des Hauptschalters . . . . .	27
4.7. Schaltplan der Not-AUS-Schaltung . . . . .	28
4.8. Schaltplan des Mähwerks . . . . .	29
4.9. Schaltplan des Spannungswandlers . . . . .	29
4.10. einfache I <sup>2</sup> C-Pegelanpassung [Sem] . . . . .	30
4.11. I <sup>2</sup> C-Pegelanpassung mit Schutzisolierung [Sem] . . . . .	31
4.12. Schaltplan der Schrittmotorsteuerung . . . . .	32
4.13. Schaltplan der Leuchtdiodenansteuerung . . . . .	33
4.14. START-Taster, Not-AUS-Kontrolle und Frontleiste . . . . .	33
4.15. zu erwartende Signalverläufe der Begrenzungsschleifensensoren . . . . .	35
4.16. Sensorlog der Begrenzungsschleife . . . . .	36
4.17. Anordnung der einzelnen Bauelemente im Roboter . . . . .	46

A.1. Entfernen der Akkus aus dem Roboter . . . . .	58
A.2. Mähwerk des Roboters . . . . .	59

# 1. Einleitung

Das Mähen des Rasens ist eine Aufgabe, die in einem privaten Garten, aber auch auf für Sport- und Freizeitwecke genutzten Rasenflächen immer wieder erledigt werden muss. In der Regel werden hierfür durch Elektro- oder Verbrennungsmotoren angetriebene, handgelenkte Rasenmäher verwendet, für kleinere Flächen auch handgetriebene Spindelmäher. Für große Flächen werden Rasenmäher verwendet, die ein Mitfahren der bedienenden Person erlauben und eine große Schnittbreite besitzen.

Seit 1995 sind im Handel verschiedene, autonome Rasenmäherroboter für private Haushalte verfügbar. Ihnen ist gemein, dass sie nur einfache oder einfachste Navigationsmechanismen bieten und damit nicht für alle Flächenzuschnitte optimale Ergebnisse liefern, d. h. jeden Teil der Rasenfläche nur genau ein Mal zu mähen. So wird z. B. bei den einfachsten verfügbaren Robotern die Fläche nur durch eine Markierung begrenzt: Die Roboter fahren geradeaus, bis sie auf ein Hindernis oder die Markierung treffen, drehen sich dann unter einem zufälligen Winkel und fahren weiter geradeaus. Bei großen oder verwinkelten Flächen kann so nicht sichergestellt werden, dass der Roboter die gesamte Rasenfläche in akzeptabler Zeit mäht, während andere Teile der Rasenfläche dagegen sehr oft gemäht werden.

Einige Rasenmäherroboter wurden auch mit intelligenteren Navigationsmechanismen angeboten, diese waren aber oft wegen der daraus resultierenden höheren Gerätepreise kommerziell nicht erfolgreich und sind größtenteils wieder vom Markt genommen worden.

Das Ziel dieser Arbeit ist es, einen autonomen, fehlertoleranten Rasenmäherroboter zu entwickeln, konstruieren und fertigen, der als Plattform zum Test alternativer Navigationsmechanismen dienen soll. Hierzu soll zunächst die einfache Navigation der kommerziellen Rasenmäherroboter imitiert wer-

den, d. h. der Roboter soll bei Erkennen des Randes der Rasenfläche wenden und dann wieder zurück in die Rasenfläche fahren. Zusätzlich soll bei Auftreten eines Hindernisses in der Fläche wie z. B. einer auf dem Rasen vergessenen Gießkanne oder einem heruntergefallenen Ast, an diesen „Fehlern“ angehalten und diese Stellen des Rasens ignoriert werden.

Neben den Anforderungen an das Arbeitsergebnis des Roboters gibt es weitere Anforderungen: So sollen die Kosten des Roboters gering sein, eine einfache Installation möglich sein und Sicherheitsanforderungen sollen beachtet werden. So soll z. B. das Mähwerk beim Anheben des Roboters schnell abschalten.

Die Arbeit gliedert sich dabei in mehrere Teilbereiche: Konstruktion und Fertigung des Rasenmäherroboters, die Konzeption und Implementierung eines Navigationsmechanismus, sowie Test und Auswahl geeigneter Sensorik zur Positionsermittlung, Kollisionserkennung, Test des Roboters und Evaluation der Mäh- und Navigationsfähigkeiten.

Zunächst wird daher die Mechanik des Roboters geplant und gefertigt. Dabei müssen Anforderungen zur Praxistauglichkeit und Sicherheitsaspekte beachtet werden. Nachdem das Chassis des Roboters aufgebaut wurde, wird dieser mit mehreren Motoren, Sensoren und einer Mikroprozessorplattform ausgestattet. Danach wird eine Steuersoftware implementiert, die die Navigation des Roboters übernimmt und diesen nach einem einfachen Zufallsalgorithmus über die Rasenfläche leitet. Die Steuersoftware berechnet aus den gegebenen Daten und den Fehler- und Vollzugsmeldungen der Sensorik und Motorsteuerung die notwendigen Steuerbefehle für die Aktorik. Dazu werden geeignete Sensoren ausgewählt, die die Ermittlung der Position und eine Fehlererkennung ermöglichen. So kann beispielsweise ein Infrarot- oder Ultraschallabstandssensor Gegenstände in Fahrtrichtung des Roboters erkennen und ein Ausweichen möglich machen, der Rand des Rasens kann z. B. über eine induktive Begrenzungsschleife erkannt werden.

Dieses Dokument gliedert sich in folgende Teile: zunächst werden bestehende kommerzielle Rasenmäherroboter mit ihren Eigenschaften aufgezählt und verglichen. Danach werden die bei diesem Projekt verwendeten Techniken erklärt, dann der Roboter in seinem mechanischen Aufbau sowie der verwendeten Hard- und Software beschrieben und die ungefähren Kosten des Roboters berechnet.

## **2. Bestehende Rasenmäherroboter**

Im Handel ist eine Vielzahl verschiedener Rasenmäherroboter verfügbar, die für verschiedene Kundengruppen angeboten werden. Hier wird nun eine kleine Auswahl von Robotern und deren Herstellern vorgestellt, die schon länger am Markt für Privatkunden aktiv sind und eine breite Modellpalette bieten.

### **2.1. Husqvarna Automower**

Husqvarna ist ein schwedischer Produzent von Garten- und Waldpflegegeräten und gehört zum Elektrolux-Konzern. Husqvarna stellte mit dem Automower im Jahr 1995 den ersten kommerziellen Rasenmäherroboter für den Massenmarkt vor [his]. Seitdem wurden mehrere neue Modelle vorgestellt, und eine ganze Familie von Rasenmäherrobotern entstand, die nach der maximal empfohlenen Mähfläche unterschieden werden. Husqvarna bietet mehrere Modelle in seinem Sortiment an, die Roboter werden aber zusätzlich noch mit modifizierten Gehäusen unter anderen Marken des Elektrolux-Konzerns vertrieben, u. a. der in Deutschland sehr bekannten Marke Gardena.

#### **2.1.1. Navigation**

Gemeinsam ist den Rasenmäherrobotern von Husqvarna, dass sie die Begrenzung der Rasenfläche über einen Begrenzungsdraht induktiv erkennen. An diesem Draht hält der Roboter an und wendet unter einem zufälligen Winkel, des Weiteren kann der Draht als Navigationshilfe benutzt werden um die Basis- und Ladestation zu finden.

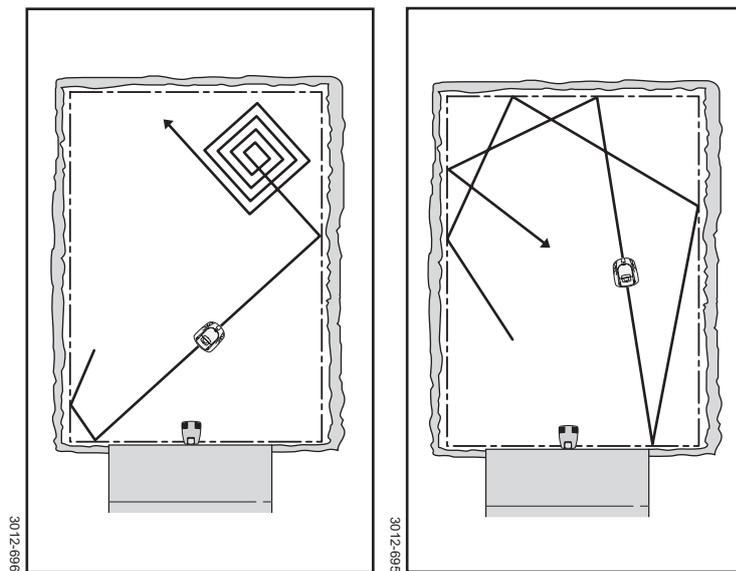


Abbildung 2.1.: Bewegungsmuster der Automower[hana]

Die teureren Modelle können zudem in einen weiteren Betriebsmodus wechseln: wird über die Betriebsparameter des Mähwerks erkannt, dass sich der Automower in einem Abschnitt der Rasenfläche mit höher gewachsenem Gras befindet, wechselt der Roboter in den sogenannten Viereckmodus (Abbildung 2.1). Hierbei wird das Gras ausgehend von der aktuellen Position aus in viereckig-spiralförmigen Bewegungen gemäht. Der Durchmesser dieser Spirale erweitert sich, bis der Automower an ein Hindernis oder den Begrenzungsdraht stößt.

### 2.1.2. Mähwerk

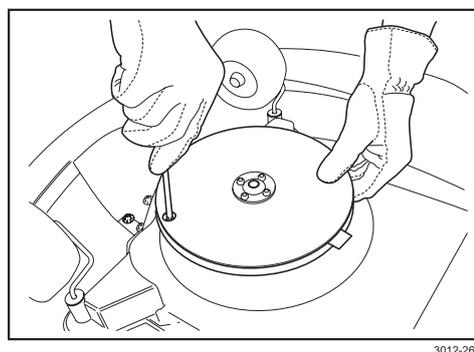


Abbildung 2.2.: Montage der Mähmesser[hana]

Das Mähwerk der Husqvarna-Rasenmäherroboter ist ein einfaches Sichelmähwerk. Vom Mähwerksmotor wird eine Metallscheibe angetrieben, an der mehrere Klingen konzentrisch angeordnet und locker gelagert sind. Sie werden mit einer einfachen Schraube befestigt (Abbildung 2.2), an der ein kurzes Stück unter dem Schraubenkopf ohne Gewinde ist – dieses Stück bildet die Drehachse für ein Messer. Dies hat den Vorteil, dass die Klingen einfach auszutauschen sind. Der einfache Aufbau der Klingen sorgt auch für einen geringen Ersatzteilpreis, dieser liegt je nach Anbieter bei ca. einem Euro je Messer. Da der Mähwerksmotor im Vergleich zu konventionellen Handrasenmähern nur eine geringe Leistung hat, müssen die Klingen auch regelmäßig gewartet und ausgetauscht oder nachgeschliffen werden, da diese scharf sein müssen um das Gras auch schneiden zu können.

## **2.2. Robomow**

Die Robomow Rasenmäherroboterfamilie wird von der israelischen Firma Friendly Robotics hergestellt. Das erste Modell wurde 1998 vorgestellt [rob]. Seitdem wurden regelmäßig neue Modelle vorgestellt und – vergleichbar mit Husqvarna – eine ganze Familie von Robotern für unterschiedliche Leistungsanforderungen erstellt. Einige Robomow-Modelle werden auch unter anderen Marken verkauft, so z. B. als Viking iMow.

### **2.2.1. Navigation**

Die Robomow-Rasenmäherroboter arbeiten ebenso wie die Automower von Husqvarna mit einem induktiven Begrenzungsdraht, den sie erkennen und an dem sie dann in einem zufälligen Winkel drehen, um auf der Rasenfläche zu bleiben – diese Betriebsart wird hier „Scan“ genannt. Zudem sind die Robomow-Roboter in der Lage, dem Begrenzungsdraht zu folgen und so die Kante der Rasenfläche zu mähen.

Die teureren Modelle bieten zusätzlich zu den beiden Betriebsmodi eine manuelle Steuerung durch eine kabelgebundene Fernbedienung. Mit dieser lässt

sich ein Robomow-Roboter z. B. auf eine zweite Rasenfläche oder an einen Lagerplatz bewegen, ohne den Roboter anheben zu müssen.

### **2.2.2. Mähwerk**



Abbildung 2.3.: Mähwerk eines Robomow [hanb]

Ebenso wie bei den Automower-Mährobotern setzen die Robomow auf ein Spindelmähwerk. Hier werden allerdings bis zu drei versetzt angeordnete Mähwerke eingesetzt, die mit einem V-förmigen Spezialmesser ausgestattet sind (Abbildung 2.3). Diese müssen nach 200 Betriebsstunden ausgetauscht werden, dies führt zu vergleichsweise hohen Kosten, da eins dieser Messer etwa 25 Euro kostet.

## **2.3. Zucchetti**

Die italienische Firma Zucchetti Centro Sistemi vertreibt seine Rasenmäherroboter unter anderem unter dem Markennamen „Wiper“, als Herstellerangabe findet man im Handel oft die Firma Ambrogio, die zum Zucchetti-Konzern gehört. Zucchetti bietet eine sehr breite Modellpalette an, die Roboter bieten unterschiedlichste Eigenschaften und sind teilweise auch mit weiterer Sensorik als Sonderzubehör bestellbar.

### **2.3.1. Navigation und Fahren**

Zuchetti bietet in einigen Modellen alternativ zum Begrenzungsdraht die Flächenerkennung über einen sogenannten „Zuchetti Gras Sensor“ (ZGS) an. Dies ist ein Feuchtigkeitssensor, mit dem Gras von der umgebenden Fläche unterschieden werden kann. Dieser ZGS minimiert daher den Installationsaufwand auf ein Minimum.

Als Bewegungsmodus wird in den Robotern ebenfalls auf das Zufallsprinzip gesetzt, einige Modelle sind auch in der Lage dazu, hohe Stellen im Gras zu erkennen und dort spiralförmig zu fahren.

Einige Modelle sind zudem symmetrisch aufgebaut und können daher auf Wendemanöver an der Rasenkante verzichten.

### **2.3.2. Mähwerk**

Die Zuchetti-Rasenmäherroboter setzen ebenfalls auf ein Sichelmähwerk. Dieses wird mit einem Sternförmigen Messer gebildet. Je nach Modell werden Messer mit vier oder acht Klingen eingesetzt. Empfehlungen zum regelmäßigen Tausch gibt es nicht, ein Ersatzmesser kostet zwischen 30 und 50 Euro.

## **2.4. Bosch Indego**

Der Bosch Indego ist der erste Rasenmäherroboter von Bosch. Dieser bietet als Neuheit ein Navigationssystem namens LogiCut.

### **2.4.1. Navigation und Fahren**

Das LogiCut Navigationssystem ist ein System, bei welchem zunächst der Rasen entlang der Induktionsschleife vermessen wird, und dann der Rasen in parallelen Bahnen zeitsparend abgemäht wird. Nach Herstellerangaben benötigt dieses Verfahren nur ein Drittel der fürs zufällige Abfahren der Rasenfläche notwendige Zeit [bos].

## **2.4.2. Mähwerk**

Der Bosch Indego arbeitet mit einem Mähwerk, welches ähnlich aufgebaut ist wie das Mähwerk der Husqvarna Automower. Allerdings sind aufgrund der Neuheit des Gerätes bisher nur Originalersatzmesser verfügbar, welche mit 18 Euro pro Stück relativ teuer sind.

## **2.5. weitere Hersteller**

Außer den oben genannten Herstellern werden auch andere Rasenmäherroboter am Markt beworben. Für den Heimbedarf sind dies aber in der Mehrzahl Produkte, die in Lizenz oder im Auftrag hergestellt wurden und bis auf die Gehäusefarbe oft identisch sind mit den Angeboten der drei größten Hersteller. Daneben gibt es noch eine Reihe weiterer Firmen wie Belrobotics SA, die Rasenmäherroboter für speziellere Anwendungen herstellen, z. B. für das Mähen von Golfplätzen, großen Parks oder Fußballplätzen [bel].

## **2.6. Zusammenfassung**

Bei den derzeit kommerziell verfügbaren Rasenmäherrobotern kann man beobachten, dass mit Ausnahme des Bosch Indego keiner der Roboter geplant vorgeht und die Rasenfläche vollständig zeitsparend abmähen kann. Vereinzelt kann auf Flächen mit einfachem, rechteckigem Zuschnitt eine bessere Variante des Wendens an der Begrenzung beobachtet werden, dies ist aber oft den teureren Modellen vorbehalten.

## **3. Grundlagen und verwendete Techniken**

In diesem Abschnitt werden die verwendeten Techniken und weitere Grundlagen erklärt. Diese reichen von grundlegenden Navigationstechniken über unterschiedliche Sensoren bis hin zur Aktorik und Steuerungstechnik.

### **3.1. Navigation**

Für die Navigation des Rasenmäherroboters gibt es unterschiedliche Sensor- und Positionsfeststellungstechniken, die hier kurz vorgestellt werden.

#### **3.1.1. Inertialnavigation**

Unter Inertialnavigation versteht man die Fortschreibung der Position über die Beschleunigungen des Fahrzeugs. Hierzu müssen Sensoren für die Beschleunigung in den drei Raumachsen sowie Drehraten um die drei Raumachsen vorhanden sein [Wen07].

Der größte Vorteil der Inertialnavigation ist die Unabhängigkeit von äußeren Einflüssen, allerdings ist die Inertialnavigation nur mit aufwendigen und sehr teuren Sensoren genau genug, um hiermit die Position eines Rasenmäherroboters auf einer Rasenfläche festzustellen.

### **3.1.2. Satellitennavigation**

Satellitennavigation arbeitet über die einfache Triangulation von Funksignalen von Satelliten auf bekannten Orbits um die Erde. Hierbei wird ein Zeitsignal von jedem Satelliten gesendet, über den Zeitunterschied der Ankunft der Signale im Empfänger und der bekannten Position des Senders kann die Position des Empfängers auf wenige Meter genau berechnet werden [Wen07]. Dies ist allerdings nicht genau genug, um damit auf einer Rasenfläche zu navigieren.

Größter Vorteil der Satellitennavigation ist, dass außer dem Empfänger keine Infrastruktur benötigt wird, um die Position festzustellen. Allerdings sind die von den Satelliten gesendeten Funksignale sehr schwach, und werden von an der Rasenfläche befindlicher Bebauung oder Bäumen abgeschirmt. Dadurch kommt es zu Ungenauigkeiten, so dass z.B. das amerikanische GPS-System nur auf ca. 30m genau ist.

Eine Erweiterung des GPS-Systems ist differenzielles GPS oder DGPS. Hierbei wird ein zweiter Empfänger an einer festen Position aufgebaut, welcher mit dem Empfänger auf dem Roboter kommunizieren kann. Hierdurch kann die Position des Roboters auf wenige Millimeter genau, relativ zum fest montierten Empfänger, bestimmt werden. Allerdings sind hierfür spezielle GPS-Empfänger notwendig.

### **3.1.3. Induktive Begrenzungsschleife**

Eine induktive Begrenzungsschleife oder Induktionsschleife ist ein Kabel, welches einmal um die zu befahrende Fläche gelegt wird. In diesem Draht wird ein wechselndes Magnetfeld erzeugt, welches von Sensoren im Rasenmäherroboter aufgenommen werden kann. Damit kann der Rasenmäherroboter feststellen, ob er sich innerhalb oder außerhalb seines Arbeitsbereiches befindet. Dazu kann die Schleife als Referenz für eine Positionsberechnung benutzt werden. Die Schaltung basiert auf einem Schaltungsvorschlag im Roboternetz-Wiki [ind].

### 3.1.3.1. Sender

Der Sender ist sehr einfach aufgebaut. Dieser besteht aus einem leistungsstarken Netzteil und einer Sendeschaltung, bei der ein Timerbaustein einen Takt erzeugt, welcher benutzt wird um einen MOS-FET-Transistor zu schalten. Dieser MOS-FET schließt dann einen Stromkreis über die Begrenzungsschleife, wodurch hier ein hoher Strom fließt und kurzzeitig ein starkes Magnetfeld aufgebaut- und dann wieder abgebaut wird.

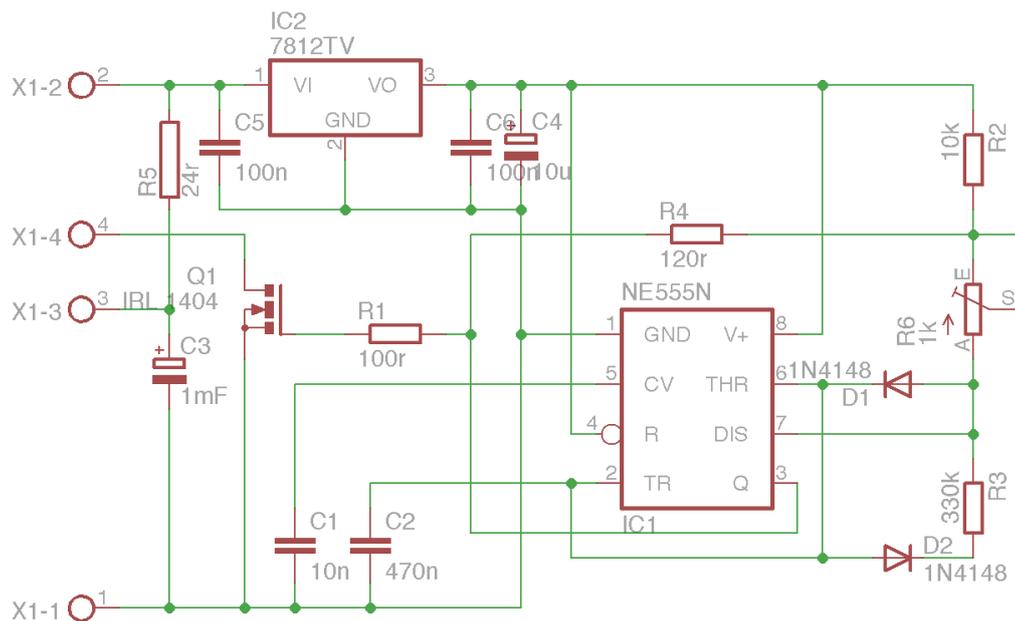


Abbildung 3.1.: Schaltplan der Sendeschaltung

In Abbildung 3.1 ist der Schaltplan der Sendeschaltung abgebildet. Bei den Anschlüssen X1-1 und X1-2 wird die Stromversorgung angeschlossen, die Induktionsschleife bei den Anschlüssen X1-3 und X1-4. Am Potentiometer R6 kann die Impulsbreite eingestellt werden.

### 3.1.3.2. Empfänger oder Sensor

Der Schleifenempfänger besteht im Kern aus einer Spule, die aus einem 24V-Relais extrahiert wurde. Hierzu wurde das Relais aufgesägt und der Kern und der Anker der Spule entfernt.

Die Impulse der Sendeschleife induzieren eine Spannung in der Spule, diese wird mit einem Differenzverstärker verstärkt (Schaltplan in Abbildung 3.2) und das Signal dann mit dem Analogeingang eines Microcontrollers ausgewertet. Dieses ausgewertete Signal wird dann an die Steuerung weitergegeben.

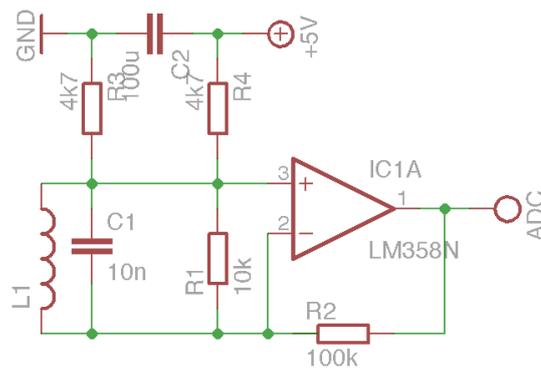


Abbildung 3.2.: Schaltplan der Empfängerschaltung

### 3.1.4. Odometrie

Als Odometrie bezeichnet man die Berechnung der Position eines Roboters anhand der Bewegung seiner Aktoren. Beispielsweise kann die zurückgelegte Entfernung aus der Umdrehungszahl der Antriebsräder errechnet werden.

Odometrie ist eine ungenaue Vorgehensweise, da Störungen wie Schlupf der Räder oder Unebenheiten in der zu befahrenden Fläche nicht bemerkt werden können. Daher muss die berechnete Position immer wieder durch weitere Sensoren oder Referenzpunkte korrigiert werden [HLN12].

### 3.1.5. Ultraschallsensoren

Ultraschallsensoren senden einen kurzen, hochfrequenten Ton aus. Dieser breitet sich über die Luft aus, und wird dann von einem Hindernis reflektiert. Durch die Reflektion wird ein Teil des Schallimpulses zurück zum Sensor geworfen und dort aufgenommen. Aus der Zeit zwischen Absenden des Tons und dem Eintreffen der Reflektion lässt sich die Entfernung zwischen dem Sensor und dem Objekt berechnen.

Ultraschallsensoren haben eine sehr gute Hintergrundausschleifung, d. h. sie detektieren zuverlässig das nächste am Sensor befindliche Objekt, welches die Schallimpulse reflektieren kann. Allerdings haben sie Probleme mit Objekten, die sich zu nah am Sensor befinden, erst ab einer modelltypischen Entfernung ist die Messung zuverlässig. Ein weiteres Problem sind weiche Gegenstände, die Schall absorbieren statt ihn zu reflektieren [HLN12].

## **3.2. Aktorik**

In diesem Abschnitt werden kurz die Antriebsmöglichkeiten für den Rasenmäherroboter erläutert. Da der Roboter autonom und ohne Fremdenergieversorgung auf einer Rasenfläche fahren soll, kommen nur zwei Typen von Elektromotoren in Frage: der Gleichstrommotor und der Schrittmotor.

### **3.2.1. Gleichstrommotor**

Ein Gleichstrommotor basiert auf dem Prinzip, dass ein Magnet innerhalb eines äußeren, festen Magnetfeldes abgestoßen bzw. angezogen wird. Dieser innere Magnet ist auf einer Rotationsachse befestigt und wird durch Wechsel der Polarität dazu gebracht, sich in dem äußeren Magnetfeld zu drehen. Das äußere Magnetfeld wird auch Stator genannt, das innere Anker oder Rotor.

Das innere Magnetfeld muss dabei durch einen Elektromagneten erzeugt werden, das äußere Magnetfeld wird je nach Motortyp und Anwendungsfall durch einen Permanentmagneten oder ebenfalls einen Elektromagneten erzeugt.

Die Rotation wird durch das Umpolen des Ankers erzeugt. Zunächst würde sich der Anker, da er frei drehend gelagert ist, am Statormagnetfeld ausrichten, so dass der Nordpol des Ankermagneten auf den Südpol des Statormagneten zeigt (Abbildung 3.3).

Ist diese Ausrichtung vollzogen, wird der Ankermagnet umgepol, so dass jetzt der Nordpol des Ankermagneten auf den Nordpol des Statormagneten zeigt. Dadurch bilden sich Abstoßungskräfte, die dazu führen dass die Rotation fortgesetzt wird. Dieser Ablauf wiederholt sich mehrmals in einer Umdrehung.

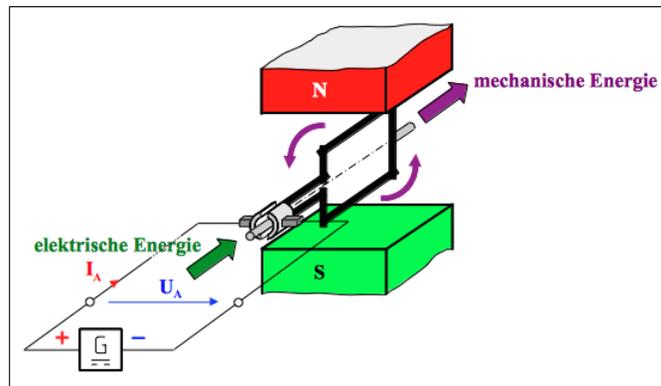


Abbildung 3.3.: prinzipielle Funktionsweise eines Gleichstrommotors [ele]

### 3.2.2. Schrittmotor

Schrittmotoren basieren auf ähnlichen Prinzipien wie Gleichstrommotoren. Auch hier wird ein Magnet innerhalb eines äußeren Magnetfeldes zur Rotation gebracht. Im Unterschied zum Gleichstrommotor wird aber nicht das Anker magnetfeld umgepolt, sondern es werden zwei unpolbare Statormagneten verwendet, die ein exaktes Steuern der Rotation ermöglichen.

Zunächst wird eins der Statormagnetfelder eingeschaltet und der Anker richtet sich in eine definierte Position aus. Dieses Magnetfeld wird nun abgeschaltet, und die andere Statorspule eingeschaltet. der Anker folgt dem neuen Magnetfeld und richtet sich am zweiten Statormagneten aus. Diese Rotation wird Schritt oder Vollschritt genannt (Abbildung 3.4)

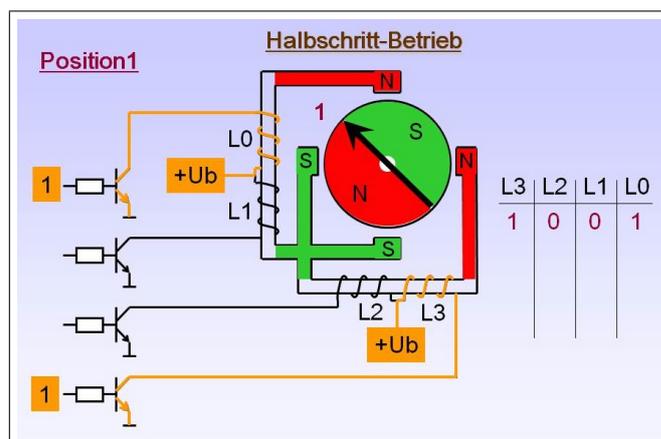


Abbildung 3.4.: Prinzipskizze eines Schrittmotors - Halbschritt [sch]

Eine weitere Möglichkeit besteht darin, das Magnetfeld der ersten Statorspule eingeschaltet zu lassen und die zweite Statorspule dazuzuschalten. Der Anker führt daraufhin eine Rotation durch, die vom Winkel einem halben Schritt entspricht. Wird dann der erste Statormagnet abgeschaltet, wird ein weiterer halber Schritt durchgeführt. Diese Betriebsart nennt man Halbschrittbetrieb.

Durch Steuerung des Spulenstroms kann das Magnetfeld der Statormagneten direkt gesteuert werden. Hierdurch ergibt sich die Möglichkeit, weitere Zwischenstationen zwischen zwei Vollschritten einzufügen. Dieses nennt man Mikroschrittbetrieb.

Durch die verschiedenen Ansteuerweisen erhält man einen sehr flexibel einsetzbaren Motor, der aber eine komplexe elektronische Steuerung benötigt.

### **3.3. Steuerung**

In der Steuerung des Roboters können unterschiedliche Steuerungshardwares verwendet werden. Hier werden kurz zwei Microcontrollerplattformen und der für die Ansteuerung vieler Sensoren notwendige I<sup>2</sup>C-Bus betrachtet.

#### **3.3.1. Raspberry Pi**

Ein Raspberry Pi ist ein kleiner, leistungsfähiger Einplatinencomputer, der in etwa die Größe einer Kreditkarte besitzt. Er ist ausgestattet mit einer ARM11-CPU, welche mit 700MHz getaktet ist. Neben der CPU befindet sich auf dem Hauptchip des Raspberry Pi, einem System On Chip, der Hauptspeicher und ein Grafikprozessor [ras]. Daneben sind auf der Platine zahlreiche Schnittstellen untergebracht:

- zwei Typ A USB-Anschlüsse
- ein HDMI-Monitoranschluss
- ein Ethernetanschluss
- ein 3,5mm Klinkenanschluss für analoge Stereo-Ton-Ausgabe
- ein Cinch-Anschluss für analoge Bildausgabe

- ein Camera Serial Interface CSI
- ein 26-poliger Anschluss für digitale Ein- und Ausgabe (GPIO)
- ein MicroUSB-Anschluss zur 5V-Spannungsversorgung

Das Betriebssystem eines Raspberry Pi wird von einer SD-Speicherkarte geladen. Seit Erscheinen des Raspberry Pi wurden diverse Betriebssysteme portiert, unter anderem einige Linuxvarianten und RISC OS 5.

Ein Raspberry Pi ist über seine GPIO-Schnittstelle einfach erweiterbar, durch den 700MHz-ARM11-Prozessor sehr leistungsfähig. Mit einem Anschaffungspreis von etwa 40 Euro bietet er ein sehr gutes Preis-/Leistungsverhältnis.

### **3.3.2. Arduino**

Die Arduino-Platinen basieren auf den megaAVR-Mikroprozessoren von Atmel. Diese bieten einen internen Programmspeicher und werden über eine serielle Schnittstelle und einen in den Prozessoren gespeicherten Bootloader programmiert. Zur Arduino-Plattform gehört auch eine Entwicklungsumgebung, die ein einfaches Programmieren in der Programmiersprache C ermöglicht.

Die Plattform besteht aus einer Basisplatine, dem Arduino Board, und mit dieser Platine verbundene sogenannte Shields. Dies sind Tochterplatinen, die auf die Basisplatine oder ein bereits vorhandenes Shield aufgesteckt werden und damit die Funktionalität der Basisplatine um diverse Sensorik, Spezialfunktionen wie Schrittmotorentreiber oder auch einfach nur Anschlussfelder für externe Geräte erweitert. Diese Shields, wie auch die Arduino Boards selbst, werden von mehreren Herstellern hergestellt. Möglich ist dies durch die offene Architektur und die freie Verfügbarkeit der Quelltexte, aber auch der Platinenlayouts [ard].

Shields sind je nach Anwendung der Plattform austauschbar, auch Eigenentwicklungen sind einfach möglich. Es gibt eine Vielzahl unterschiedlicher Shield-Platinen, die unterschiedlichste Funktionen ermöglichen, von einfachen digitalen Ein- und Ausgängen bis hin zu drahtlosen Funkverbindungen über WLAN oder UMTS/GSM. Diese Vielzahl an verfügbaren Erweiterungsmöglichkeiten erlaubt eine große Flexibilität, allerdings sind für den Umgang mit

der Arduino-Plattform Elektronikkenntnisse von Vorteil, da viele Shields nur als Bausatz verfügbar sind und Steckverbindungen nicht genormt sind – sofern überhaupt vorhanden.

Die Preise für die einzelnen Arduino-Elemente sind gering, beispielsweise kostet die Basisplatine „Arduino Mega 2560“ etwa 50 Euro, diese wird jedoch auch von anderen Anbietern produziert und ist dann günstiger. Ein Bausatz für einen Shield mit Schrittmotortreibern kostet etwa 15 Euro.

### 3.3.3. I<sup>2</sup>C-Bus

Der I<sup>2</sup>C-Bus ist ein einfacher serieller Bus, der in den 1980er Jahren von Philips Semiconductors (heute NXP) für die Interplatinen- und Interchipkommunikation entwickelt wurde. Aus dieser Anwendung wurde auch der Name des Busses abgeleitet: „Inter Integrated Circuit“. Außer der Masseleitung werden hier nur zwei weitere Leitungen benötigt, eine Taktleitung und eine Datenleitung. Aus diesem Grund wird der I<sup>2</sup>C-Bus auch „Two Wire Interface“ oder kurz „TWI“ genannt [avr04].

I<sup>2</sup>C ist ein Master-Slave-Bus. Hierbei wird ein Datentransfer immer vom Busmaster gestartet, der angesprochene Busslave antwortet dann auf den Datentransfer. Hierzu muss zunächst immer die Adresse des Slaves übertragen werden, anschließend folgen ein oder mehrere Datenbytes. Der Slave antwortet dann, entweder mit einer Bestätigung oder mit den abgefragten Daten.

Auf elektrischer Ebene ist es so, dass die Busteilnehmer mit Open-Collector-Ausgängen ausgestattet sind, und der Bus „active low“ ist. Die beiden Leitungen für Takt und Daten werden von Pull-up-Widerständen auf die Versorgungsspannung gezogen, wenn ein Busteilnehmer senden will, zieht er die Leitung auf Masse. Arbeiten zwei Busteilnehmer mit unterschiedlichen Versorgungsspannungen, ist eine Pegelanpassung notwendig.

### **3.4. Mähtechniken**

Für das Mähen von Gras kommen grundsätzlich zwei Verfahren in Frage: das Abschneiden und das Abschlagen.

Das Abschneiden von Gras kann entweder mit einer oszillierenden Schneide durchgeführt werden, dies wird z.B. bei einem Balkenmäherwerk benutzt, oder mit einer rotierenden Schneide. Ein Balkenmäherwerk ist dabei aufgebaut wie zwei übereinanderliegende Kämme, von denen einer der Kämme bewegt wird und somit das Gras abgeschnitten wird. Gemeinsam ist allen Mäherwerkstypen, dass es eine Gegenschneide gibt, also der Grashalm von zwei Schneiden berührt wird.

Beim Abschlagen gibt es keine Gegenschneide, hier wird alleine die Massesträgheit des Grashalms ausgenutzt, um diesen mit einem Schneidmesser zu durchtrennen. Der Nachteil z.B. eines Scheibenmähwerks ist, dass eine hohe Umdrehungsgeschwindigkeit des Schneidmessers vorhanden sein muss, um das Gras abzutrennen, desweiteren müssen die Messer sehr scharf sein, damit sich der Grashalm nicht einfach wegbiegt und er unter dem Mäherwerk durchtaucht. Die Vorteile eines Scheibenmähwerks liegen aber im einfachen Aufbau und im geringen Gewicht [KHH09].

### **3.5. Sicherheit**

Die Sicherheit spielt bei Rasenmäherrobotern eine große Rolle, da ein Mäherwerk eine Gefahr darstellt und es ohne weitere Sicherheitsmaßnahmen zu Verletzungen kommen kann.

Die Sicherheitsmaßnahmen werden hierbei grob in zwei Gruppen gegliedert: aktive und passive Maßnahmen.

Unter passiven Methoden versteht man solche, die beim Ausfall der Funktion selbsttätig die Sicherheit gewährleisten. Dies können Gehäuse sein, die eine Berührung verhindern, aber auch Maßnahmen, die im Normalbetrieb eine ständige Energiezufuhr benötigen und beim Ausfall der Energiezufuhr das System in einen sicheren Zustand bringen [Neu11]. Für einen Rasenmäherroboter

wären dies beispielsweise Schutzgehäuse um das Mähwerk und die Fahrmotoren, damit diese im Betrieb nicht berührt werden können. Ebenfalls kann eine Motorenansteuerung so konstruiert werden, dass das Mähwerk beim Ausfall der Spannungsversorgung schnell stoppt.

Aktive Sicherheit hingegen erfordert das Eingreifen eines Steuerungsmechanismus. Beispielsweise muss beim Anheben des Roboters das Mähwerk schnell abgeschaltet werden. Diese Schnellabschaltung erfordert allerdings einen Auslöser, dieses kann ein Berührungsschalter sein, mit dem der Gegenstand erfasst wird der das Anheben verursacht, oder ein Sensor der auf die beim Anheben wirkenden Beschleunigungen reagiert.

Insgesamt müssen die aktiven und passiven Sicherheitsmaßnahmen ausreichen, das eine Berührung des Mähwerks nicht möglich ist, wenn der Rasenmäherroboter in Betrieb ist.



## **4. Aufbau des Roboters**

Die Konstruktion des Roboters wurde zunächst in der 3D-Konstruktionssoftware Autodesk Inventor modelliert. Anschließend wurden die einzelnen Bauteile bestellt und der Roboter montiert. Dabei gab es vier Iterationen in der Entwicklung, in denen zunächst das Chassis des Roboters und der Aufbau des Roboters festgelegt, und anschließend das endgültige Gehäuse entwickelt wurden. Dieser Abschnitt beschreibt den Aufbau nach dem letzten Umbau.

### **4.1. Chassis**

Das Chassis des Roboters wurde aus Aluminium-Stebenprofilen aufgebaut. Dadurch waren Änderungen an der Konstruktion einfach durchzuführen, das Chassis ist kostengünstig und stabil. Die Profile haben eine quadratische Grundform, mit einer Nut an jeder langen Außenseite, an denen man mit Nutsteinen oder Hammerkopfschrauben einfach weitere Bauelemente befestigen kann. In der Mitte der Stirnseite haben die Profile eine Bohrung, in die man ein Gewinde schneiden und dann dort ebenfalls Bauelemente oder andere Profile befestigen kann.

Zentrales Element des Chassis ist ein Rahmen, der rechteckig aufgebaut ist – die Enden der kürzeren Stebenprofile (Nr. 1 in Abbildung 4.1) stehen hier 30mm über, um geschützten Bauraum für die Bereifung bereitzustellen.

Die Schnittkanten der Strebenprofile sind mit Abdeckkappen vor zufälligen Berührungen geschützt, um Schnittwunden bei der Handhabung mit dem Rasenmäherroboter zu vermeiden.

Die mit Winkeln befestigten, senkrecht zum Hauptrahmen stehenden Stebenprofile (Nr. 3 in Abbildung 4.1) dienen zur Motorenbefestigung. Durch die

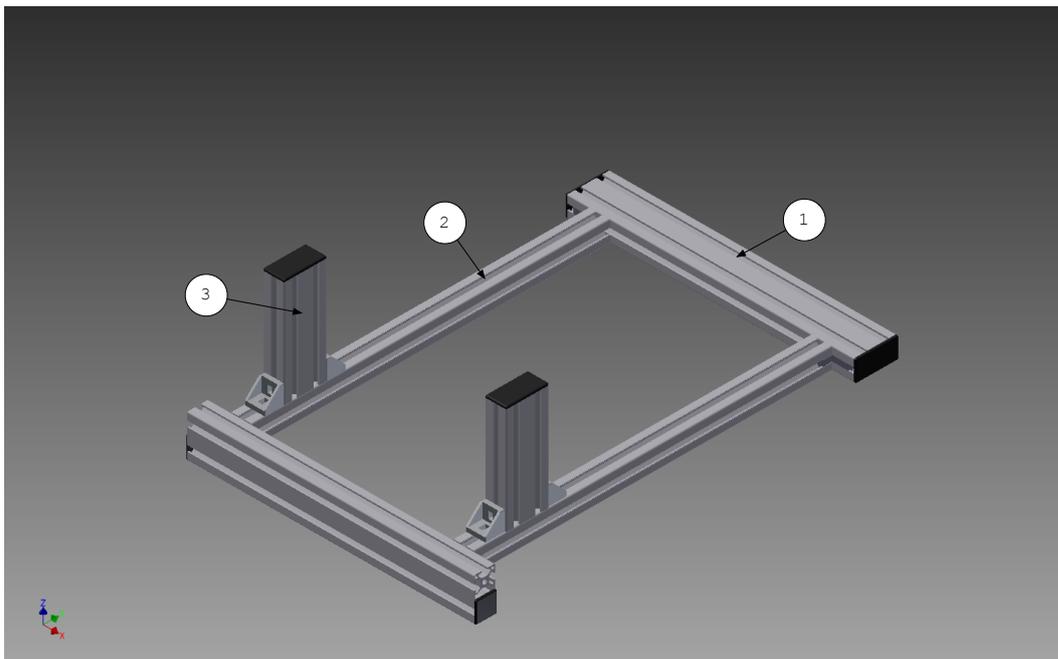


Abbildung 4.1.: Aufbau des Chassis des Rasenmäherroboters

durchgehende, hier senkrecht stehende Nut kann jeder Motor von der Radachse weggezogen und die Antriebszahnriemen damit gespannt werden.

Am Chassis befestigt ist das Gehäuse, welches aus 9mm starker Siebdruckplatte gefertigt wurde. Siebdruckplatten bieten gegenüber anderen Materialien wie Metallblechen den Vorteil, dass sie einfach bearbeitbar, kostengünstig und leicht verfügbar sind. Zudem sind Siebdruckplatten durch die Verwendung von wasserfesten Klebstoffen resistent gegenüber leichten Regenschauern, und sind vergleichsweise leicht. In Abbildung 4.2 ist das Chassis mit montiertem Gehäuse abgebildet.

Im vorderen Teil des Roboters bildet sich so ein großer Bauraum, in dem ausreichend Platz für die Motoren, die Leistungselektronik und die Akkus vorhanden ist. Durch die Anbringung zweier Scharniere ist die obere Platte des Gehäuses als Deckel ausgeführt, so ist ein einfacher Zugang für Wartungszwecke gewährleistet. Im hinteren Teil des Roboters ist nur eine Polystyrolscheibe als Abdeckung auf das Chassis gelegt, zwischen Bodenplatte und der Polystyrolscheibe ist so ein geschützter Raum für die Steuerungselektronik, der aber trotzdem auch während der Benutzung des Roboters gut einsehbar ist.

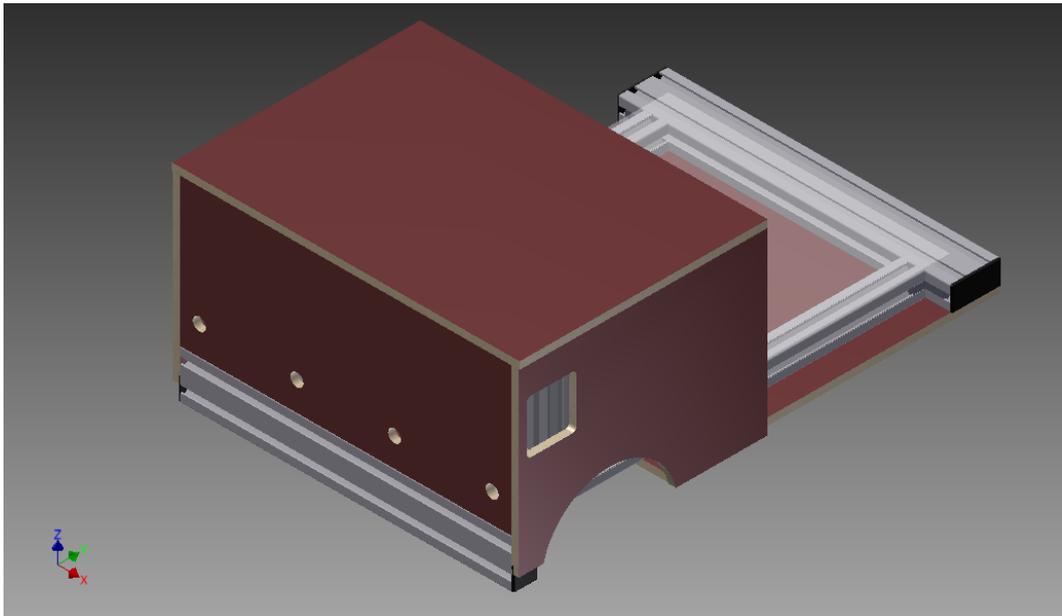


Abbildung 4.2.: Chassis mit Gehäuse des Rasenmäherroboters

## 4.2. Antrieb

Der Antrieb des Rasenmäherroboters wurde mit zwei Schrittmotoren aufgebaut. Hierbei treibt jeweils ein Motor ein in Fahrtrichtung links und ein in Fahrtrichtung rechts montiertes Antriebsrad über ein Zahnriemengetriebe an (Abbildung 4.3).

Durch die Verwendung von zwei Motoren, je einer links und rechts, erhält man die Möglichkeit, den Roboter um den Mittelpunkt zwischen den Antriebsrädern drehen zu lassen.

Schrittmotoren bieten prinzipbedingt die Möglichkeit, Odometrie als Wegstreckenmessung einzusetzen. Dies erleichtert den mechanischen Aufbau, da keine zusätzlichen Sensoren an den Antriebsachsen befestigt werden müssen. Allerdings sind Schrittmotoren nur mit relativ geringen Drehmomenten verfügbar, oder haben einen hohen Energiebedarf. Daher ist ein Getriebe notwendig, welches mit Zahnriemen aufgebaut wurde.

Für die Zahnriemen wurden zwei Riemenscheiben eingesetzt, mit 15 Zähnen auf der Motorseite und mit 72 Zähnen auf der Antriebsachse. Dies ergibt ein Übersetzungsverhältnis von 1:4,8. Da die Schrittmotoren ein maximales Halte-

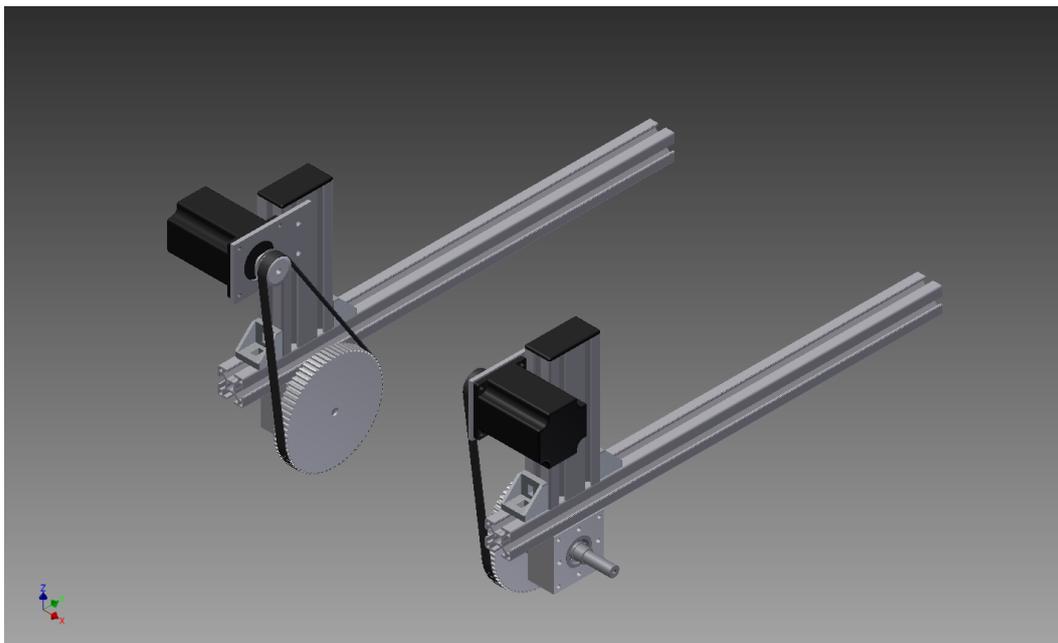


Abbildung 4.3.: Aufbau des Antriebs

moment von 2,0Nm besitzen, beträgt das Haltemoment auf der Antriebsachse durch das Zahnriemengetriebe 9,6Nm.

Als Zahnriemenprofil wurde das HTD-Profil ausgewählt. Dieses bietet im Vergleich zu den üblicheren Trapezprofilen durch seinen halbkreisförmigen Aufbau einen spielarmen Antrieb. Die Zahnriemen können durch Verschiebung der Motoren am vertikal stehenden Strebenprofil gespannt werden, eine zusätzliche Spannrolle ist bei den geringen Riemengeschwindigkeiten nicht notwendig.

Durch das Zahnriemengetriebe gewinnt man auch Genauigkeit: Statt einer Motorumdrehung mit 200 Schritten sind 960 Schritte notwendig, um die Antriebsachse einmal zu drehen.

### 4.3. Mähwerk

Das Mähwerk des Roboters wurde aus Ersatzteilen des Mähwerks eines Husqvarna Automower 105 und einem Gleichstromtriebemotor aufgebaut. Die Verwendung von Ersatzteilen eines bestehenden Mähwerks ermöglichte einen kostengünstigen und einfachen Aufbau.

Das Mähwerk besteht aus einer Scheibe, an der drei Klingen drehbar gelagert sind, jeweils in gleichem Abstand zur Nabe. Die Nabe ist mit einem kurzen Rohr als Antriebswelle direkt mit dem Getriebe des Getriebemotors verbunden.

Der Getriebemotor wird über ein Relais ein- und ausgeschaltet, welches von der Steuerung angesprochen werden kann. Im ausgeschalteten Zustand sind über das Relais die zwei Anschlüsse des Motors kurzgeschlossen, dies sorgt für eine starke Bremswirkung bei Abschaltung, die durch das induzierte Magnetfeld aufgebaut wird.

Der Getriebemotor hat ein Übersetzungsverhältnis von 1:2,5 und eine Nenn-drehzahl von 5040 U/min bei einer Nennspannung von 12V.

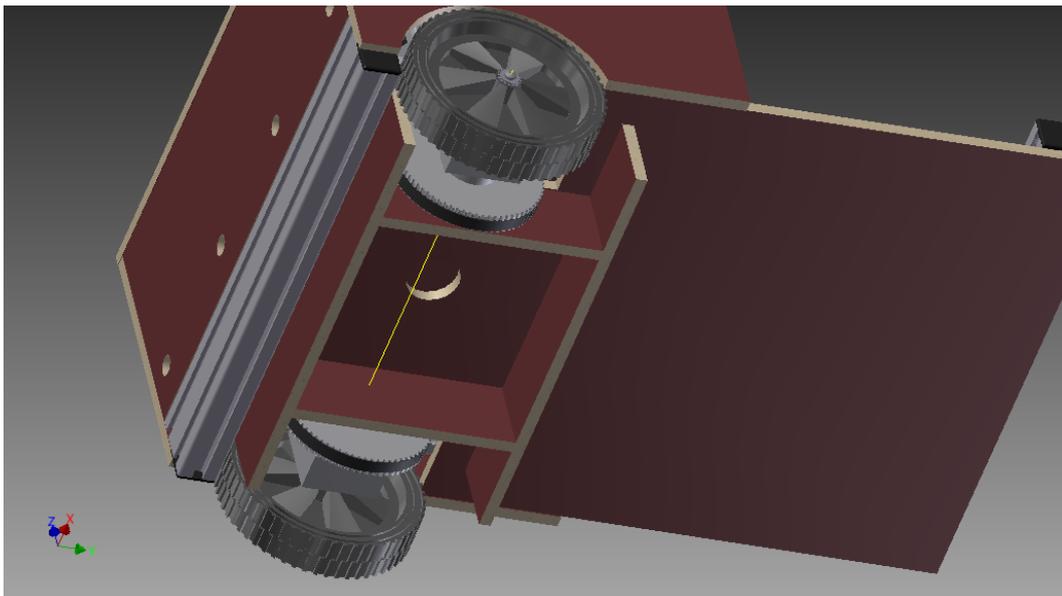


Abbildung 4.4.: Schutzabdeckung des Mähwerks

Zum Schutz des Mähwerks und der Umgebung ist an der Bodenplatte noch ein Schutz angebracht (Abbildung 4.4). Dieser besteht aus vier Siebdruckplatten und umschließt das Mähwerk vollständig. Desweiteren dient dieser Schutzaufbau der Befestigung von Abdeckplatten, welche ein Eindringen von Dreck in den Roboter verhindern sollen.

## 4.4. Schaltplan

Der Schaltplan des Rasenmäherroboters ist in mehrere Abschnitte gegliedert, welche jeweils eine Baugruppe genauer beschreiben. In Abbildung 4.5 ist zunächst eine Gesamtübersicht des Schaltplans des Roboters abgedruckt.

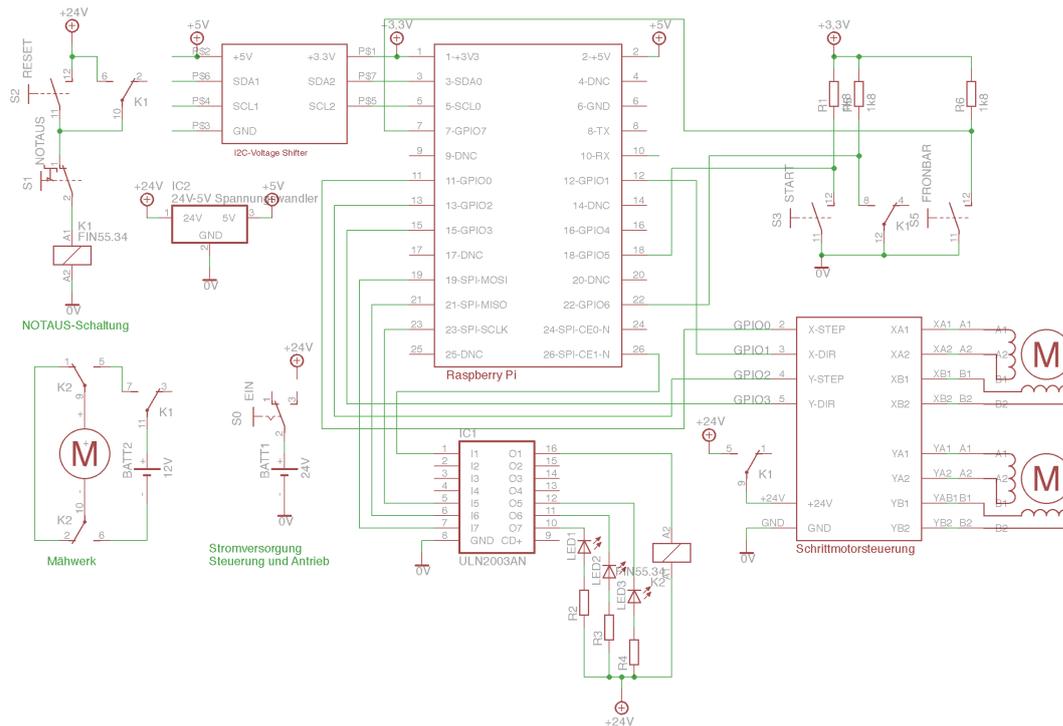


Abbildung 4.5.: Gesamtschaltplan des Rasenmäherroboters

### 4.4.1. Hauptschalter

Über einen Hauptschalter (Abbildung 4.6), der oben auf dem Rasenmäherroboter montiert ist, kann die Spannungsversorgung des 24V-Akkus unterbrochen bzw. eingeschaltet werden.

Nach Abschalten des Hauptschalters ist die Spannungsversorgung des Mähwerks nur indirekt, über das Relais der Not-AUS-Schaltung, unterbrochen.

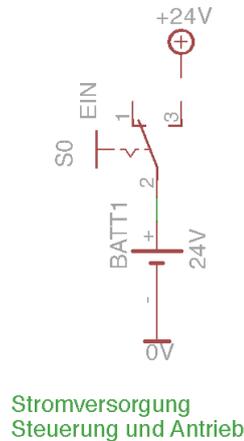


Abbildung 4.6.: Schaltplan des Hauptschalters

#### 4.4.2. Not-AUS

Die Not-AUS-Schaltung in Abbildung 4.7 ist unabhängig von der Steuerungshardware realisiert, die Steuerungshardware wird nur über den Zustand des Not-AUS-Relais informiert, indem einer der Relaiskontakte an einem digitalen Eingang des Raspberry Pi angebunden ist.

Die Schaltung funktioniert wie folgt: Wird der Roboter eingeschaltet, ist das Not-AUS-Relais inaktiv, und die Antriebsmotoren sowie der Mähwerksmotor werden nicht mit Spannung versorgt. Durch Betätigen des RESET-Tasters wird das Not-AUS-Relais eingeschaltet, durch eine Selbsthaltung bleibt es auch nach Loslassen des RESET-Tasters angezogen. Die Antriebsmotoren bzw. die Schrittmotorsteuerung und der Mähwerksmotor kann nun von der Steuerung aus eingeschaltet werden. Wird nun der Not-AUS-Taster betätigt, wird die Selbsthaltung unterbrochen und das Notaus-Relais wird abgeschaltet. Wenn die Rastung des Not-AUS-Tasters gelöst wird, bleibt das Relais zunächst abgeschaltet, bis es wieder mit dem RESET-Taster freigeschaltet wird.

#### 4.4.3. Mähwerk

Der Schaltplan des Mähwerks ist in Abbildung 4.8 dargestellt. Das Mähwerk wird durch ein Relais K2 eingeschaltet. Als Besonderheit besitzt das Mähwerk einen eigenen, exklusiven Akku. Dies hat zur Folge, dass das Mähwerk komplett



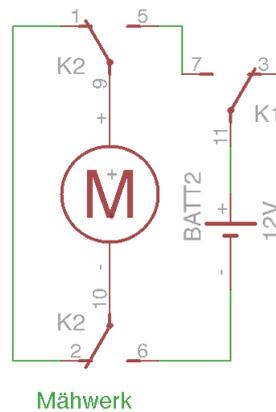


Abbildung 4.8.: Schaltplan des Mähwerks

#### 4.4.4. Spannungswandler

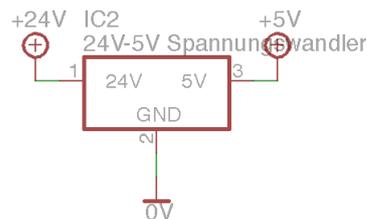


Abbildung 4.9.: Schaltplan des Spannungswandlers

Um aus den 24V des Akkus die für die Versorgung des Raspberry Pi und der Sensortechnik notwendigen 5V zu erhalten, ist ein kleiner DC/DC-Wandler (Abbildung 4.9) im Roboter eingebaut. Dieser kann ca. 5W an Leistung übertragen.

Die 5V-Versorgung ist unabhängig vom Zustand der Not-AUS-Schaltung, damit die Steuerung nicht in einen unbestimmten Zustand gebracht wird, und die Sensorik aktiv bleibt.

#### 4.4.5. I<sup>2</sup>C-Bus

Über den I<sup>2</sup>C-Bus werden die Ultraschallsensoren und der für die Induktionsschleife zuständige Microcontroller angesprochen und die Sensorwerte abgefragt. Hierbei war vor Allem zu beachten, dass die Signalspannung der Sen-

soren 5V beträgt, aber der I<sup>2</sup>C-Port des Raspberry Pi nur mit 3,3V umgehen kann. Daher wurden die Busleitungen mit einer Schaltung versehen, die die Spannungswerte umsetzen kann und dabei die digitalen Ein- und Ausgänge am Raspberry Pi schützt.

Die I<sup>2</sup>C-Busteilnehmer arbeiten mit Open-Collector-Ausgängen, so dass eine Spannungsanpassung der Datenübertragung vergleichsweise einfach realisierbar ist – einziges Problem ist die Bidirektionalität, wodurch Lösungen mit Optokopplern, welche die Busteilnehmer galvanisch voneinander trennen würden, aufwendig sind. Daher wird die Spannungsanpassung mit einer einfachen Transistorschaltung realisiert, die auf der Application Note AN97055 von Philips Semiconductors/NXP basiert [Sem].

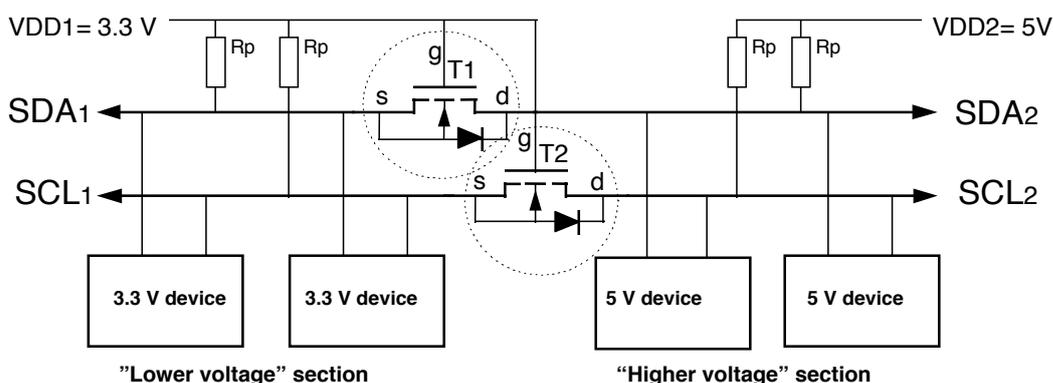


Abbildung 4.10.: einfache I<sup>2</sup>C-Pegelanpassung [Sem]

In Abbildung 4.10 ist der grundlegende Schaltplan der Spannungsanpassung dargestellt. Die Schaltung funktioniert nun wie folgt:

- Wenn kein Gerät auf den Bus schreiben will, sind alle Datenleitungen über die Pull-Up-Widerstände auf die jeweilige Versorgungsspannung gezogen. An den Gate- und den Source-Anschlüssen der MOS-FET-Transistoren T1 und T2 ist dadurch keine Spannungsdifferenz, die MOS-FET leiten nicht.
- Wenn ein Gerät auf der niedrigeren 3.3V-Seite auf eine Leitung schreiben will, wird der Source-Anschluss des jeweiligen MOS-FET auf 0V gezogen. Dadurch steigt die Gate-Source-Spannung  $V_{GS}$  über den Schwellwert des

Transistors und dieser wird leitend. Dadurch wird die zweite Seite mit der höheren Spannung ebenfalls auf 0V gezogen.

- Schreibt ein Gerät von der Seite mit der höheren Spannung auf eine Leitung, wird über die in den MOS-FET vorhandene Drain-Substrat-Diode die Leitung auf der Seite mit der niedrigeren Spannung ebenfalls auf 0V gezogen. Dadurch steigt die  $V_{GS}$  über den Schwellwert und er wird leitend.

Für die korrekte Funktion der Schaltung in Abbildung 4.10 muss die Spannung an beiden Seiten mindestens gleich sein, oder auf der zweiten Seite höher. Außer einer Spannungspegelanpassung ist mit dieser Schaltung auch gleichzeitig ein Schutz der niedrigeren Seite vor Spannungsspitzen realisiert, da nur bei einem Low-Pegel die MOS-FET leitend werden.

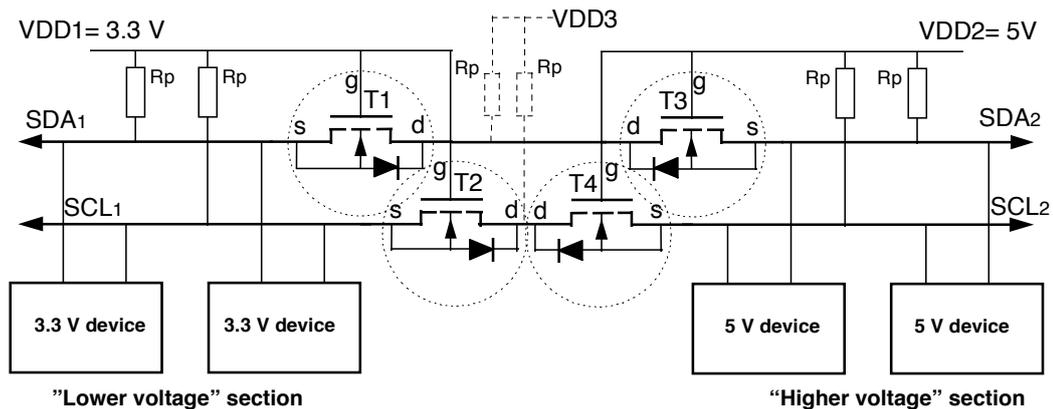


Abbildung 4.11.: I<sup>2</sup>C-Pegelanpassung mit Schutzisolierung [Sem]

Zusätzlich zur Spannungspegelanpassung ist in Abbildung 4.11 mit zwei weiteren Transistoren T3 und T4 eine Isolation der 3.3V-Seite von der 5V-Seite aus realisiert, die zum Schutz der 3.3V-Seite dient wenn die 5V-Seite abgeschaltet ist. Falls die Spannung VDD2 abgeschaltet ist, können die beiden MOS-FET nicht leitend werden. Ist die 3.3V-Spannung VDD1 abgeschaltet, sperren T1 und T2 den Stromfluss.

Die Spannung VDD3 ist nicht notwendig, diese dient dazu, eine eventuell auftretende Spannungsschwankung an den Drain-Anschlüssen der Transistoren abzufangen.

#### 4.4.6. Schrittmotorsteuerung

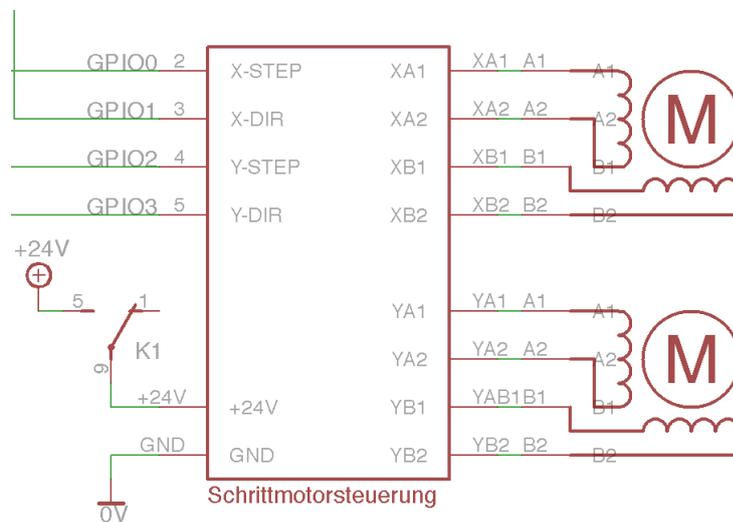


Abbildung 4.12.: Schaltplan der Schrittmotorsteuerung

Die Schrittmotorsteuerung ist an vier den digitalen Ausgängen des Raspberry Pi GPIO0-GPIO3 angeschlossen (Abbildung 4.12) und wird darüber gesteuert. Die weiteren Möglichkeiten, die die Steuerplatine bietet, werden derzeit nicht genutzt. Dies sind unter anderem die Möglichkeit, einen dritten Schrittmotor anzuschließen und die Ansteuerung zweier Relais.

An der Schrittmotorplatine können noch Einstellungen für die einzelnen Schrittmotoransteuerungen getätigt werden, so kann hier der maximale Phasenstrom, die Stromabsenkung im Stillstand und die Schrittteilung eingestellt werden. Diese müssen für beide Antriebsmotoren wie folgt eingestellt werden:

- Der Phasenstrom ist auf 0,75A eingestellt
- Die Stromabsenkung ist auf 100% eingestellt
- Die Mikroschritteinsteilung ist auf Vollschritt (1) eingestellt

#### 4.4.7. Taster und Signalleuchten

Zur Bedienung sind auf dem Roboter ein Start-Taster (Abbildung 4.14) und drei Leuchtdioden (Abbildung 4.13) angebracht. Damit die digitalen Ausgänge

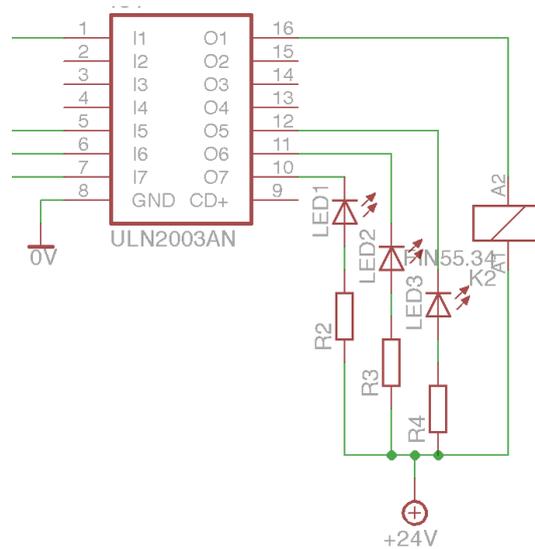


Abbildung 4.13.: Schaltplan der Leuchtdiodenansteuerung

des Raspberry Pi nicht durch den von den Leuchtdioden benötigten Strom überlastet werden, ist für deren Ansteuerung ein Ausgangstreiberbaustein ULN2003 zwischen die LED und den Ausgang geschaltet.

Neben dem START-Taster ist noch ein Kontakt des Not-AUS-Relais und eine Aluminiumleiste als Taster an den digitalen Eingängen des Raspberry Pi angeschlossen.

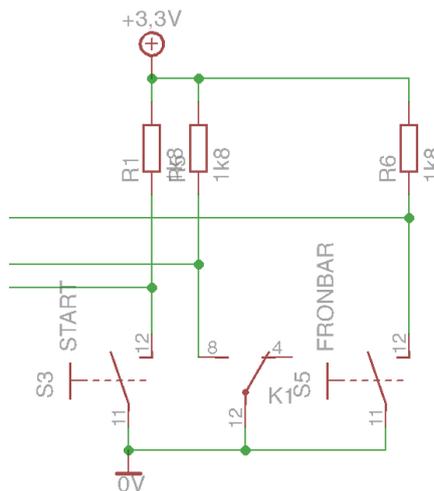


Abbildung 4.14.: START-Taster, Not-AUS-Kontrolle und Frontleiste

## **4.5. Steuerung**

Die Steuerung des Rasenmäherroboters besteht aus der Hardware, der Sensorik und der auf der Hardware laufenden Software.

### **4.5.1. Hardware**

Die zentrale Rolle in der Steuerung übernimmt eine Raspberry Pi-Platine, auf der ein Raspbian Wheezy Linux installiert ist.

### **4.5.2. Sensorik**

Der Rasenmäherroboter ist mit drei unterschiedlichen Typen von Sensoren ausgestattet. Zunächst sind hier vier Ultraschallsensoren zu nennen, die in einer Reihe 17cm über dem Boden montiert sind. Sie messen, ob sich ein Hindernis in Fahrtrichtung des Roboters befindet und melden dies über den I<sup>2</sup>C-Bus an die Steuerungssoftware. Da die Ultraschallsensoren prinzipbedingt eine Mindestentfernung zum Hindernis haben müssen, um es zu detektieren, ist zusätzlich eine Sicherheitsleiste oberhalb der Sensoren angebracht. Wenn diese von einem Hindernis eingedrückt wird, wird dies ebenfalls der Steuerungssoftware mitgeteilt.

Der zweite Typ Sensor ist ein virtueller Sensor. Dieser entsteht durch die verwendeten Schrittmotoren und dient zur Odometrie. Mit Hilfe der Schrittbefehle, die jedem Motor gegeben werden, kann die zurückgelegte Entfernung und Drehwinkel berechnet werden.

Der dritte Sensortyp ist ein Empfänger für eine induktive Begrenzungsschleife, welcher zweimal auf dem Roboter angebracht ist, einmal in Fahrtrichtung vorne rechts und einmal vorne links. Hier wurde zunächst eine Auswertungsschaltung hergestellt, mit der das Signal der umliegenden Schleife erkannt werden kann. Als Empfängerspule wurde eine Spule aus einem kleinen 24V-Relais extrahiert, diese hat einen elektrischen Widerstand von 900 Ohm. Die in der Spule induzierte Spannung wird zunächst von einem Differenzverstärker verstärkt, und anschließend über einen Schmitt-Trigger der Zustand, also ob sich

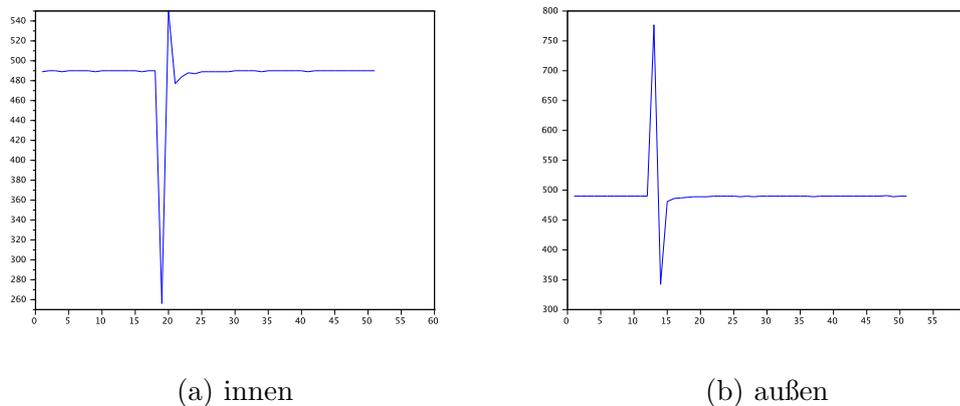


Abbildung 4.15.: zu erwartende Signalverläufe der Begrenzungsschleifensensoren

die Sensorspule innerhalb oder außerhalb der Sensorschleife befindet, als digitales Signal ausgegeben. In Funktionstests hat sich herausgestellt, dass diese Schaltung nur dann zuverlässige Ergebnisse produziert, wenn sich keine Störquellen in Form von sich verändernden Magnetfeldern in nächster Nähe zu den Sensorspulen befinden. Damit ist diese Schaltung für den Rasenmäherroboter unbrauchbar, da sowohl die Schrittmotoren als auch der Mähwerksmotor entsprechende Magnetfelder durch den Strom der Zuleitungen erzeugen.

Diese Problematik wurde umgangen, indem ein zweiter Empfänger aufgebaut wurde. Dieser ist schaltungstechnisch einfacher aufgebaut (siehe Abbildung 3.2), auch hier wird zunächst die in der Sensorpule induzierte Spannung von einem Differenzverstärker verstärkt. Danach wird das Signal aber nicht von einer analogen Schaltung ausgewertet, sondern einem Analog-Digitalwandler eines Microcontrollers, in diesem Fall ein Atmel ATmega328 auf einer Arduino Platine, zugeführt.

Diese Schaltung funktioniert im Labormaßstab, bei einer kleinen Fläche, zuverlässig. Beim Wechsel auf eine Rasenfläche hat sich gezeigt, dass die Schaltung außerhalb des Labormaßstabs nicht zuverlässig arbeitet. Grund hierfür können zusätzliche Störquellen sein, die im Labor nicht vorhanden sind: Feuchtigkeit des Rasens, Temperaturschwankungen, andere vertikale Abstände zwischen Sensorspule und Induktionsschleife.

Abbildung 4.15 zeigt die Signalverläufe der Begrenzungsschleifensensoren, die bei korrekter Funktion zu erwarten gewesen wären. Dabei ist die Sensor-  
spule innerhalb der Begrenzungsschleife, wenn zunächst der Spannungsverlauf  
steil abfällt und dann beim Abschalten des Stroms auf der Schleife stark über  
den im Ruhezustand anliegenden Wert ansteigt.

Außerhalb der Begrenzungsschleife sollte der Spannungsverlauf ein genau  
gegenläufiges Verhalten zeigen. Dabei ist aber zu beachten, dass der genaue  
Verlauf von der Polarität der Begrenzungsschleife abhängt, d. h. ob nun zuerst  
eine Spannungsspitze nach oben oder nach unten erfolgt, ist anpassbar durch  
einfaches Tauschen der Anschlüsse am Begrenzungsschleifensender.

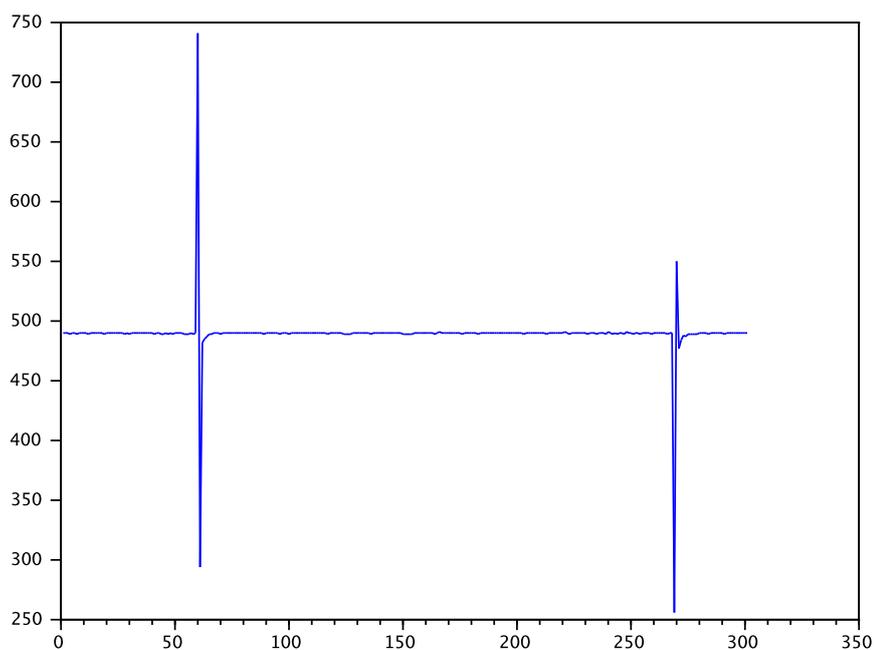


Abbildung 4.16.: Sensorlog der Begrenzungsschleife

Bei Fahrversuchen auf einer großen Rasenfläche, die mit einer Begrenzungsschleife von ca 3x4m Größe abgesteckt war, hat sich herausgestellt, dass diese Annahmen nur teilweise korrekt sind und die Aufnahmen der Sensoren zeigten, dass kein zuverlässiges, stabiles Muster messbar ist. In Abbildung 4.16

ist eine solche Datenreihe beispielhaft dargestellt, hier sieht man ein Datenlog über einen Zeitraum von etwa 100ms. Der Roboter stand 50cm von der Begrenzungsschleife entfernt innerhalb der Schleife, und die Sensordaten zeigen, dass keine Aussage darüber getroffen werden kann, auf welcher Seite der Begrenzungsschleife sich der Roboter befindet. Auch in längeren Datenlogs ist kein stabiler Zustand erkennbar, so dass ein Ausfiltern fehlerhafter Spannungsspitzen allein schon an der Menge der benötigten Messungen und der daraus resultierenden nicht genau feststellbaren Position der Sensorschleife scheitert.

### **4.5.3. Software**

Die Steuerungssoftware ist in der Programmiersprache C verfasst und läuft auf dem Linux-Betriebssystem, welches auf dem Raspberry Pi installiert ist.

Durch die langsame Umdrehungsgeschwindigkeit des Antriebsmotors sind keine Beschleunigungs- und Abbremsvorgänge bei diesen notwendig, dadurch kann zugunsten einer einfacheren Programmierbarkeit auf die Verwendung eines Echtzeitbetriebssystems verzichtet werden.

Für die Kommunikation mit den Sensoren, die am I<sup>2</sup>C-Bus angebracht sind, und die Ansteuerung der digitalen Ein- und Ausgänge des Raspberry Pi wird die Bibliothek „wiringPi“ benutzt. wiringPi stellt hierzu eine einfache API bereit, mit der Zustände der digitalen Ein- und Ausgänge eingelesen bzw. gesetzt werden können und übernimmt das gesamte Management der I<sup>2</sup>C-Kommunikation, so dass auch hier über einfach zu nutzende Befehle die Sensordaten abgefragt werden können oder Sensormessungen gestartet.

Im Programm selbst wurden für einzelne Aufgaben Methoden geschrieben, die von der Hauptschleife des Steuerungsprogramms aus aufgerufen werden können. Diese lassen sich in zwei Gruppen einteilen: einmal die Grundbefehle und dann darauf aufbauende Manöver.

Eine weiteres Programm befindet sich in einer Arduino-Microcontrollerplatine, dieses dient dazu, die von den Spulen der Begrenzungsschleifensensoren gemessenen Werte aufzubereiten und für den Raspberry Pi am I<sup>2</sup>C-Bus bereitzustellen.

```
1  /*****  
2  * step:  
3  * make one motor step  
4  *****/  
5  */  
6  void step(int direction) {  
7      switch (direction) {  
8          case LFRF:  
9              digitalWrite (XDIR, XFORWARD);  
10             digitalWrite (YDIR, YFORWARD);  
11             digitalWrite (XSTEP, 1);  
12             digitalWrite (YSTEP, 1);  
13             break;  
14             ...  
15             case LBRB:  
16                 digitalWrite (XDIR, XBACKWARD);  
17                 digitalWrite (YDIR, YBACKWARD);  
18                 digitalWrite (XSTEP, 1);  
19                 digitalWrite (YSTEP, 1);  
20                 break;  
21             default:  
22                 break;  
23             }  
24             usleep (1000);  
25             digitalWrite (XSTEP, 0);  
26             digitalWrite (YSTEP, 0);  
27             usleep (4000);  
28     }
```

Listing 4.1: Auszug aus der step()-Methode

### 4.5.3.1. Grundbefehle

Die Grundbefehle sind die Methoden, in denen die einzelnen Grundelemente realisiert wurden, aus welchen später komplexere Manöver zusammengesetzt werden können

**step(int direction)** Der atomare Vorgang der Fahrt des Mähroboters ist ein sogenannter Schritt. Als diesen bezeichnet man die Drehung eines Schrittmotors um seinen bauartbedingten Winkel, in diesem Fall sind es  $1,8^\circ$ . Hierzu muss der Ausgang des Raspberry Pi, an dem der STEP-Eingang der Schrittmotorsteuerung angeschlossen ist, einmal auf ein high-Signal und anschließend wieder auf low gebracht werden. Zusätzlich muss noch am DIR-Eingang der Schrittmotorsteuerung eine Information über die Drehrichtung angegeben werden, dies geschieht über einen weiteren Ausgang am Raspberry Pi.

Da der Rasenmäherroboter mit zwei Schrittmotoren ausgestattet ist, gibt es insgesamt neun verschiedene Schritttypen: links vorwärts rechts vorwärts, links Stopp rechts vorwärts, etc.. Diese Schritttypen werden über eine der step-Methode übergebene Variable angewiesen und entsprechend ausgewertet.

```

1  /*****
2  * routeFree:
3  * check Ultrasonic Sensors
4  * check Frontbar Switch
5  *****/
6  */
7  int routeFreeCounter = 0;
8  int valueS0, valueS1, valueS2, valueS3;
9  int valueS0l, valueS1l, valueS2l, valueS3l;
10 int routeFree() {
11     switch (routeFreeCounter++) {
12         case 0:
13             //Messung S0 starten
14             wiringPiI2CWriteReg8(sensorsrf0fd, 0, 0x51);
15             break;
16         case 1:
17             //Messung S1 starten
18             wiringPiI2CWriteReg8(sensorsrf1fd, 0, 0x51);
19             break;
20         case 2:
21             //Messung S2 starten
22             wiringPiI2CWriteReg8(sensorsrf2fd, 0, 0x51);
23             break;
24         case 3:
25             //Messung S3 starten
26             wiringPiI2CWriteReg8(sensorsrf3fd, 0, 0x51);
27             break;
28         case 4:
29             // Sensor S0 ersten Wert holen
30             valueS0l = wiringPiI2CReadReg8(sensorsrf0fd, 2) << 8;
31             break;
32         case 5:
33             // Sensor S0 zweiten Wert holen
34             valueS0 = valueS0l + wiringPiI2CReadReg8(sensorsrf0fd, 3);
35             if (valueS0 == 0) {
36                 routeFreeCounter -= 2;
37             }
38             break;
39         case 6:
40             // Sensor S1 ersten Wert holen
41             valueS1l = wiringPiI2CReadReg8(sensorsrf1fd, 2) << 8;
42             break;
43         case 7:
44             // Sensor S1 zweiten Wert holen
45             valueS1 = valueS1l + wiringPiI2CReadReg8(sensorsrf1fd, 3);
46             break;
47         case 8:
48             // Sensor S2 ersten Wert holen
49             valueS2l = wiringPiI2CReadReg8(sensorsrf2fd, 2) << 8;
50             break;
51         case 9:
52             // Sensor S2 zweiten Wert holen
53             valueS2 = valueS2l + wiringPiI2CReadReg8(sensorsrf2fd, 3);
54             break;
55         case 10:
56             // Sensor S3 ersten Wert holen
57             valueS3l = wiringPiI2CReadReg8(sensorsrf3fd, 2) << 8;
58             break;
59         case 11:
60             // Sensor S3 zweiten Wert holen
61             valueS3 = valueS3l + wiringPiI2CReadReg8(sensorsrf3fd, 3);
62             break;
63         default:
64             routeFreeCounter = 0;
65             break;
66     }
67     if (digitalRead(FRONTBAR) != 1) {
68         return 1;
69     }
70     if ((valueS0 > 25) && (valueS1 > 25) && (valueS2 > 25) && (valueS3 > 25)) {
71         return 1;
72     }
73     return 0;
74 }

```

Listing 4.2: routeFree()-Methode

```
1 /*****  
2  * checkInductiveLoop:  
3  * get inductive loop value  
4  *****/  
5 */  
6 int checkInductiveLoop() {  
7     //return wiringPiI2CRead(sensorinductived);  
8     return 42;  
9 }
```

Listing 4.3: checkInductiveLoop()-Methode

**routeFree()** In der routeFree()-Methode werden die Senordaten der Ultraschallsensoren über den I<sup>2</sup>C-Bus abgefragt, sollte die Fahrstrecke des Roboters in Fahrtrichtung durch Hindernisse blockiert sein, gibt die Methode einen Fehler zurück und die Antriebsmotoren bekommen keine weiteren Schrittbefehle zugeteilt.

Da aus Timinggründen nicht in jedem Antriebsschritt alle vier Ultraschallsensoren abgefragt werden können, ist dieses in mehrere Teile aufgeteilt und von diesen wird maximal einer je Motorschritt bei normaler Geradeausfahrt ausgeführt. Zunächst werden an jeden der vier Ultraschallsensoren je ein „Starte Messung“-Befehl gesendet, laut Datenblatt der Ultraschallsensoren dauert eine Messung etwa 20ms. Da die Messwerte als 16-bit-Wert in zwei Registern gespeichert werden, müssen danach für jeden Sensor zwei Abfragebefehle über den Bus geschickt werden. Die Werte werden dazu in Variablen gespeichert und später zur Auswertung zusammengefügt.

Nachteil an dieser Vorgehensweise ist, dass die Messwertspeichervariablen nach jeder Hindernisserkennung und in der Einschaltphase zuerst neu initialisiert werden müssen, da Fehler bis zu 20 Methodenaufrufe lang gespeichert werden.

**checkInductiveLoop()** In dieser Methode wird einfach nur ein Integer-Wert über den I<sup>2</sup>C-Bus von dem Mikrocontroller, welcher die Sensorschleifenmesswerte auswertet, geholt und zurückgegeben. Dieser kann vier verschiedene Werte annehmen:

- 39 beide Sensorspulen befinden sich außerhalb der Sendeschleife
- 40 die in Fahrtrichtung links angebrachte Sensorspule befindet sich außerhalb der Sendeschleife

```

1  /*****
2  * drive:
3  * drive robot, backwards if distance is negative
4  * returns if driving forward and route is not free
5  *****/
6  */
7  int drive(int distance){
8      int steps = distance * 96000 / 4712;
9      if (steps < 0) {
10         steps *= -1;
11     }
12     int i;
13     for (i = 0; i < steps; i++) {
14         //route not free, return unfinished steps
15         if ((distance > 0 && (checkInductiveLoop() < 42 || routeFree() != 1))
16             || digitalRead(EMERGENCYOFF) == 1 ) {
17             if (distance < 0) {
18                 return -1 * i * 96000 / 4712;
19             } else {
20                 return i * 96000 / 4712;
21             }
22         }
23         if (distance > 0) {
24             step(LFRF);
25         } else {
26             step(LBRB);
27         }
28     }
29     return 0;
30 }

```

Listing 4.4: drive()-Methode

41 die in Fahrtrichtung rechts angebrachte Sensorspule befindet sich außerhalb der Sendeschleife

42 beide Sensorspulen befinden sich innerhalb der Sendeschleife

Da die Messungen der Begrenzungsschleife sehr instabil sind und es nicht zuverlässig festgestellt werden kann, ob der Roboter sich inner- oder außerhalb der zu mähenden Fläche befindet, ist die Ausgabe zu Testzwecken auf 42 festgelegt, damit Manöverfahrten und das Mähwerk ohne störenden Einfluss getestet werden können.

#### 4.5.3.2. Manöver

Manöver sind aus einzelnen Schritten, anderen Manövern und Sensorabfragen zusammengesetzte Methoden.

**drive(int distance)** Nachdem in der step()-Methode die grundlegende Vorgehensweise für die Fortbewegung des Roboters gelegt wurde, ist die drive()-Methode eine Ebene abstrakter. Die drive()-Methode bekommt als Parameter eine Strecke in Zentimetern übergeben, die der Roboter in gerader Strecke vor-

```
1  /*****
2  * turn:
3  * rotate robot, clockwise if degrees is positive
4  *****/
5  */
6  int turn(int degrees){
7      int steps = degrees * 1254 / 90;
8      if (steps < 0) {
9          steps *= -1;
10     }
11     int i;
12     for (i = 0; i < steps; i++) {
13         //route not free, return unfinished steps
14         if ((checkInductiveLoop() < 42 || routeFree() != 1)
15             || digitalRead(EMERGENCYOFF) == 1) {
16             if (degrees < 0) {
17                 return -1 * i * 90 / 1254;
18             } else {
19                 return i * 90 / 1254;
20             }
21         }
22         if (degrees > 0) {
23             step(LFRS);
24         } else {
25             step(LSRF);
26         }
27     }
28     return 0;
29 }
```

Listing 4.5: turn()-Methode

oder zurückbewegt werden soll. Die genaue Strecke wird hierbei bestmöglich in die Anzahl Schritte umgerechnet, die notwendig ist um diese Strecke zurückzulegen. Dies ist durch den Durchmesser des Antriebsrades und der Übersetzung des Zahnriemengetriebes eine bekannte Konstante, eine Radumdrehung entspricht 960 Schritten und diese ist 471,2mm lang.

**turn(int degrees)** In der turn()-Methode bekommt der Roboter den Befehl, sich um einen Winkel zu drehen. Dabei kann eine positive oder eine negative Zahl übergeben werden, bei einer positiven Zahl dreht sich der Roboter im Uhrzeigersinn um das rechte Vorderrad. Dabei wird nur das kurvenäußere Rad angetrieben, der Mittelpunkt der Drehbewegung befindet sich also in der Mitte des nicht angetriebenen Rades.

**turnfast(int degrees)** Die turnfast()-Methode funktioniert ähnlich wie die turn()-Methode, hier werden jedoch beide Antriebsräder benutzt um die Drehung durchzuführen. Dadurch befindet sich der Mittelpunkt der Drehbewegung in der Mitte zwischen beiden Antriebsachsen und der Roboter kann sich fast auf der Stelle um sich selbst drehen.

```

1  /*****
2  * turnfast:
3  * rotate robot around its center, clockwise if degrees is positive
4  *****/
5  */
6  int turnfast(int degrees){
7      int steps = degrees * 627 / 90;
8      if (steps < 0) {
9          steps *= -1;
10     }
11     int i;
12     for (i = 0; i < steps; i++) {
13         //route not free, return unfinished steps
14         if ((checkInductiveLoop() < 42 || routeFree() != 1)
15             || digitalRead(EMERGENCYOFF) == 1) {
16             if (degrees < 0) {
17                 return -1 * i * 90 / 627;
18             } else {
19                 return i * 90 / 627;
20             }
21         }
22         if (degrees > 0) {
23             step(LFRB);
24         } else {
25             step(LBRF);
26         }
27     }
28     return 0;
29 }

```

Listing 4.6: turnfast()-Methode

```

1  /*****
2  * align:
3  * align robot to inductive loop
4  *****/
5  */
6  void align() {
7      delay(5); //wait for i2c activity
8
9      if (checkInductiveLoop() == 39) {
10         delay(5); //wait for i2c activity
11     } else if (checkInductiveLoop() == 40) {
12         //left outside
13         delay(5); //wait for i2c activity
14         while (checkInductiveLoop() != 39) {
15             step(LSRF);
16         }
17     } else if (checkInductiveLoop() == 41) {
18         //right outside
19         delay(5); //wait for i2c activity
20         while (checkInductiveLoop() != 39) {
21             step(LFRS);
22         }
23     } else if (checkInductiveLoop() < 39) {
24         //Error: Inductive Loop not ready
25         delay(5); //wait for i2c activity
26         while (checkInductiveLoop() != 42) {
27             delay(100);
28             digitalWrite (LED.ORANGE, 1);
29             delay(100);
30             digitalWrite (LED.ORANGE, 0);
31         }
32     }
33 }

```

Listing 4.7: align()-Methode

**align()** Die align()-Methode wird dazu benutzt, den Roboter bei Erreichen der Induktionsschleife senkrecht zu dieser Auszurichten. Dazu wird nachgeschaut, welche Seite des Roboters die Sensorschleife überschritten hat, und die andere Seite wird nachgezogen. Treten hierbei Fehler auf, so dass kein gültiger Wert vom Schleifensensor geliefert wird, bleibt der Roboter solange stehen, bis wieder gültige Werte der Sensorschleife eingelesen werden.

Diese Methode funktioniert nur dann, wenn die Sensorschleife auch korrekt arbeitet. Da die Sensorschleife aber sehr unzuverlässig arbeitet (siehe 4.5.2), ist diese Methode derzeit im Einsatz außerhalb des Labors nicht benutzbar.

#### **4.5.3.3. main()-Methode**

In der main()-Methode (Listing 4.8) ist das Verhalten des Roboters festgelegt. Zunächst werden die benutzen Ausgänge auf feste Startwerte gesetzt und das Mähwerk abgeschaltet, damit Fehler durch vorherige Programmdurchläufe ausgeschlossen werden. Danach wird zuerst gewartet, bis das Not-AUS-Relais mit dem RESET-Taster zurückgesetzt wird, danach wartet das Programm auf die Betätigung des START-Tasters.

Danach ist das Programm gestartet und der Roboter kann losfahren. In diesem Demonstrationsprogramm fährt der Roboter zunächst 15cm vorwärts, dreht sich dann um 90 Grad im Uhrzeigersinn und wiederholt das ganze viermal, so dass er ein Quadrat abfährt. Danach wird diese Fahrt mit einer um 15cm größer werdenden Fahrtstrecke wiederholt, bis eine Kantenlänge von 150cm erreicht ist.

Danach startet das Fahrtprogramm erneut. Das Programm wird abgebrochen und neugestartet, wenn während einer Vorwärts- oder Kurvenfahrt ein Fehler auftritt. Dann wird der Roboter um 30cm zurückversetzt und dreht sich um 30 Grad. Mögliche Fehler können sein: Betätigung des Not-AUS-Tasters, Meldung der Ultraschallsensoren über eine zu kurze Entfernung oder ein Auslösen des Tasters an der Frontseite.

```

1  /*****
2  * main:
3  * demo program
4  *****/
5  */
6  int main (void) {
7      int i = 0;
8      setup ();
9      int started = 0;
10     deactivateMower();
11     digitalWrite (LED_GREEN, 0);
12     digitalWrite (LED_RED, 1);
13     while( 1 ) {
14         //wait for emergency off relais
15         while (digitalRead(EMERGENCYOFF) == 1 && started == 0) {
16             delay(100);
17         }
18         //wait for start button, led red blinking
19         while (digitalRead(STARTBUTTON) == 1 && started == 0) {
20             step(LSRS);
21             delay(100);
22             digitalWrite (LED_RED, 1);
23             delay(100);
24             digitalWrite (LED_RED, 0);
25         }
26         digitalWrite (LED_RED, 0);
27         digitalWrite (LED_GREEN, 1);
28         if (started == 0) printf ("Started");
29         started = 1;
30         activateMower();
31         int restart = 0;
32         for (i = 15; i < 150; i+=15) {
33             int j = 0;
34             for (j=0; j<4; j++) {
35                 if (drive(i) != 0) {
36                     printf("Hallo");
37                     //error: deactivate mower and drive back
38                     deactivateMower();
39                     drive(-30);
40                     for (i=0; i<30; i++) {
41                         //reinit ultrasonic sensors
42                         routeFree();
43                         delay(20);
44                     }
45                     turn(30);
46                     for (i=0; i<30; i++) {
47                         //reinit ultrasonic sensors
48                         routeFree();
49                     }
50                     if (digitalRead(EMERGENCYOFF) != 1) started = 0;
51                     restart = 1;
52                     break;
53                 }
54                 if (turn(90) != 0) {
55                     //error: deactivate mower and drive back
56                     deactivateMower();
57                     drive(-30);
58                     for (i=0; i<30; i++) {
59                         //reinit ultrasonic sensors
60                         routeFree();
61                         delay(20);
62                     }
63                     turn(30);
64                     if (digitalRead(EMERGENCYOFF) != 1) started = 0;
65                     restart = 1;
66                     break;
67                 }
68             }
69             if (restart == 1) { //restart after break
70                 break;
71             } } }
72     return EXIT.SUCCESS;
73 }

```

Listing 4.8: main()-Methode

## 4.6. Gesamtaufbau des Roboters

In Abbildung 4.17 ist eine Skizze abgebildet, in der der Gesamtaufbau der einzelnen Baugruppen dargestellt ist. Diese Abbildung dient zur Orientierung bei der Suche nach einzelnen Komponenten und ist nicht maßstabsgetreu.

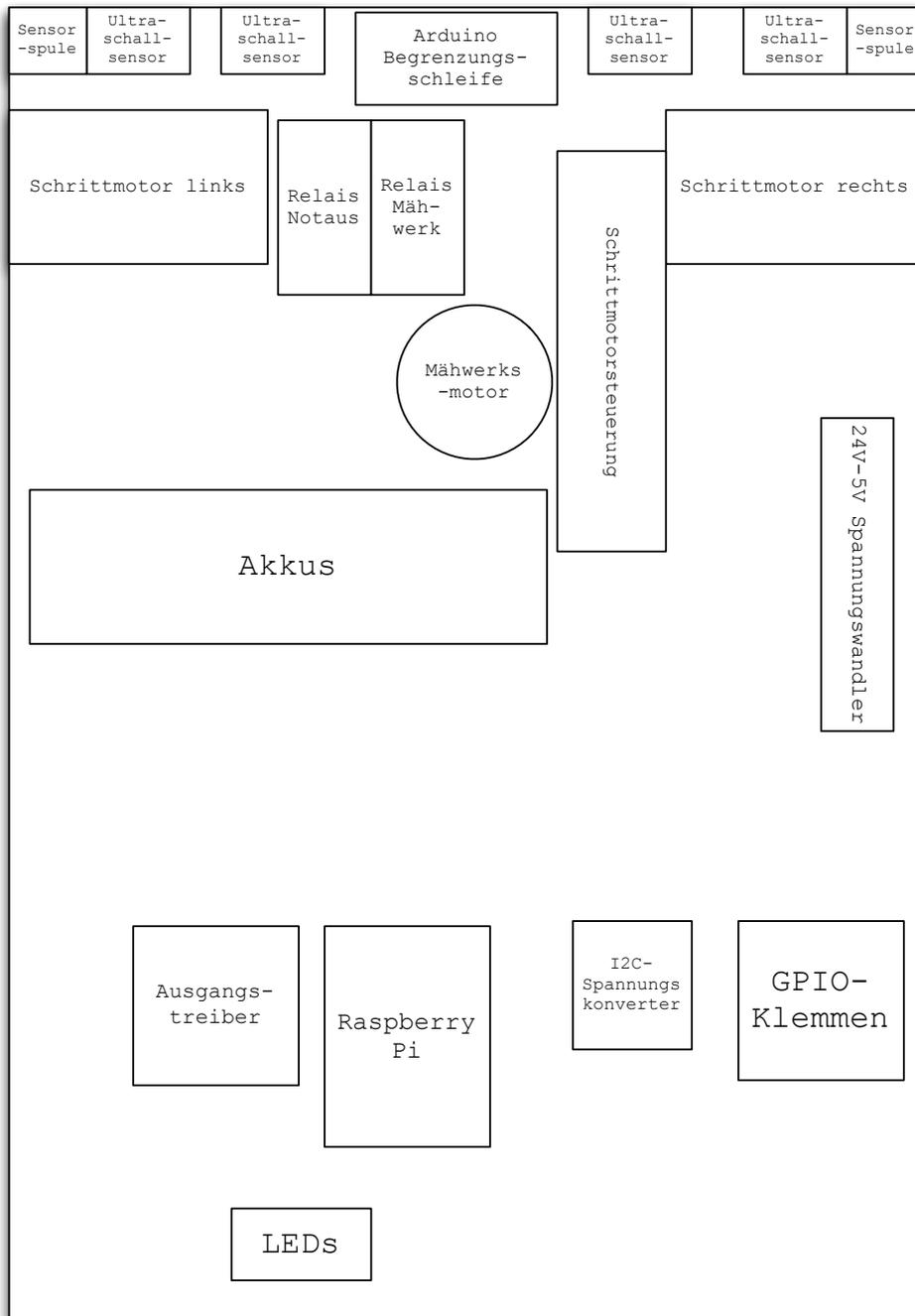


Abbildung 4.17.: Anordnung der einzelnen Bauelemente im Roboter

## **5. Stücklisten und Kostenberechnung**

Eins der Ziele dieser Diplomarbeit ist es, einen Rasenmäherroboter zu konstruieren und zu bauen, der eine kostengünstige Basis für die Weiterentwicklung von Navigationsalgorithmen bietet. In diesem Kapitel werden daher die ungefähren Kosten der einzelnen Bauteile aufgezeigt, indem für die jeweiligen Baugruppen getrennt voneinander Stücklisten erstellt werden. Am Ende des Abschnitts werden noch kurz Bauteile aufgelistet, die angeschafft, aber im Laufe der Entwicklung des Roboters nicht mehr benötigt wurden.

### **5.1. Chassis und Gehäuse**

Die wesentlichen Bauteile und Kosten des Chassis und des Gehäuses werden in Tabelle 5.1 aufgelistet.

### **5.2. Antrieb**

Die wesentlichen Bauteile und Kosten der Antriebsbaugruppe werden in Tabelle 5.2 aufgelistet. Zu beachten ist, dass der Preis der Antriebswelle nur geschätzt ist, da die Antriebswelle als einziges Bauteil der Baugruppe speziell für den Rasenmäherroboter angefertigt wurde.

Bauteil	Anzahl	Einzelpreis	Gesamtpreis
Aluprofil 30x30 Nut 8 1100mm	1	16 Eur	16 Eur
Aluprofil 30x60 Nut 8 1000mm	1	8 Eur	8 Eur
Winkel 30x30	4	2 Eur	8 Eur
Innenwinkel Nut 8	4	3 Eur	3 Eur
Abdeckkappen 30x60	5	1 Eur	5 Eur
Nutenstein M5	13	1 Eur	13 Eur
Bügelgriff M6 Nut 8	1	5 Eur	5 Eur
Siebdruckplatte 9mm 0,5m <sup>2</sup>			18 Eur
Kistenscharnier	2	2 Eur	4 Eur
Gesamt			80 Eur

Tabelle 5.1.: Stückliste der Baugruppe Antrieb

Bauteil	Anzahl	Einzelpreis	Gesamtpreis
Schrittmotoren 2,8A 2,0Nm	2	29 Eur	58 Eur
Schrittmotorsteuerung 3A	1	95 Eur	95 Eur
Festlagerblock	2	25 Eur	50 Eur
Zahnriemenscheibe 15 Zähne	2	7 Eur	14 Eur
Zahnriemenscheibe 72 Zähne	2	29 Eur	58 Eur
Zahnriemen 600mm 120 Zähne	2	6 Eur	12 Eur
Kugellager 6200 2RS	4	1 Eur	4 Eur
Kunststoff Abstandshalter	2	1 Eur	2 Eur
PVC-Reifen	2	3 Eur	6 Eur
Lenkrollen	2	10 Eur	20 Eur
Antriebswelle	2	25 Eur	50 Eur
Gesamt			369Eur

Tabelle 5.2.: Stückliste der Baugruppe Antrieb

Bauteil	Anzahl	Einzelpreis	Gesamtpreis
Getriebemotor 2,5:1 540er	1	23 Eur	23 Eur
Metallwinkel 50mm	1	1 Eur	1 Eur
Aluminiumrohr 8mm/6mm	1	3 Eur	3 Eur
Messernabe	1	9 Eur	9 Eur
Messerscheibe	1	13 Eur	13 Eur
Messer incl. Messerschraube	3	2 Eur	6 Eur
Gesamt			45 Eur

Tabelle 5.3.: Stückliste der Baugruppe Mähwerk

Bauteil	Anzahl	Einzelpreis	Gesamtpreis
Akku 12V 4500 mAh	3	49 Eur	147 Eur
Steckverbinder	3	1 Eur	3 Eur
Ladegerät	1	75 Eur	75 Eur
Gesamt			225 Eur

Tabelle 5.4.: Stückliste der Baugruppe Energieversorgung

### 5.3. Mähwerk

Die wesentlichen Bauteile und Kosten der Mähwerksbaugruppe werden in Tabelle 5.3 aufgelistet.

### 5.4. Energieversorgung

Die wesentlichen Bauteile und Kosten der Mähwerksbaugruppe werden in Tabelle 5.4 aufgelistet. Das Ladegerät für die Akkus war in der Abteilung bereits vorhanden, der geschätzte Anschaffungspreis ist hier der Vollständigkeit halber eingerechnet.

### 5.5. Sensorik

Die wesentlichen Bauteile und Kosten der Sensorik und des Begrenzungsschleifensenders werden in Tabelle 5.5 aufgelistet. Die Ultraschallsensoren waren be-

Bauteil	Anzahl	Einzelpreis	Gesamtpreis
Ultraschallsensoren SRF02	4	22 Eur	88 Eur
Arduino Uno (China Klon)	1	10 Eur	10 Eur
Lochrasterplatine	2	3 Eur	6 Eur
Relais 24V	2	2 Eur	4 Eur
Operationsverstärker	1	1 Eur	1 Eur
diskrete Bauelemente Sensor			2 Eur
Steckverbinder	10	1 Eur	10 Eur
Netzteil 24V	1	25 Eur	25 Eur
diskrete Bauelemente Sender			5 Eur
Gesamt			151 Eur

Tabelle 5.5.: Stückliste der Sensorik und Begrenzungsschleife

reits in der Abteilung vorhanden, der geschätzte Anschaffungspreis ist hier der Vollständigkeit halber eingerechnet.

## 5.6. Steuerungselektronik

Die wesentlichen Bauteile und Kosten der Steuerungselektronik und der Not-AUS-Schaltung werden in Tabelle 5.6 aufgelistet.

## 5.7. nicht benötigte Bauteile

In der Tabelle 5.7 sind die Kosten für gekaufte, aber am Ende nicht mehr benötigte Bauteile aufgelistet. Teilweise haben sich diese Bauteile ergeben, weil ein Mehrkauf zu niedrigeren Gesamtkosten geführt hat, oder sind während der Entwicklung des Roboters durch andere Bauteile ersetzt worden.

## 5.8. Gesamtkosten

Die Gesamtkosten des Roboters in Tabelle 5.8 ergeben sich aus den Kosten der einzelnen Baugruppen. Zusätzlich ist noch eine Pauschale für die Versandkosten der einzelnen Anbieter der Kaufteile und nicht genannte Kleinteile wie

Bauteil	Anzahl	Einzelpreis	Gesamtpreis
Raspberry Pi Modell B Rev. 1	1	40 Eur	40 Eur
WLAN Adapter	1	7 Eur	7 Eur
SD-Speicherkarte 8GB	1	10 Eur	10 Eur
Lochrasterplatine	1	3 Eur	3 Eur
Spannungswandler 24V-5V 1A	1	10 Eur	10 Eur
Kabelklemmen	6	2 Eur	12 Eur
diskrete Bauelemente			5 Eur
Koppelrelais	2	10 Eur	20 Eur
Kontaktgeber	3	5 Eur	15 Eur
Drucktaster Edelstahl	2	14 Eur	28 Eur
Not-AUS-Taster	1	9 Eur	9 Eur
Gesamt			159 Eur

Tabelle 5.6.: Stückliste der Steuerungselektronik und der Not-AUS-Schaltung

Bauteil	Anzahl	Einzelpreis	Gesamtpreis
Motorhalter NEMA23	2	18 Eur	36 Eur
Kugellager 6200	6	1 Eur	6 Eur
Innenwinkel	4	3 Eur	12 Eur
Nutensteine	7	1 Eur	7 Eur
Bügelgriff	1	5 Eur	5 Eur
Abdeckkappe	1	1 Eur	1 Eur
Relais 24V	8	2 Eur	16 Eur
Gesamt			83 Eur

Tabelle 5.7.: Liste der nicht benötigten Bauteilen

<b>Baugruppe</b>	<b>Kosten</b>
Chassis und Gehäuse	80 Eur
Antrieb	369 Eur
Mähwerk	45 Eur
Energieversorgung	225 Eur
Sensorik	151 Eur
Steuerungselektronik	159 Eur
Kleinteilepauschale	30 Eur
nicht benötigte Bauteile	Eur
Gesamt	1059 Eur

Tabelle 5.8.: Gesamtkosten des Roboters

Schrauben, Unterlegscheiben etc. hinzugefügt. Die Kosten der nicht benötigten Bauteile werden hier nochmals genannt, aber in der Gesamtsumme nicht berücksichtigt.

## 6. Zusammenfassung und Ausblick

In der jetzigen Form ist der Roboter in der Lage, in größer werdenden Quadraten über eine Rasenfläche zu fahren. Der Aufbau des Roboters ermöglicht auch eine leichte Erweiterbarkeit um weitere Sensoren, die dann eine genaue Positionsbestimmung auf der Rasenfläche ermöglichen werden. Möglichkeiten bestehen hier z. B. darin, einen Empfänger für Satellitennavigation einzusetzen, der gleichzeitig realzeitkinematikfähig ist, beispielsweise sei hier der in einem Kickstarterprojekt entwickelte Piksi RTK GPS Empfänger. Dieser bietet eine Genauigkeit im cm-Bereich und ist damit prädestiniert für Navigationsanwendungen im eigenen Garten.

Das Chassis aus leichten Aluminiumprofilen ist sehr flexibel und macht den Roboter leicht erweiterbar, das Gehäuse aus Siebdruckplatten macht den Roboter resistent gegenüber leichten Regenschauern. Allerdings ist durch den großen Innenraum und den nur locker mit Scharnieren und Klettband befestigten Deckel ein großer Resonanzkörper vorhanden, was den Roboter sehr laut macht.

Die Reifen des Roboters sind kostengünstig, aber haben auf Rasen nicht genügend Bodenhaftung, was zu Schlupf der Antriebsräder und damit zu Positionsverfälschungen in der Odometrie führen kann. Damit die großen Riemenscheiben an der Antriebsachse des Roboters nicht mehr über den Rasen schleifen, müssten sie zudem einen größeren Durchmesser besitzen. Die Räder sind zudem derzeit nur mit Federscheiben an der Achse geklemmt angebracht, die Räder mit Klebstoff zu befestigen würde aber die Wartbarkeit der Antriebsbaugruppe massiv verschlechtern.

Die Schrittmotoren ermöglichen eine einfache Ansteuerung aus der Software, man kann mit ihnen genaue Positionen anfahren ohne sich gleichzeitig harte Echtzeitanforderungen einzuhandeln, oder Probleme mit der Trägheit der

Drehbewegung zu bekommen. Allerdings besitzen Schrittmotoren im Vergleich zu Getriebemotoren nur ein geringes Drehmoment – Getriebemotoren mit angeflanschten Drehgebern bieten hier eine Alternative, mit der man sich allerdings die o.g. Probleme einhandelt.

Größtes Problem bei dieser Diplomarbeit war die Zuverlässigkeit der induktiven Begrenzungsschleife. Diese wurde zunächst im Labormaßstab zur Funktion gebracht, beim Wechsel auf eine größere Rasenfläche waren aber keine zufriedenstellenden, stabilen Messwerte erreichbar. Kommerzielle Rasenmäherroboter verwenden eine andere Technik für ihre Begrenzungsschleife als die in dieser Arbeit realisierte, welche statt auf dem Impulsverfahren auf einem sinusbasiertem Funkverfahren basiert. Dieses Verfahren bietet eine höhere Zuverlässigkeit, ist aber auch elektronisch anspruchsvoller und erfordert im Selbstaufbau eine genauere Kenntnis von Funktechniken. Da Sensoren für Begrenzungsschleifen nicht als selbststehende Sensoren verfügbar sind, würde es sich hier anbieten, die Begrenzungsschleife aus einem anderen Gerät zu demontieren und dieses dann als Fertigbauteil zu verwenden. Hier sind z. B. Halsbänder für Hunde und Katzen im Handel verfügbar.

## Literaturverzeichnis

- [ard] *A Complete Beginners Guide to the Arduino. : A Complete Beginners Guide to the Arduino*, [https://docs.google.com/file/d/0Bw\\_ruM0tRDDgNXI30TFGZXhIZ2c/edit](https://docs.google.com/file/d/0Bw_ruM0tRDDgNXI30TFGZXhIZ2c/edit), Abruf: Dienstag, 20. Mai 2014
- [avr04] *AVR Mikrocontroller Lehrbuch*. Roland Walter, 2004
- [bel] *Belrobotics Football Ground Mowers*. <http://www.belrobotics.com/football/index.php/de/>, Abruf: 20.05.2014
- [bos] *Der Roboter-Mäher Indego von Bosch mäht den Rasen systematisch und in Bahnen*. <http://www.bosch-presse.de/presseforum/details.htm?txtID=5813>
- [ele] *Elektronik Kurs*. <http://elektronik-kurs.net/>, Abruf: Freitag, 23. Mai 2014
- [hana] *Husqvarna Automower™ 230 ACX/220 AC Bedienungsanleitung. : Husqvarna Automower™ 230 ACX/220 AC Bedienungsanleitung*
- [hanb] *Robomow Gebrauchsanweisung & Sicherheitshandbuch RM200. : Robomow Gebrauchsanweisung & Sicherheitshandbuch RM200*
- [his] *Automower History*. <http://robomowerwiki.com/index.php/Automower>, Abruf: 19.05.2014
- [HLN12] HERTZBERG, Joachim ; LINGEMANN, Kai ; NÜCHTER, Andreas: *Mobile Roboter. Eine Einführung aus Sicht der Informatik*. 1. Springer-Vieweg, Berlin, Heidelberg, 2012 (eXamen.press). <http://www.springer.com/computer/ai/book/978-3-642-01725-4>. – ISBN 978-3-642-01725-4

- [ind] *Begrenzungsschleife - Induktionsschleife.* [http://www.rn-wissen.de/index.php/Begrenzungsschleife\\_-\\_Induktionsschleife](http://www.rn-wissen.de/index.php/Begrenzungsschleife_-_Induktionsschleife)
- [KHH09] KALTSCHMITT, M. ; HARTMANN, H. ; HOFBAUER, H.: *Energie aus Biomasse: Grundlagen, Techniken und Verfahren.* Springer, 2009 <http://books.google.de/books?id=QpMM93jkficC>. – ISBN 9783540850946
- [Neu11] NEUDÖRFER, Alfred: *Konstruieren sicherheitsgerechter Produkte.* 4. Auflage. Springer-Verlag, 2011
- [ras] *The Raspberry Pi Education Manual. : The Raspberry Pi Education Manual,* [http://pi.cs.man.ac.uk/download/Raspberry\\_Pi\\_Education\\_Manual.pdf](http://pi.cs.man.ac.uk/download/Raspberry_Pi_Education_Manual.pdf), Abruf: 19.05.2014
- [rob] *Robomow FAQ.* <http://www.powermow.com/FAQ.htm>, Abruf: 19.05.2014
- [sch] *Der Schrittmotor, eine Homepage von Georg Sauer und Benjamin Winheim.* <http://schrittmotor.gonline.at/homepage/funktionsweise.html>, Abruf: Freitag, 23. Mai 2014
- [Sem] SEMICONDUCTORS, Philips: *Application Note AN97055: Bi-directional level shifter for P<sup>2</sup>C-bus and other systems.*
- [Wen07] WENDEL, J.: *Integrierte Navigationssysteme: Sensordatenfusion, GPS und Inertiale Navigation.* Oldenbourg, 2007. – ISBN 9783486581607

# A. Anhang

## A.1. Bedienungsanleitung

### A.1.1. Benutzung des Roboters

Um den Roboter zu benutzen, stellt man ihn einfach auf die zu mähende Rasenfläche. Dann schaltet man den Roboter am Hauptschalter ein und wartet, bis an der LED-Anzeige am hinteren Ende des Roboters die rote, rechte LED aufleuchtet.

Ist dies geschehen, muss der Not-AUS zurückgesetzt werden. Falls am Not-AUS-Taster der grüne Ring nicht sichtbar ist, muss man den Taster durch eine Drehung entsperren. Dann kann man durch Betätigen des RESET-Tasters das Not-AUS zurücksetzen, der Roboter ist jetzt fahrbereit. Angezeigt wird dies mit der rechten, roten LED, diese beginnt zu blinken, sobald das Not-AUS entsperrt ist.

Nach betätigen des START-Tasters beginnt der Roboter, sein Programm abzufahren. Derzeit ist nur ein Demonstrationsprogramm im Roboter installiert, er beginnt, größer werdende Vierecke zu fahren, bis zu einer Kantenlänge von 150cm.

### A.1.2. Aufladen der Akkus

Um die Akkus aufzuladen, müssen diese aus dem Roboter entnommen werden. Hierzu muss zunächst die Deckelplatte des Roboters aufgeklappt werden. Danach können die beiden Klettbänder, die die drei Akkus gegen Verrutschen sichern, gelöst werden (Nr. 1 in Abbildung A.1). Danach können die Kontakte

der beiden Steuerungs- und Fahrtakkus gelöst werden (Nr. 2 in Abbildung A.1). Die obersten beiden Akkus können dann entnommen werden. Als letztes kann der Mähwerksakku entnommen werden, dieser liegt mit den Anschlüssen genau andersherum als die beiden darüberliegenden Akkus (Nr. 3 in Abbildung A.1).

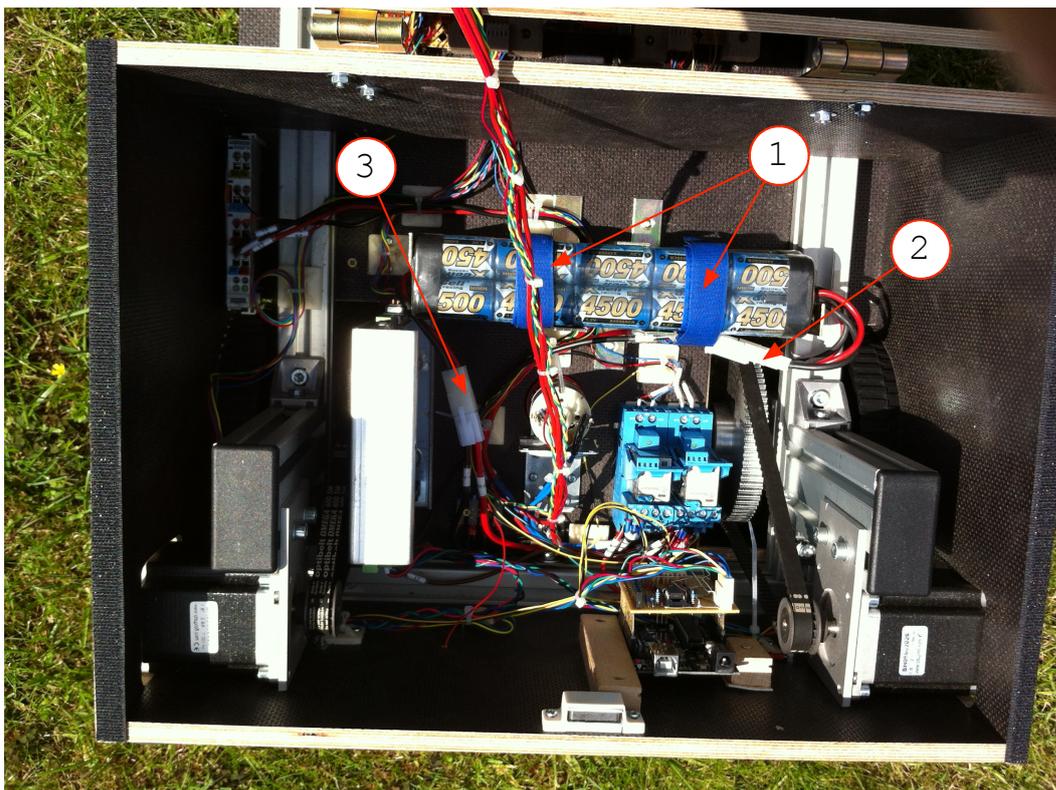


Abbildung A.1.: Entfernen der Akkus aus dem Roboter

Die Akkus müssen dann nacheinander am Ladegerät aufgeladen werden. Hierzu wird ein Akku am Ladeadapter angesteckt, und die beiden Kontakte des Ladeadapters in das Ladegerät gesteckt. Danach wird das Ladegerät eingeschaltet, hier ist darauf zu achten, dass die Akkukapazität von 4500 mAh und der Akkutyp als NiMH eingestellt ist. Der Ladevorgang startet, indem man die Taste ENTER drückt und festhält, bis das Ladegerät einen kurzen Piepton von sich gibt. Nachdem die Ladung des Akkus abgeschlossen ist, gibt das Ladegerät erneut einen Piepton aus, und es schaltet auf Erhaltungsladung, um den Akku nicht zu beschädigen.

### A.1.3. Wechsel der Mähmesser

Um die Mähmesser zu wechseln, müssen aus Sicherheitsgründen zunächst die Akkus aus dem Roboter entnommen werden (siehe A.1.2). Dann kann der Roboter auf eine Seite gelegt werden, so dass das Mähwerk zugänglich ist.

Zunächst müssen die alten Messer entfernt werden. Hierzu mit Handschuhen die Mähscheibe (Nr. 2 in Abbildung A.2) festhalten und mit einem 4,5mm Schlitz- oder einem PH2-Schraubendreher die drei Messerschrauben (Nr. 1 in Abbildung A.2) lösen. Mit den Schrauben lösen sich auch die Mähmesser. Neue Mähmesser können dann montiert werden, indem man eine Messerschraube durch die Bohrung im Messer steckt und die Messerschraube dann wieder an der Mähscheibe befestigt. Hierbei ist darauf zu achten, dass die Messerschrauben vollständig eingedreht werden, ansonsten kann sich das Messer zwischen Mähscheibe und Messerschraube verkanten und den Roboter beschädigen.

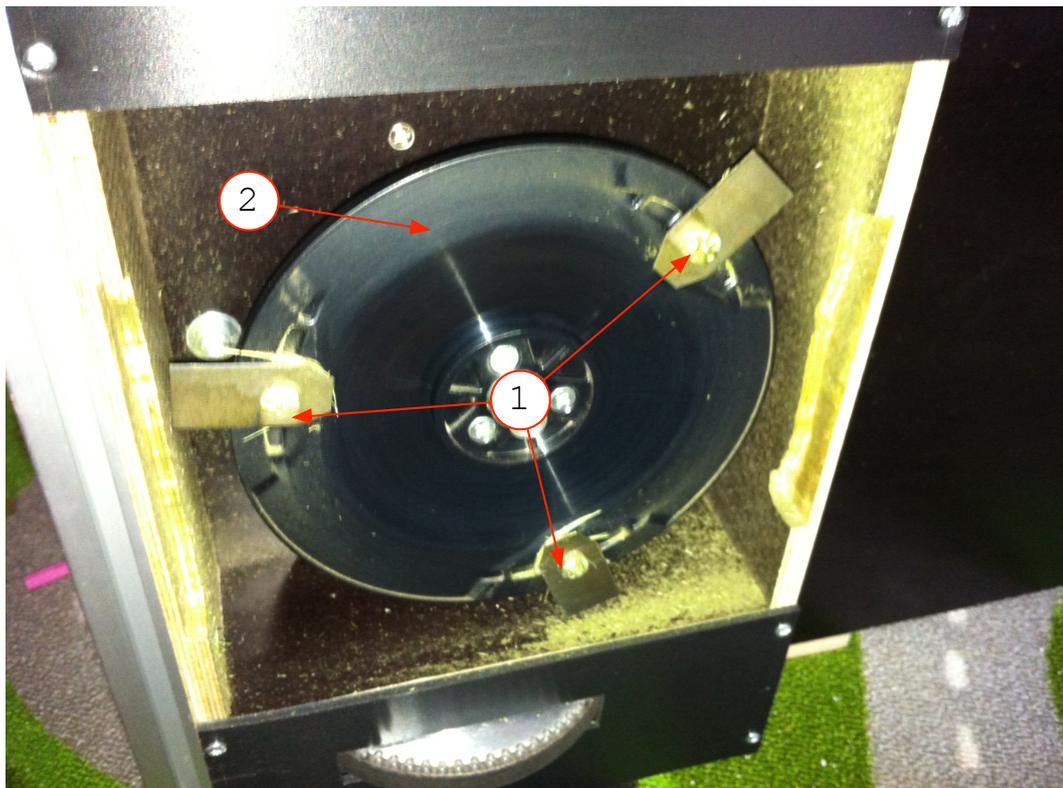


Abbildung A.2.: Mähwerk des Roboters

### A.1.4. SD-Karte des Roboters neu erstellen

Das Erstellen einer neuen SD-Karte für den Rasenmäherroboter kann auf zwei Wege geschehen. Eine Möglichkeit ist, die Image-Datei der SD-Karte zu benutzen, die auf der dieser Arbeit beiliegenden DVD vorliegt. Hierzu muss man nur das Image „ShaunRaspbian.dmg“ mit einer geeigneten Software auf eine mindestens 4GB große SD-Karte aufspielen, und erhält eine fertige Umgebung, welche allerdings kein kompiliertes Steuerungsprogramm enthält.

Ein Abbild mit einem Demoprogramm, welches den Rasenmäherroboter in größer werdenden Quadraten fahren lässt, ist in der Image-Datei „ShaunSquares.dmg“ ebenfalls auf der DVD vorhanden.

Sollte kein Zugriff auf eine der beiden oben genannten Imagedateien möglich sein, ist es ebenfalls möglich, das System von Grund auf neu aufzusetzen. Hierzu benötigt man neben einer SD-Karte mit mindestens 4GB Speicherkapazität einen kabelgebundenen Internetzugang für den Raspberry Pi, einen am Raspberry angeschlossenen Monitor und eine Tastatur. Hierfür muss man folgende Schritte durchführen:

1. Installieren eines Raspbian-Linux auf der SD-Karte.
2. Einsetzen der SD-Karte in den Raspberry Pi, durchführen des Erstininstallationsassistenten
3. Installation des `git-core`- und des `libi2c-dev`-Pakets mittels `apt-get`:

```
1 sudo apt-get install git-core libi2c-dev
```

4. Herunterladen und kompilieren von `wiringPi`:

```
1 git clone git://git.drogon.net/wiringPi
2 cd wiringPi
3 ./build
```

5. Laden des I<sup>2</sup>C-Kernelmoduls:

```
1 sudo gpio load i2c
```

```

1  DEBUG = -O3
2  CC = gcc
3  INCLUDE = -I/usr/local/include
4  CFLAGS = $(DEBUG) -Wall $(INCLUDE) -Winline -pipe -I/usr/include/SDL -D_GNU_SOURCE=1 -
   D_REENTRANT
5  LDFLAGS = -L/usr/local/lib -L/usr/lib/arm-linux-gnueabi -lSDL
6  LDLIBS = -lwiringPi -lpthread -lm
7  SRC = shaun.c shaun.h
8  OBJ = $(SRC:.c=.o)
9  BINS = $(SRC:.c=)
10 shaun: shaun.o
11     @echo [link]
12     @$ (CC) -o $$@ shaun.o $(LDFLAGS) $(LDLIBS)
13 all:
14     @echo "    $(BINS)" | fmt
15     @echo ""
16 really-all: $(BINS)
17 .c.o:
18     @echo [CC] $<
19     @$ (CC) -c $(CFLAGS) $< -o $$@
20 clean:
21     rm -f $(OBJ) *~ core tags $(BINS)
22 tags: $(SRC)
23     @echo [ctags]
24     @ctags $(SRC)
25 depend:
26     makedepend -Y $(SRC)

```

Listing A.1: Makefile zur Kompilierung der C-Steuerungsprogramme

### A.1.5. Kompilieren neuer Programme

Nachdem wie in Abschnitt A.1.4 gezeigt ein bootfähiges System erstellt wurde, kann mit dem gcc-compiler aus dem C-Quelltext eines Steuerungsprogramms eine lauffähige Version kompiliert werden. Hierzu kann der gcc-Befehl einfach mit:

```
1 gcc -Wall -o blink blink.c -lwiringPi
```

aufgerufen werden, nachdem man den Quelltext auf den Roboter kopiert und sich über ssh auf diesem eingeloggt hat. Alternativ benutzt man ein Makefile, welches man sich nach dem Vorbild von Listing A.1 erstellen kann.

### A.1.6. Netzwerkzugang

Auf dem Sender der Begrenzungsschleife ist ein WLAN-Router montiert, welcher ein Funknetz mit den folgenden Zugangsdaten aufbaut

```

1 SSID: Shaun WLAN
2 PASSWD: uln2003an
3 TYPE: WPA2

```

In dieses WLAN loggt sich der Roboter selbstständig ein, die Ethernetschnittstelle des Raspberry Pi ist ebenfalls eingerichtet. Der Roboter ist dann über SSH erreichbar:

```
1 IP-LAN: 192.168.43.2
2 IP-WLAN: 192.168.42.3
3 NETMASK: 255.255.255.0

5 LOGIN: pi
6 PASSWD: pi
```

---

# Erklärung zur Urheberschaft

Hiermit versichere ich, dass ich diese Arbeit selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Außerdem versichere ich, dass die allgemeinen Prinzipien wissenschaftlicher Arbeit und Veröffentlichung, wie sie in den Leitlinien guter wissenschaftlicher Praxis der Carl von Ossietzky-Universität festgelegt sind, befolgt habe.

Oldenburg, 30.05.2014

Ort, Datum

Unterschrift