



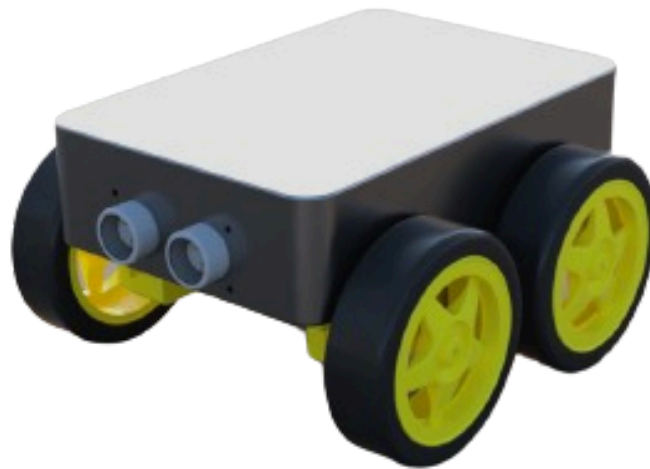
Universität Rostock
Fakultät für Informatik und Elektrotechnik
Fachbereich Elektrotechnik

Wintersemester 2023/24
Lehrveranstaltung Projekt Elektrotechnik

Iheb Mrabet

Max Balbach

Schildkröte



1. Themenstellung.....	3
2. Themenpräzisierung.....	3
3. Einleitung.....	3
3.1. Gerätebeschreibung.....	3
3.2. Funktionsbeschreibung.....	3
3.3. Einsatzmöglichkeiten des Gerätes.....	4
4. Mechanische Baugruppen.....	4
4.1. Gehäuseaufbau.....	4
4.2. Dimensionierung des Gehäuses.....	4
4.3. Konstruktive Lösung/Variantendiskussion.....	5
4.4. Einordnung und Befestigung der Baugruppen im Gehäuse.....	5
4.5. Schutzgrad und Schutzklasse.....	5
4.6. Preis und Stückliste des Gehäuses.....	7
5. Elektronische Baugruppen.....	7
5.1. Schaltungsaufbau.....	7
5.2. Spannungsversorgung.....	7
5.3. Leiterplattenkonstruktion.....	8
5.3.1. Herstellung der Leiterplatte.....	8
5.3.2. Leiterplattenbestückung.....	8
5.4. Betrachtung der Wärmeabfuhr.....	8
5.5. Preis und Stückliste der Elektronik.....	8
6. Programmierung.....	9
6.1. Beschreibung.....	9
6.2. PIN-Verbindungen.....	10
7. Testen des Gerätes.....	10
7.1. Aufgetretene Fehler - Ursachen.....	10
7.2. Fehlerbetrachtung.....	11
7.3. Kalibrierung der Skalen.....	11
8. Elektromagnetische Verträglichkeit.....	11
9. Umweltverträglichkeit und Entsorgung.....	11
10. Zusätzliche Erläuterungen zum Beleg.....	12
11. Eigenständigkeitserklärung.....	12
12. Literaturverzeichnis / Verwendete Software.....	12
Literaturverzeichnis.....	12
Verwendete Software.....	13
Präsentationen.....	13
13. Anhang.....	14
1. Schemazeichnung des Schaltplans.....	14
2. Aufbau der Platine.....	14
3. Technische Aufbau.....	14
4. Fertige Aufbau.....	16

1. Themenstellung

Im Rahmen des vorliegenden Studienprojektes wurde die Entwicklung eines autonomen Roboters, genannt "Schildkröte", durchgeführt. Der Schwerpunkt lag dabei auf der Implementierung von Sensorik zur Erfassung der Umgebung und der autonomen Navigation, basierend auf den Grundlagen der Robotik, Konstruktion, Elektronik und Programmierung.

2. Themenpräzisierung

Das Projekt "Schildkröte" verfolgt das Ziel, einen kostengünstigen, autonomen Roboter zu entwickeln, der sich für Bildungszwecke oder als Grundlage für weiterführende Entwicklungen eignet. Durch den Einsatz von Standardkomponenten und Open-Source-Programmierung soll eine modulare und anpassungsfähige Plattform geschaffen werden.

3. Einleitung

Die "Schildkröte" ist ein Projekt, das aus der Notwendigkeit heraus entstanden ist, eine Plattform zu haben, die nicht nur lehrreich ist, sondern auch bei der Durchführung verschiedener Aufgaben in verschiedenen Umgebungen verwendet werden kann.

3.1. Gerätebeschreibung

Die "Schildkröte" ist ein autonomer Roboter, der auf einem erstellten Hülle aufgebaut ist. Die Hauptsteuereinheit ist ein Genuino 101, der die Interaktion zwischen Software und Hardware ermöglicht. Der Roboter ist mit einem Ultraschallsensor HC-SR04 für die Entfernungsmessung, einem vierrädrigen Antriebssystem, betrieben durch DC-Motoren, und einem Batteriesystem für die Energieversorgung ausgestattet. Der Aufbau beinhaltet auch zwei Motor-Treiber zur Steuerung der DC-Motoren, Batterien, eine selbstgebaute Leiterplatte zur Verbindung der Komponenten, und ein manuelles Steuerelement (Schalter).

3.2. Funktionsbeschreibung

Die Hauptfunktion der "Schildkröte" ist die autonome Navigation. Der Ultraschallsensor misst kontinuierlich den Abstand zu Hindernissen und sendet die Daten an den Genuino 101. Die Programmlogik, die in den Arduino eingebettet ist, entscheidet, wie der Roboter basierend auf diesen Daten navigiert, einschließlich wann er stoppt, seine Richtung ändert oder seine Geschwindigkeit anpasst. Der Knopf ermöglicht es dem Benutzer, die Bewegung des Roboters manuell zu steuern oder anzuhalten.

3.3. Einsatzmöglichkeiten des Gerätes

Die "Schildkröte" repräsentiert ein vielseitiges robotisches Konzept mit Potenzial für erweiterte Anwendungen. Sie könnte beispielsweise für logistische Aufgaben modifiziert werden, wie den automatisierten Transport von Waren. Eine Anpassung für Reinigungsprozesse ermöglicht ihren Einsatz in verschiedenen Umgebungen. Durch Integration in größere Automatisierungsprojekte, etwa durch Verbindung mit einem Roboterarm, kann sie diverse präzise Handhabungsaufgaben erfüllen. Ihre Flexibilität und Anpassungsfähigkeit eröffnen neue Möglichkeiten in der Robotik-Forschung und praktischen Anwendung.

4. Mechanische Baugruppen

Dieser Abschnitt befasst dich mit den mechanischen Baugruppen des Gerätes, vordergründig mit dem Gehäuse, deren Dimensionen, Aufbau und Umsetzung.

Die technische Zeichnung befindet sich am Ende des Projekts.

4.1. Gehäuseaufbau

Das Gerät verfügt über ein vollständig mit einem 3D-Drucker hergestelltes Gehäuse, dessen maximale Außenabmessungen 150 mm × 80 mm × 53 mm (B×T×H) betragen. Als integraler Bestandteil des Designs wurde eine Leiterplatte, die direkt als Arduino-Shield dient, auf dem Arduino-Board montiert. Die Konstruktion des Gehäuses umfasst ferner spezielle Vorrichtungen zur Befestigung der Motorantriebe und des Ultraschallsensors sowie Aussparungen für die Durchführung der Verkabelung.

4.2. Dimensionierung des Gehäuses

Bei der Planung der Gehäuseabmessungen stand die sichere Unterbringung aller Komponenten im Fokus. Es wurde sorgfältig darauf geachtet, dass zwei 9V-Batterien mit Clips, vier DC-Motoren mit Rädern, der Arduino mit aufgesetztem Shield und ein Ultraschallsensor Platz finden. Die Breite des Gehäuses orientiert sich an der Summe der Breiten der beiden Motorantriebe zuzüglich eines angemessenen Abstands für die Verkabelung. Die Länge musste ausreichend sein, um den Arduino, die Motorantriebe und die Batterien nebeneinander anzuordnen. Die Höhe wurde so bemessen, dass sie nicht nur die Motorantriebe und Batterien aufnehmen kann, sondern im unteren Bereich auch die Höhe der DC-Motoren plus einem Sicherheitsabstand berücksichtigt. Darüber hinaus wurde im Design ein Stützeck vorgesehen, um das Gewicht des Roboters tragen zu können und zu gewährleisten, dass das Gehäuse auch unter Last nicht beschädigt wird.

4.3. Konstruktive Lösung/Variantendiskussion

Bei der konstruktiven Umsetzung des Gehäuses durch den 3D-Druckprozess wurde besonderes Augenmerk auf die Optimierung des Materialverbrauchs gelegt. Die Herausforderung bestand darin, die Stabilität des Gehäuses sicherzustellen, während gleichzeitig der Einsatz von Stützmaterial minimiert wurde, insbesondere da das Design eine große Hülle aufweist. Durch den Druck vom Unterbereich und die kompakte Form des Gehäuses konnte das Stützmaterial auf ein Minimum reduziert werden. Dies war nicht nur kosteneffizient, sondern auch umweltschonend, da weniger Abfallmaterial anfiel. Die Konstruktion der Stützecken wurde speziell angepasst, um die Lasten des Roboters zu tragen, ohne die gesamte Integrität des Designs zu beeinträchtigen. So entstand ein Gehäuse, das sowohl die Anforderungen an die Belastbarkeit als auch an die Wirtschaftlichkeit im 3D-Druckprozess erfüllt.

4.4. Einordnung und Befestigung der Baugruppen im Gehäuse

Die Anordnung der Baugruppen im Inneren des Gehäuses wurde strategisch geplant, um einen optimalen Schutz für die elektronischen Bauteile zu bieten und eine benutzerfreundliche Bedienung des Geräts zu ermöglichen. Während die meisten Komponenten mittels Schrauben und Muttern fest im Gehäuse verankert wurden, erfolgte die Befestigung des Ultraschallsensors durch eine formschlüssige und reibungs basierte Verbindung, die eine sichere und dennoch demontierbare Lösung darstellt. Für den Schalter wurde eine passgenaue Öffnung im Gehäuse vorgesehen, die eine direkte und werkzeuglose Montage ermöglicht. Die Batterieclips wurden mit Klettband fixiert, was eine stabile Stromversorgung gewährleistet und zugleich einen schnellen Batteriewechsel ohne zusätzliche Werkzeuge erlaubt. Diese spezifischen Befestigungsmethoden tragen dazu bei, die einzelnen Bauteile effektiv vor Vibrationen und Stoß-Einwirkungen zu schützen und die Wartung des Geräts zu vereinfachen.

Jedoch sind der Arduino und die Motorantriebe jeweils nur mit 2 Schrauben verbunden, obwohl normalerweise 4 Schrauben verwendet werden sollten. Dies liegt daran, dass die Motoren auf der anderen Seite und darunter platziert sind, was die Nutzung einer vollständigen Schraubenverbindung unmöglich macht. Für den Prototyp reichen 2 Schrauben aus, aber für zukünftige Versionen müssen alternative Verbindungsmethoden oder eine andere Platzierung in Betracht gezogen werden.

4.5. Schutzgrad und Schutzklasse

Das Design des Gehäuses zielt darauf ab, die internen Komponenten effektiv vor mechanischen Einwirkungen, wie Stößen durch Kollision mit Objekten, sowie vor schädlichen Umwelteinflüssen wie Staub und

Feuchtigkeit zu schützen. Der Schutz der Elektronik und die Betriebssicherheit des Geräts stehen im Vordergrund, um das Risiko von Beschädigungen zu minimieren. Entsprechend der IP-Norm DIN EN 60529 wurde das Gehäuse so konzipiert, dass es den Anforderungen des Schutzgrades IP22 gerecht wird, was bedeutet, dass es gegen das Eindringen von Fremdkörpern größer als 12,5 mm und gegen fallendes Tropfwasser geschützt ist, wenn das Gehäuse bis zu 15° geneigt ist. Gemäß der Norm DIN VDE 0702-1 fällt das Gerät in die Schutzklasse 1, was vorschreibt, dass es mit einem Schutzleiter ausgestattet ist, der im Fehlerfall einen ausreichenden Schutz gegen elektrischen Schlag bietet. Diese Schutzmaßnahmen gewährleisten, dass die Sicherheit und Langlebigkeit des Geräts auch unter anspruchsvollen Bedingungen erhalten bleiben.


















Zweite Kennziffer		Schutzgrad	Erste Kennziffer		Schutzgrad
		Kurzbeschreibung			Kurzbeschreibung
0		Nicht geschützt	0		Nicht geschützt
1		Geschützt gegen Tropfwasser	1		Geschützt gegen feste Fremdkörper größer als 50 mm
2		Geschützt gegen Tropfwasser, wenn das Gehäuse bis zu 15° geneigt ist.	2		Geschützt gegen feste Fremdkörper größer als 12 mm
3		Geschützt gegen Sprühwasser	3		Geschützt gegen feste Fremdkörper größer als 2,5 mm
4		Geschützt gegen Spritzwasser	4		Geschützt gegen feste Fremdkörper größer als 1 mm
5		Geschützt gegen Strahlwasser	5		Staubgeschützt
6		Geschützt gegen starkes Strahlwasser	6		Staubdicht
7		Geschützt gegen die Wirkungen beim zeitweiligen Untertauchen in Wasser			
8		Geschützt gegen die Wirkungen beim dauernden Untertauchen in Wasser			
9		Geschützt gegen Hochdruck und hohe Strahlwassertemperaturen			

Abbildung 1: IP-Norm DIN EN 60529

Schutzklasse	0	I	II	III
Symbol				
Eigenschaft	Neben der Grundisolation kein besonderer Schutz	Schutzleitersystem (wird im Schadensfall zuletzt getrennt)	Schutzisolierung (verstärkte oder doppelte Isolierung)	Schutzkleinspannung

Abbildung 2: DIN VDE 0702-1

4.6. Preis und Stückliste des Gehäuses

Bezeichnung	Maße (mm)	Referenz	Stück	Preis (€)	Gesamt (€)
Gehäuse	150/80/53	3D Gedruckte	2	—	—
Klettband	—	—	—	—	—
Schrauben + Muttern	—	—	14	—	—

5. Elektronische Baugruppen

Im folgenden Abschnitt geben wir einen Einblick in unsere Planung und Umsetzung der elektronischen Bauelemente, sowie die Entscheidungen, die wir getroffen haben.

5.1. Schaltungsaufbau

Die Schaltung setzt sich aus den beiden, im Anhang befindlichen Plänen (Schaltplan und Layout) zusammen. Grob zusammengefasst ist eine Genuino 101 mit einem Ultraschallsensor, den wir verwendet haben, um den Abstand zu Objekten auf der Fahrbahn zu ermitteln, den zwei DC Motoren zum Antrieb der Räder und den beiden Batterien verbunden. Zwischen dem Ground-Anschluss der Genuino und den beiden Minuspolen der Batterien befindet sich ein Schalter, mit dem der Roboter ein- und ausgeschaltet werden kann.

5.2. Spannungsversorgung

Die Spannungsversorgung erfolgt über zwei neun Volt Batterien. Die beiden Batterien haben eine zusammenhängende Masse, wobei der Pluspol der ersten Batterie mit Eingang für die Spannungsversorgung der Genuino verbunden ist und der Pluspol der anderen mit den beiden

DC Motoren. Sie sind über zwei Batterieclips mit dem Gehäuse verbunden. Da die Verbindung über ein Klettverschlussband erfolgt, stellt das Wechseln der Batterien kein Problem dar.

5.3. Leiterplattenkonstruktion

Die folgenden Abschnitte beinhalten die Entwicklung der Leiterplatte von der Planung bis hin zur Herstellung.

5.3.1. Herstellung der Leiterplatte

Das Layout der Leiterplatte ist im Anhang zu finden und wurde mit dem Programm "Eagle" entworfen. Die Produktion dieser erfolgte im Leiterplattenlabor der Universität Rostock, Fachbereich Informatik und Elektrotechnik.

5.3.2. Leiterplattenbestückung

Die Leiterplatte wurde gemäß des Schaltplans und dem dazu entsprechenden Leiterplattenlayouts bestückt. Die einzelnen Bauelemente wurden durch manuelles Lötén auf der Leiterplatte befestigt, bzw. über die Leiterplatte mit der Genuino verbunden.

5.4. Betrachtung der Wärmeabfuhr

Die meiste Wärme bei diesem Aufbau entsteht in den beiden DC Motoren. Diese haben, um dem entgegenzuwirken, zwei passive Kühlkörper aufgebaut.

5.5. Preis und Stückliste der Elektronik

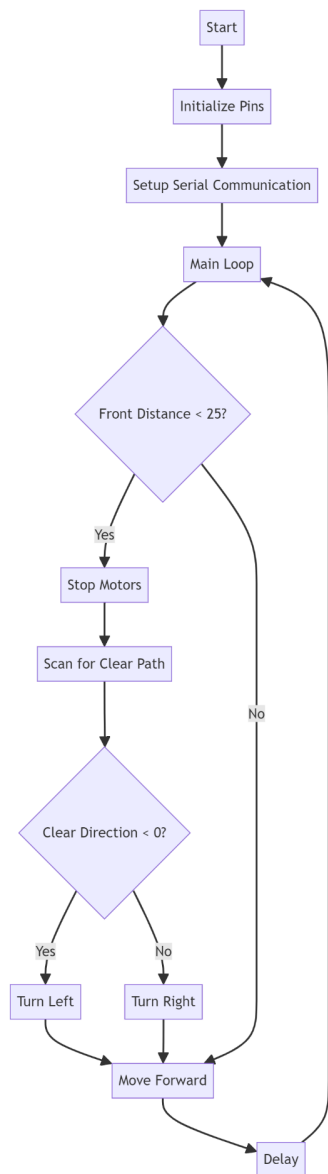
Bezeichnung	Maße (mm)	Referenz	Stück	Preis (€)	Gesamt (€)
Arduino	65/53	Genuino 101	1	42	42
Ultraschallsensor	—	HC-SR04	1	3.6	3.6
Batterien	—	9V Batterie	2	1	2
Batterie Clip	—	—	2	—	—
Leiterplatte	53/53	—	1	—	—
Schalter	13/9	—	1	1	1
Motorantrieb L298N	43/43	KeeYees	2		

DC-Motor	21/18	Motorantrie b Controller Board	4	20	20
Räder	—		4		
Kabel	—	—	—	—	—

6. Programmierung

Zur Programmierung wird die Arduino IDE verwendet.

6.1. Beschreibung



Der Code ermöglicht es einem Roboter, autonom zu navigieren, indem er Hindernisse erkennt und vermeidet. Mithilfe eines Ultraschallsensors misst der Roboter die Entfernung zu Objekten vor ihm. Wenn ein Hindernis näher als 25 cm ist, stoppt der Roboter, sucht nach einem freien Weg, indem er sich links und rechts dreht, und fährt dann in die Richtung mit dem meisten freien Raum weiter. Der Roboter nutzt Motoren für die Bewegung und kann vorwärts fahren, anhalten, sowie nach links oder rechts drehen.

6.2. PIN-Verbindungen

PIN-Bezeichnung	Verbindungstyp	Beschreibung	PIN-Nummer
in1	OUTPUT	Steuert einen der Motoren	2
in2	OUTPUT	Steuert einen der Motoren	3
in3	OUTPUT	Steuert einen der Motoren	4
in4	OUTPUT	Steuert einen der Motoren	5
trigPin	OUTPUT	Sendet Ultraschallsignale	13
echoPin	INPUT	Empfängt die reflektierten Ultraschallsignale	12

Die Tabelle zeigt, welche PINs am Mikrocontroller für Motorsteuerung und Ultraschallsensorik zuständig sind, inklusive ihrer Funktionen und PIN-Nummern.

7. Testen des Gerätes

Dieser Abschnitt befasst sich mit dem Testen und Kontrollieren des fertiggestellten Geräts und der Fehlerbetrachtung aufgetretener Fehler.

7.1. Aufgetretene Fehler - Ursachen

Der erste Fehler trat schon bei der Entwicklung des ersten Leiterplattenentwurfs auf. Die beiden Verbindungen der Enable A und B Anschlüsse der beiden DC Motoren wurden zuerst falsch verbunden, was wir leider erst bemerkten, als diese schon fertig hergestellt war und wir mit dem Löten begonnen hatten.

Des weiteren wurde die Spannungs nach ersten Belastungstest auf eine neun Volt Batterie festgelegt, was sich jedoch nach weiteren genaueren und längeren Test als zu wenig, bei unter anderem unebenen Boden herausstellte, bzw. die Laufzeit des Roboters auf eine sehr kurze Zeit beschränkt war. Dieses Problem ließ sich allerdings schnell beheben, indem eine weitere neun Volt Batterie hinzugefügt wurde.

7.2. Fehlerbetrachtung

Es können verschiedene Fehler auftreten:

Zum einen kann es passieren, dass ein Hindernis auf der Fahrbahn nicht erkannt wird, wenn es sich in einem ungünstigen Winkel zu dem Ultraschallsensor befindet, bzw. zu schmal ist, um genügend Ultraschallwellen zu reflektieren.

Des Weiteren kann es sein, dass sich der Roboter in einer Umgebung befindet, in der sich zu viele Objekte befinden und es außerdem nicht viel Platz gibt. Dafür ist die Genauigkeit des Ultraschallsensors nicht hoch genug, sodass es passieren kann, dass der Roboter immer wieder vor und zurück fährt, wodurch er dann nicht mehr aus dieser Umgebung kommt.

7.3. Kalibrierung der Skalen

Die Genauigkeit des Ultraschallsensors kann überprüft werden, indem man eine Referenz, wie zum Beispiel ein Lineal unter den Sensor legt und dann in einem bestimmten Abstand ein Objekt positioniert.

Nach vielen Stichproben kann festgehalten werden, dass der Sensor bis ungefähr 300 cm mit einer Genauigkeit von $\pm 0,5$ cm messen kann. Das stimmt relativ genau mit dem Datenblatt des Herstellers überein. In diesem steht, dass der Sensor bis zu 3 m mit einer Genauigkeit von 3 mm messen können soll.

8. Elektromagnetische Verträglichkeit

Es wurde keine Überprüfung zur elektromagnetischen Verträglichkeit durchgeführt, jedoch ist davon auszugehen, dass in diesem Zusammenhang keine Probleme auftreten sollten. In dem Aufbau werden keine größeren Frequenzen erzeugt, die zur Störung anderer Geräte führen könnten.

Die gewöhnlichen elektrischen Felder, die in unserem Alltag auftreten, durch zum Beispiel Handys, Funkmasten, Radios, Stromleitungen und so weiter, haben keine messbare Auswirkung auf die Funktionsweise des Roboters.

9. Umweltverträglichkeit und Entsorgung

Die Bauelemente unseres Geräts wurden online und aus dem Uni-Labor erworben, wobei besonderer Wert auf Umweltstandards gelegt wurde. Bei der Auswahl der Bauteile wurde darauf geachtet, dass diese am Ende ihrer Nutzungsdauer einfach zu entsorgen sind. Speziell die Batterien können problemlos bei jeder Batterie-Annahmestelle abgegeben werden. Das Gehäuse, gefertigt mittels 3D-Druck aus umweltfreundlichen und recycelbaren Kunststoffen, unterstreicht unser

Engagement für Nachhaltigkeit und minimiert den ökologischen Fußabdruck des Projekts. Diese Materialwahl erleichtert nicht nur die präzise und ressourcenschonende Produktion, sondern gewährleistet auch, dass das Gehäuse am Ende seines Lebenszyklus dem Recycling zugeführt werden kann.

Die kontinuierliche Einhaltung der ROHS-Richtlinien für die online gekauften Bauelemente ist vorgesehen, um die Umweltverträglichkeit unseres Projekts zu sichern. Durch diese Maßnahmen wird sichergestellt, dass unser Gerät nicht nur in seiner Nutzung, sondern auch in seiner Entsorgung oder Recycling-Phase die Umwelt möglichst wenig belastet.

10. Zusätzliche Erläuterungen zum Beleg

Es wurde überlegt, wie die Bauteile in zwei Bauteilgruppen unterteilt werden können. Es wird unterschieden zwischen den Bauteilen, die in dem Schaltplan vorkommen und nicht vorkommen.

So wurde das Gehäuse, die Räder, das Klettverschlussband, die schrauben und Muttern, sowie die Batterieclips zu der Baugruppe des Gehäuses zugeordnet und die anderen Bauteile, also die Geduino, die DC Motoren, der Motorantrieb, die Leiterplatte die Kabel, die Batterien und der Schalter, der elektronischen Baugruppe zugeordnet.

11. Eigenständigkeitserklärung

Hiermit erklären wir, Iheb Mrabet und Max Balbach, den Konstruktionsbeleg eigenständig und mit den nachstehend angegebenen Quellen verfasst zu haben.

12. Literaturverzeichnis / Verwendete Software

Literaturverzeichnis

1. <https://www.osram.de/media/resource/HIRES/342329/technischer-anwendungsleitfaden---schutzarten-ip-codes-nach-din-en-60529-d.pdf>
2. https://www.amazon.de/KeeYees-Motorantrieb-Controller-Schrittmotor-Intelligente/dp/B07ZT5CCFV/ref=d_dp-upsell-widget_sccl_2_1/259-1141299-6641801?pd_rd_w=Q6hGw&content-id=amzn1.sym.da6a3555-3917-42ee-92a6-9f4fea00ae66&pf_rd_p=da6a3555-3917-42ee-92a6-9f4fea00ae66&pf_rd_r=WSZ7CBBQNHTWX27PCJ9M&pd_rd_wg=bETCa&pd_rd_r=2694147d-7372-4f9c-8f86-dd98e1a8b7e0&pd_rd_j=B07ZT5CCFV&psc=1

3. <https://elektro.turanis.de/html/prj121/index.html>
4. <http://eagle.autodesk.com/eagle/release-notes>
5. <https://www.autodesk.de/products/fusion-360/overview?term=1-YEAR&tab=subscription>
6. <https://www.arduino.cc/en/software>

Verwendete Software

1. Google Docs, 09.03.2006, Google LLC
2. Autodesk Eagle 9.6.2, 27.05.2020, CadSoft Computer GmbH
3. Fusion 360, 22.06.2023, Autodesk
4. Arduino IDE 2.3.1, Arduino
5. Canva

Präsentationen

Erste Präsentation:

https://www.canva.com/design/DAF0ck08wLY/KyTvZa1R2t_ZWuLOCTrIWg/edit

Zweite Präsentation:

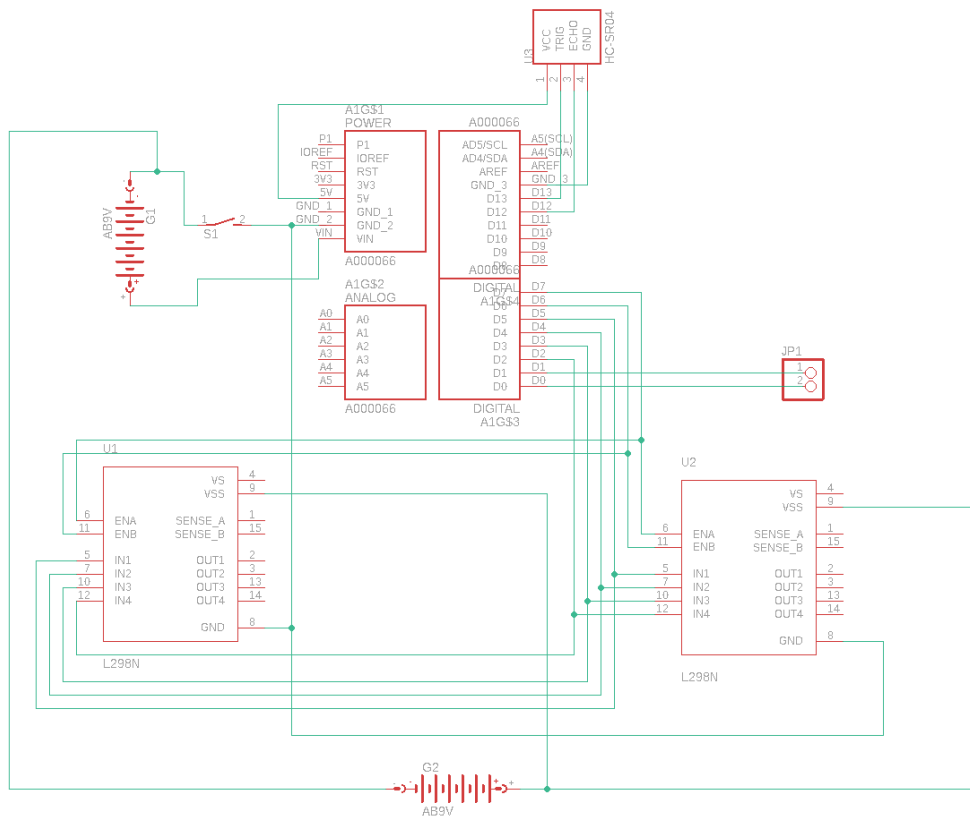
<https://www.canva.com/design/DAF3NwGtKNY/gO03OJdvzaFCXvBd5bw7GQ/edit>

Finale Präsentation:

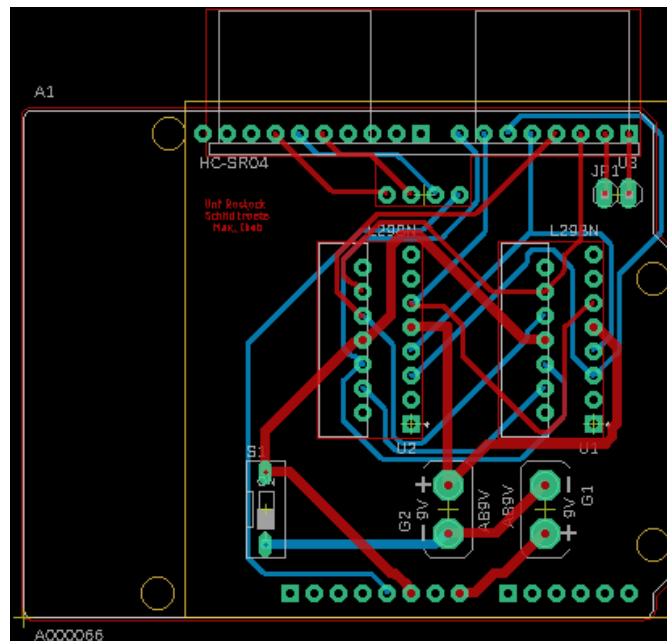
https://www.canva.com/design/DAF6i0d_T4A/OAscPg_fHkqIX4RX6LJBFW/edit

13. Anhang

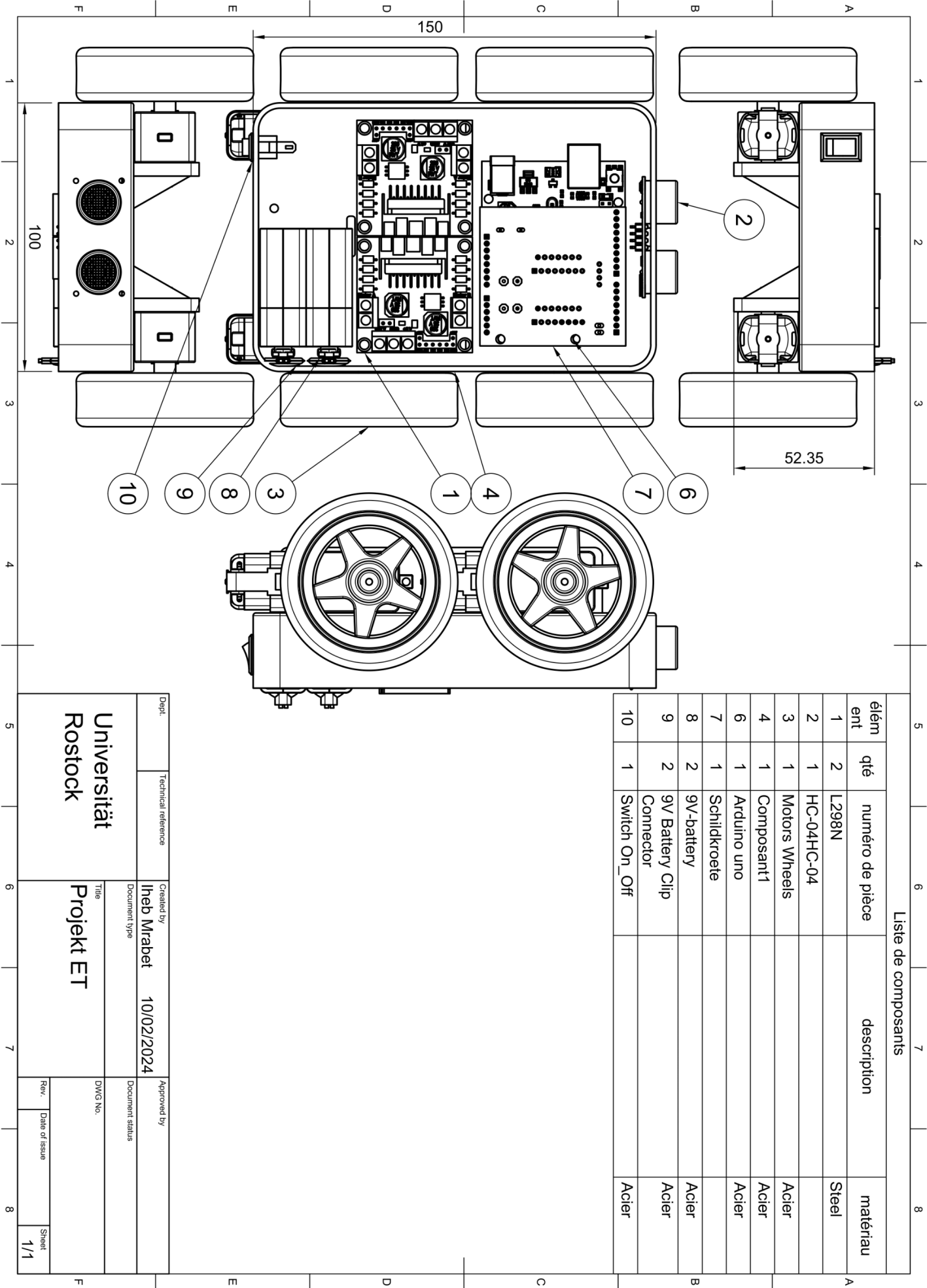
1. Schemazeichnung des Schaltplans



2. Aufbau der Platine

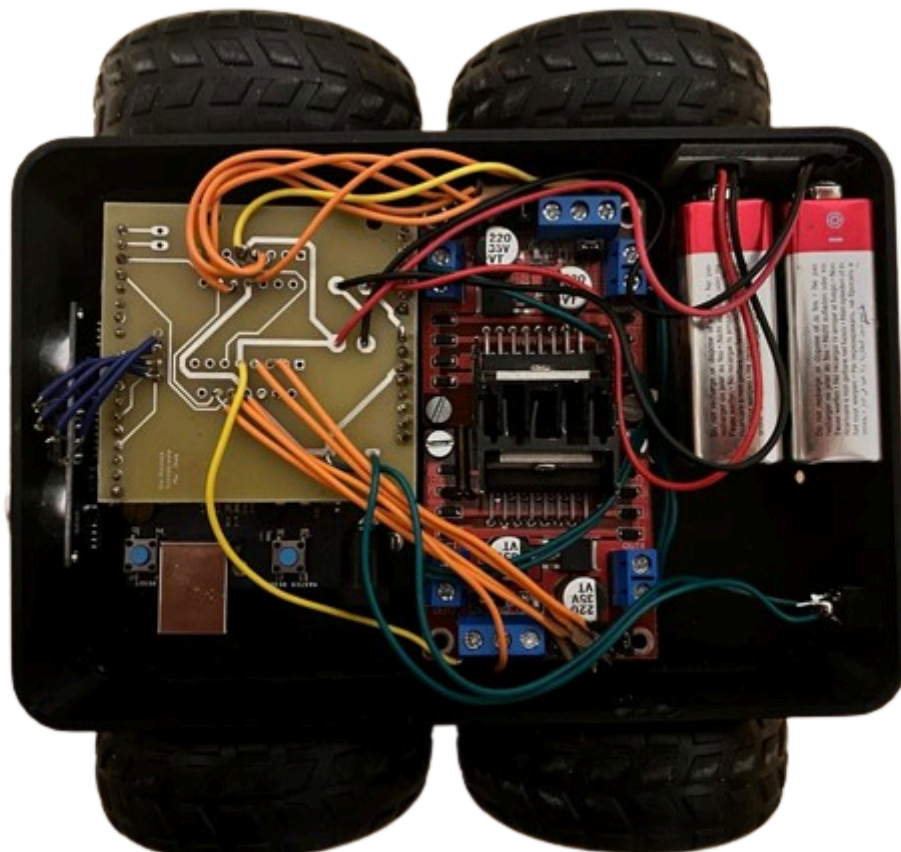


3. Technische Aufbau



Dept.	Technical reference	Created by	10/02/2024	Approved by
		Ineb Mrabet		
		Document type		Document status
Universität Rostock		Title	Projekt ET	DWG No.
		Rev.	Date of issue	Sheet
				1/1

4. Fertige Aufbau




```

int in1 = 2;
int in2 = 3;
int in3 = 4;
int in4 = 5;
int trigPin = 13;
int echoPin = 12;

void setup() {
    pinMode(in1, OUTPUT);
    pinMode(in2, OUTPUT);
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(trigPin, OUTPUT);
    pinMode(echoPin, INPUT);
    Serial.begin(9600);
}

void loop() {
    int frontDistance = getDistance();
    if (frontDistance < 25) {
        stopMotors();
        delay(500);
        int clearDirection = scanForClearPath();
        if (clearDirection < 0) {
            turnLeft(90);
        } else {
            turnRight(90);
        }
        moveForward();
        delay(1000);
    } else {
        moveForward();
    }
}

int getDistance() {
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    long duration = pulseIn(echoPin, HIGH);
    Serial.print("Distance: ");
    Serial.println(duration * 0.034 / 2);
    return duration * 0.034 / 2;
}

void moveForward() {
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
}

void moveBackward() {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
}

void stopMotors() {
    digitalWrite(in1, LOW);
    digitalWrite(in2, LOW);
    digitalWrite(in3, LOW);
    digitalWrite(in4, LOW);
}

int scanForClearPath() {
    int maxDistance = 0;
    int clearDirection = 0;
    turnLeft(45);

```

```

    int leftDistance = getDistance();
    if (leftDistance > maxDistance) {
        maxDistance = leftDistance;
        clearDirection = -1;
    }
    turnRight(90);
    int rightDistance = getDistance();
    if (rightDistance > maxDistance) {
        maxDistance = rightDistance;
        clearDirection = 1;
    }
    turnLeft(45);
    if (maxDistance < 25) {
        moveBackward();
        delay(90);
        stopMotors();
        clearDirection = scanForClearPath();
    }
    return clearDirection;
}

void turnLeft(int degrees) {
    digitalWrite(in1, LOW);
    digitalWrite(in2, HIGH);
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
    Serial.println("Turn Left");
    delay(degrees * 5); // Adjust the multiplier to calibrate turning
    stopMotors();
}

void turnRight(int degrees) {
    digitalWrite(in1, HIGH);
    digitalWrite(in2, LOW);
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
    Serial.println("Turn Right");
    delay(degrees * 5); // Adjust the multiplier to calibrate turning
    stopMotors();
}

```