



BACHELORARBEIT

Frau
Yue Wang

**Entwicklung und Umsetzung
eines Systems zum Transport
von Sensoren in
schwierigen Umgebungen**

2025

BACHELORARBEIT

Entwicklung und Umsetzung eines Systems zum Transport von Sensoren in schwierigen Umgebungen

Autorin:
Frau Yue Wang

Studiengang:
Elektro- u. Informationstechnik

Seminargruppe:
EI21s1-BC

Erstprüfer:
Prof. Dr. -Ing. Michael Kuhl

Zweitprüfer:
M. Sc. Kevin Blümel

Einreichung:
Mittweida, 01.02.2025

BACHELOR THESIS

Development and implementation of a system for transporting sensor in difficult environments

author:

Ms. Yue Wang

course of studies:

Electrical and information Technology

seminar group:

EI21s1-BC

first examiner:

Prof. Dr. -Ing. Michael Kuhl

second examiner:

M. Sc. Kevin Blümel

submission:

Mittweida, 01.02.2025

Bibliografische Angaben

Wang, Yue:

Entwicklung und Umsetzung eines Systems zum Transport von Sensoren in schwierigen Umgebungen

Development and implementation of a system for transporting sensors in difficult environments

59 Seiten, Hochschule Mittweida, University of Applied Sciences,
Fakultät Ingenieurwissenschaft, Bachelorarbeit, 2025

Abstract

Dieser Artikel beschreibt ein intelligentes Fahrzeug, das in gefährlichen Umgebungen fahren kann. Das Fahrzeug kann über eine Fernbedienung gesteuert werden und verfügt über einen automatischen Fahrmodus, in dem es eigenständig Hindernisse erkennt und umgeht, wenn sich ein Hindernis in einem Abstand von 20 Zentimetern befindet.

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abkürzungsverzeichnis	IV
Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
1 Einleitung	1
2 Aufgabenstellung und These	3
3 Stand der Technik	5
3.1 Mikrocontroller	6
3.2 Bewegungsmechanismus	8
3.3 Motorantrieb	9
3.4 Fernbedienungsfunktion	10
3.4.1 Fernbedienungsmethoden	10
3.4.2 Infrarotfernbedienung	13
3.5 Entfernungsmessung	15
3.5.1 Messmethoden zur Entfernungsmessung	15
3.5.2 Ultraschall-Abstandssensor-Modul	17
4 Konkretisierung der Aufgabenstellung	20
4.1 Geplante konkrete Funktionen	20
4.2 Zielparameter	20
1. Infrarotfernbedienungsfunktion:	20
2. Automatischer Fahrmodus:	21
3. Automatische Hindernisvermeidung:	21
4. Weitere Parameter:	21
5 Umsetzung	22
5.1 Gesamtkonzept	22

5.1.1 Hardware-Komponenten	22
5.1.2 Software-Komponenten	24
5.2 Hardware	25
5.2.1 Anforderungen	25
5.2.2 Systementwurf	27
5.2.3 Hardware-Umsetzung	29
5.3 Software	30
5.3.1 Software-Konzept	30
5.3.2 Umsetzung des Softwarekonzepts	32
6 Systemtest	41
6.1 Test der Motorsteuerung	41
6.2 Test der Infrarot-Fernbedienungsfunktion	42
6.3 Ultraschall-Abstandsmessmodul	44
7 Versuchsdurchführung	46
7.1 These	46
7.2 Fernbedienungsfunktion	46
7.2.1 Fernbedienungsrichtung	46
7.2.2 Einfluss der Sonneneinstrahlung auf die Fernbedienung	47
7.3 Ultraschall-Abstandserkennung	50
7.3.1 Test der Messgenauigkeit des Ultraschallsensors	50
7.3.2 Test der Ausweichfähigkeit	51
7.4 Autonomes Fahren	53
Zusammenfassung	56
Ausblick	58
Literaturverzeichnis	I
Anlagen	III
PCB	III
Gelötete Leiterplatte	III
Physische Zeichnung des Autos	IV
Software	V

Eigenständigkeitserklärung	VIII
---	-------------

Abkürzungsverzeichnis

I/O	Input/Output
PWM	Pulse Width Modulation
SRAM	Static Random Access Memory
EEPROM	Electrically Erasable Programmable Read-Only Memory
IR	Infrared
RF	Radio Frequency
V _{in}	Input Voltage
GND	Ground
VCC	Voltage at the Common Collector
DO	Digital Output

Abbildungsverzeichnis

Grafik 1 Arduino Mega 2560	6
Grafik 2 L298N	9
Grafik 3 Strukturdiagramm des L298N	9
Grafik 4 HX1838	13
Grafik 5 SR-04	18
Grafik 6 Arbeitszeitdiagramm des HC-SR04 Ultraschallmoduls	19
Grafik 7 Gesamtblockdiagramm	23
Grafik 8 Einfaches Software-Flussdiagramm	24
Grafik 9 Schaltbild	27
Grafik 10 PCB	29
Grafik 11 Software-Flussdiagramm	31
Grafik 12 Codeabschnitt 1	32
Grafik 13 Codeabschnitt 2	32
Grafik 14 Codeabschnitt 3	32
Grafik 15 Codeabschnitt 4	33
Grafik 16 Codeabschnitt 5	33
Grafik 17 Codeabschnitt 6	34
Grafik 18 Codeabschnitt 7	35
Grafik 19 Codeabschnitt 8	36
Grafik 20 Codeabschnitt 9	37
Grafik 21 Codeabschnitt 10	37
Grafik 22 Codeabschnitt 11	38
Grafik 23 Codeabschnitt 12	38
Grafik 24 Codeabschnitt 13	38
Grafik 25 Codeabschnitt 14	39
Grafik 26 Motoransteuerung Testcode	41
Grafik 27 Prozess der Vorwärtsbewegung des Fahrzeugs	42

Grafik 28 Testcode für die Infrarot-Fernbedienungsfunktion	42
Grafik 29 Die im seriellen Monitor angezeigten Infrarotkodierungen	43
Grafik 30 Test der Ultraschall-Abstandsmessung	44
Grafik 31 Testcode für den Ultraschallsensor	44
Grafik 32 Installationsdiagramm des Ultraschall-Abstandsmessmoduls	45
Grafik 33 Außentest	48
Grafik 34 Innentest	49
Grafik 35 Prozess der Messgenauigkeitstests für Ultraschall	50
Grafik 36 Prozess des Tests der Ausweichfähigkeit	52
Grafik 37 Szeneinstellung	53

Tabellenverzeichnis

Tabelle 1 Vergleich der Eigenschaften von Bewegungssystemen	8
Tabelle 2 Steuerungsarten und Statusübersicht für Gleichstrommotoren	10
Tabelle 3 Test des Fernsteuerwinkels	47
Tabelle 4 Tabelle der Ultraschall-Distanzmessdaten	51
Tabelle 5 Tabelle zur Analyse der Ultraschall-Distanzmessfehler	51
Tabelle 6 Tabelle zur Überprüfung der Ausweichfähigkeit	52
Tabelle 7 Automatisierter Fahrttest	54

1 Einleitung

In der modernen Robotertechnik zeigen Fahrzeuge mit Fernsteuerung und automatischer Hindernisvermeidung in vielen Bereichen ihr einzigartiges Anwendungspotential, insbesondere bei der Navigation und Durchführung von Aufgaben in gefährlichen Umgebungen. Eine präzise Wahrnehmung der Umgebung, eine rationale Planung der Bewegungsbahn sowie die dynamische Entscheidungsfindung zur Ausführung der geplanten Bahn sind die drei Hauptbestandteile des Transportsystems. Dieses System benötigt das Fachwissen aus verschiedenen Disziplinen, wie z.B. Maschinenbau, Maschinenprinzipien, elektronische Steuerungstechnik, Schaltungsdesign, Sensortechnologie und Computerprogrammierung[1][2].

Durch die Integration verschiedener Technologien kann das Transportsystem entweder über Fernsteuerung entlang der geplanten Route fahren oder autonom auf Basis der durch Sensoren gewonnenen Wahrnehmungen navigieren, wodurch die direkte Manipulation durch Menschen vermieden wird [3].

Obwohl diese Fahrzeuge nicht über eine flexible autonome Entscheidungsfähigkeit verfügen, können sie durch die Kombination eines effektiven Fernsteuerungssystems mit fortschrittlichen Hindernissensoren sicher und zuverlässig in komplexen und potenziell riskanten Umgebungen operieren. Derzeit benötigen vielerlei Anwendungsszenarien wie Katastrophenhilfe, industrielle Inspektion und Umweltüberwachung dringend Fahrzeuge, die flexibel agieren und Hindernisse effektiv vermeiden können. Ziel dieser Arbeit ist es, ein Fahrzeug mit Fernsteuerung und automatischer Hindernisvermeidung zu entwerfen. Durch die Optimierung der Sensoranordnung und der Steuerungsalgorithmen soll die Stabilität und Sicherheit in gefährlichen Umgebungen erhöht werden. Anhand von experimentellen Untersuchungen wird diese Arbeit die Leistung des Fahrzeugs unter verschiedenen

Hindernisbedingungen analysieren und somit eine theoretische Grundlage und praktische Anleitung für zukünftige technologische Fortschritte bieten.

2 Aufgabenstellung und These

Ziel der Arbeit ist die prototypische Aufbau eines fernzusteuern Systems, welches in der Lage ist, Sensoren/ Sensorsysteme in schwierigen Umgebungen zu transportieren.

In dieser Arbeit wird der Fokus auf die Implementierung der Fortbewegungsmethoden des Fahrzeugs sowie die spezifischen Realisierungsmethoden für die Infrarot-Fernsteuerung und die Ultraschall-Hindernisvermeidung gelegt. Zunächst wird das Design der Fortbewegungsmethoden des Fahrzeugs detailliert untersucht, einschließlich der Auswahl des Fahrzeugchassis und des Antriebssystems. Durch die Optimierung des Hardware-Designs wird angestrebt, die Mobilitätsflexibilität und Stabilität des Fahrzeugs unter verschiedenen Terrain- und Umweltbedingungen zu erhöhen.

Des Weiteren wird die Implementierungsmethode der Infrarot-Fernsteuerung systematisch dargestellt. Die Mechanismen der Aussendung und des Empfangs von Infrarotsignalen werden analysiert, und eine Betriebslogik auf Basis eines Infrarot-Fernbedienungssystems wird vorgeschlagen, um die präzise Bewegung des Fahrzeugs unter manueller Kontrolle zu gewährleisten. Bezüglich der Ultraschall-Hindernisvermeidung werden die Anordnung der Ultraschallsensoren, die Signalverarbeitungsalgorithmen und die Entscheidungsmechanismen zur Hindernisvermeidung erläutert. Es wird untersucht, wie die Sensoren Hindernisse in Echtzeit detektieren, Entfernungen berechnen und das Fahrverhalten des Fahrzeugs durch das Kontrollsystem anpassen, um eine autonome Hindernisvermeidung zu ermöglichen.

Abschließend wird durch die Integration und Testung der beiden Funktionseinheiten die Gesamtleistung und Zuverlässigkeit des Fahrzeugs während der

Aufgabenausführung überprüft.

3 Stand der Technik

In diesem Abschnitt wird der technische Status der in der Aufgabe verwendeten Komponenten vorgestellt.

Intelligente Fahrzeuge, die in gefährlichen Umgebungen fahren, profitieren von den rasanten Entwicklungen in der Automobiltechnologie in den letzten Jahren. Verschiedene fortschrittliche Fahrerassistenzsysteme verbessern ständig die Wahrnehmungs-, Entscheidungs- und Kontrollfähigkeiten der Fahrzeuge unter schwierigen Bedingungen.

Ein intelligentes Fahrzeug ist ein Gerät, das in der Lage ist, automatisch Arbeiten in einer Umgebung gemäß vorgegebenen Mustern zu erledigen, ohne dass menschliche Verwaltung oder Bedienung erforderlich sind, um Zielarbeiten zu vollenden. Es besteht hauptsächlich aus drei Komponenten: Sensoren, einem Controller und Aktuatoren, die zusammenarbeiten, um die Funktionsweise des Fahrzeugs zu ermöglichen.

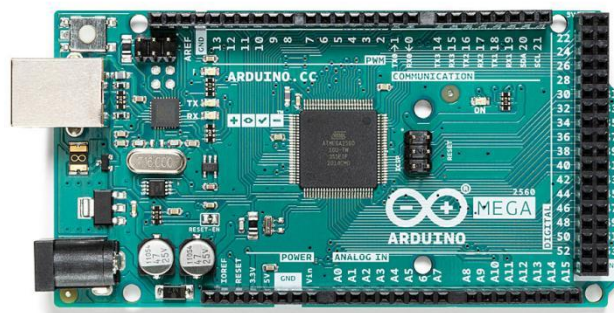
Die Sensoren fungieren als die "Sinnesorgane" des intelligenten Fahrzeugs und dienen dazu, externe Signale wahrzunehmen und zu empfangen. Die empfangenen Signale werden an den Controller zur Verarbeitung weitergeleitet.

Der Controller ist das "Gehirn" des intelligenten Fahrzeugs, das die von den Sensoren übermittelten Signale analysiert und Steuerbefehle an die Aktuatoren sendet.

Die Aktuatoren sind die "Gliedermaßen" des intelligenten Fahrzeugs und führen basierend auf den Signalen, die vom Controller empfangen werden, entsprechende Aktionen aus. Für das intelligente Fahrzeug sind die Räder die repräsentativsten Aktuatoren[4].

3.1 Mikrocontroller

Für dieses Design wurde das Arduino Mega 2560 als zentrale Entwicklungsplatine ausgewählt. Diese Entscheidung basiert auf den folgenden detaillierten Gründen und Vorteilen:



Grafik 1 Arduino Mega 2560

(<https://e.tb.cn/h.TN4uT3nxJsWVCZz?tk=5N4xe4D3C18>)

1. Umfangreiche I/O-Schnittstellen

- 54 digitale I/O-Pins: Ermöglichen den Anschluss einer Vielzahl von Sensoren und Aktoren und bieten genügend Erweiterungsmöglichkeiten für komplexe Projekte.
- 16 analoge Eingänge: In der Lage, multiple analoge Signale zu verarbeiten, ideal für die Erfassung von Sensordaten.
- 14 PWM-Ausgänge: Erleichtern die Steuerung von Motorgeschwindigkeit, LED-Helligkeit und anderen Szenarien, die analoge Ausgänge erfordern.

2. Leistungsstarke Verarbeitungskapazität

- ATmega2560 Mikrocontroller: 16 MHz Taktfrequenz, 16-Bit-Architektur, bietet

umfassende Rechenleistung [5].

- Kann mehrere Aufgaben gleichzeitig bearbeiten: Wie Infrarotsignalempfang, Ultraschallentfernungsmessung, Motorsteuerung usw.

3. Große Speicherkapazität

- 256 KB Flash-Speicher: Kann umfangreichere und komplexere Programmcodes speichern.
- 8 KB SRAM: Bietet ausreichend Laufzeitspeicher und unterstützt mehr Variablen und komplexere Datenstrukturen.
- 4 KB EEPROM: Kann zur Speicherung von Konfigurationsparametern verwendet werden, was die Projekt-Debugging und -Optimierung erleichtert [6].

3.2 Bewegungsmechanismus

Entsprechend der unterschiedlichen Bewegungsweisen und Strukturen traditioneller Transportsysteme können diese in drei Kategorien eingeteilt werden: Radtransportsysteme, Gurtförderersysteme und Kettenfahrzeugsysteme. Die Eigenschaften dieser drei Formen von Transportsystemen sind in Tabelle 1 dargestellt.

	Radtransportsysteme	Gurtförderersysteme	Kettenfahrzeugsysteme
Bewegungsgeschwindigkeit auf flachem Gelände	Relativ schnell	Relativ langsam	Mittel
Steuerungssystem	Einfach	Komplex	Mittel
Maschinenbau	Einfach	Komplex	Mittel
Stabilität auf komplexem Gelände	Relativ schlecht	Mittel	Relativ gut
Überwindungsfähigkeit	Relativ schlecht	Ausgezeichnet	Relativ gut
Typischer Roboter			

Tabelle 1 Vergleich der Eigenschaften von Bewegungssystemen

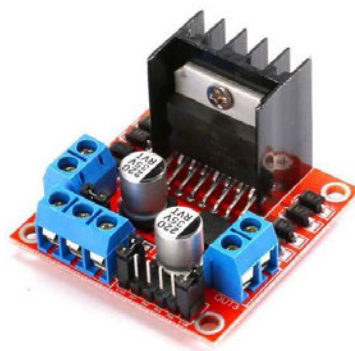
(Combinable Tracked Robot Obstacle overcoming Performance Research, Lingbin Shen, 2022)

Aus Tabelle 1 geht hervor, dass traditionelle Radtransportsysteme auf ebenem Untergrund eine hervorragende Bewegungs- und Geschwindigkeitsstabilität aufweisen. Zudem sind ihre Kontrollsysteme und mechanischen Strukturen nicht kompliziert. Aufgrund der Radgröße und -struktur ist die Hindernisüberwindungsfähigkeit im Vergleich zu den anderen beiden Typen jedoch relativ schwach. Traditionelle Gurtförderersysteme zeichnen sich durch eine hervorragende Hindernisüberwindungsfähigkeit und Stabilität auf unebenem Gelände aus, jedoch sind ihre mechanischen Strukturen und Steuerungssysteme komplexer und die Bewegungsgeschwindigkeit auf ebenem Gelände niedriger im Vergleich zu Radtransportsystemen. Kettenfahrzeugsysteme bieten sowohl hohe Bewegungsgeschwindigkeit als auch gute Hindernisüberwindungsfähigkeiten.

Deshalb haben Kettenfahrzeuge ein breites Anwendungsspektrum in den Bereichen Outdoor-Transport, Planetenerkundung, Notfallrettung und Kriegsfront.

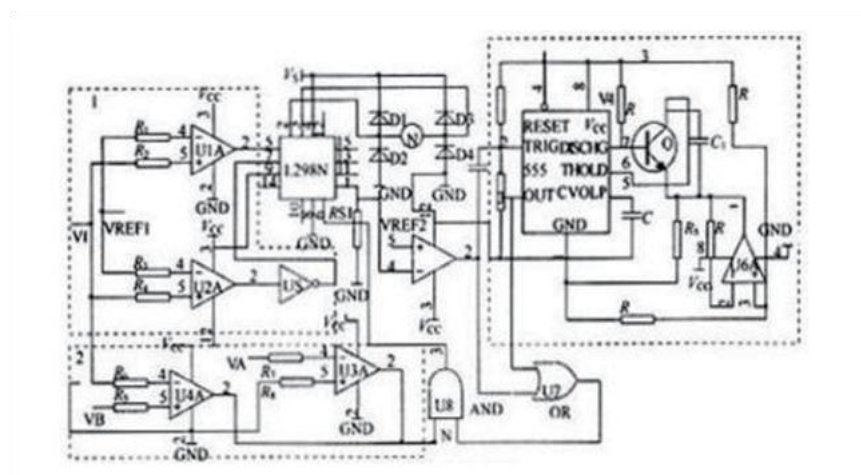
3.3 Motorantrieb

In diesem Design wird das L298N Motortreiber-Modul verwendet. Das L298N Dual-H-Brücken-Motortreiber-IC spielt eine wichtige Rolle im Motortreibersystem. L298N wird häufig in verschiedenen Motortreiberbereichen eingesetzt, einschließlich Robotik, 3D-Druckern, CNC-Graviermaschinen usw.



Grafik 2 L298N

(<https://e.tb.cn/h.TmwbTJN5bY5B3II?tk=wTrYe4OMRN6>)



Grafik 3 Strukturdiagramm des L298N

(Dong Shaowan, Guo Rui, 2018)

Funktionale Eigenschaften: Das L298N bietet zwei unabhängige H-Brücken-Motortreiber-Schaltungen und kann Gleichstrommotoren, Schrittmotoren oder andere Motortypen antreiben. Der Ausgangsstrom kann bis zu 2A betragen und der Arbeitsbereich der Spannung liegt zwischen 5V und 46V, wodurch es vielfältigen Anwendungsanforderungen gerecht wird.

Schaltungsdesign: Das L298N verwendet DMOS-Leistungsbaulemente, um eine H-Brücken-Struktur zu realisieren, und verfügt über integrierte Überstrom- und Überhitzungsschutzfunktionen, die Stabilität und Zuverlässigkeit der Treiberschaltung gewährleisten. Durch Eingabesteuersignale können Drehrichtungs- und Bremsvorgänge des Motors realisiert werden.[7]

	IN1	IN2	IN3	IN4
vorwärts	1	0	1	0
rückwärts	0	1	0	1
halten	0	0	0	0
links abbiegen	0	0	1	0
rechts abbiegen	1	0	0	0

Tabelle 2 Steuerungsarten und Statusübersicht für Gleichstrommotoren
(Bedienungsanleitung von L298N)

3.4 Fernbedienungsfunktion

3.4.1 Fernbedienungsmethoden

Als ein Fahrzeug, das in gefährlichen Umgebungen fahren kann, ist die Fernbedienungsfunktion über große Entfernungen unerlässlich.

Die Fernbedientechnologie spielt eine wichtige Rolle in der modernen Technik und im Alltag, insbesondere im Bereich der Automatisierung und intelligenten Steuerung. Mit der Entwicklung der Technologie sind verschiedene Fernbedienungsmethoden entstanden, die jeweils unterschiedliche Merkmale und Anwendungsbereiche

aufweisen.

1. Infrarot-Fernbedienung (IR)

Infrarotlicht ist eine Art von niedrigfrequentem, rotem Licht; die Infrarot-Fernbedienungstechnik ist eine Methode, die Infrarotstrahlen zur Punkt-zu-Punkt-Kommunikation nutzt.

Die Infrarot-Fernbedienung gehört zu einer Art von drahtloser, kontaktloser Steuerung. Ihre Vorteile sind signifikant, wie z. B. geringe Kosten, niedriger Energieverbrauch, einfache Kodierung und Dekodierung, zuverlässige Informationsübertragung und geringe Beeinflussung der Umgebung usw. [8]

Die Infrarot-Fernbedientechnologie ist als eine einfache und kostengünstige drahtlose Fernbedienungsmethode weit verbreitet in den Bereichen Haushaltsgeräte, Spielzeuge und intelligente Häuser.

2. Funkfrequenz-Fernbedienung (RF)

Die Funkfrequenz-Fernbedienung ist eine Methode, die Radiowellen zur Kommunikation zwischen Geräten nutzt. Ihre Eigenschaften umfassen die Fähigkeit, Wände und Hindernisse zu durchdringen, und die Steuerreichweite ist normalerweise recht groß, was sie für verschiedene Fernbedienungsanwendungen, wie z.B. Garagentoröffner und Drohnensteuerung, geeignet macht. Jedoch ist die RF-Fernbedienung anfällig für Frequenzinterferenzen und erfordert zusätzliche technische Unterstützung.

3. Bluetooth-Fernbedienung

Bluetooth ist eine sich entwickelnde drahtlose Verbindungstechnologie für kurze Distanzen mit geringem Energieverbrauch, die den Nutzern ein kostengünstiges, hochzuverlässiges und wenig komplexes drahtloses Kommunikationsnetzwerk bietet. Es ermöglicht, jederzeit und überall kabelgebundene Verbindungen durch drahtlose Schnittstellen zu ersetzen, wodurch unterschiedliche digitale Geräte im Nahbereich

bequem nahtlos Datenressourcen teilen können.[9]

Allerdings ist die effektive Steuerreichweite von Bluetooth normalerweise auf 10 Meter begrenzt, und es gibt gewisse Einschränkungen bei der Steuerung mehrerer Geräte.

4. Wi-Fi-Fernbedienung

Die Wi-Fi-Fernbedienung nutzt ein drahtloses Netzwerk zur Fernsteuerung von Geräten. Die Verbindung zwischen den Netzwerken erfolgt über Router. Der Einsatz von Routern ist äußerst flexibel und kann zwischen verschiedenen Standorten angewendet werden. Router bieten hohe Sicherheit, minimieren die Bandbreite eines lokalen Netzwerks, ermöglichen den Informationsaustausch und die Auswahl des besten Übertragungsweges. Diese Eigenschaften machen sie geeignet für großflächige Netzwerke, da sie die Belastung der Hauptgeräte reduzieren, den Kommunikationsfluss optimieren und eine schnellere Verarbeitung von Multimediadaten ermöglichen. [10]

Die Hauptvorteile bestehen in der breiten Abdeckung und der Unterstützung der Verbindung und Verwaltung mehrerer Geräte. Die Wi-Fi-Fernbedienung kann Rückmeldungen und Datenübertragungen bereitstellen, was sie besonders geeignet für intelligente Hausanwendungen macht, die eine große Menge an Daten sammeln müssen. Allerdings benötigt diese Methode eine stabile Netzwerkverbindung und hat einen höheren Energieverbrauch.

Nach einer umfassenden Analyse verschiedener Fernbedienungstechnologien gibt es folgende Gründe, warum die Auswahl der Infrarot-Fernbedienung als Steuerungssystem für das intelligente Fahrzeug sinnvoll ist:

1. Technologische Reife: Die Technologie der Infrarot-Fernbedienung hat eine lange Geschichte, und es gibt reichlich Entwicklungsressourcen, Dokumentationen und Module, die eine schnelle Umsetzung und Prototypenentwicklung erleichtern.

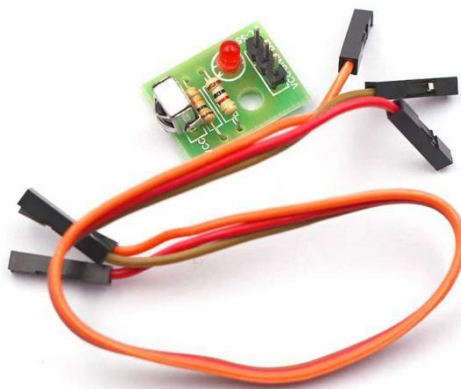
2. Einfachheit und Benutzerfreundlichkeit: Für die grundlegenden Funktionen wie Vorwärts-, Rückwärts- und Steuerung ist die Infrarot-Fernbedienung ausreichend und der Umsetzungsprozess relativ einfach.

3. Stabilität: In nicht komplexen Umgebungen bietet die Infrarot-Fernbedienung eine hohe Stabilität und vermeidet Probleme, die durch Funkfrequenzinterferenzen entstehen.

Zusammenfassend ist die Infrarot-Fernbedienung als effektive Steuerungsmethode für das Design und die Erprobung von intelligenten Fahrzeugen geeignet. Sie vereinfacht nicht nur den Entwicklungsprozess, sondern senkt auch die Gesamtkosten der Projektumsetzung.

3.4.2 Infrarotfernbedienung

In diesem Design wird das HX1838 Infrarotfernbedienungsmodul verwendet.



Grafik 4HX1838

(<https://e.tb.cn/h.TmDlITjauZNIe6?tk=EfvQe4Os9WN MF937>)

HX1838 ist ein gängiger Infrarotempfängerchip, der in den Empfangsschaltkreisen verschiedener elektronischer Produkte weit verbreitet ist. Es hat folgende Eigenschaften:

1. Arbeitsspannungsbereich: 2,4V-5,5V, kann direkt mit Mikrocontrollern oder Mikroprozessoren integriert werden.
2. Hohe Empfangsempfindlichkeit: Kann Infrarotfernbedienungssignale in einem Bereich von 3–6 Metern empfangen.
3. Standard NEK-Protokolldecodierung: Unterstützt das weit verbreitete NEC-Infrarotfernbedienungsprotokoll.
4. Kompakte Größe : SOP8-Gehäuse, benötigt wenig Platz auf der Leiterplatte.
5. Hohe Integrationsstufe: Integrierte Filter- und Decoderschaltungen, die das Design der externen Schaltungen vereinfachen[11].

Zusammenfassend lässt sich sagen, dass das HX1838 Infrarotfernbedienungsmodul aufgrund seiner stabilen Leistung, hohen Integration, einfachen Schnittstellen und niedrigen Energieverbrauchs sehr gut für die Fernsteuerung von kleinen intelligenten Fahrzeugen geeignet ist, wodurch die Leistung und Zuverlässigkeit des Fahrzeugs effektiv gesteigert werden kann.

3.5 Entfernungsmessung

3.5.1 Messmethoden zur Entfernungsmessung

Die Entfernungsmessungsfunktion bezieht sich auf das Messen der Distanz zwischen Objekten durch verschiedene technische Mittel oder Geräte. Diese Funktion ist in vielen Bereichen von großer Bedeutung, insbesondere in Anwendungen wie Robotik, autonomem Fahren, intelligentem Wohnen und industrieller Automatisierung.

Um die Hindernisvermeidung für das Fahrzeug zu realisieren, ist die Entfernungsmessungsfunktion unerlässlich.

Im Folgenden werden einige gängige Methoden zur Entfernungsmessung vorgestellt.

Gängige Methoden zur Entfernungsmessung:

1. Laserentfernungsmessung:

Durch das Aussenden eines Laserstrahls und die Messung der Rücklaufzeit wird der Abstand zwischen dem Objekt und dem Entfernungsmesser bestimmt. Diese Methode bietet hohe Messgenauigkeit und große Reichweite und kann für Langstreckenmessungen verwendet werden. Bei transparenten oder stark reflektierenden Objekten kann es jedoch zu Fehlern kommen, und die Kosten sind relativ hoch.

2. Infrarotentfernungsmessung:

Infrarotsensoren sind Geräte, die Infrarotstrahlen zur Sensormessung nutzen. Diese Sensoren verwenden Infrarotstrahlen als Medium zur Durchführung von Messungen. Infrarotstrahlen haben Eigenschaften wie Reflexion, Brechung, Streuung, Interferenz und Absorption. Bei der Anwendung der Infrarotsensoren erfolgt die Messung, ohne direkten Kontakt mit dem zu messenden Objekt, wodurch Reibung

ausgeschlossen ist. Sie bieten hohe Sensitivität und schnelle Reaktionszeiten. Allerdings sind sie stark lichtempfindlich, die Messdistanz ist begrenzt und sie sind daher für Nahbereichsmessungen geeignet.[12]

3. Ultraschallentfernungsmessung

Die Ultraschallentfernungsmessung basiert auf dem Prinzip, Ultraschallwellen auszusenden und die Zeit zu messen, die benötigt wird, um die zurückkehrenden Wellen zu empfangen, wodurch die Entfernung zu einem Objekt berechnet wird. Diese Methode ist unempfindlich gegenüber Lichtverhältnissen und kann in verschiedenen Umgebungen stabil arbeiten, was sie geeignet macht, um Entfernungen von Nahbereich bis Mitteldistanz zu messen. Allerdings reagiert sie weniger effektiv bei schallabsorbierenden Materialien oder sehr großen Entfernungen.

Analyse, warum im Design dieses Fahrzeugs Ultraschallentfernungsmessung verwendet wird:

- Umgebungsanpassungsfähigkeit: Die Ultraschallentfernungsmessung wird nicht von Lichtverhältnissen beeinflusst, was sie für den Einsatz in verschiedenen Umgebungen, sowohl innen als auch außen, geeignet macht. Sie funktioniert stabil bei Tag und Nacht.
- Geeigneter Messbereich: Der für die Ultraschallentfernungsmessung geeignete Messbereich erfüllt in der Regel die Anforderungen der Navigation und Hindernisvermeidung des Fahrzeugs, insbesondere im Bereich von einigen Zentimetern bis mehreren Metern.
- Einfache Integration: Ultraschallsensoren sind klein und leicht, was ihre Integration in das Design des Fahrzeugs erleichtert, ohne das Gesamtgewicht und die Komplexität erheblich zu erhöhen.

Insgesamt ist die Ultraschallentfernungsmessung sehr gut für die Anwendung in mobilen Robotern oder autonomer Fahrzeugtechnologie geeignet.

3.5.2 Ultraschall-Abstandssensor-Modul

Das Ultraschall-Abstandsmessmodul ist ein elektronisches Gerät zur Messung von Distanzen. Es besteht hauptsächlich aus einem Sender, einem Empfänger, einer Steuerungsschaltung und einer Schnittstelle. Das Prinzip der Abstandsmessung mit einem Ultraschallsensor beruht auf der Geschwindigkeit, mit der sich Ultraschallwellen durch die Luft bewegen. Um die Entfernung zu einem Hindernis zu bestimmen, wird das Zeitintervall zwischen dem Senden des Schallsignals und dem Empfangen des Echo-Signals berechnet.[13]

Das Messmodul sendet einen kurzen Hochfrequenzschallimpuls (in der Regel etwa 40 kHz) aus. Ultraschall breitet sich mit einer Geschwindigkeit von etwa 340 m/s in der Luft aus. Wenn der Schall auf ein Hindernis trifft, wird er reflektiert. Der Empfänger des Messmoduls erkennt die zurückgekehrten Schallwellen. Anschließend wird unter Verwendung der Laufzeit der Schallwellen und der Schallgeschwindigkeit die Entfernung berechnet.

Die Entfernung wird durch die folgende Gleichung (1) dargestellt:

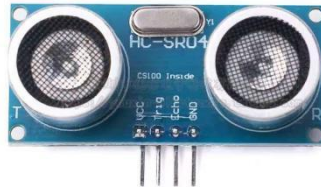
$$S = \frac{1}{2T} \times 340m/s \quad (1)$$

Formel 1

Dabei ist S die Entfernung des Sensors zum Hindernis, T die Dauer des

hochgestellten Echo-Signals und V die Geschwindigkeit des Schalls (ca. 340 m/s).

Dieses Design verwendet den SR-04 Ultraschall-Abstandssensor.



Grafik 5SR-04

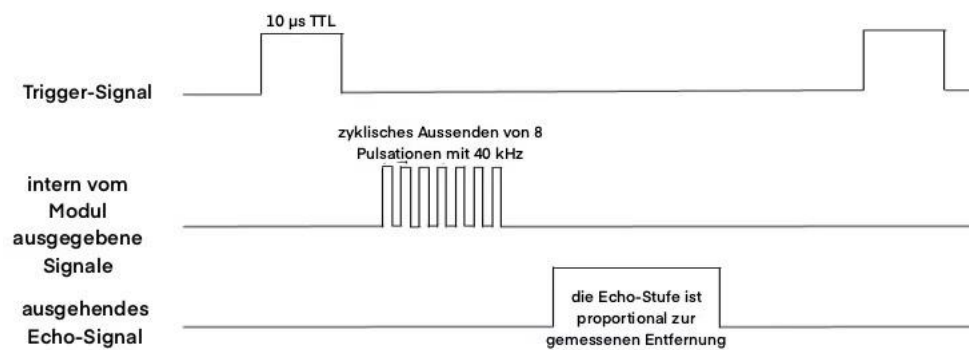
(<https://e.tb.cn/h.TmxeT9wZiHljQck?tk=Uogoe4mP1m8>)

Der SR-04 hat einen Messbereich von 2 cm bis 400 cm und ist daher besonders geeignet für den Einsatz von Fahrzeugen in Innenräumen oder engen Räumen. Das SR-04 Modul ist kompakt und leicht, was bedeutet, dass es das Volumen und Gewicht des Fahrzeugs nicht erheblich erhöht.

Der Arbeitsstrom des SR-04 liegt unter 15 mA, während der Standby-Strom unter 2 mA liegt, was es ideal für batteriebetriebene Fahrzeuge macht. Im normalen Einsatzbereich beträgt der Messfehler des Moduls in der Regel weniger als 1 cm, was den Anforderungen an die Hindernisvermeidung für das Fahrzeug gerecht wird. Darüber hinaus kann das Modul alle 40 ms eine Messung durchführen, was für die Echtzeit-Hindernisvermeidung des Fahrzeugs erforderlich ist.

Das HC-SR04 Ultraschall-Abstandsmessmodul bietet eine berührungslose Messung von 2 cm bis 400 cm. Der Arbeitsablauf des HC-SR04 ist in Abbildung 1 dargestellt. Um das Modul in den Arbeitsmodus zu aktivieren, ist ein Trigger-Signal von mehr als 10 μ s erforderlich. Nach der Aktivierung sendet das Modul 8 Pulse mit einer Frequenz von 40 kHz aus und erkennt den Empfang des Echo-Signals. Bei Erkennung eines

Echo-Signals wird ein Rücksignal ausgegeben, wobei die Pulsbreite des Rücksignals linear mit der gemessenen Distanz korreliert. Um Störungen des Rücksignals durch das ausgesendete Signal zu vermeiden, sollte der Trigger-Zyklus 60 ms oder mehr betragen.[14]



Grafik 6 Arbeitszeitdiagramm des HC-SR04 Ultraschallmoduls

4 Konkretisierung der Aufgabenstellung

Die Aufgabe ist Entwicklung und Umsetzung eines Systems zum Transport von Sensoren in schwierigen Umgebungen.

4.1 Geplante konkrete Funktionen

Das System kann entsprechend den Befehlen des Infrarotfernbedienungsgeräts Bewegungen in verschiedene Richtungen (vorwärts, rückwärts, links, rechts) durchführen. Nach Drücken einer bestimmten Taste kann es in den automatischen Fahrmodus wechseln.

Im automatischen Fahrmodus ist das Fahrzeug in der Lage, mithilfe des Ultraschall-Abstandsmessmoduls Hindernisse zu erkennen. Wenn ein Hindernis in einem Abstand von 20 cm vor dem Fahrzeug erkannt wird, wird eine Wendefunktion aktiviert, um dem Hindernis auszuweichen.

4.2 Zielparameter

1. Infrarotfernbedienungsfunktion:

- Das Fahrzeug kann durch die Infrarotfernbedienung grundlegende Operationen wie Vorwärts-, Rückwärts-, Links- und Rechtsabbiegen durchführen.
- Die Fernbedienung sollte über eine klare Tastenanordnung verfügen, um die Bedienung zu erleichtern.
- Der Empfang und die Dekodierung des Fernbedienungssignals sollten präzise und fehlerfrei sein.
- Die Reichweite der Fernbedienung sollte mindestens einen Meter betragen.
- Die Reaktionszeit der Fernbedienung sollte weniger als 200 Millisekunden

betragen.

2. Automatischer Fahrmodus:

- Nach dem Drücken einer bestimmten Taste auf der Fernbedienung kann das Fahrzeug in den automatischen Fahrmodus wechseln.

3. Automatische Hindernisvermeidung:

- Ein Ultraschallsensor an der Vorderseite wird zur Hinderniserkennung eingesetzt.

- Die Erkennungsdistanz für Hindernisse beträgt mindestens 20 cm.

- Wenn ein Hindernis direkt vor dem Fahrzeug erkannt wird, passt das Fahrzeug automatisch seinen Fahrweg an, um eine Kollision zu vermeiden.

4. Weitere Parameter:

- Fahrzeuggröße: Länge 20 cm x Breite 20 cm x Höhe 9 cm

5 Umsetzung

5.1 Gesamtkonzept

In diesem Kapitel wird der Implementierungsprozess des multifunktionalen Fernbedienungshindernisvermeidenden Fahrzeugs im Detail beschrieben. Basierend auf der vorhergehenden Bedarfsanalyse und Systemdesign wurden schrittweise die verschiedenen Funktionen des Fahrzeugs umgesetzt.

Der Implementierungsprozess gliedert sich hauptsächlich in drei Hauptphasen: Hardware-Umsetzung, Software-Entwicklung und Systemintegration. Jede Phase wurde sorgfältig geplant und streng ausgeführt, um sicherzustellen, dass das Endprodukt die festgelegten Funktions- und Leistungsanforderungen erfüllt.

5.1.1 Hardware-Komponenten

1. Eingabemodule:

- Das Infrarotfernbedienungsmodul und das Ultraschall-Abstandsmessmodul fungieren als Eingabemodule.
- Das Infrarotfernbedienungsmodul empfängt die Infrarotsignale.
- Das Ultraschall-Abstandsmessmodul misst die Entfernung zu Objekten vor dem Fahrzeug.

2. Zentrale Verarbeitungseinheit:

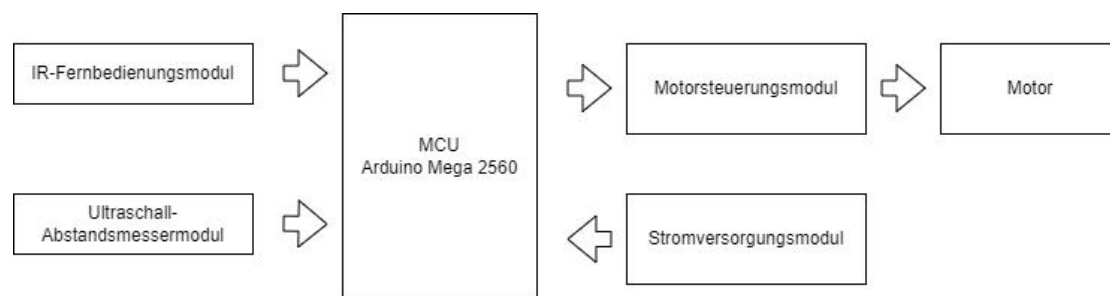
- Der Arduino Mega 2560 dient als zentrale Verarbeitungseinheit.
- Die zentrale Verarbeitungseinheit steuert die Ausgabemodule basierend auf den empfangenen Daten.
- Wenn das Signal zum Wechsel in den automatischen Fahrmodus empfangen

wird, aktiviert das System diesen Modus und führt Hindernisvermeidung basierend auf der Entfernung zu den vorderen Objekten durch.

- Die zentrale Verarbeitungseinheit wird durch eine 9V Lithium-Batterie mit Strom versorgt.

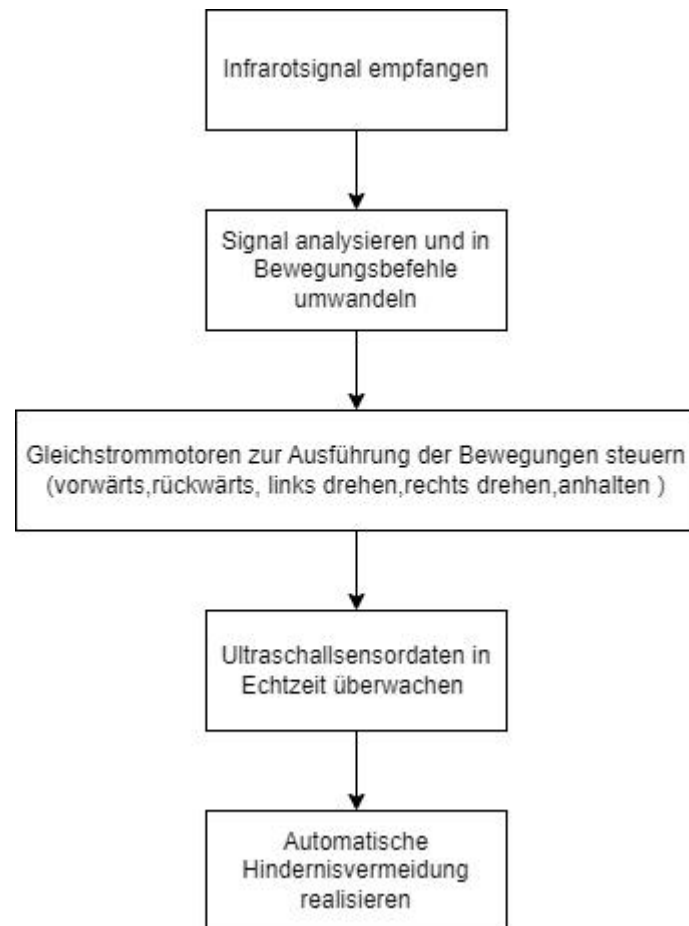
3. Ausgabemodule:

- Das Motorsteuerungsmodul fungiert als Ausgabemodul.
- Das Motorsteuerungsmodul treibt die Motoren an, um das Fahrzeug in Bewegung zu setzen.
- Das Motorsteuerungsmodul wird mit einer 12V-Stromversorgung als Antriebsspannung versorgt.



Grafik 7 Gesamtblockdiagramm

5.1.2 Software-Komponenten



Grafik 8 Einfaches Software-Flussdiagramm

Das Infrarotempfängsmodul empfängt die Signale und decodiert sie. Basierend auf den Dekodierungsergebnissen steuert es die Motoren, um entsprechende Bewegungen auszuführen.

Wenn der automatische Modus aktiviert ist, erkennt das System Hindernisse mithilfe des Ultraschallsensors und passt die Richtung an, um Kollisionen zu vermeiden.

5.2 Hardware

5.2.1 Anforderungen

Anforderungen an die Hardwarekomponenten für den ordnungsgemäßen Betrieb des Fahrzeugs:

1. Eingabemodule:

Das Eingabemodul besteht aus dem Infrarotfernbedienungsmodul und dem Ultraschall-Abstandsmessmodul.

- Das Infrarotfernbedienungsmodul dient zum Empfang der Infrarotsignale vom Fernbedienungsgerät. Um sicherzustellen, dass das Fahrzeug Infrarotsignale über einen größeren Bereich empfangen kann, werden in diesem System drei Infrarotempfänger installiert, die gleichmäßig im oberen Bereich des Fahrzeugs triangulär verteilt sind. Laut der Anleitung hat der Infrarotempfänger einen Empfangsbereich von 30 bis 60 Grad, aber in der Praxis kann der Empfangsbereich des Fernbedienungsgeräts größer sein. Die drei Infrarotempfänger decken im Wesentlichen den erforderlichen Fernsteuerungsbereich ab.

- Das Ultraschall-Abstandsmessmodul wird verwendet, um im automatischen Hindernisvermeidungsmodus die Entfernung zu Objekten vor dem Fahrzeug zu messen. Es ist an der Vorderseite des Fahrzeugs montiert. Die Systemanforderung besagt, dass das Fahrzeug bei der Erkennung eines Hindernisses in einer Entfernung von 20 cm vor sich mit der Hindernisvermeidung beginnen soll. Daher sollte der Erfassungsbereich des Moduls größer als 20 cm sein.

2. Zentrale Verarbeitungseinheit:

Für dieses Design wird der Arduino Mega 2560 als zentrale Verarbeitungseinheit verwendet, um die Daten der Eingabemodule zu verarbeiten und die Steuerung des Motorsteuerungsmoduls durchzuführen. Der Mikrocontroller wird mit einer 9V

Lithiumbatterie mit Strom versorgt.

3. Ausgabemodule:

Das Ausgabemodul besteht aus dem L298N Motorsteuerungsmodul und den Motoren.

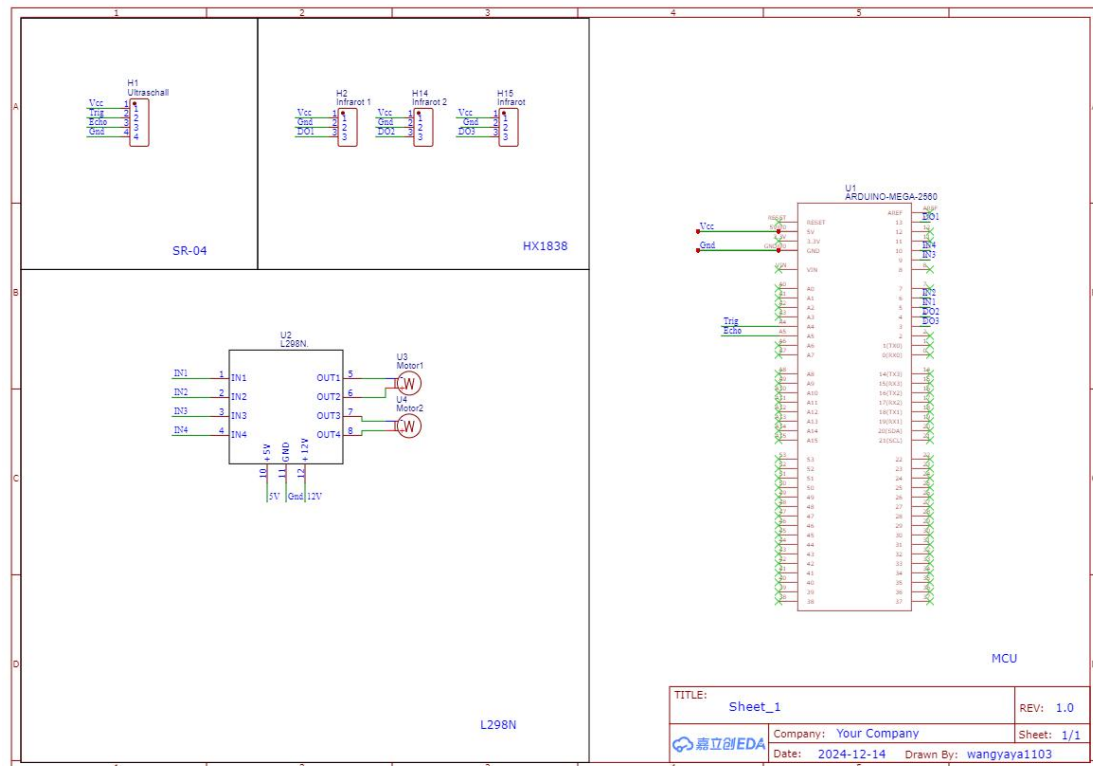
- Die vier Eingangs-Pins des L298N Motorsteuerungsmoduls sind mit der zentralen Verarbeitungseinheit verbunden, um Signale zu empfangen. Die vier Ausgangspins sind mit den Motoren verbunden, die an den Seiten des Fahrzeugs fixiert sind. Da der L298N einen höheren Strom benötigt, um die Motoren zu betreiben, ist die interne Stromversorgung, die normalerweise von demselben Mikrocontroller bereitgestellt wird, oft nicht ausreichend. Daher ist eine externe 12V-Stromversorgung erforderlich, um den Strombedarf der Motoren zu decken.
- An den beiden Seiten des Fahrzeugs werden zwei Gleichstrommotoren installiert, um die Räder auf beiden Seiten zu bewegen.

4. Fahrzeuggehäuse und Räder:

Gemäß den Anforderungen der Aufgabe wurde entschieden, Gummibandräder zu verwenden, um sicherzustellen, dass das Fahrzeug auch in gefährlichen Situationen stabil fahren kann.

5.2.2 Systementwurf

Unten ist das Schaltbild des Schaltkreises.



Grafik 9Schaltbild

1. Arduino Mega 2560 MCU

Der Arduino Mega 2560 wird als zentrale Verarbeitungseinheit verwendet und stellt das Herzstück des Schaltplans dar, da er verschiedene Eingangs- und Ausgangsanforderungen verarbeiten kann.

- Pinbelegung:

- Insgesamt 54 digitale Pins (0-53)
- Insgesamt 16 analoge Eingabepins (A0-A15)
- Stromversorgungs-Pins: Vin, 5V, 3.3V, GND

2. SR-04 Ultraschall-Abstandsmessmodul

Das SR-04 Ultraschall-Abstandsmessmodul verfügt über insgesamt 4 Pins:

- Vcc: Versorgungs-Pin, der mit dem 5V Versorgungs-Pin des Mikrocontrollers

verbunden wird.

- Trig: Trigger-Pin, der zur Aktivierung der Abstandsmessung verwendet wird und Ultraschallwellen aussendet. Verbunden mit Pin A4.
- Echo: Echo-Pin, der die Zeit des zurückgegebenen Ultraschallsignals repräsentiert. Verbunden mit Pin A5.
- GND: Erdungspin, der mit dem Erdungspunkt der Schaltung verbunden wird.

3. HX-1838 Infrarotempfänger

Das HX-1838 Modul hat drei Hauptpins. In diesem Design werden drei HX-1838 Module angeschlossen, um das Empfangen von Infrarotsignalen über einen größeren Winkel zu gewährleisten.

- Vcc: Versorgungs-Pin, der mit der 5V Stromversorgung verbunden wird.
- GND: Erdungspin, der mit der Erdung verbunden wird.
- DO: Ausgangspin, der ein High-Signal ausgibt, wenn ein Infrarotsignal empfangen wird. Die DO-Pins der drei HX-1838 Module sind mit den Pins 13, 3 und 4 verbunden.

4. L298N Motorsteuerungsmodul

Der L298N ist ein weit verbreitetes dual-polarem Gleichstrommotorsteuerungsmodul, das zwei Gleichstrommotoren gleichzeitig betreiben kann.

- IN1, IN2: Steuern die Drehrichtung von Motor 1. Verbunden mit den Pins 5 und 6.
- IN3, IN4: Steuern die Drehrichtung von Motor 2. Verbunden mit den Pins 9 und 10.
- OUT1, OUT2: Verbunden mit den Ausgängen des Motor 1.
- OUT3, OUT4: Verbunden mit den Ausgängen des Motor 2.
- +5V: Eingangsspannung, verbunden mit dem 5V Pin des Mikrocontrollers.
- GND: Erdungspin, der mit der Erdung verbunden wird.
- +12V: Betriebsstrom, eine externe 12V-Stromversorgung wird benötigt, um sicherzustellen, dass das Modul ausreichend Strom für die Motoren bereitstellt.

5. TT-Motor

Der TT-Motor ist ein gängiger Gleichstrommotor. Im Schaltkreis sind zwei Motoren

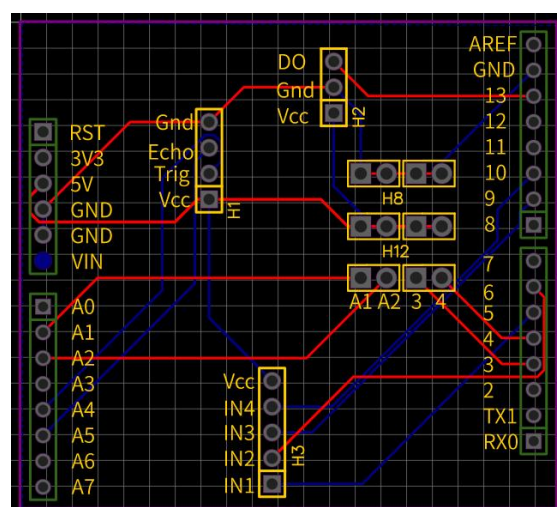
installiert, die an den Seiten des Fahrzeugs angebracht sind, um die Räder auf beiden Seiten anzutreiben.

- Die positiven und negativen Anschlüsse des TT-Motors sind mit den Ausgängen (OUT1, OUT2, OUT3, OUT4) des L298N Motorsteuerungsmoduls verbunden. Durch Ändern des Eingangssignals (High oder Low) kann die Drehrichtung des Motors gesteuert werden.

5.2.3 Hardware-Umsetzung

Um eine sinnvolle Anordnung der internen Hardwarekomponenten des Fahrzeugs zu erreichen, ist das Design einer PCB notwendig.

Da die PCB am Mikrocontroller befestigt werden muss, ist es erforderlich, die entsprechenden Steckverbinder gemäß der Pinbelegung des Mikrocontrollers anzuordnen. Das Ultraschall-Abstandsmessmodul SR-04 muss an der Vorderseite des Fahrzeugs befestigt werden, während der Infrarotempfänger HX-1838 auf dem Dach angebracht werden muss. Da der L298N Motorsteuerungsmodul relativ groß ist und nicht direkt auf der PCB befestigt werden kann, werden auf der PCB Anschlussstellen eingerichtet, um eine Verbindung herzustellen.



Grafik 10PCB

5.3 Software

5.3.1 Software-Konzept

5.3.1.1 Rahmenbedingungen

Dieses Projekt basiert auf der Entwicklungsplattform Arduino und ist daher auf die Hardware- und Software-Frameworks von Arduino angewiesen. Die Arduino IDE stellt die erforderlichen Werkzeuge zum Kompilieren und Hochladen des Codes zur Verfügung. Sie ermöglicht es uns, über die Verbindung zu einem Computer verschiedene Bibliotheken zu verwenden, um Programme zu brennen und zu debuggen.

5.3.1.2 Verwendete Softwarekomponenten

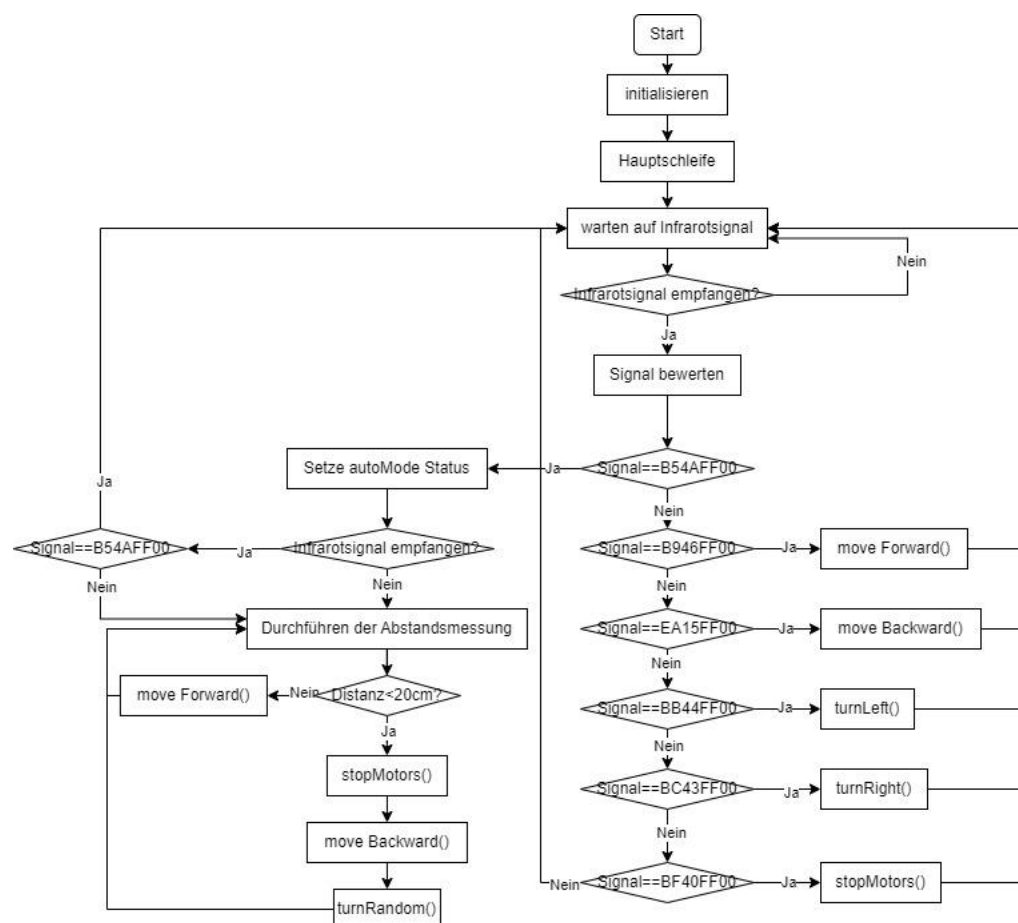
1. IRremote-Bibliothek: Die IRremote-Bibliothek wird verwendet, um Infrarotsignale zu empfangen und zu decodieren. Es handelt sich um eine Open-Source-Bibliothek, die Verarbeitung von Signalen verschiedener Infrarot-Fernbedienungen erleichtert.

2. NewPing-Bibliothek: Die NewPing-Bibliothek dient der Verwendung von Ultraschallsensoren zur Abstandsmessung. Diese Bibliothek vereinfacht die Interaktion mit der Hardware und ermöglicht eine effiziente Erfassung von Entfernungsdaten, die in der Hindernisvermeidung verwendet werden können.

5.3.1.3 Eigenentwickelte Software

Diese Software steuert die Bewegung des Fahrzeugs über Arduino und ermöglicht die Funktionen des Empfangs von Infrarotsignalen sowie der Ultraschall-Hindernisvermeidung.

Der Infrarotempfänger überträgt das empfangene Signal an den Mikrocontroller, der daraufhin das Signal analysiert und die entsprechenden Funktionen aufruft. Wenn ein Signal zum Umschalten in den automatischen Hindernisvermeidungsmodus erkannt wird, beginnt das Fahrzeug, automatisch geradeaus zu fahren und Hindernisse zu umgehen. Bei erneutem Empfang des Signals für den automatischen Hindernisvermeidungsmodus wird der Modus beendet.



Grafik 11 Software-Flussdiagramm

5.3.2 Umsetzung des Softwarekonzepts

5.3.2.1 Einbinden der Bibliotheksdateien und Definieren der Anschlüsse

```
1  #include <IRremote.hpp>
2  #include <NewPing.h>
```

Grafik 12 Codeabschnitt 1

- IRremote: Diese Bibliothek wird verwendet, um Infrarotfernbedienungssignale zu empfangen und zu decodieren.
- NewPing: Sie bietet Funktionen für den Ultraschallsensor (HC-SR04) und macht den Abstandsmessprozess effizienter.

```
5  const int IN1 = 5;
6  const int IN2 = 6;
7  const int IN3 = 9;
8  const int IN4 = 10;
9  const int IR_RECEIVER_PIN_1 = 13; // Erster Infrarotempfänger
10 const int IR_RECEIVER_PIN_2 = 3;  // Zweiter Infrarotempfänger
11 const int IR_RECEIVER_PIN_3 = 4;  // Dritter Infrarotempfänger
12 const int TRIG_PIN = A4;
13 const int ECHO_PIN = A5;
```

Grafik 13 Codeabschnitt 2

Die Steuerepins für die Motoren des Fahrzeugs, die Pins der drei Infrarotempfänger und die Pins des Ultraschallsensors wurden definiert.

5.3.2.2 Einstellungen des Ultraschallsensors

```
15 // Parameter des Ultraschallsensors
16 #define MAX_DISTANCE 200 // Maximale Messdistanz des Ultraschallsensors festlegen
17 NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);
```

Grafik 14 Codeabschnitt 3

- MAX_DISTANCE: Definiert die maximale Entfernung, die der Sensor messen kann, hier auf 200 cm eingestellt. Dieser Wert hängt von den Eigenschaften des Sensors ab, um ungenaue Messungen bei zu großer Distanz zu vermeiden.
- sonar: Ein Objekt namens sonar wird über die NewPing-Bibliothek erstellt, das für die Verwaltung der Funktionen des Ultraschallsensors verantwortlich ist. Die Parameter des Konstruktors umfassen den Trigger-Pin, den Echo-Pin und die maximale Messdistanz.

5.3.2.3 Definition des Infrarotsignals

Basierend auf der Funktionsplanung des Fahrzeugs werden insgesamt sechs Steuerknöpfe benötigt. Daher wurden sechs Infrarotcodes ausgewählt und die entsprechenden Signalwerte definiert, um die Steuerung beim Empfang zu ermöglichen.

```
19 // Definition der Infrarot-Fernbedienungssignale
20 const unsigned long FORWARD_SIGNAL = 0xB946FF00;
21 const unsigned long BACKWARD_SIGNAL = 0xEA15FF00;
22 const unsigned long LEFT_SIGNAL = 0xBB44FF00;
23 const unsigned long RIGHT_SIGNAL = 0xBC43FF00;
24 const unsigned long STOP_SIGNAL = 0xBF40FF00;
25 const unsigned long AUTO_MODE_SIGNAL = 0xB54AFF00;
```

Grafik 15 Codeabschnitt 4

5.3.2.4 Variablen- und Objektinitialisierung

```
27 IRrecv irrecv1(IR_RECEIVER_PIN_1);
28 IRrecv irrecv2(IR_RECEIVER_PIN_2);
29 IRrecv irrecv3(IR_RECEIVER_PIN_3);
30 decode_results results;
31
32 bool autoMode = false;
```

Grafik 16 Codeabschnitt 5

- irrecv: Es werden drei Instanzen von Infrarotempfängern erstellt, um die Signale zu empfangen.
- autoMode: Dient dazu, anzuzeigen, ob sich das Fahrzeug im automatischen Fahrmodus befindet.

5.3.2.5 setup()

```
34 void setup() {  
35     pinMode(IN1, OUTPUT);  
36     pinMode(IN2, OUTPUT);  
37     pinMode(IN3, OUTPUT);  
38     pinMode(IN4, OUTPUT);  
39  
40     irrecv1.enableIRIn(); // Ersten Infrarotempfänger aktivieren  
41     irrecv2.enableIRIn(); // Zweiten Infrarotempfänger aktivieren  
42     irrecv3.enableIRIn(); // Dritten Infrarotempfänger aktivieren  
43     Serial.begin(9600); // Serielle Schnittstelle initialisieren, für Debugging  
44 }
```

Grafik 17 Codeabschnitt 6

Diese Funktion wird einmal zu Beginn des Programms ausgeführt und dient hauptsächlich der Initialisierung.

In dieser Funktion:

- Die Pins zur Steuerung der Motoren (IN1, IN2, IN3, IN4) werden auf Ausgabemodus gesetzt.
- Drei Infrarotempfänger werden aktiviert, um Infrarotsignale zu empfangen.
- Die serielle Schnittstelle wird initialisiert, um Debugging zu ermöglichen.

5.3.2.6 loop()

```
46 void loop() {  
47   if (irrecv1.decode()) {  
48     processIRSignal(irrecv1);  
49   }  
50  
51   if (irrecv2.decode()) {  
52     processIRSignal(irrecv2);  
53   }  
54  
55   if (irrecv3.decode()) {  
56     processIRSignal(irrecv3);  
57   }  
58  
59   if (autoMode) {  
60     autoDrive(); // In den Automatikmodus wechseln  
61   }  
62 }
```

Grafik 18 Codeabschnitt 7

Diese Funktion ist das Kernstück des Arduino-Programms und läuft in einer kontinuierlichen Schleife. Sie ist verantwortlich für die folgenden Operationen:

- Überprüfung, ob jeder Infrarotempfänger ein Signal empfangen hat, und Aufruf der Funktion `processIRSignal()` zur Verarbeitung.
- Wenn sich das Fahrzeug im Automatikmodus befindet, wird die Funktion `autoDrive()` aufgerufen, um die automatische Steuerung des Fahrzeugs zu übernehmen.

5.3.2.7 processIRSignal(IRrecv &irrecv)

```
void processIRSignal(IRrecv &irrecv) {  
    unsigned long value = irrecv.decodedIRData.decodedRawData;  
  
    if (value == AUTO_MODE_SIGNAL) {  
        autoMode = !autoMode; // Automatikmodus umschalten  
        if (autoMode) {  
            Serial.println("Automatische Hindernisvermeidung gestartet");  
        } else {  
            Serial.println("Automatische Hindernisvermeidung gestoppt");  
            stopCar(); // Auto stoppen  
        }  
    } else if (!autoMode) {  
        switch (value) {  
            case FORWARD_SIGNAL:  
                moveForward();  
                break;  
            case BACKWARD_SIGNAL:  
                moveBackward();  
                break;  
            case LEFT_SIGNAL:  
                turnLeft();  
                break;  
            case RIGHT_SIGNAL:  
                turnRight();  
                break;  
            case STOP_SIGNAL:  
                stopCar();  
                break;  
        }  
    }  
  
    irrecv.resume(); // Nächstes Signal empfangen  
}
```

Grafik 19 Codeabschnitt 8

Diese Funktion dient der Verarbeitung der empfangenen Infrarotsignale. Sie führt hauptsächlich die folgenden Operationen aus:

- Lesen des decodierten Infrarotsignalwerts.
- Bestimmen, ob in den Automatikmodus gewechselt oder aus diesem herausgegangen werden soll, um den Status umzuschalten und entsprechende Informationen auszugeben.
- Im Handbetrieb werden je nach unterschiedlichem Signalwert die entsprechenden Steuerfunktionen (vorwärts, rückwärts, links abbiegen, rechts abbiegen, stoppen) aufgerufen.
- Die Funktion `irrecv.resume()` wird verwendet, um sich auf den Empfang des

nächsten Infrarotsignals vorzubereiten.

5.3.2.8 Funktion zur Steuerung der Fahrzeugbewegung

Jede Funktion zur Steuerung der Bewegung des Fahrzeugs steuert die Bewegung des Fahrzeugs, indem sie die High- und Low-Pegel der Motorpins einstellt.

moveForward()

```
void moveForward() {  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
}
```

Grafik 20 Codeabschnitt 9

Diese Funktion steuert das Fahrzeug, sodass es sich vorwärts bewegt. Sie setzt die High- und Low-Pegel der Motorpins, um die Motoren in eine bestimmte Richtung drehen zu lassen.

moveBackward()

```
void moveBackward() {  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
}
```

Grafik 21 Codeabschnitt 10

Diese Funktion steuert das Fahrzeug, sodass es sich rückwärts bewegt. Ähnlich wie bei moveForward() werden jedoch die Pegelzustände umgekehrt, um die Bewegungsrichtung zu ändern.

turnLeft()

```
void turnLeft() {  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, HIGH);  
    digitalWrite(IN3, HIGH);  
    digitalWrite(IN4, LOW);  
}
```

Grafik 22Codeabschnitt 11

Diese Funktion lässt das Fahrzeug nach links abbiegen. Durch das Ändern der Pegelzustände von zwei Motoren kann das Fahrzeug an Ort und Stelle nach links drehen.

turnRight()

```
void turnRight() {  
    digitalWrite(IN1, HIGH);  
    digitalWrite(IN2, LOW);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, HIGH);  
}
```

Grafik 23Codeabschnitt 12

Diese Funktion lässt das Fahrzeug nach rechts abbiegen. Ähnlich wie bei turnLeft() ist der Zustand der gesteuerten Motoren jedoch genau umgekehrt, sodass das Fahrzeug an Ort und Stelle nach rechts drehen kann

stopCar()

```
void stopCar() {  
    digitalWrite(IN1, LOW);  
    digitalWrite(IN2, LOW);  
    digitalWrite(IN3, LOW);  
    digitalWrite(IN4, LOW);  
}
```

Grafik 24Codeabschnitt 13

Diese Funktion stoppt die Bewegung des Fahrzeugs. Durch Setzen aller Steuerpins der Motoren auf Low-Pegel wird die Drehung der Motoren gestoppt.

turnRandom()

```
void turnRandom() {  
  int randomDirection = random(0, 2); // Zufällig 0 oder 1 auswählen  
  if (randomDirection == 0) { // Nach links abbiegen  
    turnLeft();  
  } else { // Nach rechts abbiegen  
    turnRight();  
  }  
  delay(500); // Abbiegen für 0,5 Sekunden  
}  
// Funktion für das autonome Fahren
```

Um die Erfolgsquote des Fahrzeugs bei der Hindernisvermeidung zu erhöhen, wählt das Fahrzeug zufällig zwischen einer Links- oder Rechtsdrehung und hält diese Bewegung für 0,5 Sekunden aufrecht.

5.3.2.9 autoDrive()

```
141 void autoDrive() {  
142   moveForward(); // Auto fährt automatisch vorwärts  
143  
144   int distance = sonar.ping_cm(); // Distanz messen  
145   Serial.print("Distanz: ");  
146   Serial.print(distance);  
147   Serial.println(" cm");  
148  
149   if (distance > 0 && distance < 20) { // Wenn ein Hindernis erkannt wird  
150     Serial.println("Hindernis erkannt, rückwärts und rechts abbiegen");  
151     moveBackward(); // Auto rückwärts fahren  
152     delay(500); // 0.5 Sekunde rückwärts  
153     turnRandom();  
154   }  
155 }
```

Grafik 25 Codeabschnitt 14

Diese Funktion ermöglicht dem Fahrzeug die automatische Fahrt und Objekterkennung. Im Automatikmodus:

- Das Fahrzeug bewegt sich vorwärts.
- Mit dem Ultraschallsensor wird der Abstand zu Objekten vor dem Fahrzeug gemessen. Wenn die Funktion `sonar.ping_cm()` aufgerufen wird, sendet der Sensor automatisch Ultraschallwellen aus und misst die Zeit des zurückkehrenden Signals, um die Entfernung zu berechnen. Diese Funktion gibt die gemessene Entfernung in

Zentimetern zurück.

- Wenn der Abstand weniger als 20 Zentimeter beträgt, wird angezeigt, dass ein Hindernis vorliegt, woraufhin das Fahrzeug rückwärts fährt und nach rechts oder links abbiegt.

Diese Funktionen bilden gemeinsam die zentrale Operationlogik des ferngesteuerten Fahrzeugs, indem sie Infrarotsignale verarbeiten und die Daten des Ultraschallsensors kombinieren, um die Bewegungssteuerung im manuellen und automatischen Modus zu realisieren.

6 Systemtest

6.1 Test der Motorsteuerung

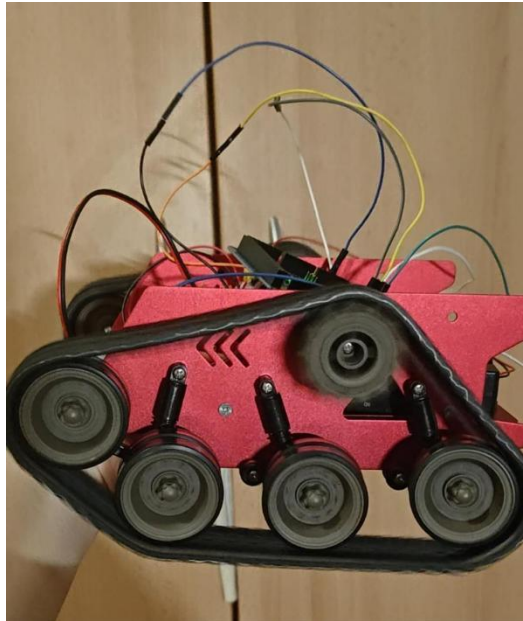
Ziel: Sicherstellen, dass der Motorantrieb korrekt funktioniert.

Schritte: Verwenden Sie den einfachsten Steuerungscode, um die Motoransteuerung einzeln zu testen. Das Fahrzeug fährt 2 Sekunden lang vorwärts, stoppt dann für eine Sekunde und fährt anschließend 2 Sekunden lang rückwärts.

```
void setup() {  
    pinMode(5, OUTPUT); // IN1  
    pinMode(6, OUTPUT); // IN2  
    pinMode(9, OUTPUT); // IN3  
    pinMode(10, OUTPUT); // IN4  
  
    // vorwärts  
    digitalWrite(5, HIGH);  
    digitalWrite(6, LOW);  
    digitalWrite(9, HIGH);  
    digitalWrite(10, LOW);  
    delay(2000);  
  
    stop();  
    delay(1000);  
  
    // rückwärts  
    digitalWrite(5, LOW);  
    digitalWrite(6, HIGH);  
    digitalWrite(9, LOW);  
    digitalWrite(10, HIGH);  
    delay(2000);  
  
    digitalWrite(5, LOW);  
    digitalWrite(6, LOW);  
    digitalWrite(9, LOW);  
    digitalWrite(10, LOW);  
}
```

Grafik 26 Motoransteuerung Testcode

Nach den Tests konnten die Motoren gemäß dem Code normal betrieben werden, und das Fahrzeug konnte problemlos vorwärts und rückwärts fahren.



Grafik 27 Prozess der Vorwärtsbewegung des Fahrzeugs

6.2 Test der Infrarot-Fernbedienungsfunktion

Ziel: Sicherstellen, dass die Signale der Fernbedienung korrekt empfangen und interpretiert werden.

Schritte: Schließen Sie den Infrarotempfänger an den Mikrocontroller an und stellen Sie sicher, dass der DO-Pin mit dem digitalen Pin 13 verbunden ist.

Laden Sie den folgenden Code hoch, um die Infrarotkodierung der Tasten im Serienmonitor beobachten zu können.

```
1  #include <IRremote.h>
2  const int RECV_PIN = 13; // Infrarotempfänger an digitalen Pin 13 anschließen
3  IRrecv irrecv(RECV_PIN);
4  decode_results results;
5
6  void setup() {
7    Serial.begin(9600); // Serielle Verbindung initialisieren, um Ausgaben zu drucken
8    irrecv.enableIRIn(); // Infrarotempfänger aktivieren
9  }
10
11 void loop() {
12   if (irrecv.decode(&results)) { // Wenn Daten empfangen werden
13     Serial.print("Dekodiertes IR-Signal: ");
14     Serial.println(results.value, HEX); // Dekodierten Wert in hexadezimaler Darstellung ausgeben
15     irrecv.resume(); // Bereit für den Empfang des nächsten Signals
16   }
17 }
```

Grafik 28 Testcode für die Infrarot-Fernbedienungsfunktion

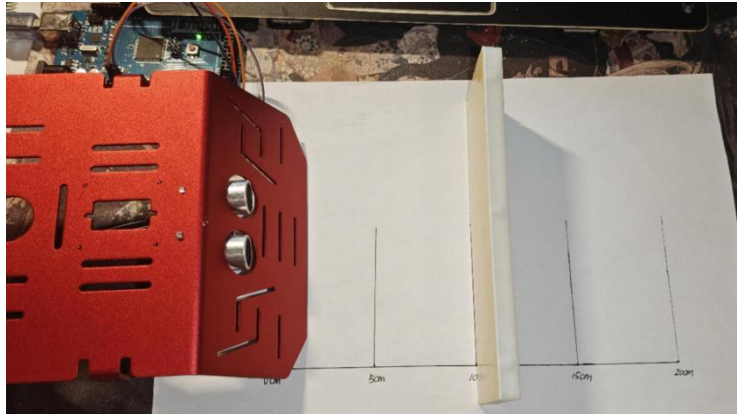
Führen Sie den Code aus und drücken Sie verschiedene Tasten auf der Fernbedienung, um die im seriellen Monitor ausgegebenen Codes zu beobachten und zu bestätigen, dass die Codes korrekt sind. Der Infrarotempfänger funktioniert ordnungsgemäß.

```
IR TYPE:NEC VALUE:BB44FF00
BB44FF00
IR TYPE:NEC VALUE:0
0
IR TYPE:NEC VALUE:BC43FF00
BC43FF00
IR TYPE:NEC VALUE:0
0
IR TYPE:NEC VALUE:BB44FF00
BB44FF00
IR TYPE:NEC VALUE:0
0
IR TYPE:NEC VALUE:BC43FF00
BC43FF00
IR TYPE:NEC VALUE:0
0
IR TYPE:NEC VALUE:BF40FF00
BF40FF00
IR TYPE:NEC VALUE:0
0
IR TYPE:NEC VALUE:B54AFF00
B54AFF00
```

Grafik 29 Die im seriellen Monitor angezeigten Infrarotkodierungen

Es kann jedoch festgestellt werden, dass unbekannte Signale das Modul kontinuierlich über den seriellen Monitor stören. Tests haben ergeben, dass diese Störsignale verschwinden, wenn sich das Fahrzeug von dem Computer entfernt. Daher kann geschlossen werden, dass elektronische Geräte möglicherweise eine gewisse Störung bei der Fernsteuerung des Fahrzeugs verursachen.

6.3 Ultraschall-Abstandsmessmodul



Grafik 30 Test der Ultraschall-Abstandsmessung

Ziel: Bestätigen, dass der Ultraschallsensor in der Lage ist, Abstände korrekt zu messen.

Schritte: Laden Sie einen einfachen Abstandsmessungscode hoch.

```
long duration, distance;
void setup() {
  Serial.begin(9600);
  pinMode(A4, OUTPUT);
  pinMode(A5, INPUT);
}

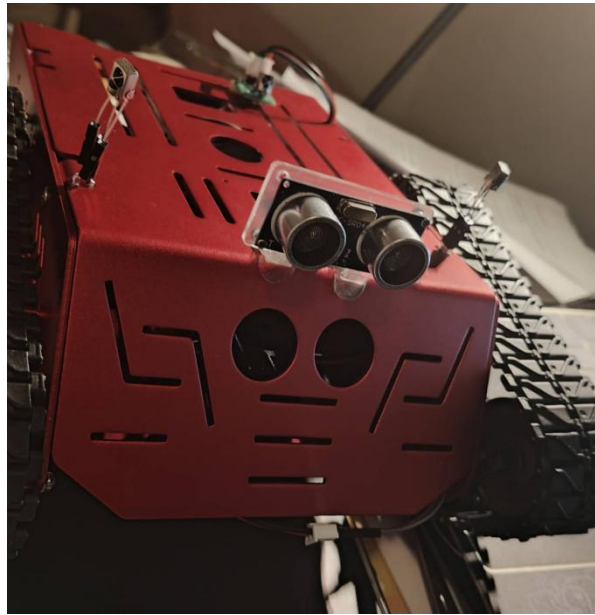
void loop() {
  digitalWrite(A4, LOW);
  delayMicroseconds(2);
  digitalWrite(A4, HIGH);
  delayMicroseconds(10);
  digitalWrite(A4, LOW);

  duration = pulseIn(A5, HIGH);
  distance = (duration * 0.034) / 2;
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");
  delay(1000);
}
```

Grafik 31 Testcode für den Ultraschallsensor

Bei den Tests traten Probleme auf, da der serielle Monitor anzeigte, dass der Abstand nach vorne kontinuierlich 0 cm betrug. Nach Tests konnte festgestellt werden, dass

der Abstand korrekt gemessen wurde, als der Winkel des Ultraschall-Abstandsmessmoduls verändert wurde. Daraus kann geschlossen werden, dass möglicherweise das Gehäuse des Fahrzeugs das Ultraschall-Abstandsmessmodul blockierte. Daher ist es notwendig, die Position des Ultraschall-Abstandsmessmoduls zu ändern. Um einen ordnungsgemäßen Betrieb des Ultraschall-Abstandsmessmoduls sicherzustellen, wurde dieses auf der oberen Ebene des Fahrzeugs montiert.



Grafik 32 Installationsdiagramm des Ultraschall-Abstandsmessmoduls

Nach den Tests konnte der serielle Monitor die Abstandsdaten ordnungsgemäß ausgeben.

7 Versuchsdurchführung

7.1 These

Um sicherzustellen, dass das Fahrzeug in gefährlichen Situationen fähig ist, zu fahren, werden die folgenden Tests für seine Funktionen durchgeführt.

1. Fernbedienungsfunktion:

- Test der Fernbedienungsreaktion in verschiedenen Richtungen.
- Vergleich des maximalen Fernbedienungsabstands im Innenbereich mit dem maximalen Fernbedienungsabstand im Freien bei Sonnenlicht.

2. Ultraschall-Abstandserkennung:

- Test der Präzision des Ultraschallsensors.
- Test der Ausweichfähigkeit

3. Test des autonomen Fahrzeugs im Hindernisvermeidungsmodus

7.2 Fernbedienungsfunktion

7.2.1 Fernbedienungsrichtung

Das Fahrzeug ist mit drei Infrarotempfängern ausgestattet, die gleichmäßig auf der Oberseite des Fahrzeugs verteilt sind, um eine Steuerung aus verschiedenen Richtungen zu ermöglichen.

Um die Fähigkeit des Fahrzeugs zu testen, auf Fernbedienungsbefehle aus unterschiedlichen Richtungen zu reagieren, wurde ein Rechtsdrehbefehl an das Fahrzeug gesendet, während der Fernbedienungsabstand gleich blieb. Da das

Fahrzeug nicht mit einem WLAN-Modul ausgestattet ist, war eine Messung über große Distanzen nicht möglich. Daher wurde eine Methode gewählt, bei der das Fahrzeug mit einem Computer verbunden ist und die Empfangszeit des Infrarotsignals über den seriellen Monitor beobachtet wurde.

Die Tests wurden bei einem Winkel von 0 Grad direkt vor dem Fahrzeug durchgeführt, und es wurden insgesamt 8 Winkel getestet. Um die Genauigkeit der Daten zu gewährleisten, wurde jeder Winkel fünfmal wiederholt.

Im Folgenden sind die gemessenen Reaktionszeiten (ms) aufgeführt.

Fernsteuerwinkel	0°	45°	90°	135 °	180°	225°	270°	315°
1	10	10	10	9	10	10	10	9
2	9	10	10	10	10	10	10	10
3	10	10	10	10	9	9	10	10
4	10	9	11	10	10	10	9	9
5	11	9	10	9	9	10	10	10

Tabelle 3 Test des Fernsteuerwinkels

Nach den Tests konnten in allen acht Richtungen die Infrarotsignale erfolgreich empfangen werden, und die Reaktionszeiten waren im Wesentlichen identisch.

7.2.2 Einfluss der Sonneneinstrahlung auf die Fernbedienung

Um den Einfluss von Infrarotstrahlung aus Sonnenlicht auf die Übertragung von Fernbedienungssignalen an das Fahrzeug zu testen, werden die maximalen Fernbedienungsdistanzen unter Innen- und Außenbedingungen verglichen, um den Grad der Störung durch Sonnenlicht auf die Funktionsweise der Infrarotfernbedienung zu bewerten.

Außentest

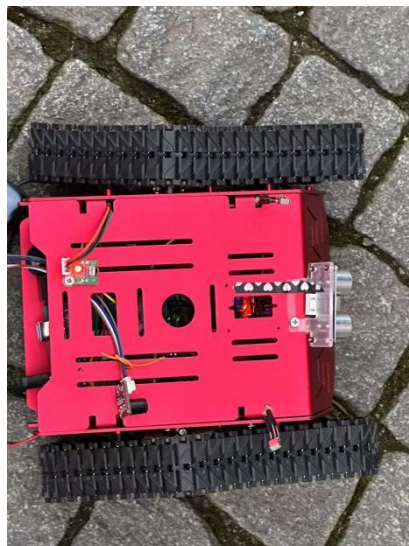
1. Umgebungsbedingungen: Wählen Sie ein offenes Gebiet ohne Hindernisse mit

starker Sonneneinstrahlung als Testgelände.

2. Startdistanz: Das Fahrzeug wird in der Ausgangsposition platziert, bei gleichmäßiger Anordnung zum Fernbediener.

3. Entfernung erhöhen: Der Abstand zwischen dem Fernbediener und dem Fahrzeug wird schrittweise erhöht, wobei alle 0,5 Meter ein Signal gesendet wird, während die Reaktion des Fahrzeugs beobachtet wird, bis kein Signal mehr empfangen wird.

4. Ergebnisse dokumentieren: Bei jedem Test wird die maximale Entfernung aufgezeichnet, bis zu der das Fahrzeug erfolgreich Befehle empfangen konnte. Laut den Testergebnissen beträgt die maximale effektive Fernbedienungsreichweite 2,7 Meter.



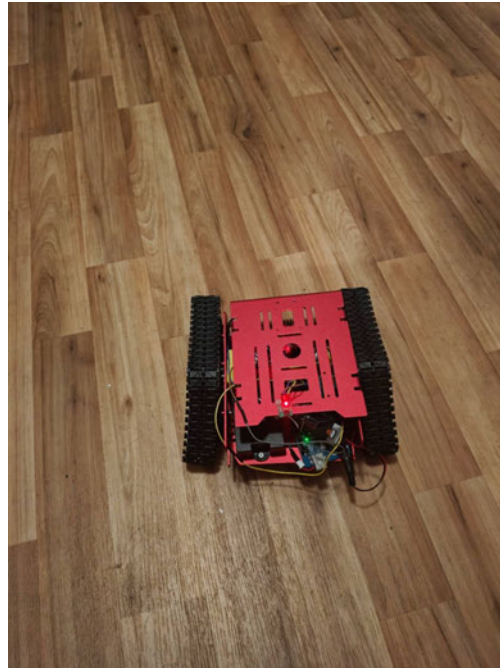
Grafik 33 Außentest

Innentest

1. Umgebungsbedingungen: Wählen Sie einen lichtschwachen Innenraum, der keine starken Störquellen für Infrarot enthält.

2. Entfernung erhöhen: Wie im Außentest wird der Abstand zwischen dem Fernbediener und dem Fahrzeug schrittweise erhöht, bis das Fahrzeug den Befehl nicht mehr empfangen kann.

3. Ergebnisse dokumentieren: Bei jedem Test wird die maximale Entfernung notiert, bis zu der das Fahrzeug erfolgreich Befehle empfangen konnte. Laut den Testergebnissen beträgt die maximale effektive Fernbedienungsreichweite 8 Meter.



Grafik 34Innentest

Experimentelle Analyse und Bewertung

- Ergebnisvergleich:

- Die maximale Fernbedienungsdistanz im Freien beträgt 2,7 Meter, während sie im Innenraum 8 Meter erreicht. Dies zeigt, dass die Empfangsleistung der Infrarotfernbedienung des Fahrzeugs in indoor-Umgebungen aufgrund der besseren Lichtverhältnisse und der geringeren Störquellen erheblich besser ist als im Freien.

- Einflussfaktorenanalyse:

- Intensität des Sonnenlichts: Die Infrarotanteile im Sonnenlicht können die effektive Übertragung von Infrarotsignalen stören, was zu einer Verringerung der Empfangsdistanz führt.

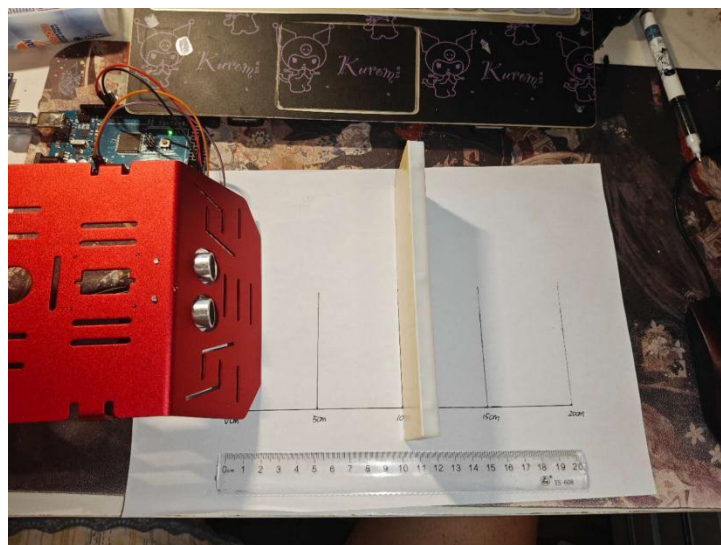
- Umgebungsstörungen: Im Freien können Lichtverschmutzung und andere Störungen (wie Fahrzeuge, fluoreszierende Lampen) die Signalqualität beeinträchtigen. Im Innenraum hingegen trägt die geschlossene Umgebung zur Klarheit des Signals bei.

7.3 Ultraschall-Abstandserkennung

7.3.1 Test der Messgenauigkeit des Ultraschallsensors

Das Fahrzeug muss in der Lage sein, Hindernisse zu umgehen, die sich in einer Distanz von 20 cm befinden. Daher ist es erforderlich, dass das Fahrzeug in der Lage ist, einen bestimmten Abstand vor sich zu messen. Im Rahmen dieser Untersuchung wurde die Messgenauigkeit des Ultraschallsensors im Ultraschall-Abstandserkennungsfahrzeug im Bereich von 0 bis 40 Zentimetern getestet. Durch mehrfache Messungen in einer stabilen Testumgebung wurde der Fehler zwischen den Messergebnissen des Sensors und der tatsächlichen Distanz analysiert.

Um den Test durchzuführen, wurden mehrere Testziele in unterschiedlichen Distanzen (0, 5, 10, 15, 20, 25, 30, 35, 40 cm) vorbereitet, wobei darauf geachtet wurde, dass ihre Oberflächen eben sind.



Grafik 35 Prozess der Messgenauigkeitstests für Ultraschall

Jede durchgeführte Messung wird in einer Tabelle festgehalten, und der durchschnittliche Fehler für jede experimentelle Distanz wird berechnet.

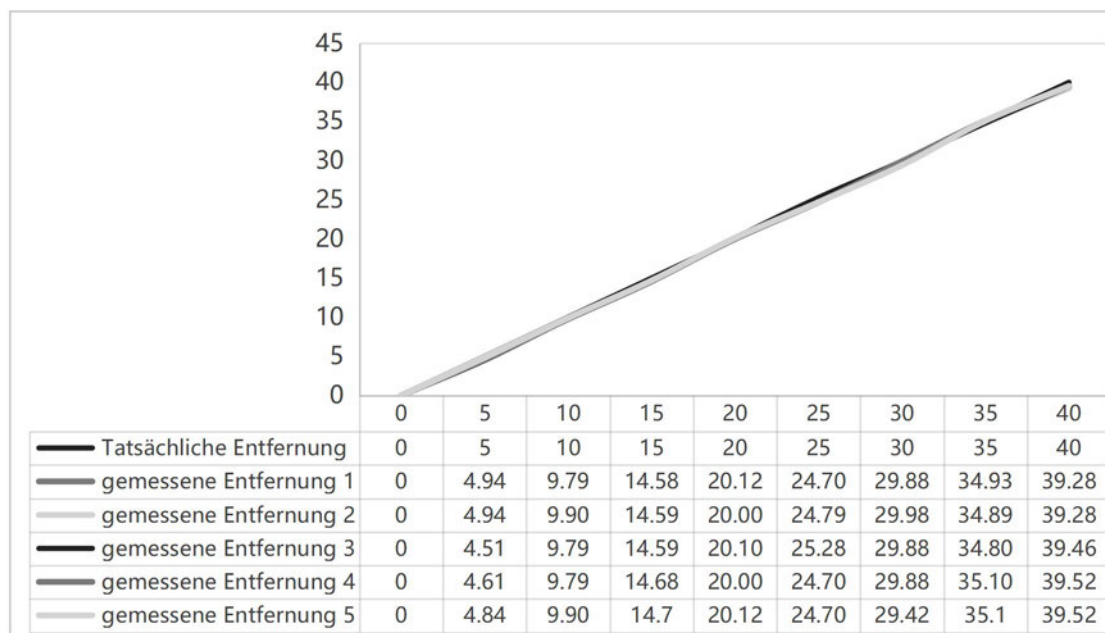


Tabelle 4 Tabelle der Ultraschall-Distanzmessdaten

Tatsächliche Entfernung	0	5	10	15	20	25	30	35	40
Durchschnittlicher Fehler	0	0.232	0.186	0.372	0.068	0.278	0.192	0.116	0.588

Tabelle 5 Tabelle zur Analyse der Ultraschall-Distanzmessfehler

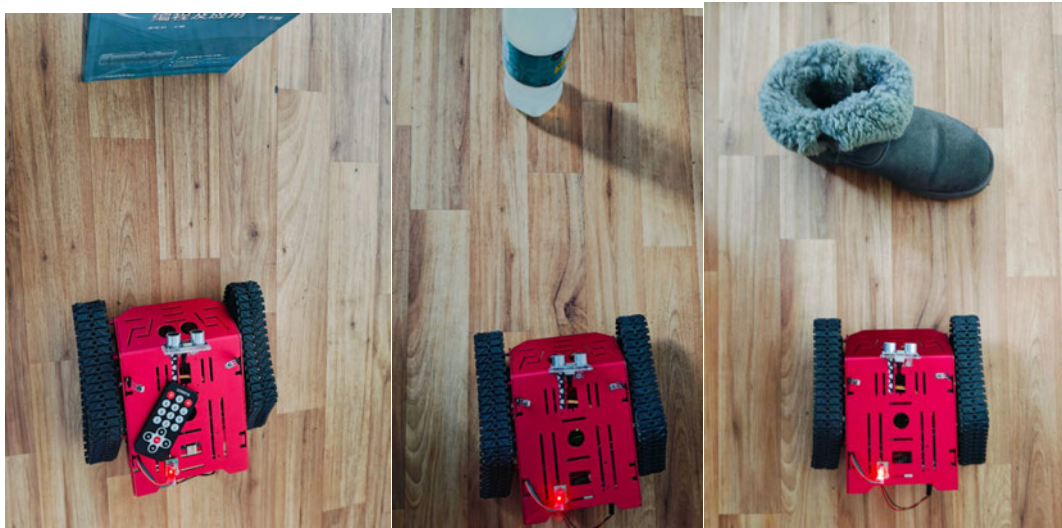
Obwohl die durchschnittlichen Fehler zwischen den verschiedenen Gruppen schwanken, sind die Fehlerwerte im Allgemeinen relativ gering (zwischen 0,1 cm und 0,6 cm) und liegen im akzeptablen Rahmen. Dies zeigt, dass die Ultraschallmessungen in diesen Distanzen insgesamt relativ genau sind.

7.3.2 Test der Ausweichfähigkeit

In diesem Test wird die Ausweichfähigkeit des Ultraschall-Abstandserkennungsfahrzeugs bei unterschiedlichen Arten von Hindernissen überprüft, einschließlich der Wirksamkeit beim Rückwärtsfahren,

Rechtsabbiegen und Wiederherstellen der Vorwärtsbewegung.

Jedes Hindernis wird in einer Entfernung von 20 Zentimetern vor dem Fahrzeug platziert, und die Reaktion des Fahrzeugs bei jedem Hindernis wird aufgezeichnet. Für jede Art von Hindernis werden jeweils 3 Testdurchläufe durchgeführt, und die Experimentergebnisse werden dokumentiert.



Grafik 36 Prozess des Tests der Ausweichfähigkeit

Testdurchlauf	Hindernisse	Rückschritt	Rechtsabbiegen	Vorwärts	Reaktionszeit
1	Buch	✓	✓	✓	1.2
2	Buch	✓	✓	✓	1.1
3	Buch	✓	✓	✓	1.3
1	Flasche	✓	✓	✓	1.0
2	Flasche	✓	✓	✓	1.1
3	Flasche	✓	✓	✓	1.2
1	Schuh	✓	✓	✓	0.9
2	Schuh	✓	✓	✓	1.0
3	Schuh	✓	✓	✓	1.1

Tabelle 6 Tabelle zur Überprüfung der Ausweichfähigkeit

Erfolgsquote der Ausweichfähigkeit:

- Bücher: 100% (3/3)

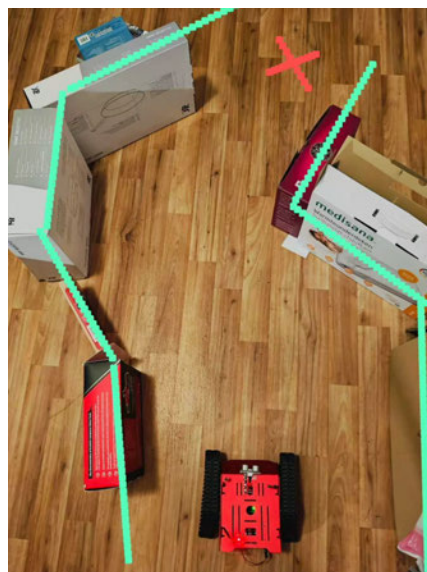
- Wasserflasche: 100% (3/3)
- Schuhe: 100% (3/3)

- Durchschnittliche Reaktionszeit:
 - Bücher: 1,2 Sekunden
 - Wasserflasche: 1,1 Sekunden
 - Schuhe: 1,0 Sekunden

Das Fahrzeug konnte erfolgreich allen drei unterschiedlichen Arten von Hindernissen – Bücher, Wasserflasche und Schuhe – ausweichen, was darauf hindeutet, dass die Ausweichfunktion ordnungsgemäß funktioniert. Die durchschnittliche Reaktionszeit des Fahrzeugs lag stabil zwischen 1,0 und 1,2 Sekunden.

7.4 Autonomes Fahren

Um die Fähigkeit des Fahrzeugs zur automatischen Hindernisvermeidung zu überprüfen, wurde ein einfaches Szenario entworfen, um die Durchfahrtstfähigkeit und die Durchfahrtszeit des Fahrzeugs in einer Kurvensituation zu testen.



Grafik 37Szeneinstellung

Um die Genauigkeit der Daten sicherzustellen, wurden hier fünf Tests durchgeführt und die benötigte Zeit aufgezeichnet.

Test-Rundenanzahl	Zugangszustand	Zugangszeit (s)
1	×	
2	✓	40.69
3	×	
4	✓	50.38
5	✓	27.57

Tabelle 7 Automatisierter Fahrttest

Bewertung

1. Vorteile:

- Das Fahrzeug zeigt eine gewisse Erfolgsquote bei der Überwindung von Hindernissen (60%), was darauf hindeutet, dass das System unter bestimmten Bedingungen effektiv Hindernisse vermeiden kann.

- Die erfolgreiche Durchfahrtszeit ist relativ schnell (maximal 27,57 Sekunden), was bedeutet, dass das Fahrzeug bei der effektiven Erkennung von Hindernissen und der Umsetzung entsprechender Maßnahmen schnell fahren kann.

2. Nachteile:

- Mit einer Fehlerrate von 40% (2 Tests) deutet dies darauf hin, dass das Fahrzeug in bestimmten Situationen möglicherweise nicht in der Lage ist, Hindernisse rechtzeitig zu erkennen oder angemessen zu reagieren.

- Die Schwankungen in der Durchfahrtszeit sind erheblich; die 50,38 Sekunden im zweiten Test zeigen, dass die Fahreffizienz des Fahrzeugs in bestimmten Situationen verbessert werden muss.

Erfolgsquote: 60%

Durchschnittliche erfolgreiche Durchfahrtszeit: 39,55 Sekunden

Die Fähigkeit des Fahrzeugs zur automatischen Hindernisvermeidung hat in den

aktuellen Tests eine gewisse Effektivität gezeigt, jedoch muss die Reaktionsgeschwindigkeit sowie die Hindernisvermeidungsstrategie optimiert werden, um die Gesamt Erfolgsquote und Effizienz des Fahrens zu erhöhen.

Zusammenfassung

Das Ziel dieses Projekts war die Entwicklung eines Transportsystems, das in gefährlichen Umgebungen operieren kann, um die Effizienz und Sicherheit im Transport in komplexen oder potenziell gefährlichen Bereichen zu verbessern. Durch den Bau eines Ultraschall-Obstacle-Avoidance-RC-Cars haben wir erfolgreich die grundlegenden Funktionen und die automatische Navigationsfähigkeit des Transportsystems realisiert.

Bei der Umsetzung des Projekts wurden folgende Schlüsselkomponenten verwendet:

1. L298N-Treibermodul: Dieses Modul steuert die Motoren des Fahrzeugs und ermöglicht das Vorwärts- und Rückwärtsfahren, die Richtungsänderung und das Anhalten, und sorgt für die stabile Energieversorgung.
2. HX1838-Infrarot-Fernbedienungsmodul: Durch den Einsatz von drei Infrarotempfängern konnten wir die Fernsteuerung realisieren, die eine Vielzahl von Fahrbefehlen unterstützt und die Bedienflexibilität erhöht.
3. SR04-Ultraschallsensor: Dieser Sensor wird zur Objekterkennung verwendet, und im automatischen Fahrmodus kann das Fahrzeug effektiv Hindernisse erkennen und sicher umfahren.
4. Gummikettenräder: Das Fahrzeug verwendet Gummikettenräder, um die Stabilität und die Geländegängigkeit unter verschiedenen Bodenbedingungen zu erhöhen.

Durchführung und Ergebnisse

Nach mehreren Experimenten und Anpassungen zeigte das Fahrzeug in verschiedenen Umgebungen eine gute Leistung. Die Fernsteuerungsfunktion arbeitete stabil, und das Fahrzeug reagierte schnell auf empfangene Befehle. Im automatischen Fahrmodus konnte das Fahrzeug Hindernisse im 20 cm Bereich vor

sich genau erkennen und automatische Ausweichmanöver durchführen, um Kollisionen zu vermeiden.

Die Ergebnisse zeigen, dass das Fahrzeug nach der Integration aller Module die festgelegten Fahr- und Ausweichfunktionen erfolgreich umgesetzt hat. Darüber hinaus erhöht die Kombination aus Fernsteuerungssteuerung und automatischer Hindernisvermeidung die Anpassungsfähigkeit des Fahrzeugs in komplexen Umgebungen, wodurch es in bestimmten gefährlichen Situationen sicher betrieben werden kann.

Ausblick

Die erfolgreiche Umsetzung dieses Projekts legt eine solide Grundlage für die weitere Entwicklung des Ultraschall-Obstacle-Avoidance-RC-Cars für den Einsatz in gefährlichen Umgebungen. Zukünftige Arbeiten werden sich auf die folgenden Aspekte konzentrieren, um die Leistung und das Anwendungsspektrum des Fahrzeugs zu verbessern:

1. Algorithmusoptimierung: Weitere Forschung und Optimierung der Ausweichalgorithmen, um die Reaktionsgeschwindigkeit und Genauigkeit des Fahrzeugs in komplexen Umgebungen zu erhöhen. Durch die Einführung von Techniken des maschinellen Lernens kann das Fahrzeug seine Fähigkeit zur Erkennung und Anpassung an die Umgebung kontinuierlich verbessern.
2. Sensorintegration: Die Erwägung der Integration weiterer Sensortypen (wie Lasermesssensoren, Lichtsensoren usw.), um die Umgebungswahrnehmung des Fahrzeugs zu verstärken und die Navigations- und Ausweichfähigkeiten in komplexeren Szenarien zu erhöhen.
3. Erweiterung der Kommunikationsfunktionen: Untersuchung der Einführung eines drahtlosen Kommunikationsmoduls, um den Status und die Position des Fahrzeugs aus der Ferne zu überwachen. Dies wird dazu beitragen, Daten zu den Operationen des Fahrzeugs in Echtzeit abzurufen und die Sicherheit zu erhöhen.
4. Haltbarkeitstests und Umweltanpassungsfähigkeit: Durchführung von Tests zur Haltbarkeit und Anpassungsfähigkeit des Fahrzeugs, um sicherzustellen, dass es unter verschiedenen klimatischen Bedingungen und Geländearten ordnungsgemäß funktioniert. Verbesserung der verwendeten Materialien und Strukturen, um die Stoßfestigkeit, Fallfestigkeit und Wasserbeständigkeit des Fahrzeugs zu erhöhen.

Durch die Umsetzung dieser Maßnahmen erwarten wir, dieses Projekt weiterzuentwickeln und die Zuverlässigkeit und Anwendbarkeit des Ultraschall-Obstacle-Avoidance-RC-Cars in gefährlichen Umgebungen zu erhöhen, um dessen Verbreitung und Nutzung in der Praxis voranzutreiben.

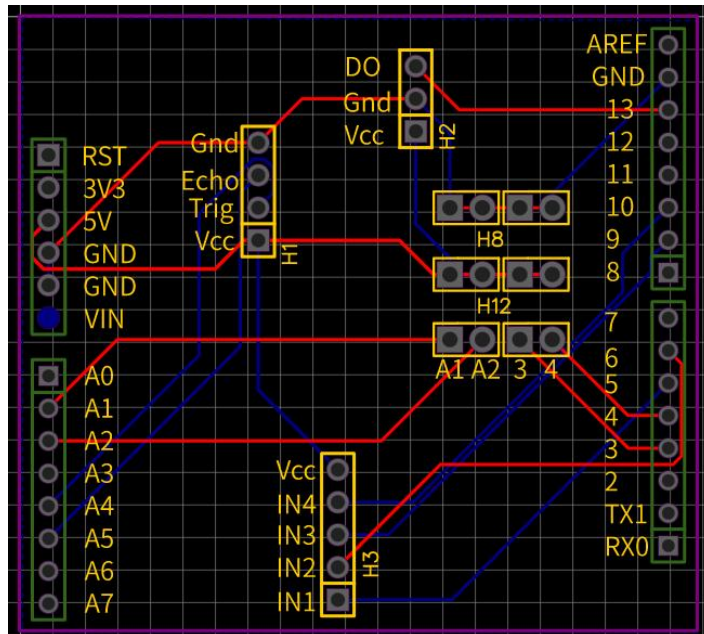
Literaturverzeichnis

- [1] Zi-rong Luo, Jian-zhong Shang, Zhi-xiong Zhang. A reconfigurable tracked mobile robot based on four-linkage mechanism[J]. Journal of Central South University, 2013, 20(1)
- [2] Wang Shuo. Research on Self-adaptive Tracked Mobile Mechanisms for Obstacle Crossing [D]. Beijing Jiaotong University, 2018
- [3] Xu Guobao, Yin Yixin, Zhou Meijuan. Current Status and Prospects of Intelligent Mobile Robot Technology [J]. Robotics Technology and Applications, 2007 (02): 29-34
- [4] He, Wujian, Qiu, Haofeng, Qu, Xiaokun, Chen, Jiale, Sun, Junqiao, and Yu, Tenglun. "Research on the Development and Application of Intelligent Vehicles." Tianjin Vocational Technical Normal University; Tianjin High-end Intelligent CNC Machine Tool Engineering Research Center, Tianjin 300222, 2022.
- [5] Arduino. (2023). Arduino Mega 2560 Rev3.
<https://store.arduino.cc/products/arduino-mega-2560-rev3>
- [6] Microchip. (2014). ATmega2560 Datasheet.
http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-2549-8-bit-AVR-Microcontroller-ATmega640-1280-1281-2560-2561_datasheet.pdf
- [7] Shuanglong Electronics Technology (2020). L298 Dual H-Bridge DC Motor Driver Board (Upgraded Version) Instruction Manual.
- [8] L. H. Wang, "Design of infrared remote control key coding," Times Economic and Trade, no. 5, pp. 1, 2013.
- [9] G. P. Huang and J. L. Wang, "Design and implementation of Bluetooth remote control system," School of Communication Engineering, PLA University of Science and Technology, Nanjing 210007, China, 2004
- [10] L. Yang, Y. Chen, H. Q. Zhang, et al., "A wireless synchronous electronic clock system," China, 203480243 UEP, 2014.
- [11] Yuxin Electronics (2019). YX62876 - Infrared Remote Control Module

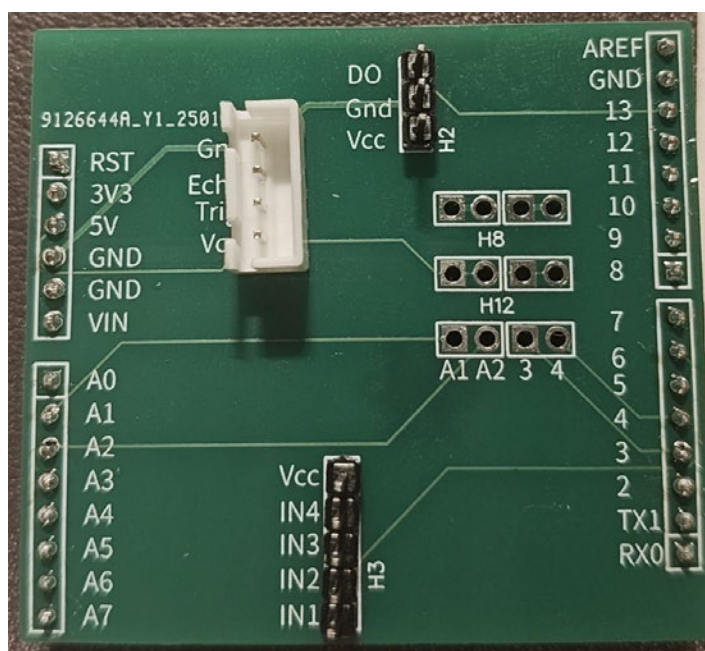
-
- [12] T. Liu, "Design and implementation of an infrared distance measurement sensor based on STC microcontroller," Shenzhen Technician College, 2018.
- [13] Wen, Qiufang. "Output-Oriented Approach" and Teaching Chinese as a Foreign Language [J]. World Chinese Teaching, 2018, 32(3): 387-400
- [14] Wu, Ruirui, Zhu, Xiaofeng, and Song, Zongfeng. "Obstacle Avoidance Technology Based on HC-SR04 Multi-Ultrasound." (Guangzhou College of Commerce, Foshan, Guangdong 528000)

Anlagen

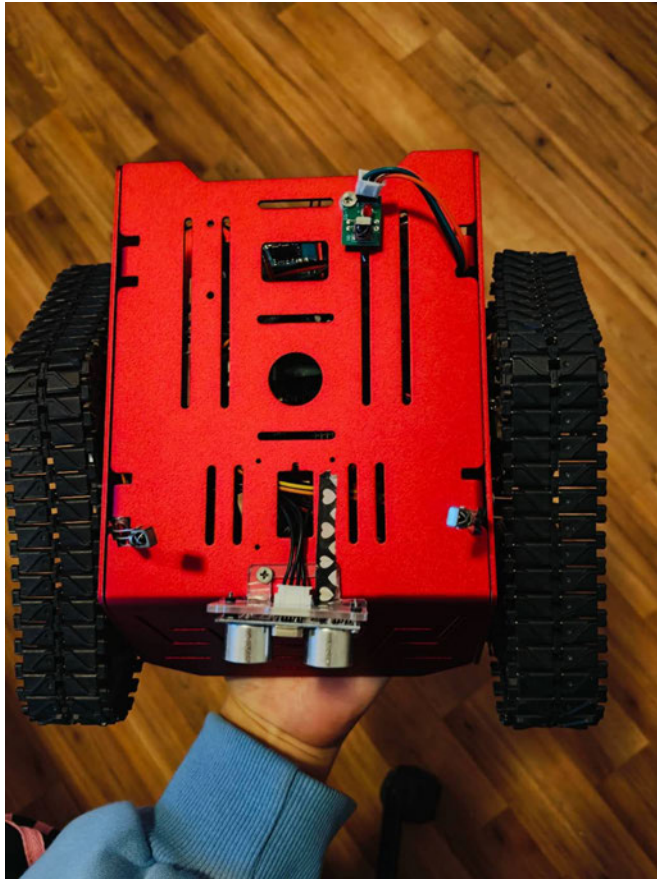
PCB



Gelötete Leiterplatte



Physische Zeichnung des Autos



Software

```
1  #include <IRremote.hpp>
2  #include <NewPing.h>
3
4  // Definition der Pins
5  const int IN1 = 5;
6  const int IN2 = 6;
7  const int IN3 = 9;
8  const int IN4 = 10;
9  const int IR_RECEIVER_PIN_1 = 13; // Erster Infrarotempfänger
10 const int IR_RECEIVER_PIN_2 = 3;  // Zweiter Infrarotempfänger
11 const int IR_RECEIVER_PIN_3 = 4;  // Dritter Infrarotempfänger
12 const int TRIG_PIN = A4;
13 const int ECHO_PIN = A5;
14
15 // Parameter des Ultraschallsensors
16 #define MAX_DISTANCE 200 // Maximale Messdistanz des Ultraschallsensors festlegen
17 NewPing sonar(TRIG_PIN, ECHO_PIN, MAX_DISTANCE);
18
19 // Definition der Infrarot-Fernbedienungssignale
20 const unsigned long FORWARD_SIGNAL = 0xB946FF00;
21 const unsigned long BACKWARD_SIGNAL = 0xEA15FF00;
22 const unsigned long LEFT_SIGNAL = 0xBB44FF00;
23 const unsigned long RIGHT_SIGNAL = 0xBC43FF00;
24 const unsigned long STOP_SIGNAL = 0xBF40FF00;
25 const unsigned long AUTO_MODE_SIGNAL = 0xB54AFF00;
26
27 IRrecv irrecv1(IR_RECEIVER_PIN_1);
28 IRrecv irrecv2(IR_RECEIVER_PIN_2);
29 IRrecv irrecv3(IR_RECEIVER_PIN_3);
30 decode_results results;
31
32 bool autoMode = false;
33
34 void setup() {
35     pinMode(IN1, OUTPUT);
36     pinMode(IN2, OUTPUT);
37     pinMode(IN3, OUTPUT);
38     pinMode(IN4, OUTPUT);
39
40     irrecv1.enableIRIn(); // Ersten Infrarotempfänger aktivieren
41     irrecv2.enableIRIn(); // Zweiten Infrarotempfänger aktivieren
42     irrecv3.enableIRIn(); // Dritten Infrarotempfänger aktivieren
43     Serial.begin(9600); // Serielle Schnittstelle initialisieren, für Debugging
44 }
45
46 void loop() {
47     if (irrecv1.decode()) {
48         processIRSignal(irrecv1);
49     }
50
51     if (irrecv2.decode()) {
52         processIRSignal(irrecv2);
53     }
54
55     if (irrecv3.decode()) {
56         processIRSignal(irrecv3);
57     }
58
59     if (autoMode) {
60         autoDrive(); // In den Automatikmodus wechseln
61     }
62 }
```

```
63 void processIRSignal(IRrecv &irrecv) {
64     unsigned long value = irrecv.decodedIRData.decodedRawData;
65
66     if (value == AUTO_MODE_SIGNAL) {
67         autoMode = !autoMode; // Automatikmodus umschalten
68         if (autoMode) {
69             Serial.println("Automatische Hindernisvermeidung gestartet");
70         } else {
71             Serial.println("Automatische Hindernisvermeidung gestoppt");
72             stopCar(); // Auto stoppen
73         }
74     } else if (!autoMode) {
75         switch (value) {
76             case FORWARD_SIGNAL:
77                 moveForward();
78                 break;
79             case BACKWARD_SIGNAL:
80                 moveBackward();
81                 break;
82             case LEFT_SIGNAL:
83                 turnLeft();
84                 break;
85             case RIGHT_SIGNAL:
86                 turnRight();
87                 break;
88             case STOP_SIGNAL:
89                 stopCar();
90                 break;
91         }
92     }
93
94     irrecv.resume(); // Nächstes Signal empfangen
95 }
96
97 void moveForward() {
98     digitalWrite(IN1, HIGH);
99     digitalWrite(IN2, LOW);
100     digitalWrite(IN3, HIGH);
101     digitalWrite(IN4, LOW);
102 }
103
```

```
104 void moveBackward() {
105     digitalWrite(IN1, LOW);
106     digitalWrite(IN2, HIGH);
107     digitalWrite(IN3, LOW);
108     digitalWrite(IN4, HIGH);
109 }
110
111 void turnLeft() {
112     digitalWrite(IN1, LOW);
113     digitalWrite(IN2, HIGH);
114     digitalWrite(IN3, HIGH);
115     digitalWrite(IN4, LOW);
116 }
117
118 void turnRight() {
119     digitalWrite(IN1, HIGH);
120     digitalWrite(IN2, LOW);
121     digitalWrite(IN3, LOW);
122     digitalWrite(IN4, HIGH);
123 }
124
125 void stopCar() {
126     digitalWrite(IN1, LOW);
127     digitalWrite(IN2, LOW);
128     digitalWrite(IN3, LOW);
129     digitalWrite(IN4, LOW);
130 }
131 void turnRandom() {
132     int randomDirection = random(0, 2); // Zufällig 0 oder 1 auswählen
133     if (randomDirection == 0) { // Nach links abbiegen
134         turnLeft();
135     } else { // Nach rechts abbiegen
136         turnRight();
137     }
138     delay(500); // Abbiegen für 0,5 Sekunden
139 }
140 // Funktion für das autonome Fahren
141 void autoDrive() {
142     moveForward(); // Auto fährt automatisch vorwärts
143
144     int distance = sonar.ping_cm(); // Distanz messen
145     Serial.print("Distanz: ");
146     Serial.print(distance);
147     Serial.println(" cm");
148
149     if (distance > 0 && distance < 20) { // Wenn ein Hindernis erkannt wird
150         Serial.println("Hindernis erkannt, rückwärts und rechts abbiegen");
151         moveBackward(); // Auto rückwärts fahren
152         delay(500); // 0.5 Sekunde rückwärts
153         turnRandom();
154     }
155 }
```

Eigenständigkeitserklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe. Stellen, die wörtlich oder sinngemäß aus Quellen entnommen wurden, sind als solche kenntlich gemacht. Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt.

Mittweida, 01.02.2025

Yue Wang

