

# NavDriveX

Projekt: Autonom fahrendes Auto

---

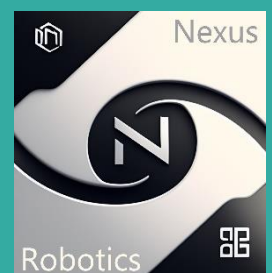


24. JULI 2023

---

**Nexus Robotics**

**Verfasst von: Anna-Lena Bensch**



---

## Inhalt

<b>Unser Ziel .....</b>	<b>4</b>
<b>Ist-Analyse.....</b>	<b>4</b>
<b>Soll-Konzept.....</b>	<b>4</b>
Projektziel: .....	4
Projektteam: .....	4
Projektorganisation: .....	5
Projektumfang: .....	5
Zeitplan: .....	5
Ressourcen: .....	5
Qualitätssicherung: .....	5
Risikomanagement:.....	5
Kommunikation: .....	5
Abschlusstest und -abnahme:.....	6
<b>Projektstrukturplan .....</b>	<b>6</b>
<b>Marktanalyse .....</b>	<b>6</b>
<b>Kostenplan .....</b>	<b>7</b>
<b>Pflichtenheft .....</b>	<b>8</b>
Zielbestimmung .....	8
Hardwareanforderung.....	8
Softwareanforderung.....	9
Branding & Vermarktung.....	9
Kosten- Zeit- & Qualitätsmanagement .....	9
Projektdokumentation .....	10
<b>Qualitätssicherungsplan .....</b>	<b>10</b>
Verantwortlichkeiten.....	10
Überprüfungs- und Bewertungskriterien .....	11
Prozesse und Verfahren .....	12
Zeitplanentwurf für den Zeitraum vom 24. Juli 2023 bis zum 08. September 2023 .....	13
Ressourcen.....	13
Risikomanagement.....	14
Kommunikation .....	15
Abweichungs- und Korrekturmaßnahmen .....	16
Überprüfung und Aktualisierung.....	17
<b>Projektzeitplan.....</b>	<b>19</b>
<b>Ressourcenplan .....</b>	<b>19</b>

---

Ressourcenplan für Nexus Robotics - Projekt NavDriveX .....	19
Ressourcentypen.....	19
Ressourcenbedarf .....	20
Ressourcenverfügbarkeit.....	21
Ressourcenüberwachung und -kontrolle .....	21
Risikobewertung und -management .....	21
Dokumentationen .....	22
Dokumentation Website .....	22
Dokumentation Software.....	23
Dokumentation Add-on Programm Video-Objekterkennung .....	27
Projekt-Dokumentation: Raspberry Pi Roboter-Auto.....	29

---

## Unser Ziel

Das Ziel von Nexus Robotics ist es, bis zum 8. September dieses Jahres einen funktionsfähigen Prototyp eines autonom fahrenden Miniaturautos zu entwickeln und zu realisieren. Der Prototyp soll sicher und zuverlässig auf einer vordefinierten Teststrecke autonom navigieren können, ohne dass ein Fahrer eingreifen muss. Die Erfolgsrate des Prototyps bei der Einhaltung der Teststrecke soll mindestens 80% betragen. Das autonome Miniaturauto soll ein bahnbrechendes Fahrerlebnis bieten, dass Kinder und Erwachsene gleichermaßen begeistert und neugierig auf die innovative Technologie des autonomen Fahrens macht.

Durch unsere Vorreiterrolle in der Spielzeugindustrie, wollen wir einen wichtigen Beitrag zur Weiterentwicklung der Branche leisten, und neuen Spielspaß mit fortschrittlicher Technologie verbinden.

## Ist-Analyse

Zum aktuellen Zeitpunkt befindet sich das Projekt "NavDriveX" in einer frühen Entwicklungsphase. Das Projektteam hat mit der Arbeit an der Hardware- und Softwareentwicklung für den Prototypen eines autonom fahrenden Miniaturautos begonnen. Es wurden bereits erste Schritte unternommen.

## Soll-Konzept

### Projektziel:

Das Hauptziel des Projekts "NavDriveX" ist die Entwicklung eines funktionsfähigen Prototyps eines autonom fahrenden Miniaturautos, das ohne Eingreifen eines Fahrers sicher und zuverlässig eine vordefinierte Teststrecke befahren kann. Der Prototyp soll Hindernisse erkennen und diese autonom, und somit selbstständig, umfahren können.

### Projektteam:

Das Projektteam besteht aus 5 Mitarbeitern der Firma Nexus Robotics:

- Frau A. Bensch
- Herrn C. Cuhaci
- Herrn K. Driesener
- Herrn S. Warner
- Herrn A. Dahir

Bensch und Cuhaci übernehmen die Leitung des Teams. Für den Schwerpunkt der Hardware und hardwarenahe Software sind Warner, Cuhaci und Dahir verantwortlich,

---

während Driesener und Bensch den Schwerpunkt der Software und den reinen Code abdecken.

**Projektorganisation:**

Die Projektorganisation wird hierarchisch unterteilt, wobei Bensch und Cuhaci als Projektleiter für die Gesamtkoordination und Überwachung verantwortlich sind. Das Team wird eng zusammenarbeiten, um einen reibungslosen Entwicklungsprozess sicherzustellen.

**Projektumfang:**

Der Projektumfang umfasst die Entwicklung eines Prototyps des autonom fahrenden Miniaturautos "NavDriveX". Dieser Prototyp wird auf der vordefinierten Teststrecke autonom navigieren und Hindernisse erkennen und umfahren können.

**Zeitplan:**

Das Projekt soll bis zum 8. September dieses Jahres abgeschlossen sein, um den funktionsfähigen Prototyp rechtzeitig fertigzustellen und zu präsentieren. Es wird ein detaillierter Zeitplan erstellt, der Meilensteine und Zwischenziele enthält, um den Fortschritt des Projekts zu überwachen.

**Ressourcen:**

Das Projektteam wird die zur Verfügung gestellten Materialien von den Auftraggebern, der EIMa GmbH, nutzen, um den Prototypen zu entwickeln. Die Ressourcen des Unternehmens, einschließlich der Labore und Ausrüstung, stehen dem Team zur Verfügung.

**Qualitätssicherung:**

Die Qualität des Projekts wird kontinuierlich überwacht und sichergestellt. Regelmäßige Tests und Überprüfungen werden durchgeführt, um sicherzustellen, dass der Prototyp die Anforderungen erfüllt und sicher und zuverlässig auf der Teststrecke navigiert.

**Risikomanagement:**

Risiken, die den Projektfortschritt beeinflussen könnten, werden identifiziert und bewertet. Ein Risikomanagementplan wird erstellt, um geeignete Maßnahmen zur Risikominimierung und -bewältigung zu ergreifen.

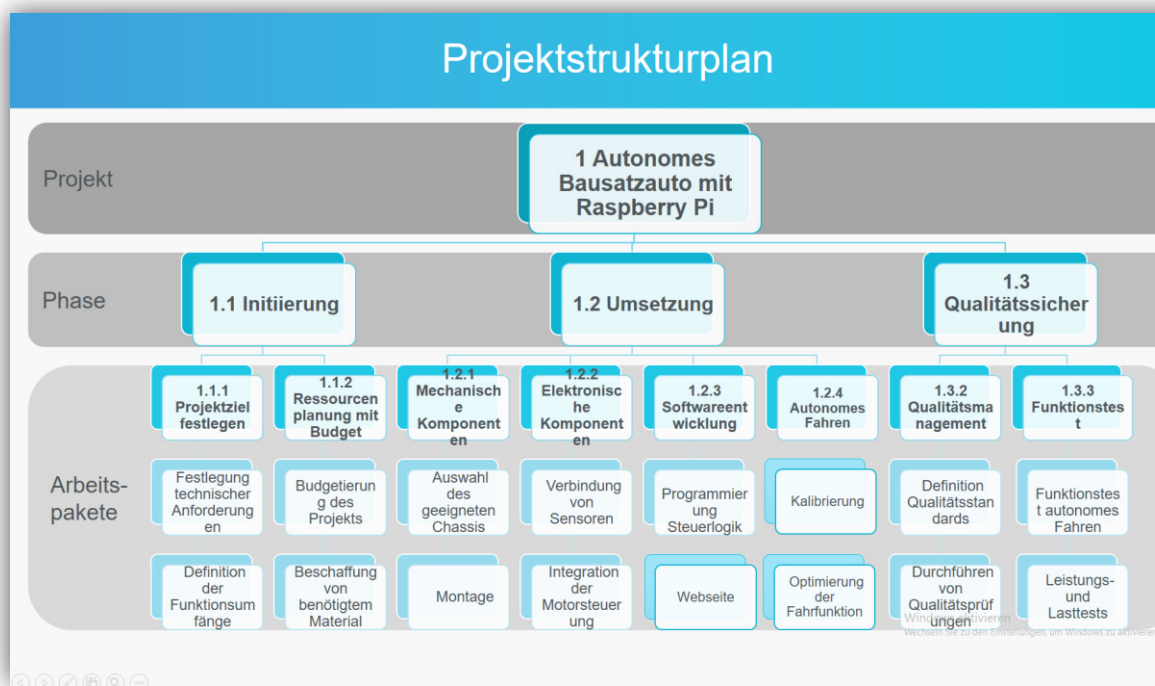
**Kommunikation:**

Die Kommunikation im Projektteam und mit den Auftraggebern ist von zentraler Bedeutung. Regelmäßige Meetings und Berichte werden abgehalten, um den Austausch von Informationen und den Fortschritt des Projekts sicherzustellen.

## Abschlusstest und -abnahme:

Vor dem Abschluss des Projekts wird der Prototyp ausgiebig getestet, um sicherzustellen, dass er alle Anforderungen erfüllt und sicher und zuverlässig auf der Teststrecke agiert. Die Abnahme erfolgt durch die Auftraggeber, die EIMA GMBH.

## Projektstrukturplan



## Marktanalyse

Elegoo

Nexus  
Robotics

Adapt Smart  
Car Kit

Sun  
Founder

<u>Eigenschaften</u>	<u>Nexus Robotics</u>	<u>Adept Smart Car Kit</u>	<u>Elegoo</u>	<u>Sun Founder</u>
<b>Fahren</b>	<b>Ja</b>	<b>Ja</b>	<b>Ja</b>	<b>Ja</b>
<b>Betriebssystem</b>	<b>Raspbian</b>	<b>Arduino</b>	<b>Arduino</b>	<b>Arduino</b>
<b>Inkl. Batterien</b>	<b>Ja</b>	<b>Nein</b>	<b>Nein</b>	<b>Nein</b>
<b>Garantie</b>	<b>2 Jahre</b>	<b>2 Jahre</b>	<b>2 Jahre</b>	<b>2 Jahre</b>
<b>Versandkosten</b>	<b>Nein</b>	<b>Ja</b>	<b>Ja</b>	<b>Ja</b>
<b><u>Verkaufspreis =</u></b>	<b><u>532,00€</u></b>	<b><u>479,00€</u></b>	<b><u>512,09€</u></b>	<b><u>495,06€</u></b>

## Kostenplan

In diesem Absatz haben wir Ihnen einen Kostenplan erstellt. Wir bedanken uns für Ihre Anfrage und das freundliche Telefonat.

Der Kostenplan wird wie folgt zusammengesetzt:

Posten	Einkauf Br	Einkauf Ne	Verkauf Ne	Verkauf Br	Marge
Materialkosten Bausatz					
Raspberry Pi 3	52,96 €	44,50 €	70,58 €	83,99 €	26,08 €
RC-Akku	20,11 €	16,90 €	49,99 €	59,49 €	33,09 €
Raspberry Pi-Kamer	25,59 €	21,50 €	36,99 €	44,02 €	15,49 €
Step-down-converter	9,95 €	8,36 €	10,79 €	12,84 €	2,43 €
Raspberry Pi Roboter Auto-Räder	24,63 €	20,70 €	21,99 €	26,17 €	1,29 €
Motosteuerung L298N	2,80 €	2,35 €	6,65 €	7,91 €	4,30 €
Micro SD-Karte 32 GB	6,82 €	5,73 €	7,20 €	8,57 €	1,47 €
Kabel für Roboter Auto	2,88 €	2,42 €	6,65 €	7,91 €	4,23 €
GPS NL-8002U	111,48 €	93,68 €	109,02 €	129,73 €	15,34 €
Ladegerät	3,45 €	2,90 €	6,98 €	8,31 €	4,08 €
Breakout board	13,09 €	11,00 €	18,00 €	21,42 €	10,42 €
Controllerboard Modul I298n Dollatek Motor Drive	8,33 €	7,00 €	12,00 €	14,28 €	5,95 €
16 Kanal 12 bit PWM Servotreiber	14,28	12,00 €	18,00 €	21,42 €	7,58 €
Wlan Adapter	26,18 €	22,00 €	34	40,46 €	14,28 €
otg Usb Adapter	3,57 €	3,00 €	5,00 €	5,95 €	2,38 €

Sowie einmalig anfallenden *tatsächlichen* Kosten:

Pos.	Bezeichnung	Menge (Std.)	Einzel (€)	Gesamt (€)		
1	Fachinformatiker AE	192	19,79 €	3.799,68 €		
2	Fachinformatiker AE	192	19,79 €	3.799,68 €		
3	Fachinformatiker SI	192	15,57 €	2.989,44 €		
4	Fachinformatiker SI	192	15,57 €	2.989,44 €		
5	Fachinformatiker SI	192	15,57 €	2.989,44 €		
			<b>Nettobetrag</b>	<b>16.567,68 €</b>		
			<b>zzgl. 19% MwSt.</b>	<b>3.147,86 €</b>		
			<b>Bruttobetrag</b>	<b>19.715,54 €</b>		

Und den einmalig *geplanten* Kosten:

Pos.	Bezeichnung	Menge (Std.)	Einzel(€)	Gesamt(€)		
1	Fachinformatiker AE	280	19,79 €	5.541,20 €		
2	Fachinformatiker AE	280	19,79 €	5.541,20 €		
3	Fachinformatiker SI	280	15,57 €	4.359,60 €		
4	Fachinformatiker SI	280	15,57 €	4.359,60 €		
5	Fachinformatiker SI	280	15,57 €	4.359,60 €		
6	Systemelektroniker	40	13,67 €	546,80 €		
			<b>Nettobetrag</b>	<b>24.708,00 €</b>		
			<b>zzgl. 19% MwSt.</b>	<b>4.694,52 €</b>		
			<b>Bruttobetrag</b>	<b>29.402,52 €</b>		

## Pflichtenheft

### Zielbestimmung

Das Ziel dieses Projekts ist die Konstruktion und Realisierung eines autonom fahrenden Modellautos, das zur Eröffnung der Spielzeugmesse Marktreife erlangt hat. Der Abgabetermin für das erfüllte Lastenheft wird auf den 8. September 2023 festgelegt, an dem die Projektabnahme vor dem Vorstand der HoMa AG stattfinden wird.

### Projektaufgabe

Konstruktion und Realisierung eines autonom fahrenden Modellautos unter Verwendung der zur Verfügung gestellten Mittel.

### Hardwareanforderung

#### 1. Sensoren

Integration von Sensoren, um die Umgebung des Fahrzeugs präzise zu erfassen.

#### 2. Aktuatoren

Verwendung von Aktuatoren zur Umsetzung der Steuerbefehle und Bewegungen des Fahrzeugs.

#### 3. Microcontroller

Implementierung eines leistungsstarken Mikrocontrollers, der die Verarbeitung der Sensordaten und die Steuerung der Entscheidungsfindung ermöglicht.

#### 4. Batterieversorgung



---

Ausstattung des Fahrzeugs mit einer langlebigen Batterie, um die autonomen Funktionen mit ausreichend Energie zu versorgen.

### **Softwareanforderung**

#### **1. Autonomes Fahrsystem**

Entwicklung eines fortschrittlichen autonomen Fahrsystems, dass das Fahrzeug eigenständig navigieren und Hindernisse sicher umfahren lässt.

#### **2. Entscheidungsfindungsalgorithmen**

Implementierung intelligenter Algorithmen zur Hindernisvermeidung und Berechnung alternativer Routen.

#### **3. Programmierung der Route**

Programmierung des Fahrzeugs, um vordefinierte Strecken autonom abfahren zu können.

### **Branding & Vermarktung**

#### **1. Entwurf eines Logos**

Gestaltung eines geeigneten Logos für das Modellauto, das als Grundlage für die neue Produktpalette dienen wird.

#### **2. Gestaltung eines Flyers**

Entwurf eines aussagekräftigen Flyers für die Messe, um das Modellauto zu präsentieren.

#### **3. Webseite**

Gestaltung einer benutzerfreundlichen Webseite ohne technische Vorgaben, um das Modellauto online zu präsentieren.

#### **4. Abschlusspräsentation**

Vorbereitung einer Abschlusspräsentation vor dem Vorstand der HoMa AG:

##### **4.1 Kommunikativer Teil**

Jeder Projektteilnehmer stellt seinen Teil vor und erläutert diesen (z.B. mit Einsatz von MS PowerPoint).

##### **4.2 Praktischer Teil**

Das Fahrzeug muss einen im Vorwege festgelegten Parcours absolvieren.

### **Kosten- Zeit- & Qualitätsmanagement**

#### **1. Kostenplan**

Erstellung eines Kostenplans mit Überschlagsrechnung der Kosten pro Gerät in einer Massenfertigung.

#### **2. Verkaufspreis**

---

Vorschlag eines Verkaufspreises basierend auf Vergleich und Marktanalyse.

### **3. Zeitplan**

Ausarbeitung eines detaillierten Zeitplans mit Arbeitsabläufen und Meilensteinen.

### **4. Qualitätssicherungsplan**

Erstellung eines Qualitätssicherungsplans mit standardisierten Testverfahren.

## **Projektdokumentation**

### **1. Aktueller Projektstatus**

Sicherstellung, dass der aktuelle Projektstatus jederzeit abrufbar ist.

### **2. Abschlussdokumentation**

Vollständige Dokumentation des Projekts zum Projektabschluss, Teil der Beurteilung bzw. Bewertung des Ausschreibungsverfahrens.

## **Qualitätssicherungsplan**

### **Qualitätssicherungsziel**

Das Ziel der Qualitätssicherung für das autonom fahrende Auto ist es, sicherzustellen, dass das Fahrzeug höchste Leistungsstandards in Bezug auf Sicherheit, Zuverlässigkeit und Funktionalität erfüllt. Wir streben an, dass das autonome Miniaturauto fehlerfrei und konsistent in der Umgebungserkennung, Entscheidungsfindung und Steuerung arbeitet.

Die Qualitätssicherungsmaßnahmen sollen sicherstellen, dass das Auto Hindernisse zuverlässig erkennt, ihnen ausweicht und das vordefinierte Ziel sicher erreicht, während es die beste Route wählt. Durch die Umsetzung strenger Tests, kontinuierlicher Überprüfungen und Optimierungen wollen wir ein qualitativ hochwertiges Produkt gewährleisten, das den Anforderungen unserer Kunden entspricht und gleichzeitig höchsten Sicherheitsstandards gerecht wird. Unser Ziel ist es, das Vertrauen der Benutzer in die Autonomie des Miniaturautos zu stärken und eine herausragende Benutzererfahrung zu bieten.

## **Verantwortlichkeiten**

### **1. Projektleitung:**

- a. Verantwortlich für die Gesamtaufsicht und Koordination der Qualitätssicherungsmaßnahmen des Projekts.
- b. Überwachung des Fortschritts und der Einhaltung der Zeitpläne für die Qualitätssicherungsaktivitäten.
- c. Sicherstellung der angemessenen Ressourcenzuweisung für die Qualitätssicherung.

### **2. Entwicklungs- und Testteam:**

- a. Entwicklung der Testpläne, Testfälle und Testskripte für die Qualitätssicherung.
- b. Durchführung von Tests zur Überprüfung der Funktionalitäten, Leistung und Sicherung des autonomen Miniaturautos.
- c. Identifizierung, Dokumentation und Meldung von Software- und Hardwarefehlern.

---

### **3. Softwareingenieur:**

- a. Programmierung der Softwarealgorithmen für die Umgebungserkennung, Entscheidungsfindung und Steuerung des autonomen Autos.
- b. Durchführung von Code-Reviews, um sicherzustellen, dass die Software den besten Praktiken Qualitätsstandards entspricht.

### **4. Hardwareingenieur:**

- a. Auswahl und Integration der richtigen Sensoren, Motoren und anderen Hardwarekomponenten für das Miniaturauto.
- b. Sicherstellung der ordnungsgemäßen Verbindung und Funktion der Hardwarekomponenten.

### **5. Qualitätsmanager:**

- a. Erstellung und Überwachung des Qualitätssicherungsplans.
- b. Durchführung von Qualitätsprüfungen, Audits und Bewertungen, um die Einhaltung der Qualitätsstandards sicherzustellen.

### **6. Benutzererfahrungs- und Interaktionsdesigner:**

- a. Sicherstellung einer benutzerfreundlichen und intuitiven Benutzeroberfläche für das autonome Miniaturauto.
- b. Durchführung von Benutzertests, um das Fahrerlebnis zu verbessern und Benutzerfeedback zu sammeln.

### **7. Dokumentationsbeauftragter:**

- a. Erstellung und Pflege von umfassender Dokumentation für das Projekt.

## **Überprüfungs- und Bewertungskriterien**

### **1. Funktionalität:**

- a. Überprüfung der Umgebungserkennungsfähigkeiten, um sicherzustellen, dass das Auto Hindernisse zuverlässig erkennt und ihnen ausweicht.
- b. Bewertung der Entscheidungsfindungsalgorithmen, um sicherzustellen, dass das Auto die bestmögliche Route autonom wählt.
- c. Überprüfung der Steuerfähigkeiten, um eine reibungslose und präzise Bewegung des Miniaturautos zu gewährleisten.

### **2. Leistung:**

- a. Bewertung der Geschwindigkeit und Reaktionszeit des autonomen Fahrzeugs bei der Umgebungserkennung und Entscheidungsfindung.
- b. Überprüfung der Akkulaufzeit und der Energieeffizienz des autonomen Miniaturautos.

### **3. Benutzererfahrung:**

- a. Durchführung von Benutzertests, um die Benutzerfreundlichkeit der Benutzeroberfläche zu bewerten.

- 
- b. Sammeln von Benutzerfeedback, um mögliche Verbesserungen im Fahrerlebnis zu identifizieren.

**4. Stabilität und Robustheit:**

- a. Durchführung von Stresstests, um die Stabilität des autonomen Miniaturautos unter verschiedenen Bedingungen zu überprüfen.
- b. Bewertung der Robustheit der Software gegenüber unvorhergesehenen Situationen und Sensorstörungen.

**5. Fehlerberichte und Fehlerbehebung:**

- a. Bewertung der Anzahl und Art der gefundenen Fehler und Probleme.
- b. Überprüfung der Wirksamkeit bei der Behebung der gemeldeten Fehler.

**6. Dokumentation:**

- a. Überprüfung der Vollständigkeit und Genauigkeit der Projektdokumentation.

**Prozesse und Verfahren**

**1. Testplanung und –vorbereitung:**

- a. Erstellung eines umfassenden Testplans, der alle Aspekte der Qualitätssicherung abdeckt, einschließlich Funktionalität, Leistung Sicherheit und Benutzererfahrung.
- b. Festlegung der Testziele, -umfang und -kriterien, um sicherzustellen, dass alle relevanten Aspekte des autonomen Miniaturautos getestet werden.
- c. Identifizierung und Auswahl geeigneter Testmethoden, Werkzeuge und Ressourcen für die Durchführung der Tests.

**2. Umsetzung des Tests:**

- a. Durchführung von manuellen und automatisierten Tests, um die Funktionalität und Leistung der Autos zu überprüfen.
- b. Simulation verschiedener Szenarien und Umgebungen, um die Robustheit und Stabilität des Fahrzeugs zu testen.
- c. Erfassung und Dokumentation der Testergebnisse, einschließlich der Fehlerberichten und den Verbesserungsvorschlägen.

**3. Fehlerbehebung und Optimierung:**

- a. Analyse der gefundenen Fehler und Probleme, um die zugrundeliegenden Ursachen zu identifizieren.
- b. Durchführung von Fehlerbehebungsmaßnahmen, um die entdeckten Probleme zu beheben.
- c. Überprüfung der Wirksamkeit der Fehlerbehebung und erneute Durchführung von Tests, um sicherzustellen, dass die Probleme behoben wurden.

**4. Code-Review und Qualitätssicherung von Software und Hardware:**

- a. Regelmäßige Code-Reviews, um sicherzustellen, dass die Software den Qualitätsstandards und Best Practices entspricht.

- 
- b. Überprüfung der Hardwarekomponenten und –integration, um sicherzustellen, dass ordnungsgemäß funktionieren und miteinander kompatibel sind.

#### **5. Qualitätssicherungsberichte und –reviews:**

- a. Regelmäßige Berichterstattung über den Fortschritt der Qualitätssicherungsaktivitäten und die Testergebnisse an das Projektmanagement und die beteiligten Teams.
- b. Durchführung von Qualitätsreviews, um sicherzustellen, dass die Qualitätsziele erreicht werden und etwaige Probleme angegangen werden.

### **Zeitplanentwurf für den Zeitraum vom 24. Juli 2023 bis zum 08. September 2023**

- Phase 1: Projektvorbereitung und Testplanung
  - 24. Juli – 09. August
    - Erstellung des Qualitätssicherungsplans und Festlegung der Testziele und –kriterien.
    - Identifizierung der benötigten Testressourcen und –werkzeuge.
    - Planung der Testphasen und Zeitpläne für die Qualitätssicherung.
- Phase 2: Hardware- und Softwareintegrationstests
  - 09. August – 01. September
    - Durchführung von Integrationstests, um sicherzustellen, dass die Hardware und Software korrekt miteinander funktionieren.
    - Überprüfung der ordnungsgemäßen Verbindung der Sensoren, Motoren und anderen Hardwarekomponenten.
    - Durchführung von Funktionstests, um die Grundfunktionen des autonom fahrenden Miniaturautos zu überprüfen.
- Phase 3: Funktions- und Leistungstests
  - 04. September – 07. September
    - Durchführung umfassender Funktions- und Leistungstests, um die Qualität und Zuverlässigkeit des autonomen Autos sicherzustellen.
    - Bewertung der Umgebungserkennungsfähigkeiten, der Entscheidungsfindungsalgorithmen und der Steuerung.
    - Überprüfung der Geschwindigkeit und Reaktionszeit des Fahrzeugs.
- Phase 4: Präsentation des Prototyps
  - 08. September
    - Präsentieren des Projektes „NavDriveX“.

### **Ressourcen**

#### **1. Personalressourcen:**

- a. Qualitätssicherungsmanager: Verantwortlich für die Planung, Überwachung und Koordination der gesamten Qualitätssicherung.

- 
- b. Test- und Entwicklungsingenieure: Zuständig für die Durchführung der Tests und die Behebung von Fehlern.
  - c. Softwareingenieure: Verantwortlich für die Überprüfung und Optimierung der Softwarealgorithmen.
  - d. Hardwareingenieure: Zuständig für die Auswahl, Integration und Überprüfung der Hardwarekomponenten.
  - e. Dokumentationsbeauftragter: Erstellung und Pflege der Projektdokumentation.

## **2. Test- und Entwicklungsumgebung:**

- a. Testumgebung für die Durchführung der funktionalen, Leistungs- und Sicherheitstests.
- b. Entwicklungsumgebung für die Softwareentwicklung und -optimierung.
- c. Simulationsumgebung für die Tests unter verschiedenen Szenarien und Umgebungen.

## **3. Testwerkzeuge und Software:**

- a. Automatisierte Testwerkzeuge für die effiziente Durchführung von Tests.
- b. Debugging-Tools zur Identifizierung und Behebung von Software- und Hardwarefehlern.
- c. Softwareentwicklungsumgebungen und Simulationssoftware.

## **4. Hardwarekomponenten:**

- a. Mikrocontroller oder Einplatinencomputer für die Steuerung des autonom fahrenden Autos.
- b. Sensoren wie Kamerasensoren für die Umgebungserkennung.
- c. Motoren und Räder für die Bewegung des Autos.
- d. Stromversorgung und Batterien für den Betrieb des autonomen Fahrzeugs.

## **5. Testfahrzeug und Testumgebung:**

- a. Ein Prototyp mit allen eingebauten Hardwarekomponenten und Softwarealgorithmen.
- b. Eine sichere Testumgebung, in der die Tests durchgeführt werden können, ohne Gefahr für Menschen oder andere Objekte.

## **6. Budget:**

- a. Finanzielle Mittel für den Kauf von Hardwarekomponenten, Testwerkzeugen und anderen benötigten Ressourcen.
- b. Budget für die Schulung und Weiterbildung des Entwicklungsteams, um sicherzustellen, dass sie über das erforderliche Fachwissen verfügen.

# **Risikomanagement**

## **1. Risikoerkennung:**

- a. Identifikation potenzieller Risiken, die sich auf die Qualitätssicherung auswirken könnten. Dazu gehören technische, organisatorische, zeitliche und ressourcenbezogene Risiken.

---

## **2. Risikobewertung:**

- a. Bewertung der Auswirkungen jedes identifizierten Risikos auf das Projekt und die Qualitätssicherung.
- b. Einschätzung der Wahrscheinlichkeit, dass das Risiko eintritt.

## **3. Risikominderung:**

- a. Entwicklung von Maßnahmen und Strategien zur Minimierung oder Beseitigung der identifizierten Risiken.
- b. Priorisierung der Risiken basierend auf ihrer Bedeutung und Wahrscheinlichkeit.

## **4. Risikovermeidung:**

- a. Implementierung von Strategien, um bestimmte Risiken zu vermeiden, wenn möglich.
- b. Identifizierung und Umgehung potenziell problematischer Komponenten oder Prozesse.

## **5. Risikoüberwachung:**

- a. Regelmäßige Überwachung der identifizierten Risiken während des gesamten Projekts.
- b. Früherkennung von Anzeichen für das Eintreten von Risiken.

## **6. Krisenmanagement:**

- a. Entwicklung eines Krisenmanagementplans, der beschreibt, wie mit unvorhergesehenen Risiken und Problemen umgegangen wird.
- b. Festlegung von Eskalationsverfahren für kritische Risiken.

## **7. Kommunikation und Dokumentation:**

- a. Klare Kommunikation aller identifizierten Risiken und der entsprechenden Maßnahmen an das Projektteam und das Management.
- b. Dokumentation aller Risiken, Bewertungen und getroffenen Maßnahmen für spätere Referenz und Lernzwecke.

## **8. Kontinuierliche Verbesserung:**

- a. Analyse von Erfahrungen und Ergebnissen aus früheren Projekten, um Verbesserungen für zukünftige Qualitätssicherungsaktivitäten abzuleiten.
- b. Integration von gewonnenen Erkenntnissen in den Qualitätssicherungsprozess, um das Risikomanagement kontinuierlich zu optimieren.

## **Kommunikation**

### **1. Kommunikationsziele:**

- a. Festlegung der Hauptziele der Kommunikation im Qualitätssicherungsprozess, wie beispielsweise den Informationsaustausch, die Berichterstattung über den Fortschritt und die Koordination zwischen den Teammitgliedern.

### **2. Kommunikationspläne:**

- 
- a. Entwicklung eines detaillierten Kommunikationsplans, der die Kommunikationsziele, die Zielgruppen, die Kommunikationskanäle und die geplanten Zeitpunkte für die Kommunikation festlegt.
  - b. Festlegung der Kommunikationswege für die interne Kommunikation innerhalb des Qualitätssicherungsteams und die externe Kommunikation mit anderen Teams und Stakeholdern.
- 3. Regelmäßige Meetings:**
- a. Festlegung von regelmäßigen Teammeetings, um den aktuellen Stand der Qualitätssicherungsaktivitäten zu besprechen und mögliche Probleme zu identifizieren.
  - b. Planung von Besprechungen mit dem Projektmanagement, um den Fortschritt zu präsentieren und Fragen oder Anliegen zu klären.
- 4. Statusberichte:**
- a. Erstellung von regelmäßigen Statusberichten, um den aktuellen Stand der Qualitätssicherungsaktivitäten und den Fortschritt gegenüber den Zielen darzustellen.
  - b. Bereitstellung von klaren und präzisen Informationen, um die HoMa AG über den Fortschritt und etwaige Herausforderungen auf dem Laufenden zu halten.
- 5. Eskalationsverfahren:**
- a. Festlegung von Eskalationsverfahren für kritische Probleme oder Risiken, die das Projekt gefährden könnten.
  - b. Klare Definition von Verantwortlichkeiten und Zuständigkeiten für die Eskalation von Problemen an die entsprechenden Instanzen.
- 6. Dokumentation:**
- a. Sorgfältige Dokumentation aller Kommunikation, Besprechungen, Beschlüsse und getroffenen Maßnahmen im Zusammenhang mit der Qualitätssicherung.
  - b. Sicherstellung, dass alle relevanten Informationen für alle Teammitglieder zugänglich sind, um eine transparente Kommunikation zu gewährleisten.
- 7. Offene Kommunikationskultur:**
- a. Förderung einer offenen und transparenten Kommunikationskultur, in der Teammitglieder ermutigt werden, ihre Meinungen, Bedenken und Vorschläge zu teilen.
  - b. Schaffen eines Umfelds, in dem Feedback willkommen ist und konstruktiv genutzt wird, um die Qualitätssicherung zu verbessern.

## **Abweichungs- und Korrekturmaßnahmen**

- 1. Abweichungserkennung:**
- a. Implementierung von Mechanismen zur systematischen Überwachung der Qualitätssicherungsaktivitäten.



- 
- b. Frühzeitige Erkennung von Abweichungen von den geplanten Zielen, Zeitplänen oder Qualitätsstandards.

**2. Abweichungsanalyse:**

- a. Untersuchung der Ursachen für festgestellte Abweichungen, um deren Auswirkungen und potenzielle Risiken zu verstehen.
- b. Analyse der zugrundeliegenden Faktoren, um mögliche Schwachstellen oder Engpässe zu identifizieren.

**3. Festlegung von Korrekturmaßnahmen:**

- a. Entwicklung von geeigneten Maßnahmen, um die festgestellten Abweichungen zu korrigieren und mögliche Probleme zu beheben.
- b. Priorisierung der Korrekturmaßnahmen basierend auf ihrer Dringlichkeit und Auswirkungen.

**4. Verantwortlichkeiten und Zuständigkeiten:**

- a. Klare Zuweisung von Verantwortlichkeiten für die Durchführung der Korrekturmaßnahmen.
- b. Festlegung von Zuständigkeiten, um sicherzustellen, dass die Maßnahmen angemessen und rechtzeitig umgesetzt werden.

**5. Zeitplan für Korrekturmaßnahmen:**

- a. Festlegung eines realistischen Zeitrahmens für die Umsetzung der Korrekturmaßnahmen.
- b. Sicherstellung, dass die Maßnahmen in angemessener Zeit abgeschlossen werden können, um die Qualitätssicherung nicht zu verzögern.

**6. Überwachung und Bewertung der Korrekturmaßnahmen:**

- a. Regelmäßige Überprüfung der Wirksamkeit der durchgeführten Korrekturmaßnahmen.
- b. Bewertung, ob die Abweichungen behoben wurden und die angestrebten Qualitätsziele erreicht wurden.

**7. Dokumentation:**

- a. Dokumentation aller Abweichungen, Korrekturmaßnahmen und Ergebnisse für zukünftige Referenz und Lernzwecke.
- b. Sicherstellung, dass alle relevanten Informationen für das Qualitätssicherungsteam und das Projektmanagement zugänglich sind.

**8. Lernprozess:**

- a. Integration von Erkenntnissen aus durchgeführten Korrekturmaßnahmen in den Qualitätssicherungsprozess, um ähnliche Abweichungen in der Zukunft zu verhindern.
- b. Etablierung eines Lernprozesses, um aus Erfahrungen zu lernen und kontinuierlich zu verbessern.

## **Überprüfung und Aktualisierung**

**1. Regelmäßige Überprüfung der Qualitätssicherungsprozesse:**

- a. Festlegung eines Zeitplans für regelmäßige Überprüfungen der Qualitätssicherungsprozesse, um sicherzustellen, dass sie effektiv und effizient sind.

- 
- b. Analyse der Ergebnisse vergangener Qualitätssicherungsaktivitäten, um mögliche Verbesserungsbereiche zu identifizieren.

**2. Kontinuierliche Verbesserung:**

- a. Integration von Erkenntnissen und Erfahrungen aus früheren Projekten in den Qualitätssicherungsprozess, um kontinuierlich zu lernen und zu verbessern.
- b. Identifizierung von bewährten Praktiken und Lösungen, die in zukünftigen Projekten angewendet werden können.

**3. Aktualisierung von Testplänen und -verfahren:**

- a. Überprüfung und Aktualisierung der Testpläne und -verfahren, um sicherzustellen, dass sie den aktuellen Anforderungen und Best Practices entsprechen.
- b. Einführung neuer Testmethoden und Werkzeuge, die die Effizienz und Genauigkeit der Qualitätssicherung verbessern.

**4. Schulung und Weiterbildung:**

- a. Schulung des Qualitätssicherungsteams und anderer beteiligte Personen, um sicherzustellen, dass sie über das erforderliche Fachwissen und die Kompetenzen verfügen.
- b. Bereitstellung von Weiterbildungsmaßnahmen, um das Team über aktuelle Entwicklungen und Trends in der Qualitätssicherung auf dem Laufenden zu halten.

**5. Feedback und Benutzererfahrung:**

- a. Einholung von Feedback von Teammitgliedern und Stakeholdern zur Qualitätssicherung und zum gesamten Entwicklungsprozess.
- b. Berücksichtigung von Benutzererfahrungen und -feedback bei der Aktualisierung von Prozessen und Verfahren.

**6. Risikomanagement und Krisenbewältigung:**

- a. Überprüfung des Risikomanagementplans und gegebenenfalls Aktualisierung, um potenzielle neue Risiken zu berücksichtigen.
- b. Weiterentwicklung des Krisenbewältigungsplans, um auf zukünftige Herausforderungen besser vorbereitet zu sein.

**7. Stakeholder-Kommunikation:**

- a. Regelmäßige Kommunikation mit der HoMA AG über den Fortschritt der Qualitätssicherung und mögliche Veränderungen im Prozess.
- b. Offene Diskussion über geplante Aktualisierungen und ihre potenziellen Auswirkungen.

## Projektzeitplan

Projektzeitplan			Zeit in Stunden	
<b>1. Definitionsphase</b>			<b>3</b>	
1.1 Kundengespräch			1	
1.2 Genehmigung und Freigabe			0,5	
1.3 Auswahl des Projektteams			1	
1.4 Auswahl des Hardware- und Softwareteams			0,5	
<b>2. Planungsphase</b>			<b>19</b>	
2.1 Zeitplan erstellen			1	
2.2 Ist-Analyse			1	
2.3 Soll-Konzept			1	
2.4 Projektstrukturplan erstellen			1	
2.5 Kostenplan erstellen			1	
2.6 Qualitätsmanagement			14	
<b>3. Durchführungsphase</b>			<b>126</b>	
3.1 Unterlagen vorbereiten			4	
3.2 Hardwarekomponenten zusammensetzen			16	
3.3 Software schreiben			14	
3.4 Software in Hardware implementieren			16	
3.5 Konfiguration Modell			16	
3.6 Testlauf			20	
3.7 Fehlerbehebung			20	
3.8 Webanwendung			20	
<b>4. Abschlussphase</b>			<b>44</b>	
4.1 Auswertung				
4.1.1 Analyse und Zusammenfassung der Ergebnisse			8	
4.1.2 Dokumentation			10	
4.2 Abschlussmeeting				
4.2.1 Abschlussbericht			10	
4.2.2 Vorbereitung der Präsentation			16	
		<b>Gesamtzeit:</b>	<b>192</b>	

## Ressourcenplan

### Ressourcenplan für Nexus Robotics - Projekt NavDriveX

#### Ziele:

Das Ziel dieses Projekts ist es, ein voll funktionsfähiges autonomes Miniaturauto unter Verwendung der bereitgestellten Materialien der HoMa AG innerhalb des genannten Zeitraums herzustellen.

## Ressourcentypen

### 1. Mitarbeiter:

- Frau A. Bensch (Projektleitung / Fachinformatikerin für Anwendungsentwicklung)

- 
- Herr C. Cuhaci (Stellvertretende Projektleitung / Fachinformatiker für Systemintegration)
  - Herr K. Driesener (Fachinformatiker für Anwendungsentwicklung)
  - Herr S. Warner (Fachinformatiker für Systemintegration)
  - Herr A. Dahir (Fachinformatiker für Systemintegration)

## **2. Materialien:**

- Buch: Roboter-Autos mit dem Raspberry PI
- Raspberry Pi 3
- Micro SD-Karte
- Platten mit Motoren
- M2 Abstandhalter Set
- M3 Abstandhalter Set
- Kabel Set
- I²C-Kabel Set
- RC-Akku
- N8 NiMH Ladegerät
- Externes Netzteil
- ANGEER PCA9685
- I²C Hub
- Sense-HAT
- DC-Spannungsregler Set
- DollaTek Mini Stepper
- DollaTek 1.8“ LCD
- FREENOVE Breakout Board
- VL53L1X Sensor (ToF)
- Sense HAT X001GKVR09
- Kamera-Modul V2
- ESD-Handschuhe
- Schnittsichere Handschuhe
- Edelstahl L Lineal
- KEMOT NAR0150
- Mysoler
- Heißklebepistole
- Heißklebesticks
- OTG USB Adapter
- USB WLAN Adapter

## **Ressourcenbedarf**

### **1. Mitarbeiter:**

Projektleitung und Koordination: Frau A. Bensch, Herr C. Cuhaci

Technische Umsetzung:

- Frau A. Bensch
- Herr C. Cuhaci,
- Herr K. Driesener,
- Herr S. Warner,

- 
- Herr A. Dahir

## **2. Materialien:**

Materialien gemäß Projektplan und technischer Anforderungen.

## **Ressourcenverfügbarkeit**

### **1. Mitarbeiter:**

Alle oben genannten Mitarbeiter sind für das Projekt verfügbar.

### **2. Materialien:**

Alle aufgeführten Materialien sind verfügbar.

## **Ressourcenallokation:**

### **1. Mitarbeiter:**

- Frau A. Bensch: Projektleitung, Projektmanagement, Programmierung
- Herr C. Cuhaci: Technische Leitung, Systemintegration
- Herr K. Driesener: Webentwicklung und Programmierung
- Herr S. Warner: Systemintegration und Testing
- Herr A. Dahir: Systemintegration, Marktanalyse

### **2. Materialien:**

Materialien werden gemäß dem Projektzeitplan und den Anforderungen verwendet.

## **Ressourcenüberwachung und -kontrolle**

### **1. Mitarbeiter:**

Tägliche oder wöchentliche Fortschrittsberichte und Besprechungen.

### **2. Materialien:**

Regelmäßige Überprüfung des Materialverbrauchs und der Bestände.

## **Risikobewertung und -management**

### **Identifizierte Risiken:**

- Verletzung durch scharfe Gegenstände → Schutzhandschuhe tragen.
- Verbrennungsgefahr durch Heißklebepistolen oder Lötkolben → Schutzhandschuhe tragen.
- Kurzschlussgefahr → Auf richtige Verkabelung achten.

### **Meilensteine:**

1. Projektstart: 24. Juli 2023
2. Planung: 26. Juli 2023
3. Durchführungsphase: 28. August 2023
4. Abschluss: 07. September 2023

---

## Dokumentationen

### Dokumentation Website

#### 1. Übersicht der Website

Der Webauftritt ist einfach aufgebaut und besteht aus einem Menü mit sechs Punkten:

- a. Home -> Startseite mit einer kurzen Vorstellung und was wir machen.
- b. Produkte -> Auf dieser Seite werden später die zur Verfügung stehenden Produkte angezeigt.
- c. Stream -> Im Stream wird der Livestream aus dem Webinterface übertragen.
- d. Downloads -> Hier stehen Codebeispiele und Anleitungen zum Download bereit.
- e. About us -> Vorstellung des Teams, deren Fachrichtung und den getätigten Aufgaben.
- f. Dokumentation -> Diese Seite beinhaltet die Dokumentationen der einzelnen Aufgaben.

#### 2. Installation

Die Installation des Webauftrittes benötigt einen Webserver aber keinerlei Software.

#### 3. Technologien

- a. Sprache: HTML 5
- b. Framework: Bootstrap 5.3
- c. Icons: Bootstrap, hiclipart

#### 4. Design und Layout

Design und Layout wurden schlicht und einfach gewählt.  
Änderungen sind möglich.

#### 5. Navigation

Die Navigation auf der Homepage wird über ein Menü auf der linken Seite ermöglicht. Menüpunkte haben keine Untermenüs und der Seiteninhalt kann gleich eingesehen werden.

#### 6. Bekannte Probleme und Einschränkungen

Zurzeit sind keine Probleme bekannt.

#### 7. Wartung und Aktualisierung

Der Webauftritt wird in regelmäßigen Abständen auf Fehler überprüft und gewartet. Bei Änderungen im Inhalt wird dieser nach eingehenden Tests und Codeversionen auf einem Testserver implementiert und auf eventuelle Fehler geprüft. Nach der Kompatibilitätsprüfung wird der Inhalt auf dem Liveserver implementiert und der Webauftritt aktualisiert.

#### 8. Kontakt

Bei Fragen oder Problemen erreichen Sie den Support von Nexus Robotics unter der E-Mail [support@nexus-robotics.com](mailto:support@nexus-robotics.com) und bei schwerwiegenden Zwischenfällen unter der Telefonnummer 0228 633 633.

#### 9. Beispielanwendungen

Der Webauftritt kann zur Informationsgewinnung der Produkte genutzt werden. Des Weiteren wird ein Livestream angeboten, welcher zu verschiedenen Tests die Prototypen eingeschaltet wird, um die Fahrt zu verfolgen.

#### 10. Versionskontrolle

Eine Versionskontrolle wird durch GitHub unter der Adresse <https://github.com/KayDriesener/Schulprojekt> bereitgestellt. Unter dieser Adresse können sich alle Entwickler die aktuelle Version herunterladen und Anpassungen vornehmen. Nach einer Revision der Änderungen kann der geänderte Code implementiert werden oder abgelehnt.

GitHub ermöglicht es mehreren Entwicklern, gleichzeitig an einem Projekt zu arbeiten um dieses in kleinere Aufgabenteile herunter zu brechen.

## Dokumentation Software

### 1. Einleitung

Diese Dokumentation beschreibt die Schritte zur Installation der Software und Konfiguration für das Roboter-Auto mit einem Raspberry Pi.

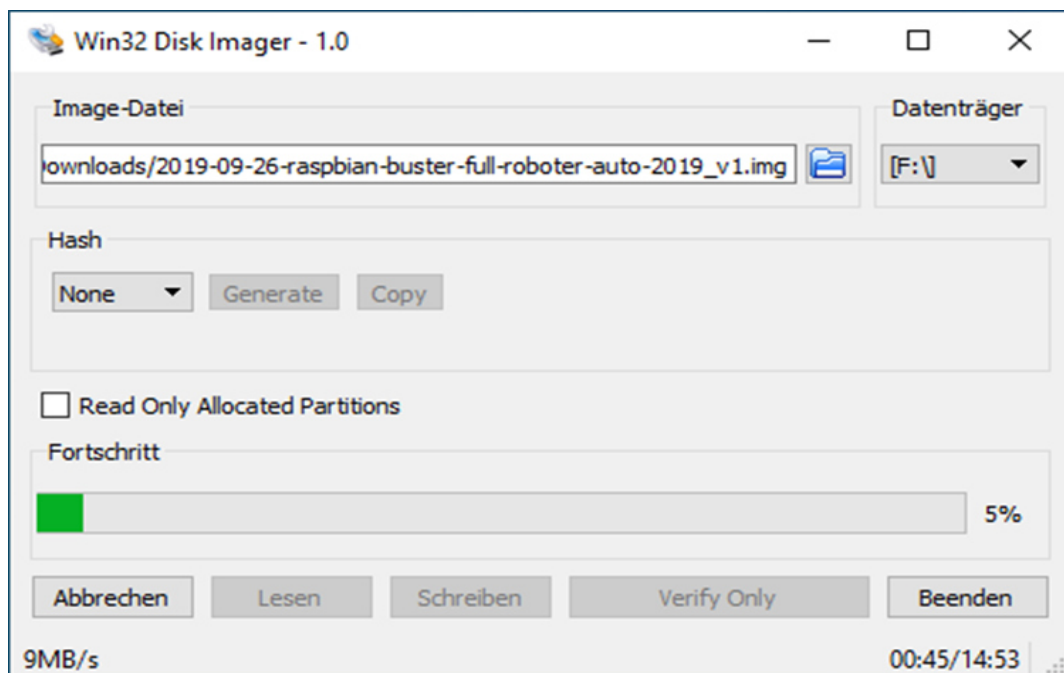
### 2. Voraussetzungen

- a. Raspberry Pi (Modell 3)
- b. SD-Karte mit ausreichender Kapazität (mindestens 32 GB)
- c. Kartenlesegerät
- d. HDMI-Monitor, Tastatur und Maus für die anfängliche Konfiguration
- e. Stabile Internetverbindung für den Software-Download und –Aktualisierung

### 3. Betriebssysteminstallation

#### Herunterladen der Raspbian Image Datei

Die neueste Raspbian Image Datei wird von der offiziellen Raspberry Pi-Website heruntergeladen.



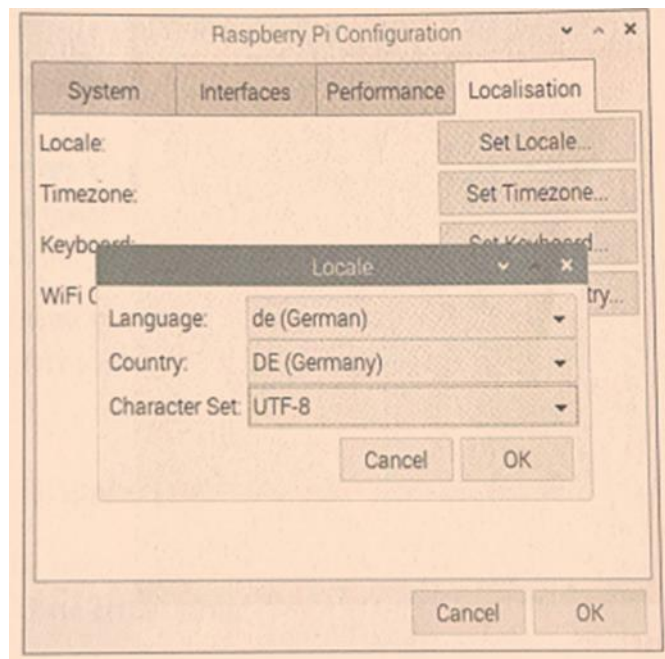
#### Schreiben des Images auf die SD-Karte

Die Raspbian Image Datei wird mithilfe des Programms "win32diskimager" auf die SD-Karte geschrieben. Die SD-Karte wird in ein Kartenlesegerät eingesteckt und den Anweisungen des Programms gefolgt.

#### 4. Erster Start und Konfiguration

##### Anschließen des Raspberry Pi

Der Raspberry Pi wird an einen HDMI-Monitor, eine Tastatur und eine Maus angeschlossen. Die vorbereitete SD-Karte wird eingesteckt.



##### Sprache und Land auswählen

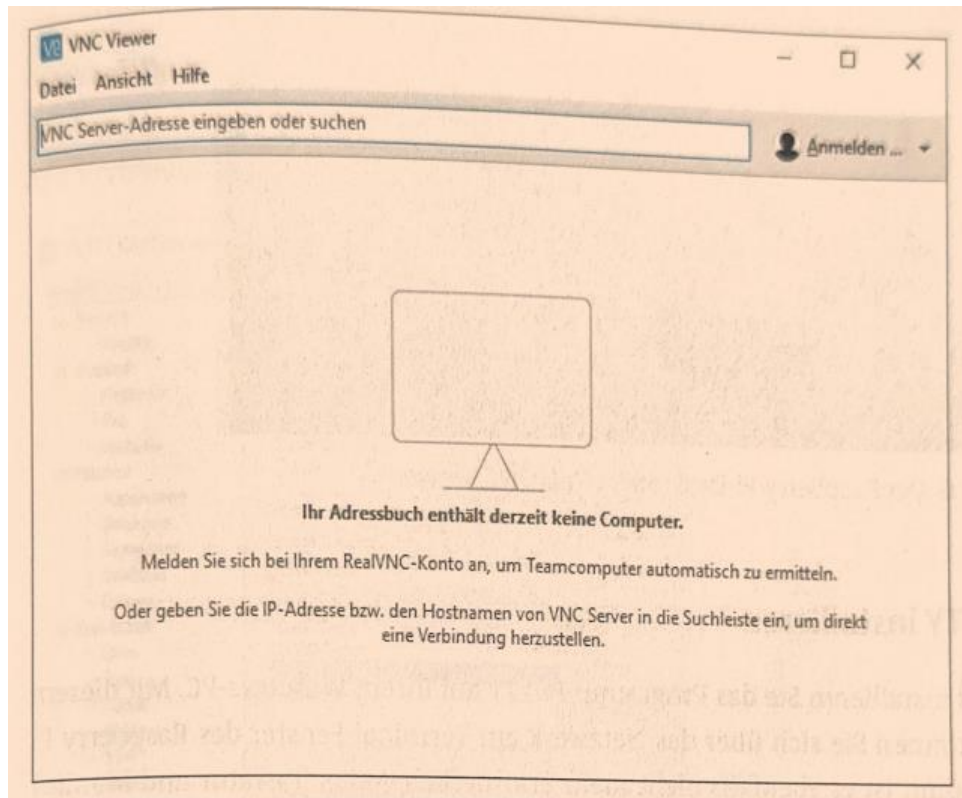
Beim ersten Start werden die gewünschte Sprache und das Land ausgewählt. Die entsprechenden Einstellungen werden gemäß den Präferenzen vorgenommen.

##### WLAN – Einrichtung

Die WLAN-Verbindung wird konfiguriert, um eine Internetverbindung herzustellen. Die Zugangsdaten für das WLAN-Netzwerk werden eingegeben.

#### 5. Installation und Konfiguration von Software





### **Installation von Zusatzprogrammen**

Die erforderlichen Programme werden installiert:

- Real VNC Server für den Remote-Zugriff
- Samba Server für die Netzwerkfreigabe
- Python als Programmbibliothek für das Roboter-Auto
- Flask für den Webserver
- MJPG-Streamer für das Live-Video-Streaming

### **Konfiguration der Programme**

Die Konfiguration jedes installierten Dienstes erfolgt gemäß den Anforderungen des Roboter-Auto-Projekts. Netzwerkfreigaben, Benutzerzugriffsrechte und IP-Einstellungen werden durchgeführt.

### **Hinzufügen der Programmcodes**

- Ein Steuerungsprogramm für den Motortreiber in Python wird erstellt.
- Ein Steuerungsprogramm für das Roboter-Auto in Python wird entwickelt.
- Ein Webinterface zur Steuerung des Roboter-Autos über den Webbrowser implementiert wird.

## **6. Aktivieren der Peripheriegeräte Kamera anschließen**

## Kameraaktivierung an Raspbian

Das Raspberry Pi Konfigurationsmenü im Raspbian-Betriebssystem wird geöffnet, um die Kamera in den Interface-Einstellungen zu aktivieren.

## 7. Konfiguration des Video – Streaming – Servers

Die Konfiguration des MJPG-Streamers für das Live-Video-Streaming von der angeschlossenen Kamera wird abgeschlossen.

## 8. Autostart der Programme konfigurieren

Der Cron-Daemon wird genutzt, um Programme zeitgesteuert auszuführen. Der MJPG-Streamer und das Steuerungsprogramm "robot-control-web.py" werden im Cron-Daemon eingetragen, damit der Video-Stream und die Webinterface-Steuerung automatisch gestartet werden.

## 9. Betriebssystem- und Firmware-Aktualisierung

Das Betriebssystem und die Raspberry Pi Firmware werden regelmäßig auf den neuesten Stand gebracht, um von den neuesten Funktionen und Sicherheitsupdates zu profitieren. Dazu werden folgende Schritte ausgeführt:

→ Das Betriebssystem wird über das Terminal mit dem Befehl `sudo apt update` gefolgt von `sudo apt upgrade` aktualisiert.

→Die Raspberry Pi Firmware wird mit dem Befehl `sudo rpi-update` aktualisiert.

```
pi@raspberrypi ~  
buster/main armhf libsmclient armhf 2:4.9.5+dfsg-5 [152 kB]  
Holen:9 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf samba-lib armhf 2:4.9.5+dfsg-5 [4.695 kB]  
Holen:10 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf libwbclient0 armhf 2:4.9.5+dfsg-5 [127 kB]  
Holen:11 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf bzip2 armhf 1.0.6-9.1 [46,1 kB]  
Holen:12 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf libbz2-1.0 armhf 1.0.6-9.1 [42,9 kB]  
Holen:13 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf nano armhf 3.2-3 [523 kB]  
Holen:14 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf bind9-host armhf 1:9.11.5.P4+dfsg-5.1 [268 kB]  
Holen:15 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf libbind9-161 armhf 1:9.11.5.P4+dfsg-5.1 [243 kB]  
Holen:16 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf libiscfgl63 armhf 1:9.11.5.P4+dfsg-5.1 [258 kB]  
Holen:17 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf libisccl61 armhf 1:9.11.5.P4+dfsg-5.1 [233 kB]  
Holen:18 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf libdns1104 armhf 1:9.11.5.P4+dfsg-5.1 [1.076 kB]  
Holen:19 http://mirror1.hs-esslingen.de/pub/Mirrors/archive.raspbian.org/raspbian  
buster/main armhf libisc1100 armhf 1:9.11.5.P4+dfsg-5.1 [421 kB]  
14% [19 libisc1100 45,1 kB/421 kB 11%] [3 libraspberrypi-doc 14,5 MB/31,4 MB 46]
```

## 10. Fazit

---

In dieser Dokumentation wurden die Schritte zur Softwareinstallation und Konfiguration für Roboter-Autos mit einem Raspberry Pi beschrieben. Mit diesen Schritten sollte das System einsatzbereit sein und die Grundlage für die Entwicklung und Programmierung des Roboter-Auto-Projekts bieten.

## Dokumentation Add-on Programm Video-Objekterkennung

### Einleitung

Diese Dokumentation beschreibt ein Python-Programm zur Objekterkennung unter Verwendung von OpenCV und einem vortrainierten SSD Mobilenet V3-Modell. Das Programm erkennt Objekte in Echtzeit auf Videodaten, sei es von einer Webcam oder einer Videodatei.

### Abhängigkeiten

Um dieses Programm auszuführen, müssen Sie die folgenden Python-Bibliotheken installieren:

- `cv2` (OpenCV): Eine umfassende Bibliothek für Computer Vision.
- `matplotlib.pyplot`: Eine Bibliothek zum Anzeigen von Bildern.
- Die Dateien `ssd\_mobilenet\_v3\_large\_coco\_2020\_01\_14.pbtxt` und `frozen\_inference\_graph.pb`, die das vortrainierte Modell enthalten.
- Eine Textdatei `Labels.txt`, die die Namen der erkannten Klassen enthält.
- Eine Videodatei (z. B. "sample.mp4") oder eine funktionierende Webcam für die Videoquelle.

### Laden der Modelldateien

Das Programm beginnt damit, die erforderlichen Modell- und Konfigurationsdateien zu laden. Die `frozen\_inference\_graph.pb`-Datei enthält das trainierte SSD Mobilenet V3-Modell, während `ssd\_mobilenet\_v3\_large\_coco\_2020\_01\_14.pbtxt` die Konfigurationsinformationen enthält.

```
# Laden der Dateien
config_file = 'ssd_mobilenet_v3_large_coco_2020_01_14.pbtxt'
frozen_model = 'frozen_inference_graph.pb'

# Modell laden
model = cv2.dnn_DetectionModel(frozen_model, config_file)
```

### Laden der Klassenbezeichnungen

Die Klassenbezeichnungen, die das Modell erkennt, werden aus der Textdatei `Labels.txt` geladen und in einer Python-Liste `classLabels` gespeichert.

```
# Leere Python-Liste
classLabels = []
# Laden der Klassenbezeichnungen
file_name = 'Labels.txt'
with open(file_name, 'rt') as fpt:
    classLabels = fpt.read().rstrip('\n').split('\n')
```

### Konfiguration des Modells

Das Modell wird auf eine Eingabegröße von 320x320 Pixeln konfiguriert, da dieses spezielle Modell diese Größe erwartet. Weitere Konfigurationen umfassen die Skalierung der Eingabewerte, die Mittelwertanpassung und die Vertauschung der Farbkanäle.

```
# Eingabegröße definieren (320x320), da unser Modell nur dieses Format unterstützt
model.setInputSize(320, 320)
model.setInputScale(1.0 / 127.5)
model.setInputMean((127.5, 127.5, 127.5))
model.setInputSwapRB(True)
```

### Videoquelle öffnen

Das Programm versucht, eine Videodatei mit dem Namen "sample.mp4" zu öffnen. Wenn dies nicht gelingt, wird die Webcam als Videoquelle verwendet.

```
cap = cv2.VideoCapture("sample.mp4")
|
if not cap.isOpened():
    cap = cv2.VideoCapture(0)
if not cap.isOpened():
    raise IOError("Kann Video nicht öffnen")
```

## Objekterkennungsschleife

In einer Endlosschleife wird kontinuierlich Video von der ausgewählten Quelle gelesen. Das Modell wird auf jedem Frame angewendet, und erkannte Objekte werden auf dem Frame markiert und beschriftet.

```
font_scale = 3
font = cv2.FONT_HERSHEY_PLAIN

while True:
    ret, frame = cap.read()
    ClassIndex, confidence, bbox = model.detect(frame, confThreshold=0.55)

    print(ClassIndex)
    if len(ClassIndex) != 0:
        for ClassInd, conf, boxes in zip(ClassIndex.flatten(), confidence.flatten(), bbox):
            if ClassInd <= 80:
                cv2.rectangle(frame, boxes, (255, 0, 0), 2)
                cv2.putText(frame, classLabels[ClassInd - 1], (boxes[0] + 10, boxes[1] + 40), font,
                            fontScale=font_scale, color=(0, 255, 0), thickness=3)
    cv2.imshow('Objekterkennungs-Tutorial', frame)
    if cv2.waitKey(2) & 0xFF == ord('q'):
        break
```

## Beendigung des Programms

Das Programm kann beendet werden, indem Sie die Taste "q" drücken. Nach Beendigung wird die Videoquelle freigegeben, und alle geöffneten Fenster werden geschlossen.

```
cap.release()
cv2.destroyAllWindows()
```

## Projekt-Dokumentation: Raspberry Pi Roboter-Auto

### Teil 1: Benötigte Materialien und Werkzeuge

Bevor wir mit dem Bau des Roboter-Autos beginnen, müssen wir sicherstellen, dass wir alle erforderlichen Materialien und Werkzeuge zur Hand haben. Hier ist eine ausführlichere Liste:

#### Materialien

- Raspberry Pi 3 (bevorzugt Raspberry Pi 3 oder neuer): Der Raspberry Pi wird das Gehirn unseres Roboter-Autos sein. Er ist ein kleiner Einplatinencomputer, der über GPIO-Pins (General Purpose Input Output) verfügt, die es uns ermöglichen, elektronische Komponenten zu steuern.
- Micro-SD-Karte (mindestens 8 GB): Die Micro-SD-Karte wird als Speichermedium für das Betriebssystem und unsere Programme im Raspberry Pi verwendet.

- 
- Roboter-Auto-Chassis-Kit (mit Rädern und Antriebsmotoren): Das Chassis-Kit bildet das Grundgerüst des Autos und enthält die Räder und Motoren, die für die Fortbewegung des Autos sorgen.
  - M2 und M3 Abstandshalter: Mit den Abstandshaltern lassen sich viele der elektronischen Komponenten des Roboter-Autos vom Raspberry Pi über den Motortreiber bis hin zur Raspberry-Pi-Kamera sicher und vor allem ohne Beschädigung am Roboter-Auto befestigen.
  - Dollatek Motor-Treiber-Board (z. B. L298N): Der Motor-Treiber ermöglicht es uns, die Motoren des Roboter-Autos zu steuern.
  - RC-Akku 7,2 V, NiMh, 5000 mAh: Der Akku liefert die Energie für die Motoren und dem Raspberry Pi. Wir verwenden einen 5000mAh Akku, um eine längere Betriebszeit zu gewährleisten.
  - Step-down Konverter-Modul: Den Step-down-Konverter benötigen wir, um den Raspberry Pi mit Strom zu versorgen. Der Konverter erzeugt aus der Versorgungsspannung der Batterie oder dem Netzteil (7V bis 24V) eine Spannung von 5Volt und bis zu 3Ampere. Wir empfehlen einen Konverter mit mehreren USB-Ausgängen, um weitere 5Volt-Komponenten wie z.B. USB Wlan-Adapter anzuschliessen.
  - N8 NiHM Ladegerät: Dieses Ladegerät ist für NiMH-Akkus mit Standard- oder Mini-Tamiya-Anschluss geeignet.
  - Netzteil für das Roboter-Auto: Der Raspberry Pi benötigt eine separate Stromquelle, um in der Testphase nicht die ganze Zeit über den Akku mit Strom versorgt zu werden.
  - Mini-Tamiya-Kabel: Wir verwenden ein Tamiya-Kabel für die Stromversorgung des Roboter-Autos. Sie eignen sich mit ihren Anschlusssteckern und Buchsen sehr gut für Verbindungen, die immer wieder gelöst werden müssen. (Netzbetrieb oder Akkubetrieb)
  - Freenove Breakout Board: Wir nutzen das Breakout Board, dass über verschiedene Anschlüsse und Steckplätze verfügt, um eine breite Palette von elektronischen Komponenten anzuschließen. Das Breakout Board ermöglicht ein einfaches Prototyping der Schaltungen. Außerdem erkennt man über die eingebauten LED's welcher GPIO-Pin an oder ausgeschaltet ist.
  - Jumper-Kabel: Wir nutzen die Jumper-Kabel, um den Motortreiber und sonstige Komponenten des Roboter-Autos mit dem Breakout Board zu verbinden. Die Kabel werden einfach auf die Pins gesteckt oder verschraubt. Das spart sehr viel Zeit und reduziert den Aufwand beim Verkabeln erheblich.



- **Angeek PCA9685 16 Kanal 12 Bit PWM Servocontroller:** Dem Servocontroller ist es mit seiner speziellen Hardware möglich, ein wesentlich exakteres PWM-Signal zu erzeugen, als es der Raspberry Pi selber kann, und das gleichzeitig für bis zu 16 Servomotoren. Daher empfehlen wir den Servocontroller zu verwenden, um eine exaktere Ansteuerung der Antriebsmotoren zu gewährleisten. Ein weiterer Vorteil des Servocontrollers ist, dass dieser über den I<sup>2</sup>C-Bus mit dem Raspberry Pi verbunden wird.
- **Seeed Studio Grove-I<sup>2</sup>C-Hub:** Für die Anbindung mehrerer I<sup>2</sup>C-Sensoren benutzen wir einen I<sup>2</sup>C-Hub. So können wir mehrere Sensoren komfortabel anschließen. (z.B. Time-of-Flight-Abstandssensoren.)
- **CQ Robot Ocean VL53L1X Sensor (ToF):** Wir haben uns für die ToF-Sensoren VL53L1X entschieden, da diese mit einem schwachen Laser ungefährlich für die Augen sind. Außerdem sind sie kostengünstiger als ein vergleichbarer I<sup>2</sup>C-Ultraschallsensor. Der Sensor verfügt über drei Modi für die Entfernungsmessung, die sich in der Reichweite und somit in der messbaren Entfernung unterscheiden.
- **I<sup>2</sup>C-Kabel Set:** Wird benötigt, um die Sensoren mit dem I<sup>2</sup>C-Hub zu verbinden.
- **Kamera-Modul V2 Noir:** Wir verwenden nicht die Original-Raspberry-Pi Kamera, sondern eine Weitwinkelkamera eines Drittanbieters z.B. von Noir wie in unserem Roboter-Auto.
- **USB-WLAN-Adapter:** Ein USB-WLAN-Adapter ist erforderlich, um den Raspberry Pi mit dem Internet zu verbinden, wenn er über keinen integrierten WLAN-Chip verfügt.

## Werkzeuge

- **Computer mit SD-Kartenleser und Internetzugang:** Sie benötigen einen Computer mit einem SD-Kartenleser, um das Betriebssystem auf die Micro-SD-Karte zu schreiben. Der Internetzugang ist erforderlich, um die benötigten Softwarepakete herunterzuladen.
- **Computer Werkzeug Set:** Für die Verschraubungen.
- **Lötkolben:** Für den Fall, das doch etwas verlötet werden muss.
- **Dritte Hand mit Klemmen:** Zur Erleichterung der Lötarbeiten.
- **Heißklebepistole und Heißklebesticks:** Zur Befestigung von Komponenten die gegebenenfalls nicht verschraubt werden können.

## Teil 2: Hardware-Zusammenbau

- 
- Beginnen wir mit dem Zusammenbau des Roboter-Auto-Chassis-Kits. Als erstes habe ich die Getriebemotoren an die dafür vorgesehenen Befestigungspunkte am Chassis verschraubt. Nachdem alle vier Motoren verschraubt sind, werden die Räder ganz einfach auf die Achsen der Getriebemotoren gesteckt.
  - Als nächstes habe ich das Motor-Treiber-Modul (L298N) an einem geeigneten Platz mit Abstandshaltern auf dem Chassis verschraubt. (ggf. müssen wie in meinem Fall noch extra Löcher gebohrt werden)
  - Danach habe ich die Getriebemotoren an das Motor-Treiber-Board (L298N) gemäß dem Schaltplan des Boards angeschlossen. Beachten Sie die Anschlüsse für die Motoren: Normalerweise M1 für die rechten und M2 die linken Motoren. Dabei ist darauf zu achten, dass jeweils die Motoren der rechten sowie auch die der linken Seite, parallel zueinander an M1 oder M2 angeschlossen werden. Bei falscher Verkabelung kann es passieren das die Räder entgegengesetzt drehen.
  - **Als weiteres Bauteil, habe ich den Servocontroller ebenfalls mit Abstandshaltern auf dem Chassis verschraubt. Hiermit ist die erste Ebene unseres Roboter-Autos erst einmal fertig, wie auf dem unteren Bild zu sehen ist.**