



Mechatronik Trinational
Mécatronique Trinationale



2020

Pflichtenheft Labirinto



Labirinto

Projekt: Labirinto

Audric Strümpfer

Adrian Rosser

Mechatronik Trinational

www.trinat.net

EINLEITUNG	3
SINN UND ZWECK DES DOKUMENTES	3
VISION (INHALT UND ZIELE)	3
DEFINITIONEN UND ABKÜRZUNGEN	3
ABLAGE, GÜLTIGKEIT UND BEZÜGE ZU ANDEREN DOKUMENTEN	3
VERTEILER UND FREIGABEN	3
KONZEPT UND RAHMENBEDINGUNGEN	4
ZIELE UND NUTZEN DES AUFTRAGGEBERS	4
ZIELE UND NUTZEN DES ANWENDERS	4
BENUTZER / ZIELGRUPPE	4
SYSTEMVORAUSSETZUNGEN	4
RESSOURCEN	4
ÜBERSICHT DER MEILENSTEINE	5
GROBSCHÄTZUNG DES AUFWANDS	6
FUNKTIONALE ANFORDERUNGEN	7
ÜBERSICHT DER ANFORDERUNGEN	7
USE-CASE ÜBERSICHT	8
USE-CASE DIAGRAMM	9
RISIKEN	9
TESTHINWEISE	10
VERGLEICH MIT BESTEHENDEN LÖSUNGEN	10
SYSTEMÜBERSICHT	11
GRUNDSÄTZLICHER AUFBAU	11
SYSTEMBESCHREIBUNG	11
SCHNITTSTELLEN	12
ÜBERSICHT	12
HARDWARESCHNITTSTELLEN	12
SOFTWARESCHNITTSTELLEN	12
MECHANIK	13
MECHANISCHE STRUKTUR	13
SENSOREN	14
ULTRASCHALL SENSOREN	14
ENCODER	15
ACCELEROMETER	15
AKTOREN	15
DC MOTOR	15
ELEKTRONIK	16
STROMVERSORGUNG	16
MOTORTREIBER	16

INFORMATIONSVERRARBEITUNG	17
BOARD-COMPUTER	17
BENUTZERINTERFACE	17
ANHANG / RESSOURCEN	18
IMPRESSUM	18
QUELLENANGABEN	18
LITERATURVERZEICHNIS	18

ABILDUNGSVERZEICHNIS

Abbildung 1: Logo Labirinto.....	3
Abbildung 2: Use-Case Diagramm	9
Abbildung 3: Professioneller Labyrinth-Löse-Roboter (Wikipedia, 2020)	10
Abbildung 4: DIY-Labyrinth-Löse-Roboter (Research Gate, 2020)	10
Abbildung 5: Blockschaltbild	11
Abbildung 6: Übersicht Schnittstellen	12
Abbildung 7: Entwurfsskizze Labirinto	13
Abbildung 8: Mechanischer Aufbau Labirinto	13
Abbildung 9: Aufbau Plattform A	14
Abbildung 10: Aufbau Plattform B	14
Abbildung 11: Ultrasonic Sensor HC-SR04 (Amazon, 2020)	14
Abbildung 12: Accelerometer MPU6050 (Engineering Giniuses, 2020).....	15
Abbildung 13: DC-Motor DG01D-E (Bastelgarage, 2020).....	15
Abbildung 14: 18650 Lithium Shield (Bastelgarage, 2020)	16
Abbildung 15: Motortreiber TB6612FNG (Bastelgarage, 2020)	16
Abbildung 16: Raspberry Pi 4 B (Wired, 2020)	17

TABELLENVERZEICHNIS

Tabelle 1: Übersicht Meilensteine.....	5
Tabelle 2: Zeitplanung Gantt-Diagramm	6
Tabelle 3: Provisorische Kostenberechnung	6
Tabelle 4: Zielkatalog Labirinto.....	7
Tabelle 5: Use-Case 1.....	8
Tabelle 6: Use-Case 2.....	8
Tabelle 7: Auszuführende Funktionstests	10

Einleitung

Sinn und Zweck des Dokumentes

Das Pflichtenheft umfasst sämtliche Anforderungen an die Projektarbeit der Studierenden im 5. Semester des Studiengangs Mechatronik Trinational. Im Folgenden werden Anforderungen an das Projekt «Labirinto» erläutert.

Vision (Inhalt und Ziele)

Unter dem Namen «Labirinto» soll ein autonomer mobiler Roboter entwickelt werden. Dieser Roboter soll eigenständig ein Labyrinth durchfahren können und anschliessend auf dem schnellsten Weg durch das Labyrinth zurückfinden. Der Roboter soll einfach und robust konstruiert sein, damit er ebenfalls als Plattform für zukünftige Projekte und Erweiterungen verwendet werden kann.



Abbildung 1: Logo Labirinto

Im Anschluss an die Projektarbeit möchte das Projektteam ein detailliertes Erklärvideo veröffentlichen, um andere Studierende und Maker für den Nachbau eines «Labirinto» zu begeistern.

Definitionen und Abkürzungen

DC	Direct current (Gleichstrom)
DIY	Do it yourself
FHNW	Fachhochschule Nordwestschweiz
PWM	Pulsweitenmodulation
SSH	Secure Shell
VCC	Voltage at the common collector

Ablage, Gültigkeit und Bezüge zu anderen Dokumenten

Dieses Dokument wurde auf Grundlage des Dokuments «00_ML_Rahmenbedingungen_HS_2020» erstellt.

Verteiler und Freigaben

Rolle / Rollen	Name	Telefon	E-Mail
Co-Projektleiter	Adrian Rosser	+41 79 871 30 31	adrian.rosser@windowslive.com
Co-Projektleiter	Audric Struempler	+33 6 32 16 52 15	audric@struempler.fr
Betreuer FHNW	Silvan Wirth		silvan.wirth@fhnw.ch
Betreuer FHNW	Patrick Grubert		patrick.grubert@fhnw.ch
Kunde / Benutzer	Robert Alard		robert.alard@fhnw.ch

Konzept und Rahmenbedingungen

Ziele und Nutzen des Auftraggebers

Der Auftraggeber erwartet von Labirinto einen zuverlässigen und optisch ansprechenden Labyrinth-Löse-Roboter, welcher für im besten Fall für Werbezwecke (Infoanlässe) verwendet werden kann. Zudem soll der Labirinto einfach zum Nachbauen sein und daher über eine ausführliche Dokumentation verfügen.

Ziele und Nutzen des Anwenders

Der Labirinto soll für den Anwender einfach und möglichst intuitiv zu bedienen sein. Zudem soll dem Anwender ein einzigartiges Fahrerlebnis geboten werden. Vom Roboter darf keine Verletzungsgefahr für den Anwender ausgehen.

Benutzer / Zielgruppe

Auftraggeber / Kunde:	Prof. Dr. Robert Alard
Stakeholder:	Betreuer FHNW (Herr Wirth, Herr Grubert)
Zielgruppe:	Studieninteressierte, Studenten, Maker

Systemvoraussetzungen

Das System ist ein eigenständiges System und wird nicht in ein anderes integriert. Das System soll gut transportierbar sein und über eine wiederaufladbare Stromversorgung verfügen. Für die volle Funktionalität des Systems ist eine funktionierende Wifi-Verbindung vorausgesetzt.

Ressourcen

Co-Projektleiter:	Audric Strümpler (Leiter Software-Entwicklung)
Co-Projektleiter:	Adrian Rosser (Leiter Mechanik- und Elektronikentwicklung)
Infrastruktur:	- Labor FHNW Mechatronik Trinational - Mechanische Werkstatt ADIMADE

Übersicht der Meilensteine

Im Folgenden sind die Aktivitäten und Meilensteine dargestellt. An den Meilensteinen 2, 3 und 4 präsentiert das Projektteam den Stand ihrer Arbeiten den Betreuern und diskutiert das weitere Vorgehen. Der 5. Meilenstein stellt die Abschlusspräsentation des Projekts und die Abgabe der Dokumentation dar.

Tabelle 1: Übersicht Meilensteine

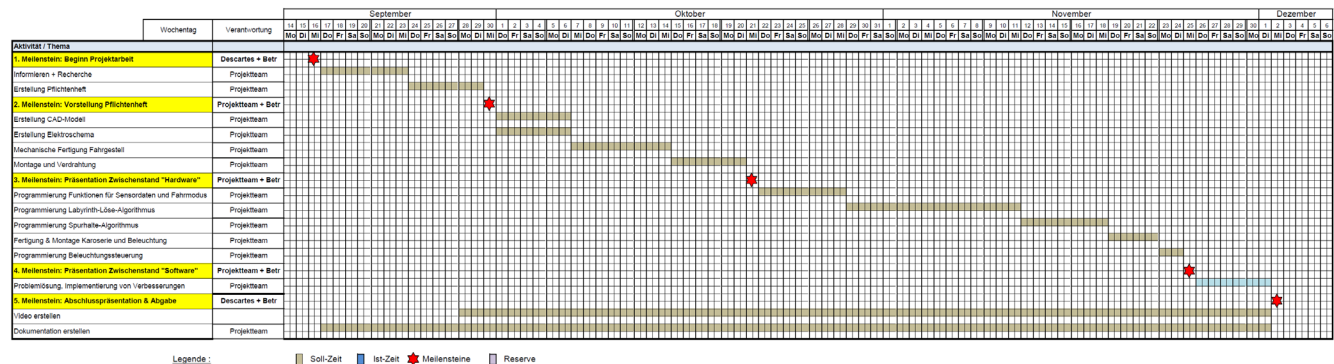
Aktivität	Deadline
1. Meilenstein: Beginn Projektarbeit	16.9.2020
Informieren + Recherche	23.09.2020
Erstellung Pflichtenheft	29.09.2020
2. Meilenstein: Präsentation Pflichtenheft	30.09.2020
Bestellung Material	01.10.2020
Erstellung CAD-Modell	06.10.2020
Erstellung Elektroschema	06.10.2020
Mechanische Fertigung Fahrgestell	14.10.2020
Montage und Verdrahtung	20.10.2020
3. Meilenstein: Präsentation Zwischenstand "Hardware"	21.10.2020
Programmierung Funktionen für Sensordaten und Fahrmodus	28.10.2020
Programmierung Labyrinth-Löse-Algorithmus	11.11.2020
Programmierung Spurhalte-Algorithmus	18.11.2020
Fertigung & Montage Karosserie und Beleuchtung	22.11.2020
Programmierung Beleuchtungssteuerung	24.11.2020
4. Meilenstein: Präsentation Zwischenstand "Software"	25.11.2020
Problemlösung, Implementierung von Verbesserungen	01.12.2020
5. Meilenstein: Abschlusspräsentation & Abgabe	02.12.2020

Grobschätzung des Aufwands

Vom Projektteam wurde vereinbart, dass pro Teammitglied 15 Stunden pro Woche in das Projekt investiert werden sollen. Dabei werden an den Mittwoch Nachmittagen jeweils rund 4 Stunden im Labor gearbeitet und die restlichen 11 Stunden zuhause gearbeitet werden. Ein Student arbeitet dementsprechend bis zur Projektabgabe rund 165 Stunden am Projekt. Gesamthaft ergibt das 330 Soll-Stunden, welche in das «Labirinto» Projekt fließen.

Im Gantt-Diagramm ist der gesamte Soll-Projektablauf dargestellt. Die Aktivitäten wurden dabei aus dem Kapitel «Übersicht der Meilensteine» auf Seite 5 übernommen.

Tabelle 2: Zeitplanung Gantt-Diagramm



Die Kosten für die anzuschaffenden Komponenten belaufen sich auf 55.50 CHF. Der Board-Computer und die Sensoren müssen nicht beschafft werden, da diese bereits im Labor vorhanden sind. Da nur ein Viertel des gesamten Budgets ausgeschöpft ist, wurde beschlossen zwei identische Roboter zu bauen. Dies ermöglicht ein flexibleres Programmieren und Testen von zu Hause.

Tabelle 3: Provisorische Kostenberechnung

Artikel	Artikel-Nummer	Lieferant	Einheit	CHF/ Einheit	Anzahl Einheiten	Kosten Soll
Getriebemotor DG01D-E 1:48 mit Encoder	421286	Bastelgarage	Stk	CHF 9.90	2	CHF 19.80
Motor Treiber TB6612FNG	420520	Bastelgarage	Stk	CHF 4.90	1	CHF 4.90
4x18650 Lithium Batterie Shield	421084	Bastelgarage	Stk	CHF 16.90	1	CHF 16.90
XL6009 Schaltregler DC-DC Step-UP	420162	Bastelgarage	Stk	CHF 5.90	1	CHF 5.90
Kugelrolle 0.4"	RB-Dfr-117	Roboshop	Stk	CHF 4.00	2	CHF 8.00
Material Total						CHF 55.50
Studenten; Soll			h	0	330	CHF 0.00
Studenten; Ist			h	0		
Herstellkosten Total						CHF 0.00
Kosten Total						CHF 55.50

Funktionale Anforderungen

Übersicht der Anforderungen

Die Anforderungen und Ziele an das Projekt wurden in einem Zielkatalog dargestellt. Dieser wurde nach der Logik des «System Engineering» erstellt.

Tabelle 4: Zielkatalog Labirinto

Zielklasse	Zieleigenschaften	Ausmass	Zeitpunkt	Zielart	Priorität
Projektziele					
Fertigungsunterlagen	Erstellung CAD-Modell		02.12.2020	M	-
Fertigungsunterlagen	Erstellen Schaltplan		02.12.2020	M	-
Dokumentation	Erstellen einer vollständigen Projekt-Dokumentation		02.12.2020	M	-
Termin	Projektabschlussstermin einhalten	02.12.2020		R	100
Summe					100
Systemfunktion					
Fortbewegung	Labyrinth vom Start zum Ziel durchfahren	Autonom	02.12.2020	M	-
Fortbewegung	Auf dem schnellsten Weg vom Ziel durchs Labyrinth an den Start fahren	Autonom	02.12.2020	M	-
Fortbewegung	Manuelles Fahren, durch Entwicklungsrechner ferngesteuert	SSH Befehle	02.12.2020	W	20
User Interface	Darstellung von Zuständen mit RGB-LED		02.12.2020	W	4
User Interface	Darstellung des zurückgelegten Wegs auf einem externen Gerät		02.12.2020	W	1
Systemeigenschaften					
Geometrie	Maximale Baubreite	> 300mm	02.12.2020	R	30
Energie	Stromversorgung mit wiederaufladbaren Akkus	5VDC	02.12.2020	M	-
Design	Ansprechendes und modernes Aussehen		02.12.2020	W	5
Finanzen					
Budget	CHF	< 200 CHF	02.12.2020	R	40
Summe					100

Zielarten

M	Muss-Ziel
W	Wunsch-Ziel
R	Restriktionsziel
O	Optimierungsziel

Use-Case Übersicht

Tabelle 5: Use-Case 1

USE CASE Nummer	Labyrinth-Lösen 1
Ziel	Labyrinth lösen und Rückweg optimieren
Kategorie	primär
Vorbereitung	Roboter ist über SSH mit Entwicklungsrechner verbunden. Akku von Roboter ist vollständig geladen
Nachbedingung falls erfolgreich	Roboter fährt autonom vom Start zum Ziel eins Labyrinths und anschliessend auf dem schnellsten Weg zurück
Nachbedingung falls Fehlschlag	Wechsel in Manuellen Fahrmodus
Hauptakteure	Bediener, Kunde
Nebenakteure	Zuschauer
Auslöser	Startbefehl per Knopfdruck
Hauptszenario	Roboter am Labyrinth-Eingang positionieren Startbefehl per Knopfdruck Autonomes Lösen des Labyrinths Autonomes Zurückfahren durch Labyrinth auf schnellstem Weg Wartemodus
Alternative	Roboter mit Entwicklungsrechner verbinden Roboter am Labyrinth-Eingang positionieren Startbefehl an Roboter per SSH übermitteln Autonomes Lösen des Labyrinths Autonomes Zurückfahren durch Labyrinth auf schnellstem Weg Wartemodus

Tabelle 6: Use-Case 2

USE CASE Nummer	Ferngesteuertes Fahren 2
Ziel	Manuelles Fahren durch SSH-Befehle
Kategorie	optional
Vorbereitung	Roboter ist über SSH mit Entwicklungsrechner verbunden. Akku von Roboter ist vollständig geladen
Nachbedingung falls erfolgreich	Roboter fährt manuell durch definierte Richtungsbefehle, welche per SSH an Roboter übermittelt werden.
Nachbedingung falls Fehlschlag	Ausgabe von Fehlermeldung
Hauptakteure	Bediener, Kunde
Nebenakteure	keine
Auslöser	Übermittlung Startbefehl per SSH an Roboter
Hauptszenario	Roboter mit Entwicklungsrechner verbinden Roboter auf freier Fläche platzieren Manuelles Fahren Wartemodus

Use-Case Diagramm

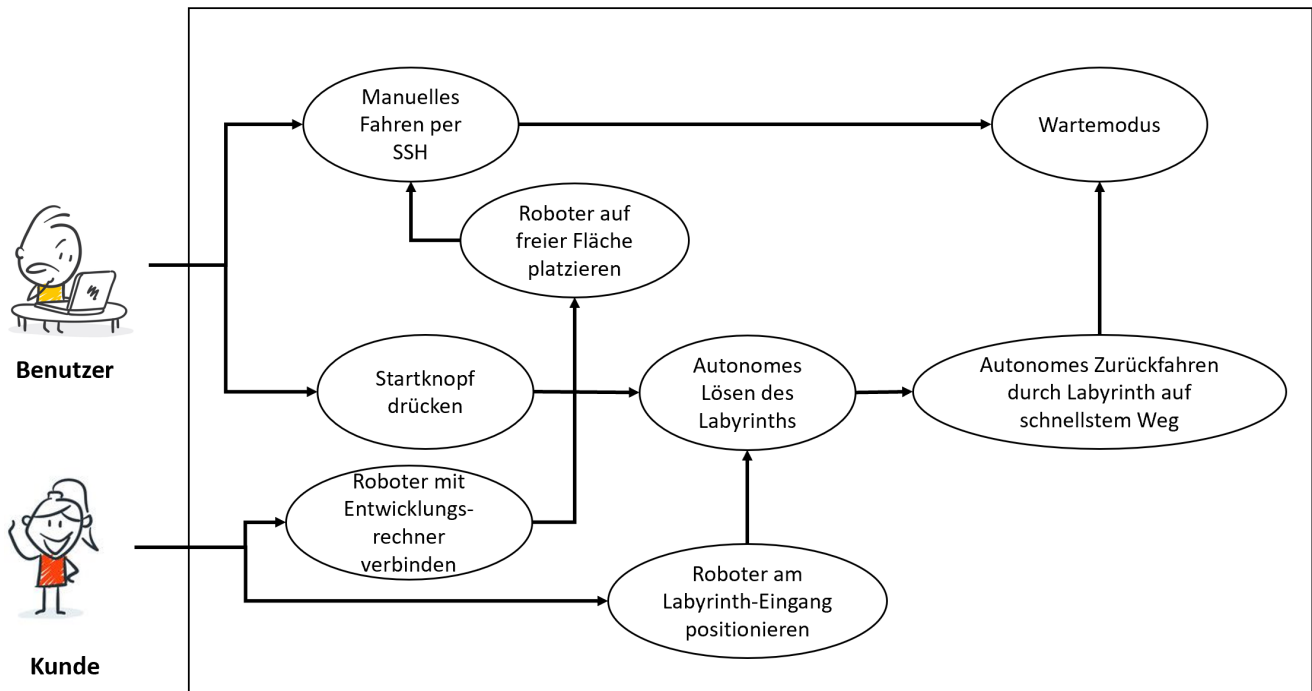


Abbildung 2: Use-Case Diagramm

Risiken

Auf Grund überhöhter Geschwindigkeit oder unkontrollierter Bewegungen kann es zu Kollisionen kommen, was Schäden an der Roboter-Hardware zufolge haben kann. Die Tests der Fahralgorithmen sollten daher in einer sicheren Umgebung durchgeführt werden. Andererseits sind Kurzschlüsse zu vermeiden, um die Elektronik Komponenten nicht zu gefährden.

Testhinweise

Tabelle 7: Auszuführende Funktionstests

Test	Projektphase	Bemerkungen
Korrekte Verdrahtung	Montage und Verdrahtung	Überprüfen ob sämtliche Leitungen gemäss Schema verbunden sind, Messung der Versorgungsspannungen
Funktionen Sensoren	Programmierung Funktionen für Sensordaten und Fahrmodus	Bei diesem Test wird die korrekte Funktion sämtlicher Sensoren und deren Messdaten als Rückgabewert der programmierten Funktionen überprüft
Funktionen Aktoren	Programmierung Funktionen für Sensordaten und Fahrmodus	Es wird die korrekte Ausführung von Fahrfunktionen wie das Drehen oder eine definierte Distanz fahren geprüft
Labyrinth-Lösen	Programmierung Labyrinth-Löse-Algorithmus	Roboter kann fehlerfrei ein Labyrinth vom Start zum Ziel durchfahren
Schnellster Weg zurück	Programmierung Labyrinth-Löse-Algorithmus	Roboter fährt fehlerfrei auf dem schnellsten Weg vom Ziel zum Start zurück
Fährt gerade aus	Programmierung Spurhalte-Algorithmus	Der Roboter fährt ohne merkliche Abweichung seiner Fahrtrichtung gerade aus und korrigiert mögliche Abweichungen automatisch
Gesamtfunktion / Schlusstest	Problemlösung, Implementierung von Verbesserungen	Sämtliche Funktionen können gemäss Zielkatalog erfüllt werden

Vergleich mit bestehenden Lösungen

Im Internet ist eine Vielzahl an DIY-Anleitungen zum Bau eines Labyrinth-Löse-Roboters vorhanden. Bei Projekten für Einsteiger wird meist ein Arduino als Steuerung verwendet. Anleitungen für komplexere Roboter mit leistungsstärkeren Board-Computern sind im Internet nur in begrenztem Ausmass verfügbar. Von verschiedenen Organisationen werden jährliche Labyrinth-Löse-Roboter Meisterschaften veranstaltet. Dabei werden schnelle Roboter eingesetzt, welche über einen leistungsstarken Board-Computer verfügen.

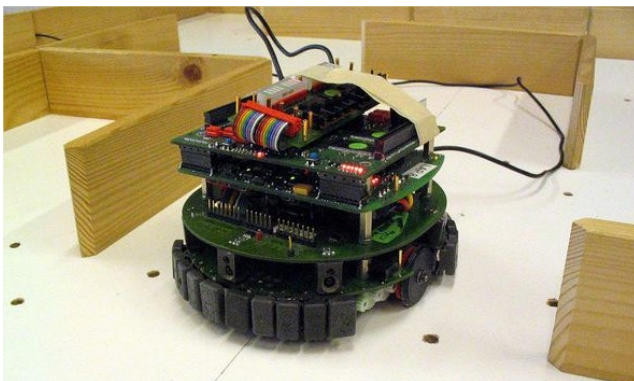


Abbildung 3: Professioneller Labyrinth-Löse-Roboter (Wikipedia, 2020)

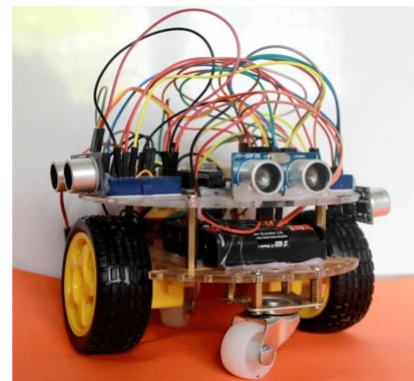


Abbildung 4: DIY-Labyrinth-Löse-Roboter (Research Gate, 2020)

Systemübersicht

Grundsätzlicher Aufbau

Das Blockschaltbild stellt das zu entwickelnde System grafisch dar und erläutert die Beziehungen der Sub-Systeme. Zudem werden die externen Einwirkungen wie physikalische Gegebenheiten oder Ein- und Ausgaben dargestellt.

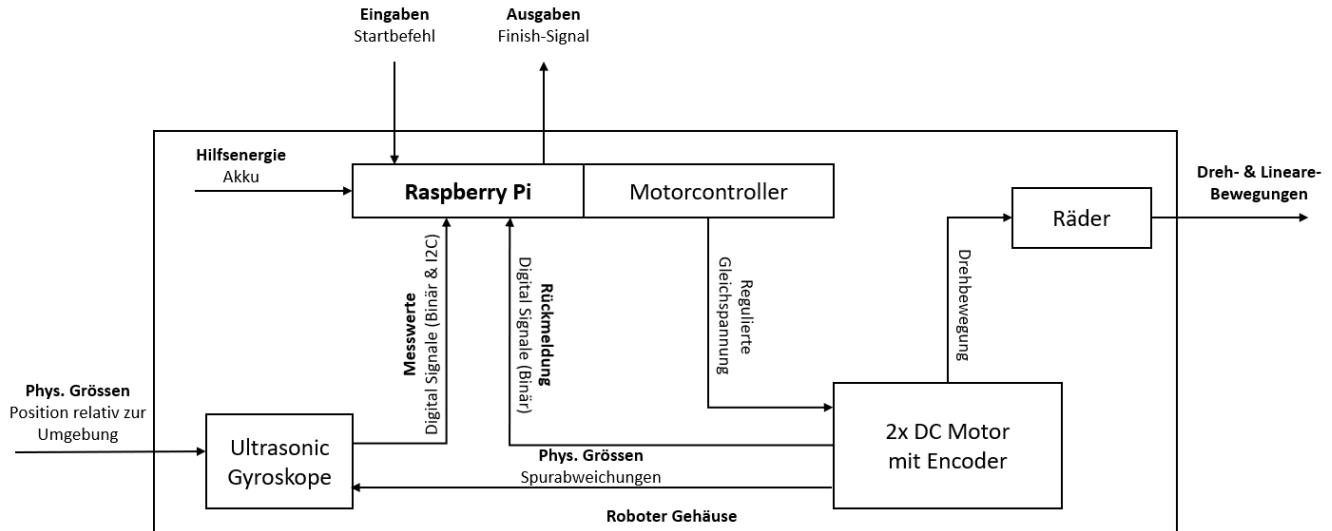


Abbildung 5: Blockschaltbild

Systembeschreibung

Der Labirinto Roboter bildet ein geschlossenes mechatronisches System und besteht aus einer zentralen Recheneinheit, einer OnBoard-Energieversorgung, sowie Aktoren und Sensoren. Das Herzstück eines Labirinto bildet der Raspberry Pi. Dieser führt den autonomen Labyrinth-Löse-Algorithmus aus, verarbeitet sämtliche Sensordaten und steuert die Aktoren an. Zudem wird über den Raspberry Pi die Kommunikation zu einem externen Entwicklungsrechner sichergestellt.

Die Eingaben für den Start des autonomen Labyrinth-Löse-Algorithmus oder die Fernsteuerung des Roboters erfolgt durch einen externen Entwicklungsrechner per SSH.

Von den Sensoren wird die Position des Roboters im Labyrinth erfasst und der Verlauf der Gänge detektiert. Zudem wird die Spurhaltung des Labirinto überwacht und bei Bedarf durch einen Spurhalte-Assistenten korrigiert.

Schnittstellen

Übersicht

Um die verschiedenen Komponenten untereinander auf Hardware- und Softwareebene miteinander zu vernetzen müssen diese mit definierten Schnittstellen verbunden werden. Auf Hardwareebene wird dies durch Kabel und Drahtlosen Verbindungen mit definierten Bus- und Binär-Verbindungen sichergestellt. Auf Softwareebene wird die Kommunikation über standardisierte Protokolle geregelt.

Hardwareschnittstellen

Die Komponenten, welche im Labirinto verbaut werden, sind durch Kabel miteinander verbunden. Die Kommunikation zwischen Raspberry Pi und Gyroskop-Sensor wird mit einem I2C-Bus ermöglicht. Der übrige Datenaustausch im Roboter wird über Binäre und PWM-Signale durchgeführt.

Die Kommunikation des Roboters nach aussen erfolgt mittels einer Wifi-Verbindung. Dies ermöglicht die Verbindung des Labirinto mit einem externen Entwicklungsrechner.

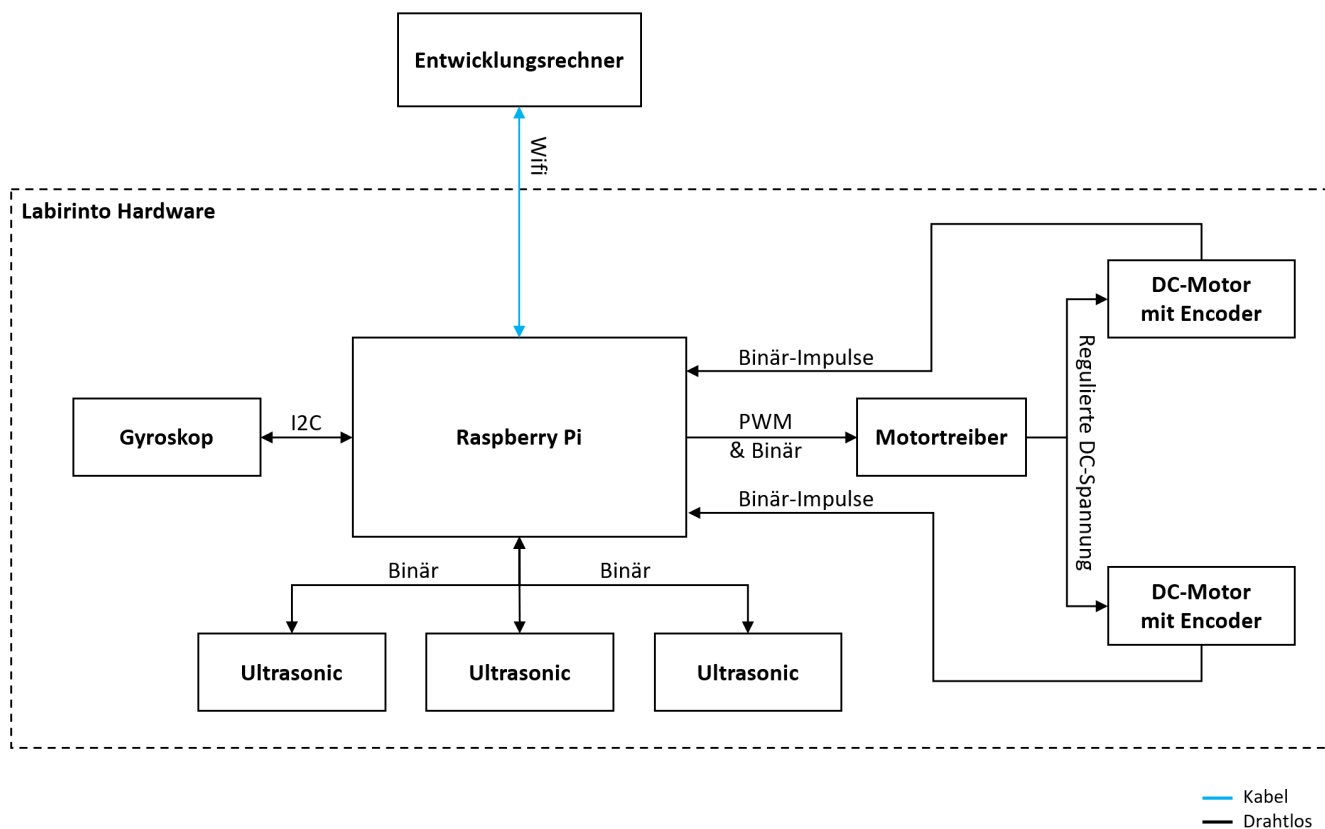


Abbildung 6: Übersicht Schnittstellen

Softwareschnittstellen

Mit dem Secure Shell (SSH) Protokoll kann der Raspberry Pi von einem externen Computer über eine Wifi-Verbindung angesteuert werden. Damit können Startbefehle für den autonomen Labyrinth-Löse-Algorithmus gesendet werden. Zudem kann man dadurch den Labirinto manuell fernsteuern.

Mechanik

Mechanische Struktur

Der Labirinto soll nach aussen als einen modernen und innovativen Roboter, mit einem soliden mechanischen Aufbau, wahrgenommen werden. Das Fahrgestell und die Elektronik sollen durch ein rundes, helles Gehäuse geschützt werden. Es werden Aussparungen am Gehäuse für die Distanzsensoren und die Energieversorgung vorgesehen.

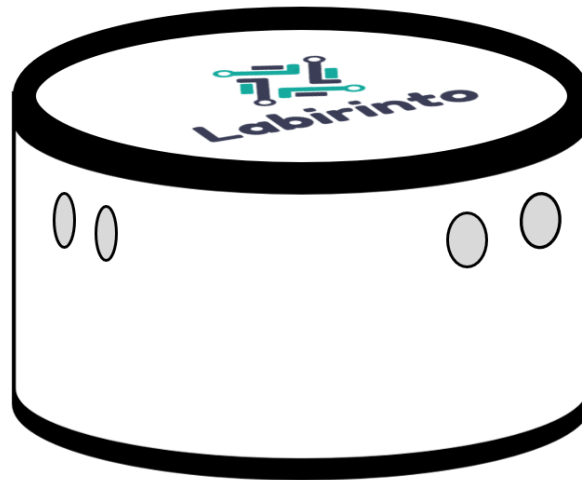


Abbildung 7: Entwurfsskizze Labirinto

Der Roboter besteht aus zwei Plattformen, welche einfach durch Distanzsäulen miteinander verbunden sind. Auf der unteren Plattform A ist die Stromversorgung, die Motorsteuerung und die DC-Motoren mit sämtlichen Rädern positioniert.

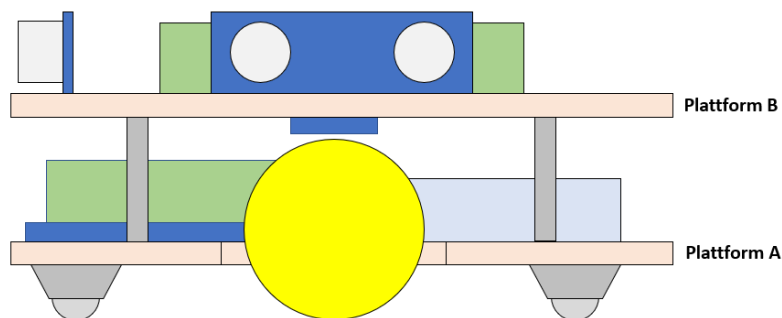


Abbildung 8: Mechanischer Aufbau Labirinto

Auf der oberen Plattform B ist der Raspberry Pi und die drei Ultraschall- und dem Gyroskop-Sensor platziert. Die Plattformen sollen modular montierbar sein.

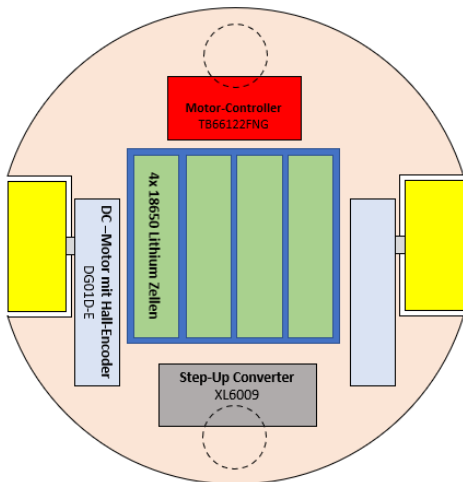


Abbildung 9: Aufbau Plattform A

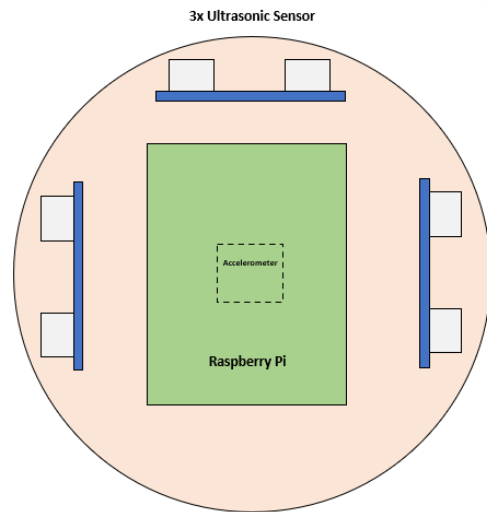


Abbildung 10: Aufbau Plattform B

Sensoren

Ultraschall Sensoren

Für die räumliche Orientierung des Roboters sind drei Ultraschall Sensoren vorgesehen. Diese dienen zur Messung der Entfernung zum nächsten Hindernis. Der Sensor sendet ein hochfrequentes Signal aus und empfängt das reflektierte Signal anschliessend. Die Distanz wird durch die zeitliche Differenz zwischen dem Sensen und Empfangen des Signals mittels der Schallgeschwindigkeit ermittelt.

$$Speed = \frac{Distance}{Time}$$

$$Distance = Speed * \left(\frac{Time}{2}\right)$$

Schallgeschwindigkeit: $343m/s = 343000mm/s$

$$Distance = \frac{343000 * Time}{2}$$

Als Sensor ist der HC-SR04 vorgesehen, da dieser in genügender Stückzahl im Labor des Studiengangs Mechatronik Trinational verfügbar ist.

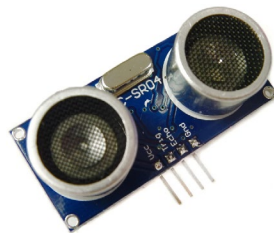


Abbildung 11: Ultrasonic Sensor HC-SR04
(Amazon, 2020)

Encoder

Die beiden Antriebsmotoren des Labirinto verfügen über integrierte Encoder. Mit den Encodern kann die Geschwindigkeit der beiden Räder individuell überprüft werden und allfällige Differenzen nachgeregelt werden. Dadurch kann eine geradlinige Fortbewegung des Roboters sichergestellt werden.

Ebenfalls kann die Umdrehungsgeschwindigkeit der Räder gemessen werden und dadurch die zurückgelegte Distanz des Roboters ermittelt werden.

Accelerometer

Um das Manövrieren des Roboters genauer zu überwachen wurde der Einbau eines Gyroskop-Sensors beschlossen. Dieser misst den Winkel beim Drehen des Roboters um seine Achse. Der Sensor muss beispielsweise über einen I2C-Bus mit dem Raspberry Pi kommunizieren können, da der Raspberry Pi über keine analogen Eingänge verfügt.



Abbildung 12: Accelerometer MPU6050
(Engineering Giniuses, 2020)

Aktoren

DC Motor

Der Antrieb des Labirinto soll über zwei DC-Motoren erfolgen. DC-Motoren über eine H-Brücke und der regulierten Versorgungsspannung einfach anzusteuern. Vorgesehen ist der DG01D-E der Firma DFRobot. Dieser Motor verfügt über ein Getriebe mit einer Übersetzung von 1:48 sowie über einen integrierten Hall-Encoder und ist somit für das Projekt bestens geeignet.

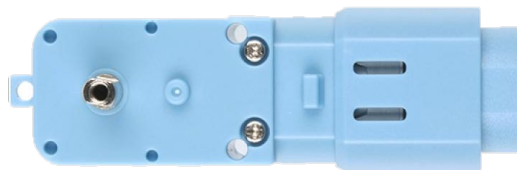


Abbildung 13: DC-Motor DG01D-E
(Bastelgarage, 2020)

Elektronik

Stromversorgung

Die Versorgung der sämtlichen Komponenten erfolgt durch wiederaufladbare 18650 Lithium-Batterien. Ein geeignetes Batterie-Shield übernimmt die Lade-Überwachung der Akkus und regelt die Ausgangsspannung auf 5V beziehungsweise auf 3.3 V. Die Aufladung erfolgt wahlweise über Micro USB oder USB-C mit 5V.



Abbildung 14: 18650 Lithium Shield
(Bastelgarage, 2020)

Motortreiber

Die DC Motoren werden vom Raspberry Pi über einen zwei Kanal-Motorentreiber angesteuert. Die integrierte H-Brücke ermöglicht das Wechseln der Drehrichtung. Die Geschwindigkeit wird über PWM-Signale vom Raspberry Pi gesteuert. Als Motortreiber ist der TB6612FNG von SparkFun vorgesehen.

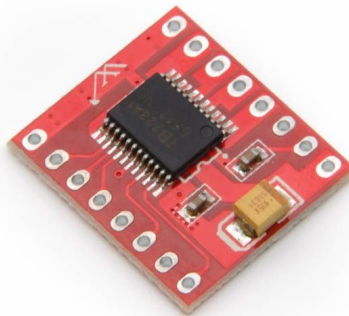


Abbildung 15: Motortreiber TB6612FNG
(Bastelgarage, 2020)

Informationsverarbeitung

Board-Computer

Das Herz des Labirinto bildet der Raspberry Pi, welcher als Board-Computer dienen soll. Das Projekt-Team hat sich für den Raspberry Pi entschieden, da man sicher mit einer fortgeschrittenen Alternative zum bestens bekannten Arduino vertraut machen will. Es gibt zudem eine Vielzahl an verfügbaren Software-Klassen in Python für Raspberry Pi Roboter.

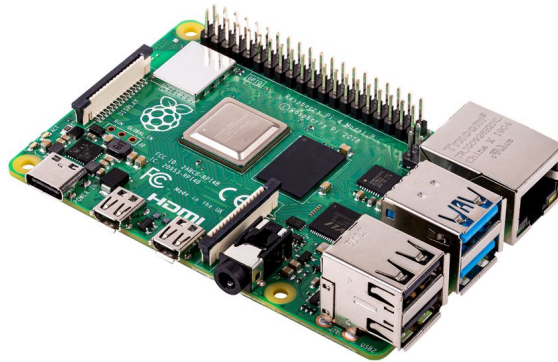


Abbildung 16: Raspberry Pi 4 B
(Wired, 2020)

Der Raspberry Pi ist ein Einplatinenrechner, mit einem ARM-Prozessor, einem Arbeitsspeicher und einem Grafikprozessor. Über folgende Schnittstellen verfügt der Raspberry Pi:

- Wireless Lan
- Bluetooth 4.1
- USB & Micro USB
- HDMI
- Ethernet
- Audio Jack
- DSI
- 40 GPIO

Für den Labirinto werden die General Purpose Input Output verwendet, um die Sensoren und Aktoren an zu steuern mit einer Spannung von 3.3V. Das WLAN dient dazu eine SSH Verbindung mit einem Computer herzustellen, um Steuer-Befehle an den Raspberry Pi zu übermitteln.

Benutzerinterface

Der Labirinto verfügt über kein integriertes Benutzerinterface. Die gesamte Kommunikation erfolgt über ein Terminalfenster eines Entwicklungsrechners.

Anhang / Ressourcen

Impressum

Datum der Erstellung des Pflichtenhefts: Herbst 2018

Projektarbeit Semester 5 Promotion Descartes – 2020 – Rosser Adrian, Strümpfer Audric

© Fachhochschule Nordwestschweiz, Studiengang Mechatronik Trinational, 2018

www.trinat.net

Quellenangaben

Literaturverzeichnis

Amazon. (28. September 2020). Von <https://www.amazon.com/HC-SR04-HC-SR04P-Ultrasonic-Distance-Measuring/dp/B07KNTQ4C2> abgerufen

Bastelgarage. (28. September 2020). Von <https://www.bastelgarage.ch/getriebemotor-dg01d-e-1-48-mit-encoder> abgerufen

Bastelgarage. (28. September 2020). Von <https://www.bastelgarage.ch/solar-lipo/4x18650-lithium-batterie-shield-5v-3a-3v-1a> abgerufen

Bastelgarage. (28. September 2020). Von <https://www.bastelgarage.ch/bauteile/stepper-motoren/motor-treiber-dual-1-2a-tb6612fng> abgerufen

Engineering Giniuses. (28. September 2020). Von <http://elec.egeniuses.net/product/mpu6050-6-axis-imu/> abgerufen

Research Gate. (28. September 2020). Von https://www.researchgate.net/figure/Figure-31-Autonomous-Maze-solving-Robot_fig4_316664613 abgerufen

Wikipedia. (28. September 2020). Von https://en.wikipedia.org/wiki/Maze_solving_algorithm abgerufen

Wired. (28. September 2020). Von <https://www.wired.com/review/raspberry-pi-4/> abgerufen