

Diplomarbeit

Ferngesteuertes LEGO-Auto

2024



Matteo Tonazzi
TEKO Luzern
Klasse:L-TEL-21
Techniker Elektrotechnik
2024

1 Inhaltsverzeichnis

1	Inhaltsverzeichnis	2
2	Management Summary	4
3	Kurzer beruflicher Lebenslauf.....	6
4	Projektinitialisierung.....	7
4.1	Pflichtenheft	7
4.2	Zielscheibe	12
5	Projektplanung	14
5.1	Vorgehensmodell	14
5.2	Projektstrukturplanung.....	14
5.3	Projektablaufplanung	15
5.4	Kommunikationsplanung	17
6	Projektrealisierung	18
6.1	Recherche.....	18
6.1.1	Mikrokontroller	18
6.1.2	Spannungsversorgung	18
6.1.3	Lenkung	18
6.1.4	Antrieb.....	20
6.1.5	Lego-Auto	21
6.2	Zusammenbau Lego-Auto	22
6.3	Programmieren	23
6.3.1	Lenkung	23
6.3.2	Antrieb.....	27
6.3.3	Visualisierung	29
6.4	Zusammenbau	31
6.4.1	Elektronischer Zusammenbau.....	31
6.4.2	Elektronik in Lego-Auto	32
6.5	Bedienungsanleitung	34
6.6	Testen	36
6.6.1	Lenkung Programmiert.....	36
6.6.2	Lenkung Eingebaut.....	36
6.6.3	Antrieb Programmiert	36
6.6.4	Antrieb Eingebaut	36
6.6.5	Bedienungsanleitung.....	37
7	Projektabschluss.....	38
7.1	Projektüberwachung	38
7.2	Evaluation der Zielerreichung.....	38
7.3	Reflexion Weg zum Ziel.....	40
7.4	Lessons learnt.....	41

7.5	Ausblicke.....	41
8	Eigenständigkeitserklärung	42
8.1	Schlusswort.....	43
9	Verzeichnisse.....	44
9.1	Abbildungsverzeichnis.....	44
10	Anhang.....	45
10.1	Projektstatusberichte	45

2 Management Summary

Im Zuge des Abschlusses des Techniker Elektrotechnik Lehrganges an der TEKO Luzern darf ich eine Diplomarbeit mit selbst gewähltem Thema bearbeiten. In der folgenden Arbeit wird ein normales Lego-Auto umgebaut und zu einem ferngesteuerten Lego-Auto gemacht. Ich habe Elektromotoren eingebaut und diese über einen Mikrocontroller gesteuert, sodass man die Motoren mit dem Handy ansteuern kann.

Ausgangslage

- Es sollen bei einem «normalen» Lego-Auto Elektromotoren so eingebaut werden, dass man es als ferngesteuertes Auto mit dem Handy steuern kann.
- Ich bin ein grosser Legofan, habe auch schon einen ferngesteuerten Bulldozer und 2 Modellautos von Lego und habe gedacht mit den in der Weiterbildung behandelten Themen habe ich das Know-How dies selber für ein Standardauto zu implementieren.
- Man Sollte nicht direkt sehen, dass das Auto nicht mehr im Originalzustand ist.
- Das Auto soll bedienerfreundlich gebaut sein und mit einer Bedienungsanleitung versehen sein, so dass es selbstständig bedient werden kann.
- Es soll genug Akku haben, um mindestens 15 Minuten zu fahren.

Vorgehen

- Als erstes habe ich eine Recherche durchgeführt, welche Bauteile die passenden sind um dieses Projekt zu realisieren und diese dann bestellt.
- Danach habe ich die Bauteile provisorisch verdrahtet um zu testen, ob sie funktionieren.
- Als nächstens kam die Programmierung, bei welcher ich Schritt für Schritt die einzelnen Funktionen programmiert habe, welche ich haben wollte.
- Für die Recherche habe ich viele elektronische Bauteile angeschaut.
- Bei der Programmierung habe ich hauptsächlich Wahrheitstabellen verwendet. Und diese mit Hilfe eines KV-Diagrammes vereinfacht.

Ergebnisse

Ich habe ein Lego-Auto fahrtüchtig zu einem ferngesteuerten Auto umgebaut.

Die vorgegebenen Funktionen wurden alle erfüllt. Ich kann das Auto mit zwei verschiedenen Modi bedienen, einem schnellen und einem langsamen. Es fährt vorwärts und rückwärts und man kann es lenken. Die Spannungsversorgung wurde so gewählt das es länger als 15 Minuten fahren kann. Die Bedienung ist simpel und kann mithilfe einer einseitigen Bedienungsanleitung selbstständig durchgeführt werden. Jedoch ist die Lenkung ein bisschen langsam und nicht mit einem Endanschlag abgesichert.

Beim langsamen Modus hört man aufgrund von mechanischen Relais ein Klappern.

Der eine Antriebsmotor ist ein bisschen gross, weswegen man nicht über Schwellen oder unebene Böden fahren kann, da er sonst den Boden berührt.

Auch wird das Auto mit zwei Motoren angetrieben. Jedoch sind die Motoren nicht für das linke und rechte Hinterrad sondern einer für die Hinterachse und einer für die Vorderachse. Dies musste ich so machen, weil die Hinterachse zu wenig Platz hatte für zwei Motoren und ein Motor nicht gereicht hat.

Ausblick

Ich habe hier ein funktionsfähiges ferngesteuertes Lego-Auto. Jedoch gibt es trotzdem noch das eine oder andere Verbesserungspotential, welches ich aus zeittechnischen Gründen nicht umsetzen konnte.

Ich würde einen schnelleren Lenkungsmotor einbauen, welcher einen Endanschlag hat, dann würde ich die Hinterachse mit einem anderen Motor antreiben, sodass man auch über unebene Flächen fahren kann ohne das der Motor den Boden berührt.

Mit den restlichen Funktionen des Autos bin ich sehr zufrieden und glücklich dass ich so etwas erreichen konnte. Ich freue mich darauf, das Auto nach der Diplomzeit noch weiter zu optimieren.

3 Kurzer beruflicher Lebenslauf

Meine berufliche Grundbildung habe ich zwischen 2015 und 2019 mit der Berufslehre zum Elektroinstallateur verrichtet. In dieser Zeit habe ich vorwiegend auf Neu- und Umbauten gearbeitet.

In meinem momentanen Betrieb sind wir in mehreren grösseren Industriebetrieben zur Unterstützung der Betriebselektriker tätig. Wir machen diverse Unterhaltsarbeiten, Umbauten oder Neubauten von Steuerungen und Anlagen und helfen bei Störungsbehebungen an Anlagen. Jetzt bin ich am Ende des 6. Semesters an der TEKO Luzern, Fachrichtung Techniker Elektrotechniker HF.

4 Projektinitialisierung

4.1 Pflichtenheft

Im Rahmen der Vorbereitungsarbeiten für die Diplomarbeit habe ich ein Pflichtenheft erstellt, welches das geforderte Endprodukt genau umschreibt und welches ich als Leitfaden verwende bei der Erarbeitung des Endprodukts.

Einleitung

Als Abschluss der Weiterbildung als Techniker-Elektrotechnik HF dürfen die Studierenden mit Hilfe einer Diplomarbeit zeigen, was sie in den letzten drei Jahren im Unterricht gelernt haben.

Grobbeschrieb Auftrag

Es sollen bei einem Modell-Legoauto Elektromotoren eingebaut werden, so dass dieses dann über das Handy gesteuert werden kann.

Ausgangslage

Es gibt diverse Bausätze, welche ein Legoauto beinhalten, welche bereits mit vorprogrammierten Elektromotoren und einer vorprogrammierter App ausgerüstet sind, mit welcher man das Auto steuern kann. Die Preisspanne liegt zwischen 60.00 CHF und 600.00 CHF. Die Kits sind jedoch kaum erweiterbar oder kombinierbar. Wenn man später noch etwas ergänzen möchte, z.B Lichter oder Soundeffekte, muss man das Auto selber ausbauen und programmieren. Ein weiterer Nachteil der auf dem Markt erhältlichen Autos ist, dass diese immer von Lego Technic sind. Deren Design hat sich in den letzten Jahren schon etwas verbessert, aber sie können immer noch nicht mit der Detailgenauigkeit der Ausstellungsmodelle von Lego mithalten.

Es ist mein Ziel, aus einem Ausstellungsmodell ein ferngesteuertes Legoauto zu machen, bei welchem man später noch andere Gadgets nachprogrammieren kann. Somit wäre das Beste aus allen Lego Varianten, in einem selbst erstellten Auto vereint - die Detailgenauigkeit eines Ausstellungsmodells und der Spass damit ferngesteuert herum zu fahren.

Fachexperte

Der Fachexperte ist Luca Stalder. Er hat nach der Kantonsschule sein Bachelor- und Masterstudium in Maschineningenieurwissenschaften an der ETH Zürich absolviert und dieses vor kurzen erfolgreich abgeschlossen. Während seines Studiums und in seiner Freizeit hat er bereits einige Projekte mit den Mikrocomputern von Raspberry durchgeführt.

Kontakt:

Luca Stalder

Kornfeld 11

6212 St. Erhard

079 365 61 71

Projektdefinierung

Das Ziel ist es, ein Modellauto von Lego auf möglichst unauffällige Art und Weise umzubauen, dass man es mit dem Handy als ferngesteuertes Auto bedienen kann.

Funktionen

Das Auto soll mit dem Handy bedienbar sein. Man soll auf einem Bedienfeld mit einem Joystick nach links und nach rechts steuern können und mit zwei Pedalen nach hinten und nach vorne fahren können. Es sollen mindestens 2 verschiedene Beschleunigungen programmiert sein, so dass man einen «Sport» und einen «eco»-Modus hat. Das Auto soll mit einer Akkuladung mindestens 15 Minuten fahren können.

Endergebnisse / Erfolgskriterien

Folgende Endergebnisse mit dazugehörigen Erfolgskriterien sollen eingehalten werden.

Endergebnis	Erfolgskriterien
Es liegt eine nachvollziehbare Dokumentation vor, welche rechtzeitig abgegeben wird.	- Am 11. November um 16:00 Uhr ist die Dokumentation in PDF im Extranet hochgeladen. - Die Dokumentation wird streng nach dem 4 Phasenmodell erarbeitet. - Die Dokumentation kann von 5 Personen mit elektrotechnischem Hintergrund durchgelesen werden, welche weniger als 3 Rückfragen aufgrund von Unklarheiten haben.
Das Auto soll 15 Minuten Fahren können.	Der Akku muss so gross gewählt werden, dass man mit dem Auto im «Sport-Modus» und bei höchster Geschwindigkeit mindestens 15 Minuten herumfahren kann.
Es soll eine allgemeinverständliche Bedieneroberfläche und Bedienungsanleitung vorliegen.	Die Bedienungsanleitung kann an 5 Personen, die nicht technisch versiert sind, kommentarlos abgegeben werden und sie können sich ohne Rückfragen mit dem Auto verbinden, den Akku wechseln, den Akku laden und das Auto bedienen.
Die Komponenten wollen so eingebaut werden, dass man es nicht direkt sieht.	Das Auto muss so umgebaut werden, dass man bei geschlossenen Türen keine Kabel oder elektrischen Komponenten sieht.

Das Auto soll mit der Vorderachse gesteuert werden und mit der Hinterachse angetrieben werden.	- Der Motor zur Steuerung der Vorderachse soll so eingebaut werden, dass sich das Lenkrad immer noch dreht. - Die Hinterachse wird mit je einem Motor pro Rad angetrieben.
Es sollen verschiedene Modis möglich sein.	Es muss mindestens 2 Modus geben (ECO und Sport) welche einen Unterschied in der Beschleunigung und der Maximalgeschwindigkeit vorweisen.

System

Die Software mit welcher es laufen soll ist Codesys. Die Verbindung zwischen Handy und Raspberry Pi soll auf W-Lan Basis sein.

Hardware

Das Auto wird mit Hilfe eines Raspberry Pi Zero gesteuert. Die Motoren sollen die Möglichkeit haben, verschiedene Geschwindigkeiten zu fahren.

Auftragsabnahme

Das Pflichtenheft enthält alle für den Auftragsgeber wichtigen Vorgaben für das Endprodukt. Abweichungen müssen schriftlich beantragt werden und per Unterschrift beider Parteien bestätigt werden.

Fachexperte:

Luca Stalder

Luca Stalder

Diplomand:

Matteo Tonazzi

M. Tonazzi

4.2 Zielscheibe

Aus dem Pflichtenheft habe ich die Richtziele und die Erfolgskriterien genommen und den Konkretisierungsgrad angepasst. Daraus habe ich die Zielscheibe erstellt, mit welcher ich den Auftrag übersichtlich in einem Dokument habe.

Richtziel: Ferngesteuertes Lego Auto

1. Es liegt eine nachvollziehbare Dokumentation vor, welche rechtzeitig abgegeben wird. 2. Das Auto soll 15 Minuten Fahren können. 3. Es soll eine allgemeinverständliche Bedieneroberfläche und Bedienungsanleitung vorliegen. 4. Die Komponenten sollen fachgerecht montiert und installiert sein. 5. Das Auto soll mit der Vorderachse gesteuert werden und mit der Hinterachse angetrieben werden. 6. Es sollen verschiedene Modis möglich sein. 7. Die Programmierung muss nachvollziehbar sein.	– Als Fachexperten habe ich Luca Stalder Master Maschinenbauwesen der ETH Zürich
Sinn und Zweck	Erfolgskriterien
– Sinn und Zweck ist es in einer Diplomarbeit als Abschluss einer Weiterbildung, die gelernten Fähigkeiten an einem praktischen Beispiel anzuwenden und umzusetzen –	1. Am 11. November um 16:00 Uhr ist die Dokumentation in PDF im Extranet hochgeladen. Die Dokumentation wird streng nach dem 4 Phasenmodell erarbeitet. Die Dokumentation kann von 5 Personen mit elektrotechnischem Hintergrund durchgelesen werden, welche weniger als 3 Rückfragen aufgrund von Unklarheiten haben. 2. Der Akku muss so gross gewählt werden, dass man mit dem Auto im «Sport-Modus» und bei höchster Geschwindigkeit mindestens 15 Minuten herumfahren kann. 3. Die Bedienungsanleitung kann an 5 Personen, die nicht technisch versiert sind, kommentarlos abgegeben werden und sie können sich ohne Rückfragen mit dem Auto verbinden, den Akku wechseln, den Akku laden und das Auto bedienen. 4. Die Komponenten sind so eingebaut, dass sichtbare und berührbare Komponenten aller Art, von Beschädigung und abreißen geschützt sind. 5. Der Motor zur Steuerung der Vorderachse soll so eingebaut werden, dass sich das Lenkrad immer noch dreht. Die Hinterachse wird mit je einem Motor pro Rad angetrieben. 6. Es muss mindestens 2 Modus geben (ECO und Sport) welche einen Unterschied in der Beschleunigung und der Maximalgeschwindigkeit vorweisen. 7. Die Programmierung wird so dokumentiert, dass man ohne in das Programm zu schauen, weiss wie dieses aussieht. Das Programm wird wo möglich mit Grafiken ausgearbeitet.

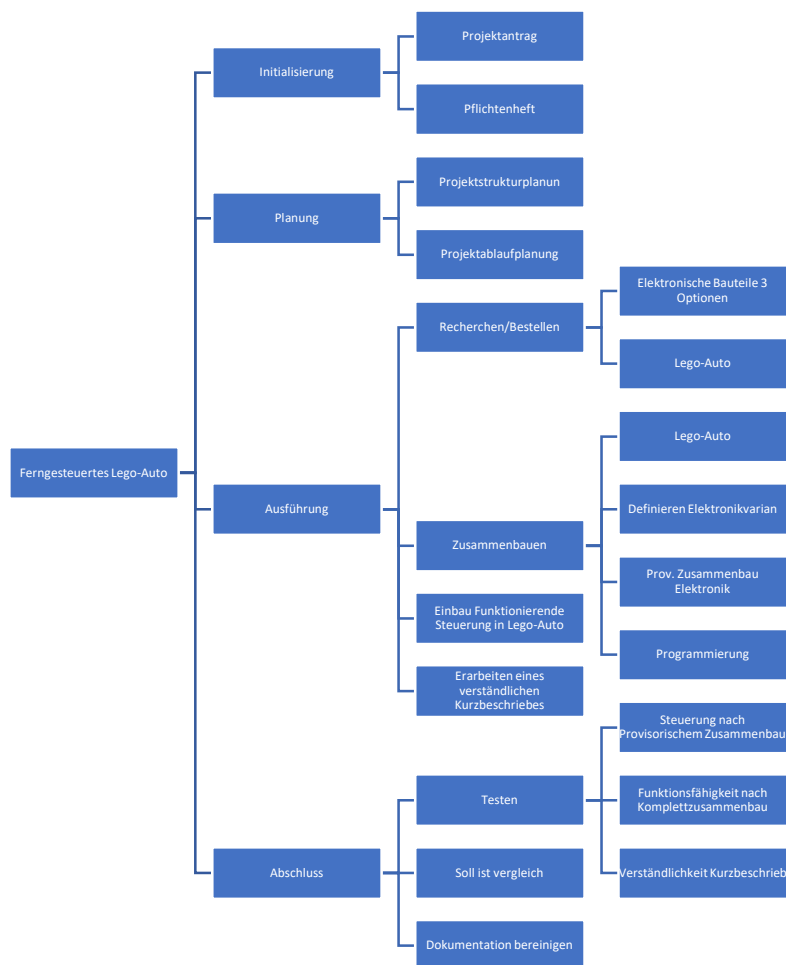
5 Projektplanung

5.1 Vorgehensmodell

Ich werde meine Diplomarbeit nach dem Wasserfallprinzip durchführen. Die Arbeitspakete können nicht parallel erledigt werden, da es ein aufbauender Prozess ist. So eignet sich das klassische Wasserfallmodell am besten, da man da ein Paket nach dem anderen erledigt. Für mich ist es am einfachsten, wenn ich eins nach dem anderen mache und mich nicht in zu vielen Paketen gleichzeitig beschäftigt bin und dadurch den Fokus verliere.

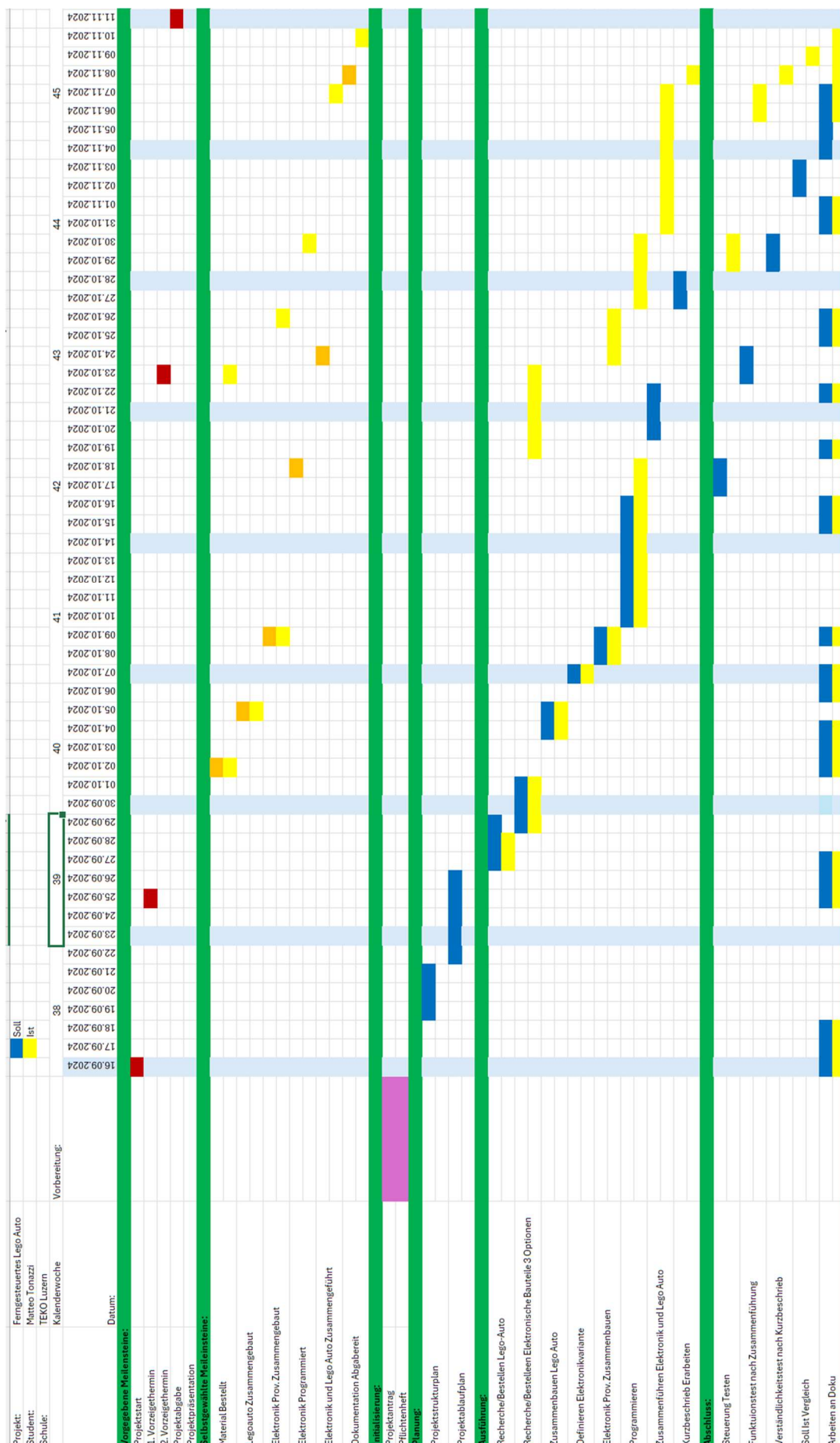
5.2 Projektstrukturplanung

Ich habe die wichtigsten Eckpunkte aufgeschrieben. Danach habe ich mir überlegt, wie ich diese verfeinern kann. Somit bin ich «from top to bottom» Stück für Stück zu einem Projektstrukturplan gelangt, welcher ziemlich verständlich und ausgearbeitet ist.



5.3 Projektablaufplanung

Aus dem Projektstrukturplan habe ich einen Projektablaufplan erstellt, bei welchem ich genau beschrieben habe, wann ich welches Arbeitspaket bearbeiten möchte. Somit habe ich einen zeitlichen Horizont, wann ich was erledigt haben muss mit selbst ausgewählten Meilensteinen.



5.4 Kommunikationsplanung

Ich werde die Diplomarbeit alleine machen. Die einzige Kommunikation welche ich habe, ist der Statusbericht an meinen Diplomlehrer, der jeden Dienstag stattfindet und zwei Vorzeigetermine. Die Vorzeigetermine sind jeweils per Microsoft Teams. Bei allfälligen Problemen dann ich den Diplomlehrer per Mail um Hilfe fragen.

6 Projektrealisierung

Mein Ziel ist es bei einem Lego-Auto Elektromotoren einzubauen, welche ich dann mit dem Handy lenken und steuern kann. Hierfür werde ich zuerst eine eingehende Recherche durchführen, welche Motoren für die Lenkung und welche Motoren für den Antrieb in Frage kommen.

6.1 Recherche

Ich brauche 4 verschiedene Bauteile. Ich brauche eine Spannungsversorgung für die Last und für den Raspberry PI für die Lenkung einen Motor, für den Antrieb und ich brauche eine Spannungsversorgung. Ausserdem brauche ich das passende Lego-Auto.

6.1.1 Mikrokontroller

Als Mikrocontroller habe ich mich für einen Raspberry PI Zero 2 entschieden. Mit diesem haben wir im 6. Semester gearbeitet. Betrieben haben wir ihn mit dem Programm Codesys. Somit habe ich einen Mikrokontroller, bei welchem es einfach ist, eine Onlinevisualisierung zu erstellen. Die Bedienung werde ich über WLAN erstellen.

6.1.2 Spannungsversorgung

Für die Spannungsversorgungen habe ich es mir einfach gemacht. Ich habe zuhause noch ein Modellbau Akkupaket für 7.2V Spannungsversorgung und eine kleine Powerbank für die Spannungsversorgung des Raspberry PI.

6.1.3 Lenkung

Für die Lenkung habe ich mir gedacht, dass ich einen Servomotor benutze. Für diesen habe ich mich entschieden, weil diese meistens eine Spannweite von 180° haben, was für die meisten Lenkungen wunderbar passt. Auf dem Markt gibt es viele verschiedene Modelle. Es gibt die digitalen Servomotoren, welche mit der sogenannten Pulsweitenmodulation gesteuert sind. Das heisst, dasin einem 20 Millisekunden Intervall ein Impuls gesendet wird, und je nach der Länge des Impulses t dreht sich der Servo nach Links oder Rechts. Ausserdem gibt es die analogen Servomotoren. Diese werden drehen sich, wenn sich die Spannung ändert. Hierfür habe ich zwei Standard-Servos gefunden. Den MASTER DS2312MG und den Futaba S3001. Ich habe mich für den digitalen Motor entschieden, da der benutzte Raspberry PI Zero 2 keinen passenden Ausgang für einen analogen besitzt.



Abbildung 1

<https://www.mhm-modellbau.de/bilder/PI-C5637.jpg>

Während dem Programmieren habe ich gemerkt, dass es mit dem Servomotor nicht geht. Entweder ist das Codesys oder der Raspberry Ausgang zu langsam. Beim Testen der Steuerung hat gar nichts funktioniert. Also musste ich mir überlegen, was ich als Alternative nehme. Die beiden anderen handelsüblichen Elektromotorentypen sind Schrittmotoren oder DC-Motoren. Die Schrittmotoren haben einen Motorentreiber, auf welchen die zwei Motorenwicklungen des Schrittmotoren geführt werden. Auf den Motorentreiber gibt man zwei Signale. Ein Richtungssignal, Wenn es High (1) ist dreht sich der Motor in die eine Richtung und wenn es Low (0) ist in die andere Richtung. Der Motor macht pro Impuls einen «Schritt» in die vorgegebene Richtung. Somit kann man mit dem Richtungssignal und dem Schrittsignal die Richtung, die Geschwindigkeit und die Anzahl Schritte für die Drehung bestimmen.

Der DC-Motor wird direkt an Plus und Minus angeschlossen und dreht solange Spannung ansteht. Wenn die Drehrichtung geändert werden soll, muss man Plus und Minus drehen. Je höher die Spannung ist, desto schneller dreht sich der Motor.

Für die Lenkung habe ich mich für einen Schrittmotor entschieden, weil man bei dem die Schritte zählen kann und somit verhindern kann, dass der Motor die physische Lenkung überdreht und somit kaputt macht.

Ich habe ein Gesamtpaket gefunden, Schrittmotor mit passendem Treiber.



Abbildung 2

https://cdn-reichelt.de/bilder/web/artikel_ws/A300%2FSBC-MOTO1_02.jpg?type=Product&



Abbildung 3

https://cdn-reichelt.de/bilder/web/artikel_ws/A300%2FSBC-MOTO1_01.jpg?type=Product&

6.1.4 Antrieb

Für den Antrieb habe ich ebenfalls zwei Optionen. Einen Schrittmotor oder einen DC-Motor. Ich habe mich für einen DC-Motor entschieden. Bei dem Antrieb ist es egal, auf welcher Position die Räder stehen. Ausserdem sind die DC-Motoren in der Regel schneller als die Schrittmotoren. Einen DC-Motor kann man ganz einfach ansteuern. Indem man mehr oder weniger Spannung anlegt, wird die Drehzahl höher oder tiefer. Das einzige Problem ist, dass ich die Drehrichtung nicht Software-technisch bestimmen kann. Aus diesem Grund habe ich noch einen Print mit 4 3-5V Relais bestellt, mit welchem ich eine kleine Schützensteuerung machen kann.

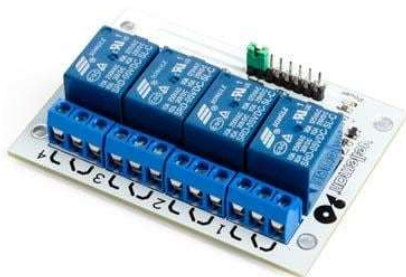


Abbildung 4

https://assets.brack.ch/images2/5/1/3/190075315/190075315_xxl.jpg

Beim Motor habe ich mich für den COM-Motor01 entschieden. Dieser hat ein aufgebautes Getriebe durch welches ich mir mehr Kraft erhoffe. Ich werde später entscheiden ob ich ihn mit oder ohne Getriebe betreiben möchte.



Abbildung 5

https://cdn-reichelt.de/bilder/web/artikel_ws/A300%2FCOM_MOTOR_RAD_03.jpg?type=Product&

Beim Testen hat sich herausgestellt, dass der DC-Motor zu wenig stark ist, um das Auto anzutreiben. deswegen habe ich mich dazu entschieden, noch einen zweiten Motor einzubauen. Jedoch wird es mit dem Platz unter dem Auto etwas knapp. Deswegen brauche ich einen möglichst kleinen DC Motor. Am besten mit einem Getriebe, so wird er stärker sein.

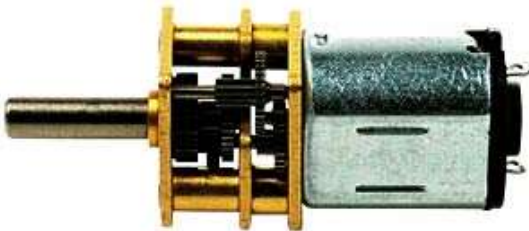


Abbildung 6

https://cdn-reichelt.de/bilder/web/artikel_ws/C160%2FGM32_3V_01.jpg?type=Product&

6.1.5 Lego-Auto

Ich habe mir überlegt, welche Voraussetzungen das Lego-Auto haben muss. Wichtig ist, dass das Auto gross ist, sodass ich die elektronischen Bauteile einfacher einbauen kann. Ausserdem soll das Auto nicht zu tief gelegen sein, sodass ich eventuell unter dem Auto noch das eine oder andere einbauen kann. Es wäre gut, wenn das Auto von Lego Technik ist, weil so nicht alles mit Steinen verbaut ist. Ausserdem ist es mir wichtig, dass das Auto schon ein intaktes Getriebe für die Lenkung hat. Leider gibt es nicht viele Autos mit diesen Anforderungen und ohne

bereits eingebauten elektrischen Antrieb. Das einzige welches ich gefunden habe, welches diesen Anforderungen entspricht, ist der Mercedes-Benz G500 Professional.



Abbildung 7

https://assets.brack.ch/images2/8/3/3/364800338/364800338_xx1.jpg

6.2 Zusammenbau Lego-Auto

Als erstes habe ich das Lego-Auto zusammengebaut, da dieses eine weniger lange Lieferfrist hatte als die elektronischen Bauteile.

Während dem Zusammenbau habe ich mich darauf geachtet, wo es gute Stellen gibt, bei welchen ich die Motoren einbauen kann. Diese habe ich gefunden und vermerkt.

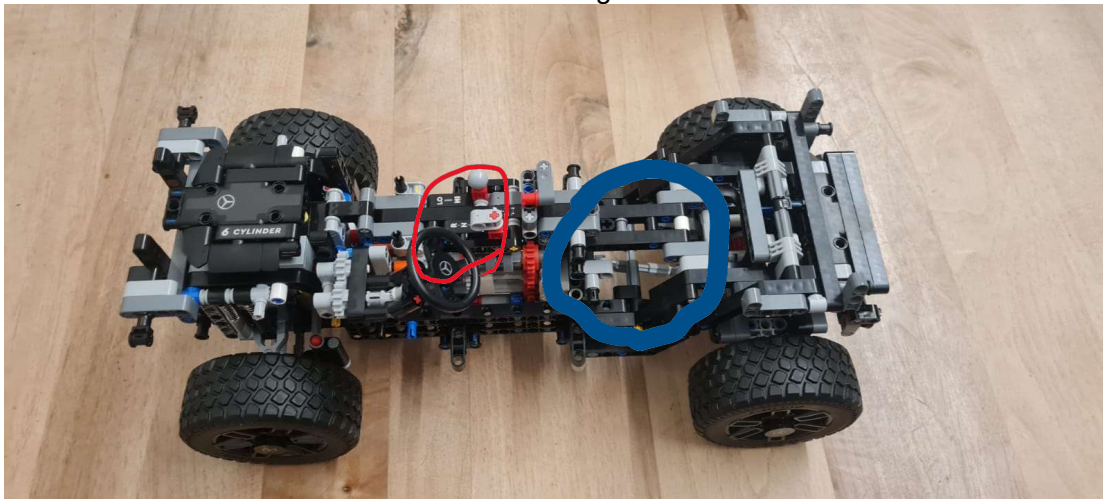


Abbildung 8

Mit blau habe ich markiert, wo ich den Antriebsmotor einbauen kann und mit rot, wo ich den Lenkmotor einbauen kann.

6.3 Programmieren

Bevor ich mit dem Programmieren beginnen konnte, musste ich meinen Raspberry PI Zero aufsetzen. Ich habe die Verbindung über einen Hotspot gemacht, für welchen ich die SSID Lego-Auto und das Passwort Lego-Auto verwendet habe. Somit kann jeder seinen Hotspot so benennen und das Auto später so steuern.

Ausserdem habe ich ein neues Codesys Programm (64 bit multi core) geöffnet.

6.3.1 Lenkung

Für die Lenkung brauche ich einen digitalen Servomotor. Die Stellung des Servomotors wird nach der Länge des Input Signales gestellt. Somit brauche ich ein Programm, mit welchem ich die Stellung des «Lenkrades» abfrage und somit die Länge des Ausgangssignales berechne.

- Stellung Links = 0.95ms
- Stellung Mitte = 1.5ms
- Stellung Rechts = 2.05ms

Es braucht immer einen abstand von 20ms zwischen den Signalen. Dies nennt man Pulsweitenmodulation (PWM).

Für die Lenkung werde ich es einfach machen, ich werde einen Knopf für links und einen für rechts machen. Ich werde einen Funktionsbaustein erstellen, in welchem die Impulszeiten definiert sind. Danach werde ich mit Hilfe von Funktionsbausteinen eine Schaltung entwickeln, mit welcher ich die Impulse so verschalte, dass sie auf ein Zeitrelais geführt werden können.

Als erstes erstelle ich eine Wahrheitstabelle, dass die LinksRechts-Mitte Schaltung richtig gemacht werden kann.

Taster Links (TL)	Taster Rechts (TR)	Links fahren	Rechts fahren	Gerade fahren
0	0	0	0	1
0	1	0	1	0
1	0	1	0	0
1	1	0	0	1

Somit können wir sagen:

- Links fahren= $TL \cdot \overline{TR}$
- Rechts fahren= $\overline{TL} \cdot TR$
- Gerade ahren= $\overline{TL} \cdot \overline{TR} + TL \cdot TR$

Dies habe ich so bei Codesys programmiert.

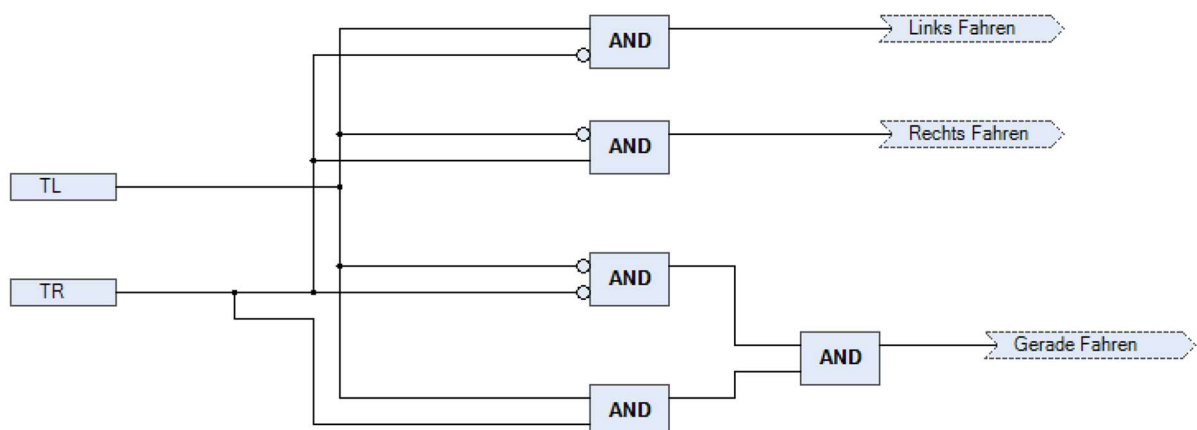
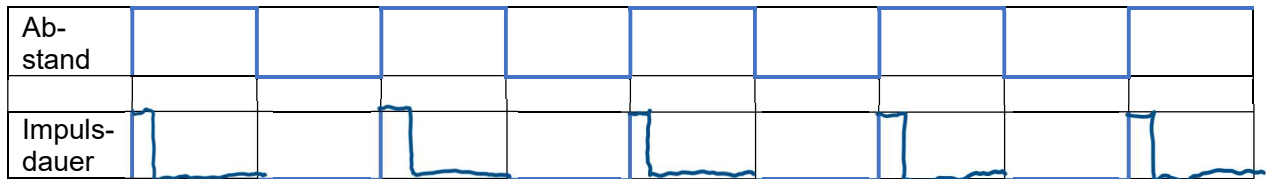


Abbildung 9

Jetzt muss ich die 3 Fahrten (Geradeaus, Links und Rechts) mit der vorgegebenen Impulsdauer und dem vorgegebenen Abstand zwischen den Impulsen verschalten.

Hierfür mache ich erst eine Skizze für eine Impulsdauer, und kann dies dann auf die drei Impulsdauern anwenden.



Somit werde ich für die Impulsdauer einen anzugverzögerten Baustein benutzen. Dessen Ausgang werde ich invertieren und mit einer «UND» Verknüpfung mit dem Abstand verschalten. Den Abstand werde ich mit zwei abfallverzögerten Bausteinen machen.

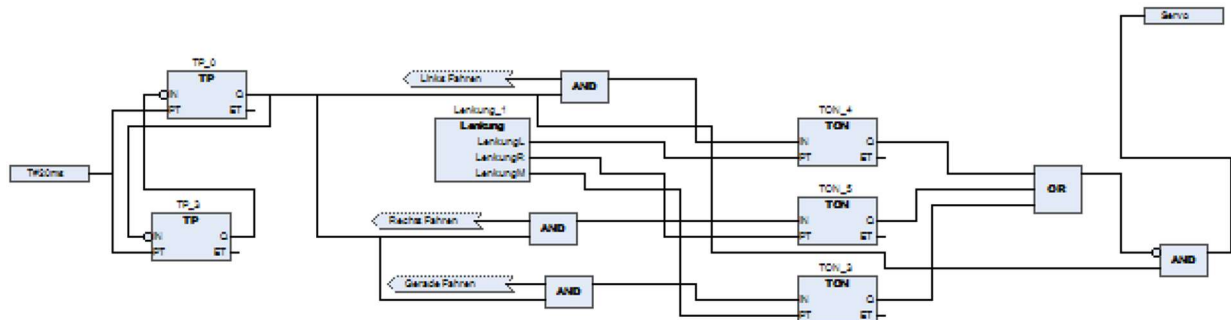


Abbildung 10

Beim Testen hat sich herausgestellt, dass der Rastperry PI und Codesys zu langsam sind um ein PWM Signal zu bearbeiten.

Jetzt habe ich noch einen Schrittmotortreiber bestellt um das PWM Signal zu umgehen. Die grosse Gefahr beim Schrittmotor ist, dass dieser keinen Anschlag hat und ich diesen softwaretechnisch einbauen muss.

Die Eingänge für das Programm bleiben gleich, ein Eingang für rechts drehen und ein Eingang für links drehen. Bei den Ausgängen, sind es mehr geworden. Ich brauche:

- Einen Enable Ausgang um den Schrittmotor zu bestromen.
- Einen Direction Ausgang um die Richtung anzugeben in welche er sich drehen soll
- Einen Pulse Ausgang, bei jedem Puls macht der Motor einen 3°schritt.

Für den Enable Ausgang werde ich einen Schalter machen um die Lenkung einzuschalten.

Wenn der Direction Ausgang auf 1 ist dreht der Motor nach Links. Also werde ich den Direction Ausgang direkt mit dem Linksdreh Eingang verknüpfen.

Für den Puls brauche ich einen Pulsgeber und eine Schaltung, mit welcher ich steuern kann, dass der Motor nicht zu weit nach links oder nach rechts dreht.

Den Pulsgeber werde ich vom PWM Signal Versuch übernehmen.

Um die Pulse zu begrenzen werde ich einen CTUD Baustein verwenden.

Gültigkeitsbereich	Name	Typ	Beschreibung
Eingang	CU	BOOL	Steigende Flanke: Erhöht cv um 1
	CD	BOOL	Steigende Flanke: Verringert cv um 1
	RESET	BOOL	TRUE: Setzt cv auf 0 zurück
	LOAD	BOOL	TRUE: Setzt cv auf den Startwert pv
	PV	WORD	Startwert für das Verringern / Obergrenze für das Erhöhen
Ausgang	QU	BOOL	TRUE wenn $CV \geq PV$
	QD	BOOL	TRUE wenn $CV = 0$
	CV	WORD	Aktueller Zählerwert

Abbildung 11

Beschreibung des Counters auf Codesys

Für PV werde ich die Anzahl Pulse, welche ich benötige um 180° zu drehen einsetzen. Diese sind $180^\circ / 0.18^\circ = 1000$ Pulse. Pulse=0 wird Stellung ganz links sein und Pulse=60 wird Stellung ganz rechts sein.

Mithilfe einer Wahrheitstabelle kann ich die Verschaltung der Eingänge definieren.

Um Fehler bei zufälliger Betätigung beider Richtungstasten zu verhindern, werde ich die für die Visualisierungstaster bei der Lenkung so beibehalten.

Links Fahren	Rechts Fahren	Puls	CU	CD
0	0	0	0	0
0	0	1	0	0
0	1	0	0	0
0	1	1	1	0
1	0	0	0	0
1	0	1	0	1
1	1	0	0	0
1	1	1	0	0

$CU = \text{Puls} * \text{Rechts Fahren} * \text{Links FahrenN}$

$CD = \text{Puls} * \text{Links Fahren} * \text{Rechts FahrenN}$

Da ich die Verschaltung $\text{Rechts Fahren} * \text{Links FahrenN}$ schon gemacht habe, muss ich diese nicht noch mal machen. So wird es nur noch:

- $CU = \text{Puls} * \text{Rechtsfahren}$
- $CD = \text{Puls} * \text{Linksfahren}$

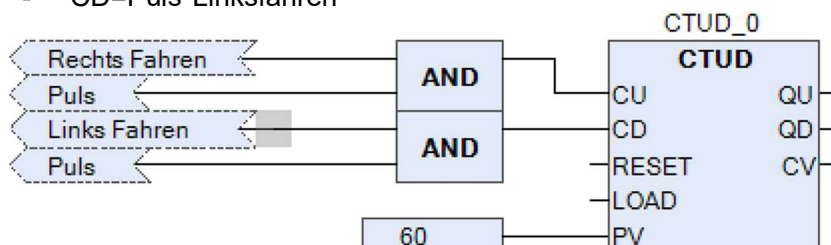


Abbildung 12

Jetzt kann ich die Schaltung machen, mit welcher ich den definitiven Puls an den Schrittmotor aus-gebe. Ich werde die beiden Ausgänge der AND Bausteine auf einen OR Baustein führen da es für den Schrittmotor egal ist, ob es jetzt ein Puls für links oder rechts ist. Danach werde ich die beiden

Ausgänge QU und QD des CTUD Bausteines je negieren und auf den AND Baustein vor dem jeweiligen CU und CD Eingang führen.

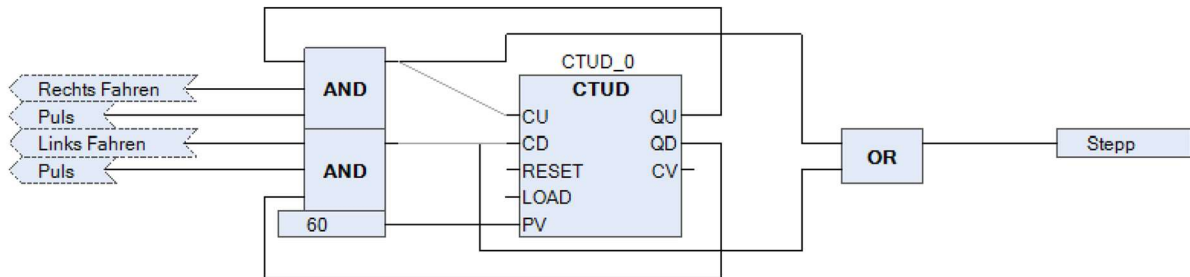


Abbildung 13

Ich habe einen ULN2003 Stepperdriver. Bei einem ULN2003 brauche ich nicht die einzelnen Steps und die Richtung, sondern habe 4 Eingänge, welche ich der Reihe nach bestromen muss um ihn zum Drehen zu bringen.

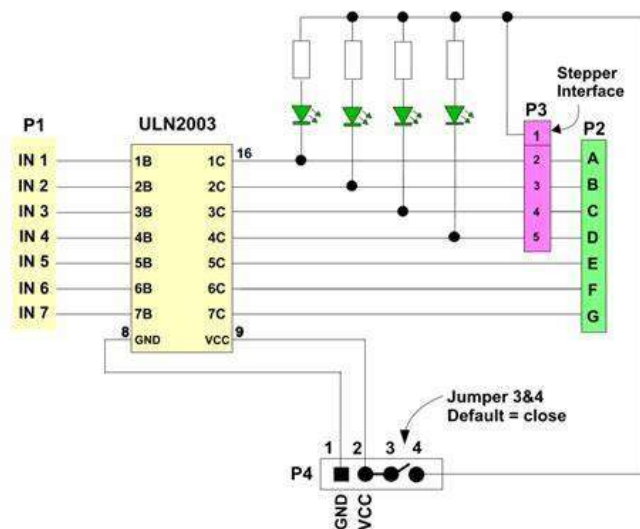


Abbildung 14

<https://th.bing.com/th/id/OIP.IQoH5UrhiRWFOemkyCfYhgHaGM?rs=1&pid=ImgDetMain>

Das heisst, ich muss bei IN1 anfangen und bis IN4 nach rechts drehen oder ich mache es umgekehrt und ich fahre nach Links.

Ich habe mir direkt gedacht, dass ich hier etwas mit rückstellbaren Countern machen kann. Ich habe 5 Counter genommen und der Reihe nach das Zählmaximum von 0 bis 4 genommen. Den Ausgang QU (Zählmaximum oben erreicht) habe ich mit dem negierten QU der darauf folgenden Zahl in einer AND Verknüpfung verschaltet und der Reihe nach an IN1-IN4 ausgegeben. Als Counter Puls habe ich den Puls für den Positionszähler genommen.

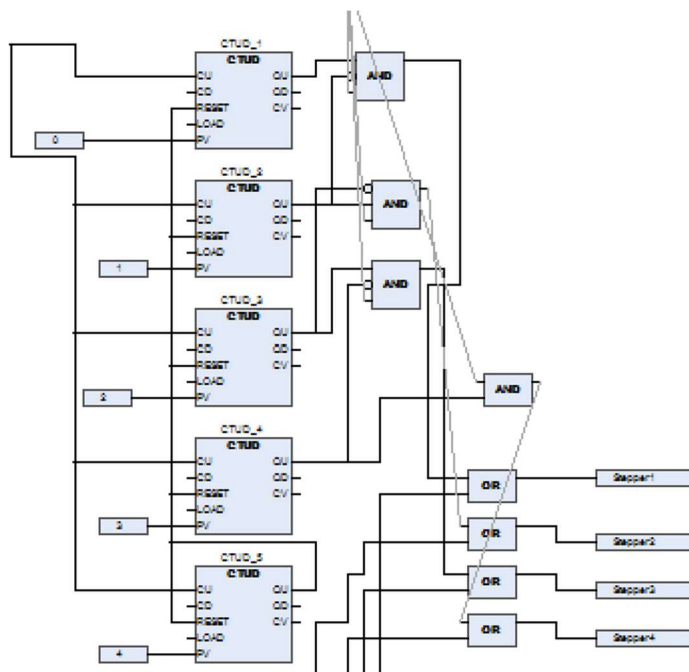


Abbildung 15

Dies hat so funktioniert, wie es soll. Jetzt habe ich dasselbe für die andere Richtung gemacht. Ich habe die gleiche Schaltung genommen, habe jedoch die Ausgänge verkehrt verschaltet.

Beim Testen hat sich herausgestellt, dass die Positionsüberwachung nicht exakt ist. Ich vermute, dass ich die Pulse so schnell eingestellt habe, da die Lenkung sonst noch langsamer ist als sie jetzt schon ist, dass der Treiber nicht genug schnell ist und ich deshalb einen Schrittverlust habe, welcher bei 180° etwa 10° ausmachen kann. Ich habe mich dazu entschieden, dass ich bei der Lenkung den Überdrehenschutz rausnehme und beim Beschrieb und auf der Visualisierung eine Warnung zu hinterlegen nicht zu überdrehen. Ausserdem werde ich beim Einbau schauen, dass ich den Motor nicht zu starr montiere, sodass eine Überdrehung nicht direkt fatale Folgen hat.

6.3.2 Antrieb

Für den Antrieb habe ich einen DC-Motor gewählt.

Die Anforderungen an den Antrieb sind:

- Vorwärts Fahren
- Rückwärts Fahren
- Beschleunigung variabel nach 2 Modi
- Geschwindigkeit variabel nach zwei Modi.

Hier für werde ich 3 Schalter benutzen. Einen Umschalter, mit welchem ich zwischen den zwei Modis «schnell» und «langsam» wechseln kann. Einen Taster fürs vorwärts fahren und einen Taster fürs rückwärtsfahren.

Somit habe ich 5 verschiedene Zustände (neutral=Stopp, vorwärts langsam=VL, vorwärts schnell=VS, rückwärts=RL langsam und rückwärts schnell=RS) welche anhand von vier Eingangssignalen gewechselt werden.

VF	RF	MS	ML	VS	VL	RS	RL	N
0	0	0	0	0	0	0	0	1
0	0	0	1	0	0	0	0	1
0	0	1	0	0	0	0	0	1
0	0	1	1	0	0	0	0	1
0	1	0	0	0	0	0	0	1

0	1	0	1	0	0	0	1	0
0	1	1	0	0	0	1	0	0
0	1	1	1	0	0	0	0	1
1	0	0	0	0	0	0	0	1
1	0	0	1	0	1	0	0	0
1	0	1	0	1	0	0	0	0
1	0	1	1	0	0	0	0	1
1	1	0	0	0	0	0	0	1
1	1	0	1	0	0	0	0	1
1	1	1	0	0	0	0	0	1
1	1	1	1	0	0	0	0	1
1	1	1	1	0	0	0	0	1

- $VS = VF \cdot RFN \cdot MS \cdot MLN$
- $VL = VF \cdot RFN \cdot MSN \cdot ML$
- $RS = VFN \cdot RF \cdot MS \cdot MLS$
- $FL = VFN \cdot RF \cdot MSN \cdot ML$

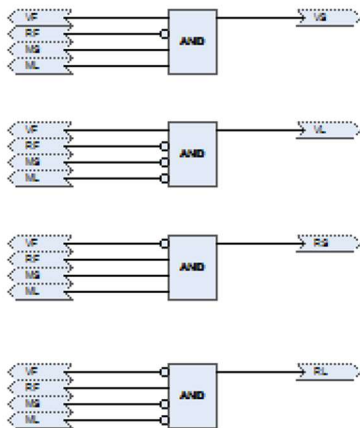


Abbildung 16

Für Neutral werde ich die Gleichung mithilfe eines KV-Diagrammes vereinfachen, da diese Gleichung sehr lange werden würde.

Neutral		VF	VF/RF	RF
	1	1	1	1
MS	1	0	1	0
MS/ML	1	1	1	1
ML	1	0	1	0

$$\text{Neutral} = VFN \cdot RFN + VF \cdot RF + MSN \cdot MLN + MS \cdot ML$$

Somit habe ich die Schaltungsgleichungen, wie ich die Eingänge verschalten muss.



Abbildung 17

Für den langsamen Modus brauche ich noch einen Taktimpuls. Für diesen habe ich wieder meine «Pulsmaschine» von der Lenkung genommen. Jetzt muss ich noch die Ausgänge verschalten.

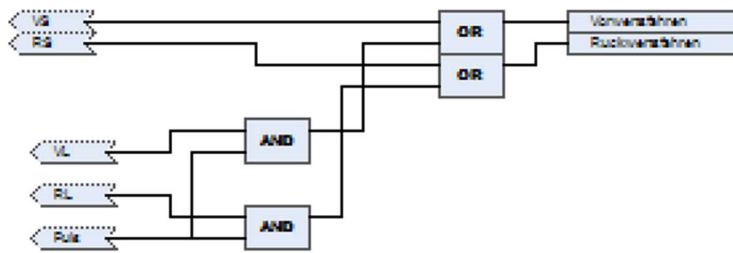


Abbildung 18

6.3.3 Visualisierung

Für die Visualisierung habe ich mir gedacht, weniger ist mehr und vor allem übersichtlicher. Ich brauche für die Lenkung einen Taster Links und einen Rechts und für die Antriebssteuerung einen vorwärts und einen rückwärts. Ausserdem brache ich noch einen Schalter für schnell und langsam.



Abbildung 19

Als Beschriftung werde ich die jeweiligen Eigenschaften anschreiben und den Warnhinweis der Lenkung.



Abbildung 20

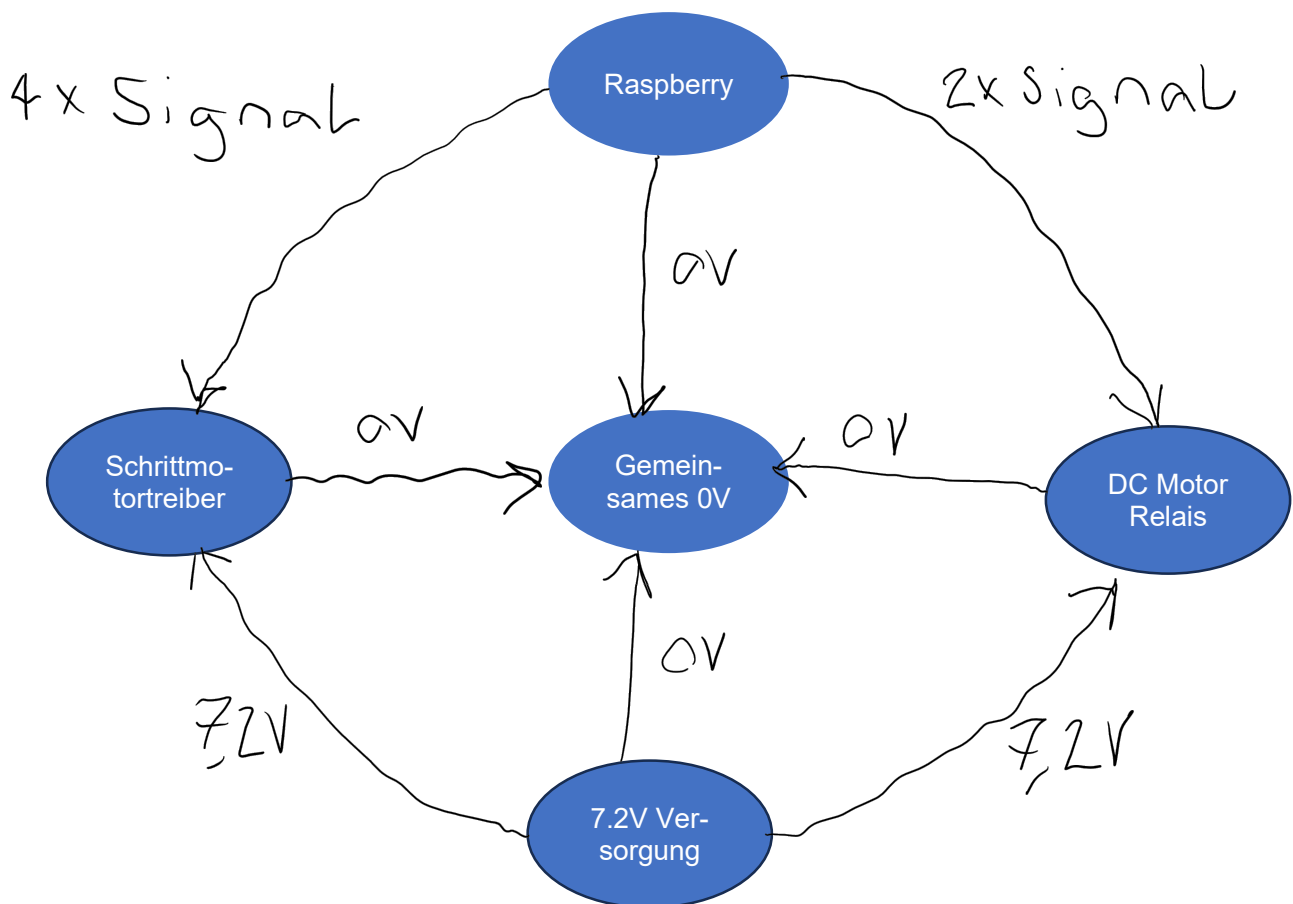
Während des Testens hat sich herausgestellt, dass man nur einen Knopf auf einmal betätigen kann. Deswegen habe ich die Antriebster zu Druckschaltern gewechselt, so dass man gleichzeitig fahren und steuern kann.

6.4 Zusammenbau

Den Zusammenbau werde ich in zwei Schritten durchführen. Im ersten Schritt werde ich alle elektronischen Bauteile zusammenhängen und im zweiten Schritt werde ich diese im Lego-Auto einbauen.

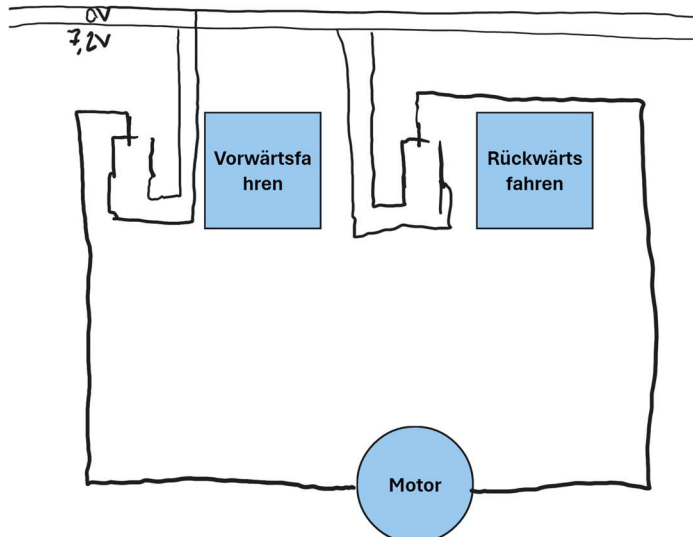
6.4.1 Elektronischer Zusammenbau

Ich habe alle elektronischen Bauteile aufgelistet, um zu sehen, welches Bauteil welche Spannungsversorgung benötigt. Die Spannungsversorgung des Raspberrys werde ich nicht einzeichnen, weil diese eine Powerbank ist.



Für den Schrittmotor ist es einfach diesen anzuhängen. Einfach die 6 Drähte für die 7.2V Spannungsversorgung und die 4 Signale anhängen und es läuft.

Beim Antrieb habe ich zwei Signale, eines fürs Vorwärtsfahren und eines fürs Rückwärtsfahren. Jetzt muss ich das mit den Relais verschalten, sodass der Plus Eingang beim Motor beim retourfahren ein Minus bekommt und umgekehrt. Ich habe mir mal ein paar Überlegungen gemacht.



Somit habe ich bei Stillstand auf beiden Motoreingängen 0V. Dies ändert sich, sobald ein Relais gezogen wird, dann wird auf einen Eingang 7.2V gegeben. Somit kann ich die Drehrichtung des Motors einfach mit dem gewünschten Signal wechseln und den Motor zum Drehen bringen. Für die Signal- und Spannungsversorgungsanschlüsse braucht es auch keine Skizze, weil diese klar definiert sind auf dem Relais.

6.4.2 Elektronik in Lego-Auto

Um die Lenkung einzubauen, habe ich die Stange demontiert, welche durchs Dach ging um das Auto von Hand zu steuern und habe auf diese Stange den Schrittmotor gebaut.



Abbildung 21

Ich habe ihn bewusst mit Kabelbindern eingebaut, so kann er sich selber noch ein bisschen überdrehen, wenn die Lenkung am Anschlag ist. So kann ich bei einer Überdrehung direkte Schäden verhindern.

Für den Antrieb habe ich die Verbindungsstange zwischen der Vorderachse und der Hinterachse demontiert und an der Halterung für die Hinterachse den Antriebsmotor montiert.

Beim Testen hat sich herausgestellt, dass ein einzelner DC-Motor zu wenig Kraft aufbringen kann, um das Auto zum Rollen zu bringen. Also habe ich das Getriebe für die Hinterachse auch noch demontiert um diese Reibungswiderstände weg zu nehmen und habe bei der Verbindung zur Vorderachse noch einen DC-Motor eingebaut, welcher die Vorderachse auch noch antreibt. Den Motor für die Vorderachse habe ich parallel zum Hinterachsenmotor angeschlossen.

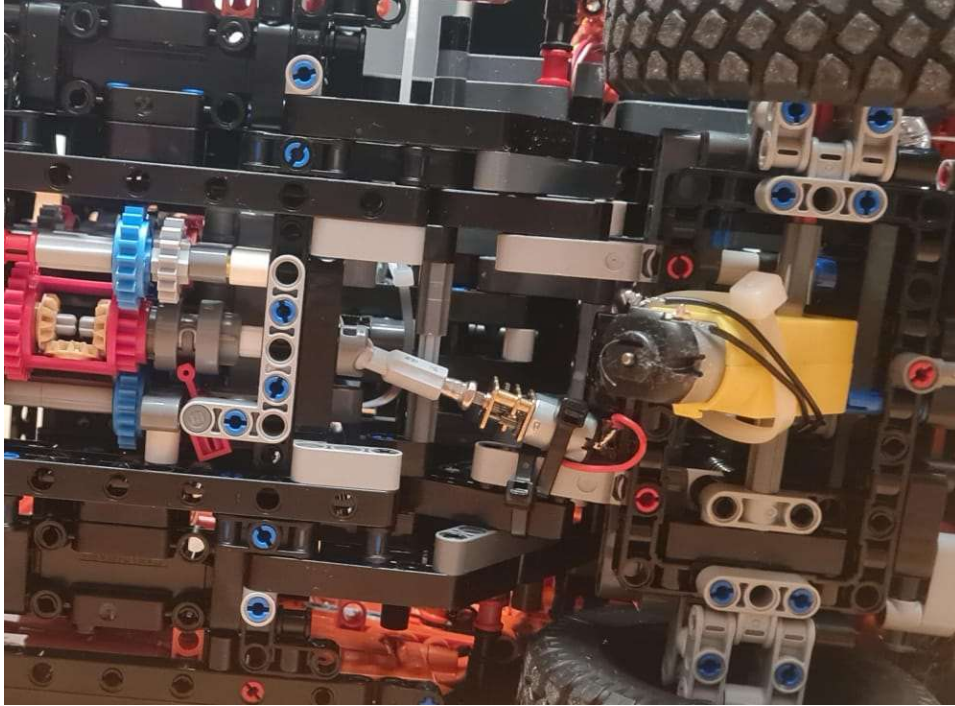


Abbildung 22

Da der Motor an der Hinterachse ein wenig nach unten schaut, werde ich in die Bedienungsanleitung einfließen lassen, dass man nicht über Schwellen fahren soll.

Die restlichen elektronischen Bauteile habe ich in den Kofferraum gelegt, und werde ich mithilfe von schwarzem Karton vor Blicken schützen. Die zwei Anschlusskabel für die Spannungsversorgungen habe ich auf die Rückbank gelegt, auf welche man das Akkupaket und die Powerbank legen kann, wenn man sie eingesteckt hat.

6.5 Bedienungsanleitung

Jetzt wo alles programmiert und eingebaut ist, kann ich eine Bedienungsanleitung machen, welche ich jemandem zusammen mit dem Auto und den Spannungsversorgungen in die Hände drücken kann und er kann das Auto selbstständig bedienen. Ausserdem sollte er auch die Akkus selbstständig wechseln können.

Als Erstes werde ich eine Liste machen, was alles wichtig ist:

- Warnhinweise
- Akkus laden
- Akkus in Auto einbauen
- Verbinden mit Handy um Auto zu steuern

Zu den Warnhinweisen gehören die beiden, welche ich beim Programmieren und beim Einbauen erwähnt habe. Lenkung nicht überdrehen, da kein Endanschlag und nicht über Schwellen fahren. Diesen zweiten Hinweis habe ich noch dahingehend abgeändert, dass nur auf glatten Böden gefahren werden soll.

Ausserdem werde ich noch darauf aufmerksam machen, dass man das Auto vorsichtig tragen soll, dass nichts abfällt.

Das Laden der Akkus wird ein kleiner Posten sein. Diese sollen mit der mitgelieferten Ladestation und dem mitgelieferten USB-C Ladekabel geladen werden.

Ich werde Fotos von den einzusteckenden Kabeln einfügen und wo man die Akkus platzieren kann.

Die Verbindung mit dem Handy wird die umfangreichste Beschreibung. Mit der Einrichtung des richtigen Hotspots und der Verbindung mit der Onlinevisualisierung.

Während der Verständlichkeitstests hat sich herausgestellt, dass iPhone-User den Hotspot für die Verbindung nicht erstellen können, da sie ihre SSID nicht so ändern können wie sie es müssten. Dies habe ich natürlich auch in die Bedienungsanleitung einfließen lassen.

Bedienungsanleitung Matteos G-Wagon

Warnhinweise:

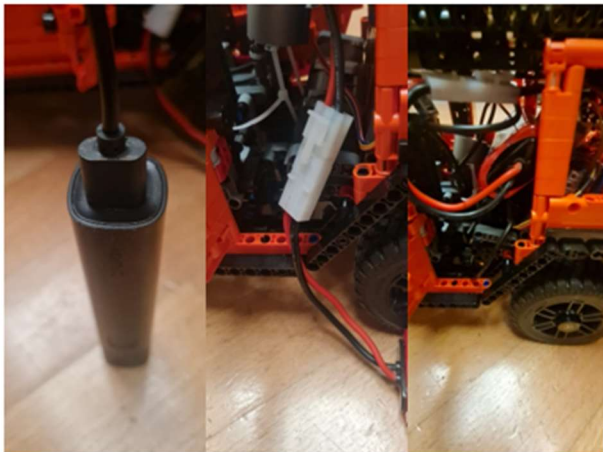
- Lenkung hat keinen Endanschlag! Lenkung nicht überdrehen.
- Das Auto nur auf Glatten Böden ohne Rampen und Schwellen benutzen, da der Antriebsmotor tief hängt!
- Nach benutzen Akkus ausstecken und Laden!!

Laden der Akkus:

- Zum Laden der Akkus das die Mitgelieferte Ladestation für das Akkupaket und das Mitgelieferte USB C Ladekabel verwenden.

Einbauen der Akkus:

- Akkupaket und Powerbank bei den dafür vorgesehenen Steckern auf dem Rücksitz einstecken und auf den Rücksitz legen.



Verbinden:

- Mobiler Hotspot auf dem Handy einrichten (kann mit iPhone nicht gemacht werden, da die SSID nicht verändert werden kann.)
 - o SSID=Lego-Auto
 - o Passwort=Lego-Auto
- Warten bis sich „Lego-Auto“ mit dem Hotspot verbunden hat.
- Auf Lego-Auto klicken und lesen was für eine IP-Adresse das Lego-Auto hat.
- Im Internetbrowser des Handys oder eines mit dem Hotspot verbundenen Gerätes die IP-Adresse mit der zusätzlichen Endung „:8080/webvisu.htm“ eingeben. Zum Beispiel „192.168.232.102:8080/webvisu.htm“.

Viel Spaß mit Matteos G-Wagon

Abbildung 23

6.6 Testen

Ich habe die meisten Funktionen Schritt für Schritt während dem Programmieren oder Zusammenbauen getestet, jedoch sind dies die finalen Tests, mit welchen ich einen Teil der Funktion abgeschlossen habe.

6.6.1 Lenkung Programmiert

Das Wichtigste für die Lenkung ist, dass sie präzise 180° fährt. Wenn dies nicht der Fall ist oder wenn sich der 0° und der 180° Punkt verschiebt, könnte es sein, dass die Lenkung beschädigt wird. Ich habe auf dem Motor zwei Striche gemacht, bei 0° und bei 180° und bin 10 mal in beide Software-Anschläge gefahren, mit dem Ergebnis, dass sich die beiden Punkte von der ersten Fahrt verschoben haben.

6.6.2 Lenkung Eingebaut

Die Lenkung wurde ohne Softwareendanschlag eingebaut. Deswegen kann ich nur testen, ob ich es richtig eingebaut habe. Ich habe die Lenkung 10 mal von links nach rechts drehen lassen und es hat immer geklappt. Aufgrund der kleinen Steps, welche der Steppermotor macht, ist die Lenkung eher langsam. Aber sie funktioniert!

6.6.3 Antrieb Programmiert

Den Antrieb werde ich testen, indem ich alle möglichen Kombinationen teste.

Modus Schnell	Vorwärts fahren	Rückwärts fahren	Soll	Ist
0	0	0	Neutral	Neutral
0	0	1	Rückwärts Langsam	Rückwärts Langsam
0	1	0	Vorwärts Langsam	Vorwärts Langsam
0	1	1	Neutral	Neutral
1	0	0	Neutral	Neutral
1	0	1	Rückwärts Schnell	Rückwärts Schnell
1	1	0	Vorwärts Schnell	Vorwärts Schnell
1	1	1	Neutral	Neutral

Somit haben alle Funktionen so funktioniert wie sie müssen.

6.6.4 Antrieb Eingebaut

Da ich die richtige Funktion des Vorwärts- und Rückwärtsfahren schon getestet habe, muss ich nur noch testen ob das Auto rollt und ich den Motor richtig rum angeschlossen habe. Das Auto bewegt sich keinen Millimeter. Wahrscheinlich hat der Motor zu wenig Kraft.

2. Versuch:

Ich habe noch einen zweiten Motor eingebaut. Jetzt bewegt sich das Auto. Die Drehrichtung hat auch gepasst. Das einzige was ein wenig unsauber ist, ist, dass die Relais mechanisch und nicht elektronisch sind. Wenn ich auf langsam fahre hört man das Klappern der Relais, da dieses Signal getaktet ist.

Als letztes habe ich noch getestet, wie lange ich mit dem Auto auf dem schnellen Modus fahren kann, ohne dass der Akku aufgibt. Da die Vorgabe 15 Minuten sind, habe ich den Test nach 20 Minuten abgebrochen.

Somit habe ich alle Funktionen des Autos getestet und es hat alles funktioniert.

6.6.5 Bedienungsanleitung

Ich wollte meine Familie als Testpersonen für die Verständlichkeit der Bedienungsanleitung nehmen, jedoch hat sich herausgestellt, dass man die SSID bei iPhones nicht so ändern kann wie bei einem Android Handy. Meine Eltern und meine beiden Schwestern haben alle ein iPhone! Somit haben sie es mit meinem Handy ausprobiert und haben es souverän geschafft das Auto zu bedienen. Ich wollte jedoch noch wissen, ob es auch mit anderen Handys geht. Deswegen habe ich noch 2 Kollegen mit Android Handys gebeten es zu versuchen und es hat sich herausgestellt, dass ich die IP-Adresse nicht so hinschreiben darf. Jedes Handy gibt dem Lego-Auto eine eigene IP-Adresse und somit muss man diese in den Hotspot Einstellungen selber nachschauen. Dies habe ich so nachgetragen.

Das grösste Problem war, dass die Testpersonen die Warnhinweise nicht gelesen haben.

7 Projektabschluss

7.1 Projektüberwachung

Ich konnte mich bis zum 18.10. an meinen Projektablaufplan halten. An diesem Tag habe ich den ersten Meilenstein nicht erfüllt. Ich habe es bis dahin nicht geschafft, den Servo zu programmieren. Von da an habe ich nochmal angefangen mit einer erneuten Recherche welchen Motor ich als Alternative verwenden kann und musste praktisch von neuem beginnen. Somit hat sich mein ganzer Plan weiter nach hinten verschoben. Ich habe die Zusammenführung des Lego-Autos und der Elektronik noch getestet, als ich die Dokumentation abgabebereit haben wollte. Somit habe ich die letzten 3 Meilensteine mit je etwa einer Woche Verspätung erreicht und musste gewisse bereits erarbeitete Meilensteine erneut erarbeiten.

Schlussendlich habe ich es mit 3-4 Nachtschichten geschafft, das Projekt 3 Tage vor Abgabetermin fertig zu haben und die Dokumentation einen Tag vor Abgabetermin.

Was die Einhaltung des Projektablaufplanes und die Termineinhaltung der selbstgewählten Meilensteine angeht werde ich in einem nächsten Projekt definitiv an mir arbeiten. Was das Endprodukt angeht, bin ich zufrieden mit dem was ich gemacht habe.

7.2 Evaluation der Zielerreichung

Das Lego-Auto funktioniert und ich bin glücklich, dass ich es geschafft habe. Jetzt werde ich schauen ob ich alle im Pflichtenheft definierten Endergebnisse mit den dazugehörigen Erfolgskriterien erreicht habe.

- Es liegt eine nachvollziehbare Dokumentation vor, welche rechtzeitig abgegeben wird.
 - o Am 11. November um 16:00 Uhr ist die Dokumentation im PDF-Format im Extranet hochgeladen.
 - o Die Dokumentation wird streng nach dem 4 Phasenmodell erarbeitet.
 - o Die Dokumentation kann von 5 Personen mit elektrotechnischem Hintergrund durchgelesen werden, welche weniger als 3 Rückfragen aufgrund von Unklarheiten haben.

Die Dokumentation ist am 10.11.2024 abgabebereit und wird am selben Abend noch im Extranet hochgeladen und dem Diplomelehrer per E-Mail zugesandt.

Wenn man das Inhaltsverzeichnis anschaut, sieht man, dass diese Arbeit streng nach dem 4 Phasenmodell erarbeitet wurde. Erst die Projektinitialisierung, dann die Projektplanung, als nächstes die Projektrealisierung und als letztes den Projektabschluss.

Ich habe die Arbeit 3 Elektronikern, einem Maschinenbauer und einem Elektroinstallateur gegeben zum Durchlesen. Sie haben gesagt meine Gedankengänge seien nachvollziehbar und klar. Die einzigen Fragen welche ich beantworten musste, waren Interessensfragen.

- Das Auto soll 15 Minuten Fahren können.
 - o Der Akku muss so gross gewählt werden, dass man mit dem Auto im «Sport-Modus» und bei höchster Geschwindigkeit mindestens 15 Minuten herumfahren kann.

Der Modus heisst jetzt nicht mehr Sport, jedoch habe ich es wie bereits im Testprotokoll dokumentiert probiert und konnte mit einer Akkuladung im schnellen Modus länger als 20 Minuten fahren.

- Es soll eine allgemeinverständliche Bedieneroberfläche und Bedienungsanleitung vorliegen.
 - o Die Bedienungsanleitung kann an 5 Personen, die nicht technisch versiert sind, kostenlos abgegeben werden und sie können sich ohne Rückfragen mit dem Auto verbinden, den Akku wechseln, den Akku laden und das Auto bedienen.

Ich habe das Auto beim Verständlichkeitstest 6 Personen mit den Akkus und der Bedienungsanleitung in die Hand gedrückt und sie haben es ohne Rückfragen geschafft, es zum Fahren zu bringen. Allfällige Unklarheiten habe ich in die Bedienungsanleitung einfließen lassen. Jedoch konnten sich iPhone User nicht mit ihrem eigenen Handy mit dem Auto verbinden, da man bei iOS die SSID des Hotspots nicht so frei ändern kann wie bei Android.

- Die Komponenten sollen fachgerecht montiert und installiert sein.
 - o Die Komponenten sind so eingebaut, dass sichtbare und berührbare Komponenten aller Art, von Beschädigung und abreißen geschützt sind.

Ich habe die meisten Bauteile in den Kofferraum gelegt und den Kofferraum mit schwarzem Karton ausgekleidet. Die einzigen Bauteile welche man wirklich sehen kann sind die Akkus, die dazugehörigen Anschlusskabel und die Motoren. Der einzige Motor welcher ein kleines Problem darstellt, ist der hintere Antriebsmotor, da dieser ein wenig tief nach unten ragt und somit bei Schwellen oder bei grobem Teppich den Boden berührt. Um dies zu vermeiden habe ich bei der Bedienungsanleitung einen Warnhinweis hinterlegt. Somit kann während dem Bedienen bei geschlossener Tür nichts passieren.

- Das Auto soll mit der Vorderachse gesteuert werden und mit der Hinterachse angetrieben werden.
 - o Der Motor zur Steuerung der Vorderachse soll so eingebaut werden, dass sich das Lenkrad immer noch dreht.
 - o Die Hinterachse wird mit je einem Motor pro Rad angetrieben. Das Auto soll mit der Vorderachse gesteuert werden und mit der Hinterachse angetrieben werden.

Der Motor zur Steuerung der Lenkung an der Vorderachse wurde so eingebaut, dass sich das Lenkrad im Auto immer noch dreht.

Bei diesem Punkt habe ich eine Abweichung. Ich konnte aus Platzgründen nicht pro Rad einen Motor einbauen. Ich habe für die Hinterachse einen Motor eingebaut und für die Vorderachse einen Motor eingebaut. Ich musste die Vorderachse auch antreiben, da ich nur mit einem Motor an der Hinterachse zu wenig Power habe.

- Es sollen verschiedene Modi möglich sein.
- Es muss mindestens 2 Modi geben (ECO und Sport) welche einen Unterschied in der Beschleunigung und der Maximalgeschwindigkeit vorweisen.

Ich habe einen Schalter in der Steuerung eingebaut, mit welchem ich vom langsamen zum schnellen Modus wechseln kann. Dies funktioniert auch, im schnellen Modus ist das Auto schneller und beschleunigt schneller als im langsamen Modus.

- Die Programmierung muss nachvollziehbar sein.
 - o Die Programmierung wird so dokumentiert, dass man ohne in das Programm zu schauen, weiss wie dieses aussieht.
 - o Das Programm wird wo möglich mit Grafiken ausgearbeitet.

Ich habe alle Programmteile mit PrintScreens in die Arbeit eingefügt, somit sieht man wie das Programm schlussendlich aussieht.

Ich habe, wo es nützlich war, die Erarbeitungsmethoden welche ich an der TEKO gelernt habe verwendet und diese auch dokumentiert. Wenn es nicht nützlich war habe ich meine Gedanken aufgeschrieben und so erklärt, was ich mir dabei überlegt habe.

Somit habe ich die meisten Endergebnisse erfolgreich umgesetzt. Ich hatte zwei Abweichungen, bei dem Einbau der Motoren und bei der fachmännischen Installation der Betriebsmittel.

7.3 Reflexion Weg zum Ziel

Meinen Weg zum Ziel hatte ich am Anfang als sehr zielstrebig und lösungsorientiert eingestuft. Jedoch habe ich beim Zusammenbauen der elektronischen Variante das erste mal gemerkt, dass ich die Recherche ausführlicher hätte machen sollen. Fragen wie zum Beispiel welche Ausgangssignale kann der Raspberry PI ausgeben oder mit welcher Geschwindigkeit er das Programm bearbeitet habe ich mir nicht gestellt. Ausserdem hätte ich mir von Anfang an schon die Frage stellen müssen, wie stark und gross der Motor sein muss um das Auto zu bewegen. Mit dieser mässigen Recherche habe ich einen Motor ausgewählt, welcher ich nicht auf die Weise bedienen konnte, wie ich es wollte. Somit musste ich mich an einem Punkt selber hinterfragen und habe mir gesagt, ich setzte auf die Motorenart welche mit geläufig ist und welche ich auch verstehe und auch schon damit gearbeitet habe. Diese Entscheidung war eine gute und würde ich wieder so machen. Wenn man schon seit Tagen am selben Problem 10 bis 20h investiert hat, sollte man sich spätestens fragen, schaffe ich es oder muss ich mir eingestehen, dass dies entweder nicht geht oder meine Fähigkeiten übersteigt. Ich habe mich dazu entschieden, meine Motorenart zu ändern und von da an ist das Projekt wieder ins Rollen gekommen. Jedoch hatte ich keine Chance mehr, die verpassten Meilensteine wieder aufzuholen oder sonst einen späteren Meilenstein wieder Termingerecht zu erreichen.

Lange Rede kurzer Sinn, hätte ich die Recherche am Anfang gewissenhafter und gründlicher durchgeführt, hätte ich schon von Beginn weg die richtigen Bauteile bestellen können und hätte die Arbeit Schritt für Schritt nach dem Projektablaufplan erarbeiten können und hätte nicht in der Hälfte der Zeit wieder von Anfang beginnen müssen. Jedoch würde ich die Entscheidung, die Lenkung neu zu überdenken und zu bestellen ein nächstes mal wieder genau gleich treffen. Dies hat mein Projekt gerettet, auch wenn ich die Entscheidung sicher schon früher hätte treffen müssen.

7.4 Lessons learnt

Ich habe in den letzten 8 Wochen viel Zeit und Energie investiert, um dieses Projekt zu realisieren im Zuge der Diplomarbeit. Ich habe schlussendlich ein Endergebnis auf welches ich stolz bin. Und natürlich habe ich wieder viel gelernt. Ich habe gelernt, auch wenn man in einer schwierigen Phase im Projekt ist, dass man nicht den «Kopf in den Sand stecken», sondern extra nochmal mehr Gas geben soll. Wenn man sich im Kreis dreht, manchmal eine komplette Neubeurteilung des Projektes auch ein Fortschritt sein kann und nicht nur ein Rückschritt.

Ich habe viele Lektionen gelernt in dieser Diplomarbeit. Was wahrscheinlich eine der wichtigsten Lektionen war, ist, dass die Recherche und die Definition der Betriebsmittel eine der wichtigsten Arbeiten bei einem Projekt sind. Dass man da lieber 3 Tage zu viel einsetzen soll als einer zu wenig. Ich habe mir gedacht, ich bestelle was meines Wissens am besten passt für mein Auto, ohne mir grosse Gedanken darüber zu machen, ob es dann in der Zusammenarbeit mit der restlichen Hardware auch funktioniert. Was mich freut ist, dass ich Lehren von früheren Arbeiten umsetzen konnte und somit aus Fehlern gelernt habe. Ich habe zum Beispiel die Erarbeitung der Steuerung gut dokumentiert und mit den passenden Grafiken untermauert. Wenn ich beim Betrachten des Problems einen Geistesblitz hatte, wie es funktionieren könnte, habe ich es versucht detailliert zu beschreiben. Die letzten Arbeiten welche ich hatte, waren alles Partnerarbeiten. Die letzten drei habe ich mit dem selben Partner gemacht. Wir hatten uns ziemlich gut ergänzt, er hat sich gerne um die Initialisierung, die Planung und den Abschluss gekümmert und ich habe mich gerne um die Realisierung gekümmert. Somit war es für mich fast Neuland alles selber zu machen und es hat mir auch ein wenig an Erfahrung gefehlt. Jetzt, wenn ich ein Projekt von A-Z alleine durchführen musste, habe ich weitere Erfahrungen gesammelt. Für ein nächstes Projekt wird es mir wahrscheinlich leichter von der Hand gehen. Wenn ich wiederum eine Weiterbildung machen würde, würde ich bei einer Teamarbeit mehr darauf schauen, dass ich in allen Bereichen Erfahrung sammeln kann und nicht nur in jenen, welche mir liegen und ich gerne mache.

Zusammengefasst:

Ich hatte viele Erfolge, ich hatte viele ernüchternde Momente und ich habe vieles gelernt. Ich habe wieder mal die Freude gespürt ein Projekt in Richtung Elektronik und Programmieren zu realisieren und freue mich wenn ich wiederum etwas realisieren möchte.

Und ich hoffe, dass die Lehren welche ich in diesem Projekt mitnehmen kann, helfen mir für zukünftige Projekte, Fehler zu vermeiden.

7.5 Ausblicke

Wenn mich nach dem TEKÖ Abschluss die Entwicklerfreude packt, habe ich schon die eine oder andere Idee, wie ich das Lego-Auto noch weiter ausbauen könnte. Ich würde den Antriebsmotor der Hinterachse durch einen kleineren, gleich starken Motor austauschen. Dadurch könnte ich mit ihm auch über Schwellen und Hindernisse fahren wie es sich gehört. Ausserdem würde ich noch LEDs als Scheinwerfer einbauen und eine Kamera, um zu sehen wo man hinfährt wäre auch noch eine coole Idee.

Somit habe ich die eine oder andere Idee, wenn ich wieder ein wenig Zeit habe an dem Lego-Auto zu basteln nach meinem Jobwechsel vom klassischen «Stromer» weg.

8 Eigenständigkeitserklärung

Der Verfasser bestätigen mit seiner Unterschrift, dass die vorliegende Arbeit selbstständig, ohne fremde Hilfe und ohne Benutzung anderer als die angegebenen Hilfsmittel erstellt wurde.

Die aus fremden Quellen (einschliesslich elektronischer Quellen) direkt oder indirekt übernommenen Inhalte sind als solche kenntlich gemacht.

Die Arbeit ist in gleicher oder ähnlicher Form noch nicht vorgelegt worden.

Unterschrift:

Datum/Ort:



Matteo Tonazzi

Geuensee 10.11.2024

8.1 Schlusswort

Wieder ist eine Arbeit fertig. Ich habe ein Endergebnis, auf welches ich stolz sein kann und wieder viel Zeit und Energie investiert habe.

Ich freue mich, die Arbeit abzugeben. Ebenfalls freue ich mich mein Endergebnis bei der Präsentation vorzustellen und zu Präsentieren.

Zum Schluss möchte ich noch Danke sagen! Der TEKO, dass ich die Möglichkeit hatte an dieser Schule viel zu lernen. Allen Dozenten, für die Geduld welche sie alle aufgebracht haben um uns das Wissen weiterzugeben. Meinen Klassenkollegen, welche den Schulalltag mit Sprüchen und Gesprächen aufgefrischt haben und in Gruppenarbeiten und gemeinsamen Lösungssuchen den Fokus wieder zurückgebracht haben.

Meiner Familie und meinen Freunden, welche mir bei Prüfungsphasen und Schlussphasen von Arbeiten viel Unterstützung, Aufmunterung und Nachsicht entgegengebracht haben.

Ich freue mich jetzt, mit einem prallgefüllten Rucksack an neuem Wissen in der Arbeitswelt weiterzugehen und bin gespannt, wohin es mich als nächstes verschlägt.

9 Verzeichnisse

9.1 Abbildungsverzeichnis

Abbildung 1.....	19
Abbildung 2.....	19
Abbildung 3.....	20
Abbildung 4.....	20
Abbildung 5.....	21
Abbildung 6.....	21
Abbildung 7.....	22
Abbildung 8.....	22
Abbildung 9.....	23
Abbildung 10.....	24
Abbildung 11.....	25
Abbildung 12.....	25
Abbildung 13.....	26
Abbildung 14.....	26
Abbildung 15.....	27
Abbildung 16.....	28
Abbildung 17.....	28
Abbildung 18.....	29
Abbildung 19.....	29
Abbildung 20.....	30
Abbildung 21.....	32
Abbildung 22.....	33
Abbildung 23.....	35

Ich habe die Abbildungen, welche nicht von mir sind oder nicht von mir erstellten Produkten sind mit dem dazugehörigen Link deklariert.

10 Anhang

10.1 Projektstatusberichte

Damit der Diplomlehrer immer auf dem neusten stand ist wie es um meine Diplomarbeit steht, habe ich jede Woche einen Statusbericht geschickt, auf welchem ich geschrieben habe, was ich gemacht habe und was als nächstes ansteht.

Projekt: Ferngesteuertes Lego-Auto

Stautsbericht: Nr.1 24.09.2024

Projektleiter Matteo Tonazzi	Projektziele Lego Auto mit DC-Motoren bestücken und zu einem Mit dem Handy steuerbaren Ferngesteuerten Auto programmieren mithilfe eines Raspberry PI.		Verteiler • Urs Gloggner		
Gesamt-beurteilung	Projektverlauf   	Projektklima   	Termine   	Risiken   	Ressourcen   
Tendenz					
Aktueller Projektstand • Der PSP und der PAP wurden erstellt. • Erste Recherchen bezüglich Elektronischer Bauteile wurden gemacht. • Das Lego-Auto welches ich verwenden möchte wurde definiert.			Was läuft gut? • Ich habe ein Ideales Lego-Auto gefunden. Was läuft nicht gut? • Es ist schwierig die passenden elektronischen Komponenten zu finden. • Diese Woche gab es einen Terminverzug bezüglich Statusbericht.		
Geplante nächste Schritte / getroffene Massnahmen • Bestellen Lego Auto • Bestellen mehrerer Optionen von Elektronischer Bauteile. • Raspberry PI aufsetzen.					

Projekt: Ferngesteuertes Lego-Auto

Stautsbericht: Nr. 2 01.10.2024

Projektleiter Matteo Tonazzi	Projektziele: Lego Auto mit DC-Motoren bestücken und zu einem Mit dem Handy steuerbaren Ferngesteuerten Auto programmieren mithilfe eines Raspberry PI.		Verteiler • Urs Gloggner		
Gesamt-beurteilung	Projektverlauf   	Projektklima   	Termine   	Risiken   	Ressourcen   
Tendenz					
Aktueller Projektstand • Projektablaufplan Überarbeitet • Lego Auto Recherchiert und bestellt • Elektronische Bauteile Recherchiert			Was läuft gut? • Ich habe das Passende Auto gefunden Was läuft nicht gut? •		
Geplante nächste Schritte / getroffene Massnahmen • Elektronische Bauteile bestellen • Lego Auto zusammenbauen • Elektrische Variante definieren und Prov. Zusammenbauen					

Projekt: Ferngesteuertes Lego-Auto

Stautsbericht: Nr. 3 08.10.2024

Projektleiter Matteo Tonazzi	Projektziele: Lego Auto mit DC-Motoren bestücken und zu einem Mit dem Handy steuerbaren Ferngesteuerten Auto programmieren mithilfe eines Raspberry PI.		Verteiler • Urs Gloggner		
Gesamt- beurteilung	Projektverlauf   	Projektklima   	Termine   	Risiken   	Ressourcen   
Tendenz	⇒	⇒	⇒	⇒	⇒
Aktueller Projektstand • Elektronische Bauteile bestellen • Lego Auto zusammenbauen • Raspberry PI Zero aufgesetzt		Was läuft gut? • Das Lego Auto ist so gebaut, dass es keine Probleme geben wird die Motoren einzubauen. Was läuft nicht gut? •			
Geplante nächste Schritte / getroffene Massnahmen • Elektrische Variante definieren und Prov. Zusammenbauen. • Mit der Programmierung beginnen.					

Projekt: Ferngesteuertes Lego-Auto

Stautsbericht: Nr. 4 15.10.2024

Projektleiter Matteo Tonazzi	Projektziele: Lego Auto mit DC-Motoren bestücken und zu einem Mit dem Handy steuerbaren Ferngesteuerten Auto programmieren mithilfe eines Raspberry PI.		Verteiler • Urs Gloggner		
Gesamt- beurteilung	Projektverlauf   	Projektklima   	Termine   	Risiken   	Ressourcen   
Tendenz	⇒	⇒	⇒	⇒	⇒
Aktueller Projektstand • Elektrische Variante definieren und Prov. Zusammenbauen. • Mit der Programmierung beginnen.		Was läuft gut? • Rasperry Pi ging gut zum Aufsetzen. Was läuft nicht gut? • Steuerung für Servomotor ist komplizierter als gedacht.			
Geplante nächste Schritte / getroffene Massnahmen • Programmierung abschliessen • Programmierung Testen • Einbau Elektronik in Lego- Auto					

Projekt: Ferngesteuertes Lego-Auto

Stautsbericht: Nr. 5 22.10.2024

Projektleiter Matteo Tonazzi	Projektziele: Lego Auto mit DC-Motoren bestücken und zu einem Mit dem Handy steuerbaren Ferngesteuerten Auto programmieren mithilfe eines Raspberry PI.		Verteiler • Urs Gloggner		
Gesamt- beurteilung	Projektverlauf   	Projektklima   	Termine   	Risiken   	Ressourcen   
Tendenz					
Aktueller Projektstand • Beim Programmieren und Testen stehen geblieben • Neuer Raspberry PI bestellt da beim Löten verbrannt. • Begonnen Recherche für alternativen Servo, da Digital nicht geht.			Was läuft gut? Was läuft nicht gut? • Raspberry PI Verbrannt beim Löten • Digitale Servosteuerung geht nicht (Warscheinlich Codeys oder Raspberry PI zu langsam.		
Geplante nächste Schritte / getroffene Massnahmen • Analoger Servo Recherche und bestellen • DC Motorentreiber Recherche und bestellen.					

Projekt: Ferngesteuertes Lego-Auto

Stautsbericht: Nr. 6 29.10.2024

Projektleiter Matteo Tonazzi	Projektziele: Lego Auto mit DC-Motoren bestücken und zu einem Mit dem Handy steuerbaren Ferngesteuerten Auto programmieren mithilfe eines Raspberry PI.		Verteiler • Urs Gloggner		
Gesamt- beurteilung	Projektverlauf   	Projektklima   	Termine   	Risiken   	Ressourcen   
Tendenz					
Aktueller Projektstand • Analoger Servo Recherche • Steppermotor mit Treiber bestellt für Lenkung • DC Motorentreiber Recherche und bestellen. • Lenkung programmiert • Antrieb begonnen zu programmieren. • Neue Raspberry PI bestellt.			Was läuft gut? Die neuen Motoren und Raspberrys sind angekommen. Was läuft nicht gut? • Durch die Verzögerung, mit den Motoren habe bin ich im Terminplan in Rückstand.		
Geplante nächste Schritte / getroffene Massnahmen • Antrieb Programmieren • Steuerung Testen • Elektronik in Lego-Auto einbauen • Legoauto mit eingebauter Elektronik testen.					

Projekt: Ferngesteuertes Lego-Auto

Stautsbericht: Nr. 7 05.11.2024

Projektleiter Matteo Tonazzi		Projektziele: Lego Auto mit DC-Motoren bestücken und zu einem Mit dem Handy steuerbaren Ferngesteuerten Auto programmieren mithilfe eines Raspberry PI.		Verteiler • Urs Gloggner	
Gesamt- beurteilung	Projektverlauf   	Projektklima   	Termine   	Risiken   	Ressourcen   
Tendenz					
Aktueller Projektstand • Antrieb programmiert • Steuerung getestet • Elektronik in Lego-Auto einbauen • Legoauto mit eingebauter Elektronik begonnen zu testen			Was läuft gut? Die Steuerung und der Lenkung und des DC-Motors funktionieren, wie sie sollen. Was läuft nicht gut? • Durch die Verzögerung, mit den Motoren bin ich im Terminplan in Rückstand.		
Geplante nächste Schritte / getroffene Massnahmen • Bedienungsanleitung Schreiben • Tests beenden • Dokumentation bereinigen • Arbeit abgeben!					