Dokumentation des Unipraktikums

Gruppe iAtePi

October 12, 2018

BERGISCHE
UNIVERSITÄT
WUPPERTAL

Contents

Chapter 1

Einleitung

1.1 Motivation

Im Moment sind selbstfahrende Autos so ziemlich das technische Thema. Universitäten rund um den Globus forschen an verschiedensten Technologien, welche das selbstfahrende Auto braucht. Ein Autohersteller nach dem anderen kauft entweder Startups zu dem Thema auf, oder investiert in eine eigene Forschungsabteilung. Als sich nun die Möglichkeit eines Praktikums im Rahmen des Elektrotechnikleistungskurses an der Bergischen Universität Wuppertal anbot, fiel die Wahl des Themas leicht. Das dieses Praktikum nicht nur eine gute Möglichkeit ist um den aktuellen technischen Entwicklungen nachzueifern, zeigt sich schon an den verschiedenen Aufgaben die zu erledigen sind. Im Laufe des Projektes müssen nämlich Motoren mithilfe von H-Brücken und Pulsweitenmodulation gesteuert, ein Programm entworfen und ein Raspberrypi in Betrieb genommen werden. Neben dem technischen Wissen, was erlernt und angewendet werden muss (und sich praktischerweise relativ gut mit dem Abiturstoff deckt), wird hier auch ein Einblick in den Universitätsalltag erlangt.

1.2 Zielsetzung

Es gibt viele mögliche Aufgaben, die man im Rahmen des Themas lösen könnte. Besonders interessant sind hier vor allem das finden von einem und das Einparken auf einem Parkplatz, sowie das Teilnehmen am Straßenverkehr. Letzteres entfällt wegen der Schwierigkeit des Simulierens von anderem Verkehr. Deshalb ist die hier gewählte Aufgabe das finden und befahren einer Parklücke in einem Straßenähnlichen Setting. Genauer gesagt wird das Auto eine Strecke entlangfahren und mit Hilfe von Sensoren eine Parklücke in die es hineinpasst identifizieren und dann dort einparken. Dies wird innerhalb von 16 Einheiten über einen Zeitraum von sieben Wochen passieren. Ein weiteres Ziel ist es das für die Vollendung des Projektes nötige Wissen für alle Gruppenmitglieder verständlich zu bearbeiten und die Gruppe damit für das Abitur vorzubereiten.

Chapter 2

Plannung

2.1 Ausblick

Wenn man das Projekt Zukünftig fertigstellen würde, könnte man einige Eigenschaften hinzufügen wie zum Beispiel ein Bildsensor. Durch den Bildsensor kann man Aufnahmen von zweidimensionalen Abbildern aus Licht erstellen, auf dem elektrischem oder mechanischem Wege. Eine andere Alternative zum Bildsensor wäre ein Lidar, ein Gerät das Laserstrahlen ausstrahlt und die reflektierten Strahlen mit einem Sensor misst. Die letzte Alternative wäre ein Ultraschallsensor. Der Ultraschallsensor erzeugt Schallwellen, nimmt diese dann auf und berechnet somit die Entfernung zum Hindernis. Generell könnte man eine Karosserie um das Gestell bauen um die Elektronik zu schützen oder verstecken und um das Fahrzeug selbst zur Geltung zu bringen. Eine weiter Möglichkeit wäre das man bessere Reifen anbringt, damit das Auto schneller fahren kann oder je nach Reifentyp und Motor kleine Gebirge und Oberflächen bezwingen kann. Eine schöne Ergänzung für das Auto wäre eine Lenkung, sodass man das Auto fernsteuern könnte, wenn man es sich wünschen würde. Eine feine Ergänzung wäre, wenn man den Antrieb programmieren würde, um in bestimmten Situationen mit dem Auto besser fahren zu können zum Beispiel: auf Eis, Schotter (Offroad) und Neigungen. Kurz gesagt das man ein ABS-System entwickelt. Zuletzt könnte man noch eine Achsenlenkung erstellen um die Lenkung selbst zu verbessern, um das einparken zu vereinfachen und effizienter machen.

2.2 Programmierung

2.2.1 Einleitung

Bei der Programmierung des Projekts wurde ein objektorientierter Lösungsansatz gewählt. Dies wurde einerseits wegen der Thematisierung der Objekt-Orientierten-Programmierung (OOP) im Informatikunterricht und andererseits wegen dem

einfachen Bezug dem man hier zwischen realen Objekten (Sensor, Motor) und zugehörigen Klassen herstellen kann gewählt. Außerdem ist die OOP einer der Paradigmen, welche in der Realität benutzt werden.

2.2.2 Python

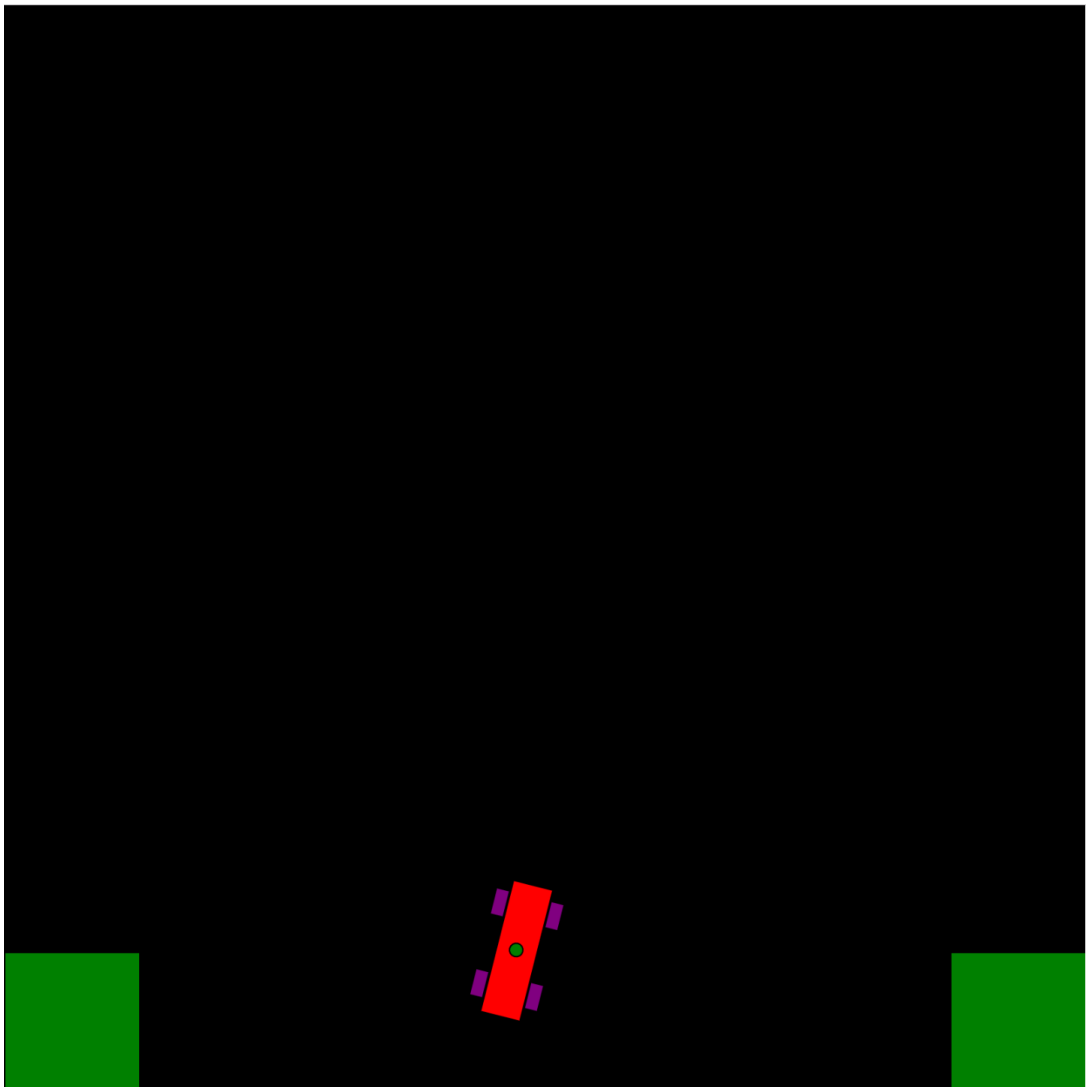
Als Programmiersprache wurde Python gewählt, da diese Sprache im Informatikunterricht verwendet wird, sie für Laien leicht verständlich ist und mehrere Personen in der Gruppe bzw. dem nahen Umfeld haben, welche diese Sprache bereits gut können. Außerdem unterstützt Python die OOP, wodurch die Sprache auch die Paradigmenwahl zulässt. Da sie eine beliebte und weitverbreitete Sprache ist unterstützt sowohl das aktuelle Image das auf dem Pi ist, als auch der Hersteller der Sensoren die Sprache. Auch kann man sagen, dass die Sprachwahl es ermöglicht Code, welcher schon frsuher von einzelnen Mitgliedern geschrieben wurde, wieder zu benutzen. Ein Nachteil ist hier, dass Python nicht alle Konzepte der OOP unterstützt (zum Beispiel Abstrakte Klassen, Interfaces/Traits).

2.2.3 Programmstruktur

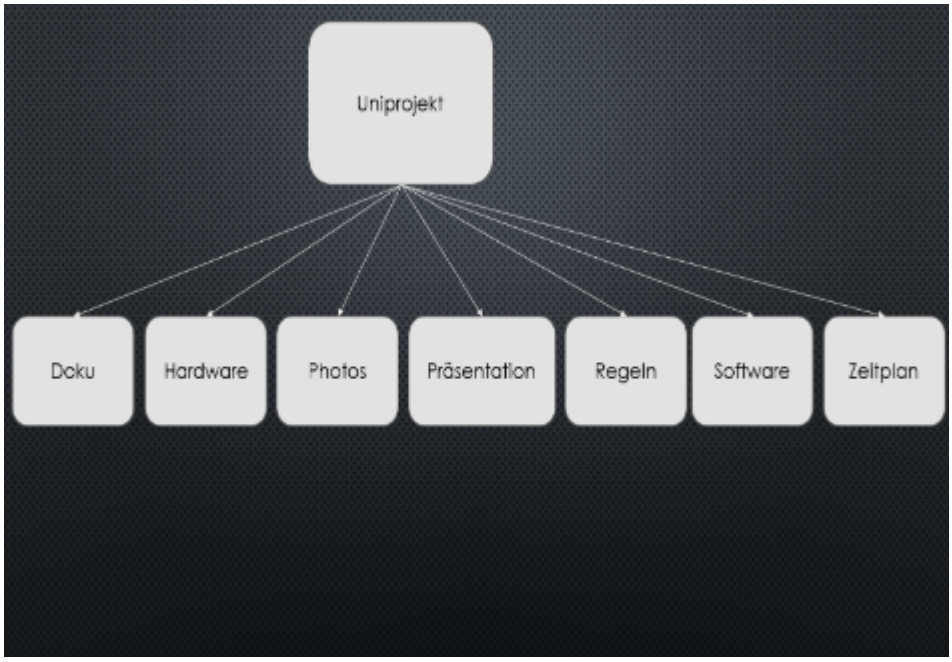
Jedes große Bauteil bekommt eine Klasse zugewiesen. Die wichtigen Bauteile sind hier: die Sensoren, die Motoren, die Motortreiber und der Raspberrypi. Für die Sensoren gibt es drei Klassen: Einmal eine abstrakte Klasse Sensor, welche definiert, dass jeder Sensor ein Signal auslesen(get_value) und die Bezeichnung des Pins ausgeben(get_location) kann. Es gibt zwei Klassen, namentlich LightSensor und DistanceSensor, welche von Sensor erben, und jeweils die Methode zum Signalauslesen, sowie den Konstruktor überschreiben.

2.2.4 Simulation

Nach dem die Realisierung des praktischen Teils nicht erfolgreich war, wurde die Informatische Leistung auf eine Simulation in Javascript fokussiert, damit wenigstens die Programmierung präsentiert werden kann. Diese Simulation benutz P5.js, was eine Javascriptbibliothek ist. In dieser Simulation soll ein Auto in eine Parklücke, welche durch zwei Hindernisse begrenzt ist, einparken. Es gibt hier momentan eine Klasse für das Auto (Car) und eine Klasse für die Hindernisse (Obstacle). Das Hauptskript findet sich in der Datei sketch.js. Später würde man auch eine Klasse für die Intelligenz des Autos (Agent) finden. Im jetzigen Zustand wird die Bewegung des Autos durch einen Geschwindigkeitsvektor und einen Beschleunigungsvektor simuliert. Außerdem wird die Ausrichtung des Autos durch die Ausrichtung des Geschwindigkeitsvektors bestimmt. In der aktuellen Version der Simulation ist der Lenkvorgang beim Einparken verfügbar, sodass für eine erste funktionsfähige Version nur noch Kollisionsdetektion und das Benutzen von Sensoren um Kollisionen zu verhindern fehlt.



2.2.5 Dateistrukturen



Alle Dateien wurden auf Github hochgeladen. Die Struktur, welche gewählt wurde ist im obigen Bild dargestellt und in diesem Text genauer erläutert. Der erste Unterordner trägt den Namen Doku. In diesem werden alle Dateien für die Dokumentation gespeichert, der Ordner ist jedoch noch mal in vier Unterordner unterteilt. Diese Ordner sind: erstens Abgabe, in welchem alle Dateien für diese Dokumentation zu finden sind, zweitens Grundwissen, in welchem Dateien sind, die alle in der Gruppe befähigen sollen, jede Aufgabe eigenständig zu lösen, drittens Quellen, in welchem alle Quellen aufgeführt sind und viertens Tagesdokus, in welchem alle Tagesdokumentationen sind. Der zweiter Unterordner Hardware enthält Datenblätter und Messungen in jeweils einem eigenen Unterordner. Es ist hier anzumerken, dass nicht alle Bauteile ein Datenblatt und Mes75sungen haben. Der Unterordner Software beinhaltet alle eigenen Programme und Skripte. Alle Dateien wurden in die Ordner Endprogramm, Simulation, Testen und Tools einsortiert. Der Ordner Endprogramm sollte das Programm, was am Ende auf dem Pi läuft, beherbergen. Das wären, im Falle Fertigstellung des Programmes, eine Datei pro Klassengruppe. Alles was im Unterordner Simulation befindet ist für die Simulation, welche Html und Javascript benutzt. In dem Ordner Test befindet sich Python-Skripte, welche während der Entwicklung des Autos zum Testen einzelner Funktionen (wie z.B. der Reichweite der Abstandssensoren) benutzt werden. Der Ordner Regeln beinhaltet Dateien, welche Regeln festhalten, die die Zusammenarbeit in der Gruppe erleichtern sollen. Der Ordner Präsentation beinhaltet unterschiedliche Versionen der Präsentation. Im Ordner Zeitplan befindet sich immer der aktuelle Zeit-

plan samt Anwesenheitsliste. Alle Fotos sind in dem Ordner Photos um die Dokumentation und die Präsentation zu vervollständigen. Durch diese Struktur sollen Dateien immer leicht zu finden sein. Außerdem wird durch diese Struktur die Abgrenzung von einzelnen Aufgabengebieten einfacher, wodurch sich alle Beteiligten leicht in jede Unteraufgabe eindenken können sollen.

2.3 Technologien

2.3.1 Github

Die Gruppe benutzt Github als Versionskontrollsystem. Github bietet sich hier an, weil einerseits git auf dem Raspberrypi ist, wodurch man immer die aktuellste Version des Programmes auf dem Pi haben können. Außerdem erleichtert Github das gleichzeitige Arbeiten, sodass mehrere Leute gleichzeitig programmieren, und die Programme schlussendlich hochladen können. Ein weiteres Plus ist, dass Github eine große Anzahl an letzten Versionen speichert, wodurch auch spät erkannte Fehler leicht überwunden werden können. Als letztes Argument für diese Technologie ist zu sagen, dass sie quasi als Dokumentation für alles was auf Github hochgeladen wird dienen kann. Da man jeden Schritt nachvollziehen kann.

2.3.2 Slack

Slack-Placeholder

Slack ist generell eine Kommunikations Software für Unterhaltung. Es können Channels eröffnet werden die man in unterschiedliche Themen unterteilen kann, je nach Kunde Team oder Projekt. Jeder Benutzer oder jedes Team kann je bei Bedarf verschiedene Channels betreten oder verlassen ganz einfach ohne Komplikationen. Bei jeder neuen Gruppe wird automatisch ein Bot (KI) erstellt der in das System von Slack einführt und Befehle oder Abstimmungen etc. in der Gruppe selbst führen/ausführen kann. In Slack kann man auch Audio- und Videochats durchführen für kurze Besprechungen oder Präsentationen. Wie man die Channels unterteilt ist jeder Gruppe selbst überlassen man kann sie für konzentrierte Diskussionen und Informations austausche benutzen oder nur für das Teilen von Dateien jeglichen Typs. Slack hat es geschafft sich mit vielen Tools und Services zu integrieren, dadurch kommen aus abisolierten Posteingängen die Dateien und Nachrichten zu den Team Channels und andersrum. Natürlich ist Slack auch abgesichert, sodass nichts verloren geht und oder was geklaut wird. Die Entwickler arbeiten ganze Zeit an der Sicherheit von Slack damit sie auf dem höchsten Niveau ist.

Benutzung

Diese Gruppe benutzte Anfangs Slack für Dateienspeicherung und Kommunikation. Mittlerweile wird auf Slack nur noch kommuniziert. Jedoch werden

die wichtigeren Dateien und generelle Textdokumente auf Github hochgeladen. Da Slack sowie Github als App herunterladbar sind werden diese auch mobil genutzt, um wenn man unterwegs ist auf dem neusten Stand zu sein.

2.3.3 LaTeX

Für die Tagesdokumentationen und die Hauptdokumentation wird LaTeX verwendet. Dies ist einerseits eine Referenz auf die echte Welt, in welcher LaTeX oft (im wissenschaftlichem und im privaten) benutzt wird. Andererseits soll von den Vorteilen profitiert werden. Die Vorteile liegen in darin, dass man einfacher alle Kapitel, Sektionen und Formeln variable gleichzeitig ändern können. Außerdem ist LaTeX eine alternative zu Word, wobei man viele der üblichen Probleme von Word umgeht. Beispiele wären hier zum Beispiel Verschiebung von Text bei dem Einfügen von Bildern, welches man in LaTeX deutlicher definieren muss.

2.3.4 Hilfsprogramm

Im Rahmen der Dokumentation gab es immer wieder Schwierigkeiten, da man in LaTeX einige Zeichen nicht einfach so schreiben darf, wenn sie entweder nicht von der Kodierung unterstützt werden (Beispiel: ä, Ü) oder schon eine Andere Belegung haben (Bsp.: _). Damit das schreiben der Dokumentation möglichst schnell passiert wurde deshalb von ein Skript geschrieben, welches alle diese problematischen Zeichen durch die LaTeX-Schreibweise ersetzt. Das Skript ist hier jedoch nur eine temporäre Lösung, weshalb man zum Beispiel den Pfad der Latex Datei nicht an das Skript weitergibt, sondern den Pfad innerhalb einer Variablen des Skriptes schreibt.

2.4 Elektrotechnik

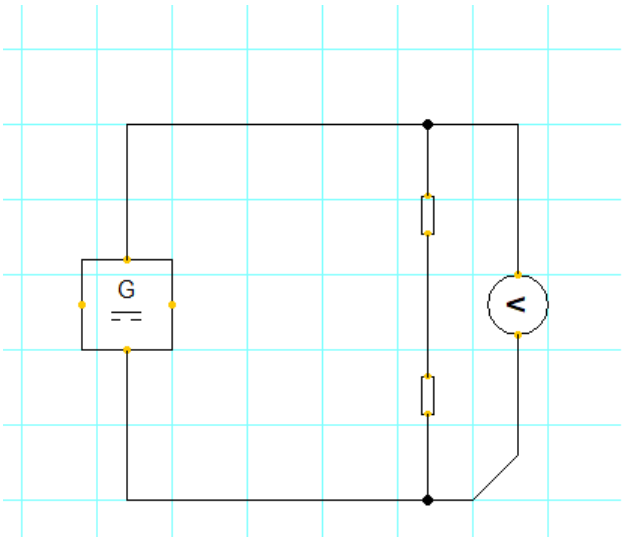
2.4.1 Grundlagen

Umgang mit Elektronik

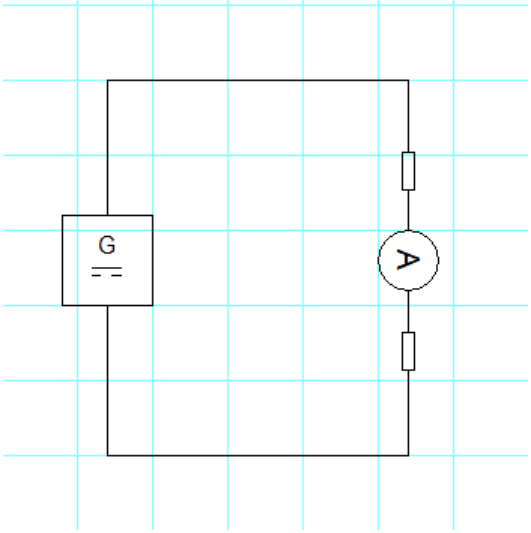
Hier würde ein Text stehen, wäre er abgegeben worden.

Messen

Wenn man eine Spannung messen will sollte man zuerst einmal das Multimeter in die Schaltung parallel einbauen und dann kann man die Spannung messen



Wenn man einen Strom messen will sollte man zuerst einmal das Multimeter in die Schaltung in Reihe einbauen somit kann man dann den Strom messen



PWM

PWM steht für PulsWeitenModulation bzw. Pulse-Width-Modulation. Diese wird zum Beispiel genutzt um durch den digitalen Ausgangswert eines Microcontrollers ein analoges Gerät (z.B. einen Motor) anzusteuern. Dies funktioniert indem innerhalb einer Periode der Dauer T für eine bestimmte Zeit T_{ein} eine Spannung anliegt die für an steht (also das high ist) und dann für eine bestimmte Zeit T_{aus} eine Spannung anlieht die für aus steht (also das low ist). Die Spannung die damit zum Beispiel bei einem Motor anliegt ist durch die Formel

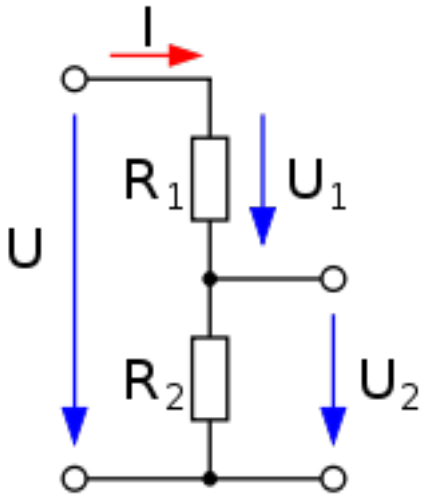
$U_m = U_{ein} * \frac{T_{ein}}{T}$ gegeben, wenn U_{aus} eine Spannung von 0V ist. Dies wäre benutzt worden um die H-Brücken und damit die Motoren anzusprechen.

AD/DA Wandler

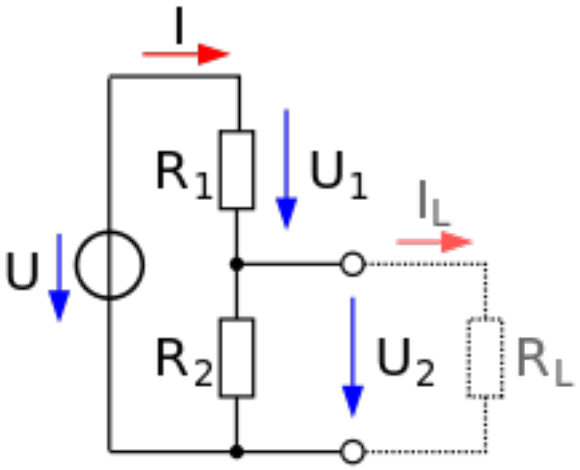
Spannungsteiler

In der Elektrotechnik kommt es oft vor, dass man zu hohen Spannungen hat um ein Bauteil zu betreiben. Eine Methode die oft angewendet wird ist der Spannungsteiler. Hier gibt es zwei grundsätzlich unterschiedliche Methoden: den unbelasteten Spannungsteiler und den belasteten Spannungsteiler.

Bei dem unbelasteten Spannungsteiler sind einfach nur zwei Bauteile in Reihe geschaltet, wovon eins das Bauteil und eins ein Widerstand ist. Durch die normalen Formeln der Reihenschaltung kann man hier die Aufteilung der Spannung berechnen: $R1 = \frac{U_1}{U_2} * R_2$.



Bei dem belasteten Spannungsteiler wird ein weiterer Widerstand zu dem ersten parallelgeschaltet. Dadurch wird der Gesamtwiderstand der Schaltung kleiner, was zu einem größeren Gesamtstrom führt. Außerdem wird die Spannung am Bauteil größer und die Spannung an den parallelen Widerständen kleiner.



Diese Schaltungen können benutzt werden um bei anderen Bauteilen als Zwischenschritt zu dienen. Dies könnte zum Beispiel dafür dienen den Output vom Pi zu verringern bevor man die Spannung an eine H-Brücke legt.

H-Brücke

Hier hätte ein nicht abgegebener Text gestanden.

Abstandssensor

Um das Projekt richtig umsetzen zu knnen, werden Abstandssensoren gebraucht um den Aufprall von dem Auto auf Hindernisse zu vermeiden. Fr diese Projekt wurden die Abstandssensoren die zu dem GrovePi passen ausgewhlt. Im Folgenden soll die Funktionsweise von den Abstandssensor erlutert werden. Diese Sensoren mssen eingestellt werden, bevor man sie benutzen kann. Im Idealfall wrde man diese Einstellung der Sensoren vor jeder Messung machen. Bei den hier gewhlten Sensoren ist es jedoch nur mglich sie vor jeder Fahrt einzustellen. Dies sollte jedoch kein Problem sein, da das Auto hier mit relativ gleichbleibenden Bedingungen zu tun hat. Um die Distanz einschtzen zu knnen sendet der Lichtsensor konstant Infrarotlicht aus und misst dann die Helligkeit. Diese Helligkeit wird dann je nach Einstellung verringert und als Spannung ausgegeben. Das Potentiometer, welches zum Einstellen eines Nullzustandes dient, ist hier ein Spannungsteiler. Dies ist auch der Grund warum man im Idealfall vor jeder Messung eine Einstellung vornimmt. Es ist hier erwhnenswert, dass die Intensitt des Lichtes, welches von dem Sensor selbst geschickt wird, proportional zu dem inversen Quadrat der Distanz, die zurckgelegt worden ist, ist. Deshalb ist bei gleich groen nderungen des Ausgangssignals nicht unbedingt mit einer gleichen nderung der Distanz zu rechnen. Die Probleme bei den hier gewhlten Sensoren liegt an zwei Stellen.: Erstens kann man das Potentiometer nicht so einstellen, dass man genug Helligkeit entnimmt um das Licht des Sensors reg-

istrieren zu knnen und zweitens gibt es bei dieses Sensoren keinen Filter der andere Wellenlgen herrausfiltert.

Lichsensor

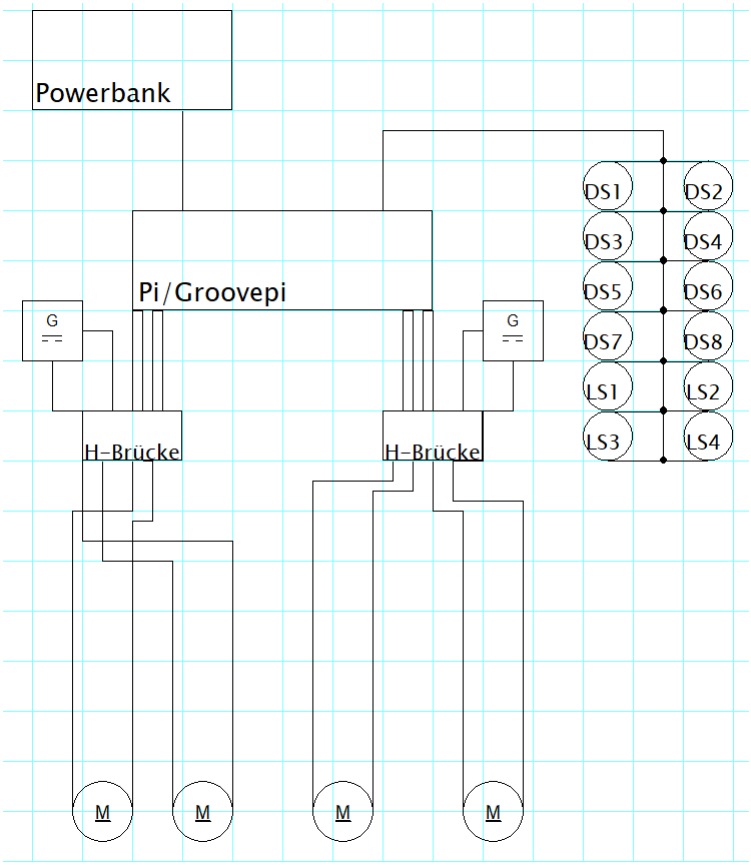
Ein weiteres ntiges Objekt fr das Projekt war der Lichtsensor. Der Lichtsensor braucht eine Spannung im Idealsten Fall von mindestens 3 bis maximal 5 n Volt. Der Versorgungsstrom liegt bei minimal 0.5 bis maximal 3 mA. Die Reaktionszeit des Sensors dauert zwischen 20-30 Sekunden und die Betriebstemperatur liegt bei -30 bis 70 Grad. Bei Licht entsteht ein Widerstand von 20 Kilo Ohm und bei Dunkelheit 1 Mega Ohm. In dem Projekt werden Lichtsensoren benutzt indem sie an den Grovepi angeschlossen sind und das Programm diese selbst justiert. Sie werden bentigt um Dunkle und Helle Oberflchen zu erkennen um sie als farbliche nderung zu identifizieren.

2.4.2 Schaltungen

Grundkonzept

Die Schaltung für das Selbsteinparkende Auto beinhaltet de Pi + Groovepi, die acht Distanzsensoren(DS 1-8) , die vier Lichtsensoren(LS 1-4),die zwei H-Brücken, vier Motoren(M 1-4),eine Powerbank und zwei 9V Blöcke.

Die Powerbank versorgt den Pi ,den Groovepi und die Sensoren mit dem nötigen Strom damit alles funktioniert, der Pi/Grovepi sendet ein Signal an die Inputs der H-Brücke welche von jeweils einem 9V Block versorgt werden und Außerdem auch die Motoren mit Strom versorgen welche von der H-Brücke auch gesteuert werden damit diese sich nach links oder rechts drehen.



2.4.3 Messprotokolle

H-Brücken

Die Messungen wurden von Murat Mete und Salome Richartz durchgeführt. Genutzt wurden zwei H-Brücken, vier Motoren, ein Labornetzgerät, ein Multimeter, ein Steckbrett und mehrere Leitungen. Damit es später gelang den Motor anzusteuern, wurde von dem Pluspol, welcher 9 Volt lieferte, eine externe Leitung gelegt. Daraufhin wurde die H-Brücke an die Spannungsversorgung angeschlossen, und diese an einen Motor. Um den Motor anzusteuern wurde die Leitung, die vorher abgezweigt wurde an einem den zugänglichen Pins, der H-Brücke, angeschlossen, um ein Signal von einem Microcontroller zu simulieren. Der Strom wurde direkt am Motor gemessen, wo schließlich das Multimeter zum Einsatz kam.

Motoren

Die Messungen wurden von Murat Mete und Salome Richartz durchgeführt. Genutzt wurden vier Motoren, ein Labornetzgerät, ein Multimeter, ein Steckbrett und mehrere Leitungen. Bei diesen Messungen wurde der Stromverbrauch bei Leerlauf und maximale Belastung des Motors festgestellt. Um dies zu erreichen wurde ein Motor nach dem anderen and die Spannungsversorgung angeschlossen. Gemessen wurde direkt am Motor. Um den Leerlauf festzustellen wurde der Motor in die Luft gehoben, somit war dieser nicht belastet. Um den Verbrauch bei maximaler Last zu dokumentieren wurde beim Messen der Motor kurz festgehalten.

Abstandssensoren

Die Messungen wurden von Phillip Schulz, Robert Grabarski, Manuel Hinz und Salome Richartz durchgeführt. Genutzt wurden ein Raspberry Pi 3 Model B, Abstandssensoren der Marke GroovePi, ein Schraubenzieher und ein Laptop. Es wurde ein Programm geschrieben, welches eine Nachricht ausgibt, ob der Sensor etwas gefunden hat oder nicht. Der Sensor wurde an den Raspberry Pi angeschlossen und das Programm wurde gestartet. Nun wurde der Sensor auf eine dunkle Ecke gerichtet und es wurde so lange an dem Potentiometer gedreht bis sich das Programm mit "nichts gefunden" meldete.

Lichtsensoren

Die Messungen wurden von Phillip Schulz, Robert Grabarski, Manuel Hinz und Salome Richartz durchgeführt. Genutzt wurden ein Raspberry Pi 3 Model B, Lichtsensoren der Marke GroovePi und ein Laptop. Es wurde ein Programm geschrieben, welches den gemessenen Helligkeitswert in einem passenden Format zurückgibt. Der Sensor wurde unterschiedlichen Helligkeitssituationen ausgesetzt und die Werte verglichen.

Chapter 3

Realisierung

3.1 Probleme

3.1.1 Gelöste Probleme

Im Rahmen des Projektes mussten einige Probleme überwunden werden. Das erste Problem war die Inbetriebnahme und Ansteuerung des Pis, da vor Ort kein Netzwerk war mit welchem man sich hätte verbinden können. Dies ist ein Problem, da alles was auf dem Pi passierte durch VNC geschah. Doch damit man VNC benutzen kann müssen das Gerät auf das man zugreifen will und das Gerät von welchem man agieren will im gleichen Netzwerk sein. Dieses Problem wurde durch das benutzen von privatem Datenvolumen aus den jeweiligen Handyveträgen gelöst. Ein weiteres Problem trat bei der Inbetriebnahme der Sensoren auf, als sich die Distanzsensoren nicht richtig kalibrieren ließen, weil die Störsignale (das Licht im Raum) zu groß bzw. hell waren. Dieses Problem ließ sich nicht direkt lösen, außer durch eine Verlegung des Ortes. Der Flur war hier schon dunkel genug, so dass man dieses Problem zu mindestens bei einer Präsentation hätte umgehen können. Doch es gab noch mehr Probleme mit diesen Sensoren, da sie scheinbar nicht zu kalibrieren waren. Das konnte jedoch später wiederlegt werden als festgestellt wurde, dass man nur den Pi nach jedem Kalibrieren neustarten sollte. Ein weiteres Problem war das Ansteuern von Pins des Pis und des GrovePis, da es hier viele widersprüchlicher Informationen im Internet gab. Nach viel Arbeit konnten jedoch die passendend Datenblätter gefunden werden. Ein weiteres Problem kam bei der Inberiebnahme des Pi‘s der anderen Gruppe auf. Hier war das Problem, dass das Image, welches auf die SDkarte geflasht wurde, nicht akutell genug war. Dies konnte durch das Hertunterladen eines neueren Images gelöst werden.

3.1.2 Ungelöste Probleme

Ausgangssignal des Pis

Eines der Probleme welches mit dem Pi aufkam war die Ansteuerung des Motors, da die Output Spannung von dem Pi bei einem High Signal nur 3,3 mV Gleichspannung war welches nicht ausreicht um in dem zugewiesen Zeitraum eine mit dieser Geringen Spannung Anständig funktionierende Schaltung zu bauen , aber dieses Problem ist eines welches bei dem Model des Pi(3b)häufiger vorkommt zumindest laut mehren Foreneinträgen welche gefunden wurden , wo auch gesagt wurde das es keine Lösung für dieses Problem gäbe außer einen neuen Pi zu holen

3.2 Teamdynamik

Am Anfang ergab sich die Problematik, da diese Gruppe nur aus fünf Individuen bestand, dass keine Teamleiter ernennen werden konnten. Darum hat sich die Gruppe entschlossen, einzelne Teilverantwortliche zu ernennen. Diese waren für einen bestimmten Bereich verantwortlich, mussten jedoch diesen nicht alleine bearbeiten, denn sie waren dazu befähigt ein anderes Teammitglied, welches zu diesem Zeitpunkt keine Arbeit hatte oder einen Leerlauf, für bestimmte Aufgaben heranzuziehen. Somit ergab sich eine stetige Kommunikation innerhalb der Gruppe, den trotz der Hauptverantwortung des einzelnen musste dieser nicht ein Experte in diesem Bereich sein, mithilfe des Informationsaustausches. Es ergaben sich jedoch auch einige Probleme durch diese Art der Aufgabenverteilung. Ein Mitglied der Gruppe war Aufgrund einer Krankheit die ersten 4 Termine nicht anwesend und musste innerhalb eines Termines auf den aktuellen Stand des Projektes gebracht werden. Dies konnte man jedoch nicht verhindern. Ein anderes Problem war die schwindende oder keinerlei vorhandene Motivation einzelner Individuen, die jedoch für schwerwiegende Bereiche verantwortlich waren, darüber hinaus arbeiteten diese auch nur wenn man sie explizit auf bestimmte Aufgaben ansetzte. Diese Probleme hätte man lösen können indem man mehrere für einzelne Bereiche Verantwortlich hätte machen sollen oder einen Hauptverantwortlichen für das komplette Projekt erklären können. Die Allgemeine Dynamik des Teams war rückblickend positiv. Jeder hat seinen Teil für das Projekt beigetragen und anderen Mitgliedern geholfen.

Chapter 4

Anhang

4.1 Quellen

Datum des Abrufes der Quellen : October 12, 2018

4.1.1 Elektrotechnik

- Datenblatt : Transistor
- Anleitung H-Brücke
- Datenblatt : Motor

4.1.2 RaspberryPi

- Ansteuerung des Pis
- ssh für Github auf dem Pi
- ssh für Github auf dem Pi

4.1.3 GrovePi

- Datenblatt / Anleitung : Abstandssensor
- Datenblatt / Anleitung : Helligkeitssensor
- Schaltplan GrovePi
- Blockschaltbild GrovePi