Dokumentation des Unipraktikums

Manuel Hinz

October 8, 2018

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 90-minütigen Einheiten über einen Zeitraum von sieben Wochen. 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 und weitere

Projekte vorzubereiten.

Chapter 2

Plannung

2.1 Ausblick

2.2 Programmierung

2.2.1 Einleitung

Bei der Programmierung des Projekts haben wir einen objektorientierten Lösungsansatz gewählt. Dies haben wir 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. Außerdem ist die OOP einer der Paradigmen, welche in der Realität benutzt werden.

2.2.2 Python

Als Programmiersprache haben wir Python gewählt, da wir diese Sprache im Informatikunterricht verwenden, sie für Laien leicht verständlich ist und wir 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 unsere Paradigmenwahl zulässt. Da sie eine beliebte und weitverbreitete Sprache ist unterstützt sowohl das aktuelle Image das wir auf dem Pi haben, als auch der Hersteller unsere Sensoren die Sprache. Auch kann man sagen, dass die Sprachwahl es uns 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

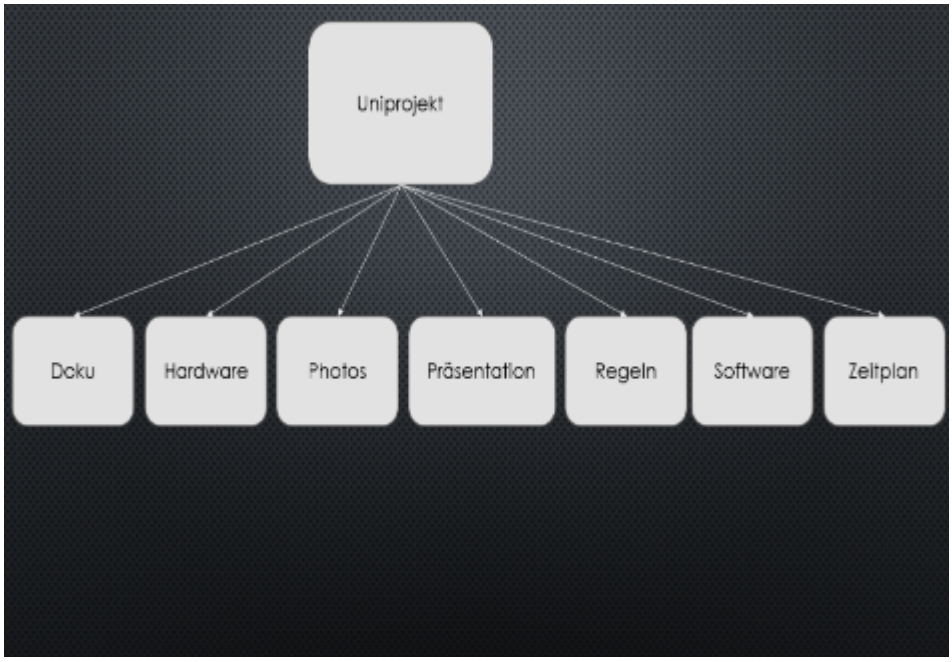
Jedes große Bauteil bekommt eine Klasse zugewiesen. Die wichtigen Bauteile sind hier: die Sensoren, die Motoren, die Motortreiber und der Raspberrypi.

Für die Sensoren haben wir 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. Wir haben dann 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 prsentiert werden kann. Diese Simulation benutz P5.js, was eine Javascriptbibliothek ist. In dieser Simulation soll ein Auto in eine Parklcke, welche durch zwei Hindernisse begrenzt ist, einparken. Es gibt hier momentan eine Klasse fr das Auto (Car) und eine Klasse fr die Hindernisse (Obstacle). Das Hauptskript findet sich in der Datei sketch.js. Spter wrde man auch eine Klasse fr die Intelligenz des Autos (Agent) finden. Im jetzigen Zustand wird die Bewegung des Autos durch einen Geschwindigkeitsvektor und einen Beschleunigungsvektor simuliert. Auerdem wird die Ausrichtung des Autos durch die Ausrichtung des Geschwindigkeitsvektors bestimmt. In der aktuellen Version der Simulation ist der Lenkvorgang beim Einparken verfgbar, sodass fr eine erste funktionsfhige 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 gewhlt wurde ist im obigen Bild dargestellt und in diesem Text genauer erlutert. Der erste Unterordner trgt den Namen Doku. In diesem werden alle Dateien fr die Dokumentation gespeichert, der Ordner ist jedoch noch mal in vier Unterordner unterteilt. Diese Ordner sind: erstens Abgabe, in welchem alle Dateien fr diese Dokumentation zu finden sind, zweitens Grundwissen, in welchem Dateien sind, die alle in der Gruppe befhigen sollen, jede Aufgabe eigenstndig zu lsen, drittens Quellen, in welchem alle unsere Quellen aufgefhrt sind und viertens Tagesdokus, in welchem alle Tagesdokumentationen sind. Der zweiter Unterordner Hardware enthlt Datenbltter und Messungen in jeweils einem eigenen Unterordner. Es ist hier anzumerken, dass nicht alle Bauteile ein Datenblatt und Messungen 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 luft, beherbergen. Das wren, im Falle Fertigstellung des Programmes, eine Datei pro Klassengruppe. Alles was im Unterordner Simulation befindet ist fr die Simulation, welche Html und Javascript benutzt. In dem Ordner Test befindet sich Python-Skripte, welche whrend 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 Prsentation beinhaltet unterschiedliche Versionen der Prsentation. Im Ordner Zeitplan befindet sich immer der aktuelle Zeitplan samt Anwesenheitsliste. Alle Fotos

sind in dem Ordner Photos um die Dokumentation und die Prsentation zu vervollstndigen. Durch diese Struktur sollen Dateien immer leicht zu finden sein. Auerdem wird durch diese Struktur die Abgrenzung von einzelnen Aufgabengebieten einfacher, wodurch sich alle Beteiligten leicht in jede Unteraufgabe eindenken knnen sollen.

2.3 Technologien

2.3.1 Github

Die Gruppe benutzt Github als unser 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 wir 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

Unsere Gruppe benutzte Anfangs Slack für Dateienspeicherung und Kommunikation mittlerweile laden wir auf Slack nur die Tagesdokus hoch und Kom-

munizieren miteinander, jedoch laden wir die wichtigeren Dateien und generelle Textdokumente auf Github hoch. Da Slack sowie Github als App herunterladbar sind benutzen wir auch diese Mobil, wenn wir unterwegs sind um auf den 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 wollen wir von den Vorteilen profitieren. 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 untersttzt werden (Beispiel: ,) oder schon eine Andere Belegung haben (Bsp.: _). Damit das schreiben der Dokumentation mglichst schnell passiert wurde deshalb von uns ein Skript geschrieben, welches alle diese problematischen Zeichen durch die LaTeX-Schreibweise ersetzt. Das Skript ist hier jedoch nur eine temporre Lsung, 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

Messen

2.4.2 Schaltungen

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 uns später gelang den Motor anzusteuern, wurde von 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 haben wir

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. Sobald der der Sensor uns ein nicht gefunden zurückgegeben hat, haben wir die Sensibilität des Sensors mithilfe des Schraubenziehers (auf dem Sensor befindet sich ein Potentiometer, welches sich mit dem Schraubenzieher verstellen lässt) auf unsere Wünsche eingestellt.

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 uns 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

3.1.2 Ungelöste Probleme

3.2 Teamdynamik

Chapter 4

Anhang

4.1 Quellen

4.1.1 Elektrotechnik

- http://cdn-reichelt.de/documents/datenblatt/A100/BC546_48-CDIL.pdf
- http://anleitung.joy-it.net/wp-content/uploads/2017/06/SBC-MotoDriver2-Anleitung.pdf
- http://cdn-reichelt.de/documents/datenblatt/A300/COM_MOTOR_RAD_DB-DE.pdf

4.1.2 RaspberryPi

- https://www.terraelectronica.ru/pdf/show?pdf_file=%252Fds%252Fpdf%252FT%252FTechicRP3.pdf
- http://www.netzmafia.de/skripten/hardware/RasPi/RasPi_GPIO_C.html
- https://help.github.com/enterprise/2.14/user/articles/generating-a-new-ssh-key-and-add
- https://devmarketer.io/learn/set-ssh-key-github/

4.1.3 GrovePi

- http://cdn-reichelt.de/documents/datenblatt/A300/GRV_IR_DISTANCE_AL-EN.pdf
- http://www.mouser.com/catalog/specsheets/Seeed_101020022.pdf
- https://github.com/DexterInd/GrovePi/blob/master/Hardware/GrovePi%2B%20v3.0.pdf

- https://github.com/DexterInd/GrovePi/blob/master/Hardware/GrovePi%20Graphical%20Datasheet.jpg