Dokumentation des Unipraktikums

Manuel Hinz

September 27, 2018

Contents

Chapter 1

Einleitung

1.1 Motivation

Im Moment sind selbstfahrende Autos so ziemlich das technische Thema. Universitten 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 Mglichkeit eines Praktikums im Rahmen des Elektrotechnikleistungskurses an der Bergischen Universitt Wuppertal anbot, fiel die Wahl des Themas leicht. Das dieses Praktikum nicht nur eine gute Mglichkeit ist um den aktuellen Technischen Entwicklungen nachzueifern, zeigt sich schon an den verschiedenen Aufgaben die zu erledigen sind. Im Laufe des Projektes mssen nmlich Motoren mithilfe von H-Brcken 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 Universittsalltag erlangt.

1.2 Zielsetzung

Es gibt viele mgliche Aufgaben, die man im Rahmen des Themas lsen knnte. Besonders interessant sind hier vor allem das finden von einem und das einparken auf einem Parkplatz, sowie das Teilnehmen am Straenverkehr. Letzteres entfllt wegen der Schwierigkeit des Simulierens von anderem Verkehr. Deshalb ist die hier gewhlte Aufgabe das finden und befahren einer Parklcke in einem straenhnlichen Setting. Genauer gesagt wird das Auto eine Strecke entlangfahren und mit Hilfe von Sensoren eine Parklcke in die es hineinpasst identifizieren und dann dort einparken. Dies wird innerhalb von 16 90-mintigen Einheiten ber einen Zeitraum von sieben Wochen. Ein weiteres Ziel ist es das fr die Vollendung des Projektes ntige Wissen fr alle Gruppenmitglieder verstndlich zu bearbeiten und die Gruppe damit fr das Abitur und weitere Projekte vorzubereiten.

Chapter 2

Theorie

2.1 Ausblick

2.2 Programmierung

2.2.1 Einleitung

Bei der Programmierung des Projekts haben wir einen objektorientierten Lsungsansatz gewhlt. 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 zugehrigen Klassen herstellen kann. Auerdem ist die OOP einer der Paradigmen, welche in der Realitt benutzt werden.

2.2.2 Python

Als Programmiersprache haben wir Python gewhlt, da wir diese Sprache im Informatikunterricht verwenden, sie fr Laien leicht verstndlich ist und wir mehrere Personen in der Gruppe bzw. dem nahen Umfeld haben, welche diese Sprache bereits gut knnen. Auerdem untersttzt Python die OOP, wodurch die Sprache auch unsere Paradigmenwahl zulsst. Da sie eine beliebte und weitverbreitete Sprache ist untersttzt 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 ermglicht Code, welcher schon frher von einzelnen Mitgliedern geschrieben wurde, wieder zu benutzen. Ein Nachteil ist hier, dass Python nicht alle Konzepte der OOP untersttzt (zum Beispiel Abstrakte Klassen, Interfaces/Traits).

2.2.3 Programmstruktur

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

Fr 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 berschreiben.

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 knnen. Auerdem erleichtert Github das gleichzeitige Arbeiten, sodass mehrere Leute gleichzeitig programmieren, und die Programme schlussendlich hochladen knnen. Ein weiteres Plus ist, dass Github eine groe Anzahl an letzten Versionen speichert, wodurch auch spt erkannte Fehler leicht berwunden werden knnen. Als letztes Argument fr diese Technologie ist zu sagen, dass sie quasi als Dokumentation fr alles was auf Github hochgeladen wird dienen kann. Da man jeden Schritt nachvollziehen kann.

2.3.2 Slack

2.3.3 LaTeX

Fr 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 ndern knnen. Auerdem ist LaTeX eine alternative zu Word, wobei man viele der blichen Probleme von Word umgeht. Beispiele wren hier zum Beispiel Verschiebung von Text bei dem Einfgen von Bildern, welches man in LaTeX deutlicher definieren muss.

2.4 Elektrotechnik

Chapter 3

Praxis

3.1 Probleme und Problemlsungen

3.2 Menschliche Zusammenarbeit

Chapter 4

Anhang

4.1 Quellen

Test