

**Hochschule Flensburg**

# **SmartCars**

## **Maker's Lab - Things that Think**

Abschlussbericht

vorgelegt von:

**Simon Hauck**

660158

Hochschule Flensburg

simon.hauck@stud.hs-flensburg.de

**Nils Jensen**

670758

Hochschule Flensburg

nils.jensen2@stud.hs-flensburg.de

9. März 2020

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>2</b>
1.1 Motivation . . . . .	2
1.2 Projektidee . . . . .	2
<b>2 Konzept</b>	<b>4</b>
2.1 Umsetzung . . . . .	4
<b>3 Komponenten</b>	<b>5</b>
3.1 Hardware . . . . .	5
3.1.1 Fahrzeuge . . . . .	5
3.1.2 Visualisierung . . . . .	5
3.2 Software . . . . .	5
3.2.1 Modellierung und Slicing . . . . .	6
<b>4 1. Design-Iteration</b>	<b>6</b>
4.1 Fahrzeuge . . . . .	6
4.2 Hard- und Software . . . . .	9
4.3 Darstellung . . . . .	10
4.4 Ergebnis . . . . .	11
<b>5 2. Design-Iteration</b>	<b>12</b>
5.1 Fahrzeuge . . . . .	12
5.2 Hard- und Software . . . . .	14
5.3 Darstellung . . . . .	15
5.4 Ergebnis . . . . .	16
<b>6 3. Design-Iteration</b>	<b>17</b>
6.1 Fahrzeuge . . . . .	17
6.2 Hard- und Software . . . . .	17
6.3 Darstellung . . . . .	20
6.4 Ergebnis . . . . .	21
<b>7 4. Design-Iteration</b>	<b>23</b>
7.1 Fahrzeuge . . . . .	23
7.2 Hardware . . . . .	24
7.3 Darstellung . . . . .	27
7.4 Ergebnis . . . . .	27
<b>8 Fazit</b>	<b>30</b>
8.1 Ausblick . . . . .	31

# 1 Einleitung

Im Zuge der Veranstaltung “Maker’s Lab - Things that Think“ wird die Entwicklung interaktiver Systeme, bestehend aus Hard- sowie Software, thematisiert. Neben dem Fokus auf der Verwendung von Raspberry Pi und Arduino als Hardware-Komponenten liegt ein weiterer Schwerpunkt auf der Verwendung von Techniken und Praktiken aus den Bereichen *making* und *rapid prototyping*. Unter Berücksichtigung dieser Punkte gilt es, eine “anfassbare“ Schnittstelle zwischen Mensch und Computer zu schaffen, welche gleichzeitig thematisch dem Klimapakt Flensburg und seiner Mission entspricht. Nachfolgend soll einerseits die Motivation und andererseits die daraus resultierende Projektidee umschrieben werden.

## 1.1 Motivation

Mit dem Ziel, die Folgen alltäglichen Handelns des Einzelnen für die Umwelt besser veranschaulichen zu können, soll die tägliche Verwendung des eigenen PKWs thematisiert werden. Bei der späteren Interaktion mit dem fertigen Prototypen soll vor allem der Fokus darauf liegen, generationsübergreifend eine möglichst intuitive und zugängliche Bedienung zu bieten. Vor diesem Hintergrund sollen bereits bekannte Konzepte und Komponenten aufgegriffen werden welche die Möglichkeit bieten, auf spielerische Art und Weise zu lernen.

## 1.2 Projektidee

Es soll eine Kombination aus Eingabegerät und Darstellung geschaffen werden, die sich grob an der Bewegung von Spielzeug-Autos auf einem mit einer abstrakten Stadt und ihrer Verkehrsinfrastruktur bedruckten Teppich orientiert. Gleichzeitig soll maßstabsgetreu die Bewegung der Fahrzeuge aufgenommen und in ihre entsprechende CO<sub>2</sub>-Emmission übersetzt werden. Die so anfallende Belastung für die Umwelt soll auf einem Display optisch ansprechend aufbereitet und mit anschaulichen, zusätzlichen Informationen zusammengebracht werden. Hierdurch sollen die Konsequenzen des eigenen Handelns verdeutlicht und so einem Nutzer näher gebracht werden können, als es durch das Auflisten von Statistiken oder Ähnlichem möglich wäre. In Abbildung 1 ist ein erster *Low-Fidelity*-Prototyp der hier beschriebenen Idee zu sehen

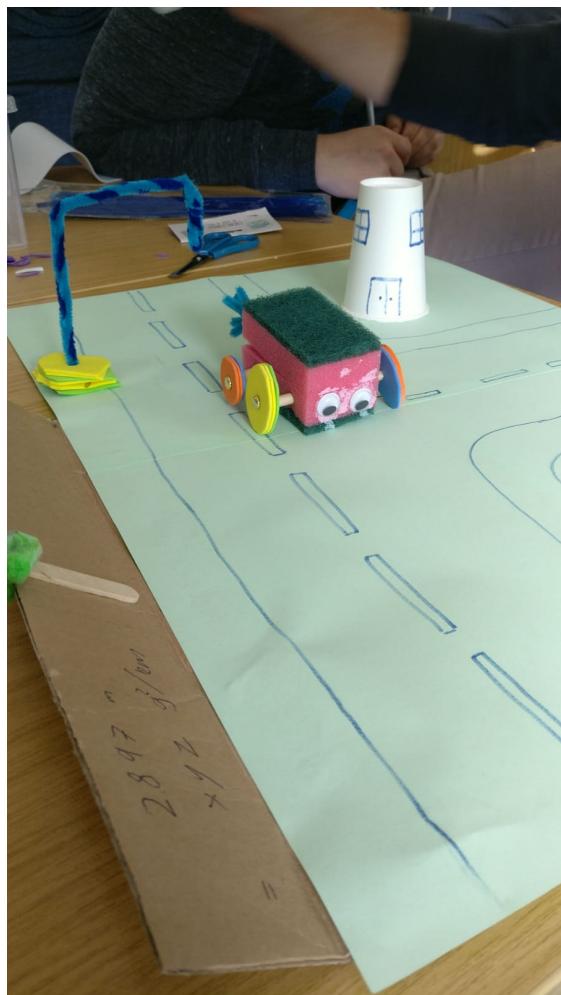


Abbildung 1: Erster Prototyp

## 2 Konzept

Im Folgenden soll nun umrissen werden, wie die in 1.2 formulierte Idee zu realisieren ist. Als Kontaktpunkt des Nutzers mit dem System sollen eines oder mehrere der erwähnten Fahrzeuge eingesetzt werden. Diese sollen per Hand bewegt werden. Im Kontrast hierzu stünde eine Umsetzung, in der die Fahrzeuge zum Beispiel mit Hilfe einer App und entsprechender Technik ferngesteuert werden können. Diese Wahl soll zum einen eine einfachere, vor allem aber auch kostengünstigere Umsetzung ermöglichen und zum Anderen ein intuitiveres Bedienerlebnis bieten. Um hierauf aufzubauen und den thematischen Schwerpunkt des Klimapakts Flensburg aufzugreifen, wird die Möglichkeit in Betracht gezogen, anstelle eines generischen Teppichs eine maßstabsgetreue Darstellung der Flensburger Innenstadt sowie eventuell des Umlands zu verwenden. Auf diesem Wege wäre es weiter möglich, die Konsequenzen auch kürzerer Fahrten zu verdeutlichen. Die Darstellung mittels des Displays könnte neben aufbereiteten Zahlen weiter um eine Repräsentation der Emissionen in weniger abstrakter Form wie eine Menge an bestimmten Lebensmitteln, deren Produktion und Transport gleiche Werte verursachen, ergänzt werden.

### 2.1 Umsetzung

Um die Anforderungen an die Fahrzeuge erfüllen zu können, wurde einerseits evaluiert, fertige Spielzeugautos zu erwerben und um die elektronischen Komponenten zu erweitern, die zur Messung der zurückgelegten Distanz und der Kommunikation benötigt werden. Allerdings hätte sich vorab nicht ohne Zweifel sagen lassen können, ob der für diese Erweiterungen erforderliche Platz vorhanden ist oder sich schaffen lässt, ohne das Auto seiner Funktion zu berauben. Andererseits bestand die Option, die Fahrzeuge von Grund auf selbst herzustellen. Auf diese Weise kann sichergestellt werden, dass genug Platz für die erforderliche Technik vorhanden ist. An dieser Stelle entsteht jedoch ein Konflikt zwischen dem Platzbedarf der zu ermittelnden Komponenten und einem der Projektidee entsprechenden Formfaktor der Fahrzeuge. Weiter sollen die Fahrzeuge in einer Form geschaffen werden, die es erlaubt, ohne weitere Konfiguration "loszufahren", sobald eine initiale, gleichsam werksseitige, Konfiguration durchgeführt wurde und die Stromversorgung hergestellt ist. Die Stromversorgung der Fahrzeuge soll mit Hilfe von Akkus umgesetzt werden, damit keine Kabel etwaiger Netzteile bei den Bewegungen der Fahrzeuge berücksichtigt werden müssen oder diese gar behindern. Ein Verzicht auf Kabel soll auch in der Kommunikation der Fahrzeuge mit der Steuerung der Darstellung gewahrt bleiben. Hierdurch entsteht die Anforderung, beide Seiten des Systems mit solchen Bauteilen auszustatten, die eine drahtlose Kommunikation erlauben. Die erforderlichen Komponenten sollen so gewählt und installiert werden, dass eine Wartung ohne unverhältnismäßigen Aufwand möglich ist. Weiter soll durch die Durchführung der Montage gewährleistet werden, dass kostspieligere Bauteile entfernt und anderweitig erneut eingesetzt werden können.

## 3 Komponenten

### 3.1 Hardware

Das Projekt besteht aus zwei Teilen, den Fahrzeugen und der Visualisierung. Die hierfür benötigte Hardware soll im Folgenden vorgestellt werden. Die genannten Komponenten beziehen sich auf das Konzept, welches in Kapitel 2 vorgestellt wurde.

#### 3.1.1 Fahrzeuge

Die Basis für die Fahrzeuge sollen Microcontroller sein. Diese benötigen wenig Strom und ermöglichen so ein längeres Spielvergnügen. Für die Fahrzeuge soll der Arduino MKR WiFi 1010 verwendet werden. Dieser ist ca. 62mm lang und 25mm breit und somit bedeutend kleiner als der Arduino Uno. Zusätzlich ist der Arduino besonders für IoT Projekte geeignet, da ein Low Power Prozessor (SAMD21) verwendet wird. Das Board besitzt zudem eine integrierte Ladeschaltung und einen Anschluss für einen Lithium-Polymer-Akku. Hierfür wird ein Modell mit einer Kapazität von 1000mAh verwendet, welches einen Kompromiss zwischen der Größe des Akkus und dem Formfaktor bietet. Des Weiteren besitzt der verwendete Arduino ein integriertes WiFi-Modul, welches später zur Kommunikation mit der Visualisierung verwendet wird. Neben dem Microcontroller wird jedes Fahrzeug mit einem oder mehreren Magneten sowie einem Hall-Sensor ausgestattet. Der Hall-Sensor gibt ein Signal an den Arduino, wenn sich ein Magnet in dessen Nähe befindet. Diese werden später zur Erkennung von Radumdrehungen verwendet. Zuletzt werden zwei LEDs mit den dazugehörigen Widerständen als Scheinwerfer verwendet.

#### 3.1.2 Visualisierung

Die Basis für die Visualisierung soll ein Raspberry Pi 4 sein. An diesen kann ein Display angeschlossen werden, welches später die erfassten Daten aufbereitet anzeigt. Der Raspberry Pi fungiert zusätzlich als WLAN *Access Point*. Die in den Fahrzeugen verwendeten Arduinos können sich direkt mit diesem verbinden womit die kabellose Übertragung der Daten gewährleistet ist. Dieser Aufbau hat den Vorteil, dass keine weitere Hardware wie zum Beispiel ein Router zwischen den Komponenten benötigt wird.

### 3.2 Software

Für das Projekt werden zwei verschiedene Programmiersprachen verwendet. Die Arduinos verwenden standardmäßig eine C- beziehungsweise C++-ähnliche Programmiersprache, für welche die Entwicklungsumgebung *Arduino IDE* verwendet wird. Für die Visualisierung stehen verschiedene Programmiersprachen zur Auswahl. Für dieses Projekt soll Python 3.7 verwendet werden. Mit dieser Sprache wird das einfache Erstellen eines Webservers, welcher zum Empfangen der Daten genutzt wird und das Ansteuern der IO-Pins beziehungsweise des Displays vom Raspberry Pi ermöglicht.

### 3.2.1 Modellierung und Slicing

Um die Fahrzeuge mit Hilfe von 3D-Druck herstellen zu können, musste eine Software gefunden werden mit der die benötigten Modelle erschaffen und angepasst werden können. Zweierlei Lösungen wurden hierfür in Betracht gezogen. Zum einen *Blender*<sup>1</sup>, da das erforderliche Dateiformat *stl* nativ unterstützt wird und derartig weit verbreitete Open Source Lösungen oft eine entsprechend große Community vorweisen können. Zum anderen *SketchUp*<sup>2</sup>, welches zwar nur mit verringertem Funktionsumfang frei verfügbar ist, aber besonders durch seine einfache Bedienung überzeugt. Die Wahl fiel darauf, zunächst erste Gehversuche mit *Blender* zu unternehmen und im Fall größerer Schwierigkeiten in der Bedienung auf *SketchUp* zurückzugreifen, da bei dieser Software auf bereits vorhandene Erfahrungen zurückgegriffen werden kann. Die Unterstützung für das *stl*-Format muss in diesem Fall jedoch über Erweiterungen sichergestellt werden. Zur Vorbereitung etwaiger Modelle für den Druck war es weiter nötig, eine geeignete Software für den *Slicing*-Vorgang zu finden. Nach erfolgter Recherche und auf Anraten des *FabLab* der Hochschule Flensburg fiel die Wahl auf *Cura*<sup>3</sup>. Neben einfacher Handhabung überzeugt *Cura* durch die Unterstützung der zur Durchführung des Projekts verfügbaren Drucker und dem *stl*-Format.

## 4 1. Design-Iteration

### 4.1 Fahrzeuge

Vor Beginn erster Modellierungsarbeiten wurden zunächst bemaßte Skizzen, zu sehen in Abbildung 2, angefertigt. Hierdurch war es möglich, den ermittelten Platzbedarf der Hardware-Komponenten abzuschätzen und einen ersten Eindruck der Dimensionen der Fahrzeuge zu erhalten.

---

<sup>1</sup><https://www.blender.org/>

<sup>2</sup><https://www.sketchup.com/de>

<sup>3</sup><https://ultimaker.com/de/software/ultimaker-cura>

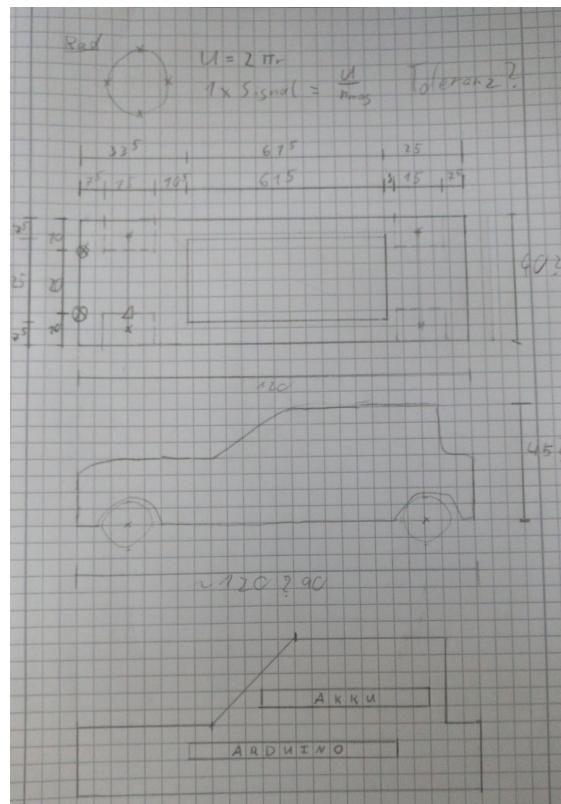


Abbildung 2: Erste Skizze eines Fahrzeugs

Bei der eigentlichen Modellierung wurde versucht, zunächst mit so wenig Details wie möglich, die grobe Form eines PKWs nachzubilden. Hierzu wurde mit *Blender* zunächst ein Quader geschaffen, der in seinen Dimensionen der Skizze entspricht.

Allerdings zeigte sich, dass das Tempo der Modellierung mit *Blender* auf Grund mangelnder Erfahrung in der Bedienung unverhältnismäßig langsam und somit als nicht zufriedenstellend zu bewerten ist. Aus diesem Grund wurde die in 3.2.1 zur Sprache gekommene Ausweichlösung in Form von *SketchUp* für den Rest der Arbeiten verwendet. Der auf diesem Wege, nun in *SketchUp*, geschaffene Quader wurde in seiner Form dahingehend manipuliert, dass er äußerlich eher als PKW zu erkennen ist. Neben einer angedeuteten Frontsektion mit Motorhaube und abgeschrägter Scheibe wurden bereits in diesem Schritt Radkästen und Aussparungen für die später benötigten Achsen geschaffen. Bei der Dimensionierung der Räder und infolgedessen auch der Radkästen galt es zu beachten, dass die verwendeten Magneten ausreichend Platz finden, ohne später die strukturelle Integrität des Drucks zu gefährden oder sich so nah bei einander befinden, dass vom Hall-Sensor keine einzelnen Magneten mehr erkannt werden können.

Um den Raum für die elektronischen Komponenten zu schaffen, wurde das Fahrzeug horizontal in etwa in der Mitte seiner Höhe in zwei Teilmodelle aufgebrochen. In beide

Hälften wurde nun eine rechteckige Aussparung eingearbeitet, die der Skizze aus Abbildung 2 entsprechend, ausreichend Platz bietet. Hierbei galt es zu beachten, Änderungen an den Dimensionen der einen Hälfte präzise auf die andere anzuwenden. Die hier beschriebenen Arbeitsschritte sind in den Abbildungen 3 und 4 zu sehen.

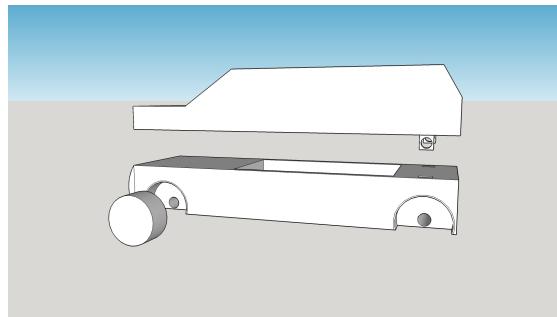


Abbildung 3: Seitenansicht des ersten Modells

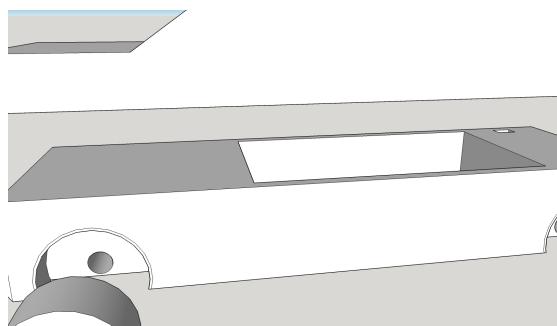


Abbildung 4: Aussparung im ersten Modell

Abbildung 3 zeigt weiter einen ersten Ansatz, um beide Fahrzeughälften miteinander verbinden und so fixieren zu können. In der unteren Hälfte werden Aussparungen vorgesehen, in die entsprechende Elemente der oberen eingelassen werden können. Diese zeichnen sich durch eine ösenartige Form aus. Somit ist es möglich, entsprechend bemessene Bolzen gleichzeitig durch beide Hälften zu führen, um diese zu fixieren. Um zu gewährleisten, dass die aus der Skizze übernommenen Maße passend sind, wurde an dieser Stelle mit einem ersten Druck begonnen. Die in *Cura* verwendeten Einstellungen wurden unter Anleitung des *FabLab*-Personals an den jeweiligen Drucker angepasst. Abbildung 5 zeigt das Ergebnis dieses Drucks.

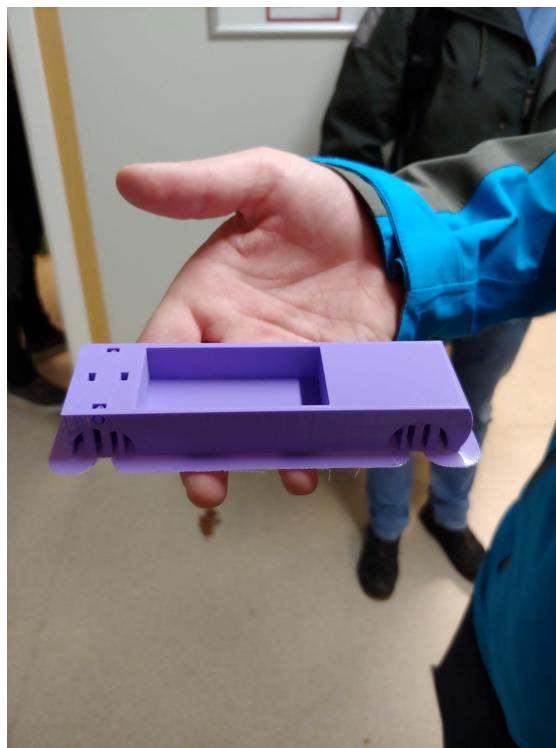


Abbildung 5: Druck des ersten Modells

## 4.2 Hard- und Software

### Fahrzeug

Nachdem die Hardware eingetroffen ist, soll in einem ersten Schritt die Funktionalität des Hall-Sensors testweise implementiert werden. Das Ziel ist herauszufinden, bis zu welchem Abstand der Sensor die Magneten erkennt und ob der Einsatz von mehreren Magneten in den Reifen möglich ist. Für den Sensor steht nur Dokumentation im Rahmen eines Datenblattes zur Verfügung. Hierbei erweist es sich als schwierig, die Bedeutung der drei Pins zu erkennen. Letztendlich stellt sich heraus, dass der linke Pin +, der mittlere *GND* und der rechte das *Output* Signal ist.

Beim Erstellen der Schaltung entsteht ein weiteres Problem. Der Sensor benötigt für den Betrieb mindestens  $3.7V$ . Somit muss der +-Pin an den  $5V$  Pin des Arduinos angeschlossen werden. Der *Output*-Pin liefert dementsprechend auch  $5V$  und muss mit einem IO-Pin des Arduinos verbunden werden, damit das Signal von diesem eingelesen werden kann. Laut Datenblatt dürfen an die Pins maximal  $3.3V$  angelegt werden, ansonsten droht eine Beschädigung des Arduinos.

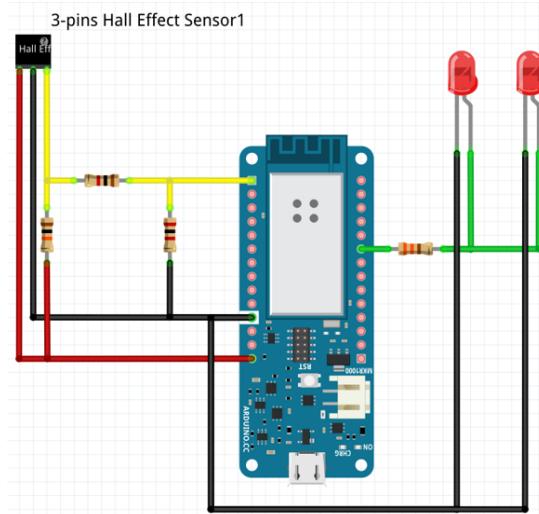


Abbildung 6: Erstes Konzept des Schaltplans für die Fahrzeuge

Mit einem Spannungsteiler, bestehend aus einem  $1k\ \Omega$  und einem  $2k\ \Omega$  Widerstand, können  $5V$  auf  $3.3V$  runter-gesetzt werden. Zusätzlich soll ein  $10k\ \Omega$  Pull-Up Widerstand verwendet werden, welcher den verwendeten Pin des Arduinos auf *HIGH* setzt, solange kein Magnet erkannt ist. Die verwendete Schaltung ist in Abbildung 6 dargestellt. Hierbei werden zusätzlich zwei LEDs hinzugefügt, welche später als Scheinwerfer im Fahrzeug verwendet werden.

### 4.3 Darstellung

Da bereits vor der ersten Materialbestellung deutlich wurde, dass geeignete Displays den finanziellen Rahmen des Projekts sprengen würden, wurden Konzepte für alternative Formen der Visualisierung erarbeitet. Obwohl die Machbarkeit zu diesem frühen Zeitpunkt der Durchführung nicht abschließend geklärt werden konnte, fiel die Wahl auf eine abstraktere Art der Darstellung in Form eines Baums mit beweglichen Ästen. Bei zunehmenden Emissionen sollen diese Äste mit Hilfe von Servomotoren abgesenkt und gleichermaßen nach einer bestimmten Zeit wieder angehoben werden können. Um das so entstehende Bild weiter ausprägen zu können, soll der gesamte Baum in einer Kiste montiert werden, welche in ihrem inneren mit LEDs umrandet ist. Diese sollen, abhängig von der Position der Äste, ihre Farbe von grün zu rot wechseln. Abbildung 7 zeigt eine frühe Konzeptskizze der Umsetzung dieses Baums.

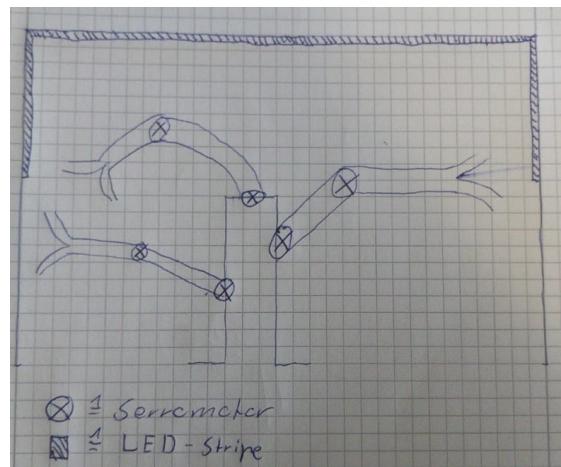


Abbildung 7: Erste Skizze des neuen Visualisierungskonzepts

#### 4.4 Ergebnis

Obwohl die in 4.1 gezeigte Skizze die Dimensionen der zu installierenden Komponenten genau berücksichtigte, war der im Inneren des Fahrzeugs vorgesehene Platz nicht ausreichend bemessen. Zu sehen ist diese Abweichung in Abbildung 8. Der Grund hierfür ist, dass die Spezifikationen des gewählten Akkus nicht berücksichtigen, dass der Akku selbst zusätzlich in einer schützenden Hülle verpackt ist. Um diese Hülle nicht entfernen zu müssen, wurden die Maße der Aussparung im Inneren des Fahrzeugs angepasst. Dadurch ergab sich weiter der Bedarf, diesen Zuwachs auch auf die Außenmaße anzuwenden. Außerdem wurde deutlich, dass das beispielsweise für die Radkästen benötigte Stützmaterial auch in den Hohlräumen der Befestigungsvorrichtung eingesetzt wurde. *Cura* verfügt nicht über die Option, Stützmaterial nur punktuell einzusetzen. Es wird entweder an jeder ermittelten Stelle verwendet oder an keiner.



Abbildung 8: Abweichende Maße des Akkus durch Hülle

Durch die Wahl einer abstrakteren Art der Darstellung besteht die Möglichkeit, dass der persönliche Bezug des Nutzers zu den simulierten Emissionen verloren geht. Die in 2 zur Sprache gekommene Unterlage verliert so, den damit verbundenen Aufwand berücksichtigend, an Wert. Obwohl also durch die Umsetzung mit Hall-Sensoren eine recht präzise Messung der zurückgelegten Distanz möglich gewesen wäre, fiel an dieser Stelle die Entscheidung, auf jedwede Art von Untergrund für die Fahrzeuge zu verzichten. Beim Testen der Fahrzeugschaltung wurde festgestellt, dass die Kombination aus Pegelwandler und dem Pull-Up Widerstand nicht funktioniert und kein Auslesen von Sensorwerten möglich ist.

## 5 2. Design-Iteration

### 5.1 Fahrzeuge

Um die in 4.4 identifizierten Probleme hinsichtlich des Platzbedarfs zu lösen, wurden die Maße des Fahrzeugs angepasst. Da der erste Eindruck der Beschaffenheit des in Abbildung 5 gezeigten Drucks hinsichtlich seiner Stabilität die anfänglichen Erwartungen übertroffen hat, wurden zunächst die zu den Seiten des Fahrzeugs gelegenen Außenwände

der Aussparung in ihrer Stärke halbiert. Weiter wurde das gesamte Modell in der Länge erweitert und der Hohlraum im Fahrzeug so weit wie möglich in diese Richtung vergrößert. Hierdurch soll zukünftigen Platz-Engpässen vorgebeugt werden. An dieser Stelle wurde ein weiterer Druck durchgeführt, um das Zusammenspiel mit Rädern und Befestigung zu prüfen. Hierbei wurden alle Einzelteile für den Druck in einem einzigen Drucker arrangiert.

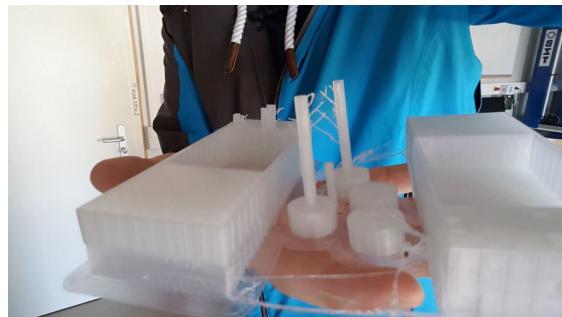


Abbildung 9: Alle Komponenten eines Fahrzeugs aus einem Druck



Abbildung 10: Fahrzeughälften mit montierten Rädern

Während Abbildung 9 das Ergebnis dieses Drucks zeigt, vermittelt Abbildung 10 einen Eindruck von Form und Dimensionen des fertigen Produkts.

## 5.2 Hard- und Software

### Fahrzeug

Weitere Nachforschungen bezüglich der Fahrzeugschaltung haben ergeben, dass der *Output*-Pin des Hall-Sensor mit einem  $1k \Omega$  Widerstand an den  $3.3V$ - sowie den gewählten IO-Pin angeschlossen werden muss. Das neue Diagramm ist in Abbildung 11 dargestellt.

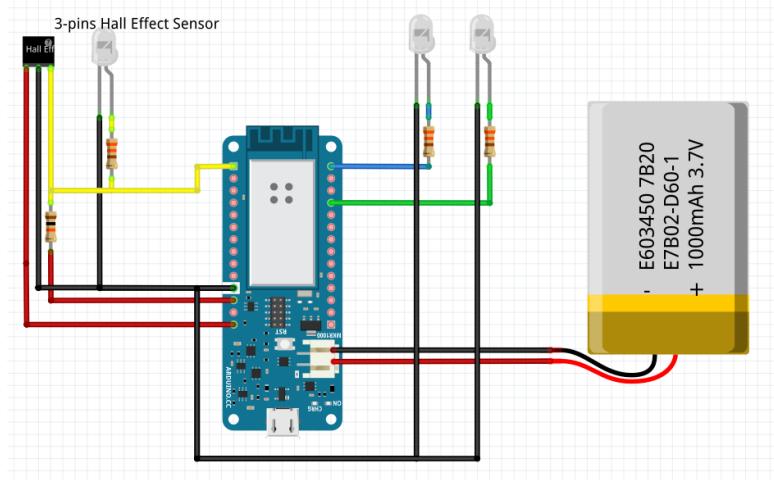


Abbildung 11: Schaltung für die Fahrzeuge in der zweiten Iteration

Mit dieser Schaltung funktioniert das Auslesen des Sensors ohne Probleme. Die maximale Distanz, bis zu welcher Magnete erkannt werden können, beträgt ca.  $1cm$ . Die Magnete müssen sich hierfür direkt vor dem Sensor befinden. Somit ist es möglich, mehrere Magnete in den Rädern des Fahrzeugs unterzubringen, was eine genauere Messung der zurückgelegten Strecke ermöglicht. Zusätzlich werden die LEDs, welche als Scheinwerfer verwendet werden an, je einen eigenen IO-Pin angeschlossen. Somit können diese einzeln angesteuert werden. Diese Eigenschaft wird genutzt, um Statusinformationen auszugeben, zum Beispiel wenn der Controller sich mit dem WLAN Netzwerk verbindet oder Fehler beim Senden der Daten auftreten. Somit ist ein leichtes Erkennen von Fehlern möglich, auch wenn das Fahrzeug nicht per USB-Kabel an einen Computer angeschlossen ist. Mit der funktionierenden Hardware wird das komplette Programm entworfen. Eine Änderung des Signals vom Hall-Sensor wird durch einen *Interrupt* erkannt. Somit gehen keine Messwerte verloren, auch wenn der Arduino eine andere Funktion wie das Senden von Daten ausführt. Der Arduino wartet eine Zeit  $x$  ab, welche von der Konfiguration abhängt. Danach überprüft er, ob das Fahrzeug sich bewegt hat und Messwerte vorliegen. Ist dies der Fall, werden die Daten in ein JSON-Objekt kodiert und anschließend an die Visualisierung gesendet. Liegen mehrere Messwerte vor, werden diese kombiniert. Neben den Messwerten enthalten die Nachrichten zusätzlich den Fahrzeugtyp, welcher für die Berechnung der erzeugten Emissionen verwendet wird. Zum Senden der Daten

wird das integrierte WLAN-Modul genutzt, welches einen *Post-Request* durchführt. Ist das Senden erfolgreich, startet der gesamte Prozess von vorne. Verliert der Arduino die Verbindung zum WLAN, versucht er diese wiederherzustellen und zeigt dies durch ein abwechselndes Blinken der LEDs an. In dieser Zeit werden alle vom Sensor erfassten Daten ignoriert. Scheitert der Sendevorgang, wird über die LEDs ein schnelles Blinken abgegeben und mit dem Programmablauf wird normal fortgefahren.

### 5.3 Darstellung

Die im Zuge der vorangegangenen Iteration erdachte Umsetzung einer abstrakteren Darstellungsweise wurde in diesem Arbeitsschritt vorangetrieben. Anhand eines ersten Papierprototypen, zu sehen in Abbildung 12, wurde begonnen, das bereits bei der Herstellung der Fahrzeugprototypen gewonnene Wissen auf die Schaffung eines druckfähigen Baummodels anzuwenden.



Abbildung 12: Papierprototyp des Baums

Mit den Maßen der Servomotoren als Referenz wurde mit den Modellierungsarbeiten begonnen. Ebendiese Maße bedingten jedoch verhältnismäßig dicke Äste in der geplanten Umsetzung des Baums. Die daraus resultierenden Dimensionen, insbesondere in die Tiefe, zeigt Abbildung 13.

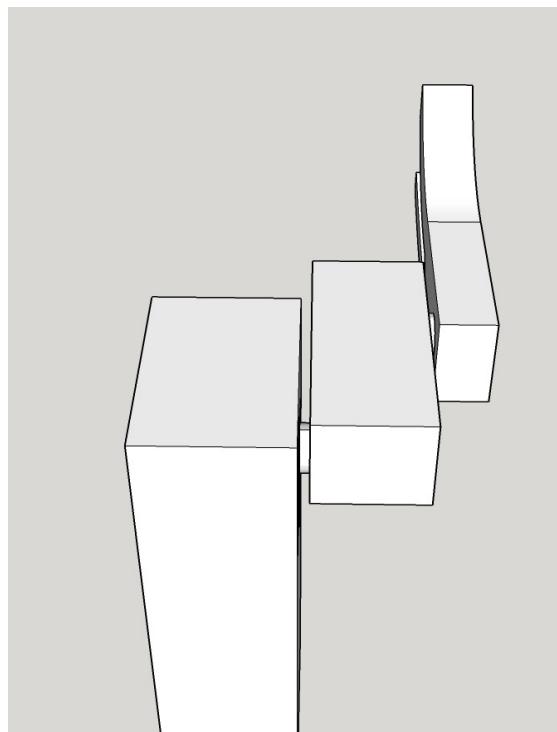


Abbildung 13: Dimensionen des Baummodels

#### 5.4 Ergebnis

Während das in 4 zum Vorschein gekommene Problem des Platzmangels im Fahrzeugginneren gelöst werden konnte, wurde deutlich, dass es nicht möglich ist, wie in 5.1 beschrieben, alle Komponenten in einem Druckvorgang herzustellen. Obwohl die beiden Hälften des Fahrzeugs mit einer Füllmaterialdichte von 30% eine ausreichende Stabilität erreichen, ist dies bei den Achsen und Bolzen zur Befestigung nicht der Fall. Bereits bei der zweiten Montage sind die an den Rädern befindlichen Achsen, wie in Abbildung 14 gezeigt, gebrochen.

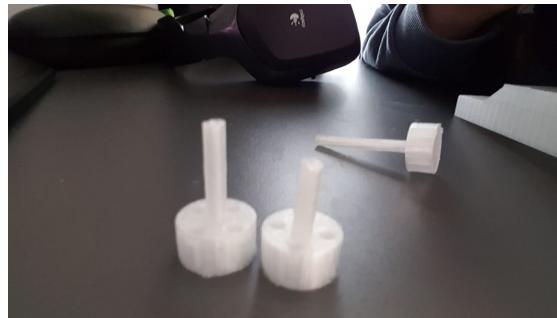


Abbildung 14: Achsbruch durch Mangel an Füllmaterial

Darüber hinaus konnte auch das Problem des Stützmaterials in eigentlich dringend benötigten Hohlräumen zur Befestigung noch nicht behoben werden. Auch die in 5.3 beschriebene Lösung zur Darstellung der Emissionen ist in dieser Form noch nicht zufriedenstellend. Die Tests der Fahrzeugschaltung sind erfolgreich verlaufen und die Grundstruktur für den Programmablauf wurde implementiert.

## 6 3. Design-Iteration

### 6.1 Fahrzeuge

Um alle Komponenten so einfach wie möglich montieren zu können, wurden weitere Anpassungen am Modell vorgenommen. Zunächst wurden im Frontbereich Aussparungen für die LED-“Scheinwerfer“ geschaffen. Hierzu wurde der Durchmesser der LEDs aufgenommen und in Form von zwei “Tunneln“ durch das Modell eingearbeitet. Mit dem Ziel einen sichereren Halt zu realisieren, wurde dieser Tunnel in etwa hinter der Länge der LEDs verengt. Auf diesem Wege soll eine LED an ihrem Platz bleiben können, während die erforderlichen Kabel trotzdem weiter in das Innere des Fahrzeugs geführt werden können. Nach demselben Prinzip wurde darüber hinaus ein Platz zur Montage des Hall-Sensors im Radkasten geschaffen. Da keine genauen Maße zur Verfügung standen, wurde der Platzbedarf näherungsweise per Hand bestimmt und etwas erweitert, um etwaigen Messfehlern vorzubeugen. Da sich das Problem um die übereifrigie Platzierung von Stützmaterial durch *Cura* in den vorherigen Iterationen nicht umgehen ließ, sollte dieser Stand des Modells nun zu Evaluationszwecken gänzlich ohne Stützmaterial gedruckt werden.

### 6.2 Hard- und Software

#### Visualisierung

Die erste funktionsfähige Fahrzeugsoftware bildet die Grundlage für die Implementation der Visualisierung. Die Basis hierfür ist das Webframework *Flask*. Mit diesem wird eine REST-Schnittstelle entworfen, welche die gesendeten Fahrzeugdaten empfängt.

In einer Konfigurationsdatei werden die verschiedenen Fahrzeugtypen und die zugehörigen Fahrzeugeigenschaften gespeichert. Die Eigenschaften umfassen eine eindeutige ID, den Namen des Typs, den Radius der Räder, die Anzahl an Magneten in diesen und die verursachte Umweltverschmutzung pro gefahrenem Zentimeter. Somit muss der Programmcode auch bei zukünftigen Änderungen nicht angepasst werden und verschiedene Fahrzeugtypen können einfach eingefügt und abgebildet werden. Wenn ein Fahrzeug sich bewegt, sendet es die Anzahl an erfassten Magnetkontakte und die ID des Fahrzeugs an den Raspberry Pi. Dieser berechnet im ersten Schritt mithilfe der spezifizierten Werten in der Konfigurationsdatei die verursachte Umweltverschmutzung. Das Ergebnis wird im zweiten Schritt mit einem Zeitstempel versehen und gespeichert. Da die Messwerte nur wenige Sekunden aktuell sind und keine persistente Speicherung benötigen, wird keine Datenbank aufgesetzt. Die Werte werden lediglich im RAM gespeichert.

Das Programm startet zu Beginn einen weiteren Thread. Dieser ist für die Steuerung der Hardware-Komponenten verantwortlich. Hierfür wird alle  $x$  Sekunden eine Funktion aufgerufen. Diese iteriert zuerst über alle gespeicherten Verschmutzungseinträge und löscht die veralteten. Hierfür wird die aktuelle Systemzeit und der Zeitstempel des Eintrags verglichen. Die Gültigkeit von diesen kann in der Konfigurationsdatei angepasst werden. Danach werden die verbleibenden, gültigen Werte addiert und ergeben die gesamte Verschmutzung. Diese wird an die einzelnen Hardware Komponenten weitergeleitet. Für die Steuerung der Servo Motoren wird ein PWM Signal verwendet. Mit der zuvor berechneten Verschmutzung wird die Rotation von diesen bestimmt. Dabei entspricht  $0^\circ$  keiner und  $180^\circ$  der maximalen Verschmutzung. Für die LEDs wird der WS2812B LED-Streifen der Firma Adafruit verwendet. Dieser wird über die dazugehörige Python Bibliothek angesteuert. Je nach Verschmutzung setzt sich die Farbe aus grün und rot zusammen. Die hierfür verwendete Schaltung ist in Abbildung 15 dargestellt.

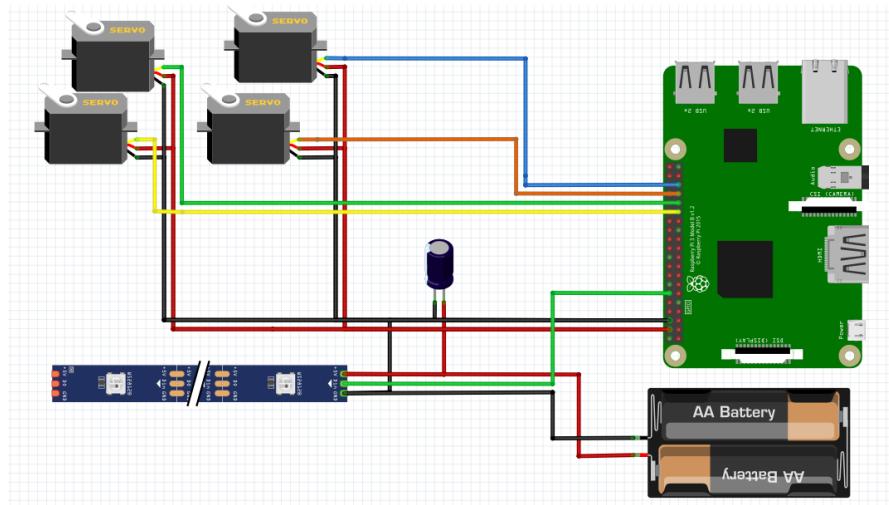


Abbildung 15: Schaltung für den Baum

Diese enthält zusätzlich einen Kondensator und eine Schnittstelle für eine externe Stromversorgung. Der Kondensator soll Schwankungen der Spannung ausgleichen. Das Netzteil wird verwendet, da der Raspberry Pi nicht genug Strom für die LEDs bereitstellen kann. Mit der vorgestellten Schaltung kann das externe Netzteil zusätzlich den Raspberry Pi über den 5V-Pin mit Strom versorgen. Somit wird nur ein Netzteil für alle Komponenten benötigt.

## Fahrzeug

Beim Testen der Visualisierung mit der Fahrzeugschaltung ist ein Problem aufgetreten. Ist das Fahrzeug mit einem USB Kabel an einen Computer angeschlossen, funktioniert die Sensorerkennung. Wird diese nur mit dem Akku betrieben, können keine Sensorwerte erkannt werden. Beim Ausmessen der verfügbaren Spannung zeigt sich, dass im Betrieb mit dem Akku nur 3.3V an dem 5V-Pin verfügbar sind. Wie in Kapitel 4.2 beschrieben, benötigt der Hall-Sensor mindestens 3.7V.

Zum Lösen dieses Problems muss die Spannung angehoben werden. Der erste Lösungsansatz ist ein *Logic-Level-Converter*, welcher zwischen dem Sensor und dem Arduino eingebaut werden kann. Dieser kann Signale in bidirektionaler Richtung zwischen zwei Spannungen, wie zum Beispiel 5V  $\longleftrightarrow$  3.3V, wandeln. Das Problem hierbei ist, dass an einem Pin des Moduls eine 5V Spannung angelegt werden muss, welche nicht verfügbar ist. Der zweite Ansatz ist das Verwenden eines *Step-Up Boost-Converters*, mit welchem das Problem erfolgreich gelöst werden kann. Dieses Modul akzeptiert eine Eingangsspannung zwischen 2 – 24V und hat eine Ausgangsspannung von 5 – 28V. Die Ausgangsspannung kann mit einem Potentiometer eingestellt werden. Die Schaltung des Fahrzeugs wird mit diesem Modul abgeändert. Die neue Version ist in Abbildung 16 dargestellt.

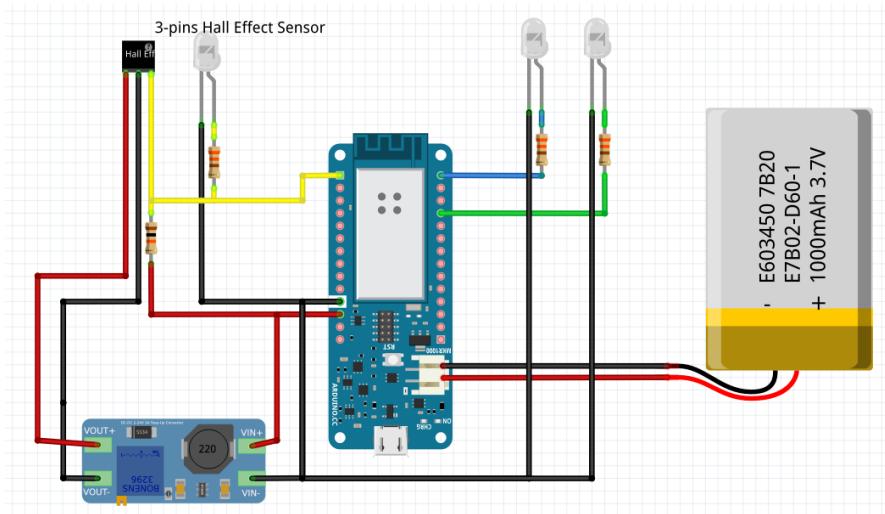


Abbildung 16: Schaltung eines Fahrzeugs mit Boost-Converter

Der +- und *GND*-Pin des Sensors werden an den Output des *Step-Up Boost-Converters* angeschlossen, welcher auf 5V eingestellt wird. Die benötigte Eingangsspannung für das Modul wird vom 3.3V-Pin des Arduinos geliefert. Somit funktioniert die Schaltung sowohl im Betrieb mit einem USB-Kabel als auch mit einem Akku.

### 6.3 Darstellung

Da wegen der Dimensionen der Servomotoren keine zufriedenstellende Lösung mit in den Ästen integrierten Motoren gefunden werden konnte, wurde ein neues Konzept erarbeitet. Während an der Idee, den Baum in einer mit LEDs umrandeten Kiste zu montieren, festgehalten wurde, sah der neue Ansatz vor, die Äste des Baums über Schnüre oder Ähnliches zu bewegen. Diese sollen idealerweise von oberhalb des Baums, bewegt durch die Servomotoren, eine marionettenartige Funktionalität herstellen. Der schematische Aufbau dieser Lösung ist in Abbildung 17 dargestellt.

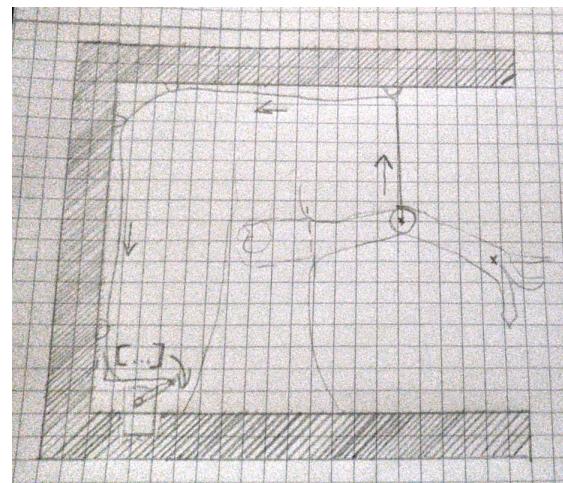


Abbildung 17: Überarbeitetes Darstellungskonzept

Aufgrund der Tatsache, dass sich keine geeignete Umsetzung des Baums mittels 3D-Druck erzielen ließ, wurde ein Bild einer Baumsilhouette gesucht. Mit diesem war es unter der Voraussetzung eines geeigneten Formats möglich, einen qualitativ hochwertigen Baum mit Hilfe des Lasercutters des *FabLabs* herzustellen. Auf dem gleichen Weg war es weiter ein Leichtes, die benötigte Kiste zu realisieren<sup>4</sup>. In Abbildung 18 ist ein erster Eindruck hiervon zu sehen.

---

<sup>4</sup><https://de.makercase.com/#/basicbox>



Abbildung 18: Ausgeschnittener Baum in Kiste

#### 6.4 Ergebnis

Obgleich die gewünschten Freiräume im Fahrzeuginneren erhalten werden konnten, indem ohne Stützmaterial gedruckt wurde, ergaben sich hierdurch neue Probleme. Da nun gar keine Stützen mehr gesetzt wurden, konnten einige Außenflächen wie beispielsweise an der Motorhaube nicht zufriedenstellend gedruckt werden. Auch die Unterseite des Fahrzeugs ist betroffen, wenn auch weniger stark. Beide Stellen sind in Abbildung 19 zu sehen. Des Weiteren wurde deutlich, dass sich durch Zusammenspiel aus modellierten Maßen und den vom Drucker erreichbaren Präzisionsintervallen Aussparungen ergaben, die kleiner als geplant produziert wurden. Hiervon betroffen waren die Scheinwerfer und der Hall-Sensor.

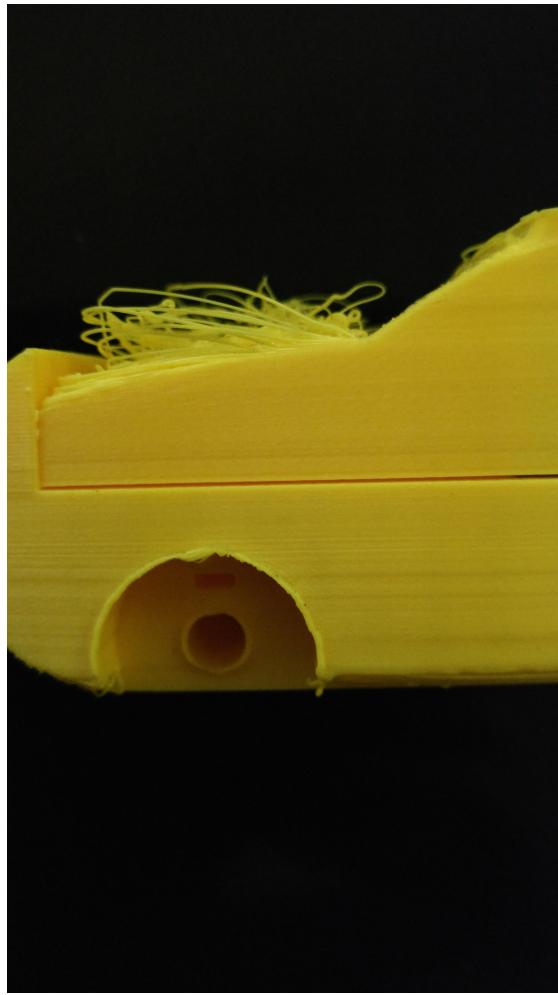


Abbildung 19: Probleme beim Druck ohne Stützmaterial

Während der neue Ansatz zur Visualisierung konzeptionell durchaus vielversprechend schien, so konnte mit den in Abbildung 18 gezeigten Bauteilen nicht fortgefahren werden. Sowohl die Box als auch der Baum selbst wurden aus Holzplatten von 3mm Stärke ausgeschnitten. Insbesondere der Baum war auf Grund der Kombination aus filigranen Blättern und Ästen mit dem zu schwach gewählten Material nicht weiter zu verwenden. Bereits durch den Transport aus den Räumlichkeiten des *FabLab* verlor er eine Vielzahl an Blättern. Weiter hätten die Äste geteilt werden müssen, um das in 6.3 vorgestellte Konzept umzusetzen. Die Box selbst wurde als zu instabil beurteilt, als dass sie die Montage der Motoren und des Raspberry Pi hätte verkraften können.

Die Fahrzeugschaltung wurde um einen *Step-Up Boost-Converter* erweitert. Hierdurch ist der reibungslose Betrieb auch bei Stromversorgung durch den Akku gewährleistet. Des Weiteren wurde auf Basis der Hardware der Visualisierung eine Schaltung erstellt und

entsprechender Programmcode implementiert. Die Kommunikation zwischen den einzelnen Komponenten ist ab diesem Zeitpunkt funktionsfähig, sodass Verschmutzungswerte ausgewertet und dargestellt werden können.

## 7 4. Design-Iteration

### 7.1 Fahrzeuge

Da die vorangegangene Iteration in Kapitel 6 gezeigt hat, dass auch ohne Stützmaterial kein zufriedenstellendes Ergebnis erzielt werden kann, wurde der Versuch unternommen, eine alternative Befestigungslösung zu erarbeiten. Die dem Druck mit Stützmaterial zugrunde liegende Problematik äußert sich in den horizontal durch das Fahrzeug geführten Hohlräumen für die Bolzen. Deshalb wurde der Versuch unternommen, diese nun vertikal anzubringen. Darüber hinaus sollte zusätzliche Stabilität in Längsrichtung gewonnen werden, indem eine Vorrichtung zum “Einhängen“ geschaffen wird. Beides ist in den Abbildungen 20 und 21 zu sehen.

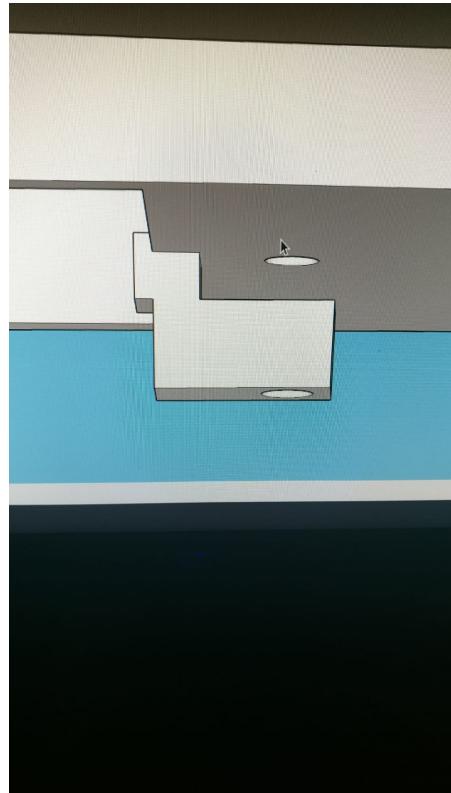


Abbildung 20: Neues Befestigungskonzept (obere Hälfte)

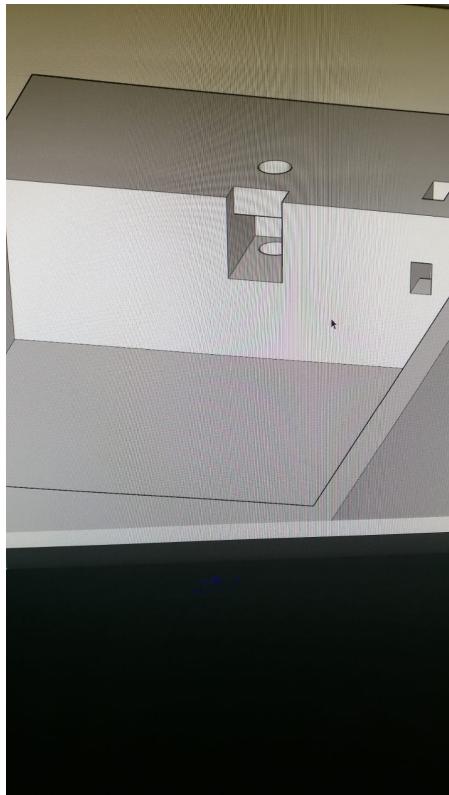


Abbildung 21: Neues Befestigungskonzept (untere Hälfte)

Außerdem wurden die Aussparungen für Scheinwerfer und Hall-Sensor erweitert, um umständlichen Berechnungen und Probendrucken entgegenzuwirken.

## 7.2 Hardware

### Visualisierung

Der Programmcode und die benötigte Hardware für die Visualisierung funktionieren bereits, trotzdem müssen noch kleine Optimierungen vorgenommen werden. Zuerst wird die Funktionsweise des LED-Streifens überarbeitet. Zuvor haben alle LEDs dieselbe Farbe angezeigt, welche sich aus Rot und Grün zusammensetzt. Die Farbe Grün ist hierbei intensiver als die Farbe Rot. Infolgedessen wirken die LEDs auch bei mittlerer Verschmutzung grün statt, wie gewünscht, orange. Erst bei starker Verschmutzung färben sie sich die LEDs orange und danach rot. Mit der neuen Implementierung zeigen die LEDs unterschiedliche Werte an. Je nach Verschmutzungsgrad färbt sich ein Teil der LEDs rot (Abbildung 22). Ist die maximale Verschmutzung erreicht, sind alle LEDs rot. Hierdurch wird die gesamte Verschmutzung eindeutig dargestellt.



Abbildung 22: Visualisierung mit neuer LED-Streifen Implementierung

Im nächsten Schritt wird noch eine Platine für den Raspberry Pi erstellt. Diese kann wie ein Shield auf die Pins gesteckt werden und die Servo-Motoren sowie der LED-Streifen können direkt angeschlossen werden. Dies reduziert die Anzahl an benötigten Kabeln und erhöht die Übersichtlichkeit. Zuletzt wird die Konfiguration des Raspberry Pi überarbeitet. Der Flask-Webserver soll automatisch mit jedem Hochfahren des Systems gestartet werden. Somit muss wie auch bei den Fahrzeugen nur die Stromversorgung hergestellt werden, um den Prototypen zu benutzen.

### Fahrzeug

Für die voll funktionsfähige und getestete Schaltung muss eine Prototyp Platine erstellt werden. Die Basisplatine hat dieselbe Größe wie der Arduino. Somit kann die fertige Platine wie ein Shield auf den Arduino gesteckt werden und verbraucht wenig Platz. Allerdings muss das Layout wegen der begrenzten Größe vorher geplant werden. Eine Skizze hierfür ist in Abbildung 23 dargestellt.

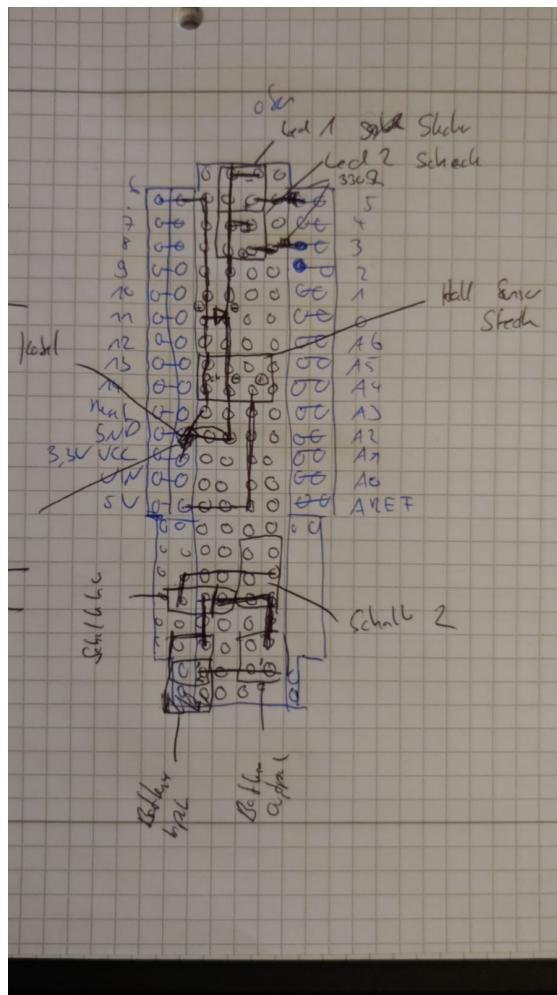


Abbildung 23: Konzept für die Fahrzeugplatine

Die fertige Platine kann auf den Arduino gesteckt werden, welcher zusammen mit dem Akku, dem *Step-Up Boost-Converter*, dem Sensor und den LEDs in das Auto eingebaut wird. Zusätzlich ermöglicht ein Schalter das einfache An- beziehungsweise Ausschalten der Batterie. Alle Verbindungen zwischen den Komponenten sind mit JST-Steckern realisiert. Hierdurch ist es möglich, jede Komponente einzeln auszutauschen, sollte ein Teil defekt sein oder für ein anderes Projekt wiederverwendet werden. Zudem wird so das Zusammensetzen der Fahrzeuge erleichtert, da einige Komponenten wie die LEDs fest verklebt sind. Zuletzt wird noch die Konfiguration der Fahrzeuge optimiert. Zuvor haben diese jede Sekunde einmal mal die erfassten Sensordaten gesendet. Da die durchschnittliche Ausführungszeit der *loop*-Funktion bei ca. 100ms liegt, wird dieser Wert auf 250ms Sekunden reduziert. Die Fahrzeuge senden häufiger Daten und die Bewegungen des Baums werden infolgedessen weicher.

### 7.3 Darstellung

Aufbauend auf den in 6 gewonnenen Erkenntnissen wurden ein neuer Baum und eine neue Box aus 6mm dickem Holz ausgeschnitten. Obgleich auch hier wieder einige Blätter verloren gingen, war das Resultat weitaus stabiler und wurde für die folgenden Montageschritte als geeignet befunden.

Das in 17 gezeigte Konzept sah noch eine Montage der Servomotoren hinter dem Baum, am Boden der Kiste vor. Um die Bewegung einfacher umsetzen zu können und die Montage des Raspberry Pis im Inneren zu erleichtern, wurde aus Holzresten des *FabLabs* eine Art Zwischenboden über dem Baum geschaffen. Die Aufhängung der Äste soll auf diesem Weg ohne weitere Umwege am Rand der Box realisiert werden können.

### 7.4 Ergebnis

Mit den in 7.1 vorgenommenen Änderungen am Modell konnte zwar genügend Platz für die Montage der kleineren Komponenten geschaffen werden, jedoch war der Einsatz einer Heißklebepistole erforderlich, um einen sicheren Halt der Bauteile zu gewährleisten. Auch wenn die vertikale Anordnung der Befestigung die Problematik um das Stützmateri al löste, so konnte die in Abbildung 20 gezeigte Vorrichtung die Montage nicht überdauern. Da eine Erhöhung der Fülldichte bei den ohnehin schon größeren Fahrzeughälften die Druckzeit unverhältnismäßig verlängert hätte und die vorgesehenen Schächte für die Bolzen einwandfrei waren, wurde eine konventionellere Form der Befestigung gewählt. In den Räumen der Analog-Werkstatt des *FabLab* fanden sich geeignete Werkzeuge. Nach kurzer Recherche nach den genauen Maßen wurden die bereits vorhandenen Löcher in der unteren Hälfte und die eigentlich für Bolzen vorgesehenen Schächte mittels eines Bohrers erweitert. In die untere Hälfte wurden nun für M3 Schrauben passende Muttern eingearbeitet. Unter Zuhilfenahme eines Hammers und der Heißklebepistole konnten diese fest mit dem Druck vereint werden. Mit den Maßen des Fahrzeugs wurden entsprechend gekürzte Schrauben hergestellt mit welchen es möglich ist, die Hälften sicher zu vereinigen. Abbildung 24 zeigt das Ergebnis der hier beschriebenen Arbeitsschritte. Auch die transparenten Plastik-Muttern sind hier zu erkennen.



Abbildung 24: Befestigung mit konventionellen Mitteln

Da das erste Fahrzeug die gestellten Anforderungen erfüllt, wurde ein weiteres in Form eines Busses hergestellt. Hierzu war es ausreichend, die äußere Form des Modells zu bearbeiten und einige Details hinzuzufügen, die den Wiedererkennungswert steigern. Hinsichtlich der Befestigung wurde jedoch gleich auf die oben ausgeführte Methode mit Schrauben und Muttern gesetzt.

Auch die stabileren Ausführungen von Baum und Box konnten nun montiert werden. Zunächst wurden die Äste des Baums an ihren jeweils dicksten Stellen mit einer feinen Holzsäge getrennt. An beiden so entstandenen Verbindungsstellen wurden mit einem Bohrer Löcher geschaffen, die in ihrem Durchmesser leicht über die Maße einer M3-Schraube hinaus gehen. Solche Schrauben wurden nun durch die Löcher geführt und die Äste mit Muttern in Position gebracht. Auf diese Weise konnten die Äste beweglich gemacht werden. Der Baum selbst wurde mit Hilfe eines einfachen Winkels an seiner endgültigen Position fixiert. Auf die gleiche Weise wurde der in 7.3 erdachte Zwischenboden montiert. Im Anschluss konnten die Servomotoren hierauf angebracht werden. Um diese etwas kostspieligeren Komponenten wiederverwendbar zu halten, kam erneut die Klebepistole zum Einsatz.

Die Verbindung zwischen den Servomotoren und den Ästen, die sie bewegen sollen, übernahm konventioneller Faden, auch wenn die Befestigung über Knoten unerwartet

zeitintensiv war. Der Grund hierfür war, dass es sich schwierig gestaltete, ein Gleichgewicht zwischen erforderlichem Halt und optisch ansprechender Anfangs- sowie Endposition zu finden.

Weiter fand die Steuerung der Visualisierung ihren Platz auf Abstandhaltern auf dem Zwischenboden, welche ebenfalls mit Schmelzkleber fixiert wurden. In die Außenwand der Kiste wurde auf dieser Höhe ein Loch gebohrt, um die Stromversorgung ins Innere führen zu können. Abschließend wurde der LED-Streifen um den Baum herum montiert, wobei statt der oberen Hülle der Box nun die Unterseite des Zwischenbodens gewählt wurde. Auch hier konnte sicher gestellt werden, dass das Bauteil erneut verwendbar ist, indem die Befestigung mit doppelseitigem Klebeband realisiert wurde. Um den Fokus des Betrachters nicht vom Baum abzulenken, wurde weiter eine Blende installiert, welche die Vorgänge auf dem Zwischenboden verbirgt.

Die Konfiguration der Fahrzeuge und des Baums wurden zu diesem Zeitpunkt lediglich optimiert. Zusätzlich wurden für alle Komponenten Prototyp-Platinen gefertigt, welche die Montage erleichtern und eine bessere Übersichtlichkeit gewährleisten.

## 8 Fazit



Abbildung 25: Fertiger Prototyp

Der im Zuge der vier genannten Iterationen entwickelte und geschaffene Prototyp, zu sehen in Abbildung 25, wird nachfolgend mit Bezug auf die Projektidee bewertet. Mit den beiden Fahrzeugen ist eine anfassbare Schnittstelle zum System realisiert worden. Die Fahrzeuge sind mittels 3D-Druck hergestellt und mit je einem Arduino ausgestattet. Hierdurch ist die drahtlose Kommunikation mit dem dahinterstehenden System möglich. Die Kombination aus der verwendeten Hardware und dem Herstellungsprozess der Fahrzeuge ermöglicht es, den gewünschten Formfaktor zu erreichen. Trotz der verbauten Hardware sind die Fahrzeuge leicht und können auch von Kindern in eine Hand genommen und bedient werden. Zur Inbetriebnahme muss lediglich die Stromversorgung der Fahrzeuge mittels eines Schalters hergestellt werden. Die Arduinos sind so programmiert und konfiguriert, dass sie automatisch die Kommunikation mit dem dahinterliegenden System aufnehmen.

Dieses System übernimmt Empfang und Verarbeitung der übermittelten Daten sowie die Koordination der Darstellung. Obwohl die ursprüngliche Idee der Verwendung eines Displays verworfen werden musste, wird eine gleichwertige Darstellung auf Basis eines beweglichen Baums erzielt. Auch wenn es so nicht mehr möglich ist, genaue Messwerte anzuzeigen, wird der Umwelteinfluss verschiedener Fahrzeugtypen deutlich. Dieser

Eindruck wurde durch den Kontakt mit projektfremden Personen, unter anderem im *FabLab*, bestätigt. Wie zuvor bei den Fahrzeugen auch muss für den Betrieb lediglich der Raspberry Pi mit Strom versorgt werden.

Abschließend lässt sich festhalten, dass die zu Beginn formulierte Idee trotz angefallener Änderungen erfolgreich umgesetzt werden konnte. Die Komponenten des Gesamtsystems interagieren reibungslos miteinander und können den gewünschten Lerneffekt bieten, beziehungsweise damit zu einer Verhaltensänderung anregen.

## 8.1 Ausblick

Da im Rahmen der Veranstaltung auf Grund der begrenzten Zeit und Ressourcen nur die Entwicklung eines einfachen Prototypen möglich ist, werden im Folgenden etwaige Erweiterungen und Optimierungen vorgestellt.

- **Anpassung des SUV-Modells:**

Das erste SUV-Fahrzeugmodell diente nur zum Herstellen der Funktionalität. Deshalb wurde auf eine detailreiche Modellierung verzichtet. Vor einem weiteren Druck des Modells sollte es überarbeitet werden.

- **Weitere Fahrzeugtypen:**

Da die für die Funktion erforderliche Elektronik einen fest definierten Platzbedarf hat, können weitere Fahrzeuge einfach erstellt werden. Hierzu muss lediglich die äußere Form angepasst und auf dem Arduino der Fahrzeugtyp hinterlegt werden.

- **Baum:**

Für den Baum wurde eine Vorlage aus dem Internet verwendet. Für einen langfristigen Einsatz sollte eine auf den Anwendungskontext zugeschnittene Version erstellt werden.

- **Kiste für den Baum:**

Für eine einfachere Montage und ein stimmigeres Gesamtbild kann die Kiste mit Aussparungen und Einbauhilfen für die Komponenten ausgestattet werden. Somit sind keine manuellen Anpassungen mehr nötig.

- **Display:**

Mit weiteren finanziellen Mitteln kann die Darstellung, entsprechend dem ursprünglichen Konzept, um ein Display erweitert werden.

- **Unterlage:**

Mit dem Hall-Sensor kann die gefahrene Strecke sehr genau ermittelt werden. Stünden eine maßstabsgerechte Karte als Unterlage und das oben genannte Display zur Verfügung, könnten gefahrene Strecken und die Emissionen daraus genau berechnet und abgebildet werden.

- **Platine:**

Da die manuelle Fertigung der Platinen sehr aufwändig ist, sollten diese bei größeren Produktionsmengen professionell hergestellt werden.