

Projektarbeit: The amazing Buggy

Von: Hendrik Riese, Mike Frank Peddinghaus

Fach: Skriptsprachen

Dozent: Prof. Dr. Gawron

Inhaltsangabe:

- Einleitung
- benötigte Teile
- Aufbau des Buggy's
- Analyse des Problems
- Programmietechnischer Teil
- aufgetretene Probleme
- Quellen

Einleitung

Das Ziel dieses Projekts ist der Bau eines fahrenden Models, hier genannt „Buggy“. Dieser soll durch unser Python Programm ausgeführt auf einem Raspberry Pi selbstständig durch ein Labyrinth fahren. Zur Erkennung der Wege berechnet der Buggy den Abstand zu den Wänden auf allen Seiten mithilfe von Ultraschallsensoren, um so auftretende Wände oder Entscheidungsmöglichkeiten zu erkennen und diese zu umfahren.

Benötigte Teile

---Benötigte Teile---

Aufbau des Buggy's

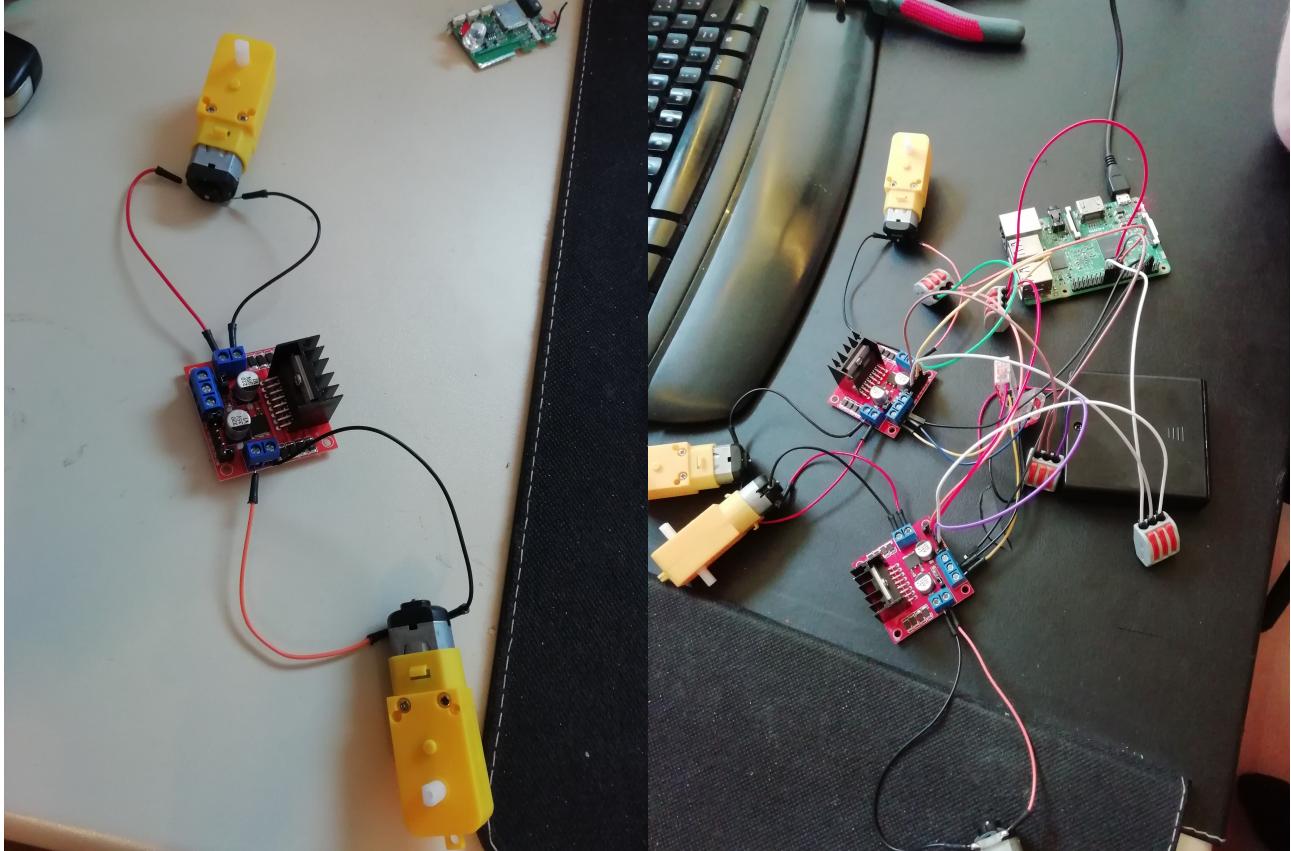


Abbildung 1: Motoren mit Board

Abbildung 2: Verkabelung mit dem Pi

Zum Aufbau des Buggys haben wir zunächst die Motoren mit 2 Male-to-Male Kabeln verlötet und anschließend mit dem Board verbunden(s. Abb. 1). Anschließend haben wir die Boards mit den GPIO-PINS des Raspberry Pi's verbunden und mit Strom versorgt. Den Strom haben wir Anfangs mittels eines Batterie-Packs bezogen, später haben wir uns für Blockbatterien entschieden, da diese weniger Probleme mit der Ausgangsspannung gemacht haben.

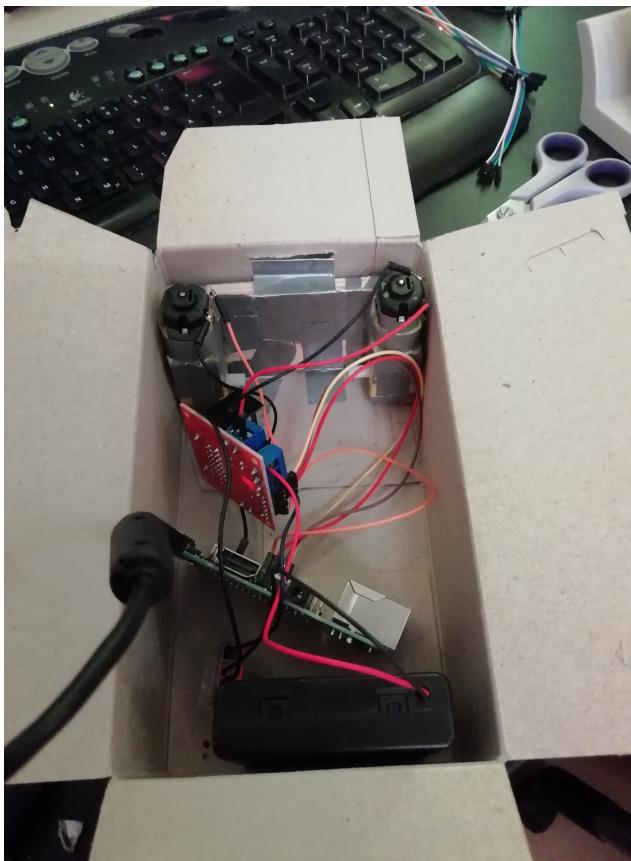


Abbildung 3: Buggy aus Pappe

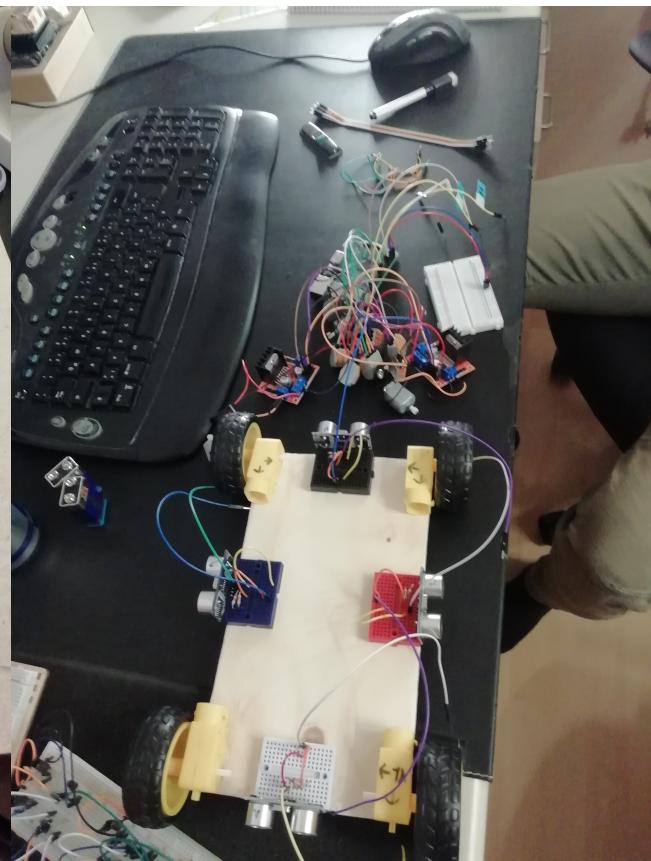


Abbildung 4: Buggy aus Holz

Beim Material, aus dem das Grundgerüst des Buggy's ist, haben wir variiert. Zunächst haben wir Pappe genutzt(s. Abb. 3), da diese sehr leicht ist und zudem relativ dünn, wodurch die Reifen wenig Reibung mit dieser haben. Da wir damit jedoch auf verschiedene Probleme gestoßen sind (s. Kapitel aufgetretene Probleme), haben wir uns für die stabilere Holzmethode entschieden(s. Abb. 4). An dieser hatten wir zudem die Möglichkeit, die Ultraschallsensoren ordentlich und stabil zu befestigen, welche wir für die Berechnungen benötigen.

Analyse des Problems

Das Hauptproblem des Projekts ist die Art und Weise, wie der Buggy seinen Weg durch das Labyrinth finden soll. Dafür sind wir auf verschiedene Algorithmen gestoßen, wie dieses Problem angegangen werden kann.

Ein Algorithmus, den wir implementiert haben, ist die rechte Hand Methode, mit welcher der Buggy sich immer an der rechten Wand orientiert und bei Entscheidungen immer den Weg zu seiner rechten Seite bevorzugt. Gibt es keine Abzweigung nach rechts, so wird der Buggy wenn möglich geradeaus fahren. Erst sobald diese beiden Möglichkeiten nicht vorhanden sind, darf der Buggy auch nach links fahren. Sind keine dieser Möglichkeiten vorhanden, so wendet der Buggy. Diese Methode funktioniert allerdings nur, wenn alle Wände zusammenhängen, sodass man aus Ihnen theoretisch einen Kreis formen könnte, da es ansonsten Ort geben kann, an denen der Buggy im Kreis fahren wird.

Eine weitere Implementierung von uns ist die randomisierte Wegwahl, wo der Buggy an jeder Abzweigung überprüft, in welche Richtungen er fahren kann, und wird dann einen der möglichen Wege per Zufall wählt. Diese Methode ist Laufzeittechnisch allerdings sehr schlecht, da die Möglichkeit, dass man mehrmals den selben Weg fährt, sehr hoch ist.

Es gibt noch weitere Algorithmen, wie z.B. der Trémaux-Algorithmus, der allerdings das Problem aufweist, dass man bei diesem den Boden markieren muss, was unserem Buggy nicht möglich ist.

Programmiertechnischer Teil

---Codeteile zeigen und kurz erklären---

Ablauf des Projekts.

---Ablauf---

aufgetretene Probleme

Das Projekt bringt eine Sammlung von Problemen mit sich, die sich neben der Software abspielen.

So haben wir anfangs Anstelle von der genutzten Holzplatte einen Pappkarton genutzt. Dieser war jedoch zu unstabil, sodass sich der Karton gebogen hat und dadurch die Reifen oftmals Kontakt zu der Pappe hatten. Dadurch kam es zu Widerständen, die entweder dazu führten, dass die Seiten nicht mehr gleich schnell waren und der Buggy dadurch nicht mehr geradeaus fuhr oder die Motoren aufgrund einer eingebauten Sicherung stoppten.

Ein weiteres Problem war das Befestigen der Motoren. So musste erstens darauf geachtet werden, dass diese einen gewissen Abstand zur Holzplatte benötigen, um das vorangegangene Problem der Widerstände vorzubeugen, und zweitens die Motoren exakt gerade und in selber Höhe auf dem Brett liegen, da es sonst passieren kann, dass ein Reifen in der Luft schwebt und somit der Buggy erneut nicht geradeaus fahren kann.

Des Weiteren gab es das Problem, dass die Ultraschallsensoren zwischenzeitlich falsche Messwerte rausgegeben haben. Dies ist dahingehend problematisch, da wir über diese Messwerte überprüft haben, ob es an der jeweiligen Stelle eine Wand gibt oder nicht. Als Lösungsansatz haben wir uns entschieden, das wir uns nicht nur auf einen Messwert verlassen, sondern den nachfolgenden Wert daraufhin ebenfalls überprüfen, um somit das Risiko, auf falsche Werte zu reagieren, einzuschränken.

Quellen

---Quellen---