

PiCar Dokumentation

Student: Patrick Furtwängler

Python Vorkenntnisse: Keine

Beschreibung:

Ein kleines Auto mit dem Raspberry Pi und Python Programmieren, welches selbst fahren kann, aber auch fernsteuerbar ist. Hierzu habe ich ein Kit von Sunfounder gekauft, welches einen Ultra Sonic Distance Sensor nutzt und bereits mit dem im Kit enthaltenen Code, etwas von selbst fährt. Dieses würde ich gerne erweitern, vor allem würde ich gerne die Funktion einbauen, es fernzusteuern.

In dem Kit ist noch ein Line Tracker enthalten, inwiefern ich diesen Anbinde, kann ich noch nicht sagen.

Grün - umgesetzt

Rot - nicht umgesetzt

Orange – so halb umgesetzt

In-Scope:

- Autonomer Modus erweitern/verbessern
- Fernsteuerung
- Wechsel zwischen Modus

Out-of-scope:

- Voll Autonomes fahren
- Objekterkennung

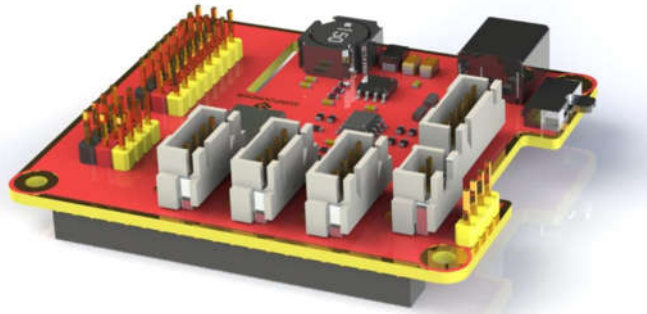
Nice-to-have:

- Soundmodul
- Line Tracker Einbindung als eigener Modus

Zielplattform: Raspberry Pi

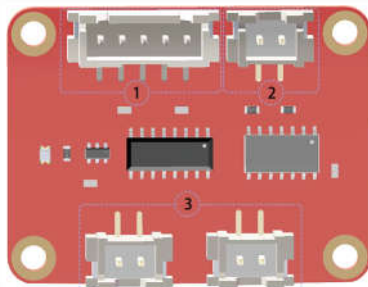
Externe Komponenten:

- **Raspberry 3b**
- **Sunfounder Picar S kit**
 - **3 Extension Boards**
 - **Hauptboard Robot HATS**
Speziell für 40-pin Raspberry Pi, wird direkt draufgesteckt und liefert dem Raspberry Pi Strom über die GPIO ports

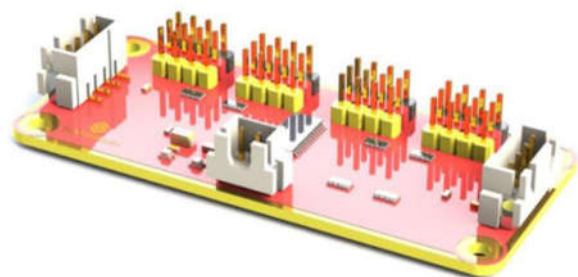


2 zusätzliche Boards, die für die Spannungswandlung der Motoren zuständig sind

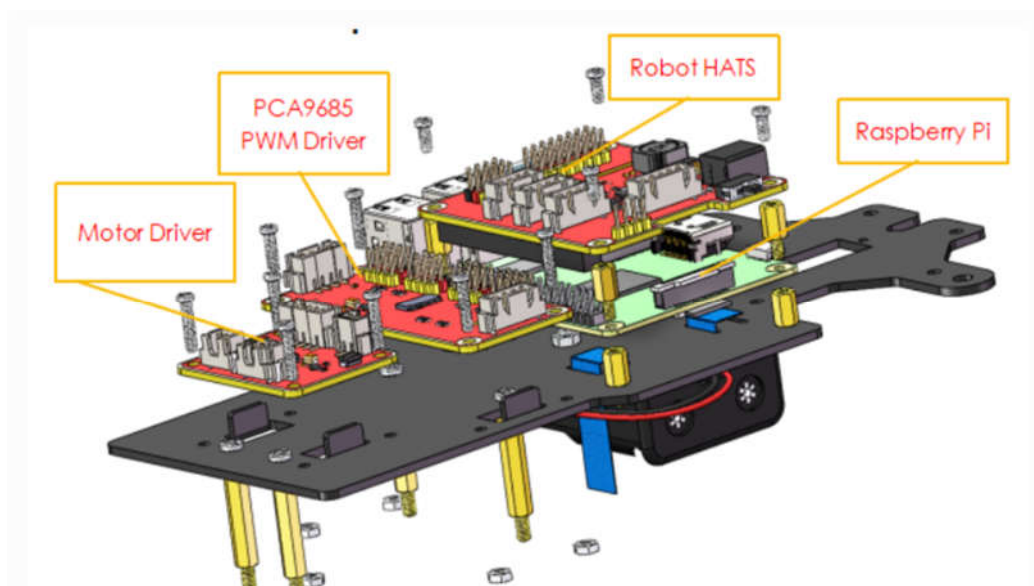
Motor Module



PCA9865



Aufbau:



- **Ultraschall Sensor (SF-SR02)**

Eigenentwickelter Sensor von Sunfounder

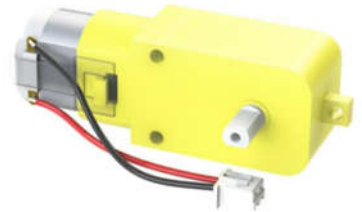
Er hat die Besonderheit für Trigger und Echo nur

Einen Anschluss zu benötigen



- **Servomotor (SF0006C)**

- **2 Elektromotoren**

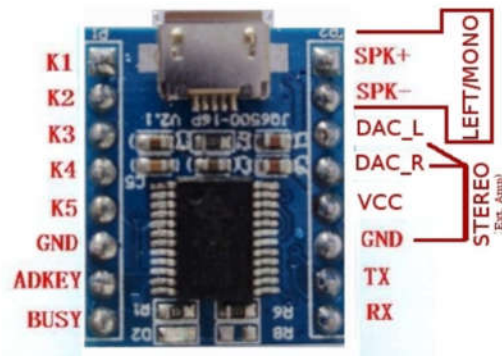


Eigene Externe Module:

- **2x Ultraschallsensoren (HCSR04)**

- **MP3 Player Modul (JQ6500)**

MP3 Modul mit einem Flash Speicher



- **Lautsprecher**

- **zwei Lithium-Ionen Batterien (je 3,7V)**

- **PS4 Controller**



Genutzte Module / Klassen

- **Sunfounder PiCar**

Gibt bereits Code um mit den Motoren und dem Servo zu interagieren

Aufgeteilt in Klassen „forward wheel“ und „backward wheel“

Front Wheels Klasseset zur Steuerung des Servos

Back Wheels Klasse zur Steuerung der Motoren

- **Sunfounder PiCarS**

Klasse „Ultrasonic_Avoidance“ um mit dem Sunfounder Ultraschallsensor zu interagieren und „Ultrasonic_HCSR04“ für die eigenen HCSR04 Ultraschallsensoren

- **pyPS4Controller**

Ermöglicht es bei Python PS4 Controller Inputs abzufangen und mit ihnen zu Arbeiten.

Genutzte Module / Klassen

- **Soundmodul**

Klasse welche mithilfe von GPIO das Soundmodul ansteuert.

- **Mainprogramm**

In diesem File läuft das Programm ab, es werden Objekte zu „front wheel“, „back wheel“, „Controller“ und „Soundmodul“ erstellt und die Funktionen des Controllers so bearbeitet, dass man das PiCar steuern kann, mithilfe des PS4 Controllers. Hier liegt auch der Code um das PiCar von selbst fahren zu lassen.

Eigenleistung:

In dem Projekt habe ich mit dem gegebenen Code von Sunfounder gearbeitet, ich liste auf welche Files aus Eigenleistung stammen:

- Mainprogramm
- Soundmodul
- Ultrasonic Avoidance (Die Klasse für den neuen Sensor aus Eigenleistung, die Klasse zu dem Sunfounder Sensor war gegeben)
- Back wheels (Hier habe ich versucht die hinteren Motoren einzeln anzusteuern, damit das Picar in der Kurve schön fährt und immer das innere Rad langsamer dreht, da es momentan quasi über das vordere Rad rutscht, da beide Motoren hinten gleich schnell drehen. Also um im Prinzip ein Differential zu simulieren, das konnte ich bis jetzt allerdings nicht umsetzen).

Zusammenfassung

Die meisten Spezifikationen wurden erfüllt, das Einzige was ich nur so mäßig hinbekommen habe ist der Wechsel zwischen dem Controller Input Modus und dem Autonomen fahren. Ich habe es mit Polling und einem Interrupt probiert, habe es nicht zufriedenstellend hinbekommen und daher rausgenommen. Wenn man das Picar jetzt startet, ist man im Manuellen Controller Modus und sobald man die Dreiecks Taste drückt, fährt es von selbst. Wenn man zurück möchte, muss man das Programm eben neustarten. Das Linetracker Modul habe ich nicht implementiert, um mich auf die Ultraschall Sensoren zu konzentrieren.

Im Ganzen finde ich das Projekt ist ein Erfolg, klar damit es richtig gut von selbst fährt würde man einen richtig guten Algorithmus benötigen, was aber noch über meiner Gehaltsklasse liegt. Ich weiß jetzt auch warum viele von einem Hardware Projekt zurückgeschreckt haben, da die Elektronik oft Fehler verursacht, mit denen man lange kämpfen muss und zusätzlich noch recht teuer sind.

Das Projekt hat mir aber trotzdem viel Spaß bereitet und mir Python richtig gezeigt, ich werde zukünftig versuchen es zu erweitern.

Quellen:

<https://sparks.gogo.co.nz/jq6500/index.html>

<https://docs.sunfounder.com/projects/picar-s/en/latest/>

https://github.com/sunfounder/SunFounder_PiCar.git

<https://pypi.org/project/pyPS4Controller/>