
Anleitung Projektphase 1

Camp2Code

Florian Edenhofner und Robert Heise
Education4Industry GmbH



Zuletzt aktualisiert: 2024-01-13

Abbildungsverzeichnis

3.1	Download des Raspberry Pi Imagers für Windows.	5
3.2	Auswahl des OS - Raspberry Pi OS (other).	6
3.3	Auswahl des OS - Debian Buster.	6
3.4	Auswahl der SD-Karte.	7
3.5	Konfigurationen	8
6.1	Montage des Raspberry Pi	17
6.2	Anschluss der Stromversorgung. Der Anschluss des Batteriefach erfolgt über einen Stecker. Kabel: 50 mm 2-Pin Jumper Wire (siehe Anhang)	18
6.3	Verbindung einzelner Module. Kabel: 100 mm 5-Pin Jumper Wire, 50 mm 4-Pin Jumper Wire (siehe Anhang)	19
6.4	Anschluss des Servomotors der Lenkung.	20
6.5	Anschluss der Antriebsmotoren.	20
6.6	Finaler Zustand aller Anschlüsse.	21
6.7	Anschluss des Ultraschallmoduls.	21
6.8	Anschluss der Infrarotmoduls.	22
7.1	Auswahl eines eigenen Images im Raspberry PI-Imager.	27
A.1	Übersicht Plattenkomponenten. Im Weiteren werden diese Plattenkomponenten mittels der in der Grafik dargestellten Nummer identifiziert.	30
A.2	Übersicht Servokomponenten für Lenkung. SunFounder SF006C Servo Set: (1) Servomotor, (2) Kipphebel, (5) kürzere Schraube, (6) 2 längere Schrauben. Die Komponenten (3) und (4) werden nicht benötigt, können aber als Ersatzteile dienen.	31
A.3	Übersicht Kleinteile. Gesamtheit und Anzahl aller Schrauben (engl. screws), Muttern (engl. nuts) und Abstandshalter (engl. standoff). (Senkkopfschraube engl.countersunk)	32
A.4	Übersicht Jumperkabel.	33
A.5	Übersicht Leiterplatten und Module. Am Bauteil Robots HATS befindet sich ein Schalter für die Stromversorgung für die Akkus. Das Bauteil Light Follower Module wird nicht benötigt.	34

A.6	Übersicht weiterer Komponenten. Vorderräder (front wheels) und Hinterräder (rear wheels) besitzen unterschiedliche Bauart! Das blaue Bändchen (engl. ribbon) dient der vereinfachten Entnahme der Akkus.	35
A.7	Übersicht Werkzeuge.	35

Inhaltsverzeichnis

Abbildungsverzeichnis	i
Inhaltsverzeichnis	iii
1 Überblick	1
2 Materialien	2
2.1 Lieferumfang	2
2.2 Umgang mit Lithium-Ionen-Akkus	3
3 Installation und Konfiguration des Betriebssystems	4
3.1 Installation des Raspberry Pi OS mit Admin-Rechten	4
3.2 Konfigurationen des RPi	9
4 Softwarevoraussetzungen	10
4.1 Bereitstellen des Repository Camp2Code auf dem RPi	10
4.2 Installation notwendiger Softwarepakete	10
4.3 Installation optionaler Softwarepakete	11
5 Remotezugriff auf Raspberry Pi	13
5.1 Remotezugriff auf den RPi mittels Remoteverbindung und xrdp	13
5.2 Remotezugriff auf den RPi via VSCode	14
5.3 Remotezugriff unter Verwendung von JupyterLab	14
6 Montage und Funktionstest des Modellautos	16
6.1 Montage des RPi am Modellauto	16
6.2 Verbinden der einzelnen Bauteile mittels der Kabel	17
6.3 Test der einzelnen Komponenten	22
7 Weitere Hinweise	26
7.1 Erstellung eines ssh-keys für den RPi	26
7.2 Verwendung eines vorgefertigten RPiOS-Images für die Projektphasen	26
7.3 Verwendung eines vorgefertigtes WinPython	27
7.4 Autostart eines Programms nach dem Booten des RPi	27

8 Problembehandlung	29
A Anhang	30
A.1 Komponentenliste Bausatz	30

1. Überblick

Ziel der Projektphase 1 ist die Entwicklung einer Software für den Betrieb eines Modellautos mittels Python. Zu diesem Zwecke steht das Modellauto (Sunfounder PiCar-S) mit Ultraschallmodul und einer Infrarotsensorleiste zur Verfügung.

Diese Anleitung fasst verschiedene vorbereitende Schritte für den Einstieg in die Projektphase 1 zusammen und kann als Leitfaden dienen, falls einzelne dieser Schritte, z.B. auf Grund eines Defektes, erneut durchgeführt werden müssen.

- Prüfen des Lieferumfangs (falls nicht bereits geschehen)
- Installation des Betriebssystem Raspberry Pi OS Buster (falls nicht bereits geschehen)
- Konfiguration des Raspberry Pi (falls nicht bereits geschehen)
- Softwareinstallation und Download des Repository für das *Camp2code* (falls nicht bereits geschehen)
- Inbetriebnahme und Test des Modellautos (falls nicht bereits geschehen)
- Remoteverbindung zum Raspberry Pi (optional, aber empfohlen)

Zusätzlich enthält dieses Dokument Abschnitte für weitere Hinweise und zur Problembehandlung.

2. Materialien

2.1. Lieferumfang

Für die Verwendung in den Projektphasen stehen Ihnen folgende Materialien zur Verfügung. Bitte prüfen Sie die Vollständigkeit der Materialien.

- 1 x Modelauto Bausatz Sunfounder PiCar-S
- 1 x Ladegerät
- 2 x Akkus (Lithium-Ionen-Akkus 18650 3.7V)
- 1 x Raspberry Pi Model 4
- 1 x Netzteil für Raspberry Pi
- 1 x micro-SD-Karte
- 1 x Klebeband schwarz
- 1 x Klebeband blau
- 1 x USB-Stick (optional)
- 1 x USB-Kartenleser (optional)
- 1 x Kamera mit CSI-Flachkabel (Verwendung in Projektphase 2)
- 1 x Kamerahalterung (Verwendung in Projektphase 2)
- Maus, Monitor, Tastatur
- 1 x Anschlusskabel micro-HDMI/HDMI

2.2. Umgang mit Lithium-Ionen-Akkus

Gehen Sie bewusst und sorgsam mit den Lithium-Ionen-Akkus um. Bei sachgemäßer Handhabung ist der Umgang mit Lithium-Ionen-Akkus sicher. Lithium-Ionen-Akkus sind jedoch brennbar und können giftige Gase oder Flüssigkeiten abgeben.



Entnehmen sie die Akkus dem Modelauto, wenn Sie es nicht verwenden und lagern sie diese sorgsam!

Hinweise:

- Bei unsachgemäßer Benutzung und Lagerung können lithiumhaltige Batterien und Akkus Brände verursachen.
- Verwenden Sie keine defekten, beschädigten, verformten oder aufgeblähten Akkus.
- Batterien und Akkus gehören nicht in den Hausmüll. Entsorgen Sie Altbatterien und Altakkus sachgerecht in den Sammelboxen im Handel oder bei kommunalen Sammelstellen.

Weitere Hinweise finde Sie auf der Webseite des Bundesumweltamtes:

<https://www.umweltbundesamt.de/umwelttipps-fuer-den-alltag/elektrogeraete/lithium-batterien-lithium-ionen-akkus>

3. Installation und Konfiguration des Betriebssystems

3.1. Installation des Raspberry Pi OS mit Admin-Rechten



Um, wie folgend beschrieben, das Betriebssystem auf einer SD-Karte zu übertragen, benötigen Sie eine Computer mit Admin-Rechten!

Um einen Raspberry Pi zu betreiben, muss ein Betriebssystem (engl. Operating System, OS) auf einem boot-fähigen Speichermedium, wie z.B. einer SD-Karte, bereit gestellt werden. Für das Aufspielen des Betriebssystem auf eine SD-Karte kann die Software Raspberry-Pi-Imager verwendet werden. Sie erlaubt Ihnen verschiedene Versionen des Raspberry-Pi-OS in Form eines Images aus dem WWW zu Laden und auf eine SD-Karte zu kopieren. Ein solches Image stellt eine Kopie eines Betriebssystems in Form eines Files dar. (Ein Image können Sie auch selbst erzeugen, um es zum als Backup für ein Betriebssystem zu verwenden.)

Laden Sie den Raspberry-Pi-Imager von <https://www.raspberrypi.com/software/> und installieren Sie ihn (siehe Abbild.3.1). Hierzu benötigen Sie Admin-Rechte auf dem verwendeten Computer. Starten Sie das Programm.

Raspberry Pi OS

Your Raspberry Pi needs an operating system to work. This is it. Raspberry Pi OS (previously called Raspbian) is our official supported operating system.



Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 45-second video](#) to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

[Download for Windows](#)

[Download for macOS](#)

[Download for Ubuntu for x86](#)

Abbildung 3.1.: Download des Raspberry Pi Imagers für Windows.

Im Folgenden wird beschrieben wie Sie mittels des Raspberry-Pi-Imagers **Raspberry Pi OS Version Buster** auf einer SD-Karte installieren. Sie sollen zwingend die **Version Buster** verwenden, da es sonst zu Konflikten verschiedener weiterer Softwarekomponenten kommen kann. (Die zu verwendende Softwarezusammenstellung ist nur auf Buster getestet!)



Alternativ besteht die Möglichkeit, statt der Version Buster, ein bereits vorgefertigtes Image des OS zu verwenden. Dieses Image basiert auf der Version Buster und enthält bereits die notwendigen Softwareinstallationen. Laden Sie in diesem Fall das vorgefertigt Image auf Ihren Computer. Die weitere Installation dieses Images erfolgt analog zur Installation der Version Buster mitels des Raspberry-Pi-Imagers. Zur weitere Informationen siehe Abschnitt 7.

3.1.1. Auswahl des Betriebssystems

Verwenden Sie nicht die aktuelle Version des OS, sondern die ältere Version Buster!

Wählen Sie **Raspberry Pi OS 32bit (other)** (siehe Abbild.3.2).

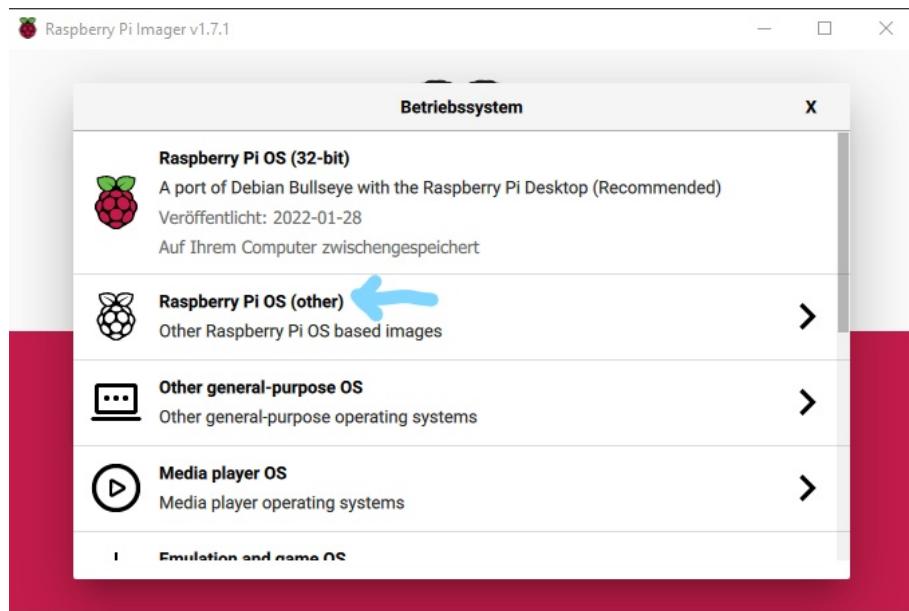


Abbildung 3.2.: Auswahl des OS - Raspberry Pi OS (other).

Wählen Sie **Raspberry Pi OS (Legacy)** (siehe Abbild.3.3).

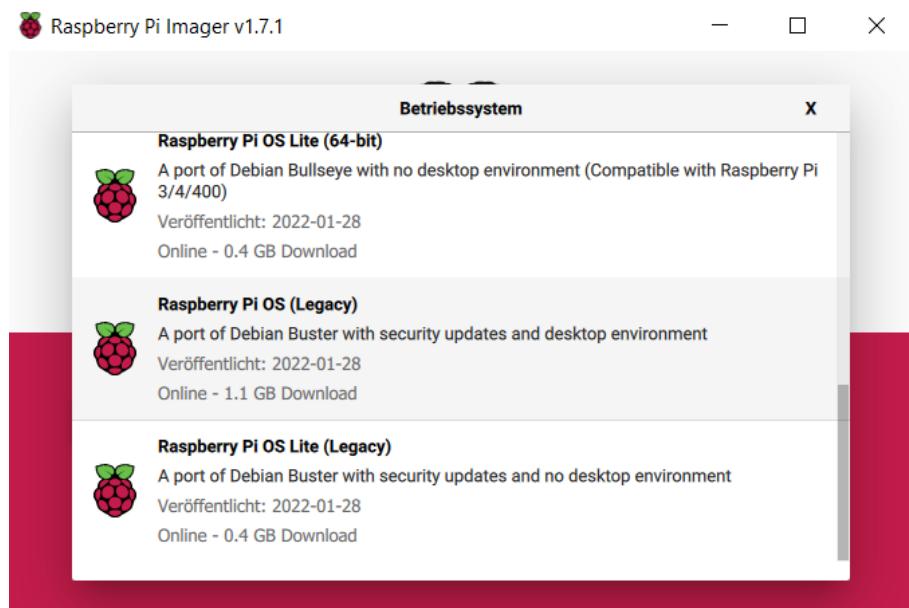


Abbildung 3.3.: Auswahl des OS - Debian Buster.

3.1.2. Auswahl der SD-Karte



Abbildung 3.4.: Auswahl der SD-Karte.

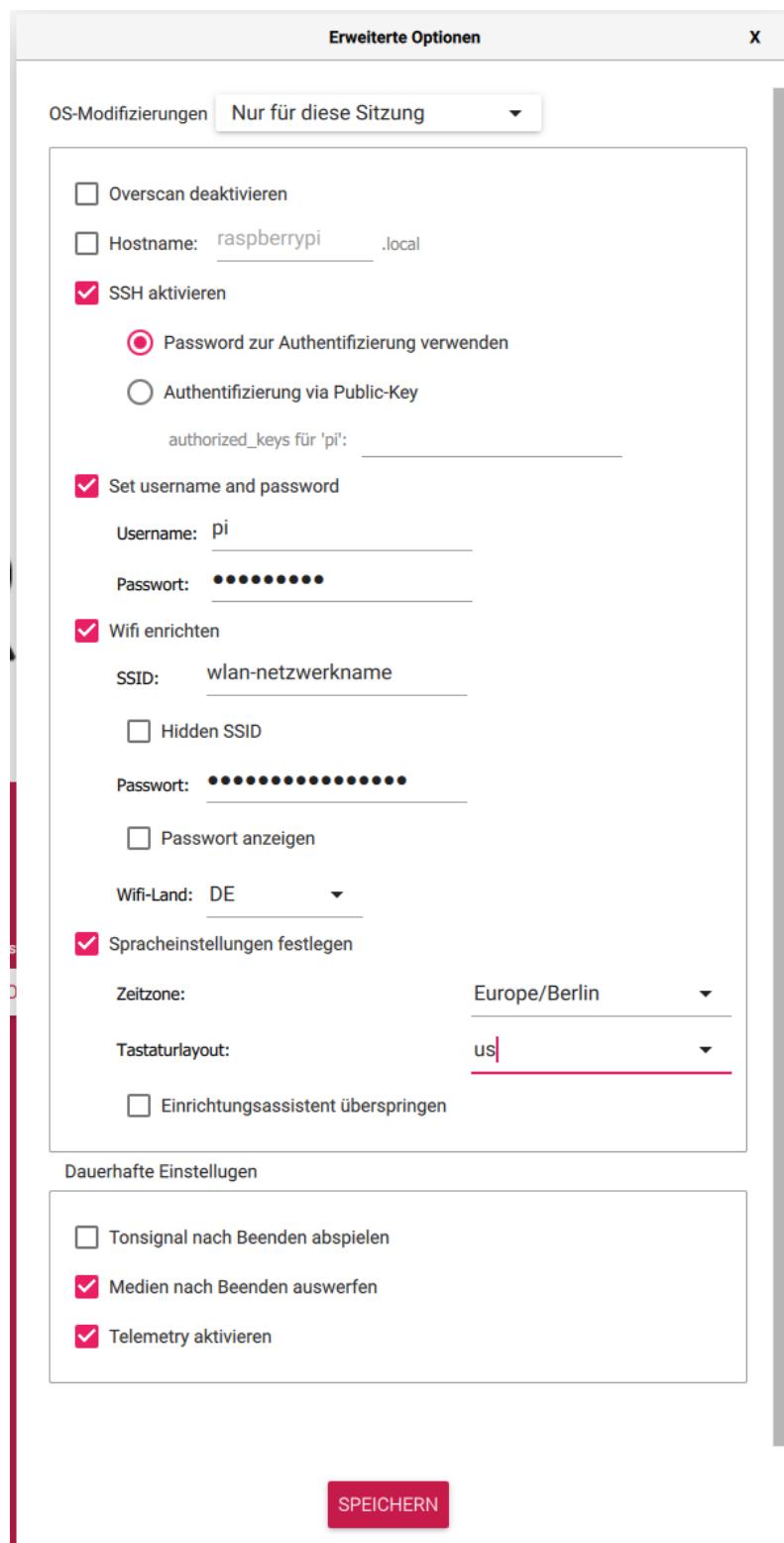
Hier unbedingt darauf achten, dass die richtige SD Karte ausgewählt wird (siehe Abbild.3.4).

3.1.3. Konfigurationen

Einige Konfigurationen des Betriebssystems können nun bereits festgelegt werden (siehe Abbild.3.5). Dies kann jedoch auch später direkt am RPi geschehen.

Bei Bedarf können Sie folgende Einstellungen vornehmen:

- **SSH aktivieren** (mit Passwort) für den Remotezugriff
- **Set username and password**, verwenden Sie als Username am besten *pi*, merken Sie sich das Passwort
- **Wifi einrichten**, damit der Raspberry Pi mit dem WLAN verbunden werden kann
- **Spracheinstellungen festlegen**

**Abbildung 3.5.: Konfigurationen**

3.1.4. Beschreiben der SD-Karte

Klicken Sie nun auf “schreiben” und das Betriebssystem wird heruntergeladen und auf der SD-Karte übertragen.

3.2. Konfigurationen des RPi

Wählen Sie im Menü des Raspberry Pi den Punkt “Preferences/Einstellungen” und klicken auf den Reiter “Raspberry Pi Configuration/Konfigurationen”. Klicken Sie auf den Reiter “Interfaces/Schnittstellen”. Aktivieren Sie die Punkte:

- I2C (dient der Ansteuerungen von Komponenten des Modellautos)
- SSH (für Remotezugriff per SSH)
- Camera (Zugriff auf Kamera in Projektphase 2)

Schließen Sie den Vorgang mit “Ok” ab.

Alternativ kann man den Befehl `sudo raspi-config` in einem Terminal verwenden, um die Einstellungen aufrufen.

4. Softwarevoraussetzungen

4.1. Bereitstellen des Repository Camp2Code auf dem RPi

4.1.1. Klonen des Repositorys

Machen Sie das Repository *camp2code* (<https://gitlab.com/u4i/camp2code.git>) auf Ihrem Raspberry Pi verfügbar. In diesem Repository finden Sie die Basisklassen (basisklassen.py) für den Zugriff auf die einzelnen Fahrzeugbauteile und Sensoren.

Um das Repository auf Ihren RPi zu klonen, führen Sie das folgende Kommando aus. (Genauer gesagt, klonen Sie durch die folgende Anweisung den Branch *project_phase_1* des Repositorys.)

```
1 cd /home/pi/  
2 git clone --recursive -b project_phase_1 https://gitlab.com/u4i/camp2code.git
```

4.1.2. Wechseln des Branches bei bereits vorhandenem Repository

Sollte sich das Repository bereits auf Ihrem RPi befinden, so müssen Sie das folgenden Kommando im Ordner des Repository's ausführen, um in den Branch *project_phase_1* wechseln.

```
1 git checkout project_phase_1
```

Aktualisieren Sie den Branch durch Ausführen des folgenden Kommandos.

```
1 git pull
```

4.2. Installation notwendiger Softwarepakete

In der Projektphase wird folgende Software benötigt.

4.2.1. Python 3

```
1 sudo apt-get update  
2 sudo apt-get install python3
```

4.2.2. Python-Module

Die Python-Module können mit pip3 installiert werden.

Numpy

```
1 pip3 install numpy
```

Falls ein Importproblem auftritt:

```
1 pip3 install -U numpy
```

Pandas

```
1 pip3 install pandas
```

Falls beim Import des Pakets Pandas die Bibliothek `libf77blas.so` fehlt:

```
1 sudo apt-get install libatlas-base-dev
```

Plotly

```
1 pip3 install plotly
```

Dokumentation: <https://plotly.com/python/>

Dash

```
1 pip3 install dash  
2 pip3 install dash_bootstrap_components
```

Dokumentation: <https://dash.plotly.com/>

4.3. Installation optionaler Softwarepakete

Folgende Software ist nicht zwingend notwendig.

Matplotlib

```
1 pip3 install matplotlib
```

Jupyter Notebook

```
1 pip3 install jupyter
```

Falls das nicht klappt:

```
1 pip3 install --upgrade pip
2 pip3 install jupyter
```

Nach der Installation ist ein Neustart notwendig, damit es im Execute Path automatisch hinzugefügt wird und nicht nur .local Verzeichnis installiert ist.

Jupyter Lab

```
1 pip3 install jupyterlab
```

VSCode

```
1 sudo apt install code
```

5. Remotezugriff auf Raspberry Pi

Ein Remotezugriff auf den Raspberry Pi kann es Ihnen ermöglichen bedeutend komfortabler zu arbeiten. Zusätzlich vereinfacht es den Fahrbetrieb des Modellautos erheblich, da das Auto nicht über das Kabel an den Monitor gebunden ist! Um *remote* auf den Raspberry zugreifen zu können, muss dieser mit Ihrem lokalen WLAN verbunden sein. Im folgenden finden Sie verschiedene Möglichkeiten für den Remotezugriff. Welche dieser Möglichkeit für Sie sind umsetzbar sind, wird durch Ihre jeweiliges technisches Setup bestimmt. Präferieren Sie den den Zugriff mittels Remoteverbindung und xrdp.

5.1. Remotezugriff auf den RPi mittels Remoteverbindung und xrdp

Für diese Variante des Zugriffs benötigen Sie das Windowsprogramm **Remotedesktopverbindung**. Der gesamte Bildschirm des Raspberry Pi wird an Ihren PC übertragen und in einem Fenster Ihres PC angezeigt. Dies ermöglicht Ihnen so auf dem Raspberry Pi zu arbeiten, wie Sie es direkt am Raspberry Pi tun würden.

Installieren Sie xrdp auf dem RPi.

```
1 sudo apt-get update
2 sudo apt-get install xrdp
```

Starten Sie unter Windows die Remotedesktopverbindung. Geben Sie die IP-Adresse des RPi's in Ihrem WLAN an. Geben Sie dann den Nutzernamen und das Passwort für den Raspberry an. Um die IP-Adresse Ihres RPi's zu ermitteln können Sie das folgende Kommando im Terminal Ihres RPi's nutzen.

```
1 hostname -I
```

Um eine Remoteverbindung zum Raspberry Pi aufzubauen, muss die grafische Benutzeroberfläche des Raspberry Pi aktiv sein und entsprechend nach dem Booten gestartet werden. Dies geschieht normalerweise nur, wenn ein angeschlossener Monitor bzw. HDMI-Kabel erkannt wird. Um die grafische Benutzeroberfläche auch ohne angeschlossenen Monitor zu starten, muss in der Konfigurationsdatei /boot/config.txt die Zeile `hdmi_force_hotplug=1` aktivieren. Diese geschieht durch das entfernen des Zeichens # am Beginn der Zeile.

5.2. Remotezugriff auf den RPi via VSCode

Für diese Möglichkeit benötigen Sie den Quellcode-Editor VSCode und dessen Erweiterung Remote Development. Zusätzlich muss es Ihnen möglich sein mittels Ihres PC's eine SSH-Verbindungen zum Raspberry Pi aufzubauen. Dies ist jedoch nicht auf allen betrieblichen PCs möglich!

Sie können VSCode direkt auf ihrem PC installieren oder eine vorgefertigte WinPython-Version nutzen, welches Sie unter <https://downloads.fida.de/data-science/02-C2C/> auf Ihren privaten PC downloaden können. Diese WinPython-Version enthält u.a. VSCode.

Konfigurieren der Remote-Verbindung bei erstmaliger Verwendung:

- Öffnen Sie VSCode
- Installieren Sie die Erweiterung **Remote Development**
- Öffnen Sie mittels **CTRL+SHIFT+P** (STRG+SHIFT+P) die Kommandozeile VSCode's und suchen nach *Remote SSH: Connect to a Host*. Klicken Sie dann auf **+Add New SSH Host**. Geben Sie nun Ihren Nutzernamen (RPi) und die IP-Adresse ihres RPi in der Form “[USERNAME@IP](#)” ein (z.B. [pi@192.168.178.54](#)). (Alternative können Sie statt der IP-Adresse den Namen verwenden unter welchen Sie den RPi im Netzwerk registriert haben.)
- Wählen Sie nun eines der vorgeschlagenen Konfigurationsfiles aus. Typischerweise **C:\User\IhrName\.ssh\config** auf einem Windowsrechner.

Starten der Remote-Verbindung:

- Öffnen Sie mittels **CTRL+SHIFT+P** (STRG+SHIFT+P) die Kommandozeile VSCode's und suchen nach *Remote SSH: Connect to a Host* und wählen den Eintrag für ihren Nutzernamen und Ihren RPi (z.B. [pi@192.168.178.54](#))
- Geben Sie Ihr Passwort für den RPi ein
- Öffnen Sie mittels des VSCode-Exploreres den Ordner Ihrer Wahl

5.3. Remotezugriff unter Verwendung von JupyterLab

In dieser Variante starten Sie Jupyter auf dem Raspberry Pi und greifen von Ihrem PC über Ihre WLAN auf dieses Jupyter-Server zu. Um die Jupyter-App auch für andere Rechner im lokalen Netz freizugeben, muss die App an die IP-Adresse Ihres RPi's gebunden werden. Dies können Sie erreichen indem Sie Jupyter mittels folgenden Kommandos starten.

```
1 jupyter-notebook --ip "192.168.178.54" \
2   --port 8888 --notebook-dir="/home/pi/git/camp2code/" --no-browser
```

IP-Adresse, Port und das Arbeitsverzeichnis des Notebook müssen angepasst werden!

Passwort statt Token setzen für Jupyter-Notebook

Gegenenfalls kann es notwendig sein Jupyter so zu konfigurieren, dass statt einem Token ein Passwort verwendet wird.

```
1 jupyter notebook password
```

Man wird aufgefordert ein Passwort einzugeben. Dadurch wird die Konfiguration von Jupyter geändert und es wird statt eines Tokens ein Passwort verwendet.

Autostart Jupyter-Notebook Um JupyterLab nicht nach jedem Neustart des RPi manuell starten zu müssen, können Sie JupyterLab automatisch nach dem Bootvorgang starten. Dazu muss folgendes Codefragment in die Datei “/etc/rc.local” eingefügt werden (Siehe dazu auch Autostart eines Programm nach dem Booten unter Linux).

Beispiel:

```
1 sleep 10 #seems to be needed. maybe in order to wait for network connection
2 cd /home/pi
3 sudo -u pi /home/pi/.local/bin/jupyter-notebook --ip "192.168.178.54" \
4   --port 8888 --notebook-dir="/home/pi/git/camp2code" --no-browser &
5
6 exit 0
```

IP-Adresse, Port, das Arbeitsverzeichnis des Notebook und der Path zu “jupyter-notebook” müssen gegebenfalls angepasst werden!

6. Montage und Funktionstest des Modellautos

Der Raspberry Pi muss nun entsprechend der Abbildung 6.1 auf dem Auto montiert werden. Danach wird die Komponente Robots HATS mit dem Raspberry verbunden, angeschraubt und es erfolgt der Anschluss der einzelnen Platinen mittels verschiedener Kabel. Im Anhang befindet sich eine Übersicht der einzelnen Bauteile des Modellautos und deren Bezeichnungen.

6.1. Montage des RPi am Modellauto

Der Raspberry Pi wird mittels der Abstandshalter M2.5x8 auf die bereits montierten Abstandhalter geschraubt. Dann wird die Komponente Robot HATS auf die GPIOs des Raspberry Pi gesteckt und mit vier Schrauben M2.5x6 befestigt. Diese Komponente besitzt einen Schalter für die Stromversorgung durch die Batterien.

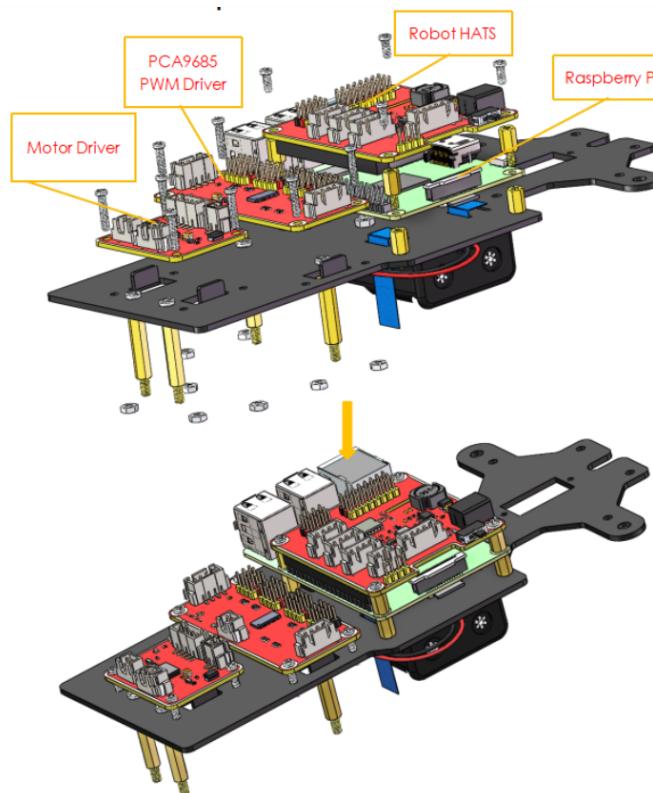


Abbildung 6.1.: Montage des Raspberry Pi

6.2. Verbinden der einzelnen Bauteile mittels der Kabel

Die einzelnen Bauteile werden nun mittels verschiedener Kabel verbunden. Die Abbildungen 6.2 bis 6.5 zeigen diese Anschlüsse und die zu verwendenden Kabel.

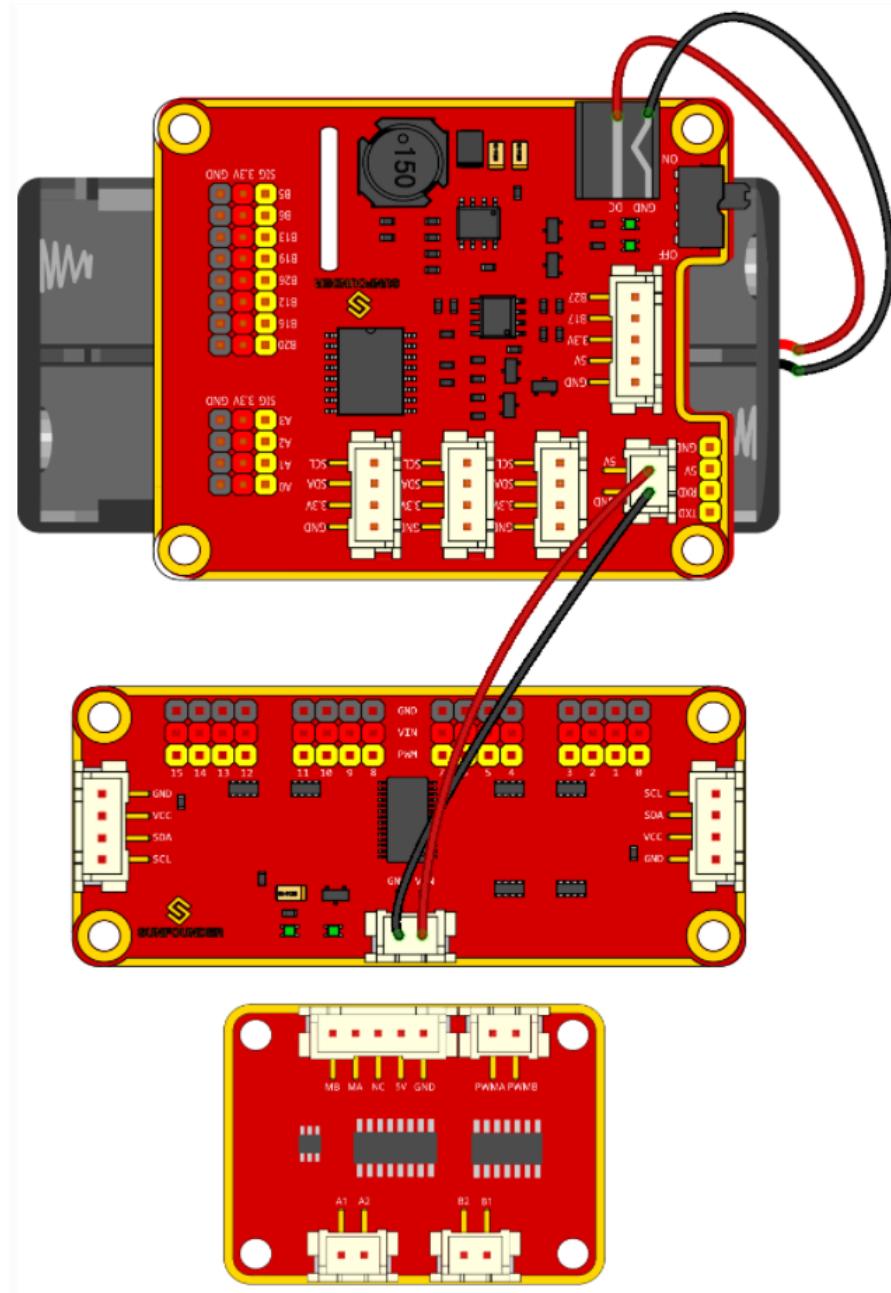


Abbildung 6.2.: Anschluss der Stromversorgung. Der Anschluss des Batteriefach erfolgt über einen Stecker. Kabel: 50 mm 2-Pin Jumper Wire (siehe Anhang)

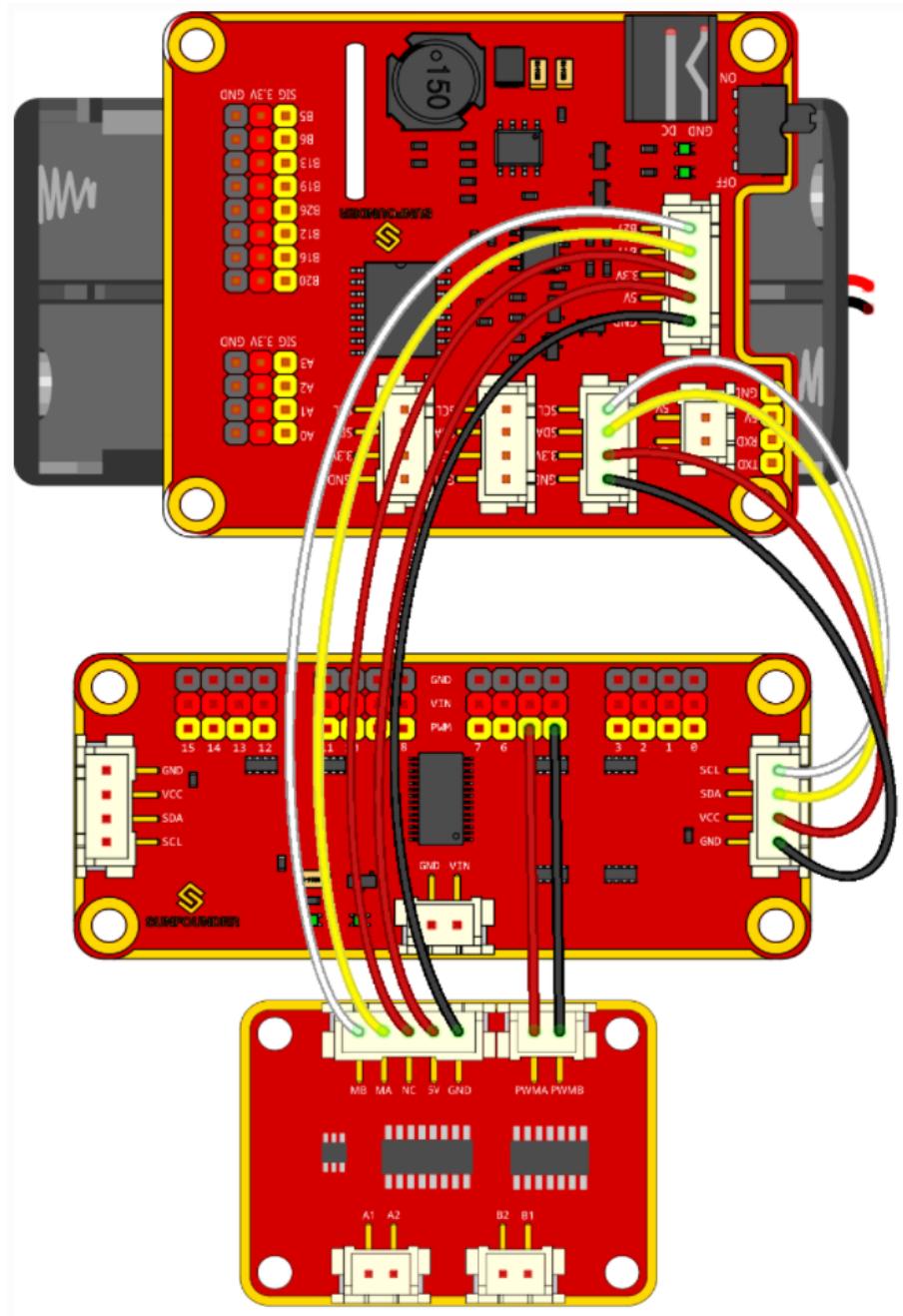


Abbildung 6.3.: Verbindung einzelner Module. Kabel: 100 mm 5-Pin Jumper Wire, 50 mm 4-Pin Jumper Wire (siehe Anhang)

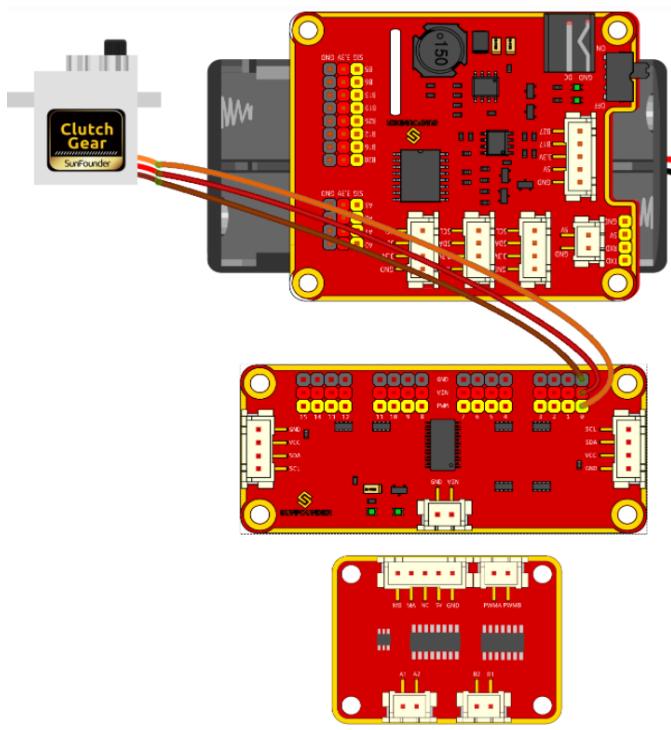


Abbildung 6.4.: Anschluss des Servomotors der Lenkung.

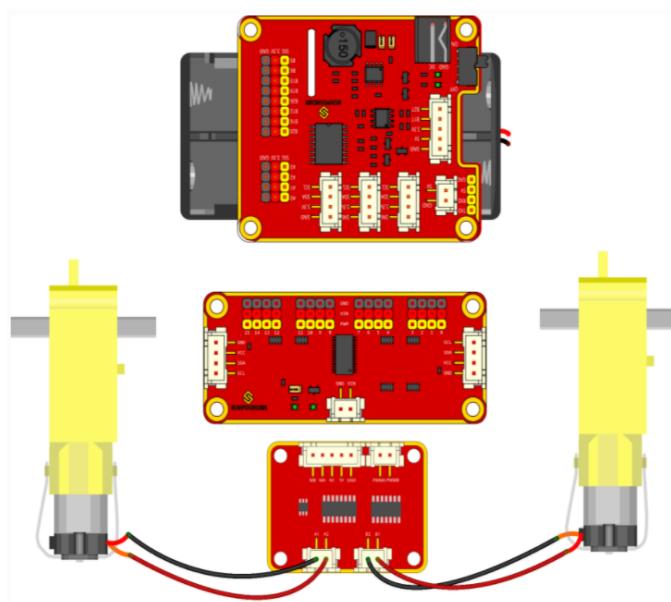


Abbildung 6.5.: Anschluss der Antriebsmotoren.

Die folgende Abbildung 6.5 zeigt die finalen Zustand aller Anschlüsse an den Hauptplatinen und

dient der Kontrolle.

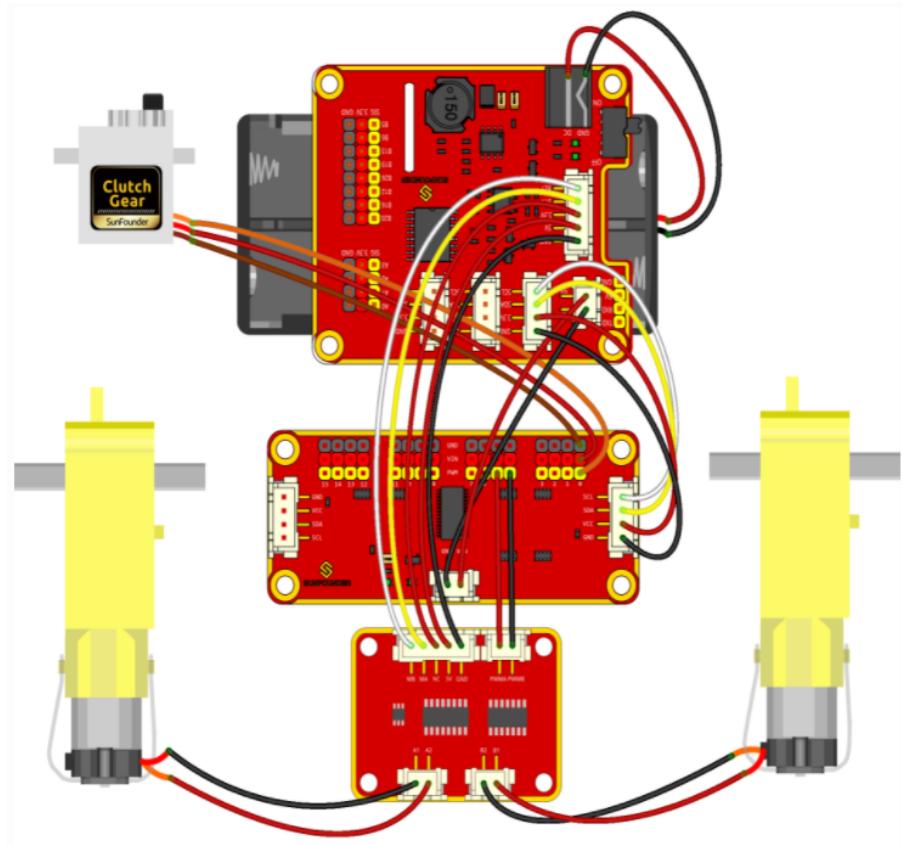


Abbildung 6.6.: Finaler Zustand aller Anschlüsse.

Es folgen Abbildungen für den Anschluss der Sensoren.

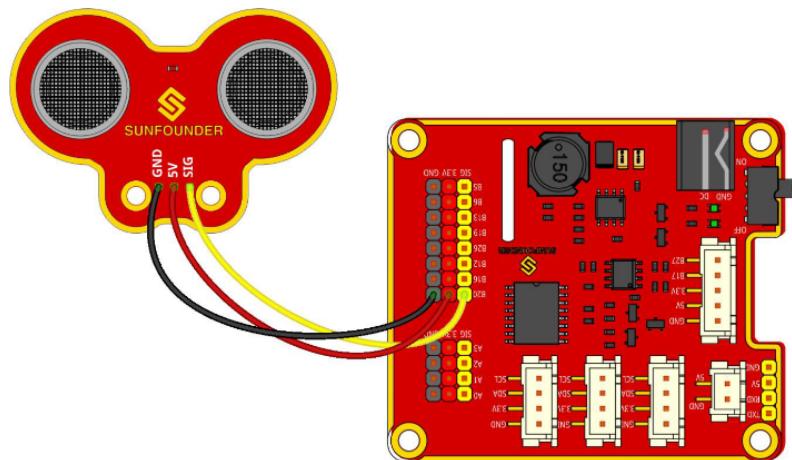


Abbildung 6.7.: Anschluss des Ultraschallmoduls.

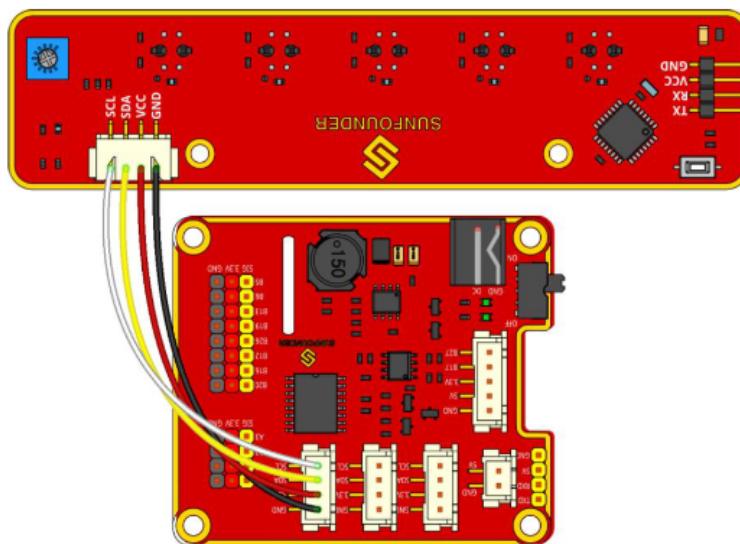


Abbildung 6.8.: Anschluss der Infrarotmodule.

6.3. Test der einzelnen Komponenten

Nun soll die prinzipielle Funktionsfähigkeit der einzelnen Bauteile des Modellautos getestet werden. Da sowohl die Servo der Lenkung als auch die Antriebsmotoren **nur** im Batteriebetrieb mit Energie versorgt werden, muss nun der Batteriebetrieb aktiviert werden. **Legen Sie die Akkus in das Batteriefach und achten Sie dabei auf die Polung der Batterien!** Legen Sie zuvor die Bänder in das jeweilige Fach, um das jeweilige Akku später leichter entfernen zu können. **Schalten Sie den Batteriebetrieb mittels des Schalter am Bauteil Robot Hats ein.** Sie sollten nun zusätzliche leuchtende Dioden sehen.

Für den Funktionstest bzw. den Betrieb des Autos muss die Schnittstelle/interface I2C unter Konfigurationen aktiviert sein!

Das Repository **camp2code** muss ebenfalls auf dem RPi vorhanden sein (siehe auch Abschnitt 4.1). Im Folgenden wird davon ausgegangen, dass Sie dieses Repository als Ordner ‘~/camp2code’ geklont haben. Anderfalls müssen Sie alternative in den entsprechenden Ordner wechseln.

6.3.1. Test des Hinterradantriebs

Führen Sie folgendes Kommando im Terminal aus.

```

1 cd ~/camp2code/software/
2 python3 basisklassen.py --modus 1

```

```
pi@raspberrypi:~/camp2code/Software $ python3 basisklassen.py --modus 1
-- DEMO BASISKLASSEN-----
ACHTUNG! Das Auto wird ein Stück fahren!
Dücken Sie ENTER zum Start.
```

Die Räder des Autos müssen sich nun, wie im Terminal angezeigt, vorwärts und rückwärts drehen. Überprüfen Sie, ob die Drehrichtung mit den angezeigten Vorgaben im Terminal übereinstimmt.



Sollten sich die Räder in die falsche Richtung drehen, so kann die Drehrichtung durch die Software (basisklassen.py) geändert werden. Merken sich diesen Sachverhalt!

6.3.2. Test der Lenkung

Führen Sie folgendes Kommando im Terminal aus.

```
1 cd ~/camp2code/software/
2 python3 basisklassen.py --modus 2
```

```
pi@raspberrypi:~/camp2code/Software $ python3 basisklassen.py --modus 2
-- DEMO BASISKLASSEN-----
Test Vorderräder
```

Das Modellauto führt nun, wie im Terminal angezeigt, Lenkmanöver aus. Überprüfen Sie, ob das Auto korrekt geradeaus lenkt und die eingeschlagenen Lenkwinkel (für rechts und links) gleich sind.



Sollten sich kleine Abweichungen im Lenkwinkel bis etwa 15 Grad zeigen, können diese durch die Software angepasst werden. Sollte die Abweichung im Lenkwinkel über 15° liegen müssen Konfiguration und Montage des Servomotors wiederholt werden.

6.3.3. Konfiguration von Lenkung und Drehrichtung der Antriebsräder

Sowohl die Drehrichtung der Antriebsräder als auch kleine Abweichung im Lenkwinkel können softwaretechnisch, als Funktionalität der Basisklassen, angepasst werden. Diese geschieht durch die Konfigurationsdatei ‘config.json’.

Bearbeiten Sie hierzu die Datei ‘config.json’ im Ornder ‘camp2code/Software’. Die Parameter ‘forward_A’ und ‘forward_B’ können die Wert 0 oder 1 zugewiesen werden. Diese Werte kodieren die Drehrichtung.

Der Wert ‘turning_offset’ kann eine Zahl im Interval von -20 bis 20, mit welchem Sie Abweichungen im Lenkwinkel korrigieren können. Diese Abweichung zeigt sich deutlich, wenn die Räder

auf Geradeaus stehen sollten. Auf diese Weise lässt sich der Lenkwinkel der Servo anpassen. Positive Zahlen beschreiben eine starke Lenkung nach rechts, negative eine nach links. Der voreingestellte Wert beträgt 0. Die Werte für ‘turning_offset’ können Sie zu einem späteren Zeitpunkt erneut anpassen. Die korrekte Wahl des Werte zeigt sich ohnehin erst bei einer Fahrt!

Sollten Sie diese Schritte durchführen, so editieren Sie die Werte in der Datei config.json und überprüfen Sie die Lenkeigenschaften erneut mittels des folgenden Kommandos. Der Modus 5 gibt dabei an, dass die Werte aus der Datei config.json verwendet werden. Es werden Lenkwinkel und Drehrichtung getestet.

```
1 cd ~/camp2code/software/
2 python3 basisklassen.py --modus 5
```

Sollte sich der Lenkwinkel auf die oben beschriebene Weise nicht anpassen lassen, so ist eventuell der Kipphebel, welche an der Welle des Servomotor befestigt ist und die Lenkstange bewegt, falsch montiert oder verrutscht. In diesem Fall muss der Kipphebel von der Welle gelöst und neu befestigt werden. Dabei muss der Servomotor während der erneuten Befestigung auf Geradeaus gestellt und der Kipphebel so befestigt werden, dass er in die Fahrtrichtung zeigt. Um die Welle des Servomotor auf Geradeaus zu stellen, verwenden Sie das folgende Kommando mit dem Modus 0.

```
1 cd ~/camp2code/software/
2 python3 basisklassen.py --modus 0
```

6.3.4. Test des Ultraschall-Moduls

Zum Testen des Ultraschallmoduls führen Sie das Pythonprogramm “basisklassen.py” aus. Führen Sie dazu die folgenden Anweisungen in einem Terminal aus.

```
1 cd ~/camp2code/Software
2 python3 basisklassen.py --modus 3
```

```
pi@raspberrypi:~/camp2code/Software $ python3 basisklassen.py --modus 3
-- DEMO BASISKLASSEN-----
Test Ultrasonic
```

Prüfen Sie nun die Ausgaben des Programms im Terminal, indem Sie z.B. einen geeigneten flachen Gegenstand in verschiedenen Entfernung von 30-100 cm vor den Ultraschallsensor halten.



Sie werden womöglich feststellen, dass die Entfernungsangaben nicht exakt stimmen. Für die Zwecke des Projektes ist die Qualität des Messung jedoch ausreichend.

6.3.5. Test des Infrarot-Moduls

Für einen Funktionstest des Moduls führen Sie das Programm “basisklassen.py” aus.

```
1 cd ~/camp2code/Software  
2 python3 basisklassen.py --modus 4
```

```
pi@raspberrypi:~/camp2code/Software $ python3 basisklassen.py --modus 4  
-- DEMO BASISKLASSEN-----  
Test Infrared
```

Prüfen Sie nun die Ausgaben des Programms im Terminal indem Sie schwarze und weiße Gegenstände unter einzelne der fünf Sensoren halten.



Das Testprogramm gibt eine Liste von jeweils fünf Zahlen aus. Die Elemente der Listen stehen dabei für einzelne Infrotsensoren. Die Liste gibt analoge Messdaten wieder. Sie sollten sensibel auf die Helligkeit des Untergrundes unter einem Sensor reagieren.

7. Weitere Hinweise

7.1. Erstellung eines ssh-keys für den RPi

Die Verwendung von SSH-KEYs vereinfacht die Verwendung eines Remote-Repositorys wie z.B. GitHub.

Das obige Kommando erzeugt u.a. einen Public-Key für Ihren RPi und speichert ihn in der Datei '/home/pi/.ssh/id_rsa.pub'.

```
1 ssh-keygen -o
```

Mittels des folgenden Kommandos können sie diesen Public-Key ausgeben lassen.

```
1 cat /home/pi/.ssh/id_rsa.pub
```

7.2. Verwendung eines vorgefertigten RPiOS-Images für die Projektphasen

Sollte Ihre SD-Karte einmal defekt sein und Sie gezwungen sein das Betriebssystem erneut auf Ihre SD-Karte zu übertragen, so steht Ihnen ein vorbereitetes OS-Image zum Download zur Verfügung, auf welchem die Konfigurationen und Softwareinstallationen bereits durchgeführt wurden. Laden Sie hierzu das File 'fidacar_image.img' von folgender URL <https://downloads.fida.de/data-science/02-C2C/> herunter.

Nutzer: fida

Passwort: fida

Um dieses Image auf die SD-Karte zu übertragen, verfahren Sie analog zur Installation des Betriebssystems unter Verwendung des Raspberry Pi-Imagers, verwenden jedoch das heruntergeladene Image indem Sie bei der Auswahl des Betriebssystem den Punkt 'Eigenes Images' wählen (siehe auch Abbild.7.1).

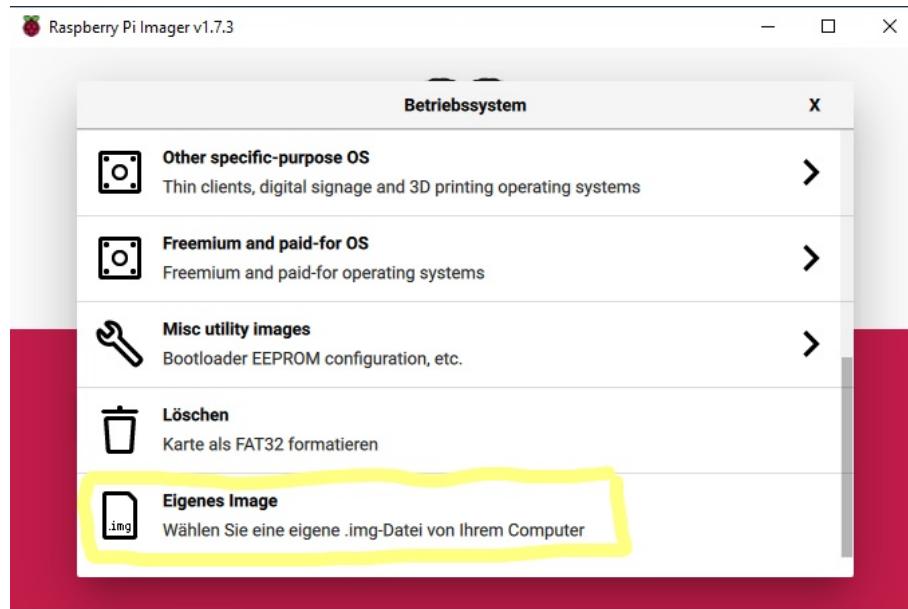


Abbildung 7.1.: Auswahl eines eigenen Images im Raspberry PI-Imager.

7.3. Verwendung eines vorgefertigtes WinPython

Eine für das Camp2Code abgestimmte WinPython-Version kann für die Verwendung auf Ihrem privaten Windows-PC heruntergeladen werden. Diese WinPython-Version beinhaltet bereits alle notwendigen Softwarepakete, sowie u.a. Jupyter Notebook, Jupyter Lab und VS Code. Laden Sie hierzu das File ‘WPy64-3940.zip’ von <https://downloads.fida.de/data-science/02-C2C/> auf Ihren Windows-PC und entpacken dieses. Im Ordner ‘WPy64-3940’ finden Sie alle relevanten Dateien und können von dort z.B. VS Code direkt starten.

Nutzer: fida

Passwort: fida

7.4. Autostart eines Programms nach dem Booten des RPi

Falls es Ihnen nicht möglich ist das Auto bzw. den Raspberry Pi remote (z.B. per W-Lan) anzusprechen, kann es hilfreich sein Programme direkt nach dem Bootvorgang automatisch zu starten. Diese Vorgehensweise ermöglicht es Ihnen das Monitorkabel vom RPi bzw. Auto zu lösen, den RPi zu rebooten und dann direkt das gewünschte Python-Prgramme, z.B. ein Fahrprogramm, zu starten.

Das automatische Ausführen eines Programm nach Abschluss des Bootvorgangs kann unter Linux durch Einfügen des Kommandos in die Datei ‘/etc/rc.local’ erfolgen. Die letzte Zeile muss jedoch den Inhalt ‘exit 0’ aufweisen. In folgenden Beispielen müssen z.B. IP-Adressen, Port und Dateipfade angepasst werden. Im Beispiel wird das Pythonprogramm ‘start_after_boot.py’ mittels des Pythoninterpreters gestartet.

Autostart Python-Programm nach dem Booten unter Linux

Beispiel:

```
1 sleep 2
2 cd /home/pi/git/rpicar
3 sudo -u pi /usr/bin/python3 /home/pi/git/camp2code/software/start_after_boot.py
4
5 exit 0
```

Ein Programm names ‘start_after_boot.py’ ist im Repository camp2code enthalten. Sie können diese Programm wiederum nutzen um ein weiteres Pythonprogramm zu starten. Auf diese Weise können Sie das nach dem Booten zu startende Programm ändern indem Sie, statt der Datei ‘/etc/rc.local’ das Programm ‘start_after_boot.py’ editeren. In der im Repository camp2code vorliegenden Version des Programms ‘start_after_boot.py’ wird das Programm ‘greet.py’ gestartet, welches das Auto kurz mit den Rädern wackeln lässt.

8. Problembehandlung

Dieser Abschnitt dient vorrangig der Vereinfachung der Dokumentation von technischen Problemen und deren Lösung.

Die Remoteverbindung funktioniert nur wenn zusätzlich der Monitor am RPi angeschlossen ist.

Lösung: In der Konfigurationsdatei **/boot/config.txt** die Zeile **hdmi_force_hotplug=1** aktivieren.

Weiteres: https://elinux.org/RPi_Configuration

Man verwendet die RPiOS-Version Bullseye und erhält bei der Verwendung von Dash einen Fehler bzgl Jinja2

Lösung: Flask von Updaten auf Version 2.2.2 Dies führt auch zum Update von Werkzeug auf die Version 2.2.2

```
1 pip3 install Flask --upgrade
```

Falls beim Import des Pakets Pandas die Bibliothek libf77blas.so fehlt:

Lösung:

```
1 sudo apt-get install libatlas-base-dev
```

A. Anhang

A.1. Komponentenliste Bausatz

Die folgenden Abbildungen enthalten alle Bauteile und Werkzeuge des Bausatzes des Modellautos. Bitte prüfen Sie die Vollständigkeit. Kleinteile tragen teilweise eine Beschriftung.

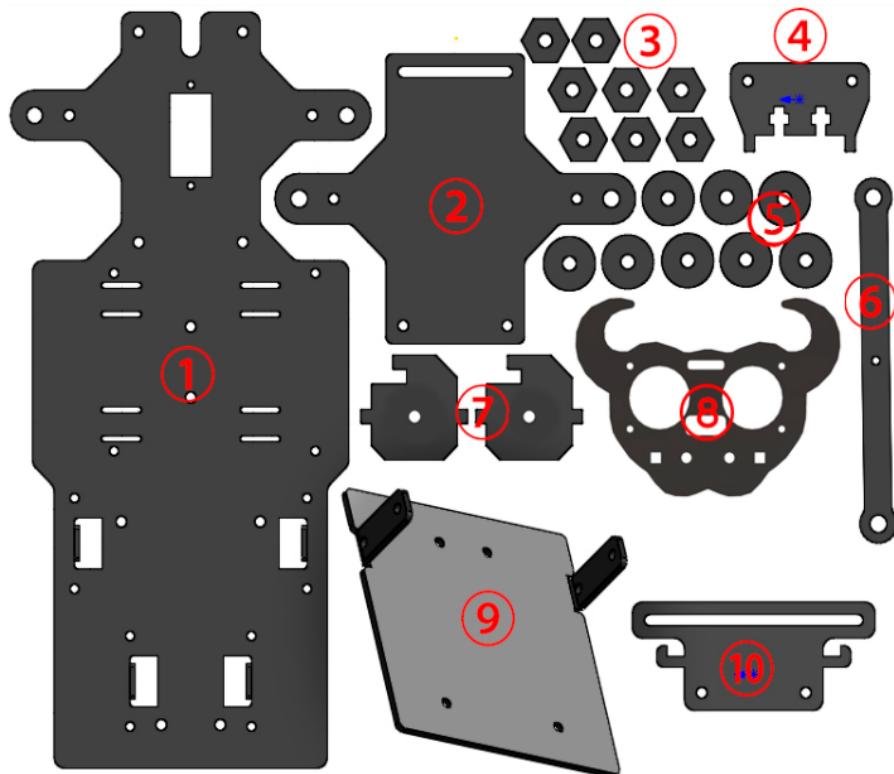


Abbildung A.1.: Übersicht Plattenkomponenten. Im Weiteren werden diese Plattenkomponenten mittels der in der Grafik dargestellten Nummer identifiziert.



Abbildung A.2.: Übersicht Servokomponenten für Lenkung. SunFounder SF006C Servo

Set: (1) Servomotor, (2) Kipphebel, (5) kürzere Schraube, (6) 2 längere Schrauben. Die Komponenten (3) und (4) werden nicht benötigt, können aber als Ersatzteile dienen.

Name	Component	Qty.
M2x8 Screw		2
M2.5x6 Screw		4
M2.5x12 Screw		8
M3x8 Screw		8
M3x8 Countersunk Screw		2
M3x10 Screw		9
M3x25 Screw		4
M4x25 Screw		2
M2 Nut		2
M2.5 Nut		12
M3 Nut		23
M4 Self-locking Nut		2
M2.5x8 Copper Standoff		8
M3x25 Copper Standoff		8
4x11x4 F694ZZ Flange Bearing		2

Abbildung A.3.: Übersicht Kleinteile. Gesamtheit und Anzahl aller Schrauben (engl. screws), Muttern (engl. nuts) und Abstandshalter (engl. standoff). (Senkkopfschraube engl.countersunk)

100mm 5-Pin Jumper Wire		1
50mm 4-Pin Jumper Wire		1
50mm 2-Pin Jumper Wire		1
100mm 2-Pin Jumper Wire		1
200mm 5-Pin Jumper Wire		1
200mm 4-Pin Jumper Wire		1
200mm 4-Pin Jumper Wire		1

Abbildung A.4.: Übersicht Jumperkabel.

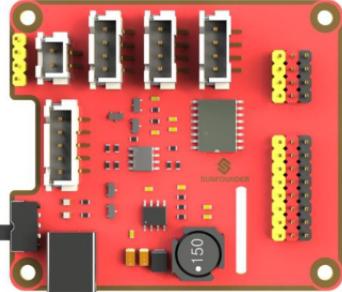
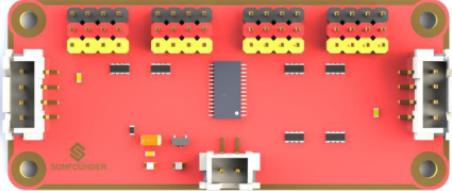
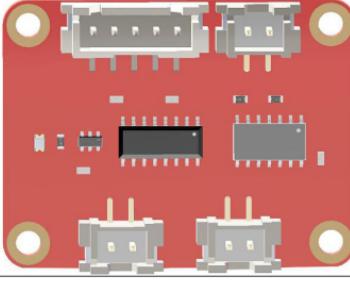
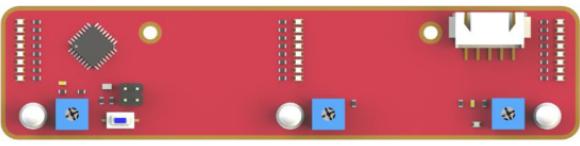
Robot HATS		1
PCA9685 PWM Driver		1
Motor Driver Module		1
5-CH Line Follower Module		1
Ultrasonic Obstacle Avoidance Module		1
Light Follower Module		1

Abbildung A.5.: Übersicht Leiterplatten und Module. Am Bauteil Robots HATS befindet sich ein Schalter für die Stromversorgung für die Akkus. Das Bauteil Light Follower Module wird nicht benötigt.

2x18650 Battery Holder		1
DC Gear Motor		2
Rear Wheel		2
Front Wheel		2
Ribbon (30cm)		1

Abbildung A.6.: Übersicht weiterer Komponenten. Vorderräder (front wheels) und Hinterräder (rear wheels) besitzen unterschiedliche Bauart! Das blaue Bändchen (engl. ribbon) dient der vereinfachten Entnahme der Akkus.

Cross Screwdriver		1
Cross Socket Wrench		1
M2.5/M4 Small Wrench		1
M2/M3 Small Wrench		1

Abbildung A.7.: Übersicht Werkzeuge.