
SunFounder PiCar-X Kit

www.sunfounder.com

05.03.2024

1	Einführung	3
1.1	Die Geschichte der selbstfahrenden Autos	3
1.2	Über PiCar-X	5
1.3	Deep Learning und neuronale Netze	5
2	Montageanleitung	7
3	Servo für die Montage einstellen	9
4	Spiele mit Python	11
4.1	Schnellstartführer für Python	11
4.1.1	Was Brauchen Wir?	11
4.1.2	Betriebssystem Installieren	13
4.1.3	Richten Sie Ihren Raspberry Pi Ein	21
4.1.4	Installiere Alle Module(Wichtig)	42
4.1.5	Aktivierung des I2C-Interface(Wichtig)	45
4.1.6	Servo-Justierung(Wichtig)	47
4.2	Einschalten & Laden	49
4.2.1	Laden	49
4.2.2	Einschalten	50
4.2.3	18650 Batterie	51
4.3	0. Kalibrierung des PiCar-X	52
4.3.1	Kalibrierung von Motoren & Servo	52
4.3.2	Kalibrierung des Graustufenmoduls	55
4.4	1. Lassen Sie PiCar-X sich bewegen	57
4.5	2. Tastatursteuerung	60
4.6	3. Text-zu-Sprache & Soundeffekte	62
4.7	4. Hindernisvermeidung	66
4.8	5. Linienverfolgung	69
4.9	6. Klippenerkennung	73
4.10	7. Computervision	74
4.11	8. Blick auf Dich	81
4.12	9. Videoaufnahme	83
4.13	10. Stierkampf	85
4.14	11. Video-Auto	87
4.15	12. Schatzsuche	91
4.16	13. Durch die App gesteuert	94

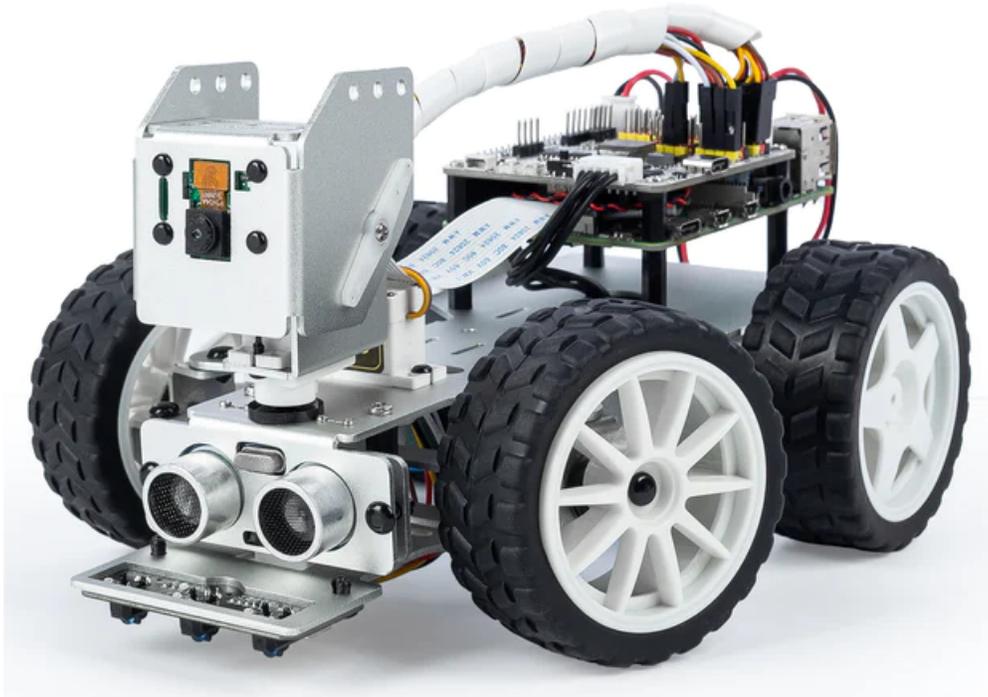
5	Python-Videokurs	99
5.1	Video A1: Einstieg mit dem Raspberry Pi	99
5.2	Video A2: Zusammenbau des PICAR-X	100
5.3	Video A3: Kalibrierung des PiCar-X	100
5.4	Video 1: Motorsteuerung und Lenkkontrolle	101
5.5	Video 2: Steuerung des PiCar-X mit der Tastatur	101
5.6	Video 3: Text in Sprache	102
5.7	Video 4: Hindernisvermeidung mit Ultraschall	102
5.8	Video 5: Graustufen-Linienverfolgung	103
5.9	Video 6: Abgrunderkennung	103
5.10	Video 7: PiCar-X Computervision	104
5.11	Video 8: PiCar-X Blickt Dir Nach	104
5.12	Video 9: Videoaufnahme	105
5.13	Video 10: Stierkampf mit dem PiCar-X	105
5.14	Video 11: PiCar-X als Videowagen	106
5.15	Video 12: Schatzsuche-Spiel	106
6	Mit Ezblock spielen	107
6.1	Schnellanleitung für EzBlock	107
6.2	Installation und Konfiguration von EzBlock Studio	111
6.3	Kalibrierung des Fahrzeugs	111
6.4	Bewegung	115
6.5	Fernsteuerung	119
6.6	Test des Ultraschallmoduls	121
6.7	Test des Graustufenmoduls	123
6.8	Farberkennung	124
6.9	Gesichtserkennung	127
6.10	Soundeffekte	130
6.11	Hintergrundmusik	132
6.12	Sag Hallo	133
6.13	Musikauto	136
6.14	Abgrund-Erkennung	137
6.15	Minenwagen	139
6.16	Minenwagen Plus	142
6.17	Stierkampf	147
6.18	Achtung Fußgänger	149
6.19	Verkehrsschilderkennung	151
6.20	Orientierungslauf	155
7	Anhang	161
7.1	Filezilla Software	161
7.2	PuTTY	163
7.3	Installieren Sie OpenSSH über Powershell	165
8	FAQ	167
8.1	Q1: Nach der Installation von Ezblock OS kann der Servo nicht auf 0° drehen?	167
8.2	Q2: Bei der Verwendung von VNC werde ich darauf hingewiesen, dass der Desktop momentan nicht angezeigt werden kann?	168
8.3	Q3: Warum kehrt der Servo manchmal ohne Grund in die Mittelposition zurück?	168
8.4	Q4: Wo finde ich ein detailliertes Tutorial zum Robot HAT?	168
9	Danke	169
10	Urheberrechtshinweis	171

Danke, dass Sie sich für unser PiCar-X entschieden haben.

Bemerkung: Dieses Dokument ist in den folgenden Sprachen verfügbar.

-
-
-

Bitte klicken Sie auf die jeweiligen Links, um das Dokument in Ihrer bevorzugten Sprache aufzurufen.



Das PiCar-X ist ein von KI angetriebenes autonomes Roboter-Auto für die Raspberry Pi-Plattform, bei dem der Raspberry Pi als Steuerzentrale fungiert. Das 2-Achsen-Kamera-Modul, das Ultraschall-Modul und die Linienverfolgungs-Module des PiCar-X ermöglichen Funktionen wie Farb-/Gesichts-/Verkehrsschilder-Erkennung, automatische Hindernisvermeidung und automatische Linienverfolgung.

Das PiCar-X unterstützt zwei Programmiersprachen: Blockly und Python. Unabhängig davon, in welcher Sprache Sie programmieren, finden Sie detaillierte Schritte, die Ihnen alles von der Konfiguration des Raspberry Pi bis zum Ausführen des relevanten Beispielcodes beibringen.

- *Spielen mit Python*

- Dieses Kapitel richtet sich an diejenigen, die gerne in Python programmieren oder die Python-Sprache erlernen möchten.
- Um das PiCar-X ordnungsgemäß zu betreiben, müssen Sie zuerst einige Bibliotheken installieren.
- In diesem Kapitel werden die Raspberry Pi-Konfiguration und Beispielcodes für das PiCar-X bereitgestellt.
- Eine APP - SunFounder Controller wird ebenfalls bereitgestellt, damit Sie das PiCar-X über Ihr mobiles Gerät fernsteuern können.

- *Mit Ezblock spielen*

- In diesem Abschnitt verwenden Sie die auf Blockly basierende APP Ezblock Studio, die es Ihnen ähnlich wie Scratch ermöglicht, Bausteine zu ziehen und abzulegen, um das PiCar-X zu bewegen.
- Es ist notwendig, die SD-Karte mit dem von uns bereitgestellten Betriebssystem neu zu installieren, das bereits mit der Ezblock-Umgebung ausgestattet ist, bevor Sie mit der Programmierung beginnen. Für diesen Abschnitt empfehlen wir die Verwendung einer neuen oder unbenutzten TF-Karte.
- Ezblock Studio ist für fast alle Gerätetypen verfügbar, einschließlich Macs, PCs und Android-Geräten.
- Ezblock Studio ist eine gute Wahl, wenn Sie zwischen 6 und 12 Jahre alt sind, keine Programmierkenntnisse haben oder das PiCar-X schnell testen möchten.

Inhalt

1.1 Die Geschichte der selbstfahrenden Autos

Schon seit den 1920er Jahren wurden Experimente mit selbstfahrenden Autos durchgeführt. In den 1950er Jahren gab es vielversprechende Versuche, und seitdem hat die Arbeit kontinuierlich Fortschritte gemacht. Die ersten selbständigen und wirklich autonomen Fahrzeuge tauchten in den 1980er Jahren auf, darunter das Navlab und ALV-Projekt der Carnegie Mellon University von 1984 sowie das Eureka Prometheus Projekt von Mercedes-Benz und der Universität der Bundeswehr München von 1987. Seit den späten 1980er Jahren haben zahlreiche Forschungseinrichtungen und große Automobilhersteller, darunter Mercedes-Benz, General Motors, Continental Automotive Systems, Autoliv Inc., Bosch, Nissan, Toyota, Audi, Volvo, Vislab der Universität Parma, die Universität Oxford und Google, funktionierende autonome Fahrzeuge entwickelt. Im Juli 2013 demonstrierte Vislab BRAiVE, ein Fahrzeug, das autonom auf einer für den öffentlichen Verkehr freigegebenen Strecke fuhr. Bis 2019 hatten bereits 29 US-Bundesstaaten Gesetze verabschiedet, die autonome Autos auf öffentlichen Straßen erlauben.

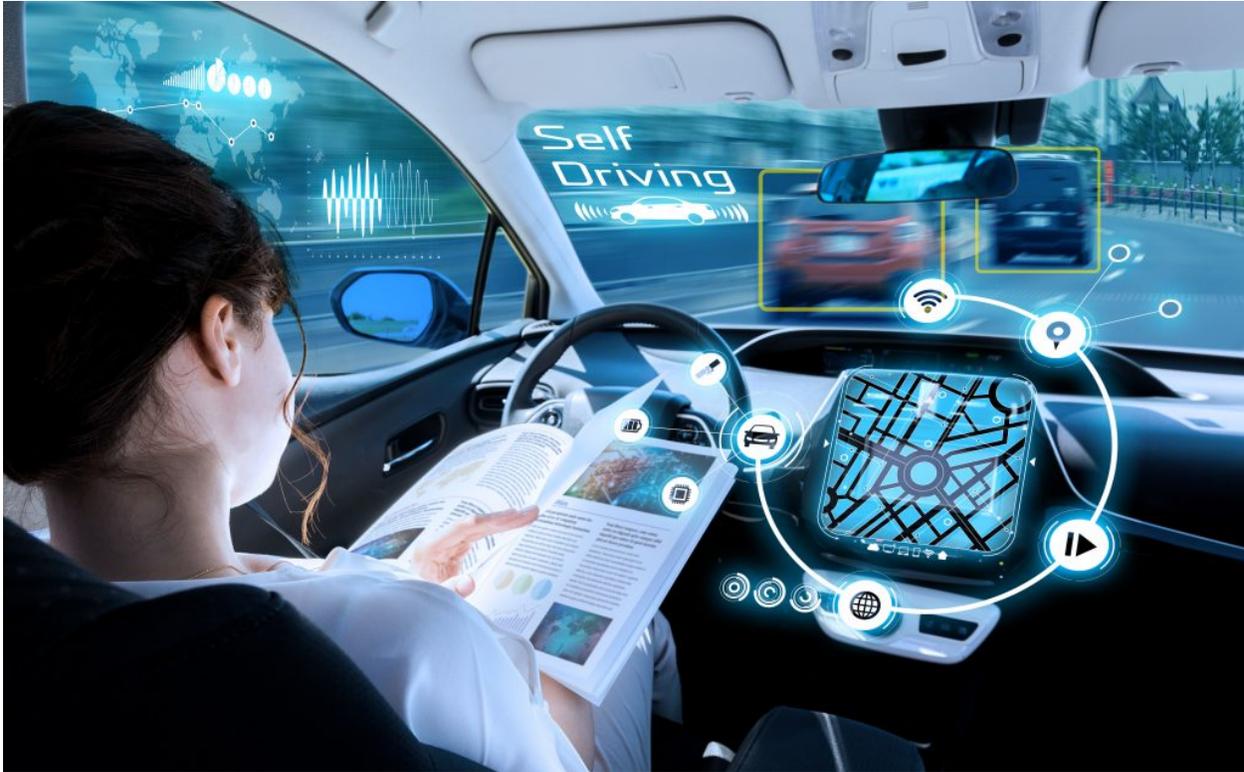
Einige Mitglieder der UNECE und der EU, einschließlich des Vereinigten Königreichs, haben Vorschriften und Regelungen für automatisierte und vollautomatisierte Autos erlassen. In Europa haben Städte in Belgien, Frankreich, Italien und dem Vereinigten Königreich Pläne zum Betrieb von Transportsystemen für fahrerlose Autos, während Deutschland, die Niederlande und Spanien bereits Tests mit Roboterautos im öffentlichen Verkehr erlaubt haben. Im Jahr 2020 sind das Vereinigte Königreich, die EU und Japan bereits auf dem Weg, automatisierte Autos zu regulieren.

- Quelle: [Geschichte der selbstfahrenden Autos - Wikipedia](#)

Heute sind selbstfahrende Autos die nächstgelegene technologische Revolution. Einige Experten prognostizieren, dass bis 2025 Fahrzeuge der Stufe 4 auf den Markt kommen werden. Diese Autos der Stufe 4 werden es den Fahrern ermöglichen, ihre Aufmerksamkeit komplett abzulenken und sich nicht mehr auf den Verkehr zu konzentrieren, solange das System ordnungsgemäß funktioniert.

Referenz für Stufe 4:

- [SAE Stufen der Fahrzeugautomatisierung™](#)
- [ABI Research prognostiziert 8 Millionen Fahrzeuge mit SAE Stufe 3, 4 und 5 autonomen Technologie im Jahr 2025](#)



Die jüngsten raschen Fortschritte in Software (Künstliche Intelligenz, Maschinelles Lernen), Hardware (GPUs, FPGAs, Beschleunigungssensoren usw.) und Cloud-Computing treiben diese technologische Revolution voran.

- Im Oktober 2010 benötigte ein fahrerloser LKW des italienischen Technologieunternehmens **Vislab** drei Monate für die [Reise von Italien nach China](#), insgesamt 8.077 Meilen.
- Im April 2015 fuhr ein von **Delphi Automotive** entworfenes Auto von [San Francisco nach New York](#) über 3.400 Meilen, wobei 99 Prozent dieser Strecke unter Computerkontrolle zurückgelegt wurden.
- Im Dezember 2018 startete **Alphabet's Waymo** einen [Taxi-Service der Stufe 4 in Arizona](#), wo sie seit 2008 fahrerlose Autos testeten. Ohne jemanden auf dem Fahrersitz legten die Fahrzeuge mehr als ein Jahr lang über 10 Millionen Meilen zurück.
- Im Oktober 2020 eröffnete **Baidu** seinen [Apollo Robotaxi selbstfahrenden Taxi-Service](#) in Peking vollständig. Die Fahrrouten umfassen lokale Wohngebiete, Gewerbegebiete, Freizeit- und Industrieparks und bieten ein vollautomatisches Fahrsystem.

Jedoch, trotz der riesigen Mengen an täglich gesammelten Daten, einschließlich Trainingsdaten aus realen Fahrten und simulierten Szenarien, wurde die Komplexität der KI-Modelle für selbstfahrende Autos noch nicht vollständig erfüllt.

Laut dem [RAND-Bericht](#) erfordert das Erreichen des geeigneten Niveaus des autonomen Lernens Trainingsdaten aus hunderten von Millionen oder sogar hunderten von Milliarden von Meilen, um ein Maß an Zuverlässigkeit festzulegen.

Die Zukunft der selbstfahrenden Autos sieht also vielversprechend und spannend aus, doch es stehen noch viele Entwicklungsjahre bevor, bis die Technologie ausgereift genug ist, um auf dem Markt für selbstfahrende Autos vollständig verfügbar zu sein.

Der bewährte Weg, um eine aufkommende Technologie schnell reifen zu lassen, besteht darin, sie durch Minimierung der Markteintrittsbedingungen jedem leicht zugänglich zu machen. Dies ist SunFounders Motivation für den Start von PiCar-X.

SunFounders Ziel ist es, Anfängern und Neulingen, sowie all jenen, die einfach mehr über autonomes Fahren lernen möchten, den Entwicklungsprozess, die Technologie und die neuesten Innovationen im Bereich selbstfahrender Fahr-

zeuge näherzubringen.

1.2 Über PiCar-X

Der PiCar-X ist ein KI-gesteuertes selbstfahrendes Roboter-Auto für die Raspberry Pi-Plattform, wobei der Raspberry Pi als Steuerzentrum dient. Mit der 2-Achsen-Kamera, dem Ultraschallmodul und den Linienverfolgungsmodulen des PiCar-X können Funktionen wie Farb-/Gesichts-/Verkehrsschilder-Erkennung, automatische Hindernisvermeidung und automatisches Linienverfolgen realisiert werden.

Mit der von SunFounder entworfenen Robot HAT-Platine integriert der PiCar-X Links-/Rechtsantriebsmotoren, Servomotoren für Lenkung und die Schwenk-/Neigefunktionen der Kamera und setzt die ADC, PWM und Digital I2C Pins des Robot HAT vor, um Erweiterungen der Standardfunktionen des Raspberry Pi zu ermöglichen. Ein Lautsprecher und ein Bluetooth-Chip sind in den Robot HAT integriert, um Text-zu-Sprache, Soundeffekte oder sogar Hintergrundmusikfunktionen fernzusteuern.

Alle Funktionen des PiCar-X, einschließlich GPIO-Steuerung, Computer Vision und Deep Learning, werden über die Open-Source-Programmiersprache Python, OpenCV's Computer Vision Library-Software und Googles TensorFlow für Deep Learning-Frameworks implementiert. Weitere Software wurde hinzugefügt, um die Fähigkeiten des PiCar-X zu optimieren und dem Benutzer eine nahezu grenzenlose Lernumgebung zu bieten.

1.3 Deep Learning und neuronale Netze

Um mehr über Deep Learning und neuronale Netze zu erfahren, empfiehlt SunFounder die folgenden Ressourcen:

Maschinelles Lernen - Andrew Ng : Dieser Kurs bietet eine umfassende Einführung in maschinelles Lernen, Datenabbau und statistische Mustererkennung.

Neuronale Netze und Deep Learning : Dieses E-Book behandelt sowohl neuronale Netze, ein biologisch inspiriertes Programmierparadigma, das es einem Computer ermöglicht, aus Beobachtungsdaten zu lernen, als auch Deep Learning, ein leistungsfähiges Set von Techniken für maschinelles Lernen in neuronalen Netzen.

Neudenken der Inception-Architektur für Computer Vision : Dieses hochrangige Whitepaper untersucht, wie Benutzer Netzwerke durch den effizientesten Einsatz zusätzlicher Berechnungen über faktorisierte Faltungen und aggressive Regularisierung skalieren können.

Montageanleitung

Bevor Sie den PiCar-X zusammenbauen, überprüfen Sie bitte zuerst, ob alle Teile und Komponenten enthalten sind. Sollten Teile fehlen oder beschädigt sein, kontaktieren Sie bitte sofort SunFounder unter service@sunfounder.com, um das Problem schnellstmöglich zu klären.

Befolgen Sie die Schritte in der folgenden PDF für Montageanleitungen:

[PDF]Teileliste und Montage des PiCar-X.

Montieren Sie den Raspberry Pi Zero W auf dem PiCar-X

Wenn Ihr Hauptplatine ein Raspberry Pi Zero W ist, finden Sie hier die Schritte zur Installation auf dem PiCar-X.

Danach können Sie die Anweisungen im Video unten ab 1:45 fortsetzen, um es zusammenzubauen.

Montage-Anleitungsvideo (Raspberry Pi 4/3/1 Modell)

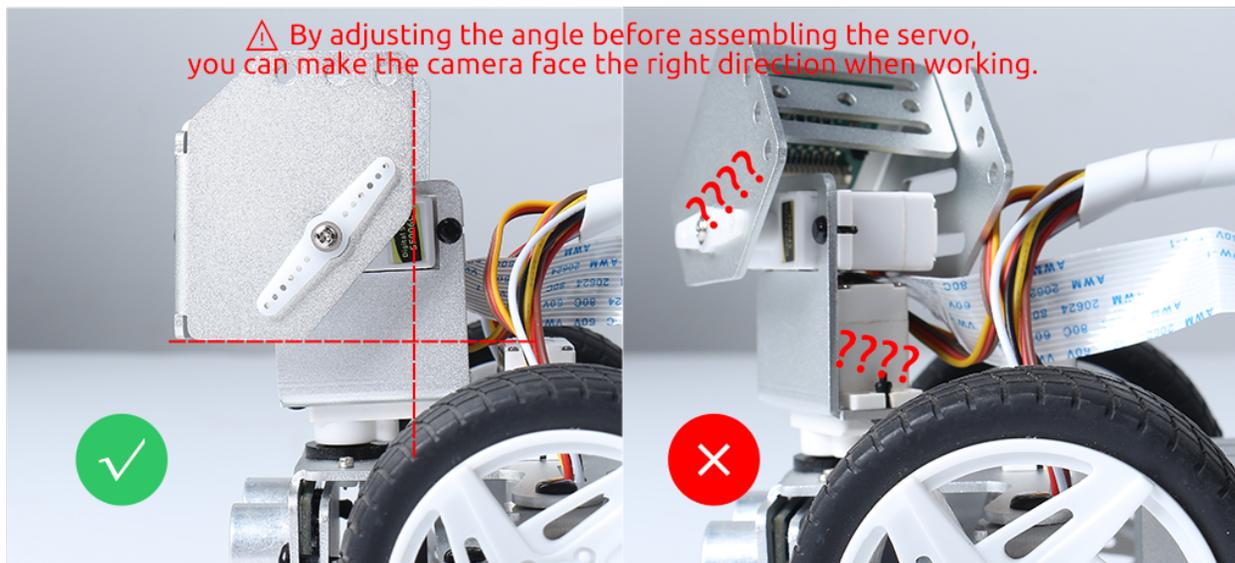
Dieses Video führt Sie Schritt für Schritt durch den Prozess des Zusammenbaus Ihres Roboters.

In diesem Tutorial erfahren Sie:

- **Vorbereitung:** Wir stellen Ihnen alle notwendigen Werkzeuge und Teile vor, sodass Sie bestens ausgerüstet sind, bevor Sie mit der Montage beginnen.
- **Montageschritte:** Wir zeigen jeden Montageschritt systematisch.
- **Tipps und Hinweise:** Während des gesamten Prozesses geben wir Ihnen wichtige Tipps und Tricks an die Hand, damit Sie häufige Fehler vermeiden und Ihr Auto reibungslos funktioniert.
- **Servo Nullstellen:** Bevor jeder Servo festgelegt wird, muss er zuerst auf Null gestellt werden. Die Schritte zum Nullstellen sind: Installieren Sie zuerst das Raspberry Pi OS, dann die benötigten Module und führen Sie dann ein Skript aus (stellen Sie den Winkel aller PWM-Pins auf 0). Danach stecken Sie das Servo-Kabel ein, um den Servo auf Null zu stellen.

Servo für die Montage einstellen

Vor der Montage des Servos muss der Winkel auf null Grad eingestellt werden. Dies liegt daran, dass der Servomotor einen begrenzten Bewegungsumfang hat. Indem man den Winkel auf null Grad einstellt, stellt man sicher, dass der Servo in seiner Ausgangsposition ist und seinen Bewegungsbereich nicht überschreitet, wenn er eingeschaltet wird. Wenn der Servo vor der Montage nicht auf null Grad eingestellt ist, könnte er versuchen, seinen Bewegungsbereich zu überschreiten, wenn er eingeschaltet wird, was den Servo oder das mechanische System, an das er angeschlossen ist, beschädigen könnte. Daher ist es wichtig, den Winkel auf null Grad einzustellen, um einen sicheren und normalen Betrieb des Servomotors zu gewährleisten.



Für Python-Nutzer

Bitte beziehen Sie sich auf *Schnellstartführer für Python*, um die Installation des Raspberry Pi OS abzuschließen und den Winkel der Servos anzupassen.

Für Ezblock-Nutzer

Nachdem Sie das Ezblock-System installiert haben, kann der P11-Pin verwendet werden, um den Servo anzupassen. Weitere Informationen finden Sie unter [*Schnellanleitung für EzBlock*](#).

Für Anfänger und Einsteiger, die in Python programmieren möchten, sind grundlegende Programmierkenntnisse in Python sowie Wissen über das Betriebssystem des Raspberry Pi erforderlich. Um mit der Konfiguration des Raspberry Pi zu beginnen, beziehen Sie sich bitte auf den Schnellstartführer für Python:

4.1 Schnellstartführer für Python

Dieser Abschnitt lehrt Sie, wie Sie das Raspberry Pi OS installieren, das WLAN für den Raspberry Pi konfigurieren und den Fernzugriff auf den Raspberry Pi einrichten, um den entsprechenden Code auszuführen.

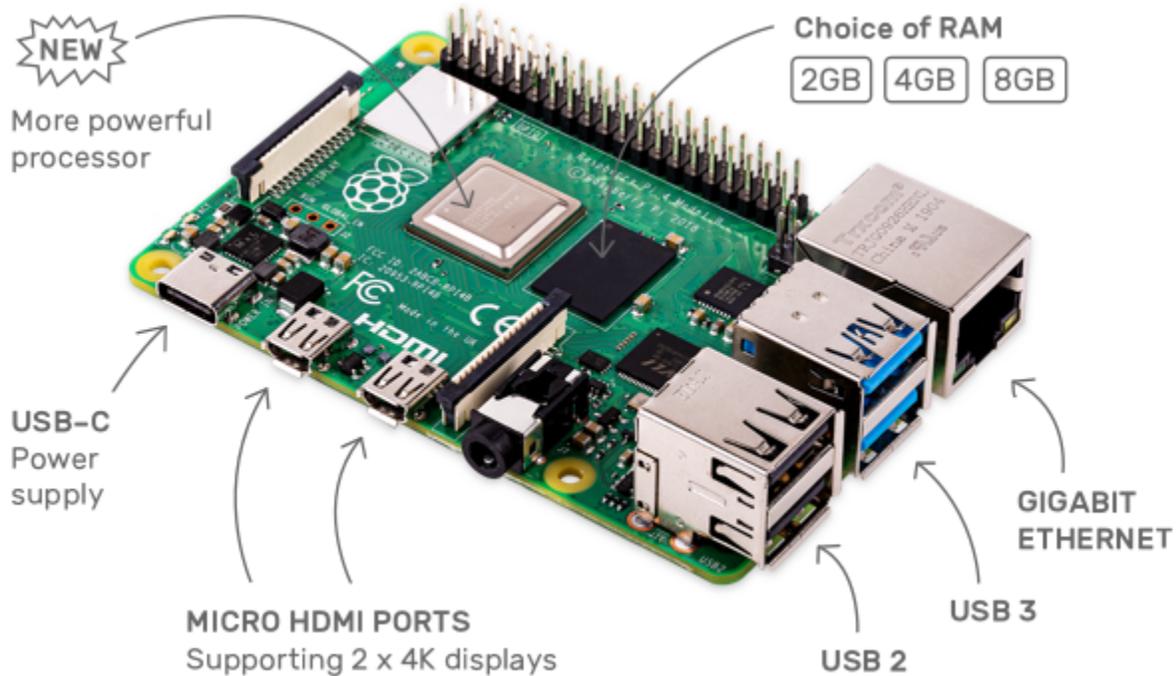
Wenn Sie mit dem Raspberry Pi vertraut sind und die Befehlszeile erfolgreich öffnen können, können Sie die ersten 3 Teile überspringen und dann den letzten Teil abschließen.

4.1.1 Was Brauchen Wir?

Benötigte Komponenten

Raspberry Pi

Der Raspberry Pi ist ein kostengünstiger, kreditkartengroßer Computer, der an einen Computermonitor oder Fernseher angeschlossen wird und eine Standardtastatur sowie eine Maus verwendet. Es ist ein leistungsfähiges kleines Gerät, das Menschen aller Altersgruppen ermöglicht, die Welt der Computer zu erkunden und das Programmieren in Sprachen wie Scratch und Python zu erlernen.



Netzteil

Für den Anschluss an eine Steckdose verfügt der Raspberry Pi über einen Micro-USB-Anschluss (wie er auch bei vielen Mobiltelefonen zu finden ist). Sie benötigen ein Netzteil, das mindestens 2,5 Ampere liefert.

Micro SD-Karte

Ihr Raspberry Pi benötigt eine Micro SD-Karte, um alle seine Dateien und das Raspberry Pi OS zu speichern. Sie benötigen eine Micro SD-Karte mit einer Kapazität von mindestens 8 GB.

Optionale Komponenten

Bildschirm

Um die Desktop-Umgebung des Raspberry Pi zu nutzen, benötigen Sie einen Bildschirm, der ein Fernsehbildschirm oder ein Computermonitor sein kann. Wenn der Bildschirm eingebaute Lautsprecher hat, gibt der Pi über diese den Ton wieder.

Maus & Tastatur

Wenn Sie einen Bildschirm verwenden, benötigen Sie auch eine USB-Tastatur und eine USB-Maus.

HDMI

Der Raspberry Pi verfügt über einen HDMI-Ausgangsport, der mit den HDMI-Anschlüssen der meisten modernen Fernseher und Computermonitore kompatibel ist. Wenn Ihr Bildschirm nur DVI- oder VGA-Anschlüsse hat, benötigen Sie die entsprechende Umwandlungsleitung.

Gehäuse

Sie können den Raspberry Pi in ein Gehäuse setzen; dadurch können Sie Ihr Gerät schützen.

Lautsprecher oder Kopfhörer

Der Raspberry Pi ist mit einem etwa 3,5 mm großen Audioanschluss ausgestattet, der verwendet werden kann, wenn Ihr Bildschirm keine eingebauten Lautsprecher hat oder wenn kein Bildschirm verwendet wird.

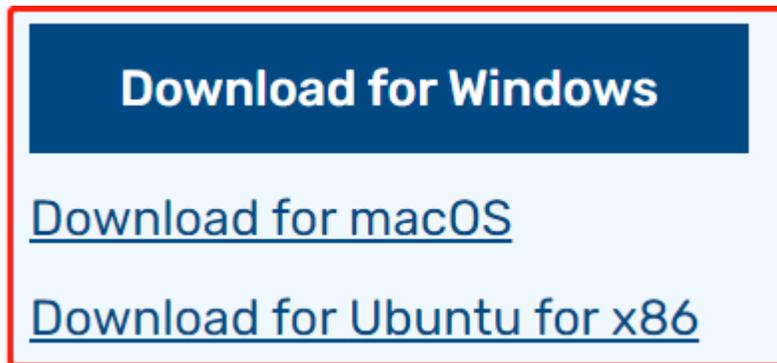
4.1.2 Betriebssystem Installieren

Benötigte Komponenten

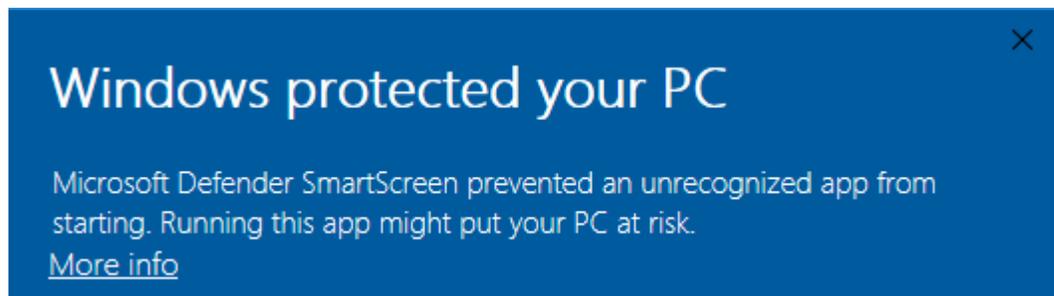
- Raspberry Pi 4B/Zero 2 W/3B 3B+/2B/Zero W
- 1 x Personal Computer
- 1 x Micro SD-Karte

Schritte

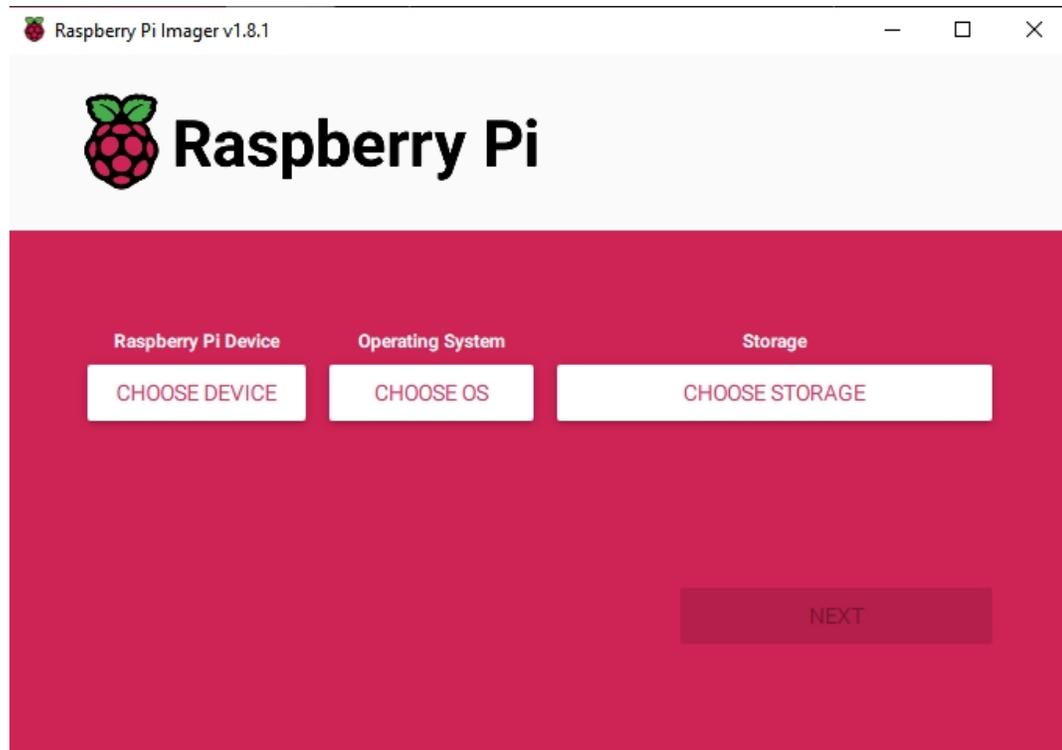
1. Besuchen Sie die Download-Seite der Raspberry Pi-Software: [Raspberry Pi Imager](#). Wählen Sie die Imager-Version für Ihr Betriebssystem aus. Nach dem Herunterladen öffnen Sie die Datei, um die Installation zu starten.



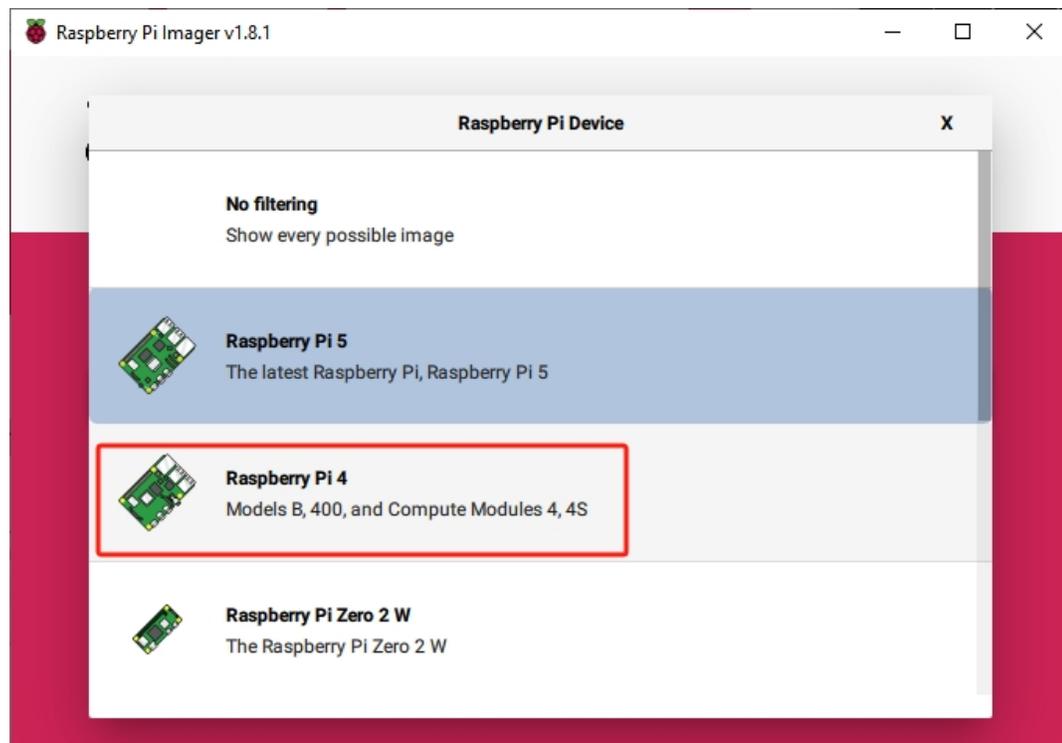
2. Nach dem Start des Installationsprogramms zeigt Ihr Betriebssystem möglicherweise eine Sicherheitswarnung an. Beispielsweise könnte Windows eine Warnmeldung anzeigen. Wenn dies der Fall ist, wählen Sie **More info** und dann **Run anyway**. Folgen Sie den Anweisungen auf dem Bildschirm, um den Raspberry Pi Imager zu installieren.



3. Legen Sie Ihre SD-Karte in den SD-Kartenslot Ihres Computers oder Laptops ein.
4. Öffnen Sie die Raspberry Pi Imager-Anwendung entweder durch Klicken auf das Symbol oder durch Ausführen von `rpi-imager` in Ihrem Terminal.



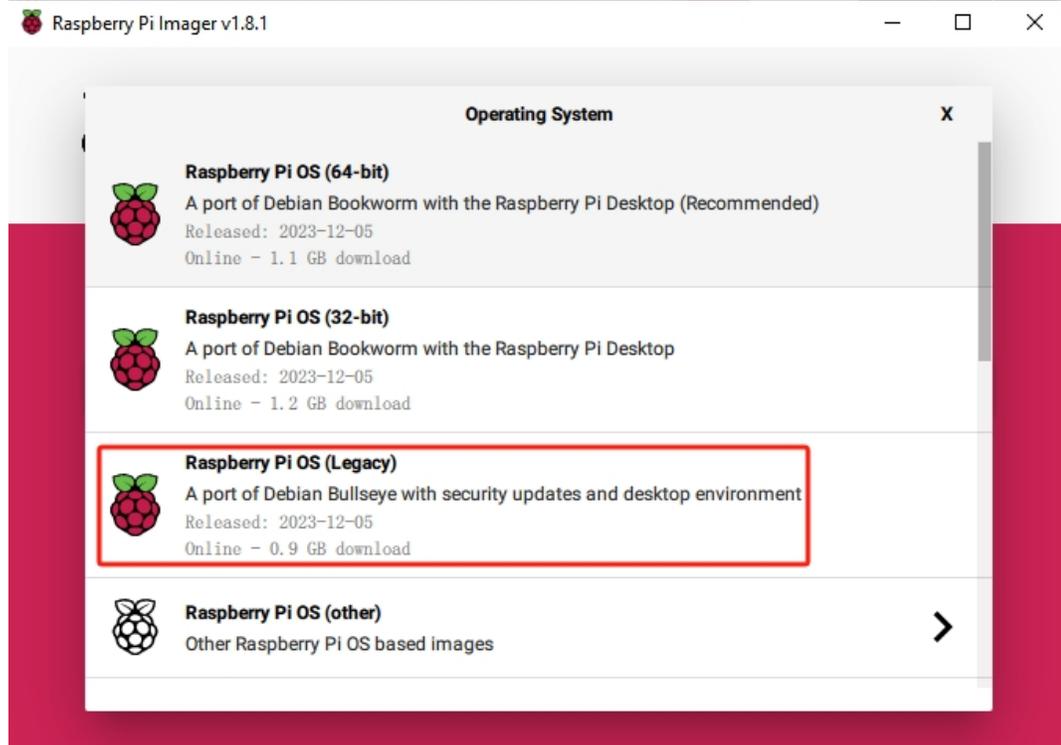
5. Klicken Sie auf **CHOOSE DEVICE** und wählen Sie Ihr spezifisches Raspberry Pi-Modell aus der Liste aus (Hinweis: Raspberry Pi 5 ist nicht anwendbar).



6. Wählen Sie **CHOOSE OS** und dann **Raspberry Pi OS (Legacy)**.

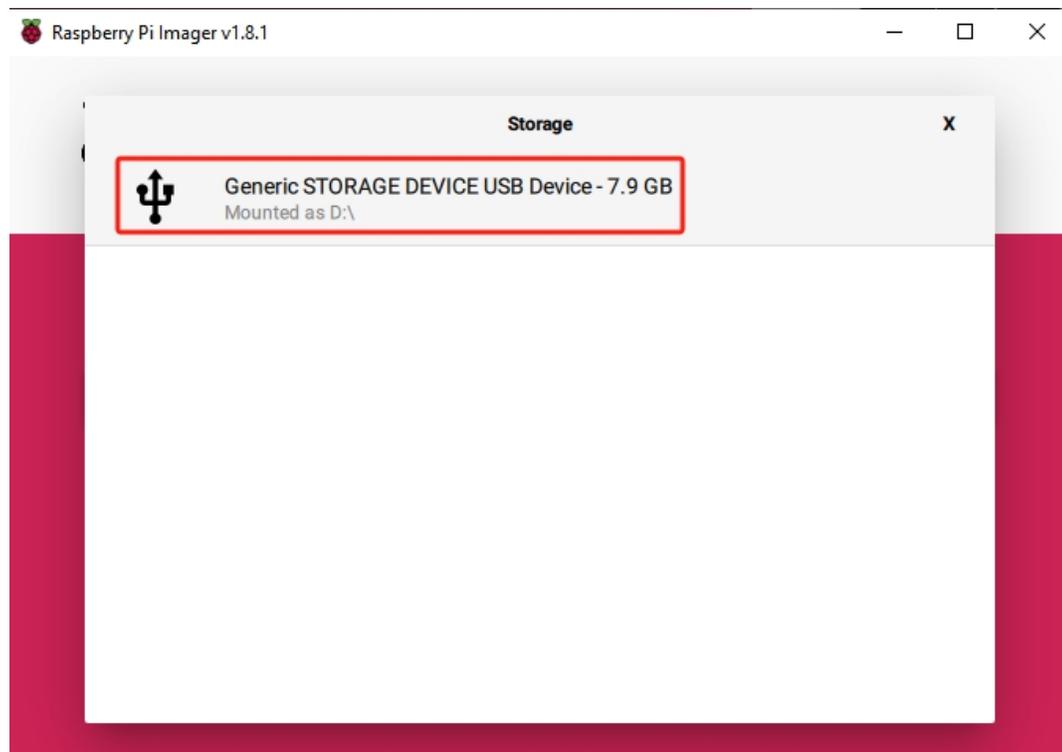
Warnung:

- Bitte installieren Sie nicht die **Bookworm**-Version, da die Lautsprecher nicht funktionieren werden.
- Sie müssen die Version **Raspberry Pi OS (Legacy) - Debian Bullseye** installieren.

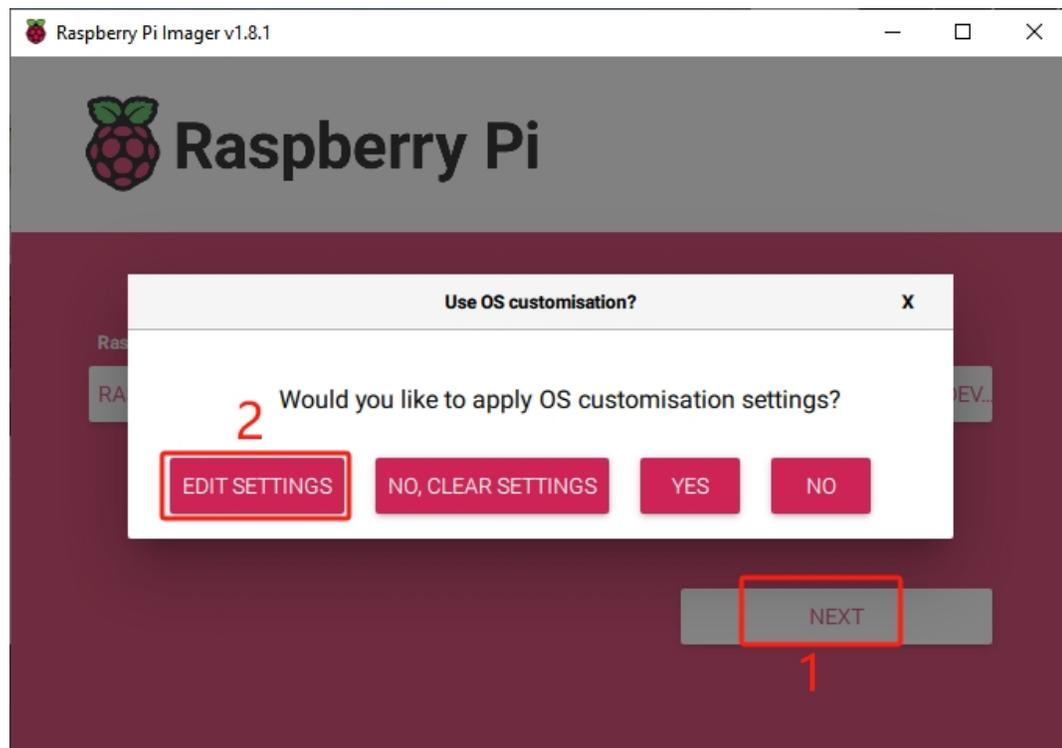


7. Klicken Sie auf **Choose Storage** und wählen Sie das richtige Speichergerät für die Installation aus.

Bemerkung: Stellen Sie sicher, dass Sie das richtige Gerät auswählen, insbesondere wenn mehrere Speichergeräte angeschlossen sind. Trennen Sie andere Geräte, wenn Sie sich nicht sicher sind.



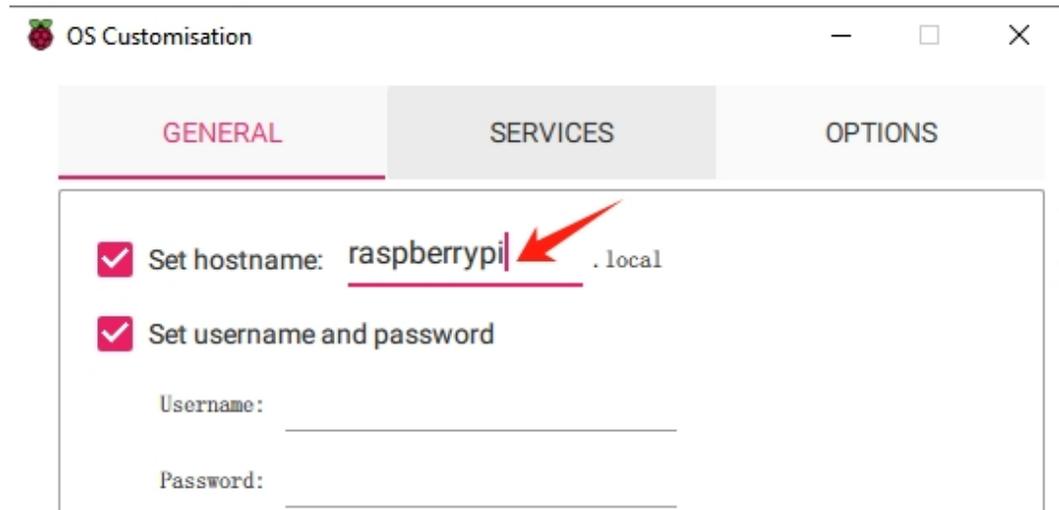
8. Drücken Sie **NEXT** und wählen Sie **EDIT SETTINGS**, um Ihre Betriebssystemeinstellungen anzupassen.



9. Legen Sie den **hostname** Ihres Raspberry Pi fest.

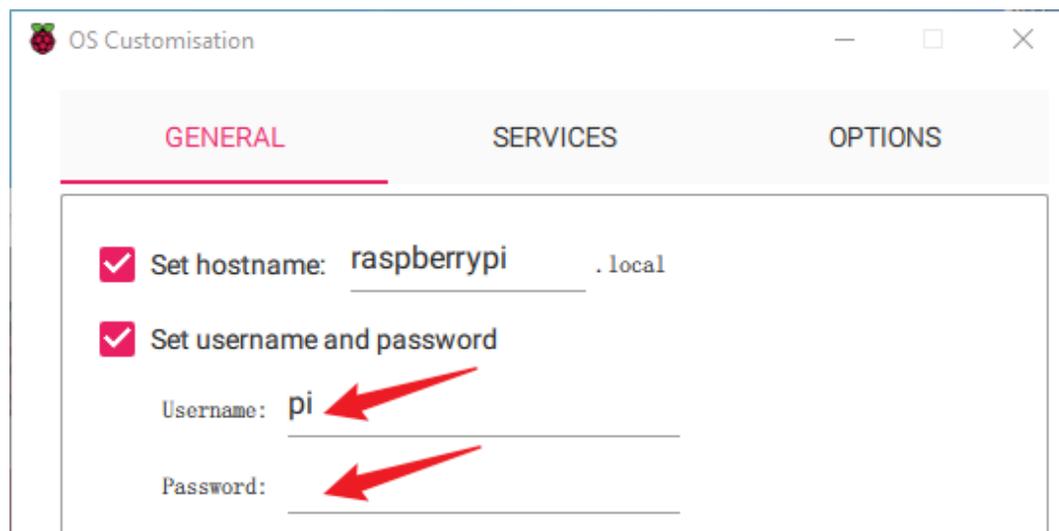
Bemerkung: Der Hostname ist der Name, mit dem sich Ihr Raspberry Pi im Netzwerk identifiziert.

Sie können sich mit `<hostname>.local` oder `<hostname>.lan` mit Ihrem Pi verbinden.



- Erstellen Sie einen **Username** und ein **Password** für das Administratorkonto des Raspberry Pi.

Bemerkung: Das Festlegen eines einzigartigen Benutzernamens und Passworts ist aus Sicherheitsgründen wichtig, da der Raspberry Pi kein Standardpasswort hat.

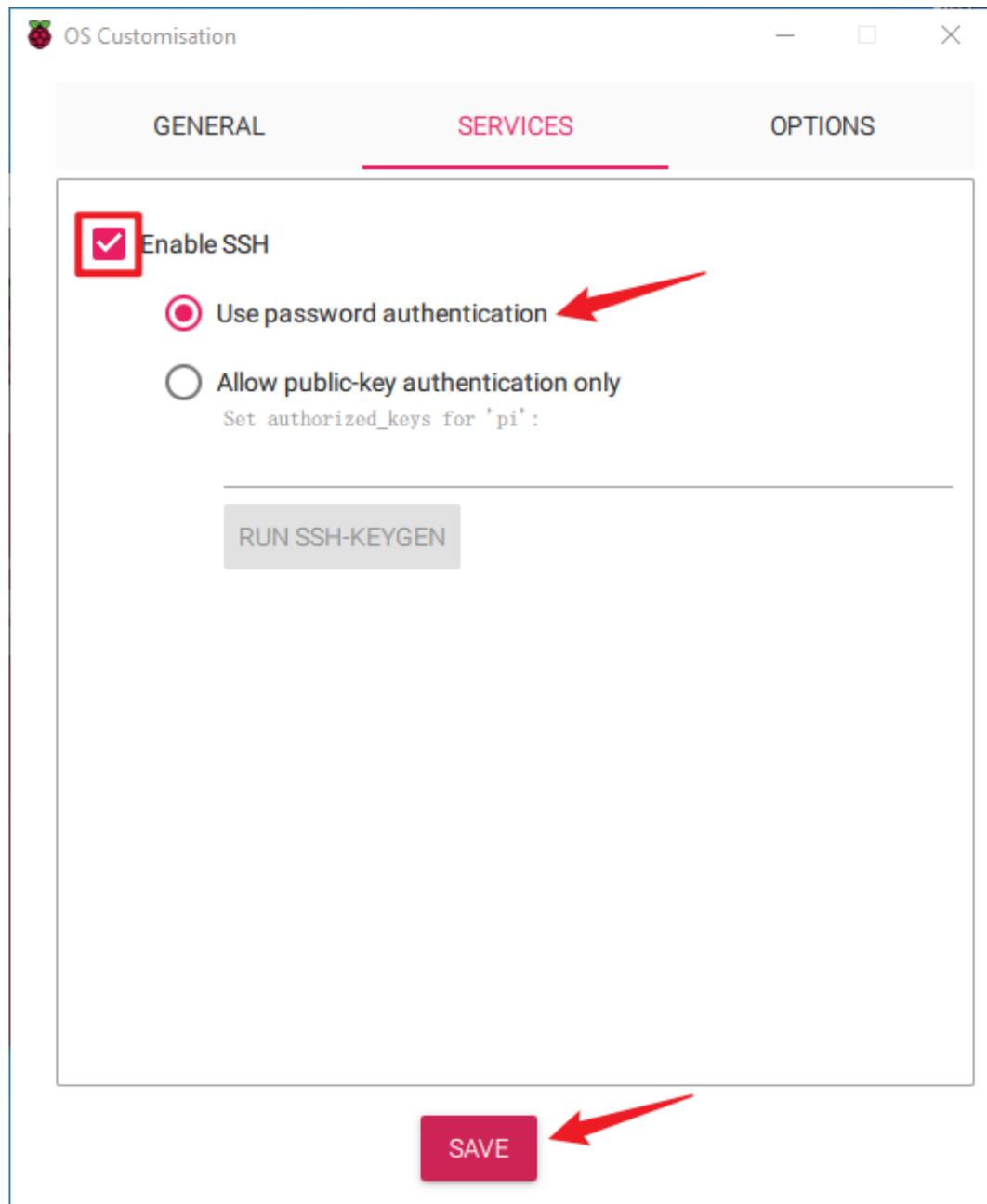


- Richten Sie das drahtlose LAN ein, indem Sie den **SSID** und das **Password** Ihres Netzwerks eingeben.

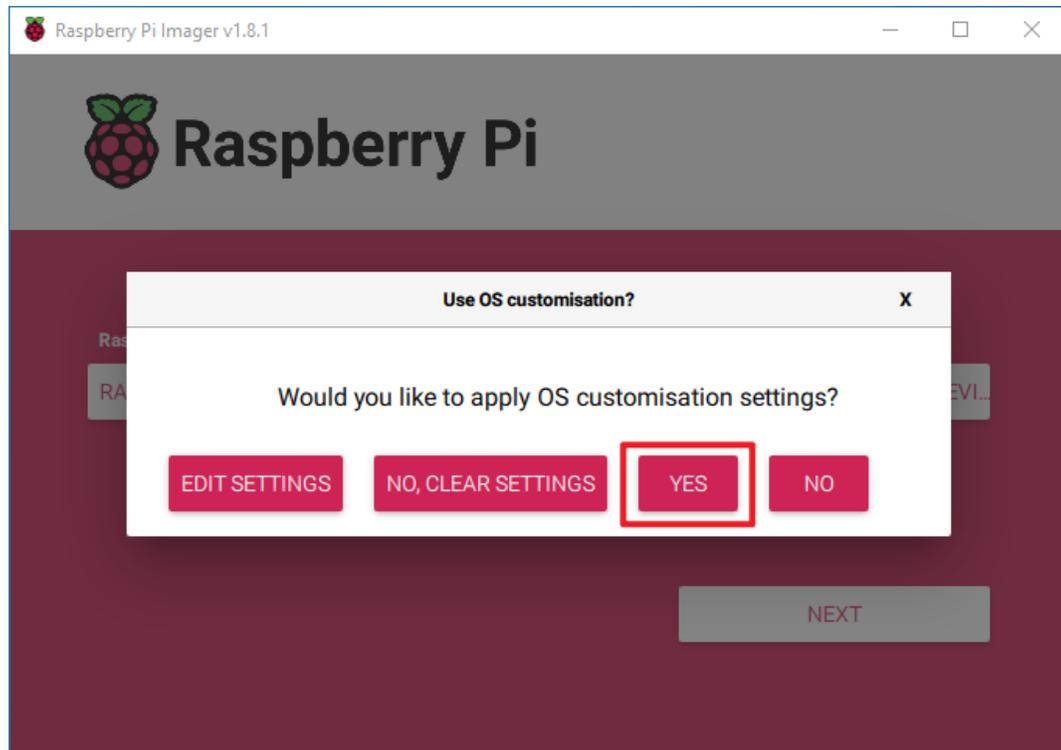
Bemerkung: Wireless LAN-Land sollte auf den zweistelligen ISO/IEC Alpha2-Code für das Land eingestellt werden, in dem Sie Ihren Raspberry Pi verwenden.

The screenshot displays a configuration window for the SunFounder PiCar-X Kit. At the top, the option "Configure wireless LAN" is checked, indicated by a red square with a white checkmark. Below this, there are two input fields: "SSID:" and "Password:", both with red arrows pointing to their respective text boxes. Further down, there are two unchecked checkboxes: "Show password" and "Hidden SSID". Below these is a dropdown menu for "Wireless LAN country:" with "GB" selected and a red arrow pointing to it. At the bottom of the configuration area, there is an unchecked checkbox for "Set locale settings". Underneath, there are two more dropdown menus: "Time zone:" with "Asia/Shanghai" selected and "Keyboard layout:" with "US" selected. At the very bottom of the window is a red "SAVE" button.

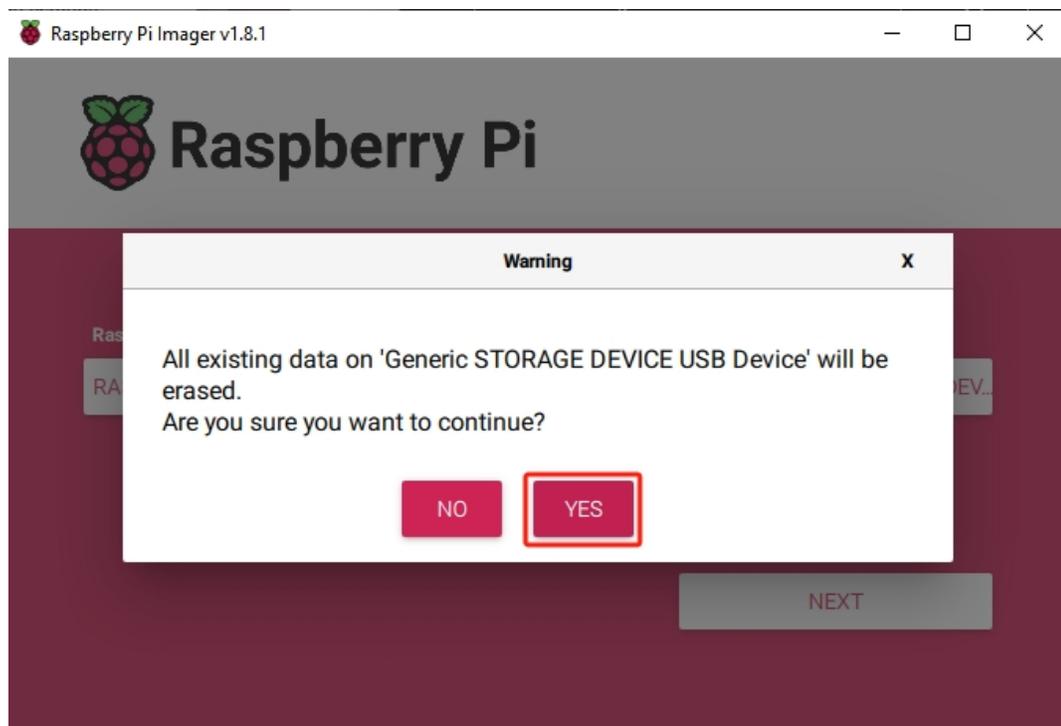
12. Klicken Sie auf **SERVICES** und aktivieren Sie **SSH** für den passwortbasierten Fernzugriff. Denken Sie daran, auf **Save** zu klicken.



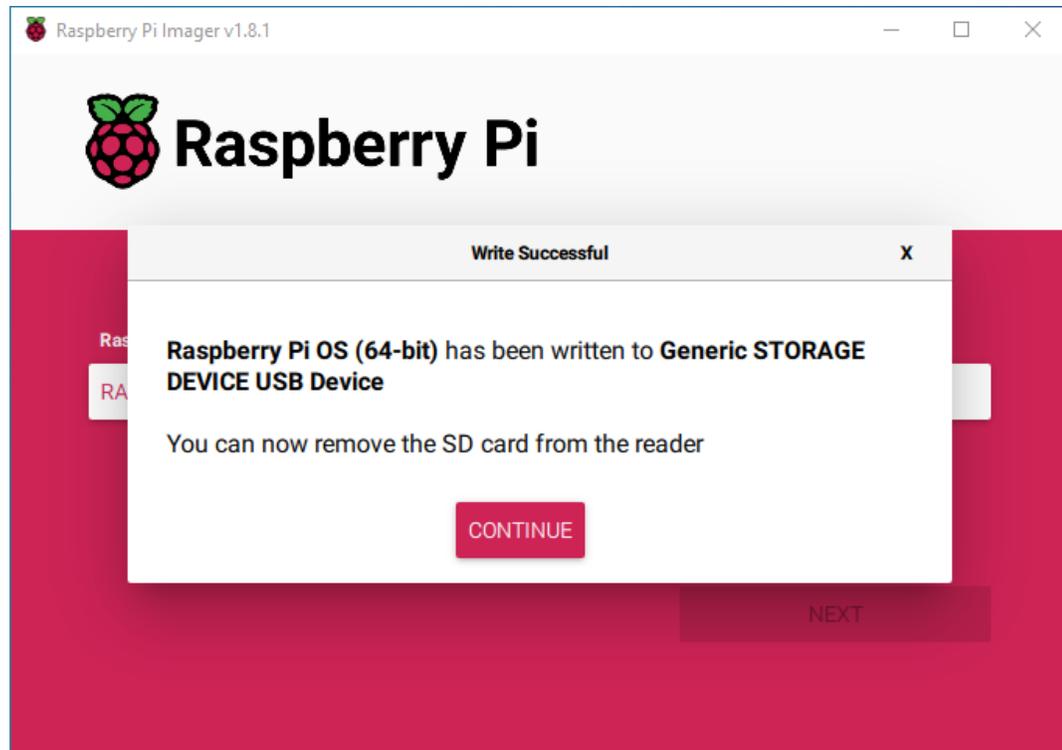
13. Bestätigen Sie Ihre Auswahl, indem Sie auf **Yes** klicken.



14. Sichern Sie vorhandene Dateien auf Ihrer SD-Karte, um Datenverlust zu vermeiden. Klicken Sie auf **Yes**, um fortzufahren, wenn keine Sicherung erforderlich ist.



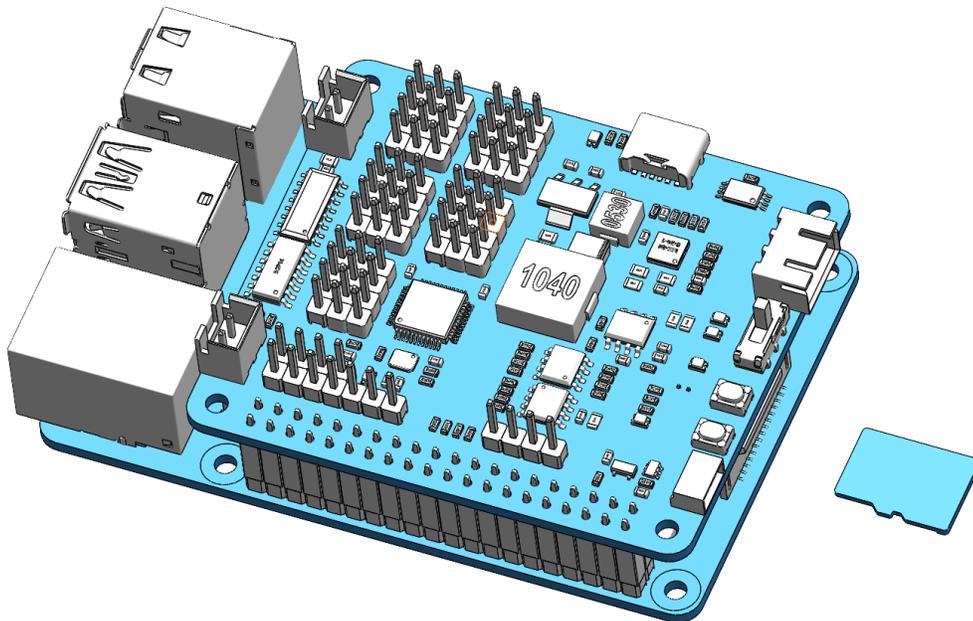
15. Warten Sie, während das Betriebssystem auf die SD-Karte geschrieben wird. Nach Abschluss erscheint ein Bestätigungsfenster.



4.1.3 Richten Sie Ihren Raspberry Pi Ein

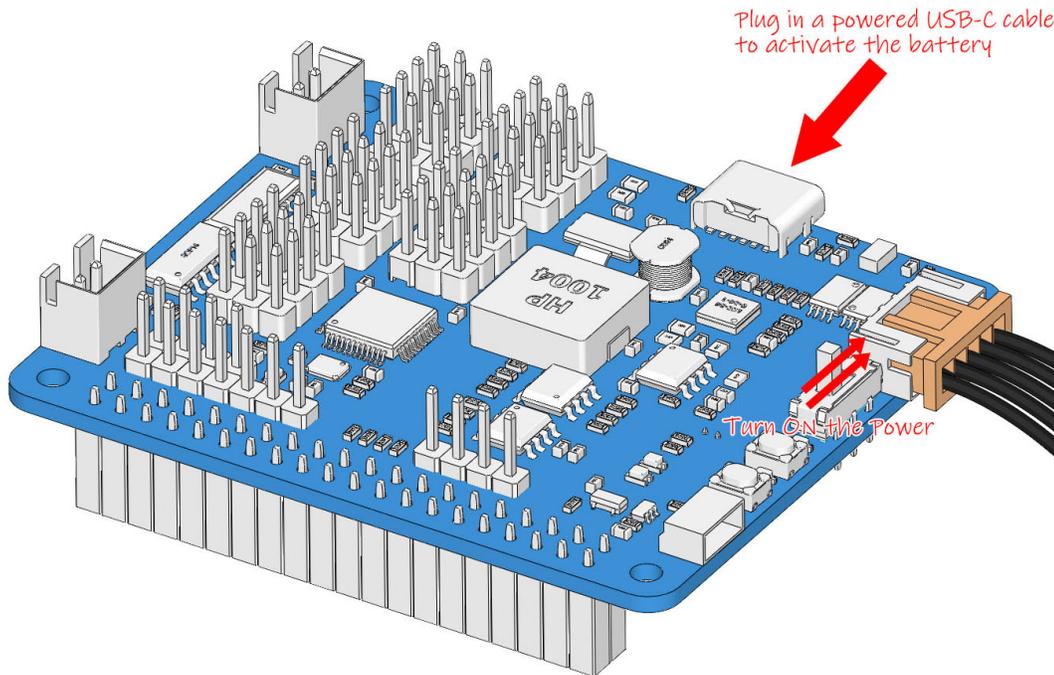
Stromversorgung für den Raspberry Pi (Wichtig)

1. Setzen Sie die mit dem Raspberry Pi OS eingerichtete SD-Karte in den Micro-SD-Kartensteckplatz auf der Unterseite des Raspberry Pi ein.



2. Befolgen Sie die Montageanleitung, stecken Sie das Batteriekabel ein und schalten Sie den Stromschalter ein.

Anschließend stecken Sie das USB-C-Kabel ein, um die Batterie aufzuladen. Warten Sie 1-2 Minuten, und Sie werden ein Geräusch hören, das anzeigt, dass der Raspberry Pi erfolgreich gestartet ist.



Bemerkung: Es wird empfohlen, das USB-C-Kabel eingesteckt zu lassen, da der nachfolgende Software-Einrichtungsprozess eine erhebliche Zeit in Anspruch nehmen kann.

Wenn Sie Einen Bildschirm Haben

Bemerkung: Der auf dem Roboter installierte Raspberry Pi ZERO lässt sich nicht leicht mit dem Bildschirm verbinden, bitte verwenden Sie die Methode ohne Bildschirm, um ihn einzurichten.

Wenn Sie einen Bildschirm haben, wird es für Sie einfacher sein, am Raspberry Pi zu arbeiten.

Benötigte Komponenten

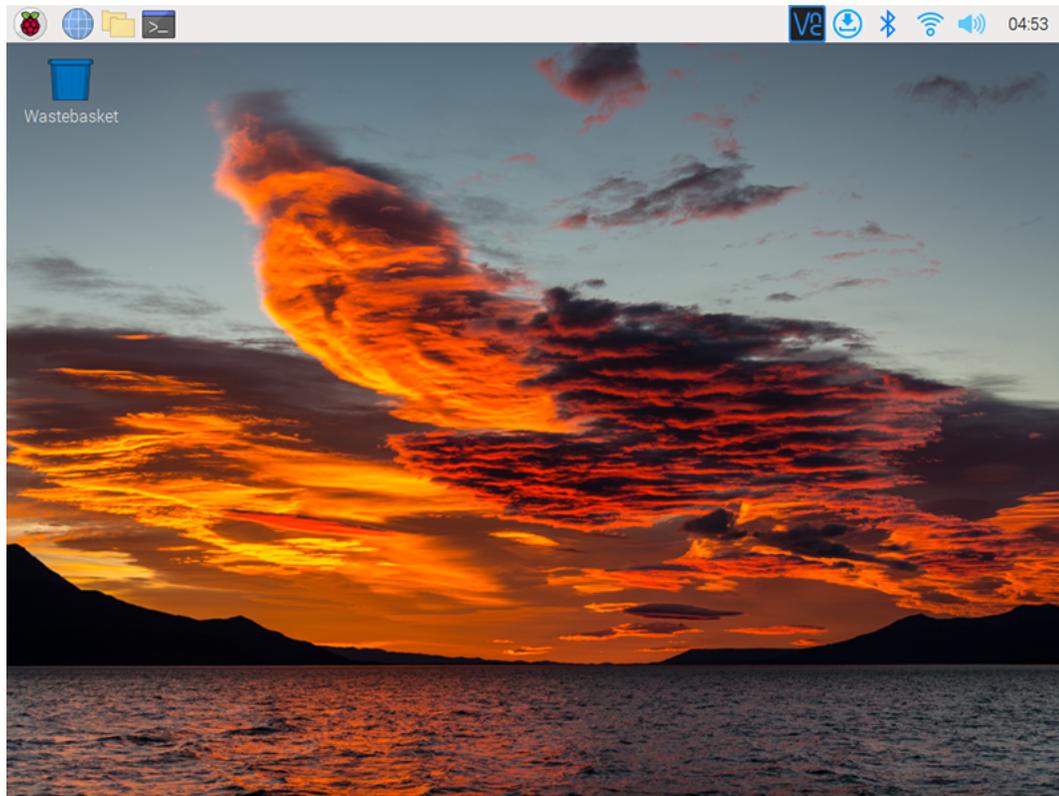
- Raspberry Pi 4B/3B 3B+/2B
- 1 * Netzteil
- 1 * Micro-SD-Karte
- 1 * Bildschirm-Netzteil
- 1 * HDMI-Kabel
- 1 * Bildschirm
- 1 * Maus
- 1 * Tastatur

1. Schließen Sie die Maus und die Tastatur an.

2. Verbinden Sie den Bildschirm mit dem HDMI-Port des Raspberry Pi und stellen Sie sicher, dass Ihr Bildschirm an eine Steckdose angeschlossen und eingeschaltet ist.

Bemerkung: Wenn Sie einen Raspberry Pi 4 verwenden, müssen Sie den Bildschirm mit dem HDMI0-Anschluss (am nächsten zum Stromanschluss) verbinden.

3. Verwenden Sie das Netzteil, um den Raspberry Pi mit Strom zu versorgen. Nach einigen Sekunden wird der Raspberry Pi OS-Desktop angezeigt.



Wenn Sie Keinen Bildschirm Haben

Wenn Sie keinen Monitor haben, können Sie sich remote in Ihren Raspberry Pi einloggen.

Benötigte Komponenten

- – Raspberry Pi 4B/Zero 2 W/3B 3B+/2B/Zero W
- 1 * Netzteil
- 1 * Micro-SD-Karte

Sie können den SSH-Befehl anwenden, um die Bash-Shell des Raspberry Pi zu öffnen. Bash ist die Standard-Standard-Shell für Linux. Die Shell selbst ist ein Befehl (Anweisung), wenn der Benutzer Unix/Linux verwendet. Das meiste, was Sie tun müssen, kann über die Shell erledigt werden.

Wenn Sie mit der Verwendung des Befehlsfensters für den Zugriff auf Ihren Raspberry Pi nicht zufrieden sind, können Sie auch die Funktion „Remote-Desktop“ verwenden, um Dateien auf Ihrem Raspberry Pi einfach über eine grafische Benutzeroberfläche (GUI) zu verwalten.

Siehe unten für detaillierte Tutorials für jedes System.

Mac OS X-Nutzer

Für Mac-Nutzer ist der direkte Zugriff auf den Raspberry Pi-Desktop über VNC bequemer als über die Befehlszeile. Sie können darauf über den Finder zugreifen, indem Sie das festgelegte Kontopasswort eingeben, nachdem Sie VNC auf der Raspberry Pi-Seite aktiviert haben.

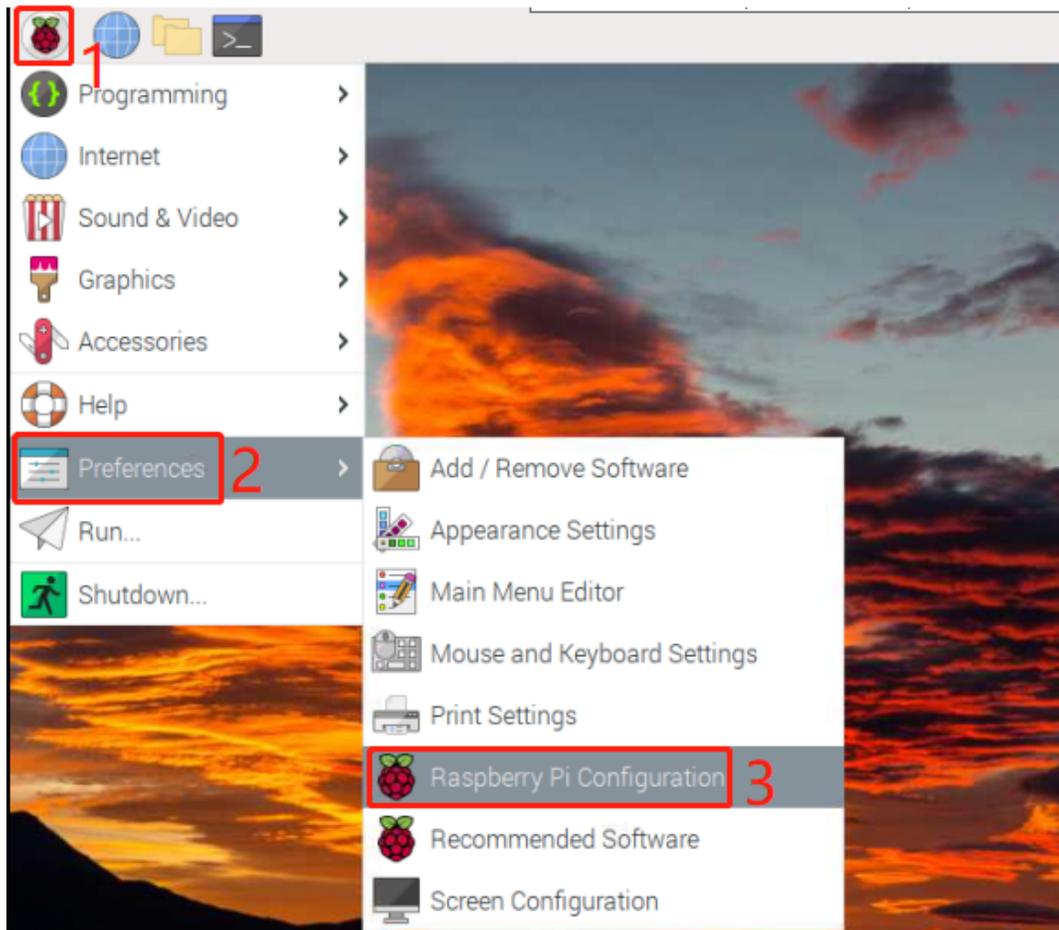
Beachten Sie, dass diese Methode die Kommunikation zwischen dem Mac und dem Raspberry Pi nicht verschlüsselt. Die Kommunikation findet innerhalb Ihres Heim- oder Geschäftsnetzwerks statt, sodass es auch ungeschützt kein Problem darstellt. Wenn Sie jedoch Bedenken haben, können Sie eine VNC-Anwendung wie [VNC® Viewer](#) installieren.

Alternativ wäre es praktisch, wenn Sie einen temporären Monitor (TV), eine Maus und eine Tastatur verwenden könnten, um den Raspberry Pi-Desktop direkt zu öffnen und VNC einzurichten. Wenn nicht, ist das kein Problem, Sie können auch den SSH-Befehl verwenden, um die Bash-Shell des Raspberry Pi zu öffnen und dann den Befehl verwenden, um VNC einzurichten.

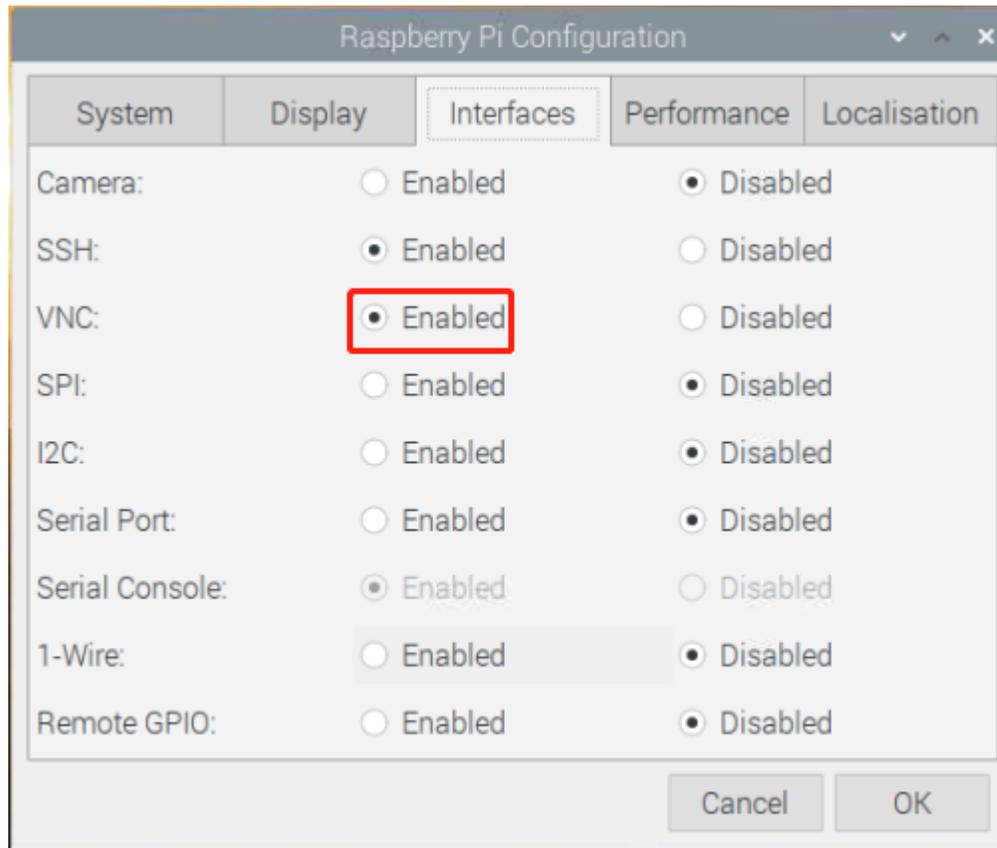
- *Haben Sie Einen Temporären Monitor (oder TV)?*
- *Keinen Temporären Monitor (oder TV)?*

Haben Sie Einen Temporären Monitor (oder TV)?

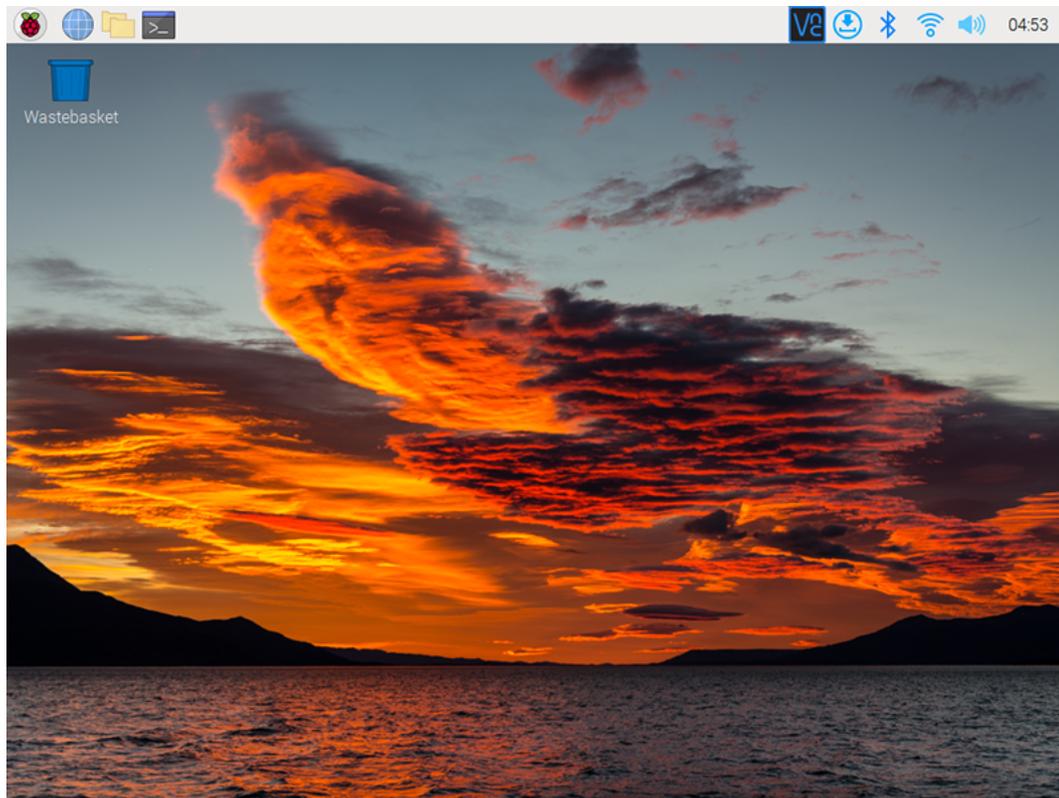
1. Schließen Sie einen Monitor (oder TV), eine Maus und eine Tastatur an den Raspberry Pi an und schalten Sie ihn ein. Wählen Sie das Menü entsprechend den Nummern in der Abbildung.



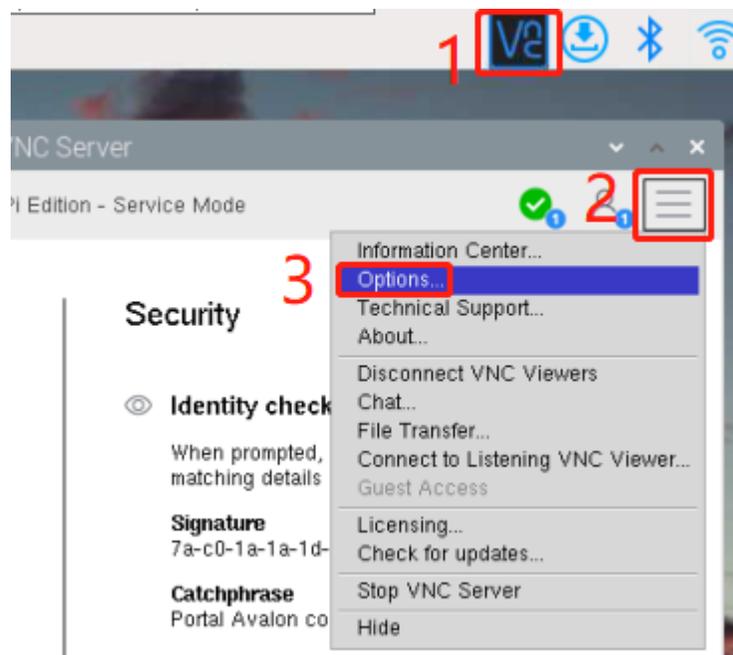
2. Der folgende Bildschirm wird angezeigt. Setzen Sie **VNC** auf **Enabled** im Reiter **Interfaces**, und klicken Sie auf **OK**.



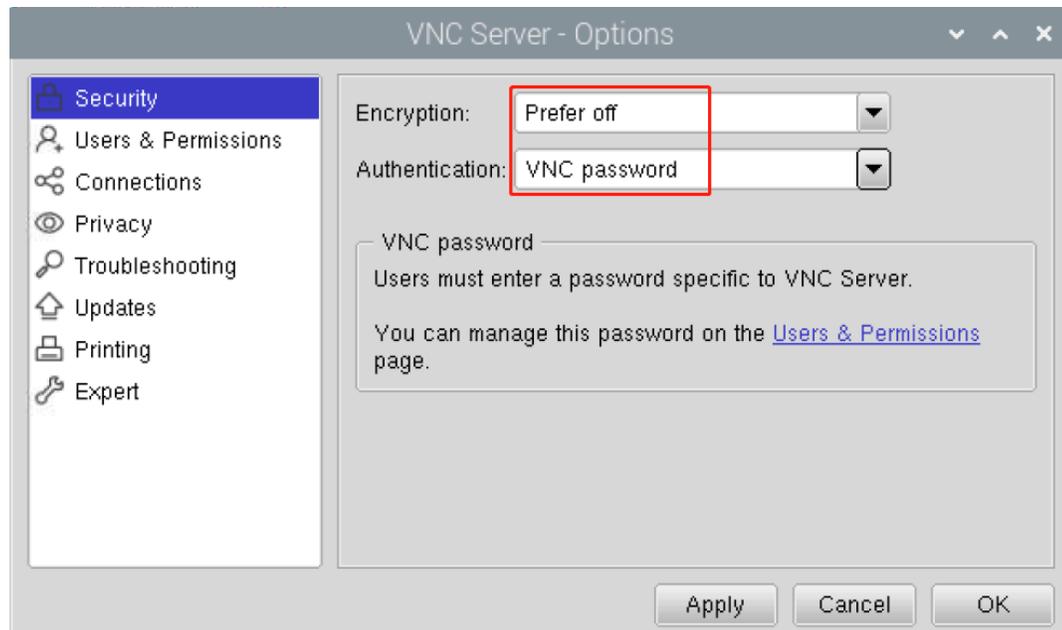
3. Ein VNC-Symbol erscheint oben rechts auf dem Bildschirm und der VNC-Server startet.



- Öffnen Sie das VNC-Serverfenster, indem Sie auf das **VNC**-Symbol klicken, dann klicken Sie auf die **Menu**-Schaltfläche in der oberen rechten Ecke und wählen **Options**.

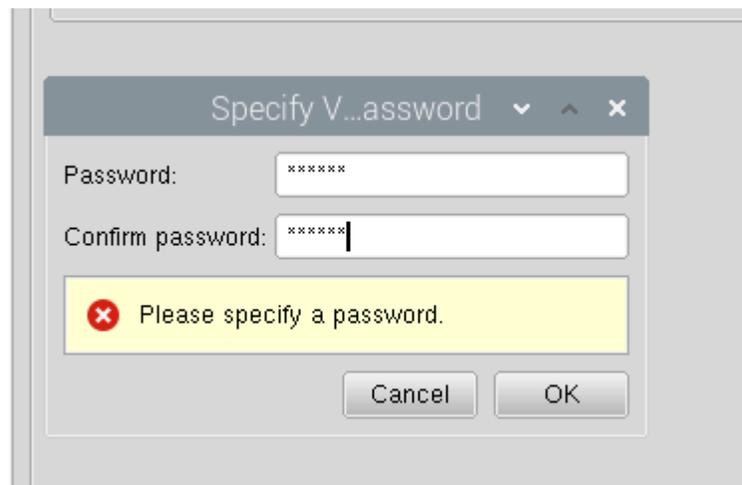


- Sie werden mit dem folgenden Bildschirm präsentiert, auf dem Sie die Optionen ändern können.



Stellen Sie **Encryption** auf **Prefer off** und **Authentication** auf **VNC password**.

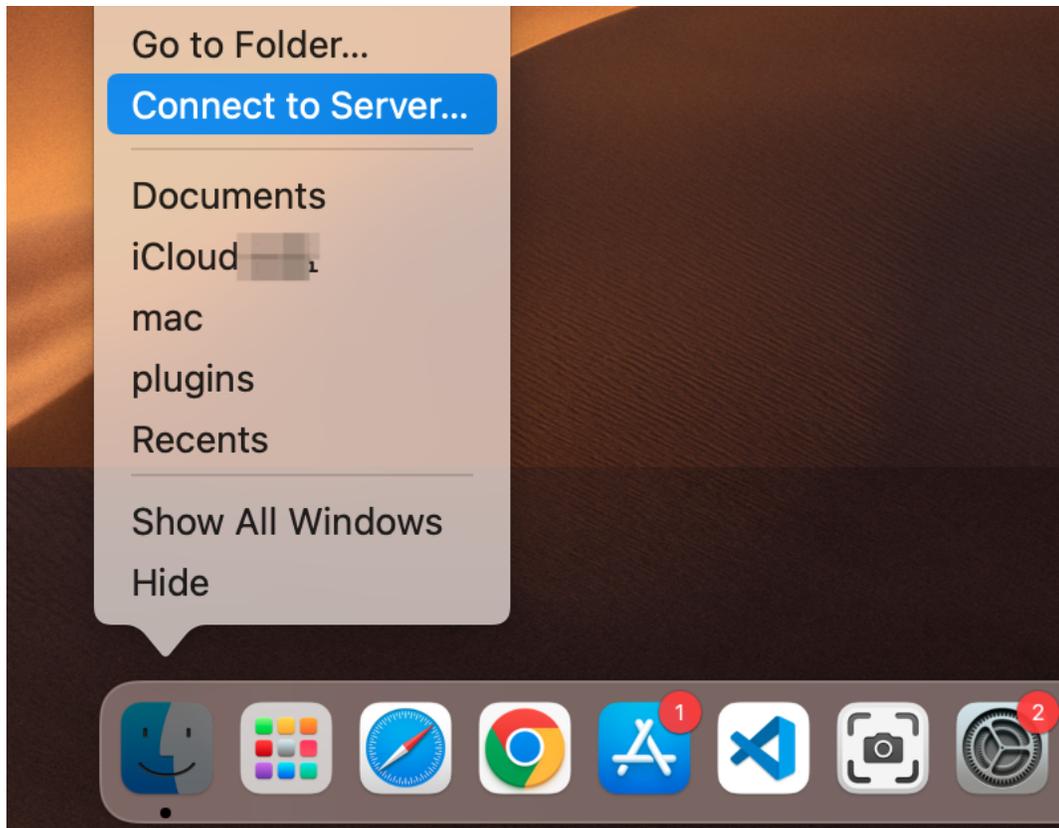
6. Wenn Sie auf die Schaltfläche **OK** klicken, wird der Passwordeingabebildschirm angezeigt. Sie können dasselbe Passwort wie das Raspberry Pi-Passwort oder ein anderes Passwort verwenden, also geben Sie es ein und klicken Sie auf **OK**.



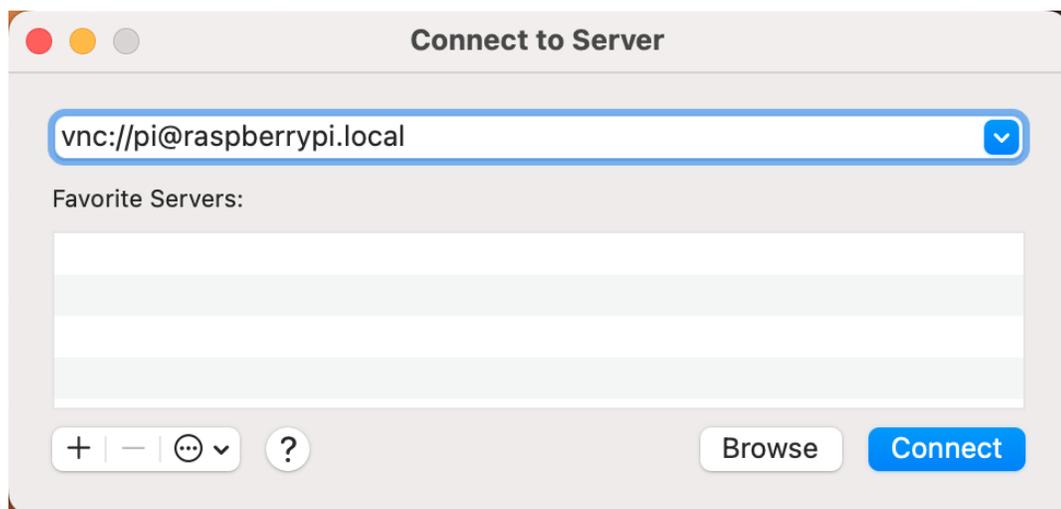
Sie sind jetzt bereit, von Ihrem Mac aus eine Verbindung herzustellen. Es ist in Ordnung, den Monitor zu trennen.

Von hier an wird es die Operation auf der Mac-Seite sein.

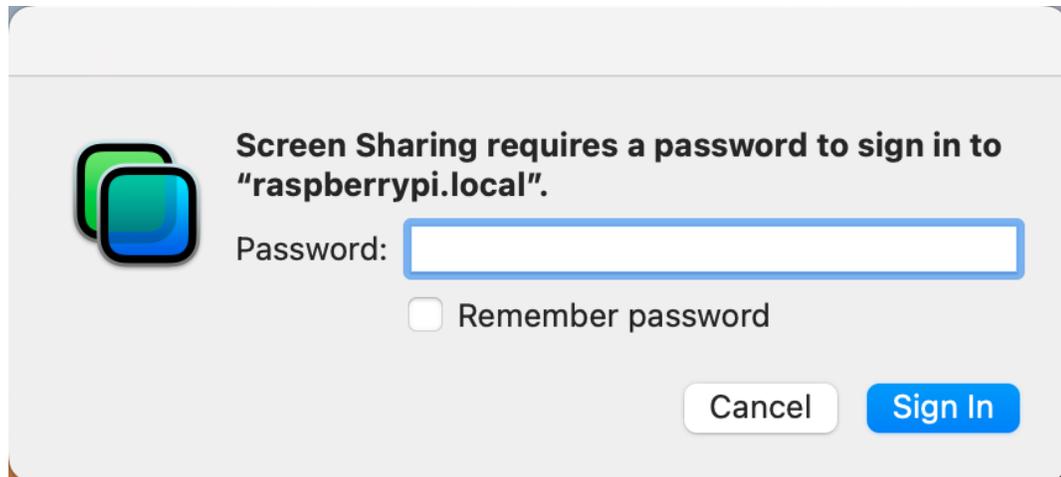
1. Wählen Sie jetzt **Connect to Server** aus dem Finder-Menü, das Sie durch Rechtsklick öffnen können.



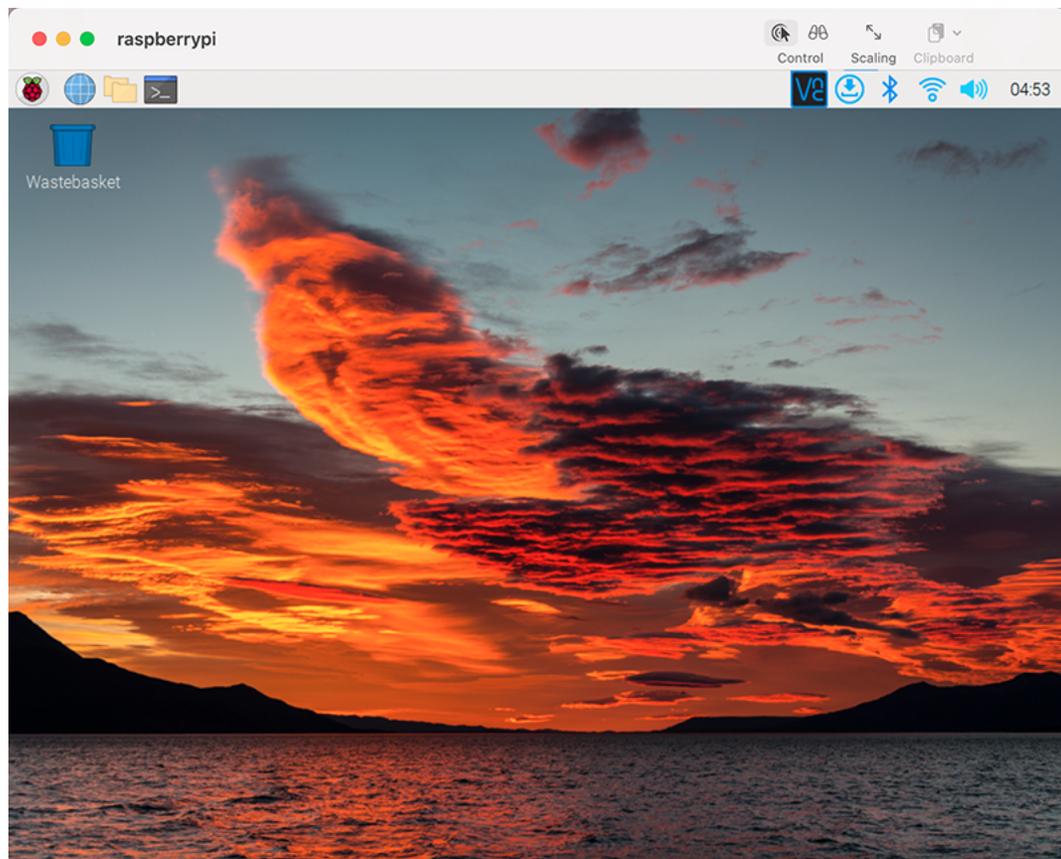
2. Geben Sie `vnc://<benutzername>@<hostname>.local` (oder `vnc://<benutzername>@<IP-Adresse>`) ein. Nach der Eingabe klicken Sie auf **Connect**.



3. Sie werden nach einem Passwort gefragt, also geben Sie es bitte ein.



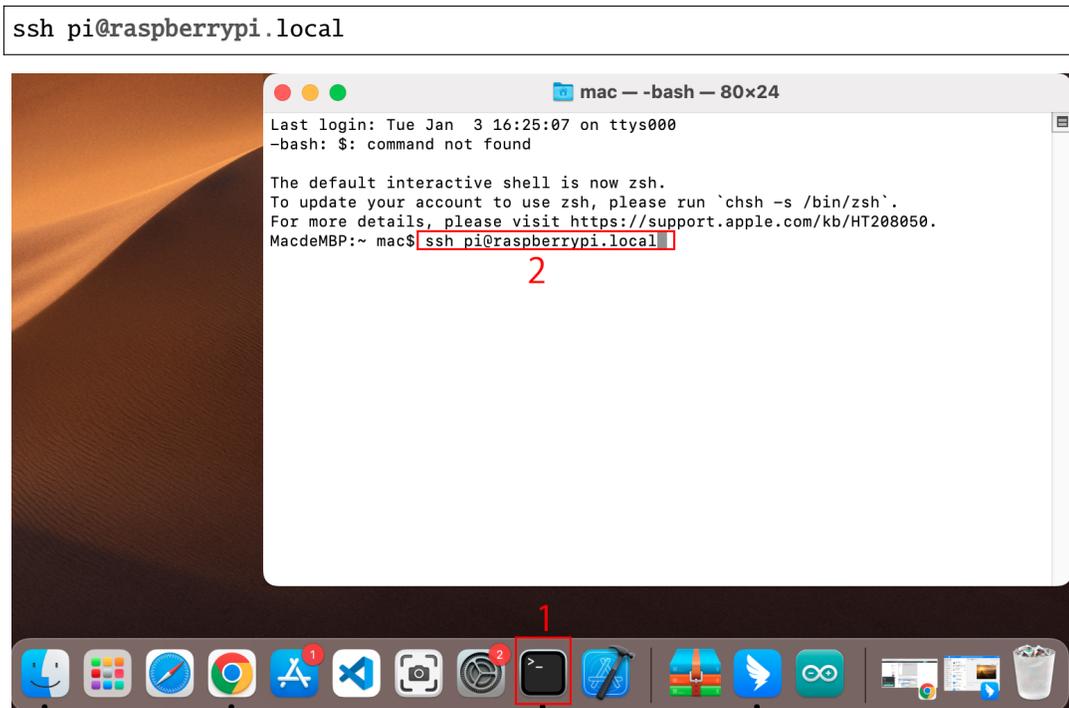
4. Der Desktop des Raspberry Pi wird angezeigt, und Sie können ihn direkt vom Mac aus bedienen.



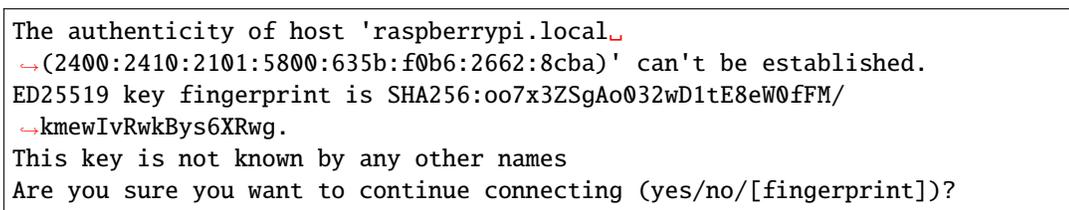
Keinen Temporären Monitor (oder TV)?

- Sie können den SSH-Befehl anwenden, um die Bash-Shell des Raspberry Pi zu öffnen.
- Bash ist die Standard-Standard-Shell für Linux.
- Die Shell selbst ist ein Befehl (Anweisung), wenn der Benutzer Unix/Linux verwendet.
- Das meiste, was Sie tun müssen, kann über die Shell erledigt werden.
- Nachdem Sie den Raspberry Pi eingerichtet haben, können Sie vom Mac aus über den **Finder** auf den Desktop des Raspberry Pi zugreifen.

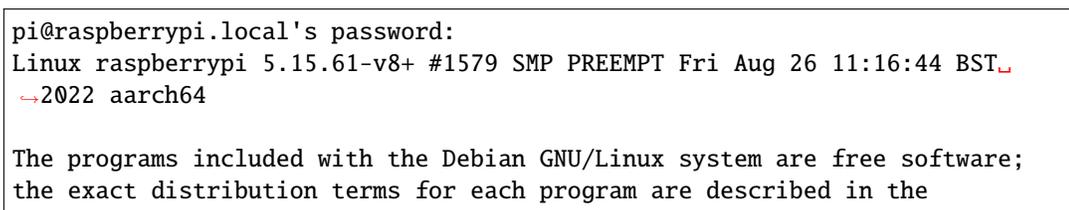
1. Geben Sie `ssh <benutzername>@<hostname>.local` ein, um sich mit dem Raspberry Pi zu verbinden.



2. Die folgende Nachricht wird nur beim ersten Anmelden angezeigt, also geben Sie **yes** ein.



3. Geben Sie das Passwort für den Raspberry Pi ein. Das eingegebene Passwort wird nicht angezeigt, seien Sie also vorsichtig, keinen Fehler zu machen.



(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
individual files in /usr/share/doc/*/copyright.
```

```
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
```

```
Last login: Thu Sep 22 12:18:22 2022
```

```
pi@raspberrypi:~ $
```

4. Richten Sie Ihren Raspberry Pi so ein, dass Sie sich über VNC von Ihrem Mac aus anmelden können, sobald Sie sich erfolgreich eingeloggt haben. Der erste Schritt besteht darin, Ihr Betriebssystem zu aktualisieren, indem Sie die folgenden Befehle ausführen.

```
sudo apt update
sudo apt upgrade
```

Möchten Sie fortfahren? [Y/n], geben Sie Y ein, wenn Sie dazu aufgefordert werden.

Es kann einige Zeit dauern, bis das Update abgeschlossen ist. (Es hängt von der Menge der Updates zu diesem Zeitpunkt ab.)

5. Geben Sie den folgenden Befehl ein, um den **VNC-Server** zu aktivieren.

```
sudo raspi-config
```

6. Der folgende Bildschirm wird angezeigt. Wählen Sie mit den Pfeiltasten auf der Tastatur **Interface Options** aus und drücken Sie die **Enter**-Taste.

```
Raspberry Pi Software Configuration Tool (raspi-config)

1 System Options      Configure system settings
2 Display Options     Configure display settings
3 Interface Options   Configure connections to peripherals
4 Performance Options Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options    Configure advanced settings
8 Update              Update this tool to the latest version
9 About raspi-config  Information about this configuration tool

<Select>                <Finish>
```

7. Wählen Sie dann **VNC**.

```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/disable connection to the Raspberry Pi Camera
P2 SSH         Enable/disable remote command line access using SSH
P3 VNC         Enable/disable graphical remote access using RealVNC
P4 SPI         Enable/disable automatic loading of SPI kernel module
P5 I2C         Enable/disable automatic loading of I2C kernel module
P6 Serial Port Enable/disable shell messages on the serial connection
P7 1-Wire      Enable/disable one-wire interface
P8 Remote GPIO Enable/disable remote access to GPIO pins

<Select>                <Back>
```

8. Verwenden Sie die Pfeiltasten auf der Tastatur, um **<Yes>** -> **<OK>** -> **<Finish>** auszuwählen und die Einrichtung abzuschließen.

```
Would you like the VNC Server to be enabled?

<Yes>                <No>
```

9. Jetzt, da der VNC-Server gestartet wurde, ändern wir die Einstellungen für die Verbindung von einem Mac aus.

Um Parameter für alle Programme für alle Benutzerkonten auf dem Computer festzulegen, erstellen Sie `/etc/vnc/config.d/common.custom`.

```
sudo nano /etc/vnc/config.d/common.custom
```

Nachdem Sie `Authentication=VncAuthenter` eingegeben haben, drücken Sie `Ctrl+X -> Y -> Enter`, um zu speichern und zu beenden.



```
mac — pi@raspberrypi: ~ — ssh pi@raspberrypi.local — 80x24
GNU nano 3.2 /etc/vnc/config.d/common.custom Modified
Authentication=VncAuth

[ New File ]
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text  ^J Justify    ^C Cur Pos
^X Exit      ^R Read File  ^\ Replace   ^U Uncut Text ^T To Spell  ^_ Go To Line
```

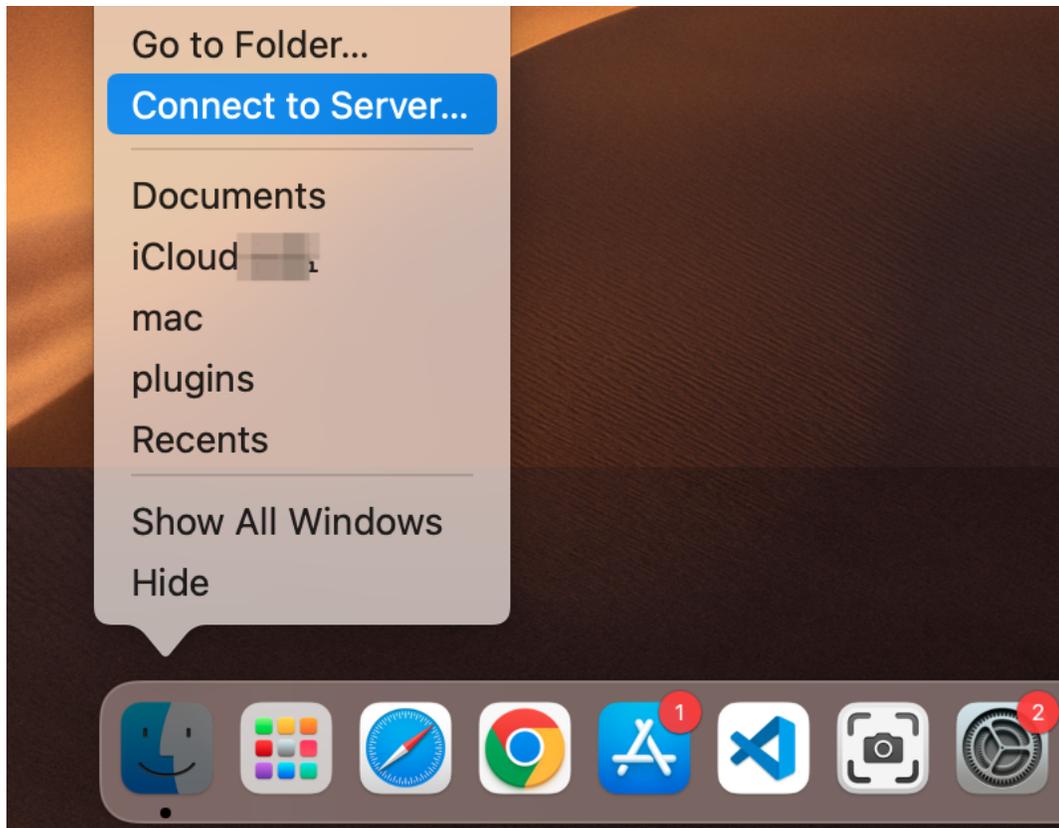
10. Setzen Sie zusätzlich ein Passwort für die Anmeldung über VNC von einem Mac aus. Sie können dasselbe Passwort wie das Raspberry Pi-Passwort oder ein anderes Passwort verwenden.

```
sudo vncpasswd -service
```

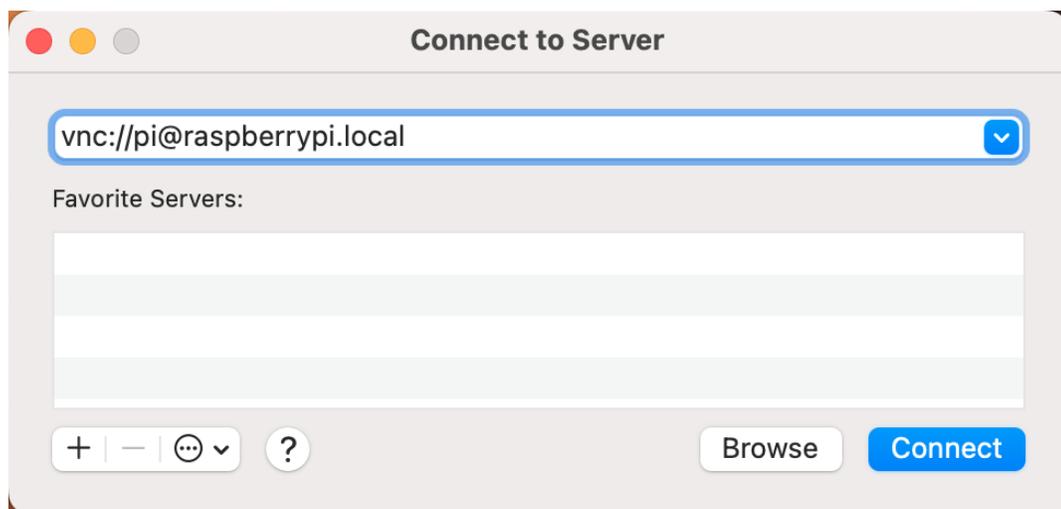
11. Nachdem die Einrichtung abgeschlossen ist, starten Sie den Raspberry Pi neu, um die Änderungen anzuwenden.

```
sudo sudo reboot
```

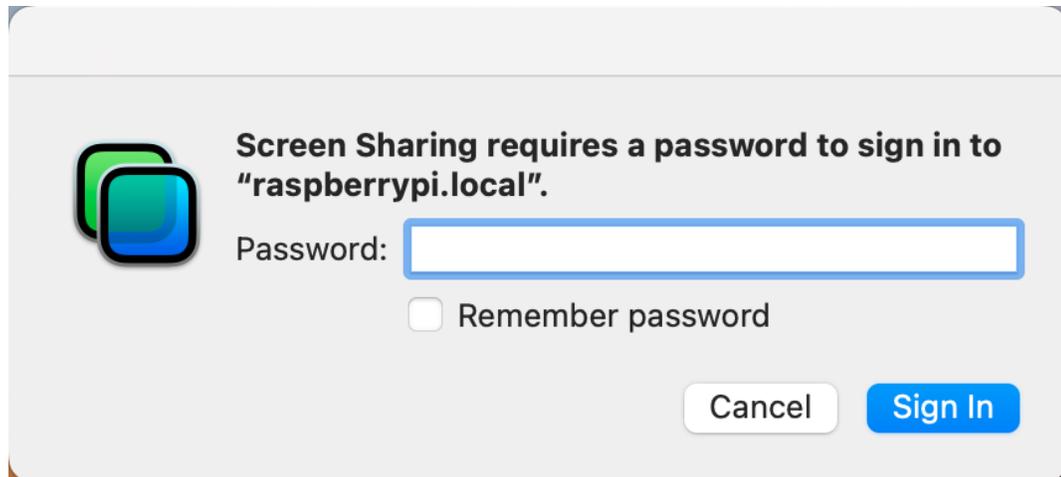
12. Wählen Sie jetzt **Connect to Server** aus dem **Finder**-Menü, das Sie durch Rechtsklick öffnen können.



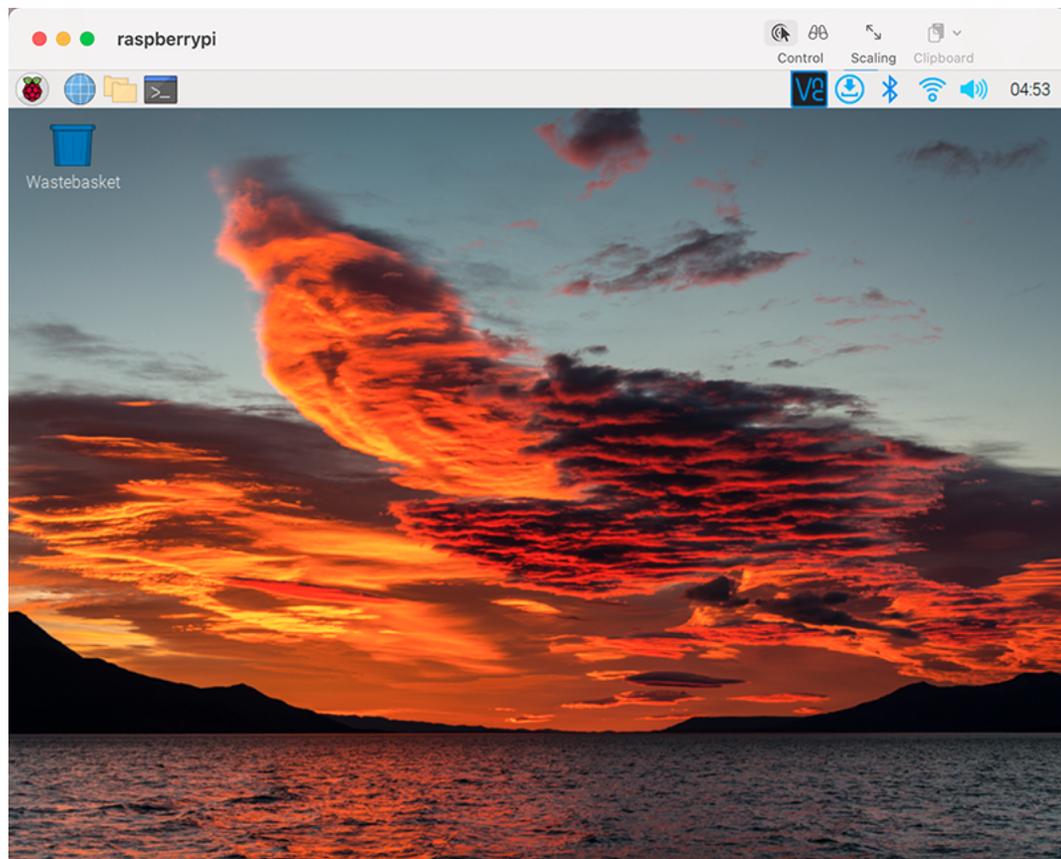
13. Geben Sie `vnc://<benutzername>@<hostname>.local` (oder `vnc://<benutzername>@<IP-Adresse>`) ein. Nach der Eingabe klicken Sie auf **Connect**.



14. Sie werden nach einem Passwort gefragt, also geben Sie es bitte ein.



15. Der Desktop des Raspberry Pi wird angezeigt, und Sie können ihn direkt vom Mac aus bedienen.

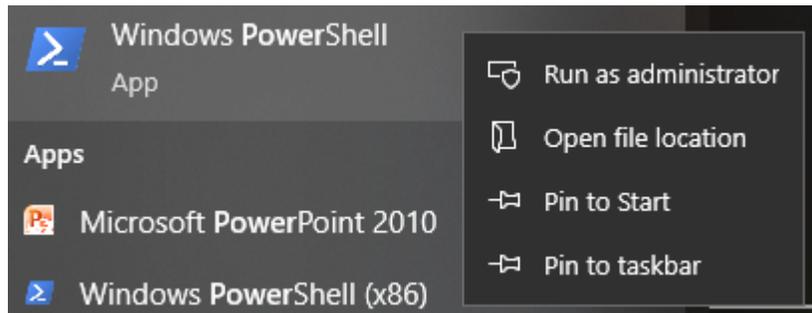


Windows-Nutzer

Anmeldung am Raspberry Pi Aus der Ferne

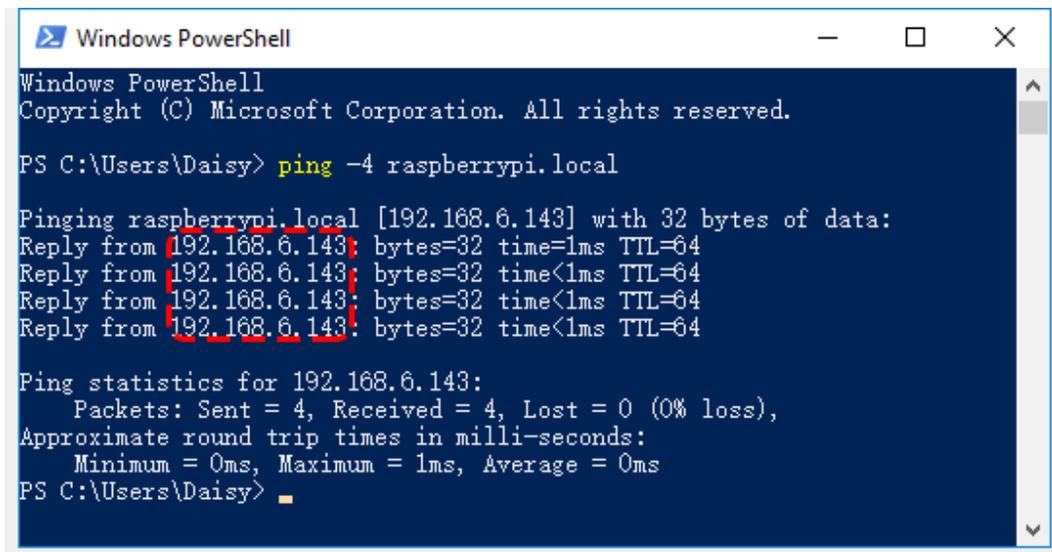
Wenn Sie win10 verwenden, können Sie folgenden Weg nutzen, um sich aus der Ferne am Raspberry Pi anzumelden.

1. Geben Sie in das Suchfeld Ihres Windows-Desktops powershell ein, klicken Sie mit der rechten Maustaste auf die Windows PowerShell und wählen Sie Run as administrator aus dem erscheinenden Menü.



2. Überprüfen Sie dann die IP-Adresse Ihres Raspberry Pi, indem Sie `ping -4 <hostname>.local` eingeben.

```
ping -4 raspberrypi.local
```



Wie oben gezeigt, können Sie die IP-Adresse des Raspberry Pi sehen, nachdem er mit dem Netzwerk verbunden wurde.

- Wenn das Terminal Ping-Anforderung konnte Host pi.local nicht finden. Bitte überprüfen Sie den Namen und versuchen Sie es erneut. anzeigt, folgen Sie den Anweisungen, um sicherzustellen, dass der Hostname, den Sie ausfüllen, korrekt ist.
- Bekommen Sie immer noch keine IP? Überprüfen Sie Ihre Netzwerk- oder WLAN-Konfiguration auf dem Raspberry Pi.

3. Jetzt können Sie sich mit `ssh <username>@<hostname>.local` (oder `ssh <username>@<IP address>`) an Ihrem Raspberry Pi anmelden.

```
ssh pi@raspberrypi.local
```

Warnung: Wenn eine Aufforderung erscheint The term 'ssh' is not recognized as the name of a cmdlet...

Das bedeutet, dass Ihr System zu alt ist und keine vorinstallierten SSH-Tools hat, Sie müssen manuell *Installieren Sie OpenSSH über Powershell* installieren.

Oder verwenden Sie ein Drittanbieter-Tool wie *PuTTY*.

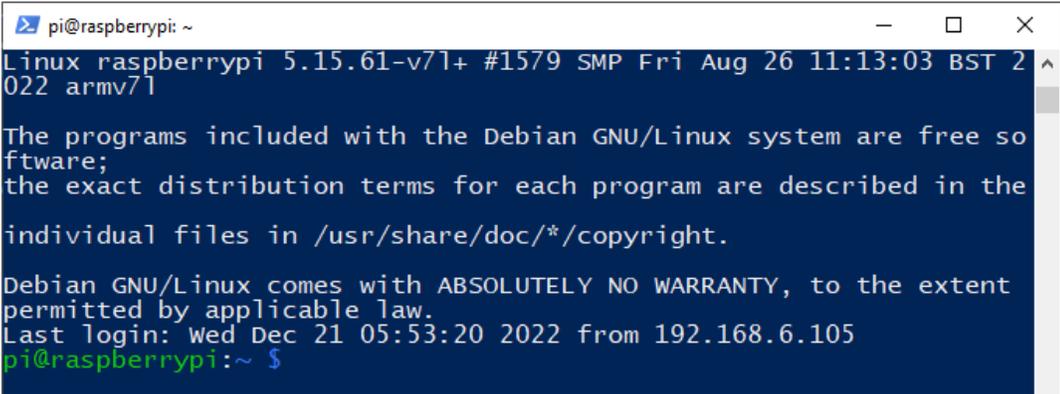
4. Die folgende Nachricht wird nur beim ersten Anmelden angezeigt, also geben Sie yes ein.

```
The authenticity of host 'raspberrypi.local
↳(2400:2410:2101:5800:635b:f0b6:2662:8cba)' can't be established.
ED25519 key fingerprint is SHA256:oo7x3ZSgAo032wD1tE8eW0fFM/
↳kmewIvRwkBys6XRwg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

5. Geben Sie das zuvor festgelegte Passwort ein. (Meines ist raspberry.)

Bemerkung: Wenn Sie das Passwort eingeben, werden die Zeichen nicht im Fenster angezeigt, was normal ist. Wichtig ist, dass Sie das korrekte Passwort eingeben.

6. Wir haben jetzt den Raspberry Pi verbunden und sind bereit für den nächsten Schritt.



```
pi@raspberrypi: ~
Linux raspberrypi 5.15.61-v7l+ #1579 SMP Fri Aug 26 11:13:03 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec 21 05:53:20 2022 from 192.168.6.105
pi@raspberrypi:~ $
```

Remote-Desktop

Wenn Sie mit der Nutzung des Befehlsfensters für den Zugriff auf Ihren Raspberry Pi nicht zufrieden sind, können Sie auch die Remote-Desktop-Funktion verwenden, um Dateien auf Ihrem Raspberry Pi einfach über eine grafische Benutzeroberfläche (GUI) zu verwalten.

Hier verwenden wir **VNC® Viewer**.

VNC-Dienst Aktivieren

Der VNC-Dienst ist im System installiert. Standardmäßig ist VNC deaktiviert. Sie müssen es in der Konfiguration aktivieren.

1. Geben Sie den folgenden Befehl ein:

```
sudo raspi-config
```

```

pi@raspberrypi: ~
login as: pi
pi@192.168.0.234's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 20 09:18:17 2017 from daisy-pc.lan
pi@raspberrypi:~ $ sudo raspi-config

```

2. Wählen Sie **3 Interfacing Options** durch Drücken der Abwärtspfeiltaste auf Ihrer Tastatur, dann drücken Sie die **Enter**-Taste.

```

Raspberry Pi Software Configuration Tool (raspi-config)

1 System Options          Configure system settings
2 Display Options        Configure display settings
3 Interface Options       Configure connections to peripherals
4 Performance Options    Configure performance settings
5 Localisation Options   Configure language and regional settings
6 Advanced Options       Configure advanced settings
8 Update                 Update this tool to the latest version
9 About raspi-config     Information about this configuration tool

<Select>                <Finish>

```

3. Dann **VNC**.

```

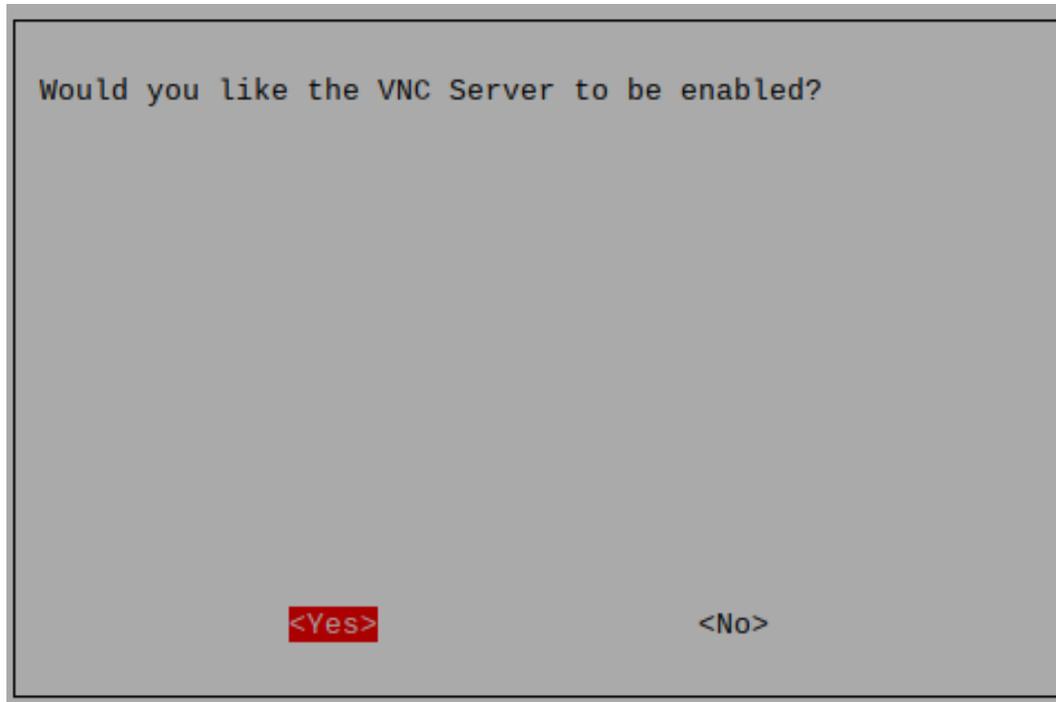
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera                Enable/disable connection to the Raspberry Pi Camera
P2 SSH                   Enable/disable remote command line access using SSH
P3 VNC                   Enable/disable graphical remote access using RealVNC
P4 SPI                   Enable/disable automatic loading of SPI kernel module
P5 I2C                   Enable/disable automatic loading of I2C kernel module
P6 Serial Port           Enable/disable shell messages on the serial connection
P7 1-Wire                Enable/disable one-wire interface
P8 Remote GPIO           Enable/disable remote access to GPIO pins

<Select>                <Back>

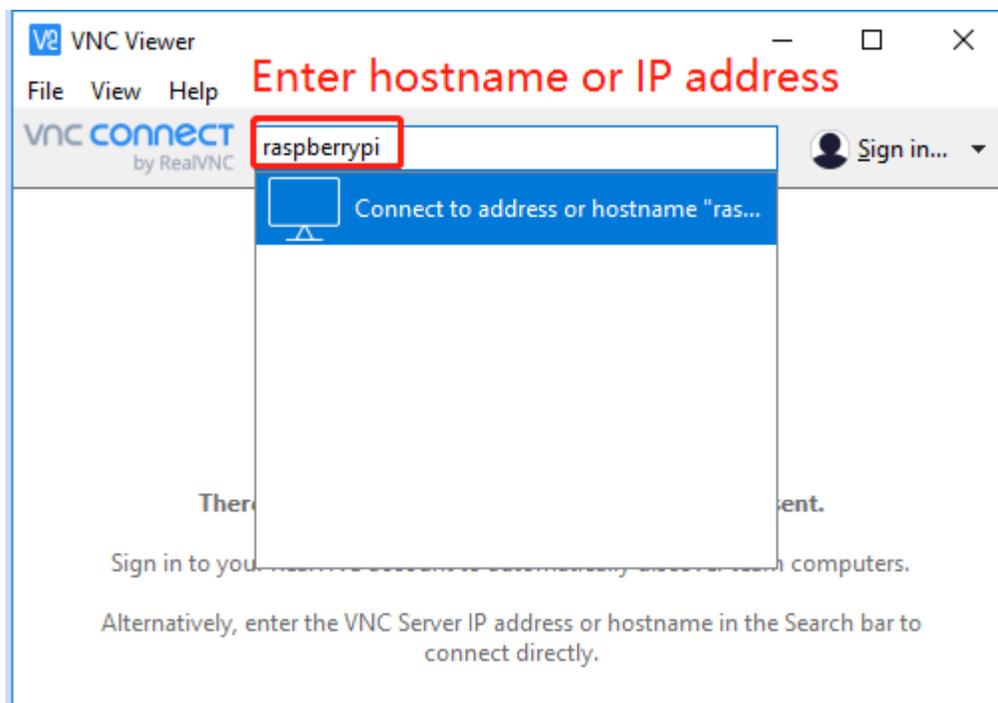
```

4. Verwenden Sie die Pfeiltasten auf der Tastatur, um **<Yes>** -> **<OK>** -> **<Finish>** auszuwählen und die Einrichtung abzuschließen.

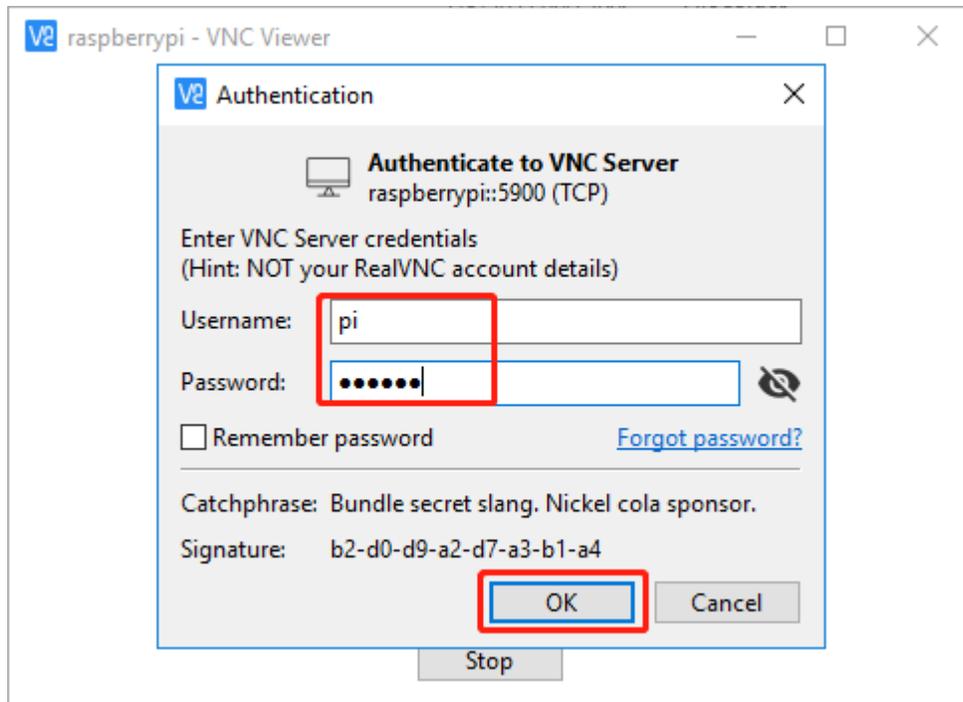


Anmeldung beim VNC

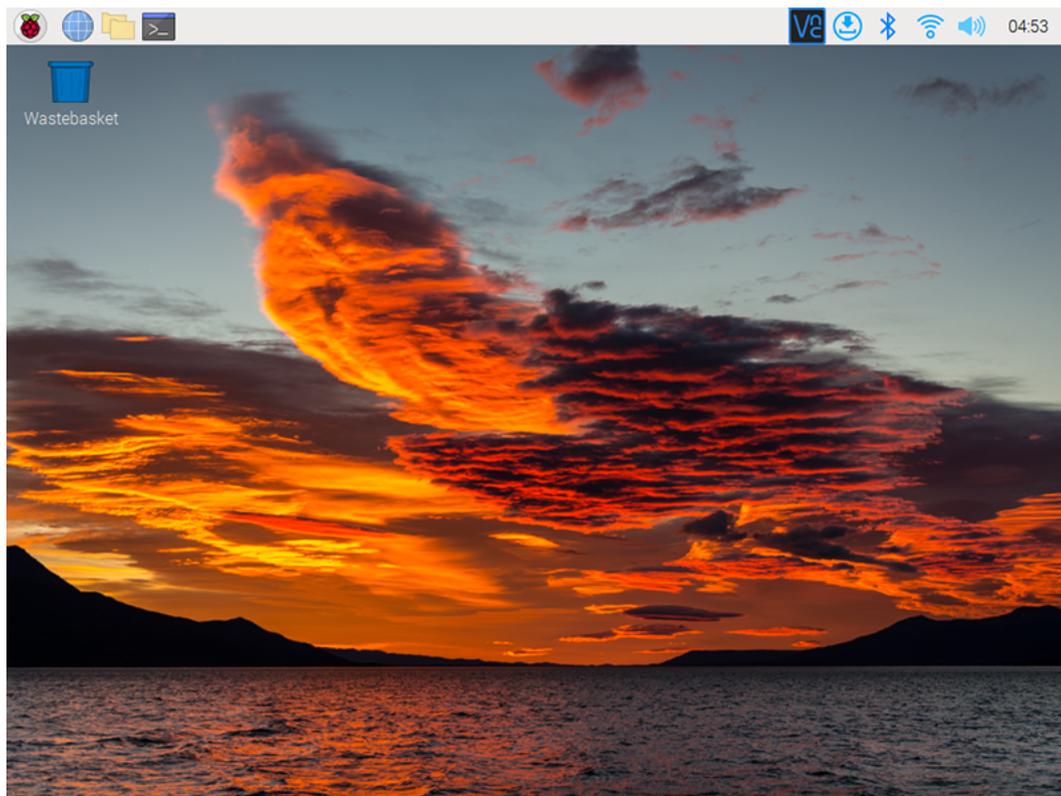
1. Sie müssen den [VNC Viewer](#) auf Ihrem persönlichen Computer herunterladen und installieren.
2. Öffnen Sie ihn, sobald die Installation abgeschlossen ist. Geben Sie dann den Hostnamen oder die IP-Adresse ein und drücken Sie Enter.



3. Nachdem Sie Ihren Raspberry Pi-Namen und Ihr Passwort eingegeben haben, klicken Sie auf **OK**.

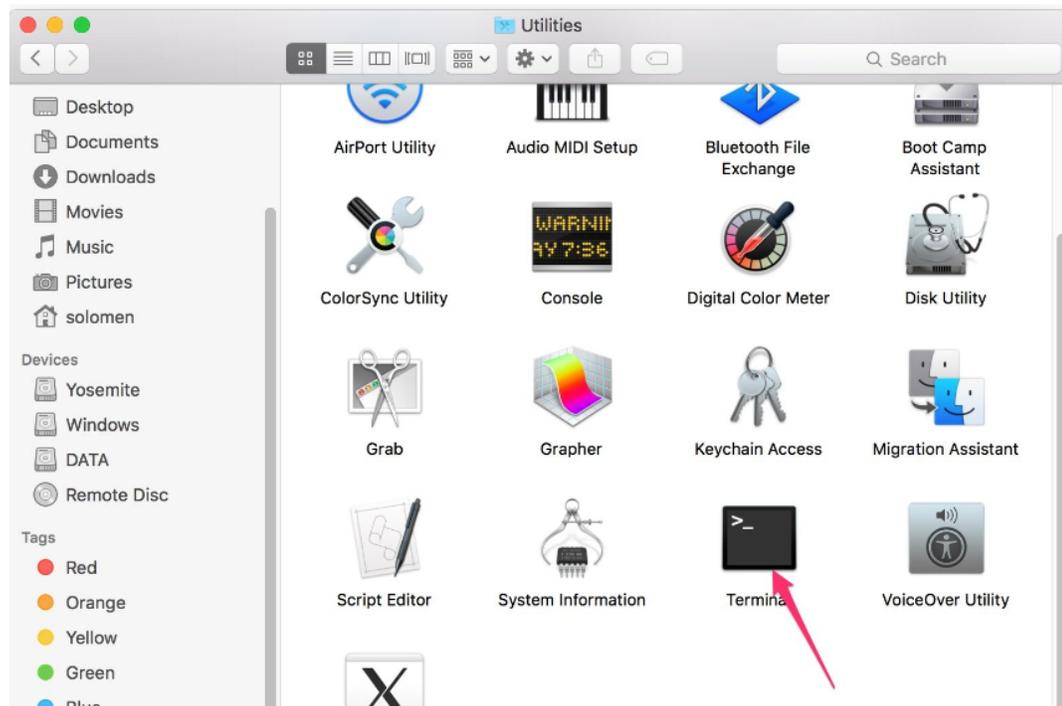


4. Jetzt können Sie den Desktop des Raspberry Pi sehen.



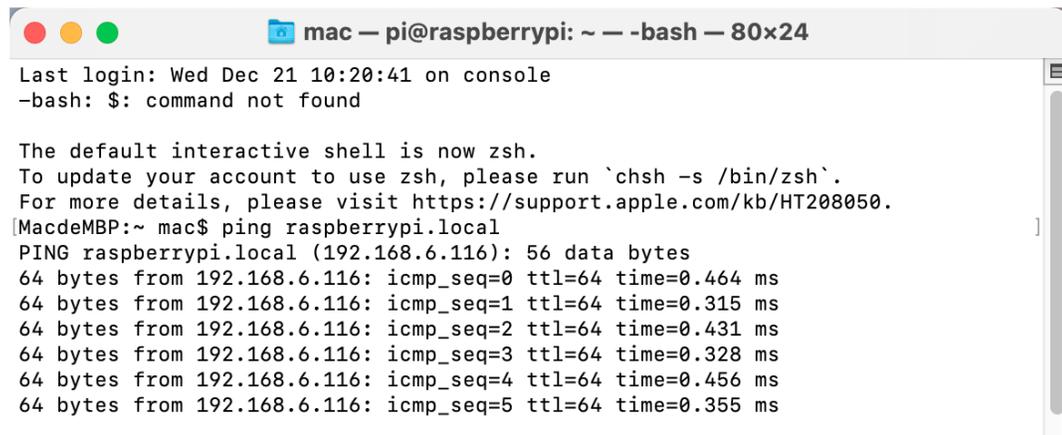
Linux/Unix-Nutzer

1. Gehen Sie zu **Applications** ->**Utilities**, finden Sie das **Terminal** und öffnen Sie es.



2. Überprüfen Sie, ob Ihr Raspberry Pi im selben Netzwerk ist, indem Sie `ping <hostname>.local` eingeben.

```
ping raspberrypi.local
```



Wie oben gezeigt, können Sie die IP-Adresse des Raspberry Pi sehen, nachdem er mit dem Netzwerk verbunden wurde.

- Wenn das Terminal Ping request could not find host pi.local. Please check the name and try again. meldet, folgen Sie den Anweisungen, um sicherzustellen, dass der Hostname, den Sie eingeben, korrekt ist.
- Bekommen Sie immer noch keine IP? Überprüfen Sie Ihre Netzwerk- oder WLAN-Konfiguration auf dem Raspberry Pi.

3. Geben Sie `ssh <username>@<hostname>.local` (oder `ssh <username>@<IP address>`) ein.

```
ssh pi@raspberrypi.local
```

Bemerkung: Wenn eine Aufforderung erscheint The term 'ssh' is not recognized as the name of a cmdlet...

Das bedeutet, dass Ihr System zu alt ist und keine vorinstallierten SSH-Tools hat, Sie müssen manuell *Installieren Sie OpenSSH über Powershell* installieren.

Oder verwenden Sie ein Drittanbieter-Tool wie *PuTTY*.

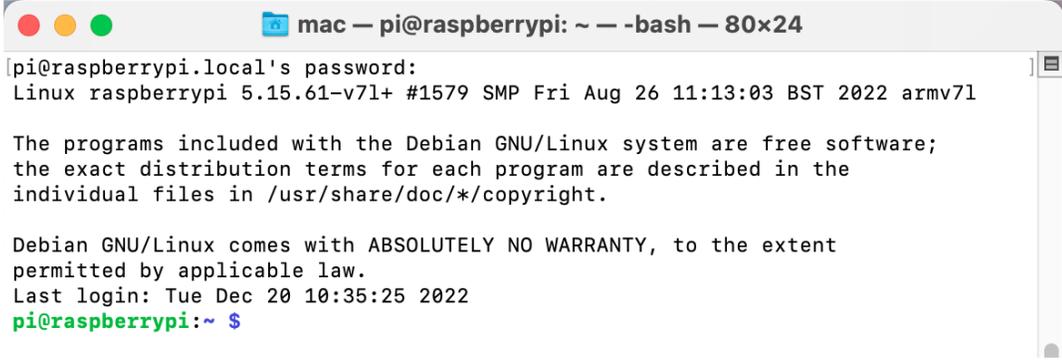
- Die folgende Nachricht wird nur beim ersten Anmelden angezeigt, also geben Sie yes ein.

```
The authenticity of host 'raspberrypi.local'
(2400:2410:2101:5800:635b:f0b6:2662:8cba) can't be established.
ED25519 key fingerprint is SHA256:oo7x3ZSgAo032wD1tE8eW0fFM/
kmewIvRwkBys6XRwg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

- Geben Sie das zuvor festgelegte Passwort ein. (Meines ist raspberry.)

Bemerkung: Wenn Sie das Passwort eingeben, werden die Zeichen nicht im Fenster angezeigt, was normal ist. Wichtig ist, dass Sie das korrekte Passwort eingeben.

- Wir haben jetzt den Raspberry Pi verbunden und sind bereit für den nächsten Schritt.



```
mac — pi@raspberrypi: ~ — -bash — 80x24
pi@raspberrypi.local's password:
Linux raspberrypi 5.15.61-v7l+ #1579 SMP Fri Aug 26 11:13:03 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 20 10:35:25 2022
pi@raspberrypi:~ $
```

4.1.4 Installiere Alle Module(Wichtig)

Stellen Sie sicher, dass Sie mit dem Internet verbunden sind und aktualisieren Sie Ihr System:

```
sudo apt update
sudo apt upgrade
```

Bemerkung: Python3-bezogene Pakete müssen installiert werden, wenn Sie die Lite-Version des Betriebssystems installieren.

```
sudo apt install git python3-pip python3-setuptools python3-smbus
```

Installieren Sie robot-hat.

```
cd ~/
git clone -b v2.0 https://github.com/sunfounder/robot-hat.git
cd robot-hat
sudo python3 setup.py install
```

Dann laden Sie das Modul vilib herunter und installieren Sie es.

```
cd ~/
git clone -b picamera2 https://github.com/sunfounder/vilib.git
cd vilib
sudo python3 install.py
```

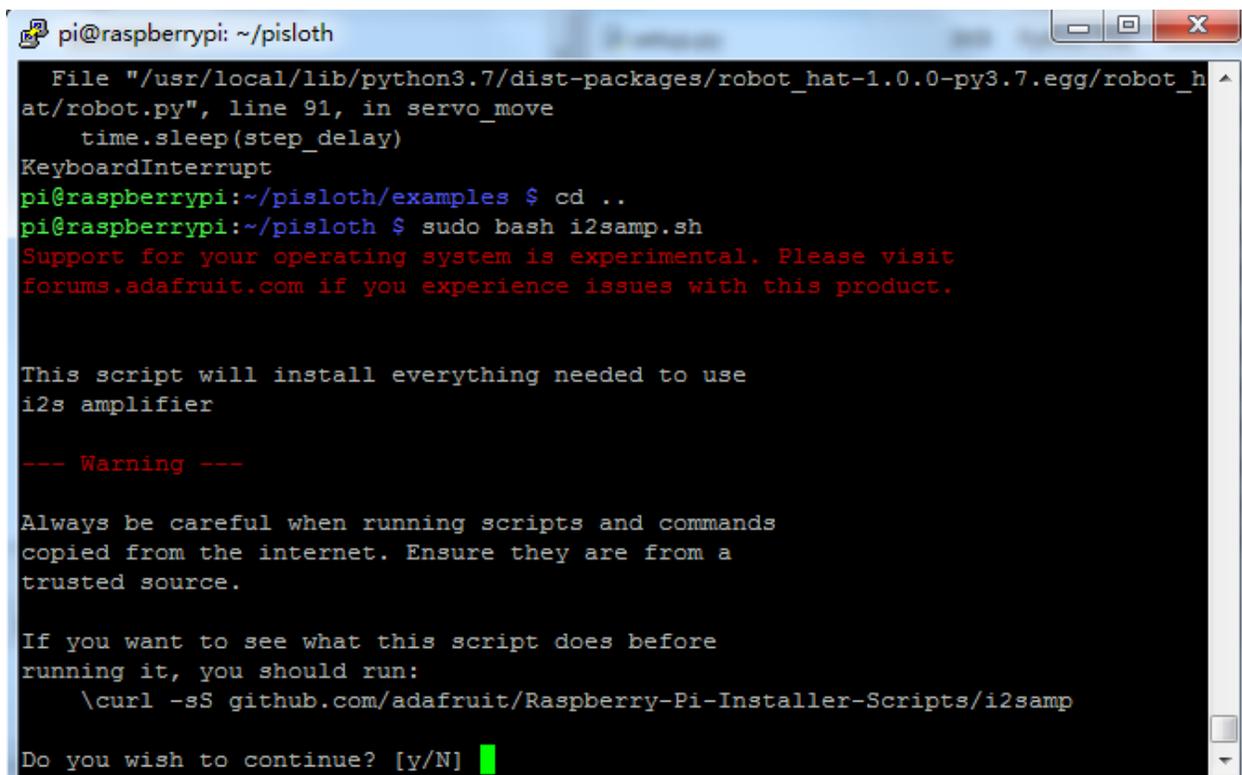
Laden Sie das Modul picar-x herunter und installieren Sie es.

```
cd ~/
git clone -b v2.0 https://github.com/sunfounder/picar-x.git
cd picar-x
sudo python3 setup.py install
```

Dieser Schritt dauert eine Weile, bitte haben Sie Geduld.

Schließlich müssen Sie das Skript i2samp.sh ausführen, um die Komponenten zu installieren, die der i2s-Verstärker benötigt, sonst hat der Picar-X keinen Ton.

```
cd ~/picar-x
sudo bash i2samp.sh
```



```
pi@raspberrypi: ~/pisloth
File "/usr/local/lib/python3.7/dist-packages/robot_hat-1.0.0-py3.7.egg/robot_hat/robot.py", line 91, in servo_move
    time.sleep(step_delay)
KeyboardInterrupt
pi@raspberrypi:~/pisloth/examples $ cd ..
pi@raspberrypi:~/pisloth $ sudo bash i2samp.sh
Support for your operating system is experimental. Please visit
forums.adafruit.com if you experience issues with this product.

This script will install everything needed to use
i2s amplifier

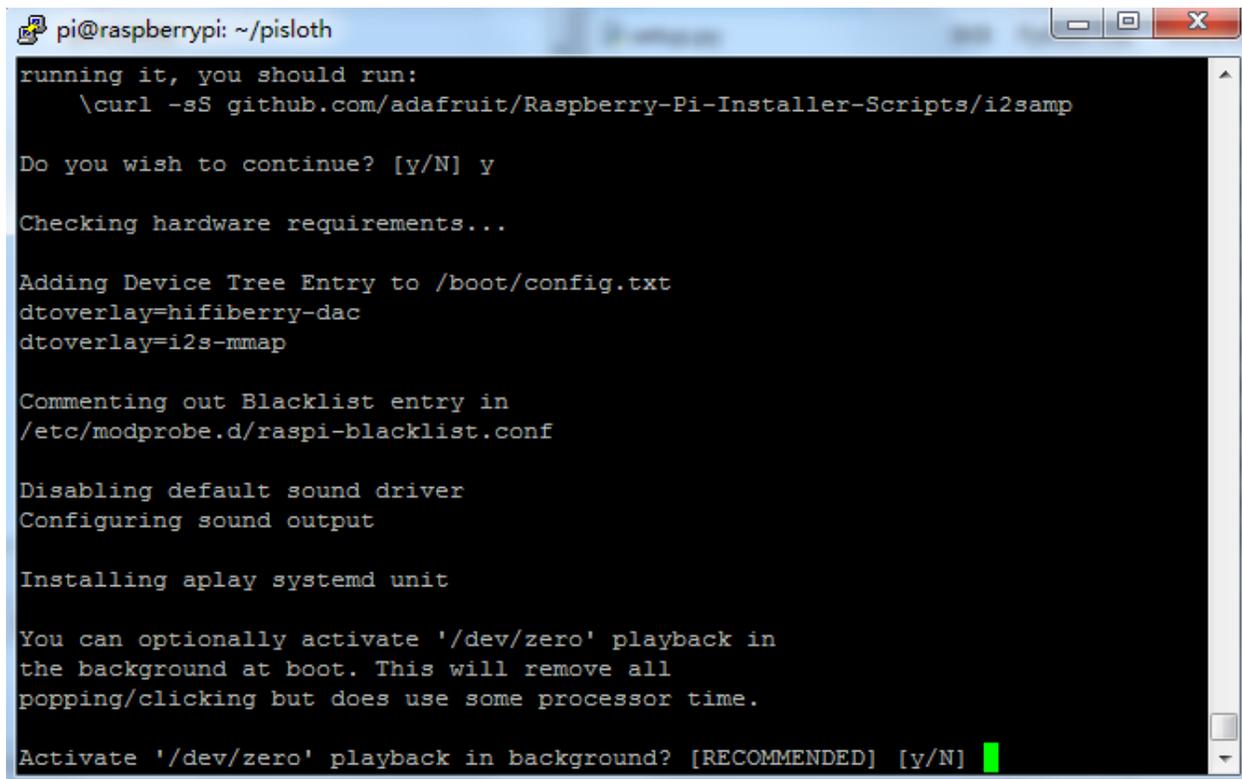
--- Warning ---

Always be careful when running scripts and commands
copied from the internet. Ensure they are from a
trusted source.

If you want to see what this script does before
running it, you should run:
    \curl -sS github.com/adafruit/Raspberry-Pi-Installer-Scripts/i2samp

Do you wish to continue? [y/N] █
```

Geben Sie y ein und drücken Sie Enter, um das Skript weiter auszuführen.



```
pi@raspberrypi: ~/pislloth
running it, you should run:
  \curl -sS github.com/adafruit/Raspberry-Pi-Installer-Scripts/i2samp

Do you wish to continue? [y/N] y

Checking hardware requirements...

Adding Device Tree Entry to /boot/config.txt
dtoverlay=hifiberry-dac
dtoverlay=i2s-mmap

Commenting out Blacklist entry in
/etc/modprobe.d/raspi-blacklist.conf

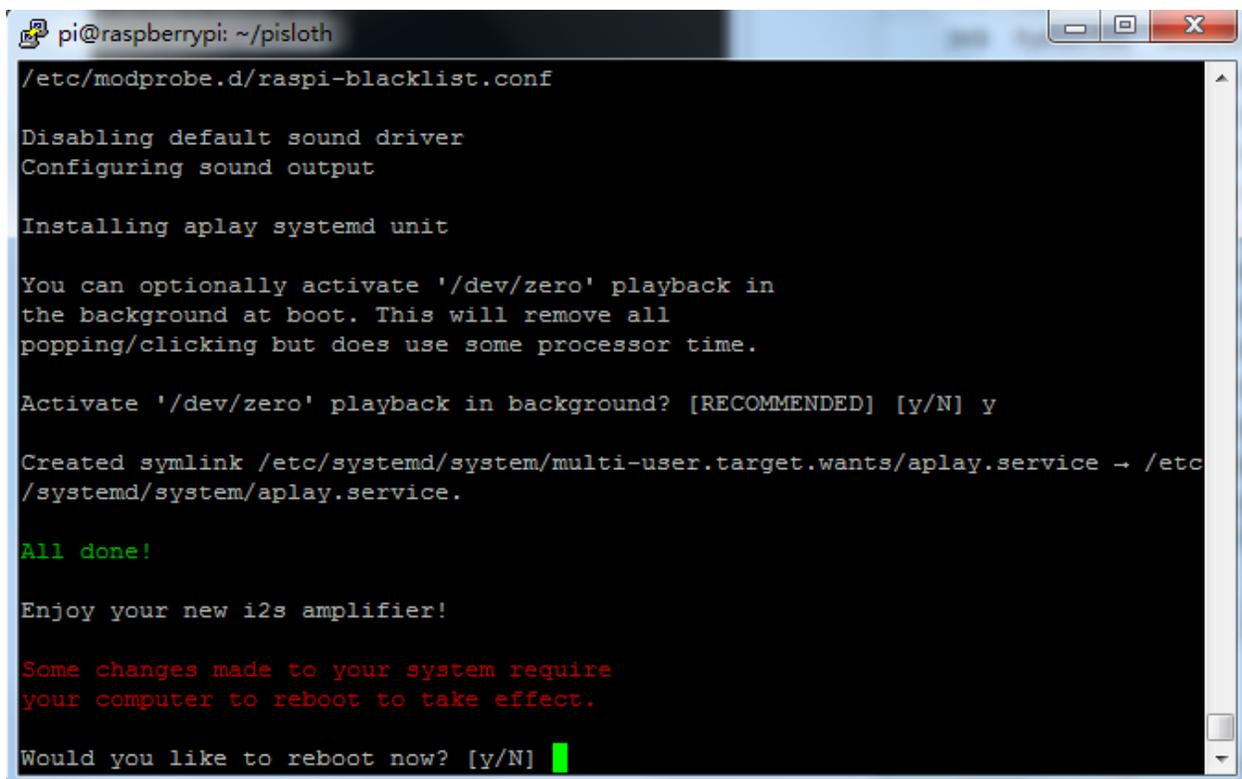
Disabling default sound driver
Configuring sound output

Installing aplay systemd unit

You can optionally activate '/dev/zero' playback in
the background at boot. This will remove all
popping/clicking but does use some processor time.

Activate '/dev/zero' playback in background? [RECOMMENDED] [y/N] █
```

Geben Sie y ein und drücken Sie Enter, um /dev/zero im Hintergrund laufen zu lassen.



```
pi@raspberrypi: ~/pislloth
/etc/modprobe.d/raspi-blacklist.conf

Disabling default sound driver
Configuring sound output

Installing aplay systemd unit

You can optionally activate '/dev/zero' playback in
the background at boot. This will remove all
popping/clicking but does use some processor time.

Activate '/dev/zero' playback in background? [RECOMMENDED] [y/N] y

Created symlink /etc/systemd/system/multi-user.target.wants/aplay.service → /etc
/systemd/system/aplay.service.

All done!

Enjoy your new i2s amplifier!

Some changes made to your system require
your computer to reboot to take effect.

Would you like to reboot now? [y/N] █
```

Geben Sie y ein und drücken Sie Enter, um den Picar-X neu zu starten.

Bemerkung: Wenn nach dem Neustart kein Ton vorhanden ist, müssen Sie möglicherweise das i2samp.sh-Skript mehrmals ausführen.

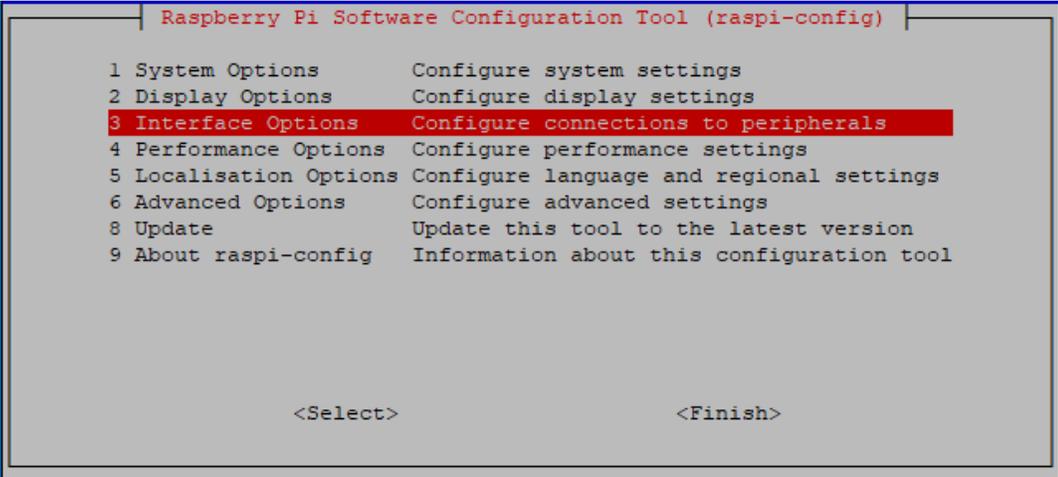
4.1.5 Aktivierung des I2C-Interface(Wichtig)

Hier verwenden wir die I2C-Schnittstellen des Raspberry Pi, standardmäßig sind diese jedoch deaktiviert, daher müssen wir sie zunächst aktivieren.

1. Geben Sie den folgenden Befehl ein:

```
sudo raspi-config
```

2. Wählen Sie **Interfacing Options** durch Drücken der Abwärtsfeiltaste auf Ihrer Tastatur, dann drücken Sie die **Enter**-Taste.

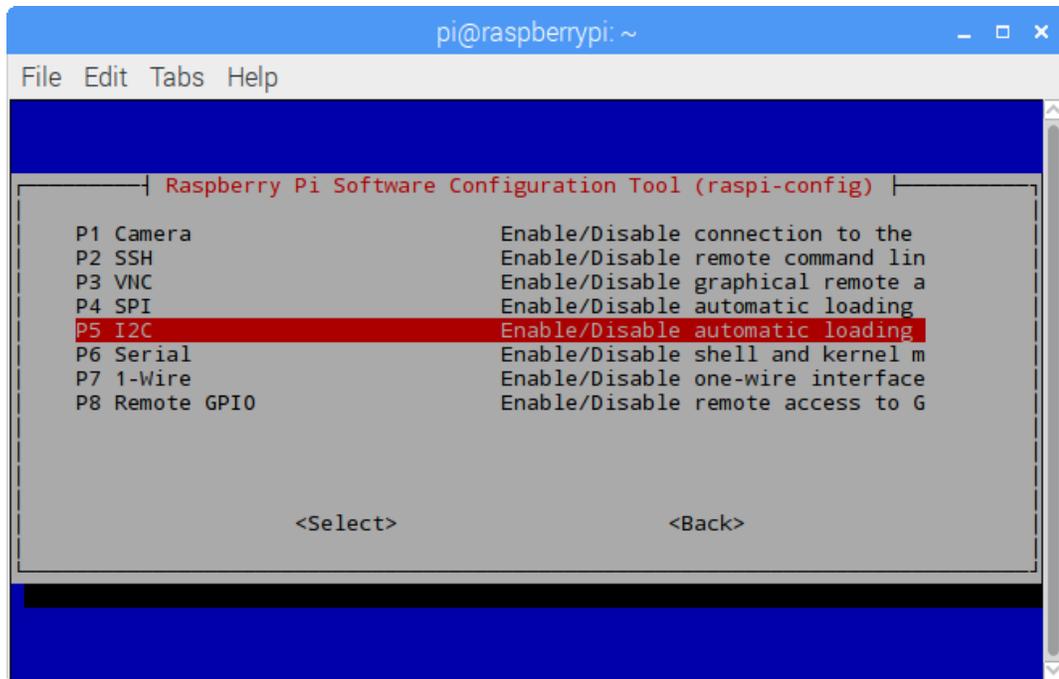


```
Raspberry Pi Software Configuration Tool (raspi-config)

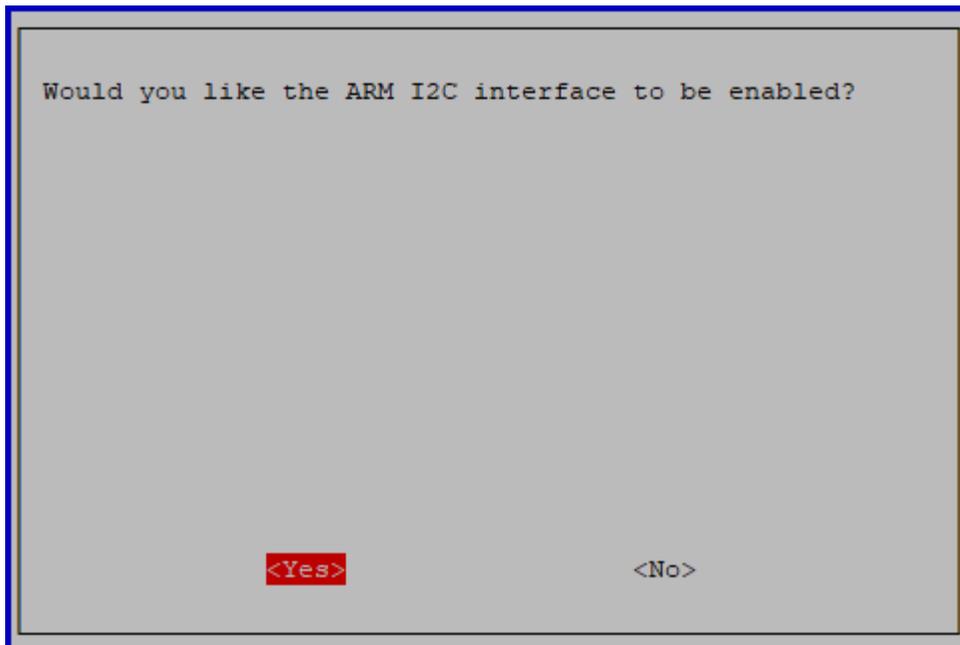
1 System Options      Configure system settings
2 Display Options     Configure display settings
3 Interface Options   Configure connections to peripherals
4 Performance Options Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options   Configure advanced settings
8 Update              Update this tool to the latest version
9 About raspi-config  Information about this configuration tool

<Select>                <Finish>
```

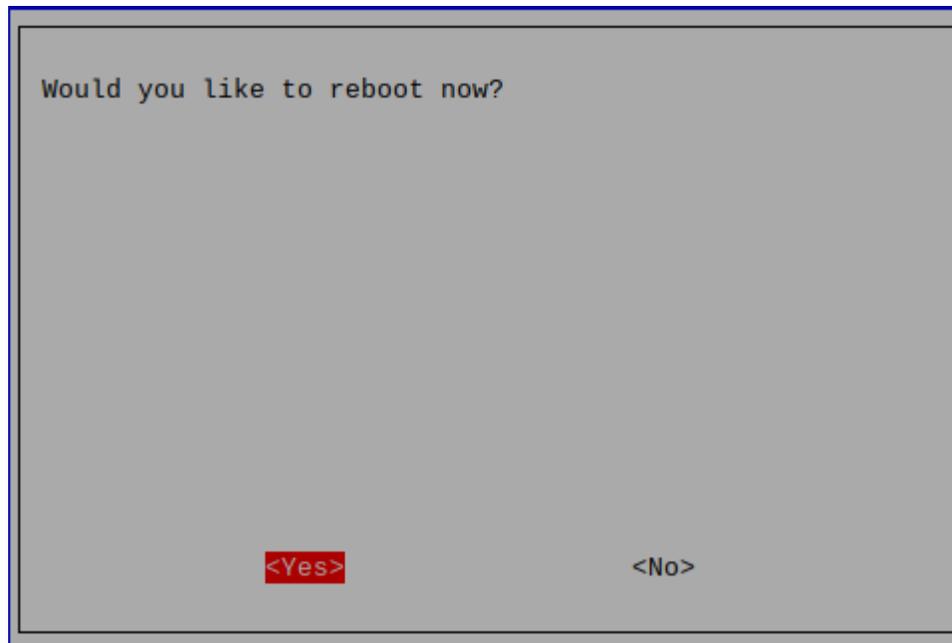
3. Dann **I2C**.



4. Verwenden Sie die Pfeiltasten auf der Tastatur, um **<yes>** -> **<OK>** auszuwählen und die Einrichtung des I2C abzuschließen.



5. Nachdem Sie **<Finish>** ausgewählt haben, erscheint ein Pop-up, das Sie daran erinnert, dass Sie neu starten müssen, damit die Einstellungen wirksam werden. Wählen Sie **<yes>**.

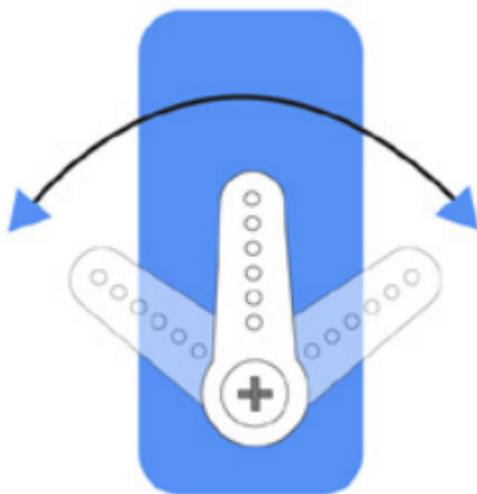


4.1.6 Servo-Justierung(Wichtig)

Der Winkelbereich des Servos liegt bei -90° ~ 90° Grad, aber der in der Fabrik eingestellte Winkel ist zufällig, vielleicht 0° , vielleicht 45° . Wenn wir ihn direkt mit einem solchen Winkel montieren, führt dies nach dem Ausführen des Codes des Roboters zu einem chaotischen Zustand oder schlimmer noch, es kann dazu führen, dass der Servo blockiert und durchbrennt.

Daher müssen wir hier alle Servowinkel auf 0° einstellen und sie dann montieren, sodass der Servowinkel in der Mitte ist, egal in welche Richtung er sich dreht.

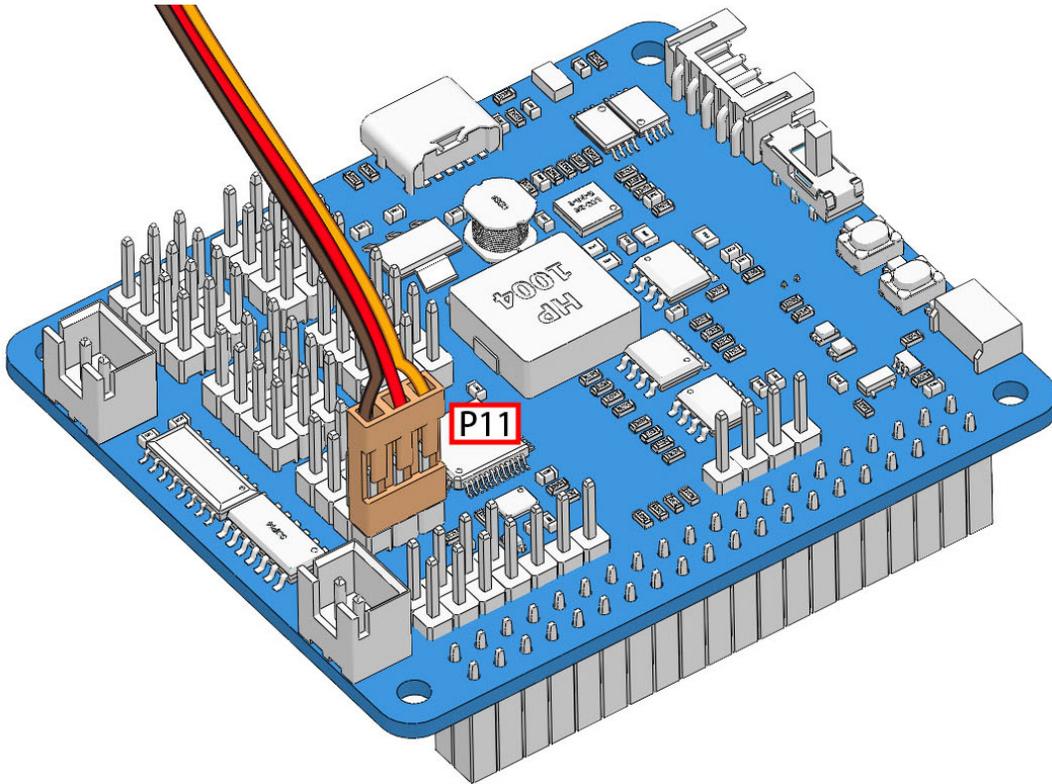
1. Um sicherzustellen, dass der Servo ordnungsgemäß auf 0° eingestellt wurde, stecken Sie zuerst den Servoarm in die Servowelle und drehen Sie dann vorsichtig den Schwenkarm in einen anderen Winkel. Dieser Servoarm dient lediglich dazu, Ihnen deutlich zu zeigen, dass der Servo sich dreht.



2. Führen Sie nun `servo_zeroing.py` im Ordner `example/` aus.

```
cd ~/picar-x/example
sudo python3 servo_zeroing.py
```

3. Stecken Sie als Nächstes das Servokabel wie folgt in den P11-Port, gleichzeitig sehen Sie, dass sich der Servoarm in eine Position dreht (Dies ist die 0°-Position, die ein zufälliger Ort ist und möglicherweise nicht vertikal oder parallel ist).



4. Entfernen Sie jetzt den Servoarm, wobei das Servokabel angeschlossen bleibt, und schalten Sie den Strom nicht aus. Fahren Sie dann mit der Montage gemäß den Papieranweisungen fort.

Bemerkung:

- Ziehen Sie dieses Servokabel nicht ab, bevor Sie es mit der Servoschraube befestigt haben; Sie können es nach der Befestigung abziehen.
- Drehen Sie den Servo nicht, während er eingeschaltet ist, um Schäden zu vermeiden. Wenn die Servowelle nicht im richtigen Winkel eingeführt ist, ziehen Sie den Servo heraus und stecken Sie ihn erneut ein.
- Bevor Sie jeden Servo montieren, müssen Sie das Servokabel in P11 einstecken und den Strom einschalten, um seinen Winkel auf 0° einzustellen.

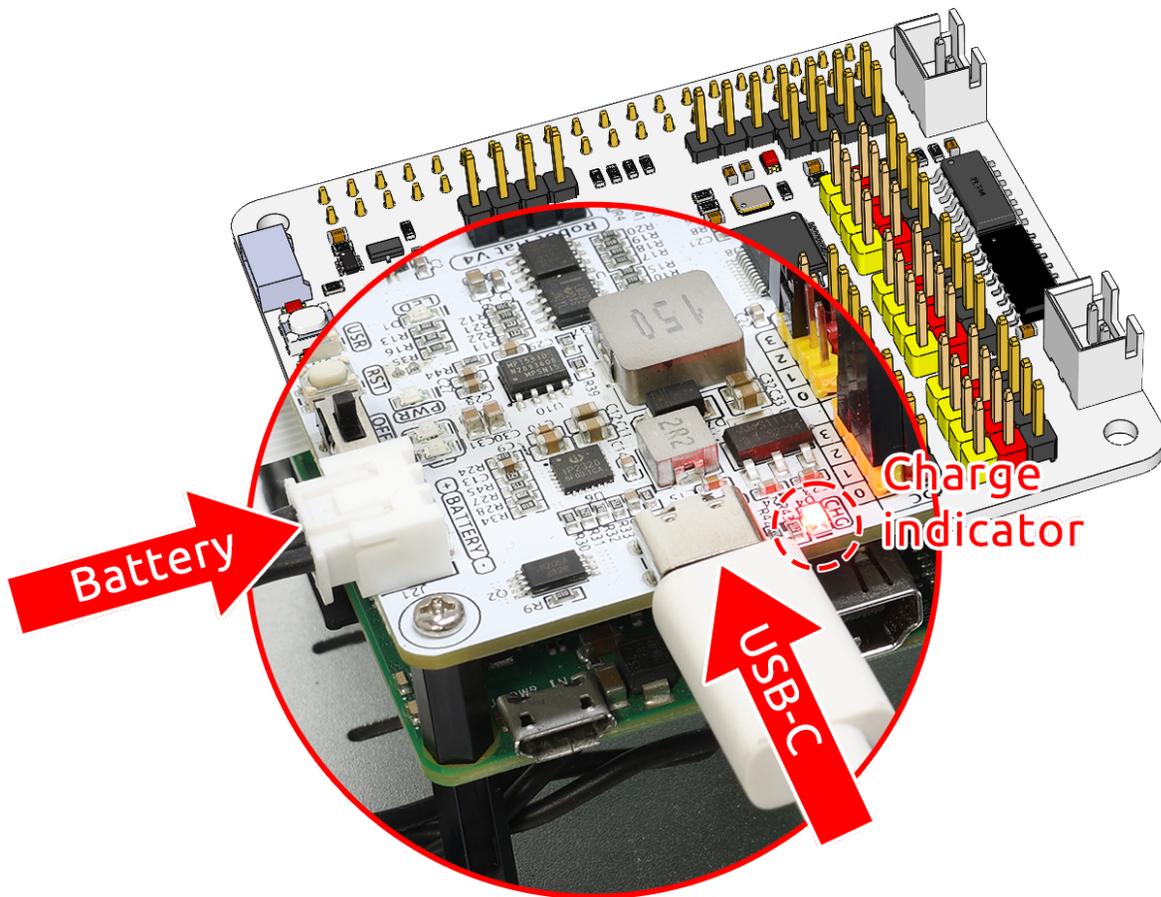
Video

In unserem Montagevideo von 6:25 bis 8:48 gibt es auch ein detailliertes Tutorial zu diesem Kapitel. Sie können den Videoanweisungen direkt folgen.

4.2 Einschalten & Laden

4.2.1 Laden

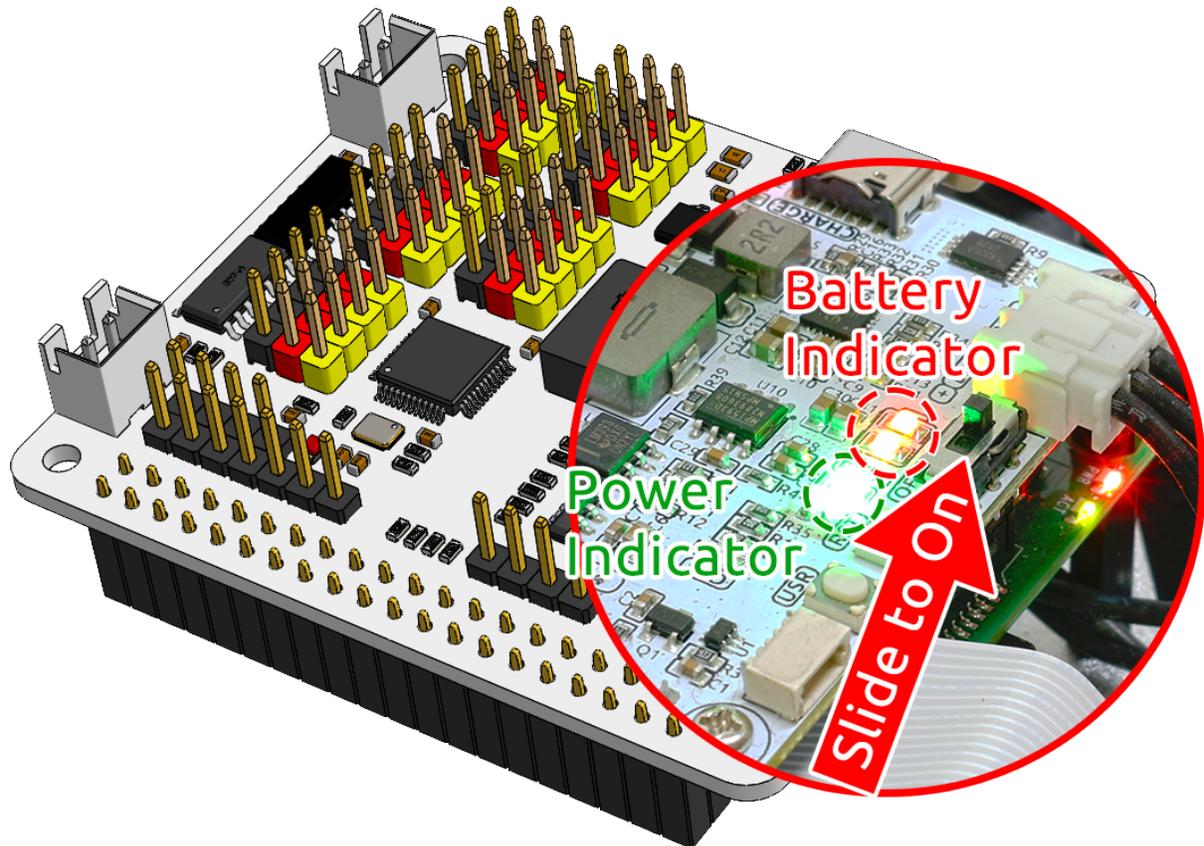
Stecken Sie das Batteriekabel ein. Anschließend stecken Sie das USB-C-Kabel ein, um die Batterie aufzuladen. Sie müssen Ihr eigenes Ladegerät bereitstellen; wir empfehlen ein 5V 3A Ladegerät, oder Ihr üblicherweise verwendetes Smartphone-Ladegerät wird ausreichen.



Bemerkung: Schließen Sie eine externe Typ-C-Stromquelle an den Typ-C-Anschluss auf dem Roboterhut an; er wird sofort mit dem Laden der Batterie beginnen, und eine rote Anzeigelampe wird aufleuchten. Wenn die Batterie vollständig aufgeladen ist, wird das rote Licht automatisch erlöschen.

4.2.2 Einschalten

Schalten Sie den Netzschalter ein. Die Power-Anzeigeleuchte und die Batteriestandanzeige werden aufleuchten.



Warten Sie einige Sekunden, und Sie werden einen leichten Piepton hören, der darauf hinweist, dass der Raspberry Pi erfolgreich gestartet wurde.

Bemerkung: Wenn beide Batteriestandanzeigeleuchten ausgeschaltet sind, laden Sie bitte die Batterie auf. Wenn Sie längere Programmier- oder Debugging-Sitzungen benötigen, können Sie den Raspberry Pi operational halten, indem Sie gleichzeitig das USB-C-Kabel einstecken, um die Batterie aufzuladen.

4.2.3 18650 Batterie



- VCC: Batteriepositivpol, hier gibt es zwei Sätze von VCC und GND, um den Strom zu erhöhen und den Widerstand zu verringern.
- Mittel: Zur Spannungsausgleich zwischen den beiden Zellen und somit zum Schutz der Batterie.
- GND: Negativer Batteriepol.

Dies ist ein maßgefertigtes Batteriepaket von SunFounder, bestehend aus zwei 18650 Batterien mit einer Kapazität von 2000mAh. Der Stecker ist XH2.54 3P, der direkt nach dem Einsetzen in den Schild aufgeladen werden kann.

Merkmale

- Batterieladung: 5V/2A
- Batterieausgang: 5V/5A
- Batteriekapazität: 3.7V 2000mAh x 2
- Batterielebensdauer: 90 Minuten
- Batterieladezeit: 130 Minuten

- Stecker: XH2.54 3P

Nachdem die Montage des PiCar-X abgeschlossen ist, versuchen Sie, die untenstehenden Projekte auszuführen:

4.3 0. Kalibrierung des PiCar-X

4.3.1 Kalibrierung von Motoren & Servo

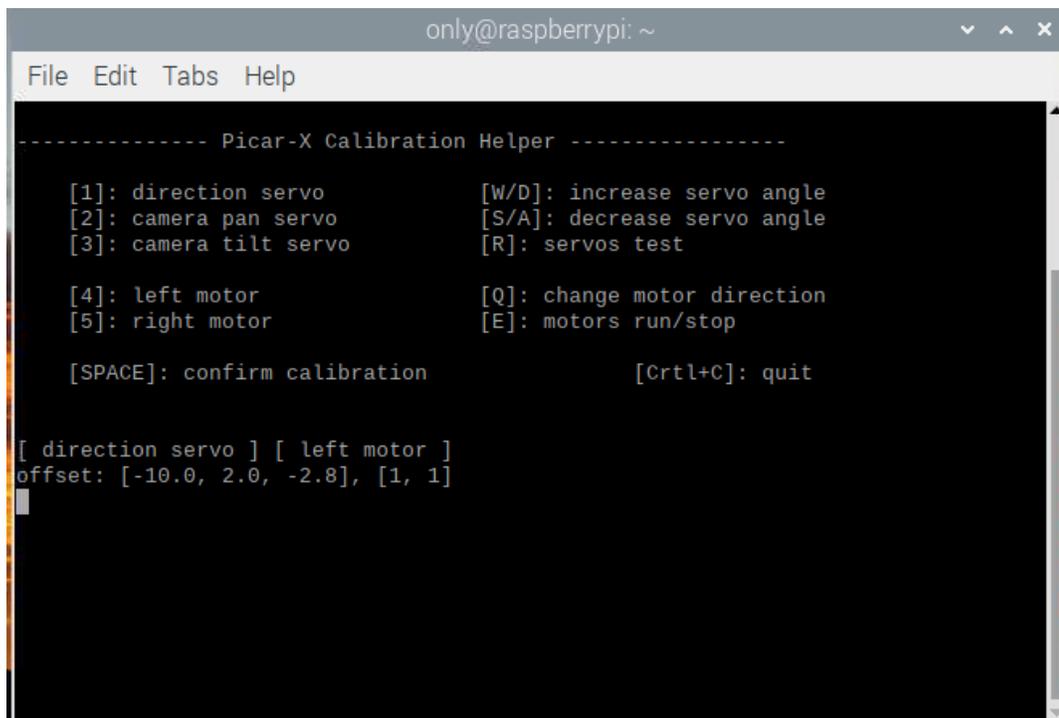
Einige Servowinkel können aufgrund möglicher Abweichungen bei der Montage des PiCar-X oder aufgrund der Einschränkungen der Servos selbst leicht geneigt sein, daher können Sie diese kalibrieren.

Natürlich können Sie dieses Kapitel überspringen, wenn Sie denken, dass die Montage perfekt ist und keine Kalibrierung erfordert.

1. Führen Sie `calibration.py` aus.

```
cd ~/picar-x/example/calibration
sudo python3 calibration.py
```

2. Nachdem Sie den Code ausgeführt haben, wird folgende Oberfläche im Terminal angezeigt.



```
only@raspberrypi: ~
File Edit Tabs Help
----- Picar-X Calibration Helper -----

[1]: direction servo           [W/D]: increase servo angle
[2]: camera pan servo         [S/A]: decrease servo angle
[3]: camera tilt servo        [R]: servos test

[4]: left motor               [Q]: change motor direction
[5]: right motor              [E]: motors run/stop

[SPACE]: confirm calibration   [Ctrl+C]: quit

[ direction servo ] [ left motor ]
offset: [-10.0, 2.0, -2.8], [1, 1]
```

3. Die Taste R wird verwendet, um zu testen, ob die 3 Servos einwandfrei funktionieren. Nachdem ein Servo mit den Tasten 1, 2 oder 3 ausgewählt wurde, drücken Sie die Taste R, um diesen Servo zu testen.
4. Drücken Sie die Zahlentaste 1, um den Servo des Vorderrads auszuwählen, und dann die Tasten W/S, damit das Vorderrad so gerade wie möglich aussieht, ohne nach links oder rechts abzuweichen.

```
only@raspberrypi: ~
File Edit Tabs Help
----- Picar-X Calibration Helper -----
[1]: direction servo      [W/D]: increase servo angle
[2]: camera pan servo    [S/A]: decrease servo angle
[3]: camera tilt servo   [R]: servos test

[4]: left motor          [Q]: change motor direction
[5]: right motor        [E]: motors run/stop

[SPACE]: confirm calibration      [Ctrl+C]: quit

[ direction servo ] [ left motor ]
offset: [-10.0, 2.0, -2.8], [1, 1]
```

5. Drücken Sie die Zahlentaste 2, um den **Pan servo** auszuwählen, dann drücken Sie die Tasten W/S, damit die Schwenk-/Neigeplattform geradeaus schaut und nicht nach links oder rechts kippt.

```
only@raspberrypi: ~
File Edit Tabs Help
----- Picar-X Calibration Helper -----
[1]: direction servo      [W/D]: increase servo angle
[2]: camera pan servo    [S/A]: decrease servo angle
[3]: camera tilt servo   [R]: servos test

[4]: left motor          [Q]: change motor direction
[5]: right motor        [E]: motors run/stop

[SPACE]: confirm calibration      [Ctrl+C]: quit

[ direction servo ] [ left motor ]
offset: [-10.0, 2.0, -2.8], [1, 1]
```

6. Drücken Sie die Zahlentaste 3, um den **tilt servo** auszuwählen, dann drücken Sie die Tasten W/S, damit die Schwenk-/Neigeplattform geradeaus schaut und nicht nach oben oder unten kippt.

```
only@raspberrypi: ~
File Edit Tabs Help
----- Picar-X Calibration Helper -----
[1]: direction servo           [W/D]: increase servo angle
[2]: camera pan servo         [S/A]: decrease servo angle
[3]: camera tilt servo        [R]: servos test

[4]: left motor               [Q]: change motor direction
[5]: right motor              [E]: motors run/stop

[SPACE]: confirm calibration   [Ctrl+C]: quit

[ direction servo ] [ left motor ]
offset: [-10.0, 2.0, -2.8], [1, 1]
```

7. Da die Verkabelung der Motoren bei der Installation umgekehrt sein könnte, können Sie E drücken, um zu testen, ob das Auto normal vorwärtsfahren kann. Wenn nicht, verwenden Sie die Zahlentasten 4 und 5, um die linken und rechten Motoren auszuwählen, und dann drücken Sie die Taste Q, um die Drehrichtung zu kalibrieren.

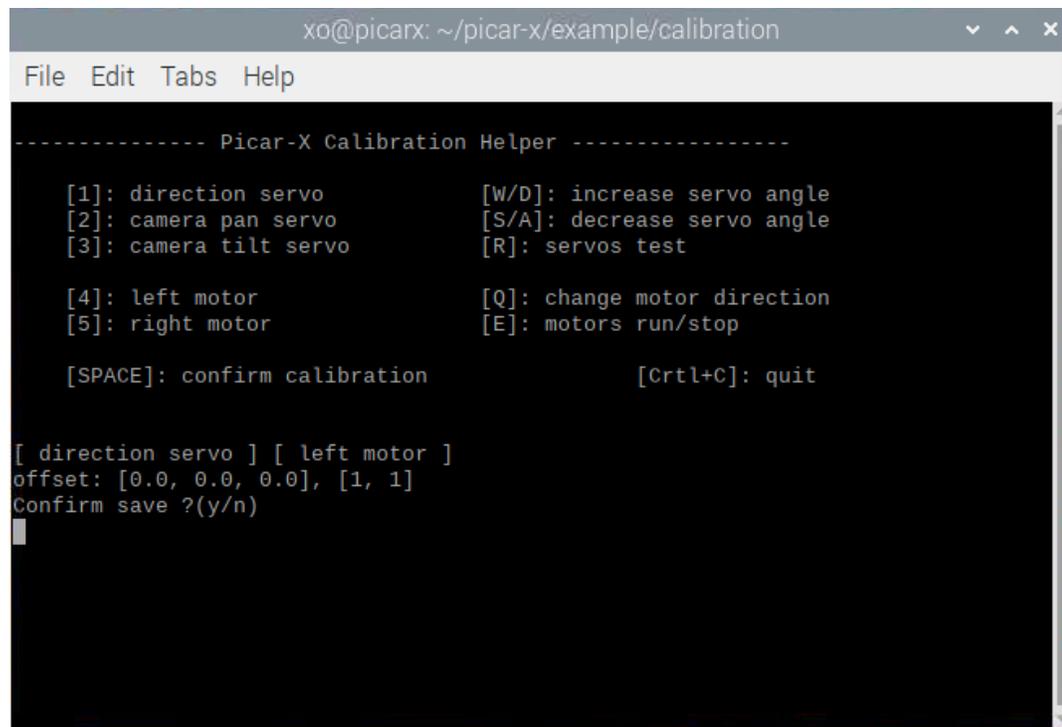
```
only@raspberrypi: ~
File Edit Tabs Help
----- Picar-X Calibration Helper -----
[1]: direction servo           [W/D]: increase servo angle
[2]: camera pan servo         [S/A]: decrease servo angle
[3]: camera tilt servo        [R]: servos test

[4]: left motor               [Q]: change motor direction
[5]: right motor              [E]: motors run/stop

[SPACE]: confirm calibration   [Ctrl+C]: quit

[ direction servo ] [ left motor ]
offset: [-10.0, 2.0, -2.8], [1, 1]
```

8. Wenn die Kalibrierung abgeschlossen ist, drücken Sie die Spacebar, um die Kalibrierungsparameter zu speichern. Es erscheint eine Aufforderung, y zur Bestätigung einzugeben, und dann drücken Sie Ctrl+C, um das Programm zu beenden und die Kalibrierung abzuschließen.



```
xo@picarx: ~/picar-x/example/calibration
File Edit Tabs Help
----- Picar-X Calibration Helper -----

[1]: direction servo          [W/D]: increase servo angle
[2]: camera pan servo        [S/A]: decrease servo angle
[3]: camera tilt servo       [R]: servos test

[4]: left motor              [Q]: change motor direction
[5]: right motor             [E]: motors run/stop

[SPACE]: confirm calibration  [Ctrl+C]: quit

[ direction servo ] [ left motor ]
offset: [0.0, 0.0, 0.0], [1, 1]
Confirm save ?(y/n)
```

4.3.2 Kalibrierung des Graustufenmoduls

Aufgrund unterschiedlicher Umgebungsbedingungen und Beleuchtungssituationen sind die voreingestellten Parameter für das Graustufenmodul möglicherweise nicht optimal. Sie können diese Einstellungen über dieses Programm feinjustieren, um bessere Ergebnisse zu erzielen.

1. Legen Sie einen Streifen schwarzes Isolierband, etwa 15 cm lang, auf einen hellen Boden. Zentrieren Sie Ihr PiCar-X, sodass es über dem Band steht. Dabei sollte der mittlere Sensor des Graustufenmoduls direkt über dem Band sein, während die beiden flankierenden Sensoren über der helleren Oberfläche schweben sollten.
2. Führen Sie `grayscale_calibration.py` aus.

```
cd ~/picar-x/example/calibration
sudo python3 grayscale_calibration.py
```

3. Nachdem Sie den Code ausgeführt haben, wird folgende Oberfläche im Terminal angezeigt.

```

<frozen importlib._bootstrap>:228: RuntimeWarning: Your system is neon capable but
pygame was not built with support for it. The performance of some of your blits c
ould be adversely affected. Consider enabling compile time detection with environm
ent variables like PYGAME_DETECT_AVX2=1 if you are compiling without cross compila
tion.

Picar-X Grayscale Module Reference
Calibration Helper

config_file: /opt/picar-x/picar-x.conf

press [Q] to start line reference calibration,
press [E] to start cliff reference calibration

[SPACE]: confirm calibration          [Ctrl+C]: quit

-----

current value: [854, 791, 922]
thresholds: [[56, 872], [82, 820], [83, 991]]
line reference: [1000.0, 1000.0, 1000.0]
cliff reference: [500.0, 500.0, 500.0]

```

4. Drücken Sie die Taste „Q“, um die Kalibrierung des Graustufenmoduls zu starten. Sie werden dann beobachten, wie das PiCar-X geringfügige Bewegungen nach links und rechts macht. Während dieses Vorgangs sollten alle drei Sensoren mindestens einmal über das Isolierband streichen.
5. Zusätzlich werden Sie drei Paare von deutlich unterschiedlichen Werten im Abschnitt „Schwellenwert“ bemerken, während „Linienreferenz“ zwei Zwischenwerte anzeigt, von denen jeder den Durchschnitt eines dieser Paare darstellt.

```

<frozen importlib._bootstrap>:228: RuntimeWarning: Your system is neon capable but
pygame was not built with support for it. The performance of some of your blits c
ould be adversely affected. Consider enabling compile time detection with environm
ent variables like PYGAME_DETECT_AVX2=1 if you are compiling without cross compila
tion.

Picar-X Grayscale Module Reference
Calibration Helper

config_file: /opt/picar-x/picar-x.conf

press [Q] to start line reference calibration,
press [E] to start cliff reference calibration

[SPACE]: confirm calibration          [Ctrl+C]: quit

Line reference auto calibration done.
Note that cliff reference values shou be less than line reference values.
current value: [860, 242, 314]
thresholds: [[44, 873], [60, 820], [63, 1112]]
line reference: [511, 517, 639]
cliff reference: [558, 555, 544]

```

6. Suspendieren Sie anschließend das PiCar-X in der Luft (oder positionieren Sie es über einer Klippenkante) und

drücken Sie die Taste „E“. Sie werden beobachten, dass auch die „Klippenreferenz“-Werte entsprechend aktualisiert werden.

```
<frozen importlib._bootstrap>:228: RuntimeWarning: Your system is neon capable but
pygame was not built with support for it. The performance of some of your blits c
ould be adversely affected. Consider enabling compile time detection with environm
ent variables like PYGAME_DETECT_AVX2=1 if you are compiling without cross compila
tion.

Picar-X Grayscale Module Reference
Calibration Helper

config_file: /opt/picar-x/picar-x.conf

press [Q] to start line reference calibration,
press [E] to start cliff reference calibration

[SPACE]: confirm calibration          [Ctrl+C]: quit

Cliff reference auto calibration done.

current value: [4, 4, 4]
thresholds: [[0, 1517], [0, 1467], [0, 1519]]
line reference: [497, 472, 624]
cliff reference: [250, 238, 314]
```

7. Sobald Sie verifiziert haben, dass alle Werte korrekt sind, drücken Sie die „Leertaste“, um die Daten zu speichern. Sie können das Programm dann durch Drücken von Strg+C beenden.

4.4 1. Lassen Sie PiCar-X sich bewegen

Dies ist das erste Projekt, testen wir die grundlegende Bewegung des Picar-X.

Code ausführen

```
cd ~/picar-x/example
sudo python3 1.move.py
```

Nachdem der Code ausgeführt wurde, wird PiCar-X vorwärts fahren, in einer S-Form abbiegen, anhalten und den Kopf schütteln.

Code

Bemerkung: Sie können den untenstehenden Code **modifizieren/zurücksetzen/kopieren/ausführen/stoppen**. Bevor Sie das tun, müssen Sie jedoch zum Quellcodepfad wie `picar-x/example` gehen. Nachdem Sie den Code modifiziert haben, können Sie ihn direkt ausführen, um den Effekt zu sehen.

```
from picarx import Picarx
import time

if __name__ == "__main__":
```

(Fortsetzung auf der nächsten Seite)

```
try:
    px = Picarx()
    px.forward(30)
    time.sleep(0.5)
    for angle in range(0,35):
        px.set_dir_servo_angle(angle)
        time.sleep(0.01)
    for angle in range(35,-35,-1):
        px.set_dir_servo_angle(angle)
        time.sleep(0.01)
    for angle in range(-35,0):
        px.set_dir_servo_angle(angle)
        time.sleep(0.01)
    px.forward(0)
    time.sleep(1)

    for angle in range(0,35):
        px.set_camera_servo1_angle(angle)
        time.sleep(0.01)
    for angle in range(35,-35,-1):
        px.set_camera_servo1_angle(angle)
        time.sleep(0.01)
    for angle in range(-35,0):
        px.set_camera_servo1_angle(angle)
        time.sleep(0.01)
    for angle in range(0,35):
        px.set_camera_servo2_angle(angle)
        time.sleep(0.01)
    for angle in range(35,-35,-1):
        px.set_camera_servo2_angle(angle)
        time.sleep(0.01)
    for angle in range(-35,0):
        px.set_camera_servo2_angle(angle)
        time.sleep(0.01)

finally:
    px.forward(0)
```

Wie funktioniert des?

Die grundlegende Funktionalität von PiCar-X ist im Modul `picarx` enthalten, welches zur Steuerung von Lenkservos und Rädern verwendet werden kann, und lässt das PiCar-X vorwärtsfahren, in einer S-Form abbiegen oder den Kopf schütteln.

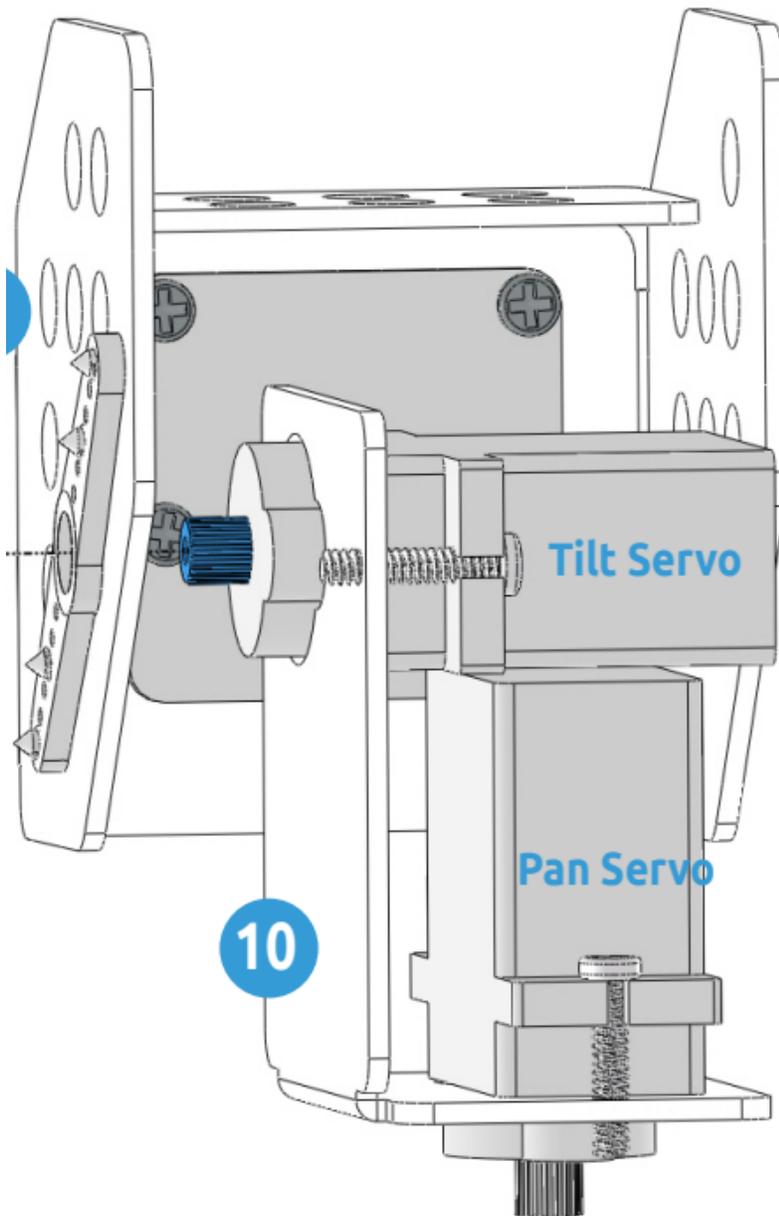
Nun werden die Bibliotheken importiert, die die grundlegende Funktionalität von PiCar-X unterstützen. Diese Zeilen erscheinen in allen Beispielen, die die Bewegung von PiCar-X involvieren.

```
from picarx import Picarx
import time
```

Die folgende Funktion mit der `for`-Schleife wird dann verwendet, um PiCar-X vorwärts zu bewegen, die Richtung zu ändern und die Schwenk-/Neigeplattform der Kamera zu bewegen.

```
px.forward(speed)
px.set_dir_servo_angle(angle)
px.set_camera_servo1_angle(angle)
px.set_camera_servo2_angle(angle)
```

- `forward()`: Befiehlt dem PiCar-X, mit einer bestimmten `speed` vorwärtszufahren.
- `set_dir_servo_angle`: Dreht den Lenkservo in einen bestimmten `angle`.
- `set_cam_pan_angle`: Dreht den Pan-Servo in einen bestimmten `angle`.
- `set_cam_tilt_angle`: Dreht den Neige-Servo in einen bestimmten `angle`.



4.5 2. Tastatursteuerung

In diesem Projekt lernen wir, wie man die Tastatur verwendet, um den PiCar-X aus der Ferne zu steuern. Sie können den PiCar-X vorwärts, rückwärts, nach links und nach rechts steuern.

Code ausführen

```
cd ~/picar-x/example
sudo python3 2.keyboard_control.py
```

Drücken Sie Tasten auf der Tastatur, um PiCar-X zu steuern!

- w: Vorwärts
- a: Links abbiegen
- s: Rückwärts
- d: Rechts abbiegen
- i: Kopf hoch
- k: Kopf runter
- j: Kopf nach links drehen
- l: Kopf nach rechts drehen
- ctrl + c: Zweimal drücken, um das Programm zu beenden

Code

```
from picarx import Picarx
from time import sleep
import readchar

manual = '''
Press keys on keyboard to control PiCar-X!
  w: Forward
  a: Turn left
  s: Backward
  d: Turn right
  i: Head up
  k: Head down
  j: Turn head left
  l: Turn head right
  ctrl + c: Press twice to exit the program
'''

def show_info():
    print("\033[H\033[J",end='') # clear terminal windows
    print(manual)

if __name__ == "__main__":
    try:
        pan_angle = 0
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
tilt_angle = 0
px = Picarx()
show_info()
while True:
    key = readchar.readkey()
    key = key.lower()
    if key in('wsadikjl'):
        if 'w' == key:
            px.set_dir_servo_angle(0)
            px.forward(80)
        elif 's' == key:
            px.set_dir_servo_angle(0)
            px.backward(80)
        elif 'a' == key:
            px.set_dir_servo_angle(-35)
            px.forward(80)
        elif 'd' == key:
            px.set_dir_servo_angle(35)
            px.forward(80)
        elif 'i' == key:
            tilt_angle+=5
            if tilt_angle>35:
                tilt_angle=35
        elif 'k' == key:
            tilt_angle-=5
            if tilt_angle<-35:
                tilt_angle=-35
        elif 'l' == key:
            pan_angle+=5
            if pan_angle>35:
                pan_angle=35
        elif 'j' == key:
            pan_angle-=5
            if pan_angle<-35:
                pan_angle=-35

        px.set_cam_tilt_angle(tilt_angle)
        px.set_cam_pan_angle(pan_angle)
        show_info()
        sleep(0.5)
        px.forward(0)

    elif key == readchar.key.CTRL_C:
        print("\n Quit")
        break

finally:
    px.set_cam_tilt_angle(0)
    px.set_cam_pan_angle(0)
    px.set_dir_servo_angle(0)
    px.stop()
    sleep(.2)
```

Wie funktioniert des?

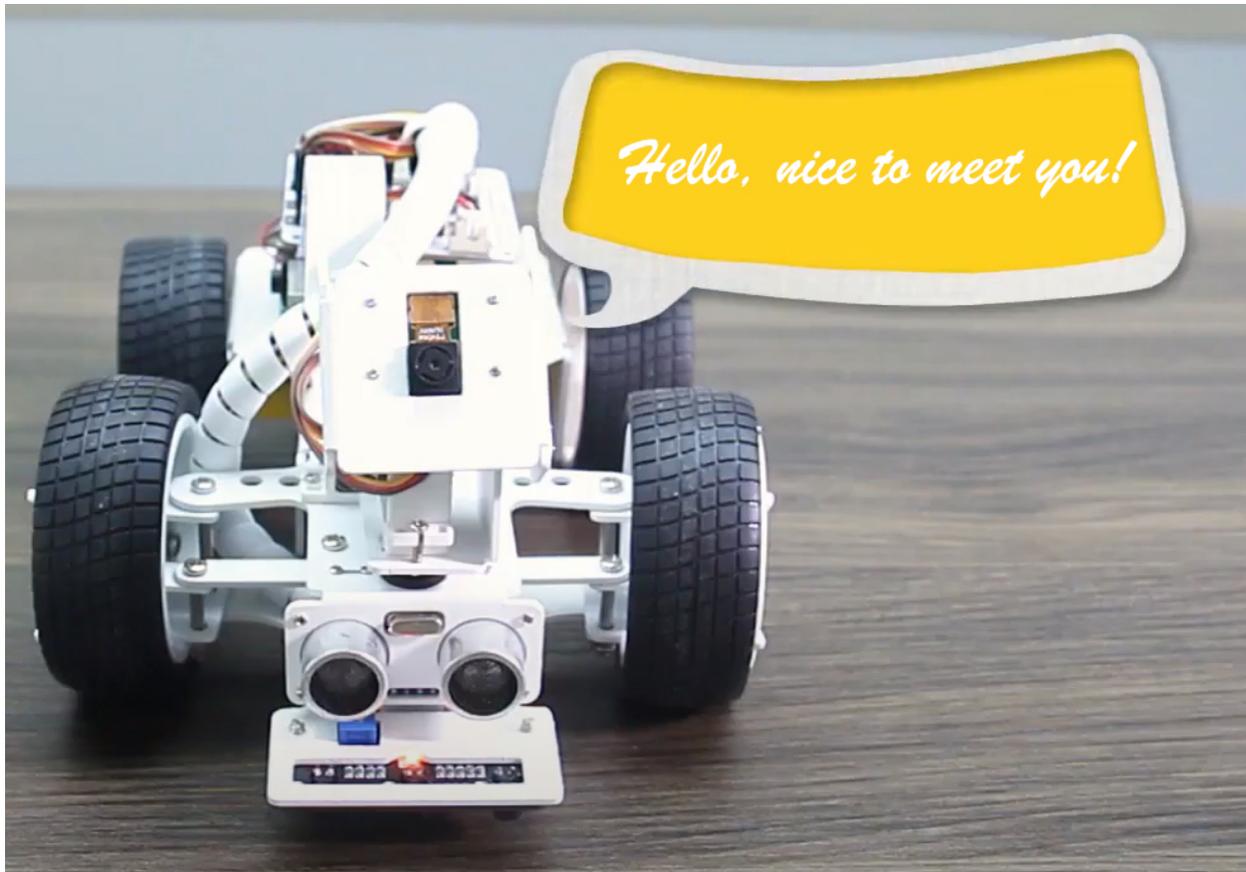
PiCar-X sollte entsprechende Aktionen basierend auf den gelesenen Tastaturzeichen ausführen. Die Funktion `lower()` wandelt Großbuchstaben in Kleinbuchstaben um, damit der Buchstabe unabhängig von der Groß- oder Kleinschreibung gültig bleibt.

```
while True:
    key = readchar.readkey()
    key = key.lower()
    if key in('wsadikjl'):
        if 'w' == key:
            pass
        elif 's' == key:
            pass
        elif 'a' == key:
            pass
        elif 'd' == key:
            pass
        elif 'i' == key:
            pass
        elif 'k' == key:
            pass
        elif 'l' == key:
            pass
        elif 'j' == key:
            pass

    elif key == readchar.key.CTRL_C:
        print("\n Quit")
        break
```

4.6 3. Text-zu-Sprache & Soundeffekte

In diesem Beispiel verwenden wir die Soundeffekte von PiCar-X (genauer gesagt, von Robot HAT). Es besteht aus drei Teilen, nämlich Musik, Sound und Text-zu-Sprache.



i2smp installieren

Bevor Sie die Funktionen Text-zu-Sprache (TTS) und Soundeffekte verwenden, aktivieren Sie zuerst den Lautsprecher, damit er eingeschaltet wird und Töne erzeugen kann.

Führen Sie `i2smp.sh` im `picar-x` Ordner aus, und dieses Skript installiert alles, was benötigt wird, um den i2s-Verstärker zu nutzen.

```
cd ~/picar-x  
sudo bash i2smp.sh
```

```

pi@raspberrypi:~/picar-x
File Edit Tabs Help
pi@raspberrypi:~ $ cd picar-x/
pi@raspberrypi:~/picar-x $ sudo bash i2samp.sh
Support for your operating system is experimental. Please visit
forums.adafruit.com if you experience issues with this product.

This script will install everything needed to use
i2s amplifier

--- Warning ---

Always be careful when running scripts and commands
copied from the internet. Ensure they are from a
trusted source.

If you want to see what this script does before
running it, you should run:
  \curl -sS github.com/adafruit/Raspberry-Pi-Installer-Scripts/i2samp

Do you wish to continue? [y/N] █

```

Es wird mehrere Aufforderungen geben, um die Anfrage zu bestätigen. Antworten Sie auf alle Aufforderungen mit einem **Y**. Nachdem die Änderungen am Raspberry Pi-System vorgenommen wurden, muss der Computer neu gestartet werden, damit diese Änderungen wirksam werden.

Nach dem Neustart führen Sie das Skript `i2samp.sh` erneut aus, um den Verstärker zu testen. Wenn ein Ton erfolgreich vom Lautsprecher abgespielt wird, ist die Konfiguration abgeschlossen.

Code ausführen

```

cd ~/picar-x/example
sudo python3 3.tts_example.py

```

Nachdem der Code ausgeführt wurde, bedienen Sie sich bitte gemäß der Aufforderung, die im Terminal angezeigt wird.

Taste drücken, um die Funktion aufzurufen!

- Leertaste: Soundeffekt abspielen (Autohupe)
- c: Soundeffekt mit Threads abspielen
- t: Text in Sprache umwandeln (Sagen Sie Hallo)
- q: Musik abspielen/stoppen

Code

```

from time import sleep
from robot_hat import Music, TTS
import readchar

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

music = Music()
tts = TTS()

manual = '''
Input key to call the function!
    space: Play sound effect (Car horn)
    c: Play sound effect with threads
    t: Text to speak
    q: Play/Stop Music
...

def main():
    print(manual)

    flag_bgm = False
    music.music_set_volume(20)
    tts.lang("en-US")

    while True:
        key = readchar.readkey()
        key = key.lower()
        if key == "q":
            flag_bgm = not flag_bgm
            if flag_bgm is True:
                music.music_play('../musics/slow-trail-Ahjay_Stelino.mp3')
            else:
                music.music_stop()

        elif key == readchar.key.SPACE:
            music.sound_play('../sounds/car-double-horn.wav')
            sleep(0.05)

        elif key == "c":
            music.sound_play_threading('../sounds/car-double-horn.wav')
            sleep(0.05)

        elif key == "t":
            words = "Hello"
            tts.say(words)

if __name__ == "__main__":
    main()

```

Wie funktioniert das?

Funktionen, die mit Hintergrundmusik zusammenhängen, umfassen Folgendes:

- `music = Music()` : Deklariert das Objekt.
- `music.music_set_volume(20)` : Stellt die Lautstärke ein, der Bereich liegt zwischen 0 und 100.
- `music.music_play('../musics/slow-trail-Ahjay_Stelino.mp3')` : Spielt Musikdateien ab, hier ist die Datei **slow-trail-Ahjay_Stelino.mp3** unter dem Pfad `../musics`.

- `music.music_stop()` : Stoppt das Abspielen von Hintergrundmusik.

Bemerkung: Sie können verschiedene Soundeffekte oder Musik zur `musics` oder `sounds`-Ordner hinzufügen über *Filezilla Software*.

Funktionen, die mit Soundeffekten zusammenhängen, umfassen Folgendes:

- `music = Music()`
- `music.sound_play('./sounds/car-double-horn.wav')` : Spielt die Soundeffekt-Datei ab.
- `music.sound_play_threading('./sounds/car-double-horn.wav')` : Spielt die Soundeffekt-Datei im neuen Thread-Modus ab, ohne den Haupt-Thread zu unterbrechen.

Die Software `eSpeak` wird verwendet, um die Funktionen von TTS zu implementieren.

Importieren Sie das TTS-Modul in `robot_hat`, das Funktionen zur Umwandlung von Text in Sprache kapselt.

Funktionen, die mit Text-zu-Sprache zusammenhängen, umfassen Folgendes:

- `tts = TTS()`
- `tts.say(words)` : Textaudio.
- `tts.lang("en-US")` : Setzt die Sprache.

Bemerkung: Stellen Sie die Sprache ein, indem Sie die Parameter von `lang("")` mit den folgenden Zeichen setzen.

Tab. 1: Language

zh-CN	Mandarin (Chinesisch)
en-US	Englisch-Vereinigte Staaten
en-GB	Englisch-Vereinigtes Königreich
de-DE	Deutschland-Deutsch
es-ES	Spanien-Spanisch
fr-FR	Frankreich-Französisch
it-IT	Italien-Italienisch

4.7 4. Hindernisvermeidung

In diesem Projekt wird PiCar-X Hindernisse vor sich erkennen, während es vorwärtsfährt, und wenn die Hindernisse zu nah sind, wird es die Fahrtrichtung ändern.

Code ausführen

```
cd ~/picar-x/example
sudo python3 4.avoiding_obstacles.py
```

Nachdem der Code ausgeführt wurde, fährt PiCar-X vorwärts.

Wenn es erkennt, dass die Entfernung des Hindernisses vor ihm weniger als 20 cm beträgt, wird es rückwärtsfahren.

Wenn ein Hindernis innerhalb von 20 bis 40 cm ist, wird es nach links abbiegen.

Wenn nach dem Linksabbiegen kein Hindernis in der Richtung ist oder die Hindernisentfernung größer als 25 cm ist, wird es weiter vorwärtsfahren.

Code

Bemerkung: Sie können den untenstehenden Code **modifizieren/zurücksetzen/kopieren/ausführen/stoppen**. Bevor Sie das tun, müssen Sie jedoch zum Quellcodepfad wie `picar-x/example` gehen. Nachdem Sie den Code modifiziert haben, können Sie ihn direkt ausführen, um den Effekt zu sehen.

```

from picarx import Picarx
import time

POWER = 50
SafeDistance = 40 # > 40 safe
DangerDistance = 20 # > 20 && < 40 turn around,
                  # < 20 backward

def main():
    try:
        px = Picarx()
        # px = Picarx(ultrasonic_pins=['D2','D3']) # tring, echo

        while True:
            distance = round(px.ultrasonic.read(), 2)
            print("distance: ",distance)
            if distance >= SafeDistance:
                px.set_dir_servo_angle(0)
                px.forward(POWER)
            elif distance >= DangerDistance:
                px.set_dir_servo_angle(30)
                px.forward(POWER)
                time.sleep(0.1)
            else:
                px.set_dir_servo_angle(-30)
                px.backward(POWER)
                time.sleep(0.5)

        finally:
            px.forward(0)

if __name__ == "__main__":
    main()

```

Wie funktioniert des?

- Import des Picarx-Moduls und Initialisierung von Konstanten:

Dieser Abschnitt des Codes importiert die Klasse `Picarx` aus dem Modul `picarx`, die für die Steuerung des Picarx-Roboters entscheidend ist. Konstanten wie `POWER`, `SafeDistance` und `DangerDistance` werden definiert, die später im Skript verwendet werden, um die Bewegung des Roboters basierend auf Distanzmessungen zu steuern.

```

from picarx import Picarx
import time

```

(Fortsetzung auf der nächsten Seite)

```
POWER = 50
SafeDistance = 40 # > 40 sicher
DangerDistance = 20 # > 20 && < 40 umdrehen,
# < 20 rückwärts
```

- Definition der Hauptfunktion und Auslesen des Ultraschallsensors:

Die `main` Funktion ist der Ort, an dem der Picarx-Roboter gesteuert wird. Eine Instanz von `Picarx` wird erstellt, die die Funktionen des Roboters aktiviert. Das Skript tritt in eine endlose Schleife ein, liest ständig die Entfernung vom Ultraschallsensor und verwendet diese Entfernung, um die Bewegung des Roboters zu bestimmen.

```
def main():
    try:
        px = Picarx()

        while True:
            distance = round(px.ultrasonic.read(), 2)
            # [Rest der Logik]
```

- Bewegungslogik basierend auf Distanz:

Die Bewegung des Roboters wird basierend auf der vom Ultraschallsensor gelesenen `distance` gesteuert. Wenn die `distance` größer als `SafeDistance` ist, bewegt sich der Roboter vorwärts. Wenn die Distanz zwischen `DangerDistance` und `SafeDistance` liegt, dreht er sich leicht und bewegt sich vorwärts. Wenn die `distance` kleiner als `DangerDistance` ist, fährt der Roboter rückwärts und dreht sich in die entgegengesetzte Richtung.

```
if distance >= SafeDistance:
    px.set_dir_servo_angle(0)
    px.forward(POWER)
elif distance >= DangerDistance:
    px.set_dir_servo_angle(30)
    px.forward(POWER)
    time.sleep(0.1)
else:
    px.set_dir_servo_angle(-30)
    px.backward(POWER)
    time.sleep(0.5)
```

- Sicherheit und Aufräumen mit dem `finally` Block:

Der `try...finally` Block stellt die Sicherheit sicher, indem er die Bewegung des Roboters im Falle einer Unterbrechung oder eines Fehlers stoppt. Dies ist ein entscheidender Teil, um unkontrollierbares Verhalten des Roboters zu verhindern.

```
try:
    # [Steuerungslogik]
finally:
    px.forward(0)
```

- Ausführungseinstiegspunkt:

Der standardmäßige Python-Einstiegspunkt `if __name__ == "__main__":` wird verwendet, um die Hauptfunktion auszuführen, wenn das Skript als eigenständiges Programm ausgeführt wird.

```
if __name__ == "__main__":
    main()
```

Zusammenfassend verwendet das Skript das Picarx-Modul, um einen Roboter zu steuern, wobei ein Ultraschallsensor zur Entfernungsmessung genutzt wird. Die Bewegung des Roboters wird anhand dieser Messungen angepasst und sorgt durch sorgfältige Steuerung und einen Sicherheitsmechanismus im finally-Block für einen sicheren Betrieb.

4.8 5. Linienverfolgung

Dieses Projekt wird das Graustufenmodul verwenden, um den PiCar-X entlang einer Linie vorwärtsfahren zu lassen. Verwenden Sie dunkelfarbiges Klebeband, um eine Linie so gerade wie möglich zu machen und nicht zu sehr gekrümmt. Einige Experimente könnten notwendig sein, wenn der PiCar-X entgleist.

Code ausführen

```
cd ~/picar-x/example
sudo python3 5.minecart_plus.py
```

Nachdem der Code ausgeführt wurde, wird PiCar-X entlang einer Linie vorwärtsfahren.

Code

Bemerkung: Sie können den untenstehenden Code **modifizieren/zurücksetzen/kopieren/ausführen/stoppen**. Bevor Sie das tun, müssen Sie jedoch zum Quellcodepfad wie `picar-x/example` gehen. Nachdem Sie den Code modifiziert haben, können Sie ihn direkt ausführen, um den Effekt zu sehen.

```
from picarx import Picarx
from time import sleep

px = Picarx()
# px = Picarx(grayscale_pins=['A0', 'A1', 'A2'])

# Please run ./calibration/grayscale_calibration.py to Auto calibrate grayscale values
# or manual modify reference value by follow code
# px.set_line_reference([1400, 1400, 1400])

current_state = None
px_power = 10
offset = 20
last_state = "stop"

def outHandle():
    global last_state, current_state
    if last_state == 'left':
        px.set_dir_servo_angle(-30)
        px.backward(10)
    elif last_state == 'right':
        px.set_dir_servo_angle(30)
        px.backward(10)
    while True:
        gm_val_list = px.get_grayscale_data()
```

(Fortsetzung auf der nächsten Seite)

```

    gm_state = get_status(gm_val_list)
    print("outHandle gm_val_list: %s, %s"%(gm_val_list, gm_state))
    currentSta = gm_state
    if currentSta != last_state:
        break
    sleep(0.001)

def get_status(val_list):
    _state = px.get_line_status(val_list) # [bool, bool, bool], 0 means line, 1 means_
↪background
    if _state == [0, 0, 0]:
        return 'stop'
    elif _state[1] == 1:
        return 'forward'
    elif _state[0] == 1:
        return 'right'
    elif _state[2] == 1:
        return 'left'

if __name__=='__main__':
    try:
        while True:
            gm_val_list = px.get_grayscale_data()
            gm_state = get_status(gm_val_list)
            print("gm_val_list: %s, %s"%(gm_val_list, gm_state))

            if gm_state != "stop":
                last_state = gm_state

            if gm_state == 'forward':
                px.set_dir_servo_angle(0)
                px.forward(px_power)
            elif gm_state == 'left':
                px.set_dir_servo_angle(offset)
                px.forward(px_power)
            elif gm_state == 'right':
                px.set_dir_servo_angle(-offset)
                px.forward(px_power)
            else:
                outHandle()
    finally:
        px.stop()
        print("stop and exit")
        sleep(0.1)

```

Wie funktioniert des?

Dieses Python-Skript steuert ein Picarx-Roboterauto mithilfe von Graustufensensoren zur Navigation. Hier ist eine Zusammenfassung seiner Hauptkomponenten:

- Import und Initialisierung:

Das Skript importiert die Picarx-Klasse zur Steuerung des Roboterwagens und die Sleep-Funktion aus dem Time-Modul, um Verzögerungen hinzuzufügen.

Eine Instanz von Picarx wird erstellt, und es gibt eine auskommentierte Zeile, die eine alternative Initialisierung mit spezifischen Graustufensensor-Pins zeigt.

```
from picarx import Picarx
from time import sleep

px = Picarx()
```

- Konfiguration und Globale Variablen:

current_state, px_power, offset und last_state sind globale Variablen, die verwendet werden, um die Bewegung des Autos zu verfolgen und zu steuern. px_power setzt die Motorleistung, und offset wird zur Einstellung des Lenkwinkels verwendet.

```
current_state = None
px_power = 10
offset = 20
last_state = "stop"
```

- outHandle Funktion:

Diese Funktion wird aufgerufen, wenn das Auto ein ‚Aus-der-Linie‘-Szenario behandeln muss.

Sie passt die Fahrtrichtung basierend auf last_state an und überprüft die Graustufensensorwerte, um den neuen Zustand zu bestimmen.

```
def outHandle():
    global last_state, current_state
    if last_state == 'left':
        px.set_dir_servo_angle(-30)
        px.backward(10)
    elif last_state == 'right':
        px.set_dir_servo_angle(30)
        px.backward(10)
    while True:
        gm_val_list = px.get_grayscale_data()
        gm_state = get_status(gm_val_list)
        print("outHandle gm_val_list: %s, %s"%(gm_val_list, gm_state))
        currentSta = gm_state
        if currentSta != last_state:
            break
    sleep(0.001)
```

- get_status Funktion:

Sie interpretiert die Graustufensensordaten (val_list), um den Navigationszustand des Autos zu bestimmen.

Der Zustand des Autos kann vorwärts, links, rechts oder stop sein, je nachdem, welcher Sensor die Linie erkennt.

```
def get_status(val_list):
    _state = px.get_line_status(val_list) # [bool, bool, bool], 0 bedeutet
    ↳Linie, 1 bedeutet Hintergrund
    if _state == [0, 0, 0]:
        return 'stop'
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
elif _state[1] == 1:
    return 'vorwärts'
elif _state[0] == 1:
    return 'rechts'
elif _state[2] == 1:
    return 'links'
```

- Hauptschleife:

Die `while True`-Schleife überprüft kontinuierlich die Graustufendaten und passt die Bewegung des Autos entsprechend an.

Abhängig vom `gm_state` wird der Lenkwinkel und die Bewegungsrichtung eingestellt.

```
if __name__ == '__main__':
    try:
        while True:
            gm_val_list = px.get_grayscale_data()
            gm_state = get_status(gm_val_list)
            print("gm_val_list: %s, %s"%(gm_val_list, gm_state))

            if gm_state != "stop":
                last_state = gm_state

            if gm_state == 'vorwärts':
                px.set_dir_servo_angle(0)
                px.forward(px_power)
            elif gm_state == 'links':
                px.set_dir_servo_angle(offset)
                px.forward(px_power)
            elif gm_state == 'rechts':
                px.set_dir_servo_angle(-offset)
                px.forward(px_power)
            else:
                outHandle()
```

- Sicherheit und Aufräumen:

Der `try...finally`-Block stellt sicher, dass das Auto stoppt, wenn das Skript unterbrochen oder beendet wird.

```
finally:
    px.stop()
    print("stop and exit")
    sleep(0.1)
```

Zusammenfassend verwendet das Skript Graustufensensoren zur Navigation des Picarx-Roboterwagens. Es liest kontinuierlich die Sensordaten, um die Richtung zu bestimmen und passt die Bewegung und Lenkung des Autos entsprechend an. Die `outHandle`-Funktion bietet zusätzliche Logik für Situationen, in denen das Auto seinen Weg deutlich anpassen muss.

4.9 6. Klippenerkennung

Geben wir PiCar-X ein wenig Selbstschutz-Bewusstsein und lassen es lernen, sein eigenes Graustufenmodul zu nutzen, um nicht über die Klippe zu stürzen.

In diesem Beispiel wird das Auto im Ruhezustand sein. Wenn Sie es an eine Klippe schieben, wird es dringend geweckt, fährt dann zurück und sagt „Gefahr“.

Code ausführen

```
cd ~/picar-x/example
sudo python3 6.cliff_detection.py
```

Code

Bemerkung: Sie können den untenstehenden Code **modifizieren/zurücksetzen/kopieren/ausführen/stoppen**. Bevor Sie das tun, müssen Sie jedoch zum Quellcodepfad wie `picar-x/example` gehen. Nachdem Sie den Code modifiziert haben, können Sie ihn direkt ausführen, um den Effekt zu sehen.

```
from picarx import Picarx
from time import sleep
from robot_hat import TTS

tts = TTS()
tts.lang("en-US")

px = Picarx()
# px = Picarx(grayscale_pins=['A0', 'A1', 'A2'])
# manual modify reference value
px.set_cliff_reference([200, 200, 200])

current_state = None
px_power = 10
offset = 20
last_state = "safe"

if __name__ == '__main__':
    try:
        while True:
            gm_val_list = px.get_grayscale_data()
            gm_state = px.get_cliff_status(gm_val_list)
            # print("cliff status is: %s"%gm_state)

            if gm_state is False:
                state = "safe"
                px.stop()
            else:
                state = "danger"
                px.backward(80)
                if last_state == "safe":
                    tts.say("danger")
                    sleep(0.1)
```

(Fortsetzung auf der nächsten Seite)

```

        last_state = state

    finally:
        px.stop()
        print("stop and exit")
        sleep(0.1)

```

Wie funktioniert des?

Die Funktion zur Erkennung der Klippe sieht so aus:

- `get_grayscale_data()`: Diese Methode gibt direkt die Messwerte der drei Sensoren von rechts nach links aus. Je heller die Fläche, desto größer der erhaltene Wert.
- `get_cliff_status(gm_val_list)`: Diese Methode vergleicht die Messwerte der drei Sensoren und gibt ein Ergebnis aus. Wenn das Ergebnis wahr ist, wird erkannt, dass sich eine Klippe vor dem Auto befindet.

4.10 7. Computervision

Dieses Projekt wird offiziell in das Feld der Computervision eintreten!

Code ausführen

```

cd ~/picar-x/example
sudo python3 7.display.py

```

Das Bild betrachten

Nachdem der Code ausgeführt wurde, wird im Terminal folgende Aufforderung angezeigt:

```

No desktop !
* Serving Flask app "vilib.vilib" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:9000/ (Press CTRL+C to quit)

```

Dann können Sie `http://<Ihre IP>:9000/mjpg` im Browser eingeben, um den Videobildschirm zu betrachten, wie z. B.: `https://192.168.18.113:9000/mjpg`



Nachdem das Programm ausgeführt wurde, sehen Sie folgende Informationen am Ende:

- Taste drücken, um die Funktion aufzurufen!
- q: Foto machen
- 1: Farberkennung : rot
- 2: Farberkennung : orange

- 3: Farberkennung : gelb
- 4: Farberkennung : grün
- 5: Farberkennung : blau
- 6: Farberkennung : lila
- 0: Farberkennung ausschalten
- r: QR-Code scannen
- f: Gesichtserkennung EIN/AUS schalten
- s: Informationen über erkannte Objekte anzeigen

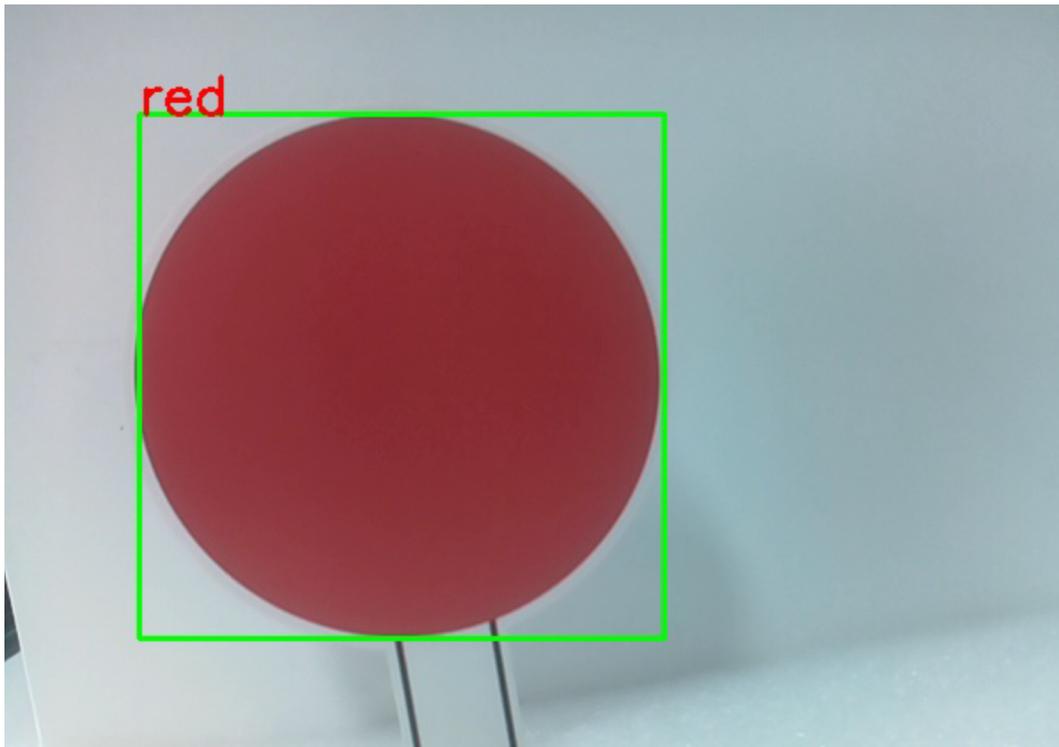
Bitte folgen Sie den Aufforderungen, um die entsprechenden Funktionen zu aktivieren.

- **Foto machen**

Geben Sie im Terminal `q` ein und drücken Sie Enter. Das von der Kamera aktuell gesehene Bild wird gespeichert (wenn die Farberkennungsfunktion eingeschaltet ist, wird auch die Markierungsbox im gespeicherten Bild angezeigt). Diese Fotos können Sie im Verzeichnis `/home/{username}/Pictures/` des Raspberry Pi sehen. Sie können Tools wie *Filezilla Software* verwenden, um Fotos auf Ihren PC zu übertragen.

- **Farberkennung**

Durch Eingabe einer Zahl zwischen 1~6 wird eine der Farben „rot, orange, gelb, grün, blau, lila“ erkannt. Geben Sie `0` ein, um die Farberkennung auszuschalten.



Bemerkung: Sie können die PDF-Farbkarten für die Farberkennung herunterladen und ausdrucken.

- **Gesichtserkennung**

Geben Sie `f` ein, um die Gesichtserkennung einzuschalten.



- **QR-Code-Erkennung**

Geben Sie `r` ein, um die QR-Code-Erkennung zu öffnen. Vor der Erkennung des QR-Codes können keine anderen Operationen durchgeführt werden. Die Dekodierungsinformationen des QR-Codes werden im Terminal gedruckt.



- **Informationen anzeigen**

Durch Eingabe von `s` werden die Informationen des Gesichtserkennungs- (und Farberkennungs-) Ziels im Terminal gedruckt. Einschließlich der Mittelkoordinaten (X, Y) und Größe (Breite, Höhe) des gemessenen Objekts.

Code

```

from pydoc import text
from vilib import Vilib
from time import sleep, time, strftime, localtime
import threading
import readchar
import os

flag_face = False
flag_color = False
qr_code_flag = False

manual = '''
Input key to call the function!
  q: Take photo
  1: Color detect : red
  2: Color detect : orange
  3: Color detect : yellow
  4: Color detect : green
  5: Color detect : blue
  6: Color detect : purple
  0: Switch off Color detect
  r: Scan the QR code

```

(Fortsetzung auf der nächsten Seite)

```
f: Switch ON/OFF face detect
s: Display detected object information
...

color_list = ['close', 'red', 'orange', 'yellow',
              'green', 'blue', 'purple',
              ]

def face_detect(flag):
    print("Face Detect:" + str(flag))
    Vilib.face_detect_switch(flag)

def qrcode_detect():
    global qr_code_flag
    if qr_code_flag == True:
        Vilib.qrcode_detect_switch(True)
        print("Waiting for QR code")

    text = None
    while True:
        temp = Vilib.detect_obj_parameter['qr_data']
        if temp != "None" and temp != text:
            text = temp
            print('QR code:%s'%text)
            if qr_code_flag == False:
                break
            sleep(0.5)
        Vilib.qrcode_detect_switch(False)

def take_photo():
    _time = strftime('%Y-%m-%d-%H-%M-%S', localtime(time()))
    name = 'photo_%s'%_time
    username = os.getlogin()

    path = f"/home/{username}/Pictures/"
    Vilib.take_photo(name, path)
    print('photo save as %s%s.jpg'%(path,name))

def object_show():
    global flag_color, flag_face

    if flag_color is True:
        if Vilib.detect_obj_parameter['color_n'] == 0:
            print('Color Detect: None')
        else:
            color_coordinate = (Vilib.detect_obj_parameter['color_x'],Vilib.detect_obj_
↪parameter['color_y'])
            color_size = (Vilib.detect_obj_parameter['color_w'],Vilib.detect_obj_
↪parameter['color_h'])
```

(Fortsetzung der vorherigen Seite)

```

        print("[Color Detect] ", "Coordinate:", color_coodinate, "Size", color_size)

    if flag_face is True:
        if Vilib.detect_obj_parameter['human_n'] == 0:
            print('Face Detect: None')
        else:
            human_coodinate = (Vilib.detect_obj_parameter['human_x'], Vilib.detect_obj_
↪parameter['human_y'])
            human_size = (Vilib.detect_obj_parameter['human_w'], Vilib.detect_obj_
↪parameter['human_h'])
            print("[Face Detect] ", "Coordinate:", human_coodinate, "Size", human_size)

def main():
    global flag_face, flag_color, qr_code_flag
    qrcode_thread = None

    Vilib.camera_start(vflip=False, hflip=False)
    Vilib.display(local=True, web=True)
    print(manual)

    while True:
        # readkey
        key = readchar.readkey()
        key = key.lower()
        # take photo
        if key == 'q':
            take_photo()
        # color detect
        elif key != '' and key in ('0123456'): # " in ('0123') -> True
            index = int(key)
            if index == 0:
                flag_color = False
                Vilib.color_detect('close')
            else:
                flag_color = True
                Vilib.color_detect(color_list[index]) # color_detect(color:str -> color_
↪name/close)
            print('Color detect : %s'%color_list[index])
        # face detection
        elif key == "f":
            flag_face = not flag_face
            face_detect(flag_face)
        # qrcode detection
        elif key == "r":
            qr_code_flag = not qr_code_flag
            if qr_code_flag == True:
                if qrcode_thread == None or not qrcode_thread.is_alive():
                    qrcode_thread = threading.Thread(target=qrcode_detect)
                    qrcode_thread.setDaemon(True)
                    qrcode_thread.start()
            else:

```

(Fortsetzung auf der nächsten Seite)

```

        if qrcode_thread != None and qrcode_thread.is_alive():
            # wait for thread to end
            qrcode_thread.join()
            print('QRcode Detect: close')
        # show detected object information
        elif key == "s":
            object_show()

        sleep(0.5)

if __name__ == "__main__":
    main()

```

Wie funktioniert des?

Das Erste, worauf Sie hier achten müssen, ist die folgende Funktion. Diese beiden Funktionen ermöglichen es Ihnen, die Kamera zu starten.

```

Vilib.camera_start()
Vilib.display()

```

Funktionen, die mit „Objekterkennung“ zusammenhängen:

- `Vilib.face_detect_switch(True)` : Gesichtserkennung EIN/AUS schalten
- `Vilib.color_detect(color)` : Für die Farberkennung, es kann nur eine Farberkennung gleichzeitig durchgeführt werden. Die einführbaren Parameter sind: "rot", "orange", "gelb", "grün", "blau", "lila"
- `Vilib.color_detect_switch(False)` : Farberkennung ausschalten
- `Vilib.qrcode_detect_switch(False)` : QR-Code-Erkennung EIN/AUS schalten, gibt die dekodierten Daten des QR-Codes zurück.
- `Vilib.gesture_detect_switch(False)` : Gestenerkennung EIN/AUS schalten
- `Vilib.traffic_sign_detect_switch(False)` : Verkehrszeichenerkennung EIN/AUS schalten

Die von dem Ziel erkannten Informationen werden im Wörterbuch `detect_obj_parameter = Manager().dict()` gespeichert.

Im Hauptprogramm können Sie es so verwenden:

```

Vilib.detect_obj_parameter['color_x']

```

Die Schlüssel des Wörterbuchs und deren Verwendungen werden in der folgenden Liste gezeigt:

- `color_x`: der x-Wert der Mittelkoordinate des erkannten Farbblocks, der Bereich ist 0~320
- `color_y`: der y-Wert der Mittelkoordinate des erkannten Farbblocks, der Bereich ist 0~240
- `color_w`: die Breite des erkannten Farbblocks, der Bereich ist 0~320
- `color_h`: die Höhe des erkannten Farbblocks, der Bereich ist 0~240
- `color_n`: die Anzahl der erkannten Farbflecken
- `human_x`: der x-Wert der Mittelkoordinate des erkannten menschlichen Gesichts, der Bereich ist 0~320
- `human_y`: der y-Wert der Mittelkoordinate des erkannten Gesichts, der Bereich ist 0~240
- `human_w`: die Breite des erkannten menschlichen Gesichts, der Bereich ist 0~320

- `human_h`: die Höhe des erkannten Gesichts, der Bereich ist 0~240
- `human_n`: die Anzahl der erkannten Gesichter
- `traffic_sign_x`: der Mittelkoordinaten-x-Wert des erkannten Verkehrszeichens, der Bereich ist 0~320
- `traffic_sign_y`: der Mittelkoordinaten-y-Wert des erkannten Verkehrszeichens, der Bereich ist 0~240
- `traffic_sign_w`: die Breite des erkannten Verkehrszeichens, der Bereich ist 0~320
- `traffic_sign_h`: die Höhe des erkannten Verkehrszeichens, der Bereich ist 0~240
- `traffic_sign_t`: der Inhalt des erkannten Verkehrszeichens, die Werteliste ist `['stop', 'right', 'left', 'forward']`
- `gesture_x`: Der Mittelkoordinaten-x-Wert der erkannten Geste, der Bereich ist 0~320
- `gesture_y`: Der Mittelkoordinaten-y-Wert der erkannten Geste, der Bereich ist 0~240
- `gesture_w`: Die Breite der erkannten Geste, der Bereich ist 0~320
- `gesture_h`: Die Höhe der erkannten Geste, der Bereich ist 0~240
- `gesture_t`: Der Inhalt der erkannten Geste, die Werteliste ist `['paper', 'scissor', 'rock']`
- `qr_date`: der Inhalt des erkannten QR-Codes
- `qr_x`: der Mittelkoordinaten-x-Wert des zu erkennenden QR-Codes, der Bereich ist 0~320
- `qr_y`: der Mittelkoordinaten-y-Wert des zu erkennenden QR-Codes, der Bereich ist 0~240
- `qr_w`: die Breite des zu erkennenden QR-Codes, der Bereich ist 0~320
- `qr_h`: die Höhe des zu erkennenden QR-Codes, der Bereich ist 0~320

4.11 8. Blick auf Dich

Dieses Projekt basiert ebenfalls auf dem 7. *Computervision*-Projekt, ergänzt um Algorithmen zur Gesichtserkennung.

Wenn Sie vor der Kamera erscheinen, wird sie Ihr Gesicht erkennen und ihr Gimbal so anpassen, dass Ihr Gesicht im Zentrum des Bildes bleibt.

Sie können den Bildschirm unter `http://<Ihre IP>:9000/mjpg` ansehen.

Code ausführen

```
cd ~/picar-x/example
sudo python3 8.stare_at_you.py
```

Wenn der Code ausgeführt wird, wird die Kamera des Autos immer auf Ihr Gesicht starren.

Code

```
from picarx import Picarx
from time import sleep
from vilib import Vilib

px = Picarx()

def clamp_number(num, a, b):
    return max(min(num, max(a, b)), min(a, b))

def main():
```

(Fortsetzung auf der nächsten Seite)

```

Vilib.camera_start()
Vilib.display()
Vilib.face_detect_switch(True)
x_angle =0
y_angle =0
while True:
    if Vilib.detect_obj_parameter['human_n']!=0:
        coordinate_x = Vilib.detect_obj_parameter['human_x']
        coordinate_y = Vilib.detect_obj_parameter['human_y']

        # change the pan-tilt angle for track the object
        x_angle +=(coordinate_x*10/640)-5
        x_angle = clamp_number(x_angle,-35,35)
        px.set_cam_pan_angle(x_angle)

        y_angle -=(coordinate_y*10/480)-5
        y_angle = clamp_number(y_angle,-35,35)
        px.set_cam_tilt_angle(y_angle)

        sleep(0.05)

    else :
        pass
        sleep(0.05)

if __name__ == "__main__":
    try:
        main()

    finally:
        px.stop()
        print("stop and exit")
        sleep(0.1)

```

Wie funktioniert des?

Diese Codezeilen in while True lassen die Kamera dem Gesicht folgen.

```

while True:
    if Vilib.detect_obj_parameter['human_n'] != 0:
        coordinate_x = Vilib.detect_obj_parameter['human_x']
        coordinate_y = Vilib.detect_obj_parameter['human_y']

        # Ändere den Schwenk-Neige-Winkel, um dem Objekt zu folgen
        x_angle += (coordinate_x*10/640) - 5
        x_angle = clamp_number(x_angle, -35, 35)
        px.set_cam_pan_angle(x_angle)

        y_angle -= (coordinate_y*10/480) - 5
        y_angle = clamp_number(y_angle, -35, 35)
        px.set_cam_tilt_angle(y_angle)

```

1. Überprüfe, ob ein menschliches Gesicht erkannt wurde

```
Vilib.detect_obj_parameter['human_n'] != 0
```

2. Wenn ein menschliches Gesicht erkannt wird, erhalte die Koordinaten (`coordinate_x` und `coordinate_y`) des erkannten Gesichts.
3. Berechne neue Schwenk- und Neigewinkel (`x_angle` und `y_angle`) basierend auf der Position des erkannten Gesichts und passe sie an, um dem Gesicht zu folgen.
4. Begrenze die Schwenk- und Neigewinkel innerhalb des angegebenen Bereichs mithilfe der `clamp_number`-Funktion.
5. Stelle die Schwenk- und Neigewinkel der Kamera mit `px.set_cam_pan_angle()` und `px.set_cam_tilt_angle()` ein.

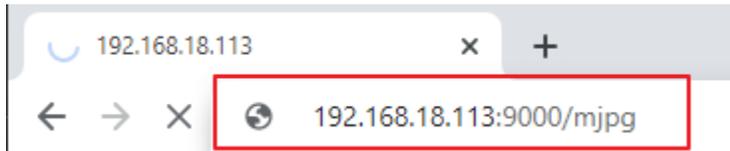
4.12 9. Videoaufnahme

Diese Anleitung zeigt Ihnen, wie Sie die Aufnahmefunktion nutzen können.

Code ausführen

```
cd ~/picar-x/example
sudo python3 9.record_video.py
```

Nachdem der Code ausgeführt wurde, können Sie `http://<Ihre IP>:9000/mjpg` in Ihren Browser eingeben, um das Video zu sehen. Zum Beispiel: `http://192.168.18.113:9000/mjpg`



Die Aufnahme kann durch Drücken der Tasten auf der Tastatur gestartet oder gestoppt werden.

- Drücken Sie `q`, um die Aufnahme zu starten oder zu pausieren/fortzusetzen, `e`, um die Aufnahme zu stoppen oder zu speichern.
- Um das Programm zu beenden, drücken Sie `ctrl+c`.

Code

```
from time import sleep, strftime, localtime
from vilib import Vilib
import readchar
import os

manual = '''
Press keys on keyboard to control recording:
    Q: record/pause/continue
    E: stop
    Ctrl + C: Quit
'''

def print_overwrite(msg, end='', flush=True):
    print('\r\033[2K', end='', flush=True)
    print(msg, end=end, flush=True)
```

(Fortsetzung auf der nächsten Seite)

```
def main():
    rec_flag = 'stop' # start,pause,stop
    vname = None
    username = os.getlogin()

    Vilib.rec_video_set["path"] = f"/home/{username}/Videos/" # set path

    Vilib.camera_start(vflip=False,hflip=False)
    Vilib.display(local=True,web=True)
    sleep(0.8) # wait for startup

    print(manual)
    while True:
        # read keyboard
        key = readchar.readkey()
        key = key.lower()
        # start,pause
        if key == 'q':
            key = None
            if rec_flag == 'stop':
                rec_flag = 'start'
                # set name
                vname = strftime("%Y-%m-%d-%H.%M.%S", localtime())
                Vilib.rec_video_set["name"] = vname
                # start record
                Vilib.rec_video_run()
                Vilib.rec_video_start()
                print_overwrite('rec start ...')
            elif rec_flag == 'start':
                rec_flag = 'pause'
                Vilib.rec_video_pause()
                print_overwrite('pause')
            elif rec_flag == 'pause':
                rec_flag = 'start'
                Vilib.rec_video_start()
                print_overwrite('continue')

        # stop
        elif key == 'e' and rec_flag != 'stop':
            key = None
            rec_flag = 'stop'
            Vilib.rec_video_stop()
            print_overwrite("The video saved as %s%s.avi"%(Vilib.rec_video_set["path"],
↪vname),end='\n')
        # quit
        elif key == readchar.key.CTRL_C:
            Vilib.camera_close()
            print('\nquit')
            break

    sleep(0.1)
```

(Fortsetzung der vorherigen Seite)

```
if __name__ == "__main__":
    main()
```

Wie funktioniert das?

Die Funktionen zur Videoaufnahme umfassen Folgendes:

- `Vilib.rec_video_run(video_name)`: Startet den Thread für die Videoaufnahme. `video_name` ist der Name der Videodatei, es sollte ein String sein.
- `Vilib.rec_video_start()`: Startet oder setzt die Videoaufnahme fort.
- `Vilib.rec_video_pause()`: Pausiert die Aufnahme.
- `Vilib.rec_video_stop()`: Stoppt die Aufnahme.

`Vilib.rec_video_set["path"] = f"/home/{username}/Videos/"` legt den Speicherort der Videodateien fest.

4.13 10. Stierkampf

Mache PiCar-X zu einem wütenden Stier! Verwende seine Kamera, um das rote Tuch zu verfolgen und darauf zuzusehen!

Code ausführen

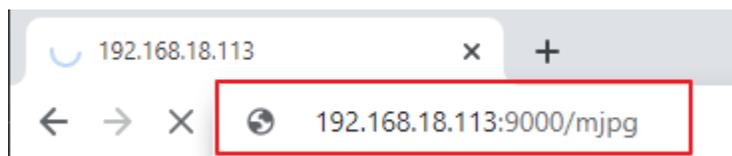
```
cd ~/picar-x/example
sudo python3 10.bull_fight.py
```

Das Bild ansehen

Nachdem der Code ausgeführt wurde, zeigt das Terminal folgende Aufforderung an:

```
No desktop !
* Serving Flask app "vilib.vilib" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:9000/ (Press CTRL+C to quit)
```

Anschließend können Sie `http://<Ihre IP>:9000/mjpg` im Browser eingeben, um das Videosignal zu sehen, z.B.: `https://192.168.18.113:9000/mjpg`

**Code**

Bemerkung: Sie können den untenstehenden Code **modifizieren/zurücksetzen/kopieren/ausführen/stoppen**. Bevor Sie das tun, sollten Sie zum Quellcodepfad wie `picar-x\beispiele` gehen. Nachdem Sie den Code modifiziert haben, können Sie ihn direkt ausführen, um die Wirkung zu sehen.

```
from picarx import Picarx
from time import sleep
from vilib import Vilib

px = Picarx()

def clamp_number(num,a,b):
    return max(min(num, max(a, b)), min(a, b))

def main():
    Vilib.camera_start()
    Vilib.display()
    Vilib.color_detect("red")
    speed = 50
    dir_angle=0
    x_angle =0
    y_angle =0
    while True:
        if Vilib.detect_obj_parameter['color_n']!=0:
            coordinate_x = Vilib.detect_obj_parameter['color_x']
            coordinate_y = Vilib.detect_obj_parameter['color_y']

            # change the pan-tilt angle for track the object
            x_angle +=(coordinate_x*10/640)-5
            x_angle = clamp_number(x_angle, -35,35)
            px.set_cam_pan_angle(x_angle)

            y_angle --(coordinate_y*10/480)-5
            y_angle = clamp_number(y_angle, -35,35)
            px.set_cam_tilt_angle(y_angle)

            # move
            # The movement direction will change slower than the pan/tilt direction
            # change to avoid confusion when the picture changes at high speed.
            if dir_angle > x_angle:
                dir_angle -= 1
            elif dir_angle < x_angle:
                dir_angle += 1
            px.set_dir_servo_angle(x_angle)
            px.forward(speed)
            sleep(0.05)

        else :
            px.forward(0)
            sleep(0.05)

if __name__ == "__main__":
    try:
        main()

    finally:
        px.stop()
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
print("stop and exit")
sleep(0.1)
```

Wie funktioniert des?

Sie sollten auf die folgenden drei Teile dieses Beispiels achten:

1. Die Hauptfunktion definieren:

- Starten Sie die Kamera mit `Vilib.camera_start()`.
- Zeigen Sie das Kamera-Feed mit `Vilib.display()` an.
- Aktivieren Sie die Farberkennung und legen Sie die Zielfarbe als „rot“ fest mit `Vilib.color_detect("red")`.
- Initialisieren Sie Variablen: `speed` für die Bewegungsgeschwindigkeit des Autos, `dir_angle` für den Richtungswinkel der Bewegung des Autos, `x_angle` für den Schwenkwinkel der Kamera und `y_angle` für den Neigungswinkel der Kamera.

2. Treten Sie in eine kontinuierliche Schleife (`while True`), um das rotfarbene Objekt zu verfolgen:

- Überprüfen Sie, ob ein rotfarbenedes Objekt erkannt wurde (`Vilib.detect_obj_parameter['color_n'] != 0`).
- Wenn ein rotfarbenedes Objekt erkannt wird, erhalten Sie dessen Koordinaten (`coordinate_x` und `coordinate_y`).
- Berechnen Sie neue Schwenk- und Neigungswinkel (`x_angle` und `y_angle`) basierend auf der Position des erkannten Objekts und passen Sie diese an, um das Objekt zu verfolgen.
- Begrenzen Sie die Schwenk- und Neigungswinkel innerhalb des festgelegten Bereichs mit der Funktion `clamp_number`.
- Stellen Sie die Schwenk- und Neigungswinkel der Kamera mit `px.set_cam_pan_angle()` und `px.set_cam_tilt_angle()` ein, um das Objekt im Blick zu behalten.

3. Steuern Sie die Bewegung des Autos basierend auf dem Unterschied zwischen `dir_angle` und `x_angle`:

- Wenn `dir_angle` größer als `x_angle` ist, verringern Sie `dir_angle` um 1, um den Richtungswinkel allmählich zu ändern.
- Wenn `dir_angle` kleiner als `x_angle` ist, erhöhen Sie `dir_angle` um 1.
- Stellen Sie den Lenkservo-Winkel mit `px.set_dir_servo_angle()` ein, um die Räder des Autos entsprechend zu steuern.
- Bewegen Sie das Auto vorwärts mit der festgelegten Geschwindigkeit mit `px.forward(speed)`.

4.14 11. Video-Auto

Dieses Programm ermöglicht eine First-Person-Ansicht vom PiCar-X! Verwenden Sie die WSAD-Tasten der Tastatur, um die Bewegungsrichtung zu steuern, sowie O und P, um die Geschwindigkeit anzupassen.

Code ausführen

```
cd ~/picar-x/beispiel
sudo python3 11.video_auto.py
```

Sobald der Code läuft, können Sie sehen, was PiCar-X filmt und es steuern, indem Sie die folgenden Tasten drücken.

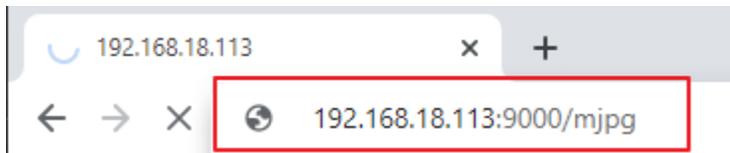
- O: Geschwindigkeit erhöhen
- P: Geschwindigkeit verringern
- W: vorwärts
- S: rückwärts
- A: links abbiegen
- D: rechts abbiegen
- F: anhalten
- T: Foto aufnehmen
- Strg+C: beenden

Das Bild ansehen

Nachdem der Code ausgeführt wurde, zeigt das Terminal folgende Aufforderung an:

```
No desktop !
* Serving Flask app "vilib.vilib" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:9000/ (Press CTRL+C to quit)
```

Anschließend können Sie `http://<Ihre IP>:9000/mjpg` im Browser eingeben, um das Videosignal zu sehen, z.B.:
`https://192.168.18.113:9000/mjpg`



Code

```
#!/usr/bin/env python3

from robot_hat.utils import reset_mcu
from picarx import Picarx
from vilib import Vilib
from time import sleep, time, strftime, localtime
import readchar

import os
user = os.getlogin()
user_home = os.path.expanduser(f'~{user}')

reset_mcu()
sleep(0.2)

manual = '''
Press key to call the function(non-case sensitive):
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

O: speed up
P: speed down
W: forward
S: backward
A: turn left
D: turn right
F: stop
T: take photo

... ctrl + c: Press twice to exit the program
...

px = Picarx()

def take_photo():
    _time = strftime('%Y-%m-%d-%H-%M-%S', localtime(time()))
    name = 'photo_%s' % _time
    path = f"{user_home}/Pictures/picar-x/"
    Vilib.take_photo(name, path)
    print('\nphoto save as %s%s.jpg' % (path, name))

def move(operate:str, speed):

    if operate == 'stop':
        px.stop()
    else:
        if operate == 'forward':
            px.set_dir_servo_angle(0)
            px.forward(speed)
        elif operate == 'backward':
            px.set_dir_servo_angle(0)
            px.backward(speed)
        elif operate == 'turn left':
            px.set_dir_servo_angle(-30)
            px.forward(speed)
        elif operate == 'turn right':
            px.set_dir_servo_angle(30)
            px.forward(speed)

def main():
    speed = 0
    status = 'stop'

    Vilib.camera_start(vflip=False, hflip=False)
    Vilib.display(local=True, web=True)
    sleep(2) # wait for startup
    print(manual)

```

(Fortsetzung auf der nächsten Seite)

```

while True:
    print("\rstatus: %s , speed: %s" % (status, speed), end='', flush=True)
    # readkey
    key = readchar.readkey().lower()
    # operation
    if key in ('wsadfop'):
        # throttle
        if key == 'o':
            if speed <=90:
                speed += 10
        elif key == 'p':
            if speed >=10:
                speed -= 10
            if speed == 0:
                status = 'stop'
        # direction
        elif key in ('wsad'):
            if speed == 0:
                speed = 10
            if key == 'w':
                # Speed limit when reversing, avoid instantaneous current too large
                if status != 'forward' and speed > 60:
                    speed = 60
                status = 'forward'
            elif key == 'a':
                status = 'turn left'
            elif key == 's':
                if status != 'backward' and speed > 60: # Speed limit when reversing
                    speed = 60
                status = 'backward'
            elif key == 'd':
                status = 'turn right'
        # stop
        elif key == 'f':
            status = 'stop'
        # move
        move(status, speed)
    # take photo
    elif key == 't':
        take_photo()
    # quit
    elif key == readchar.key.CTRL_C:
        print('\nquit ...')
        px.stop()
        Vilib.camera_close()
        break

    sleep(0.1)

if __name__ == "__main__":
    try:

```

(Fortsetzung der vorherigen Seite)

```

    main()
except Exception as e:
    print("error:%s"%e)
finally:
    px.stop()
    Vilib.camera_close()

```

4.15 12. Schatzsuche

Richten Sie ein Labyrinth in Ihrem Zimmer ein und platzieren Sie sechs verschiedenfarbige Karten in sechs Ecken. Steuern Sie dann PiCar-X, um diese Farbkarten nacheinander zu suchen!

Bemerkung: Sie können die PDF-Farbkarten für die Farberkennung herunterladen und ausdrucken.

Code ausführen

```

cd ~/picar-x/beispiel
sudo python3 12.schatzsuche.py

```

Das Bild ansehen

Nachdem der Code ausgeführt wurde, zeigt das Terminal folgende Aufforderung an:

```

No desktop !
* Serving Flask app "vilib.vilib" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:9000/ (Press CTRL+C to quit)

```

Anschließend können Sie `http://<Ihre IP>:9000/mjpg` im Browser eingeben, um das Videosignal zu sehen, z.B.: `http://192.168.18.113:9000/mjpg`



Code

```

from picarx import Picarx
from time import sleep
from robot_hat import Music,TTS
from vilib import Vilib
import readchar
import random
import threading

px = Picarx()

```

(Fortsetzung auf der nächsten Seite)

```
music = Music()
tts = TTS()

manual = '''
Press keys on keyboard to control Picar-X!
    w: Forward
    a: Turn left
    s: Backward
    d: Turn right
    space: Say the target again
    ctrl+c: Quit
'''

color = "red"
color_list=["red","orange","yellow","green","blue","purple"]

def renew_color_detect():
    global color
    color = random.choice(color_list)
    Vilib.color_detect(color)
    tts.say("Look for " + color)

key = None
lock = threading.Lock()
def key_scan_thread():
    global key
    while True:
        key_temp = readchar.readkey()
        print('\r',end='')
        with lock:
            key = key_temp.lower()
            if key == readchar.key.SPACE:
                key = 'space'
            elif key == readchar.key.CTRL_C:
                key = 'quit'
            break
        sleep(0.01)

def car_move(key):
    if 'w' == key:
        px.set_dir_servo_angle(0)
        px.forward(80)
    elif 's' == key:
        px.set_dir_servo_angle(0)
        px.backward(80)
    elif 'a' == key:
        px.set_dir_servo_angle(-30)
        px.forward(80)
    elif 'd' == key:
        px.set_dir_servo_angle(30)
        px.forward(80)
```

(Fortsetzung der vorherigen Seite)

```
def main():
    global key
    Vilib.camera_start(vflip=False,hflip=False)
    Vilib.display(local=False,web=True)
    sleep(0.8)
    print(manual)

    sleep(1)
    _key_t = threading.Thread(target=key_scan_thread)
    _key_t.setDaemon(True)
    _key_t.start()

    tts.say("game start")
    sleep(0.05)
    renew_color_detect()
    while True:

        if Vilib.detect_obj_parameter['color_n']!=0 and Vilib.detect_obj_parameter[
↪ 'color_w']>100:
            tts.say("will done")
            sleep(0.05)
            renew_color_detect()

        with lock:
            if key != None and key in ('wsad'):
                car_move(key)
                sleep(0.5)
                px.stop()
                key = None
            elif key == 'space':
                tts.say("Look for " + color)
                key = None
            elif key == 'quit':
                _key_t.join()
                print("\n\rQuit")
                break

        sleep(0.05)

if __name__ == "__main__":
    try:
        main()
    except KeyboardInterrupt:
        pass
    except Exception as e:
        print(f"ERROR: {e}")
    finally:
        Vilib.camera_close()
        px.stop()
        sleep(.2)
```

Wie funktioniert das?

Um die grundlegende Logik dieses Codes zu verstehen, sollten Sie sich auf die folgenden Schlüsselteile konzentrieren:

1. **Initialisierung und Importe:** Importanweisungen am Anfang des Codes, um die verwendeten Bibliotheken zu verstehen.
2. **Globale Variablen:** Definitionen von globalen Variablen, wie `color` und `key`, die im gesamten Code verwendet werden, um die Ziel-Farbe zu verfolgen und Tastatureingaben zu erfassen.
3. `renew_color_detect()` : Diese Funktion wählt eine zufällige Farbe aus einer Liste aus und setzt sie als Ziel-Farbe für die Erkennung. Sie verwendet auch Text-to-Speech, um die ausgewählte Farbe anzusagen.
4. `key_scan_thread()` : Diese Funktion läuft in einem separaten Thread und scannt kontinuierlich nach Tastatureingaben, wobei sie die Variable `key` mit der gedrückten Taste aktualisiert. Sie verwendet ein Lock für threadsicheren Zugriff.
5. `car_move(key)` : Diese Funktion steuert die Bewegung des PiCar-X basierend auf der Tastatureingabe (`key`). Sie legt die Richtung und Geschwindigkeit der Bewegung des Roboters fest.
6. `main()` : Die Hauptfunktion, die die gesamte Logik des Codes koordiniert. Sie macht Folgendes:
 - Initialisiert die Kamera und startet die Anzeige des Kamerabilds.
 - Erstellt einen separaten Thread, um Tastatureingaben zu scannen.
 - Kündigt den Spielstart mit Text-to-Speech an.
 - Tritt in eine kontinuierliche Schleife ein, um:
 - Nach erkannten farbigen Objekten zu suchen und Aktionen auszulösen, wenn ein gültiges Objekt erkannt wird.
 - Tastatureingaben zu verarbeiten, um den Roboter zu steuern und mit dem Spiel zu interagieren.
 - Behandelt das Beenden des Spiels und Ausnahmen wie `KeyboardInterrupt`.
 - Stellt sicher, dass die Kamera geschlossen und der PiCar-X beim Beenden angehalten wird.

Indem Sie diese Schlüsselteile des Codes verstehen, können Sie die grundlegende Logik erfassen, wie der PiCar-X-Roboter auf Tastatureingaben reagiert und Objekte einer bestimmten Farbe mit der Kamera und den Audioausgabefähigkeiten erkennt und interagiert.

4.16 13. Durch die App gesteuert

Der SunFounder-Controller wird verwendet, um auf Raspberry Pi/Pico basierende Roboter zu steuern.

Die App integriert Schaltflächen, Schalter, Joysticks, D-Pads, Schieberegler und Gashebel-Widgets; digitale Anzeigen, Ultraschallradar, Graustufenerkennung und Geschwindigkeitsmesser-Eingabewidgets.

Es gibt 17 Bereiche A-Q, in denen Sie verschiedene Widgets platzieren können, um Ihren eigenen Controller anzupassen.

Zusätzlich bietet diese Anwendung einen Live-Videostreaming-Dienst.

Lassen Sie uns einen PiCar-X-Controller mit dieser App anpassen.

Wie geht das?

1. Installieren Sie das Modul `sunfounder-controller`.

Die Module `robot-hat`, `vilib` und `picar-x` müssen zuerst installiert werden, Details siehe: [Installiere Alle Module\(Wichtig\)](#).

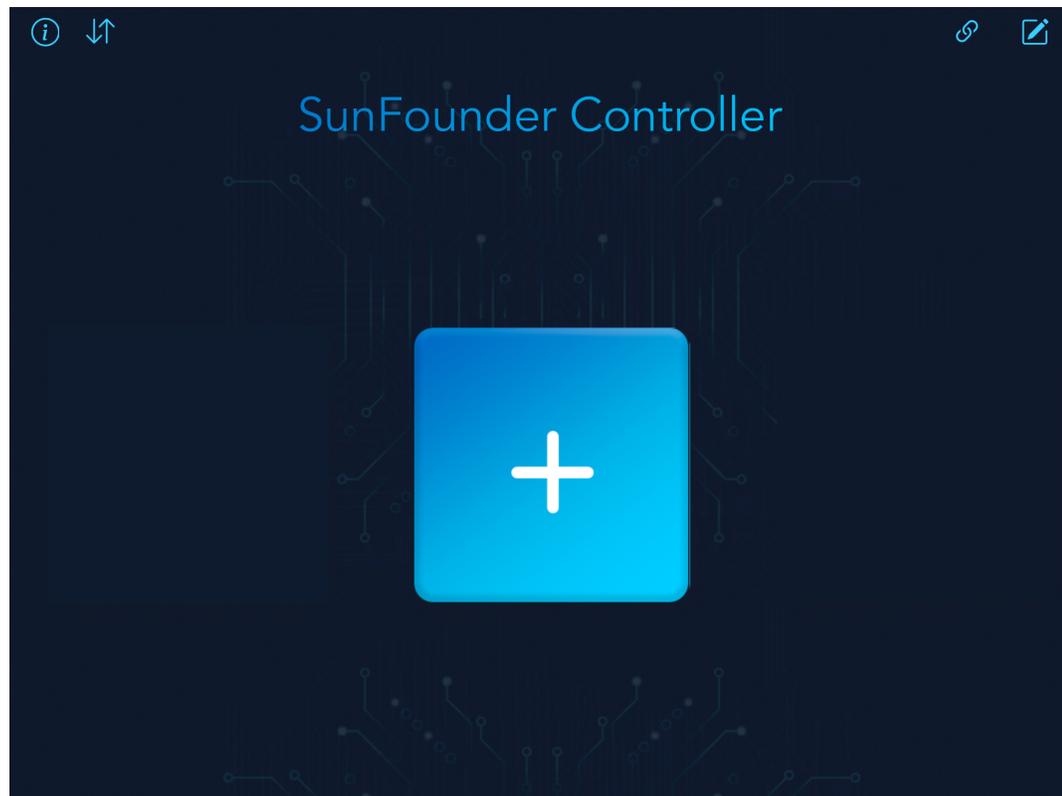
```
cd ~
git clone https://github.com/sunfounder/sunfounder-controller.git
cd ~/sunfounder-controller
sudo python3 setup.py install
```

2. Führen Sie den Code aus.

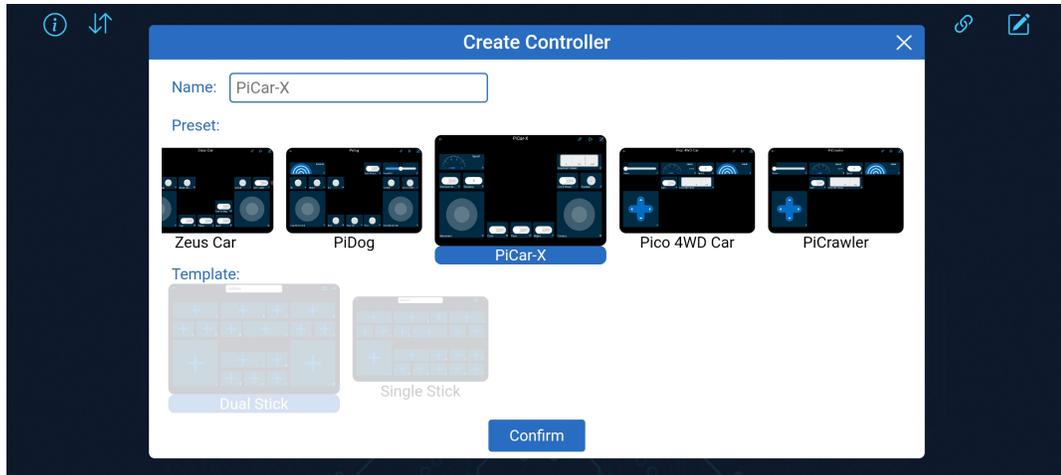
```
cd ~/picar-x/example
sudo python3 13.app_control.py
```

3. Installieren Sie [SunFounder Controller](#) aus dem **App Store(iOS)** oder **Google Play(Android)**.
4. Öffnen und erstellen Sie einen neuen Controller.

Erstellen Sie einen neuen Controller, indem Sie im SunFounder Controller-App auf das + Zeichen klicken.

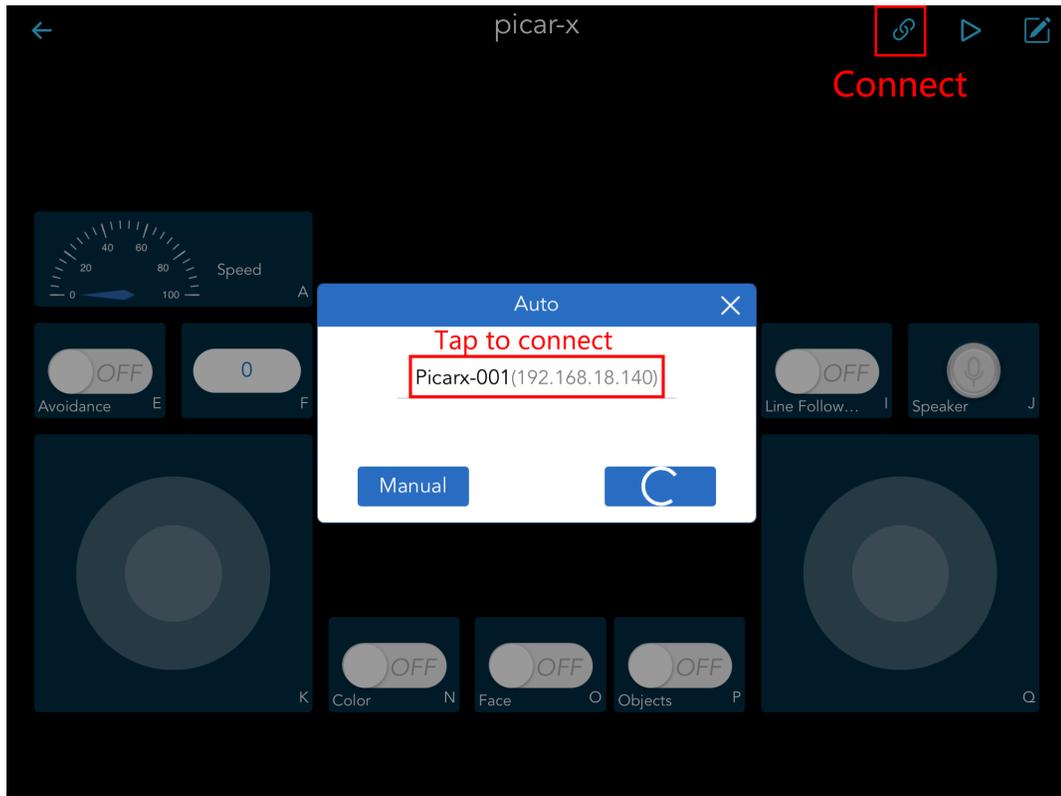


Im Abschnitt ‚Voreinstellungen‘ gibt es voreingestellte Controller für einige Produkte, die Sie bei Bedarf verwenden können. Hier wählen wir PiCar-X aus.

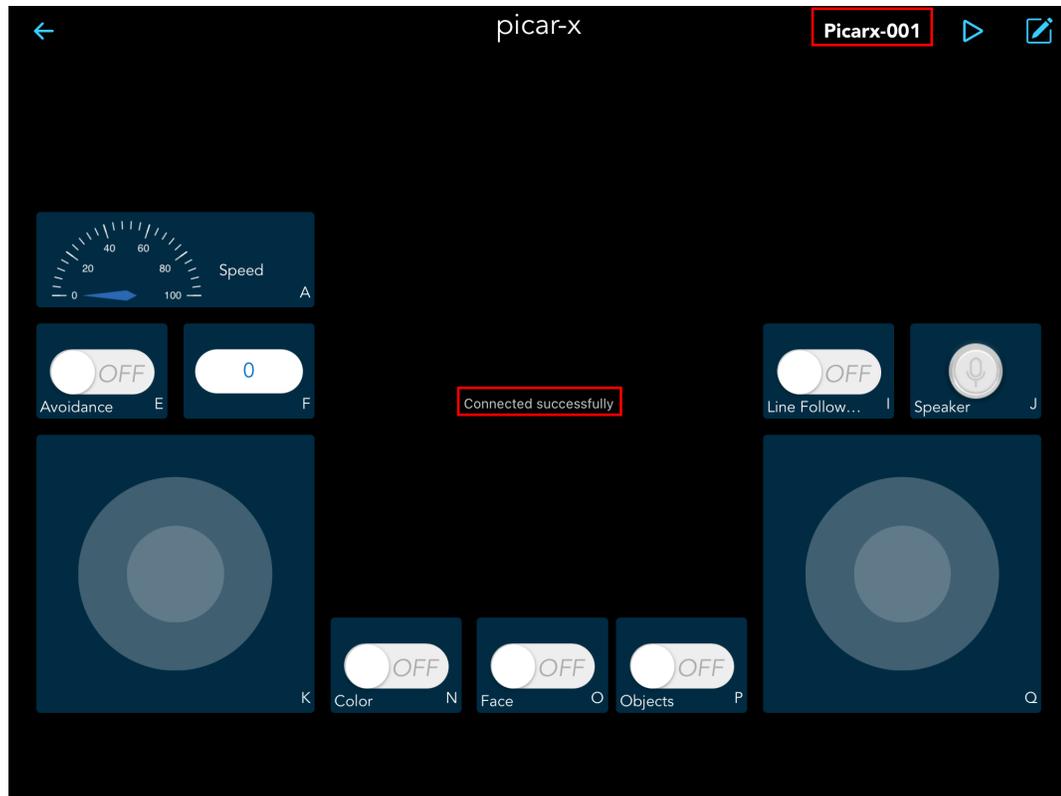


5. Verbinden Sie sich mit PiCar-x.

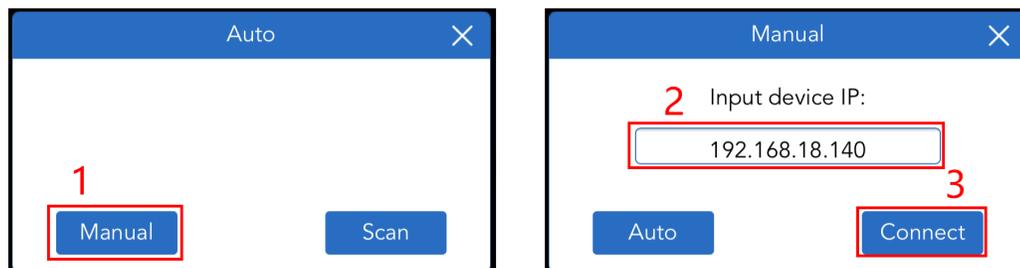
Wenn Sie auf den **Connect**-Button klicken, wird automatisch nach Robotern in der Nähe gesucht. Der Name ist in `picarx_control.py` definiert und muss ständig laufen.



Sobald Sie auf den Produktnamen klicken, erscheint die Meldung „Connected Successfully“ und der Produktname wird in der oberen rechten Ecke angezeigt.

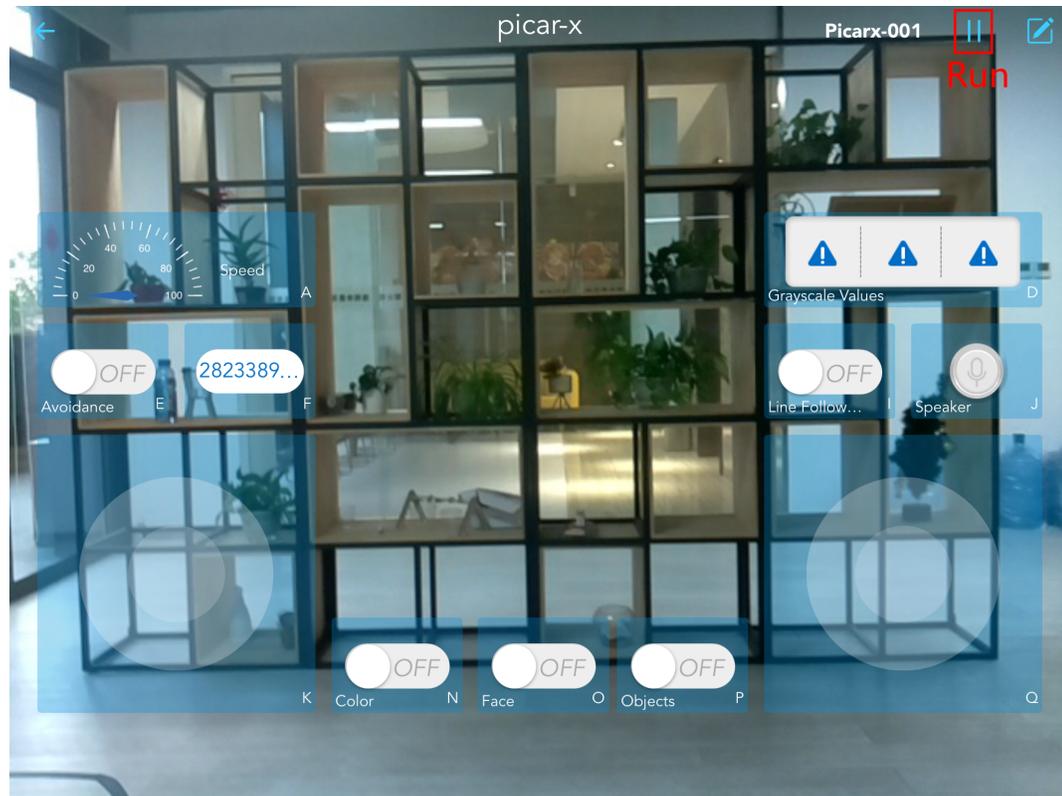
**Bemerkung:**

- Sie müssen sicherstellen, dass Ihr mobiles Gerät mit demselben LAN wie PiCar-X verbunden ist.
- Wenn es nicht automatisch sucht, können Sie auch manuell die IP eingeben, um sich zu verbinden.



6. Verwenden Sie diesen Controller.

Klicken Sie auf den **Run**-Button, um den Controller zu starten, Sie sehen das Filmmaterial des Autos und können jetzt Ihr PiCar-X mit diesen Widgets steuern.



Hier sind die Funktionen der Widgets.

- **A:** Zeigt die aktuelle Geschwindigkeit des Autos an.
- **E:** Schaltet die Hindernisvermeidungsfunktion ein.
- **I:** Schaltet die Linienverfolgungsfunktion ein.
- **J:** Spracherkennung, halten Sie dieses Widget gedrückt, um zu sprechen, und es zeigt die erkannte Stimme an, wenn Sie es loslassen. Wir haben die 4 Befehle `forward`, `backard`, `left` und `right` im Code festgelegt, um das Auto zu steuern.
- **K:** Steuert Vorwärts-, Rückwärts-, Links- und Rechtsbewegungen des Autos.
- **Q:** Dreht den Kopf (Kamera) nach oben, unten, links und rechts.
- **N:** Schaltet die Farberkennungsfunktion ein.
- **O:** Schaltet die Gesichtserkennungsfunktion ein.
- **P:** Schaltet die Objekterkennungsfunktion ein, sie kann fast 90 Arten von Objekten erkennen, für die Liste der Modelle, siehe: https://github.com/sunfounder/vilib/blob/master/workspace/coco_labels.txt.

Dieser Videokurs ist ein Online-Tutorial für Python, das speziell für interaktives Lernen konzipiert wurde. Er umfasst drei Einführungsvideos, die die wesentlichen Grundlagen abdecken: angefangen bei der Einrichtung des Raspberry Pi, dem Zusammenbau des PiCar-X und der Installation der erforderlichen Roboter-Module. Diese Anfangsphase stellt sicher, dass alles vorbereitet ist, bevor mit den verschiedenen Projekten mit dem PiCar-X begonnen wird.

Nach dem Einführungsteil bietet der Kurs 12 Projektvideos. Diese Projekte entwickeln schrittweise Fähigkeiten, beginnend mit der grundlegenden Bewegung des PiCar-X, Tastatursteuerung, Text-zu-Sprache (TTS), Hindernisvermeidung, Linienverfolgung bis hin zu Anwendungen in der Computer Vision. Anschließend werden komplexere Projekte vorgestellt, die mehrere Funktionalitäten kombinieren. Neben dem Ausführen von Beispielen aus dem Online-Tutorial bieten die Videos auch zusätzliche Erweiterungen zu jedem Thema, die ein tieferes Verständnis jeder Funktion ermöglichen.

Erste Schritte

Der Kurs beginnt mit drei Einführungsvideos, die die Grundlage für die Arbeit mit dem PiCar-X bilden. Diese Videos umfassen:

5.1 Video A1: Einstieg mit dem Raspberry Pi

Dies ist das erste Video für den PiCar-X.

Dieses Video bietet ein umfassendes Tutorial zur Einrichtung eines Raspberry Pi für die Verwendung mit dem PiCar-X Roboter-Auto. Es umfasst:

- Die Funktionen des PiCar-X.
- Wie man das Raspberry Pi OS abbildet.
- Verschiedene Programmiermethoden, einschließlich der Verwendung von HDMI, PowerShell, Remote Desktop und SunFounder Create Agent.
- Installation der notwendigen Roboter-Module.

Wenn Sie Anfänger sind, wird vorgeschlagen, den Schritten nacheinander zu folgen. Wenn Sie bereits mit dem Raspberry Pi vertraut sind, können Sie direkt den letzten Teil des Videos verfolgen, um die notwendigen Module zu installieren.

Video

Zugehörige Online-Tutorials

- *Schnellstartführer für Python*

5.2 Video A2: Zusammenbau des PICAR-X

In diesem Video liegt der Fokus auf dem Zusammenbau des PICAR-X.

Wichtige Aspekte des Tutorials umfassen:

- **Auspacken:** Vorstellung der Komponenten des Kits, einschließlich der Montageanleitung, Werkzeuge, Module und Motoren.
- **Montageanleitung:** Detaillierte Schritte zum Zusammenbau des PICAR-X, beginnend mit dem Anbringen von Schrauben am Raspberry Pi, dem Verbinden von Kabeln und dem Einrichten des Roboter-Huts.
- **Servo-Kalibrierung:** Verwendung von SunFounders Create Agent zum Nullen der Servos.
- **Montage von Motor und Batterie:** Anbringen der Motoren am Rahmen und Befestigen der Batterie.
- **Sensor- und Kamera-Installation:** Installation und Verbindung des Graustufensensors, des Ultraschallsensors und der Kamera.
- **Verkabelung und Abschluss-Setup:** Organisieren und Verbinden der Kabel für verschiedene Komponenten, gefolgt von abschließenden Überprüfungen und ordentlichem Kabelmanagement.

Das Video ist so konzipiert, dass es Sie Schritt für Schritt durch den Montageprozess führt und ein gründliches Verständnis dafür vermittelt, wie jede Komponente zusammenpasst.

Video

Zugehörige Online-Tutorials

- *Montageanleitung*

5.3 Video A3: Kalibrierung des PiCar-X

Dieses dritte Video der Serie konzentriert sich auf die Kalibrierung des PiCar-X und die Einführung des Robot HAT. Das Tutorial ist in mehrere Schlüsselabschnitte unterteilt:

- **Kalibrierung des Graustufensensors:** Anweisungen zum Zugriff auf und zur Kalibrierung des Graustufensensors.
- **Kalibrierung der Gleichstrommotoren und Servos:** Schritte zur Kalibrierung der Gleichstrommotoren für Richtung und Geschwindigkeit sowie der Lenk- und Kamera-Pan-/Tilt-Servos.
- **Skript-Automatisierung beim Start:** Wie man Skripte einstellt, die automatisch beim Start des Raspberry Pi ausgeführt werden.
- **Überblick über den Robot HAT:** Detaillierter Überblick über den Robot HAT, einschließlich seiner Funktionen und Funktionalitäten.

Dieses Tutorial ist entscheidend für diejenigen, die ihren PiCar-X feinjustieren und die technischen Aspekte des Robot HAT verstehen möchten.

Video

Zugehörige Online-Tutorials

- *0. Kalibrierung des PiCar-X*

Projekte

Nach der Einrichtung widmet sich der Kurs zwölf projektbasierten Videos. Diese Videos erweitern schrittweise die Fähigkeiten des PiCar-X, angefangen bei grundlegenden Bewegungen bis hin zu komplexeren Aufgaben, einschließlich:

5.4 Video 1: Motorsteuerung und Lenkkontrolle

Dieses Video dient als erstes Projekt-Tutorial in der PiCar-X-Reihe und konzentriert sich darauf, wie man die Motoren und den Lenkservo des PiCar-X steuert. Die Schlüsselinhalte umfassen:

- **Motoren Starten und Stoppen:** Demonstriert, wie man die Bewegung des PiCar-X steuert, einschließlich Vorwärts- und Rückwärtsfahren.
- **Geschwindigkeitskontrolle der Motoren:** Zeigt, wie man die Geschwindigkeit der Motoren erhöht und verringert.
- **Lenksteuerung:** Lehrt, wie man durch Steuerung des vorderen Lenkservos nach links und rechts abbiegt.
- **Skriptausführung:** Veranschaulicht, wie man ein Programm automatisch beim Start des Raspberry Pi ausführt.
- **Move.py Code-Analyse:** Bietet eine ausführliche Erklärung des Move.py-Skripts und erläutert die Logik hinter der Steuerung von Motoren und Lenkung.
- **Testen und Debuggen:** Praktisches Testen, um sicherzustellen, dass sowohl der Code als auch die Hardware korrekt funktionieren.

Dieses Video ist ein ausgezeichnete praktischer Leitfaden für Anfänger im Umgang mit PiCar-X und bietet detaillierte Anweisungen zur Steuerung von Motoren und Lenkung, was eine solide Grundlage für die weitere Projektentwicklung legt.

Video

Zugehörige Online-Tutorials

- *1. Lassen Sie PiCar-X sich bewegen*

5.5 Video 2: Steuerung des PiCar-X mit der Tastatur

In diesem Video-Tutorial lernen Sie, wie Sie den PiCar-X-Roboter mit einer Tastatur steuern. Es umfasst:

- **Grundlegende Steuerung:** Demonstration der Steuerung des PiCar-X mit Tastaturbefehlen - W für vorwärts, S für rückwärts, A für Linkskurven und D für Rechtskurven.
- **Neigung und Schwenk der Kamera:** Lehrt die Steuerung der Neigung und des Schwenks der montierten Kamera mit I (hoch), K (runter), J (links) und L (rechts).
- **Code-Erklärung:** Bietet eine detaillierte Erklärung des Python-Codes für die Tastatursteuerung, einschließlich Import von Bibliotheken und Steuerungslogik.
- **Code-Ausführung:** Zeigt, wie der Python-Code für die Tastatursteuerung ausgeführt wird, einschließlich der Verbindung zum PiCar-X.
- **Steuerung Beenden:** Erklärt das Beenden der Tastatursteuerung durch Drücken von `Strg + C`, wodurch der Roboter in seinen Ausgangszustand zurückkehrt.

Dieses Tutorial ist ideal für Anfänger und Robotik-Enthusiasten und bietet klare Anweisungen sowie eine praktische Demonstration zur Steuerung des PiCar-X-Roboters mit einer Tastatur.

Video

Zugehörige Online-Tutorials

- *2. Tastatursteuerung*

5.6 Video 3: Text in Sprache

Dieses Tutorial behandelt die Text-zu-Sprache-Funktionen des PiCar-X-Roboters:

- **Roboter-Setup und Erste Bewegungen:** Demonstration der Einsatzbereitschaft des Roboters und seiner ersten Bewegungen, einschließlich Anhalten und Hinderniserkennung.
- **Text-zu-Sprache-Fähigkeiten:** Zeigt, wie der Roboter vordefinierte Phrasen spricht und einen Countdown mittels Text-zu-Sprache-Technologie durchführt.
- **Demonstration eines Benutzerdefinierten Skripts:** Führt ein benutzerdefiniertes Skript aus, in dem der Roboter spricht, sich bewegt, auf Hindernisse reagiert und Wendemanöver durchführt.
- **Musikwiedergabe:** Lehrt, wie Musikdateien auf dem Roboter mit Python-Skripten abgespielt werden.

Die Lektion bietet ein umfassendes Tutorial zur Integration der Text-zu-Sprache-Funktionalität in den PiCar-X-Roboter, einschließlich praktischer Demonstrationen und Code-Details.

Video

Zugehörige Online-Tutorials

- *3. Text-zu-Sprache & Soundeffekte*

5.7 Video 4: Hindernisvermeidung mit Ultraschall

Dieses Video-Tutorial behandelt die Hindernisvermeidung mithilfe eines Ultraschallsensors im PiCar-X-Roboter:

- **Einführung in den Ultraschallsensor:** Erklärt, wie der Ultraschallsensor zur Messung von Entfernungen und zur Steuerung der Bewegung des Autos verwendet wird.
- **Python-Code-Durchgang:** Erläutert im Detail den Python-Code für die Hindernisvermeidung, einschließlich Variablendefinitionen und bedingter Bewegungslogik.
- **Erstellen und Ausführen Benutzerdefinierter Skripte:** Zeigt, wie man benutzerdefinierte Skripte für Wendemanöver und Reaktionen auf Hindernisse schreibt und ausführt.
- **Praktische Demonstrationen:** Bietet Demonstrationen der Hindernisvermeidungsfähigkeiten des Roboters auf verschiedenen Oberflächen.
- **Speichern und Aktualisieren des Codes:** Erklärt, wie man Skripte auf dem Roboter speichert und aktualisiert.

Diese Lektion bietet einen wesentlichen Leitfaden zur Implementierung von ultraschallbasierten Hindernisvermeidung im PiCar-X-Roboter.

Video

Zugehörige Online-Tutorials

- *4. Hindernisvermeidung*

5.8 Video 5: Graustufen-Linienverfolgung

Dieses Video-Tutorial befasst sich mit der Graustufen-Linienverfolgung mithilfe des PiCar-X-Roboters:

- **Linientdeckung:** Demonstriert, wie der Roboter mit einem Graustufensensor eine schwarze Linie auf einer weißen Oberfläche erkennt, was ihm ermöglicht, der Linie zu folgen oder darüber zu fahren.
- **Python-Code-Erklärung:** Erklärt den Python-Code, der in der Linienverfolgung verwendet wird, und geht detailliert auf den Import von Modulen, die Erstellung von Objekten und die Logik zur Reaktion auf Linientdeckung ein.
- **Praktische Demonstrationen:** Bietet Demonstrationen, wie der Roboter Linien auf verschiedenen Oberflächen erkennt und verfolgt.
- **Fehlerbehebung und Anpassung:** Diskutiert die Notwendigkeit von Anpassungen und Korrekturen am Code für eine bessere Leistung bei der Linienverfolgung.

Diese Lektion bietet einen umfassenden Leitfaden zur Implementierung der Graustufen-Linienverfolgung im PiCar-X, einschließlich praktischer Demonstrationen, Code-Durchläufen und Tipps zur Fehlerbehebung.

Video

Zugehörige Online-Tutorials

- [5. Linienverfolgung](#)

5.9 Video 6: Abgrunderkennung

Dieses Tutorial bietet wesentliche Einblicke in die Programmierung und Nutzung der Abgrunderkennung im PiCar-X-Roboter.

- **Einführung:** Erklärt die Verwendung eines Graustufensensors im PiCar-X zur Abgrunderkennung durch Messung von Oberflächenreflexionen.
- **Python-Code-Durchgang:** Behandelt den Code für die Abgrunderkennung, einschließlich der Sensor-Setup und der Programmlogik.
- **Demonstration und Testen:** Zeigt, wie der Roboter Abgründe erkennt und darauf reagiert, mit Schritten zum Ausführen und Testen des Codes.
- **Auto-Start-Einrichtung:** Leitet zur Konfiguration des Raspberry Pi an, um das Skript zur Abgrunderkennung beim Booten auszuführen.
- **Praktische Anwendung:** Demonstriert die Reaktion des Roboters auf verschiedene Oberflächen und Kanten, wodurch seine Fähigkeit zur Abgrunderkennung hervorgehoben wird.

Video

Zugehörige Online-Tutorials

- [6. Klippenerkennung](#)

5.10 Video 7: PiCar-X Computervision

Dieses Video-Tutorial konzentriert sich auf die Fähigkeiten der Computervision im PiCar-X:

- **Einführung in die Computervision:** Lehrt, wie man mit der Kamera ausgestatteten PiCar-X verwendet, um Handbewegungen, Finger, Farben, Gesichter zu erkennen und QR-Codes zu lesen.
- **Remote-Desktop und Python-Code-Ausführung:** Demonstriert die Verwendung von VNC für den Remote-Desktop-Zugriff auf den Raspberry Pi und das Ausführen von Python-Code für Aufgaben der Computervision.
- **Farberkennung und Fotoaufnahme:** Zeigt, wie man verschiedene Farben erkennt und Fotos mit Kamerasteuerungen aufnimmt.
- **QR-Code- und Gesichtserkennung:** Erklärt, wie man zwischen QR-Code-Lesen und Gesichtserkennungsfunktionen wechselt.
- **Objekterkennung:** Diskutiert die Verwendung von integrierten Funktionen aus der SunFounder VI-Bibliothek für die Objekterkennung.
- **Videoanzeige im Browser und auf dem Handy:** Lehrt, wie man das Kamerabild auf einen Browser oder ein Mobiltelefon streamt, vorausgesetzt, sie sind mit demselben Netzwerk wie der PiCar-X verbunden.
- **Handerkennung:** Behandelt die Verwendung der Handerkennungsfunktion aus der VI-Bibliothek und das Ausführen des entsprechenden Python-Codes.
- **Codebearbeitung und -ausführung:** Demonstriert das Erstellen, Bearbeiten und Ausführen von Python-Skripten für verschiedene Aufgaben der Computervision.

Diese Lektion bietet einen umfassenden Leitfaden zur Erforschung der Funktionen der Computervision im PiCar-X, einschließlich praktischer Demonstrationen von Farberkennung, QR-Code-Lesen, Gesichtserkennung und Verfolgung von Handbewegungen.

Video

Zugehörige Online-Tutorials

- [7. Computervision](#)

5.11 Video 8: PiCar-X Blickt Dir Nach

Dieses Tutorial lehrt, wie man eine Kamera und einen Servo im PiCar-X-Roboter für die Objektverfolgung verwendet:

- **Kamera-Servo-Integration:** Demonstriert das Verbinden einer Kamera mit einem Servo zur Bewegungsverfolgung.
- **Python-Code-Überblick:** Erklärt den Code zur Steuerung der Kamerabewegungen des PiCar-X.
- **Kamera Starten & Videoanzeige:** Zeigt das Starten der Kamera und das Anzeigen des Videofeeds.
- **Gesichtsverfolgung:** Beschreibt, wie der Roboter ein menschliches Gesicht verfolgt und seine Kamerawinkel anpasst.
- **Demonstrationen:** Beinhaltet Demonstrationen, wie die Kamera des Roboters Bewegungen verfolgt und sich anpasst.
- **Videostreaming:** Behandelt das Streamen des Kamera-Feeds in einen Browser oder auf ein Mobiltelefon.

Diese Lektion bietet einen prägnanten Leitfaden, um den PiCar-X in die Lage zu versetzen, Objekte oder Gesichter mit seiner Kamera zu verfolgen und zu fokussieren.

Video

Zugehörige Online-Tutorials

- [8. Blick auf Dich](#)

5.12 Video 9: Videoaufnahme

Dieses Tutorial bietet eine prägnante, aber detaillierte Anleitung zur Aufnahme und Verwaltung von Videos mit dem PiCar-X-Roboter.

- **Überblick:** Lehrt, wie man HD 1080p-Videos mit dem PiCar-X, einem Raspberry Pi Selbstfahr-Roboterkit, aufzeichnet.
- **Dokumentation und Code:** Führt durch die PiCar-X-Dokumentation zur Videoaufnahme und erklärt das Python-Skript zur Steuerung der Aufnahme (Starten, Pausieren, Fortsetzen, Stoppen).
- **Remote-Desktop-Verbindung:** Demonstriert, wie man sich remote mit dem Raspberry Pi für die Videoaufnahme verbindet.
- **Praktische Aufnahme-Demonstration:** Zeigt, wie das Videoaufnahme-Skript ausgeführt und über einen Remote-Desktop bedient wird.
- **Videowiedergabe:** Veranschaulicht, wie aufgenommene Videos gefunden und abgespielt werden, um die Qualität zu überprüfen.
- **Zusätzliche Funktionen:** Stellt das PiCamera2 GitHub-Repository für weitere Aufnahmeoptionen wie Zeitraffer und einfache Aufnahme vor.
- **Einfache Video-Skripterstellung:** Hebt das Erstellen und Ausführen von einfachen Videoaufnahme-Skripten mit verschiedenen Formaten und Konfigurationen hervor.

Video

Zugehörige Online-Tutorials

- [9. Videoaufnahme](#)

5.13 Video 10: Stierkampf mit dem PiCar-X

Dieses Tutorial behandelt die Verwendung des PiCar-X-Roboters für ein „Stierkampf“-Spiel, wobei der Fokus auf Farberkennung und Bewegung liegt:

- **Stierkampf-Konzept:** Verwendet die Kamera des PiCar-X, um die rote Farbe zu verfolgen und mit Schwenk- und Neigungssteuerung zu folgen.
- **Python-Code-Überblick:** Erläutert die Programmierung hinter der Farberkennung und den Roboterbewegungen.
- **Remote-Desktop-Zugang:** Demonstriert das Verwalten des PiCar-X-Codes über einen Remote-Desktop.
- **Mechanik der Farbverfolgung:** Erklärt, wie der Roboter rote Objekte verfolgt und seine Bewegung entsprechend anpasst.
- **Demonstration:** Zeigt den PiCar-X in Aktion, wie er einem roten Ziel folgt.

Diese Lektion bietet einen umfassenden Leitfaden zur Programmierung des PiCar-X für ein Farbverfolgungsspiel, einschließlich Code-Erklärungen und einer Live-Demonstration.

Video

Zugehörige Online-Tutorials

- *10. Stierkampf*

5.14 Video 11: PiCar-X als Videowagen

Dieses Tutorial lehrt, wie man den PiCar-X als Videowagen mit Kamerasteuerung verwendet:

- **Steuermechanik:** Stellt Tastatursteuerungen für Bewegungen vor – vorwärts, rückwärts, links, rechts und anhalten.
- **Videowagen-Funktionen:** Konzentriert sich auf die Verwendung des PiCar-X zum Fahren, Geschwindigkeitsanpassung, Abbiegen und Fotografieren.
- **Python-Code-Überblick:** Erklärt kurz den Python-Code zur Steuerung des Wagens und der Kamera.
- **Remote-Desktop und Programm-Setup:** Zeigt das Einrichten des Programms des Wagens über einen Remote-Desktop und das Ausführen beim Start.
- **Live-Demonstration:** Beinhaltet Demonstrationen der Steuerung des PiCar-X mit Tastatureingaben und der Verwendung seiner Kamera.
- **Fotoaufnahme und -ansicht:** Lehrt das Aufnehmen und Ansehen von Fotos, die vom PiCar-X gemacht wurden.

Die Lektion bietet einen prägnanten Überblick über den Betrieb des PiCar-X als Videowagen, mit Schwerpunkt auf praktischer Steuerung und Kameranutzung.

Video

Zugehörige Online-Tutorials

- *11. Video-Auto*

5.15 Video 12: Schatzsuche-Spiel

Das Tutorial bietet eine spannende und lehrreiche Erfahrung in der Programmierung und Robotik mit dem PiCar-X.

- **Überblick:** Demonstriert ein Schatzsuche-Spiel, bei dem der PiCar-X-Roboter zufällig ausgewählte Farben erkennt und darauf zusteuert.
- **Python-Code:** Detaillierte Erklärung des Python-Codes, der für die Farberkennung, die Steuerung der Roboterbewegung und die Text-zu-Sprache-Ansagen verwendet wird.
- **Spielverlauf:** Der Roboter bewegt sich mit Tastatureingaben, um eine Ziel-Farbe zu finden und zu erreichen, die mittels Text-zu-Sprache angesagt wird.
- **Praktische Demonstration:** Zeigt den Roboter in Aktion, der erfolgreich verschiedene Farben wie Rot, Gelb und Blau identifiziert und darauf zusteuert.
- **Spiel-Exit-Anweisungen:** Behandelt, wie man das Spiel sicher beendet, einschließlich des Anhaltens des Roboters und des Ausschaltens der Kamera.

Video

Zugehörige Online-Tutorials

- *12. Schatzsuche*

Mit Ezblock spielen

Für Anfänger ist EzBlock eine Softwareentwicklungsplattform, die von SunFounder für den Raspberry Pi angeboten wird. EzBlock bietet zwei Programmierumgebungen: eine grafische Umgebung und eine Python-Umgebung.

Sie ist für fast alle Arten von Geräten verfügbar, einschließlich Mac, PC und Android.

Hier ist ein Tutorial, das Ihnen hilft, die Installation, den Download und die Nutzung von EzBlock abzuschließen.

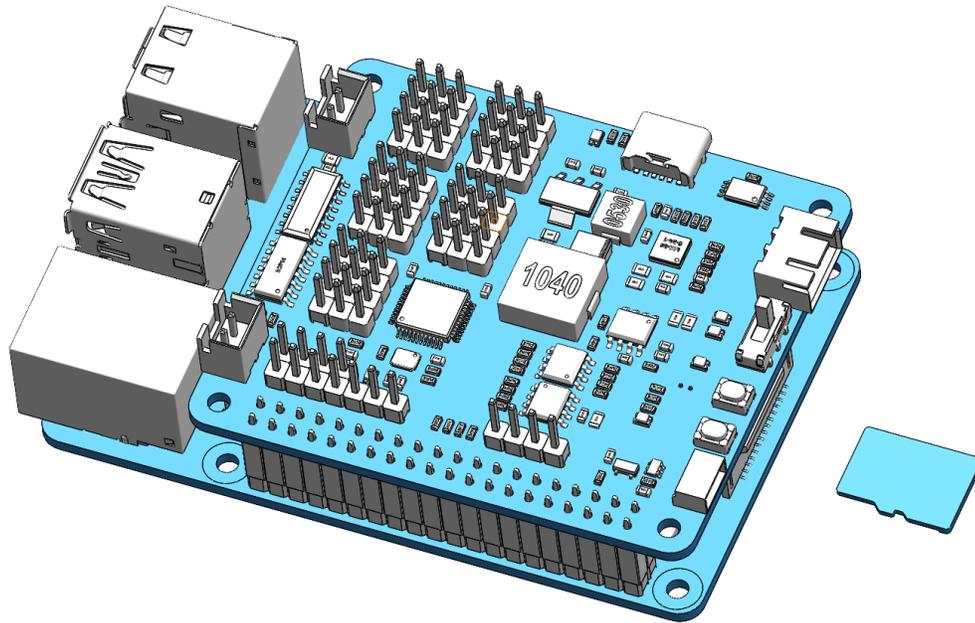
6.1 Schnellanleitung für EzBlock

Der Drehwinkelbereich des Servos liegt zwischen -90° und 90° . Jedoch ist der im Werk eingestellte Winkel zufällig, vielleicht 0° oder vielleicht 45° . Wenn wir ihn direkt mit solch einem Winkel montieren, führt dies zu einem chaotischen Zustand, sobald der Roboter den Code ausführt. Schlimmer noch, es kann dazu führen, dass der Servo blockiert und durchbrennt.

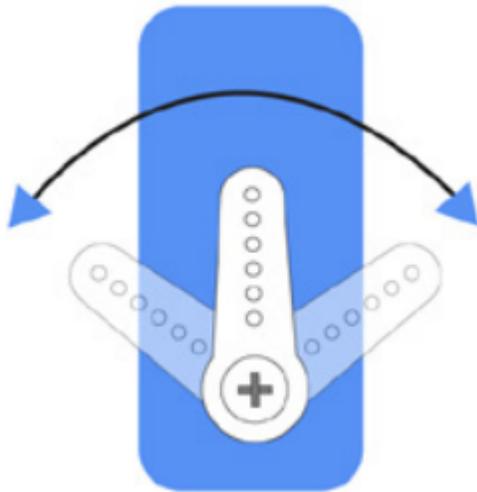
Deshalb müssen wir alle Servo-Winkel auf 0° einstellen und dann installieren. So ist der Servo-Winkel immer mittig, egal in welche Richtung er dreht.

1. Zunächst [Install EzBlock OS](#) (EzBlocks eigene Anleitungen) auf eine Micro-SD-Karte installieren. Nach Abschluss der Installation diese in den Raspberry Pi einsetzen.

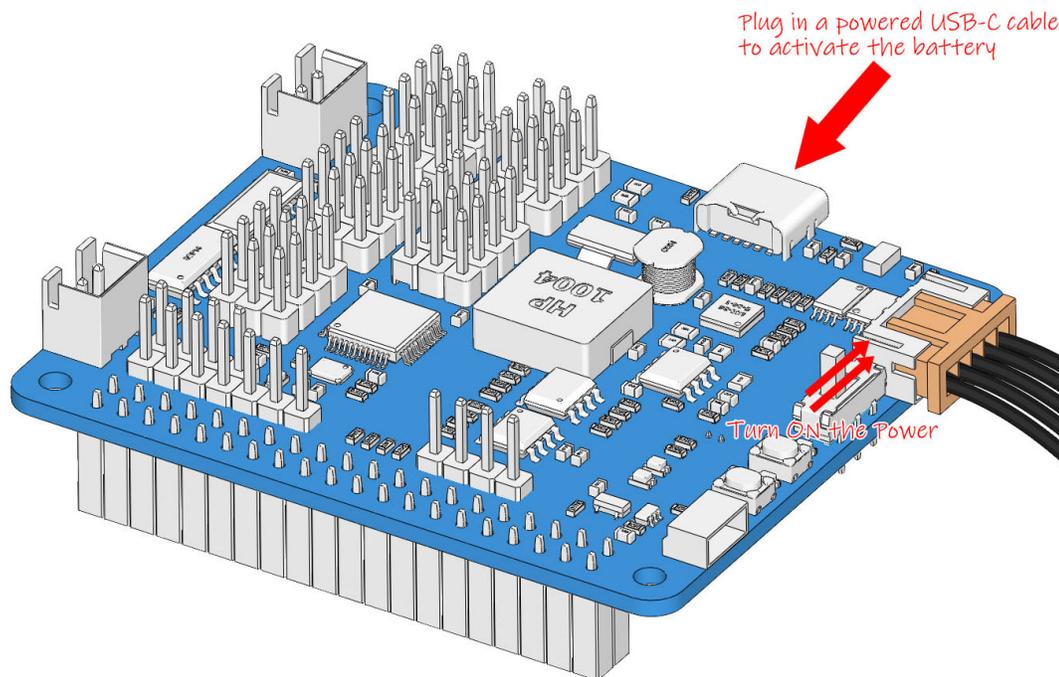
Bemerkung: Nach Abschluss der Installation kehren Sie bitte zu dieser Seite zurück.



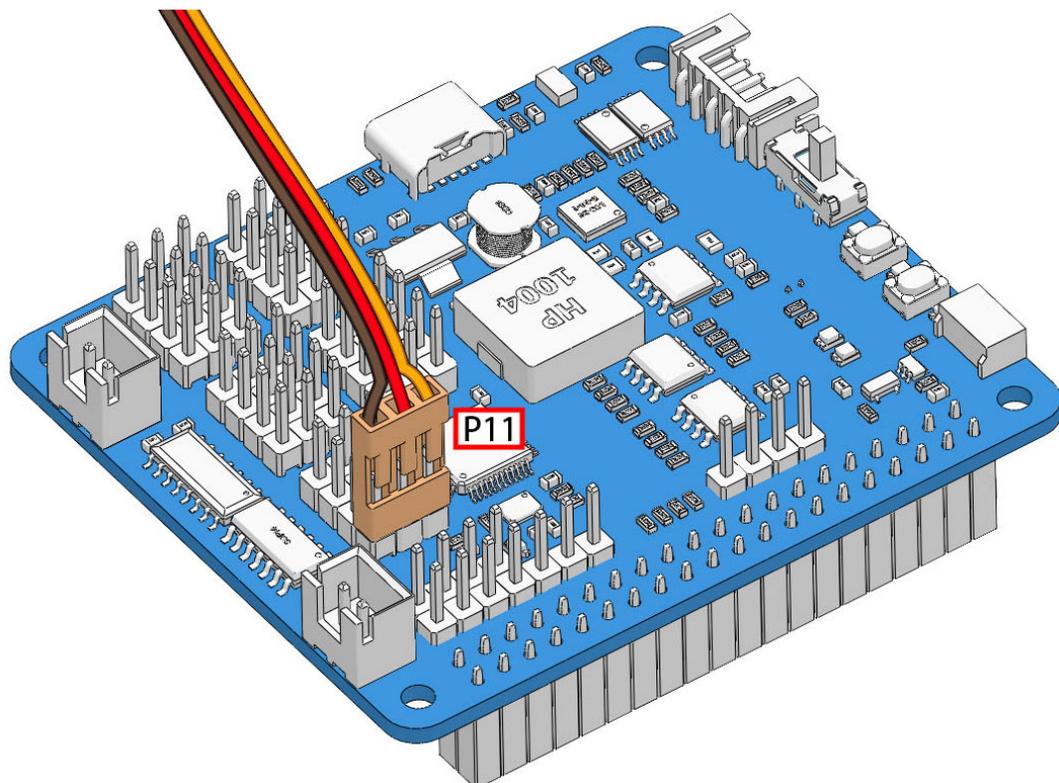
2. Um sicherzustellen, dass der Servo korrekt auf 0° eingestellt ist, den Servoarm in die Servoachse einstecken und dann den Hebelarm sanft in einen anderen Winkel drehen. Dieser Servoarm dient lediglich dazu, deutlich zu machen, dass sich der Servo dreht.



3. Den Anweisungen auf dem Montageblatt folgen, das Batteriekabel einstecken und den Ein-/Ausshalter auf ON stellen. Dann ein USB-C-Kabel mit Stromquelle anschließen, um die Batterie zu aktivieren. 1-2 Minuten warten; ein Ton signalisiert das erfolgreiche Hochfahren des Raspberry Pi.

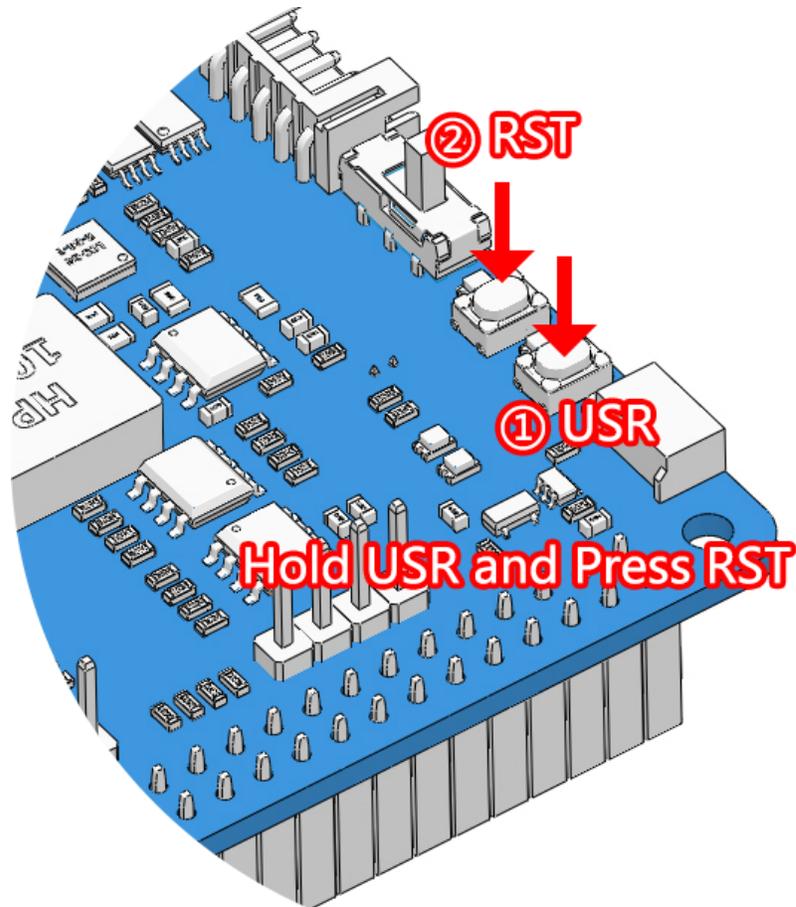


4. Nun das Servokabel wie folgt in den P11-Port einstecken.



5. Die **USR**-Taste gedrückt halten und dann die **RST**-Taste drücken, um das Servo-Nullstellungs-Skript im System auszuführen. Wenn der Servoarm sich dreht und eine Position einnimmt (dies ist die 0°-Position, die sich an einer zufälligen Stelle befindet und möglicherweise nicht senkrecht oder parallel ist), zeigt dies an, dass das Programm läuft.

Bemerkung: Dieser Schritt muss nur einmal durchgeführt werden; danach einfach andere Servokabel einstecken, und diese werden automatisch auf Null gestellt.



6. Jetzt den Servoarm entfernen, sicherstellen, dass das Servokabel verbunden bleibt und die Stromversorgung nicht abschalten. Dann die Montage gemäß der schriftlichen Montageanleitung fortsetzen.

Bemerkung:

- Das Servokabel nicht vor dem Festziehen des Servos mit der Servoschraube abziehen. Nach dem Festziehen kann es abgezogen werden.
 - Den Servo nicht drehen, solange er eingeschaltet ist, um Beschädigungen zu vermeiden. Wenn die Servoachse im falschen Winkel eingesetzt ist, den Servo herausziehen und erneut einsetzen.
 - Vor der Montage jedes Servos muss das Servokabel in P11 eingesteckt und die Stromversorgung eingeschaltet werden, um seinen Winkel auf 0° einzustellen.
 - Diese Nullstellungsfunktion wird deaktiviert, wenn Sie später ein Programm mit der EzBlock APP auf den Roboter herunterladen.
-

6.2 Installation und Konfiguration von EzBlock Studio

Sobald der Roboter montiert ist, sind einige grundlegende Schritte erforderlich.

- **Install EzBlock Studio:** Laden Sie EzBlock Studio auf Ihr Gerät herunter und installieren Sie es oder nutzen Sie die webbasierte Version.
- **Connect the Product and EzBlock:** Konfigurieren Sie Wi-Fi, Bluetooth und kalibrieren Sie vor der Benutzung.
- **Open and Run Examples:** Sehen Sie sich das zugehörige Beispiel direkt an oder führen Sie es aus.

Bemerkung: Nachdem Sie die Picar-x verbunden haben, gibt es einen Kalibrierungsschritt. Dies ist aufgrund möglicher Abweichungen im Montageprozess oder Einschränkungen der Servos selbst notwendig, die dazu führen könnten, dass einige Servowinkel leicht abweichen. Sie können diese in diesem Schritt kalibrieren.

Wenn Sie jedoch der Meinung sind, dass die Montage perfekt ist und keine Kalibrierung erforderlich ist, können Sie diesen Schritt auch überspringen.

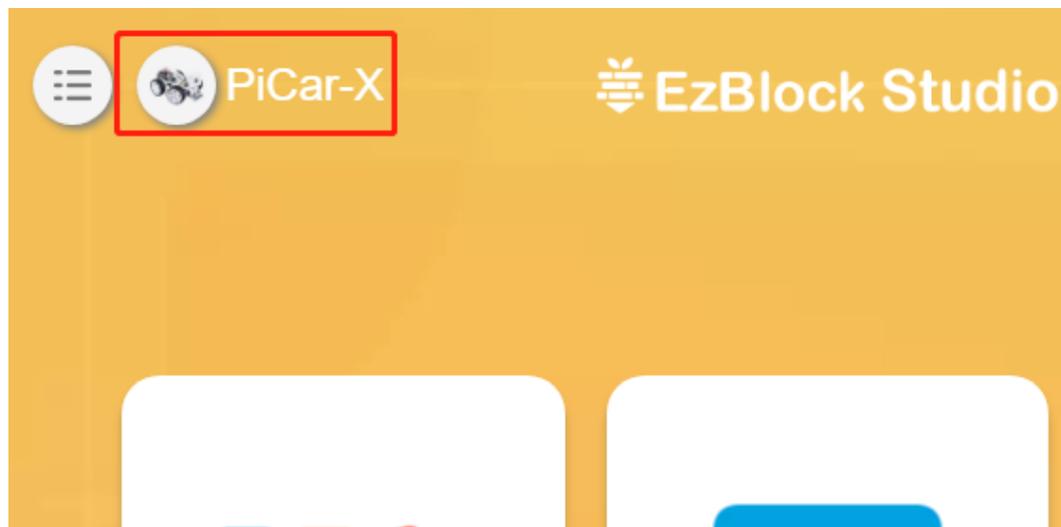
6.3 Kalibrierung des Fahrzeugs

Nachdem Sie die PiCar-X verbunden haben, folgt ein Kalibrierungsschritt. Dies ist notwendig, da es während der Montage zu möglichen Abweichungen oder durch die Begrenzungen der Servos selbst zu leicht schrägen Servo-Winkeln kommen kann. Diese können Sie im nächsten Schritt kalibrieren.

Falls Sie jedoch der Meinung sind, dass die Montage fehlerfrei ist und keine Kalibrierung erforderlich ist, können Sie diesen Schritt auch überspringen.

Bemerkung: Wenn Sie den Roboter während des Betriebs neu kalibrieren möchten, befolgen Sie bitte die unten aufgeführten Schritte.

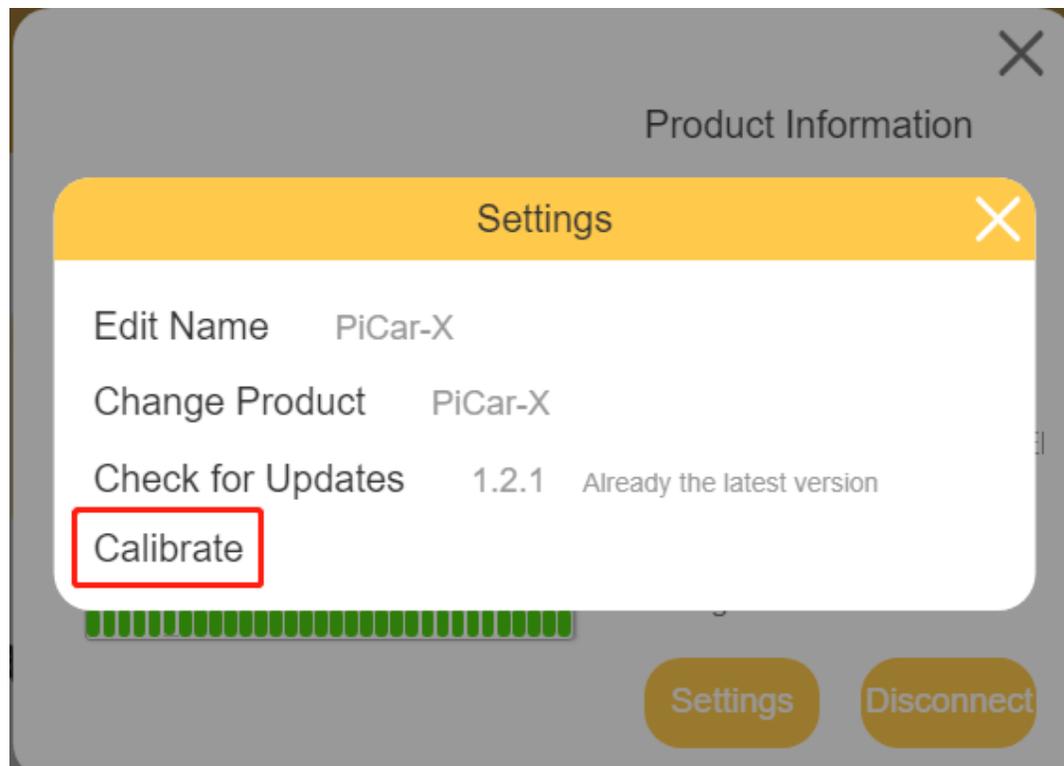
1. Öffnen Sie die Produktinformationsseite, indem Sie das Verbindungssymbol in der oberen linken Ecke anklicken.



2. Klicken Sie auf den **Einstellungen**-Button.



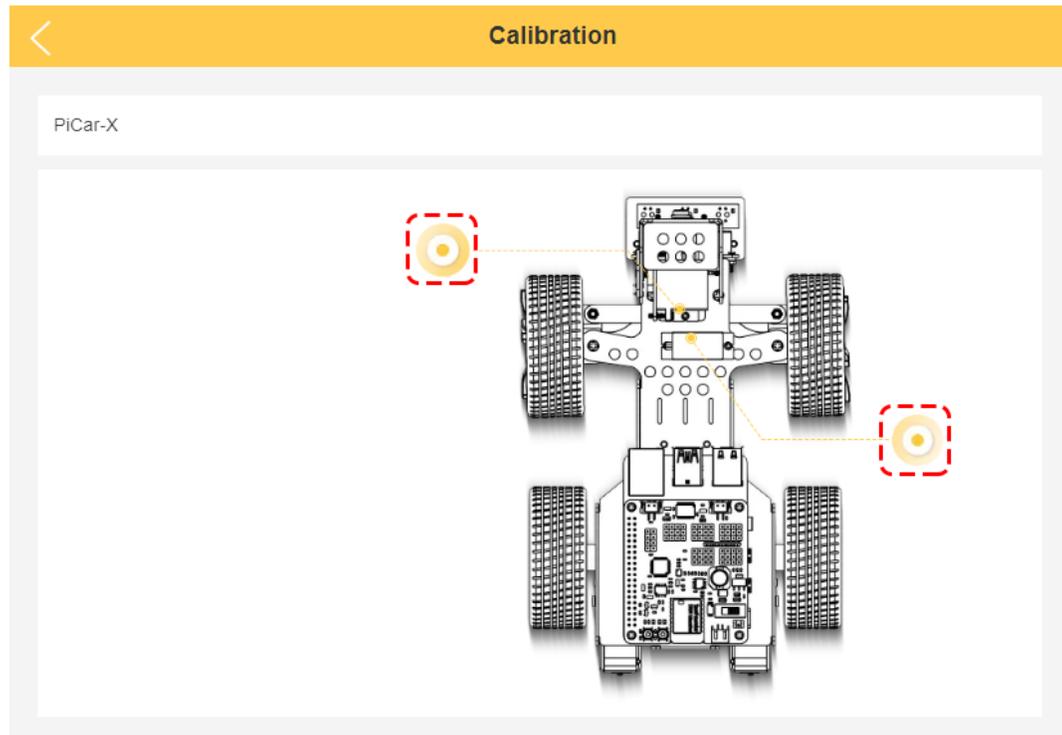
3. Auf dieser Seite können Sie den Produktnamen ändern, den Produkttyp auswählen, die App-Version einsehen oder den Roboter kalibrieren. Klicken Sie auf **Kalibrieren**, um zur Kalibrierungsseite zu gelangen.



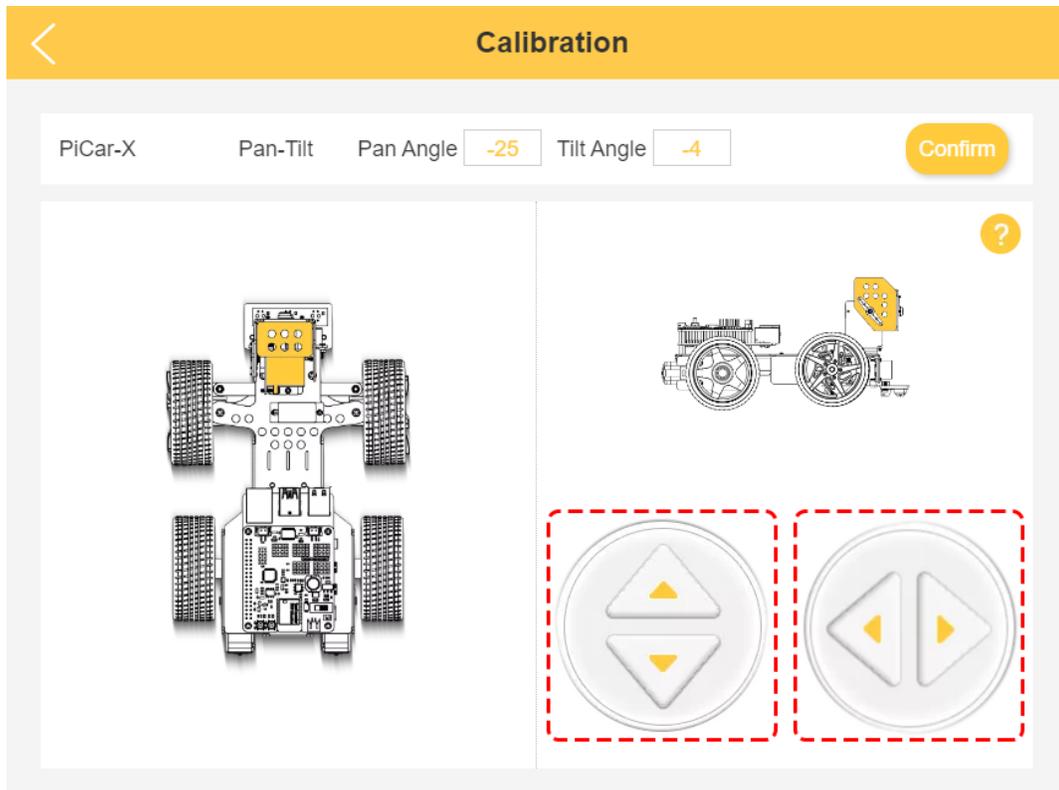
Die Kalibrierungsschritte lauten wie folgt:

1. Sobald Sie auf der Kalibrierungsseite sind, gibt es zwei Hinweispunkte, die Ihnen zeigen, wo kalibriert werden soll.

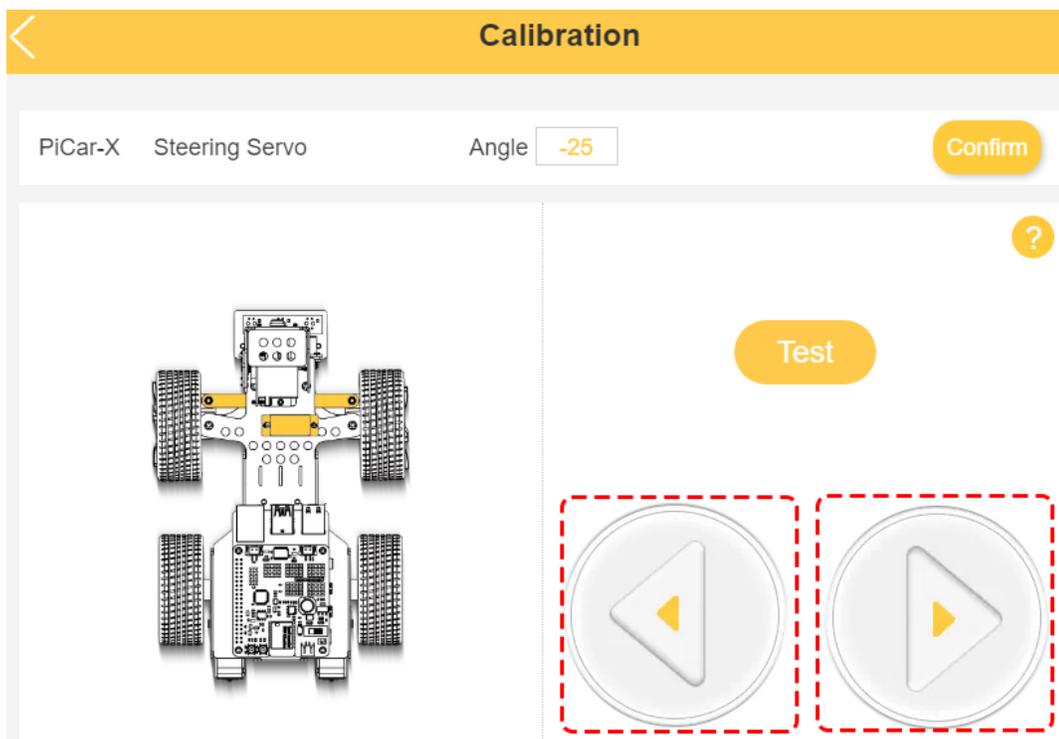
Bemerkung: Die Kalibrierung ist ein Prozess der Feineinstellung. Wenn Sie eine Taste bis zum Anschlag drücken und das Teil immer noch nicht passt, wird empfohlen, es abzunehmen und erneut zu montieren.



2. Klicken Sie auf den linken Hinweispunkt, um die Schwenk-Neige-Funktion (den Kamerabereich) der PiCar-X zu kalibrieren. Mit den beiden Tastensätzen auf der rechten Seite können Sie die Ausrichtung des Schwenk-Neige-Systems langsam anpassen und deren Winkel einsehen. Wenn die Einstellung abgeschlossen ist, klicken Sie auf **Bestätigen**.



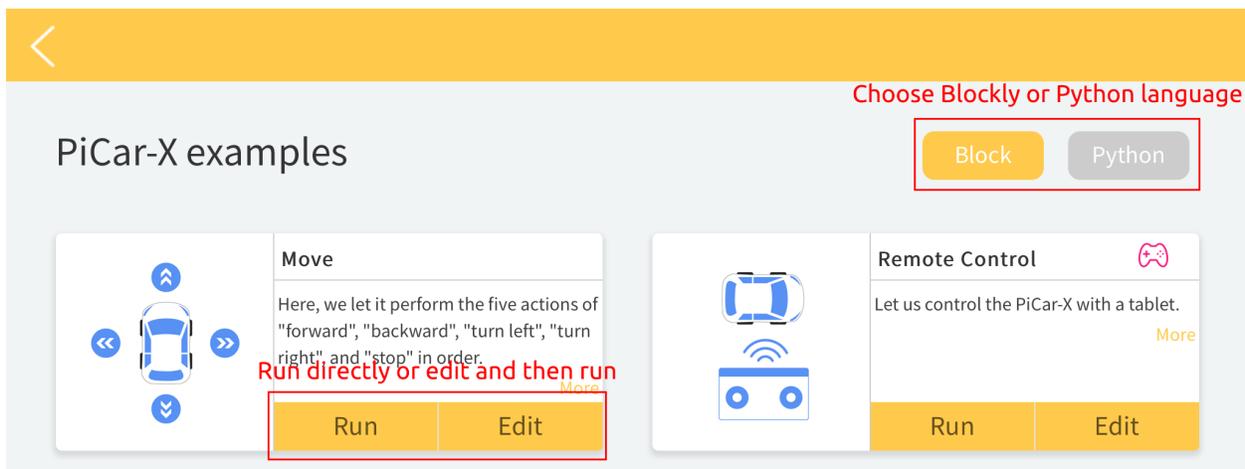
- Um die Ausrichtung des Vorderrads zu kalibrieren, klicken Sie auf den rechten Hinweispunkt. Verwenden Sie die beiden Tasten auf der rechten Seite, um das Vorderrad geradeaus auszurichten. Ist die Einstellung abgeschlossen, klicken Sie auf **Bestätigen**.



Dieser Abschnitt beginnt mit grundlegenden Programmierfunktionen für den PiCar-X und führt weiter zur Erstellung fortgeschrittenerer Programme im Ezbloc Studio. Jedes Tutorial enthält TIPPS, die neue Funktionen vorstellen, sodass Benutzer das entsprechende Programm schreiben können. Es gibt auch einen vollständigen Referenzcode im Beispielbereich, der direkt verwendet werden kann. Wir empfehlen, die Programmierung ohne den Code aus den Beispielabschnitten zu versuchen und den Spaß am Lösen von Herausforderungen zu genießen!

Alle Ezbloc-Projekte wurden auf die Beispielseite von Ezbloc Studio hochgeladen. Von der Beispielseite aus können Benutzer die Programme direkt ausführen oder die Beispiele bearbeiten und sie im Ordner „Meine Projekte“ des Benutzers speichern.

Die Beispielseite ermöglicht es den Benutzern, zwischen Block- oder Python-Sprache zu wählen. Die Projekte in diesem Abschnitt erklären nur die Block-Sprache. Für eine Erklärung des Python-Codes überprüfen Sie diese [Datei](#), um den Python-Code zu verstehen.



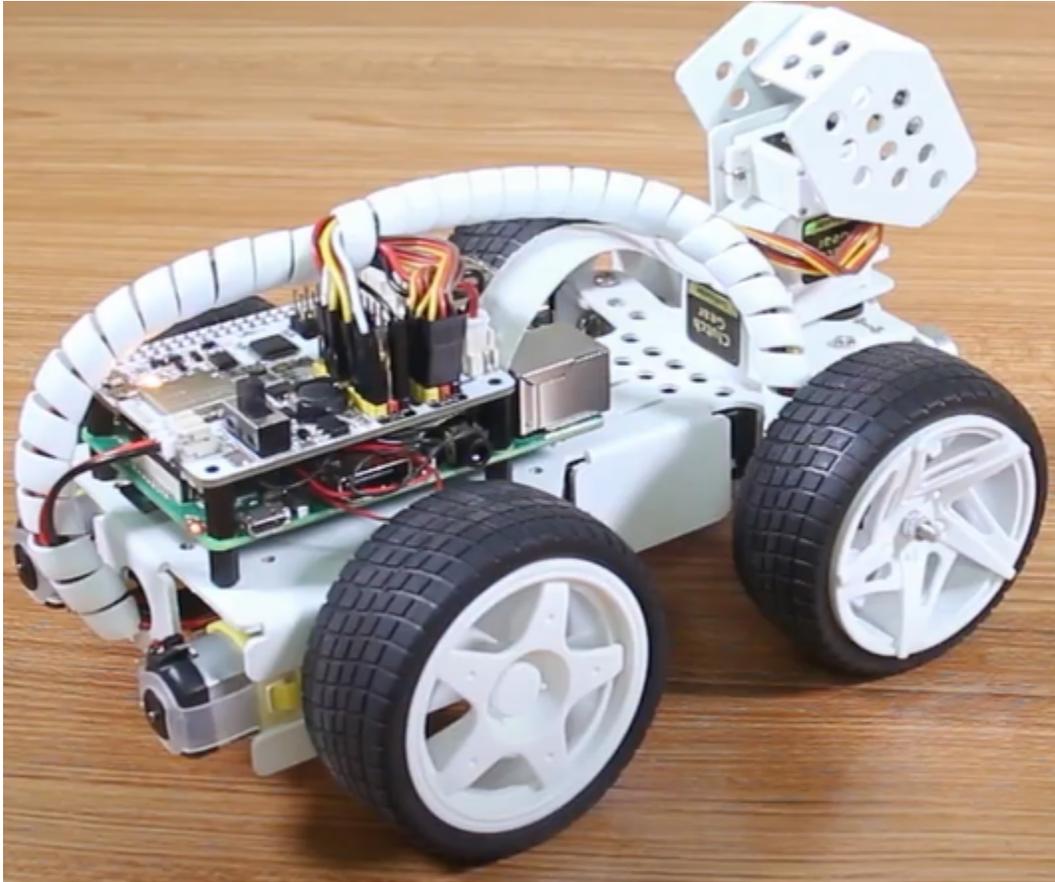
Grundlagen

6.4 Bewegung

Dieses erste Projekt zeigt Ihnen, wie Sie Bewegungsabläufe für die PiCar-X programmieren können. In diesem Projekt wird das Programm die PiCar-X anweisen, fünf Aktionen in folgender Reihenfolge auszuführen: „Vorwärts“, „Rückwärts“, „Links abbiegen“, „Rechts abbiegen“ und „Stoppen“.

Um die grundlegende Bedienung von Ezbloc Studio zu erlernen, lesen Sie bitte die folgenden beiden Abschnitte:

- [How to Create a New Project?](#)



TIPPS

forward at a speed of 50

Dieser Block veranlasst die PiCar-X, mit einer Geschwindigkeit vorwärts zu fahren, die sich an einem Prozentsatz der verfügbaren Leistung orientiert. Im untenstehenden Beispiel bedeutet „50“, dass die Geschwindigkeit bei 50 % der Leistung oder Halbgeschwindigkeit liegt.

backward at a speed of 50

Dieser Block veranlasst die PiCar-X, mit einer an einem Prozentsatz der verfügbaren Leistung orientierten Geschwindigkeit rückwärts zu fahren.

turn steering angle to 0

Dieser Block passt die Ausrichtung der Vorderräder an. Der Bereich liegt bei „-45“ bis „45“. Im untenstehenden Beispiel bedeutet „-30“, dass die Räder um 30° nach links drehen werden.



Dieser Block bewirkt eine zeitliche Verzögerung zwischen den Befehlen, basierend auf Millisekunden. Im untenstehenden Beispiel wartet die PiCar-X 1 Sekunde (1000 Millisekunden) bevor der nächste Befehl ausgeführt wird.



Dieser Block bringt die PiCar-X zum vollständigen Anhalten.

BEISPIEL

Bemerkung:

- Sie können das Programm entsprechend dem folgenden Bild schreiben. Bitte beachten Sie das Tutorial: [How to Create a New Project?](#).
 - Oder suchen Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite von EzBlock Studio und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.
-

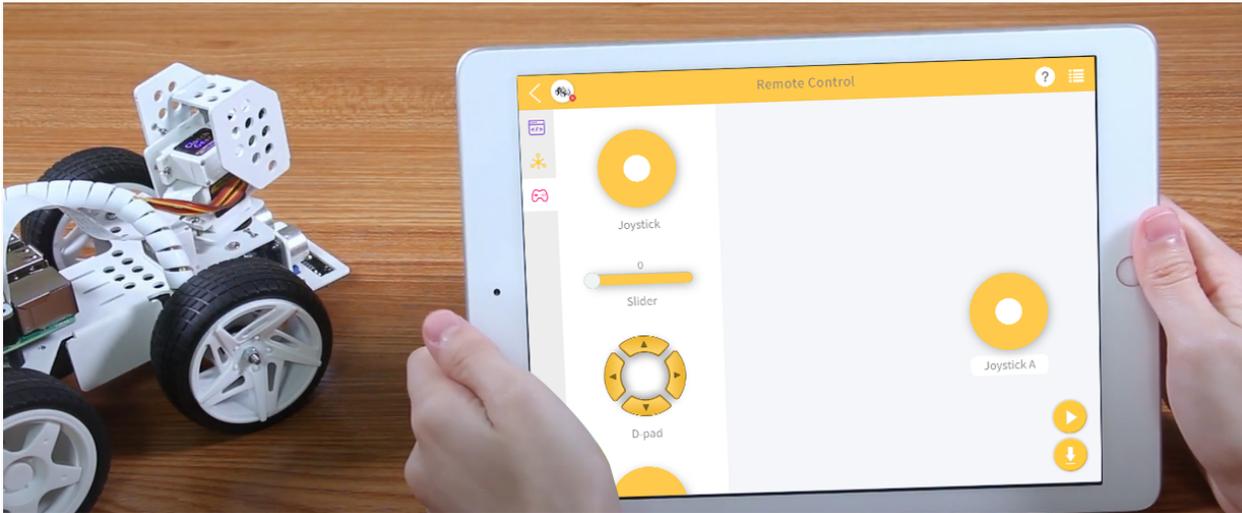


```
Forever
  forward at 50 % speed
  delay 1000
  backward at 50 % speed
  delay 1000
  forward at 50 % speed
  turn steering angle to -30
  delay 1000
  forward at 50 % speed
  turn steering angle to 30
  delay 1000
  turn steering angle to 0
  stop
  delay 2000
```

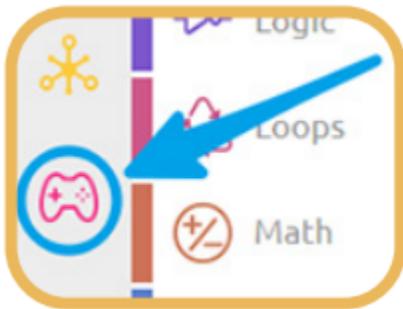
6.5 Fernsteuerung

Dieses Projekt zeigt, wie Sie die PiCar-X mithilfe des Joystick-Widgets fernsteuern können. Hinweis: Nachdem Sie das Joystick-Widget von der Fernsteuerungsseite gezogen und abgelegt haben, verwenden Sie die „Map“-Funktion, um die X- und Y-Achsen-Werte des Joysticks zu kalibrieren. Weitere Informationen zur Fernsteuerungsfunktion finden Sie unter folgendem Link:

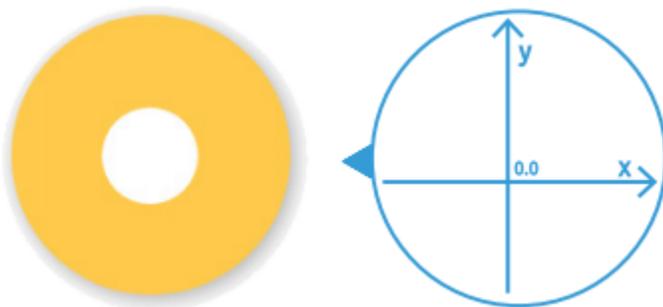
- [How to Use the Remote Control Function?](#)



TIPPS



Um die Fernsteuerungsfunktion zu nutzen, öffnen Sie die Fernsteuerungsseite über die linke Seite der Hauptseite.

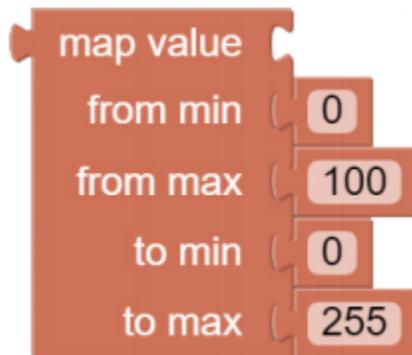


Ziehen Sie einen Joystick in den mittleren Bereich der Fernsteuerungsseite. Durch Verschieben des weißen Punktes in der Mitte und sanftes Ziehen in eine beliebige Richtung werden ein (X, Y)-Koordinatenpaar erzeugt. Der Bereich der

X- oder Y-Achse ist standardmäßig von „-100“ bis „100“ festgelegt. Verschieben Sie den weißen Punkt und ziehen Sie ihn direkt zum äußersten linken Rand des Joysticks, ergibt dies einen X-Wert von „-100“ und einen Y-Wert von „0“.



Nachdem ein Widget auf der Fernsteuerungsseite gezogen und abgelegt wurde, erscheint eine neue Kategorie - Remote - mit dem obigen Block. Dieser Block liest den Joystick-Wert auf der Fernsteuerungsseite. Im Dropdown-Menü können Sie auf die Y-Achsen-Lesung umschalten.

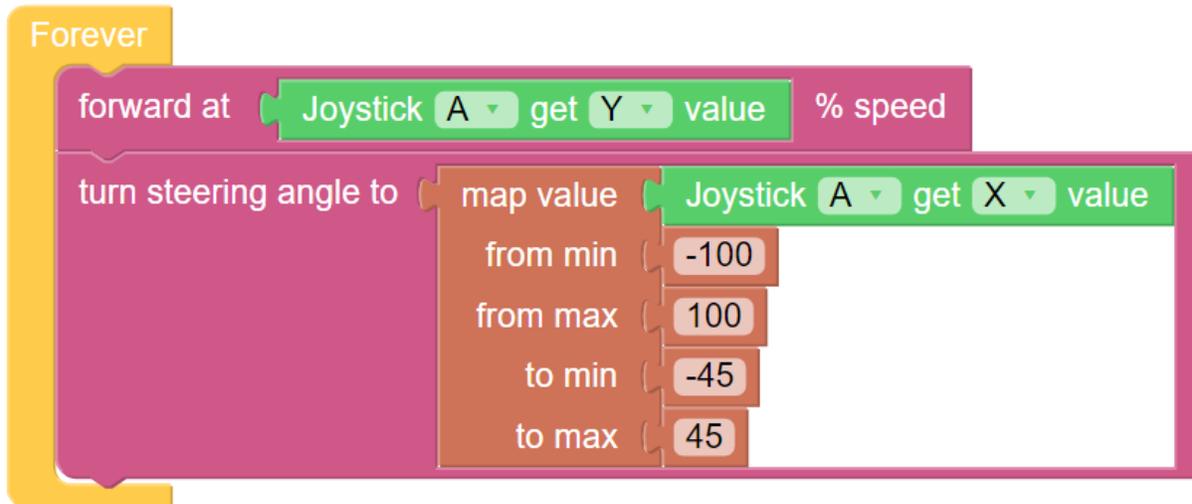


Der „Map Value“-Block kann eine Zahl von einem Bereich in einen anderen umkodieren. Wenn der Bereich von 0 bis 100 festgelegt ist und die „Map Value“-Nummer 50 beträgt, dann entspricht das einer Position von 50 % des Bereichs oder „50“. Wenn der Bereich von 0 bis 255 festgelegt ist und die „Map Value“-Nummer 50 beträgt, dann entspricht dies einer Position von 50 % des Bereichs oder „127,5“.

BEISPIEL

Bemerkung:

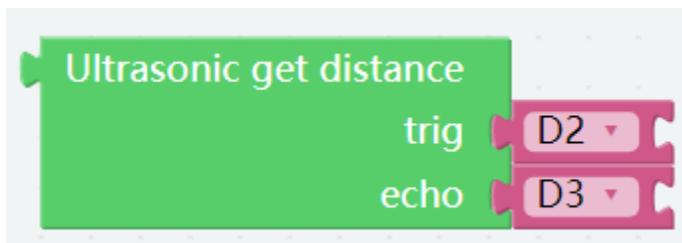
- Sie können das Programm gemäß dem folgenden Bild erstellen. Bitte beziehen Sie sich auf das Tutorial: [How to Create a New Project?](#).
 - Oder finden Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite des EzBlock Studios und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.
-



6.6 Test des Ultraschallmoduls

Der PiCar-X verfügt über ein integriertes Ultraschall-Sensormodul, das für Hindernisvermeidung und automatische Objektverfolgungsexperimente verwendet werden kann. In dieser Lektion wird das Modul eine Entfernung in Zentimetern messen (24 cm = 1 Zoll) und die Ergebnisse in einem **Debug-Fenster ausgeben**.

TIPPS



Der Block **Ultraschall Entfernung ablesen** ermittelt die Entfernung von PiCar-X zu einem direkt vorausliegenden Hindernis.



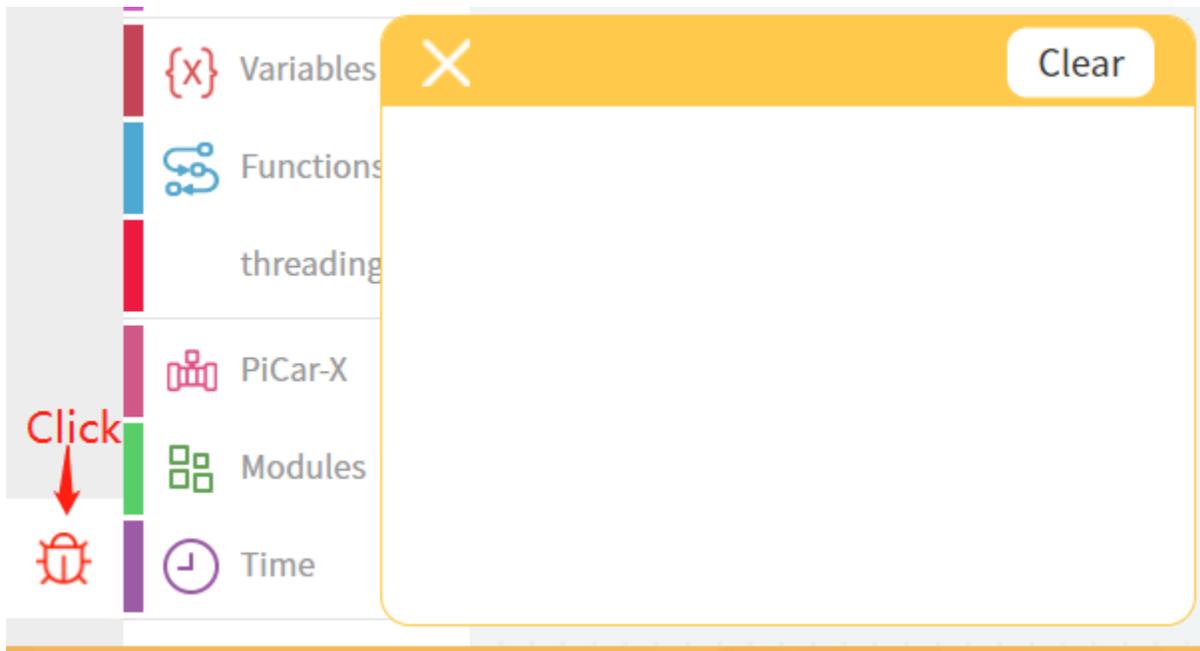
Dieses Programm wird durch eine **Variable** vereinfacht. Wenn beispielsweise mehrere Funktionen in einem Programm jeweils die Entfernung zu einem Hindernis nutzen müssen, kann eine **Variable** verwendet werden, um den gleichen Entfernungswert an jede Funktion zu übermitteln, statt dass jede Funktion den Wert separat ausliest.

Create variable...

Klicken Sie auf die Schaltfläche **Variable erstellen...** in der Kategorie **Variablen** und verwenden Sie den Dropdown-Pfeil, um die Variable mit dem Namen „Entfernung“ auszuwählen.



Die **Ausgabefunktion** kann Daten wie Variablen und Text zur einfachen Fehlersuche ausgeben.

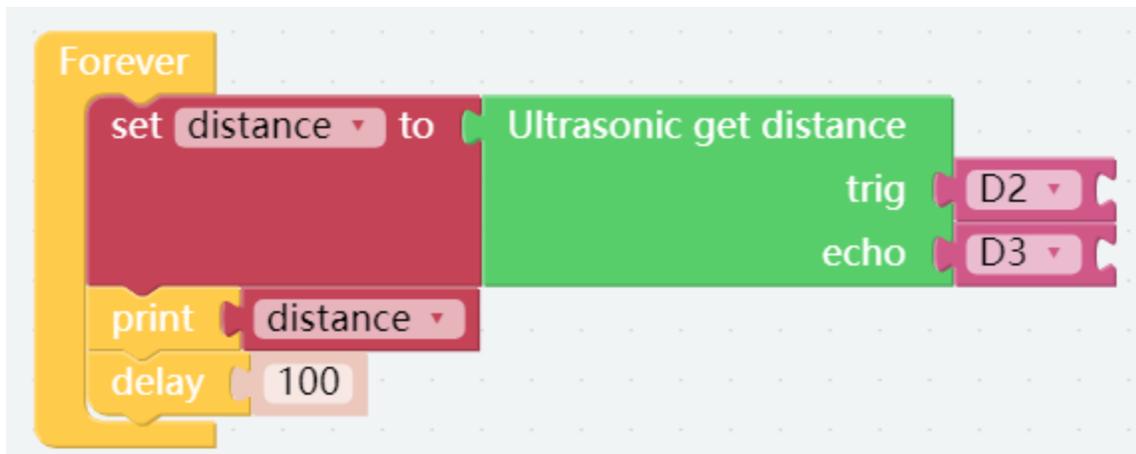


Sobald der Code läuft, aktivieren Sie den Debug-Monitor, indem Sie auf das **Debug**-Symbol in der unteren linken Ecke klicken.

BEISPIEL

Bemerkung:

- Sie können das Programm gemäß dem folgenden Bild erstellen. Bitte beziehen Sie sich auf das Tutorial: [How to Create a New Project?](#).
- Oder finden Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite des EzBlock Studios und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.



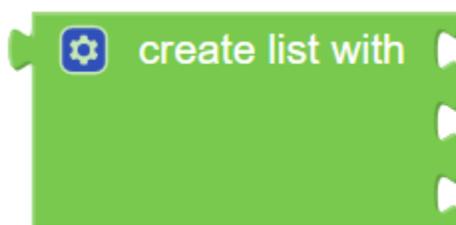
6.7 Test des Graustufenmoduls

Der PiCar-X beinhaltet ein Graustufenmodul, das sich für Anwendungen wie Linienverfolgung, Abgrund-Erkennung und weitere spannende Experimente eignet. Das Graustufenmodul besitzt drei Detektionssensoren, die jeweils einen Wert entsprechend der erkannten Farbschattierung zurückgeben. Zum Beispiel wird ein Sensor, der reines Schwarz erkennt, den Wert „0“ zurückgeben.

TIPPS



Verwenden Sie den Block **Graustufenmodul** um den Wert eines der Sensoren auszulesen. Im obigen Beispiel ist der Sensor „A0“ der Sensor ganz links am PiCar-X. Verwenden Sie den Dropdown-Pfeil, um den Sensor auf „A1“ (Mittelsensor) oder „A2“ (rechter Sensor) zu wechseln.



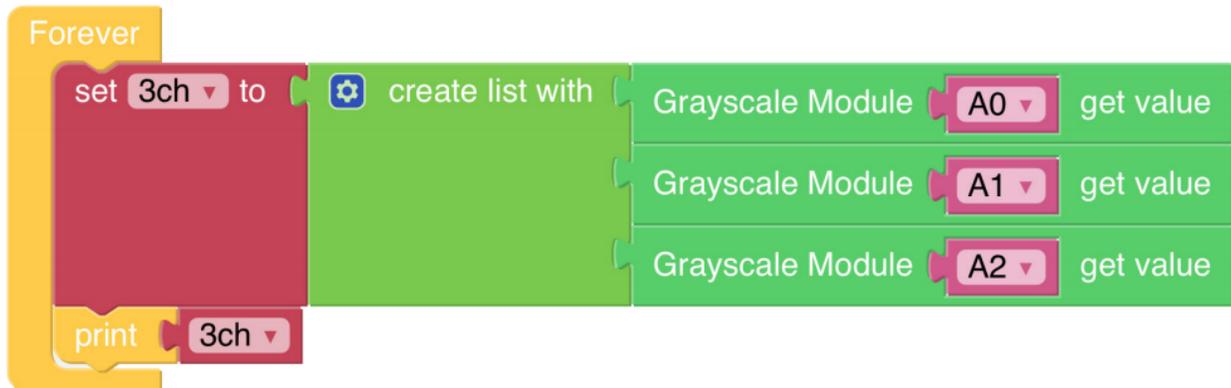
Das Programm wird mit einem Block **Liste erstellen mit** vereinfacht. Eine **Liste** wird ähnlich wie eine einzelne **Variable** verwendet, ist in diesem Fall jedoch effizienter als eine einzelne **Variable**, da das **Graustufenmodul** mehrere Sensorwerte zurückmeldet. Der Block **Liste erstellen mit** wird separate **Variablen** für jeden Sensor anlegen und sie in einer Liste zusammenfassen.

BEISPIEL

Bemerkung:

- Sie können das Programm gemäß dem folgenden Bild erstellen. Bitte beziehen Sie sich auf das Tutorial: [How to Create a New Project?](#).

- Oder suchen Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite des EzBlock Studios und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.
-



6.8 Farberkennung

Der PiCar-X ist ein selbstfahrendes Auto mit einer integrierten Kamera, die es ermöglicht, Ezblock-Programme für Objekterkennung und Farberkennung zu nutzen. In diesem Abschnitt wird Ezblock verwendet, um ein Programm für die Farberkennung zu erstellen.

Bemerkung: Stellen Sie sicher, dass das Flachbandkabel der Raspberry Pi-Kamera ordnungsgemäß und sicher angeschlossen ist, bevor Sie diesen Abschnitt beginnen. Für detaillierte Anweisungen zum sicheren Anschluss des FFC-Kabels siehe: [Montageanleitung](#).

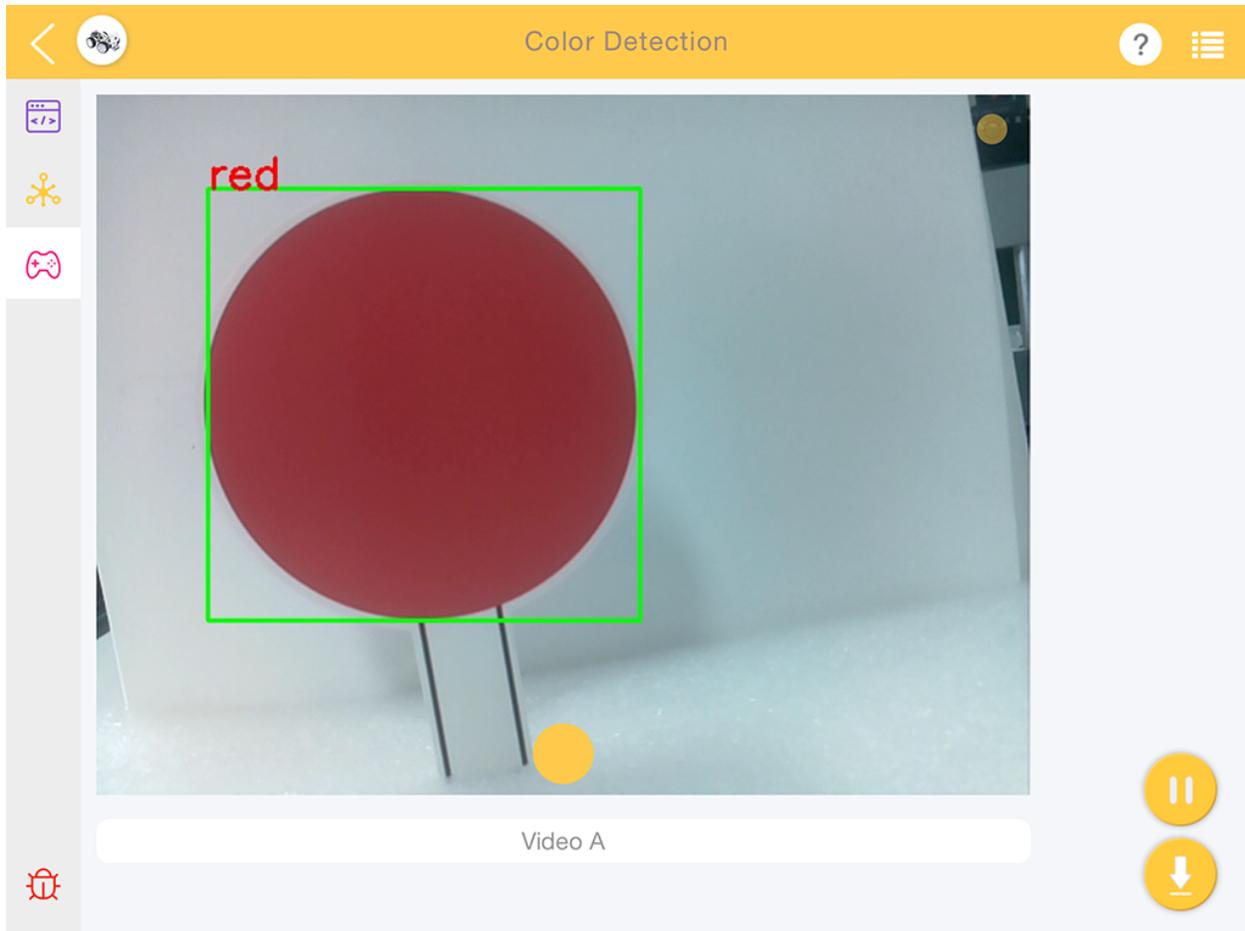
In diesem Programm wird Ezblock zunächst der Farbraum Hue-Saturation-Value (HSV) der zu erkennenden Farbe mitgeteilt. Anschließend wird OpenCV verwendet, um die Farben im HSV-Bereich zu verarbeiten, den Hintergrund rauschen zu entfernen und schließlich die passende Farbe einzurahmen.

Für den PiCar-X bietet Ezblock sechs Farbmodelle: „red“, „orange“, „yellow“, „green“, „blue“, „purple“ („rot“, „orange“, „gelb“, „grün“, „blau“ und „lila“). Farbkarten sind im folgenden PDF vorbereitet und müssen mit einem Farbdrukker ausgedruckt werden.

- [PDF] Farbkarten



Bemerkung: Aufgrund von Unterschieden im Druckertoner oder dem bedruckten Medium, wie zum Beispiel sandfarbenem Papier, können die ausgedruckten Farben einen leicht abweichenden Farbton im Vergleich zu den Ezblock-Farbmodellen aufweisen. Dies kann zu einer weniger genauen Farberkennung führen.



TIPPS



Ziehen Sie das Video-Widget von der Fernsteuerungsseite, um einen Videomonitor zu erzeugen. Für weitere Informationen zur Verwendung des Video-Widgets siehe das Tutorial zu Ezblock-Video hier: [How to Use the Video Function?](#).



Aktivieren Sie den Videomonitor, indem Sie den Block **Kameramonitor** auf **ein** setzen. Hinweis: Wenn Sie den **Kameramonitor** auf **aus** setzen, wird der Monitor geschlossen, aber die Objekterkennung bleibt weiterhin verfügbar.



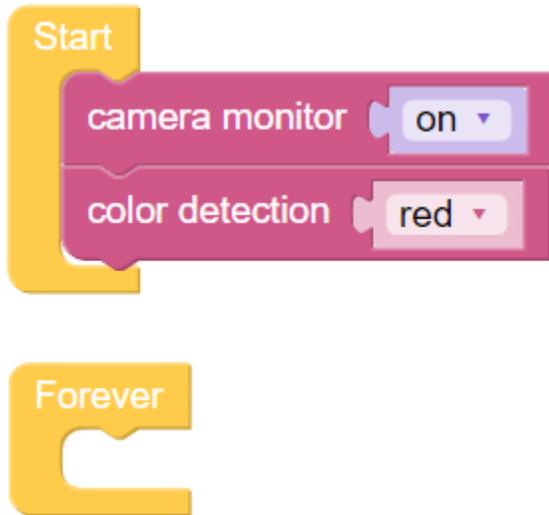
Verwenden Sie den Block **Farberkennung**, um die Farberkennung zu aktivieren. Hinweis: Es kann immer nur eine

Farbe gleichzeitig erkannt werden.

BEISPIEL

Bemerkung:

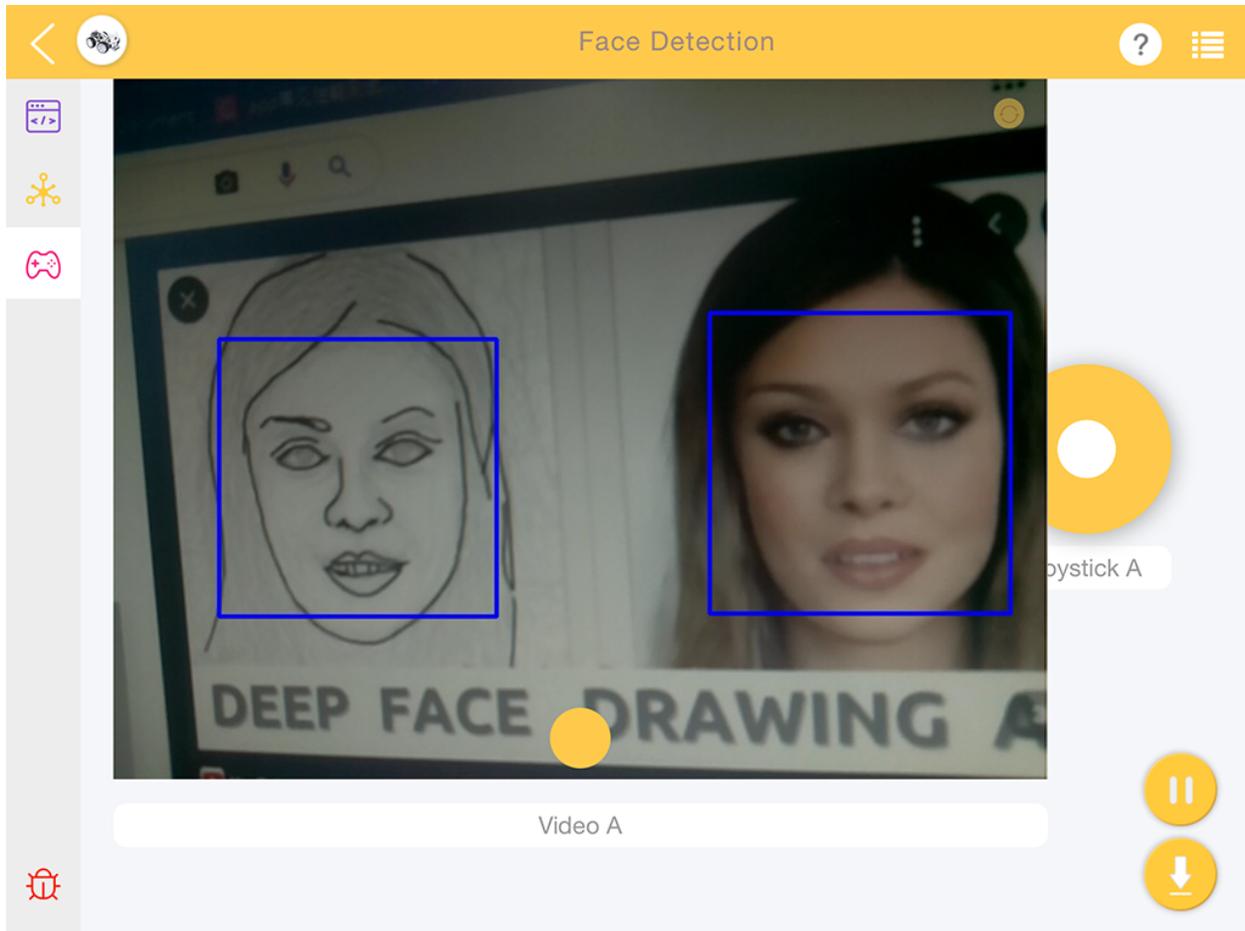
- Sie können das Programm gemäß dem folgenden Bild erstellen. Bitte beziehen Sie sich auf das Tutorial: [How to Create a New Project?](#).
- Oder suchen Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite des EzBlock Studios und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.



6.9 Gesichtserkennung

Neben der Farberkennung bietet der PiCar-X auch eine Funktion zur Gesichtserkennung. Im folgenden Beispiel wird das Joystick-Widget zur Steuerung der Kamerarichtung verwendet, und die Anzahl der erkannten Gesichter wird im Debug-Monitor angezeigt.

Für weitere Informationen zur Verwendung des Video-Widgets verweisen wir auf das Tutorial zu Ezblock-Video: [How to Use the Video Function?](#).



TIPPS

face detection on ▼

Aktivieren Sie das **Gesichtserkennung**-Widget, um die Gesichtserkennungsfunktion zu nutzen.

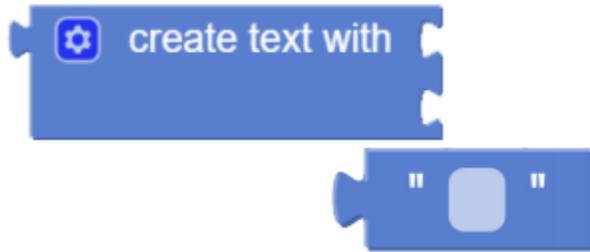
turn camera pan angle to 0

turn camera tilt angle to 0

Diese beiden Blöcke dienen zur Anpassung der Ausrichtung der Schwenk-Neige-Kamera, ähnlich wie bei der Steuerung des PiCar-X im Tutorial *Fernsteuerung*. Ein steigender Wert lässt die Kamera nach rechts oder oben schwenken, ein sinkender Wert nach rechts oder unten.

number ▼ of detected face

Die Ergebnisse der Bilderkennung werden über den **detected face**-Block zurückgegeben. Verwenden Sie die Dropdown-Menüoptionen, um zwischen der Ausgabe der Koordinaten, der Größe oder der Anzahl der Ergebnisse aus der Bilderkennungsfunktion zu wählen.

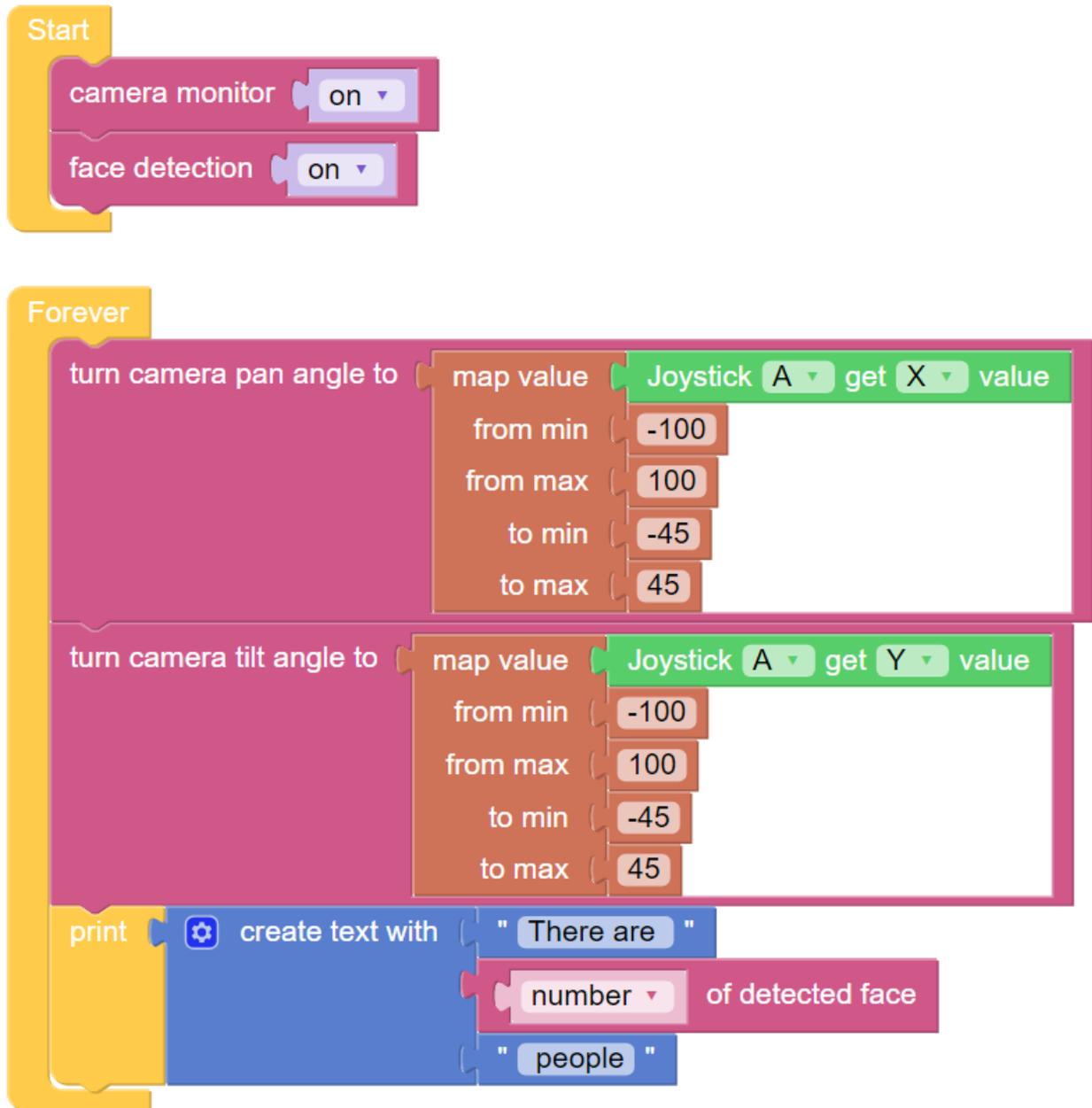


Verwenden Sie den **create text with**-Block, um die Kombination aus **Text** und **detected face**-Daten auszugeben.

BEISPIEL

Bemerkung:

- Das Programm kann gemäß dem folgenden Bild erstellt werden. Bitte beziehen Sie sich auf das Tutorial: [How to Create a New Project?](#).
 - Oder finden Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite des EzBlock Studios und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.
-



6.10 Soundeffekte

Der PiCar-X verfügt über einen integrierten Lautsprecher, der für Audio-Experimente genutzt werden kann. Mit Ezblock können Nutzer Text eingeben, um den PiCar-X sprechen zu lassen oder bestimmte Soundeffekte zu erzeugen. In diesem Tutorial wird der PiCar-X nach einem 3-Sekunden-Countdown das Geräusch eines Schussabfeuers mittels einer do/while-Funktion erzeugen.

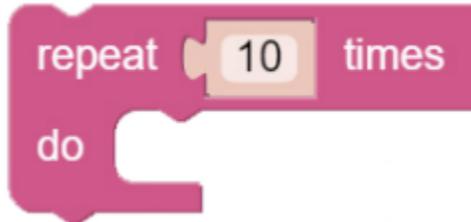
TIPPS



Verwenden Sie den **sagen**-Block zusammen mit einem **Text**-Block, um einen Satz für den PiCar-X zu formulieren. Der **sagen**-Block kann sowohl für Text als auch für Zahlen verwendet werden.



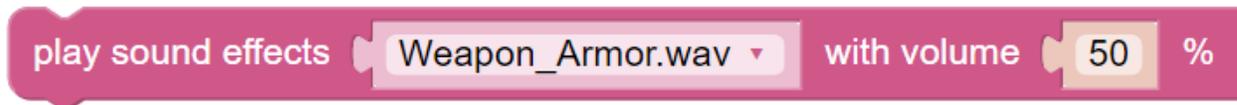
Der **Zahlen**-Block.



Mit dem **wiederholen**-Block wird dieselbe Anweisung wiederholt ausgeführt, was den Code verkürzt.



Der **mathematische Operationen**-Block kann übliche mathematische Funktionen ausführen, wie „+“, „-“, „x“ und „÷“.

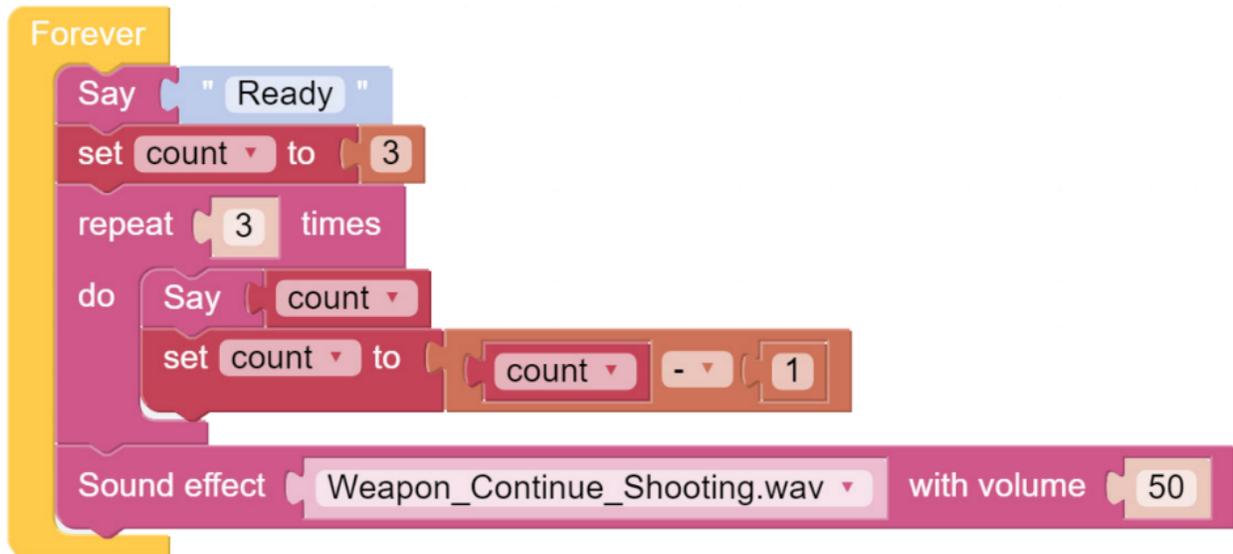


Der Block **Soundeffekte abspielen - mit Lautstärke - %** bietet voreingestellte Soundeffekte wie etwa ein Sirengeräusch, ein Schussgeräusch und andere. Der Lautstärkebereich kann von 0 bis 100 eingestellt werden.

BEISPIEL

Bemerkung:

- Sie können das Programm gemäß dem folgenden Bild schreiben. Bitte beziehen Sie sich auf das Tutorial: [How to Create a New Project?](#).
- Oder finden Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite des EzBlock Studios und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.



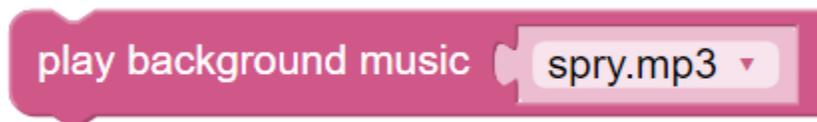
6.11 Hintergrundmusik

Neben der Programmierung des PiCar-X, um Soundeffekte oder Text-to-Speech (TTS) abzuspielen, kann der PiCar-X auch Hintergrundmusik wiedergeben. In diesem Projekt wird außerdem ein **Schieberegler**-Widget zur Anpassung der Musiklautstärke verwendet.

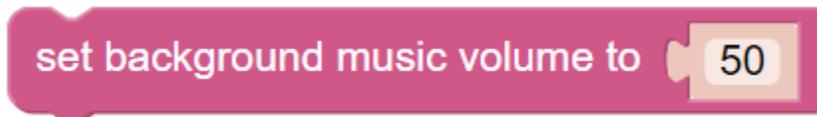
- [How to Use the Remote Control Function?](#)

Für ein detailliertes Tutorial zu den Fernsteuerungsfunktionen von Ezblock verweisen wir auf das *Fernsteuerung* Tutorial.

TIPPS



Der **Hintergrundmusik abspielen**-Block muss der **Start**-Funktion hinzugefügt werden. Verwenden Sie das Dropdown-Menü, um unterschiedliche Hintergrundmusiken für den PiCar-X auszuwählen.



Mit dem Block **Hintergrundmusik-Lautstärke einstellen auf** wird die Lautstärke im Bereich von 0 bis 100 angepasst.



Ziehen Sie eine **Schieberegler**-Leiste von der **Fernsteuerung**-Seite, um die Musiklautstärke anzupassen.



Der **Schieberegler [A] Wert abrufen**-Block liest den Wert des Schiebereglers aus. Im obigen Beispiel ist Schieberegler ‚A‘ ausgewählt. Wenn es mehrere Schieberegler gibt, verwenden Sie das Dropdown-Menü, um den entsprechenden auszuwählen.

BEISPIEL

Bemerkung:

- Sie können das Programm gemäß dem folgenden Bild schreiben. Bitte beziehen Sie sich auf das Tutorial: [How to Create a New Project?](#).
- Oder finden Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite des EzBlock Studios und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.



6.12 Sag Hallo

Dieses Projekt kombiniert mehrere Funktionen aus den vorangegangenen Projekten. Die Bewegung des PiCar-X wird ferngesteuert, und die Kamera des PiCar wird ebenfalls durch zwei Joystick-Controller ferngesteuert. Erkennt das PiCar das Gesicht einer Person, wird es höflich nicken und „Hallo!“ sagen.

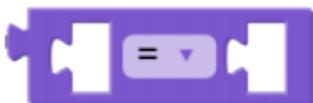
- [How to Use the Video Function?](#)
- [How to Use the Remote Control Function?](#)



TIPPS



Der **if do**-Block wird verwendet, um einmal höflich zu nicken, sobald die bedingte Überprüfung von „if“ wahr ist.

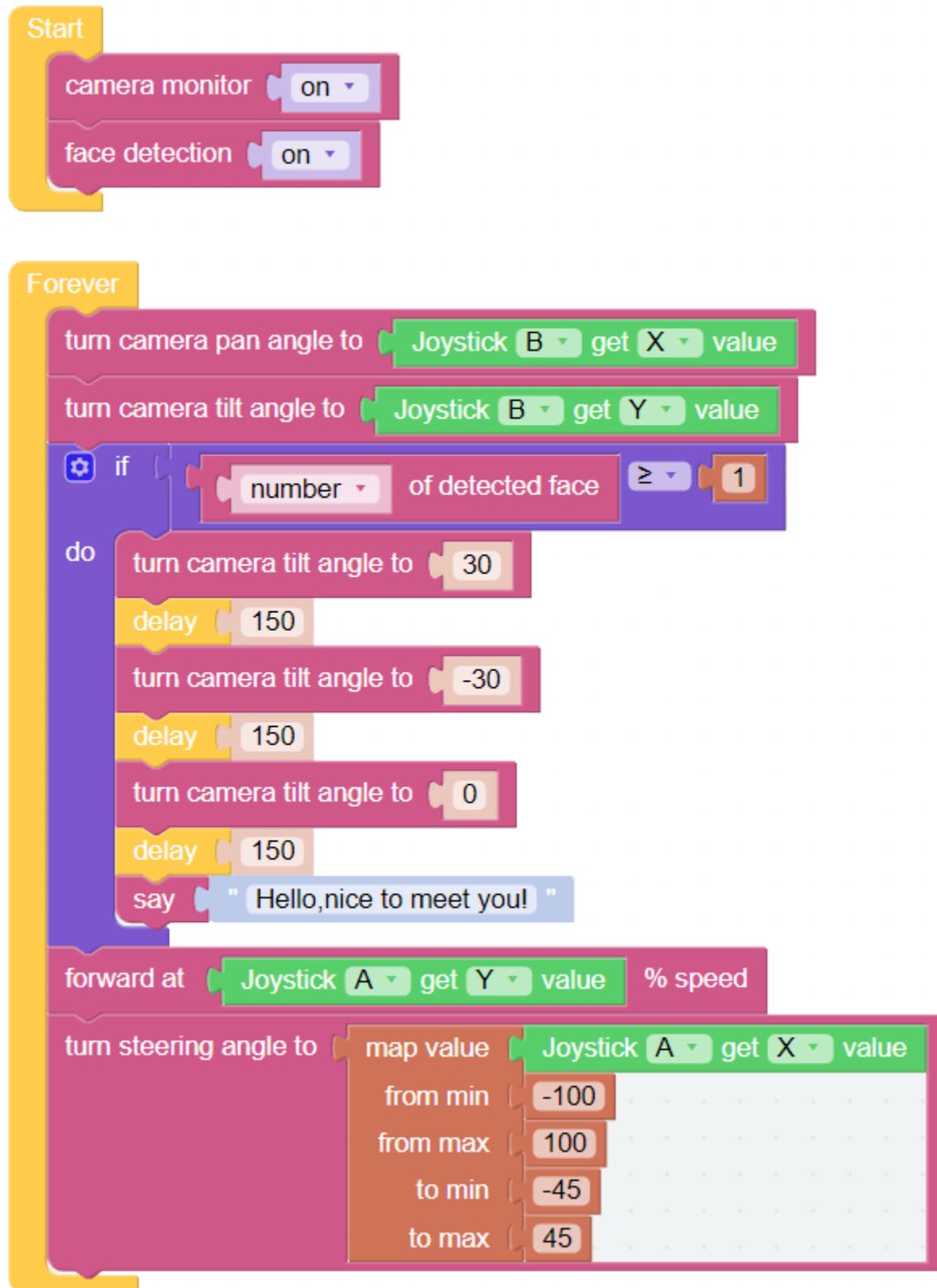


Der **Bedingungen**-Block wird zusammen mit dem **if do**-Block verwendet. Die Bedingungen können „=“, „>“, „<“, „>“, „<“ oder „“ sein.

BEISPIEL

Bemerkung:

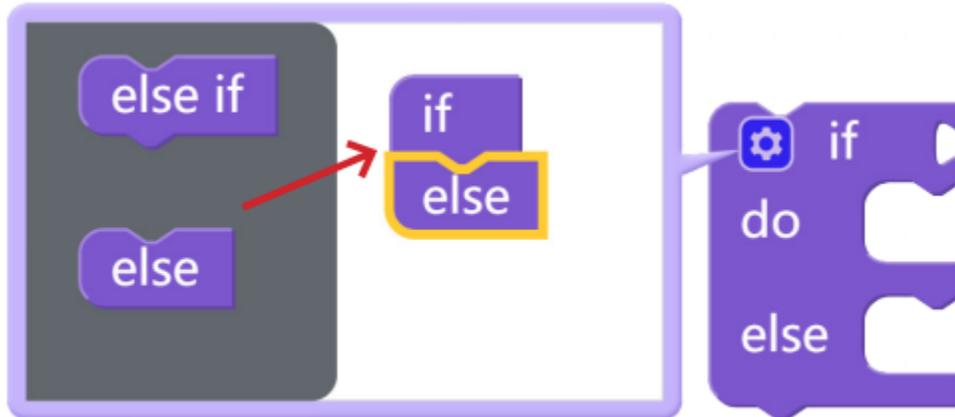
- Sie können das Programm gemäß dem folgenden Bild erstellen, bitte beziehen Sie sich dazu auf das Tutorial: [How to Create a New Project?](#).
- Oder finden Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite des EzBlock Studios und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.



6.13 Musikauto

Dieses Projekt verwandelt den PiCar-X in ein Musikauto, das durch Ihr Zuhause fährt und dabei fröhliche Musik abspielt. Außerdem wird gezeigt, wie der PiCar-X mithilfe des integrierten Ultraschallsensors Kollisionen mit Wänden vermeidet.

TIPPS

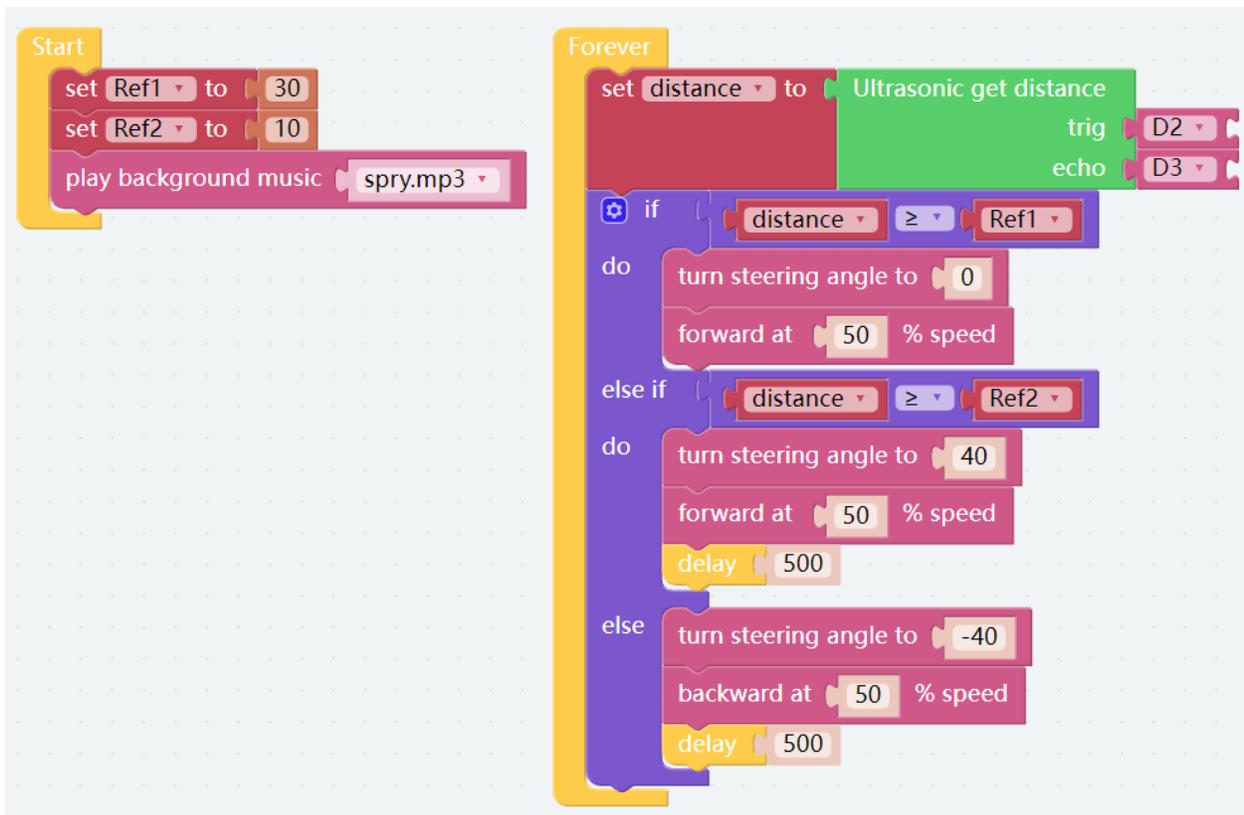


Um mehrere bedingte Abfragen durchzuführen, ändern Sie den einfachen „if do“-Block in einen „if else do / else if do“-Block. Dies erfolgt durch Klicken auf das Einstellungssymbol, wie oben gezeigt.

BEISPIEL

Bemerkung:

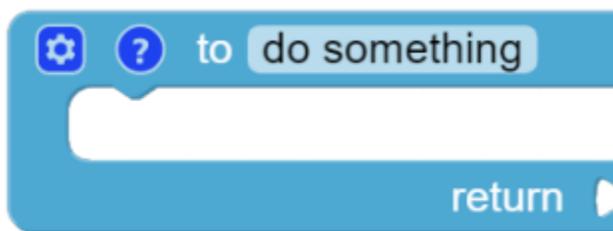
- Sie können das Programm gemäß dem folgenden Bild erstellen. Bitte beziehen Sie sich dazu auf das Tutorial: [How to Create a New Project?](#).
 - Oder suchen Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite des EzBlock Studios und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.
-



6.14 Abgrund-Erkennung

Dieses Projekt nutzt das **Grauskalierungsmodul**, um zu verhindern, dass der PiCar-X in Ihrem Zuhause von einer Klippe oder einer Treppe stürzt. Dies ist ein unverzichtbares Projekt für Häuser mit Treppen.

TIPPS

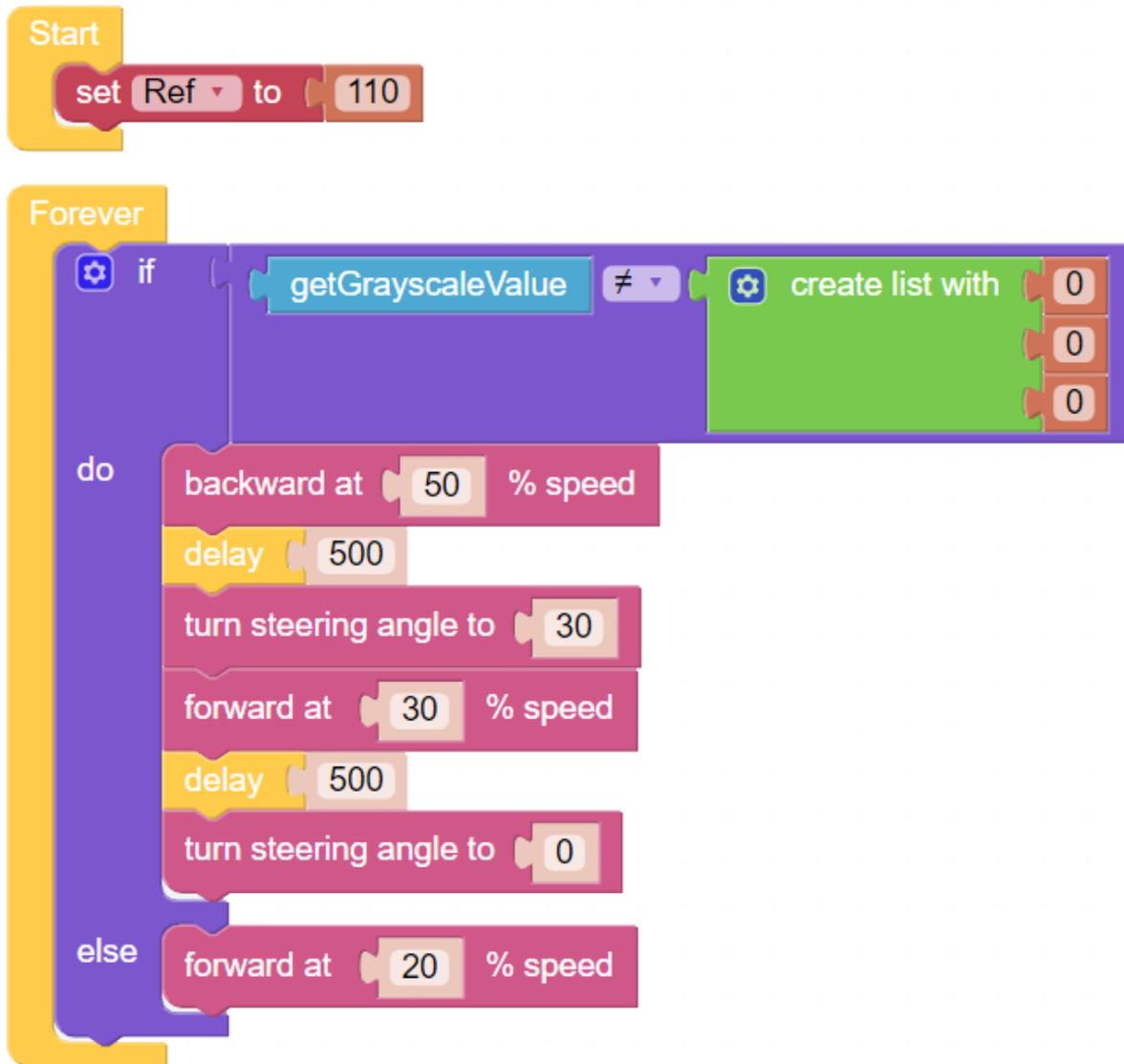


Das **Grauskalierungsmodul** wird dieselbe Operation mehrmals ausführen. Um das Programm zu vereinfachen, führt dieses Projekt eine **Funktion** ein, die dem **Endlosschleifen-Block** eine **Liste** als Variable zurückgibt.

BEISPIEL

Bemerkung:

- Sie können das Programm gemäß dem folgenden Bild erstellen. Bitte beziehen Sie sich dazu auf das Tutorial: [How to Create a New Project?](#).
- Oder suchen Sie den Code mit dem gleichen Namen auf der **Beispiele**-Seite des EzBlock Studios und klicken Sie direkt auf **Ausführen** oder **Bearbeiten**.





6.15 Minenwagen

Lass uns ein Minenwagen-Projekt machen! In diesem Projekt verwenden wir das Grayscale-Modul, um den PiCar-X entlang einer Strecke vorwärts zu bewegen. Verwende dunkel gefärbtes Klebeband, um eine möglichst gerade und nicht zu kurvig verlaufende Strecke auf dem Boden zu markieren. Es kann sein, dass einige Experimente erforderlich sind, falls der PiCar-X entgleist.

Beim Fahren entlang der Strecke werden die Sonden auf der linken und rechten Seite des Grayscale-Moduls helle Bodenflächen erkennen, während die mittlere Sonde die Strecke erkennt. Wenn die Strecke einen Bogen hat, wird die Sonde auf der linken oder rechten Seite des Sensors das dunkel gefärbte Klebeband erkennen und die Räder in diese Richtung lenken. Wenn der Minenwagen das Ende der Strecke erreicht oder entgleist, erkennt das Grayscale-Modul das dunkel gefärbte Klebeband der Strecke nicht mehr und der PiCar-X bleibt stehen.

TIPPS

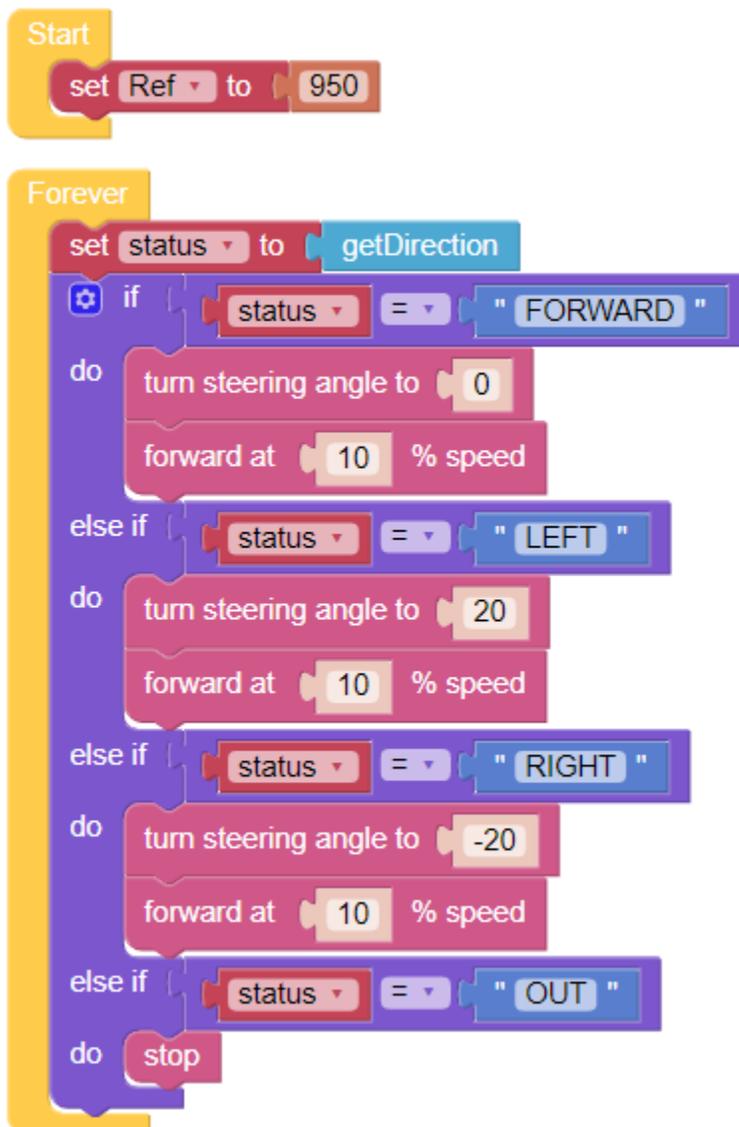
- Der Block **Set ref to ()** wird verwendet, um den Graustufen-Schwellenwert einzustellen. Du musst ihn entspre-

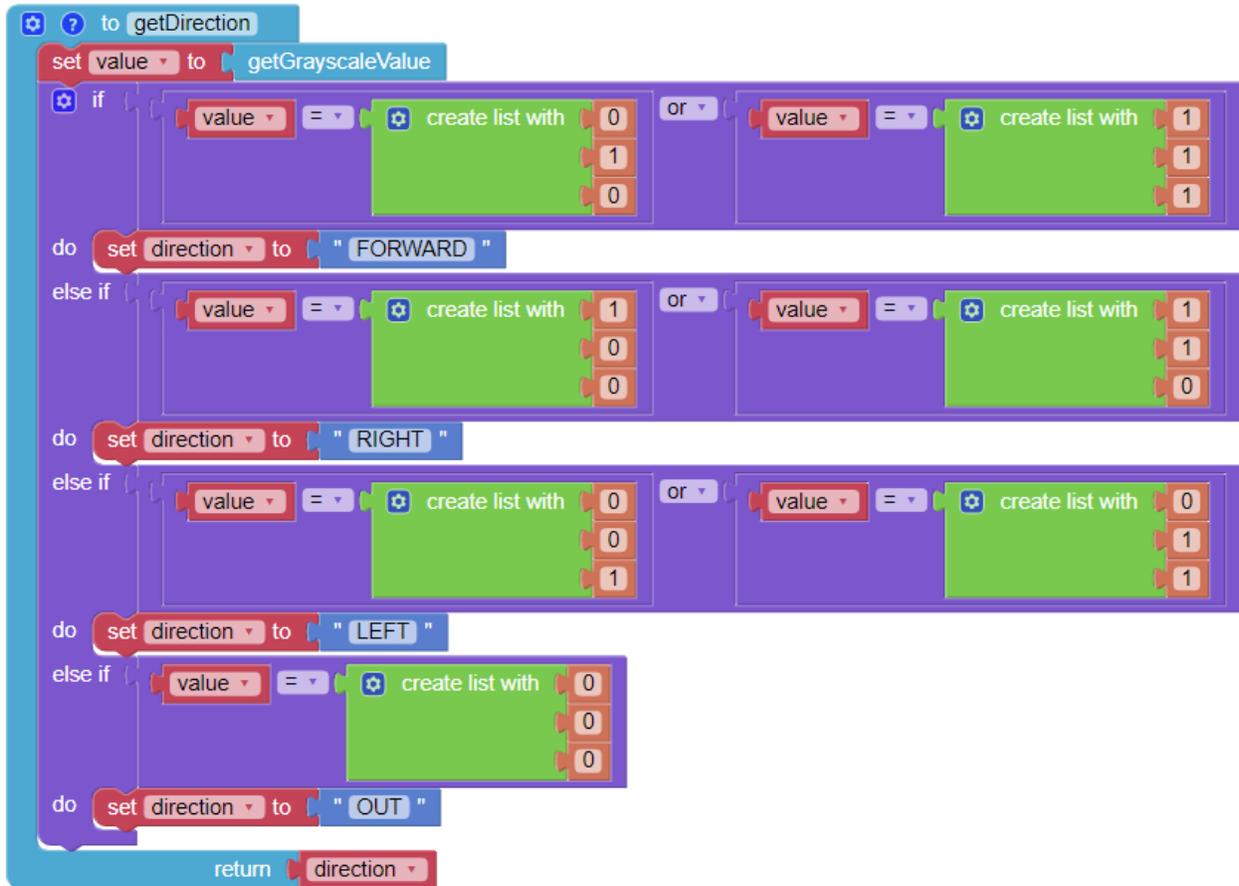
chend der tatsächlichen Situation anpassen. Du kannst *Test des Graustufenmoduls* ausführen, um die Werte des Graustufen-Moduls auf weißen und schwarzen Flächen zu sehen, und deren Mittelwerte in diesem Block eintragen.

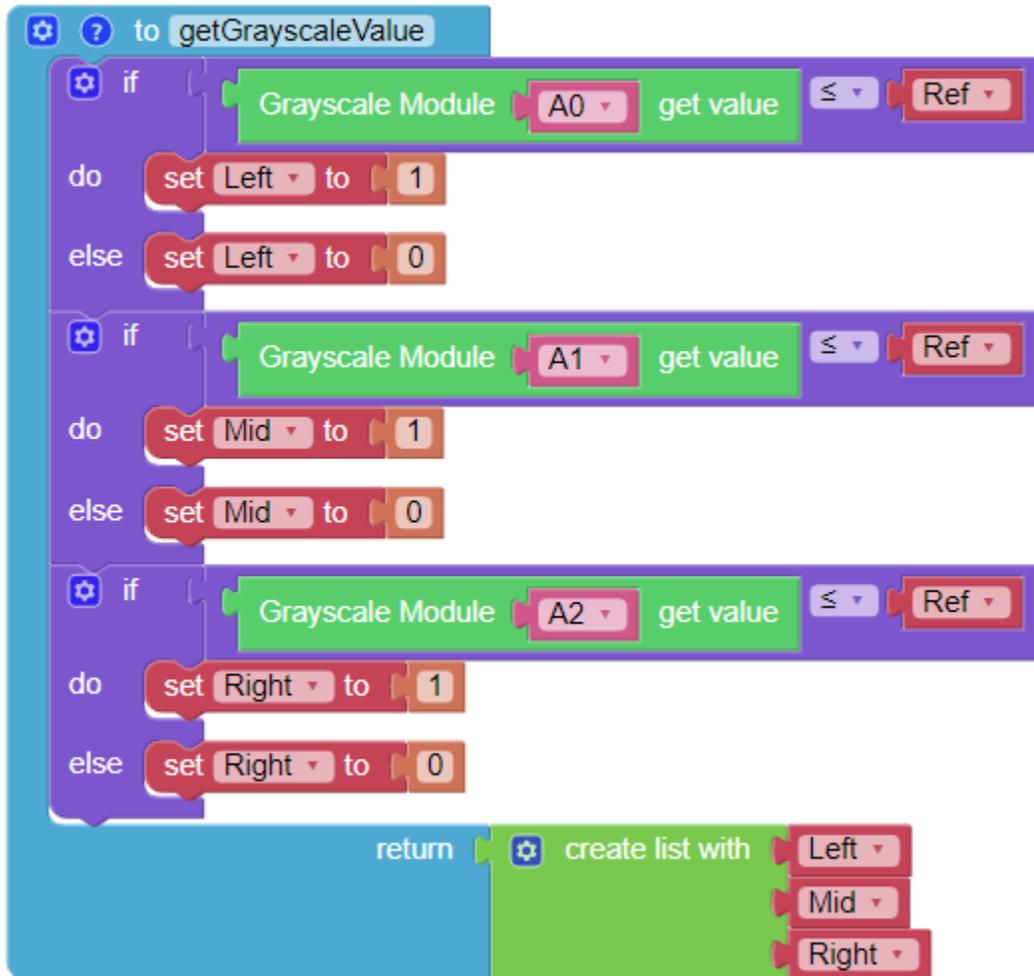
BEISPIEL

Bemerkung:

- Du kannst das Programm entsprechend dem folgenden Bild schreiben. Bitte beachte das Tutorial: [How to Create a New Project?](#).
- Oder finde den Code mit demselben Namen auf der **Examples**-Seite des EzBlock Studio und klicke direkt auf **Run** oder **Edit**.

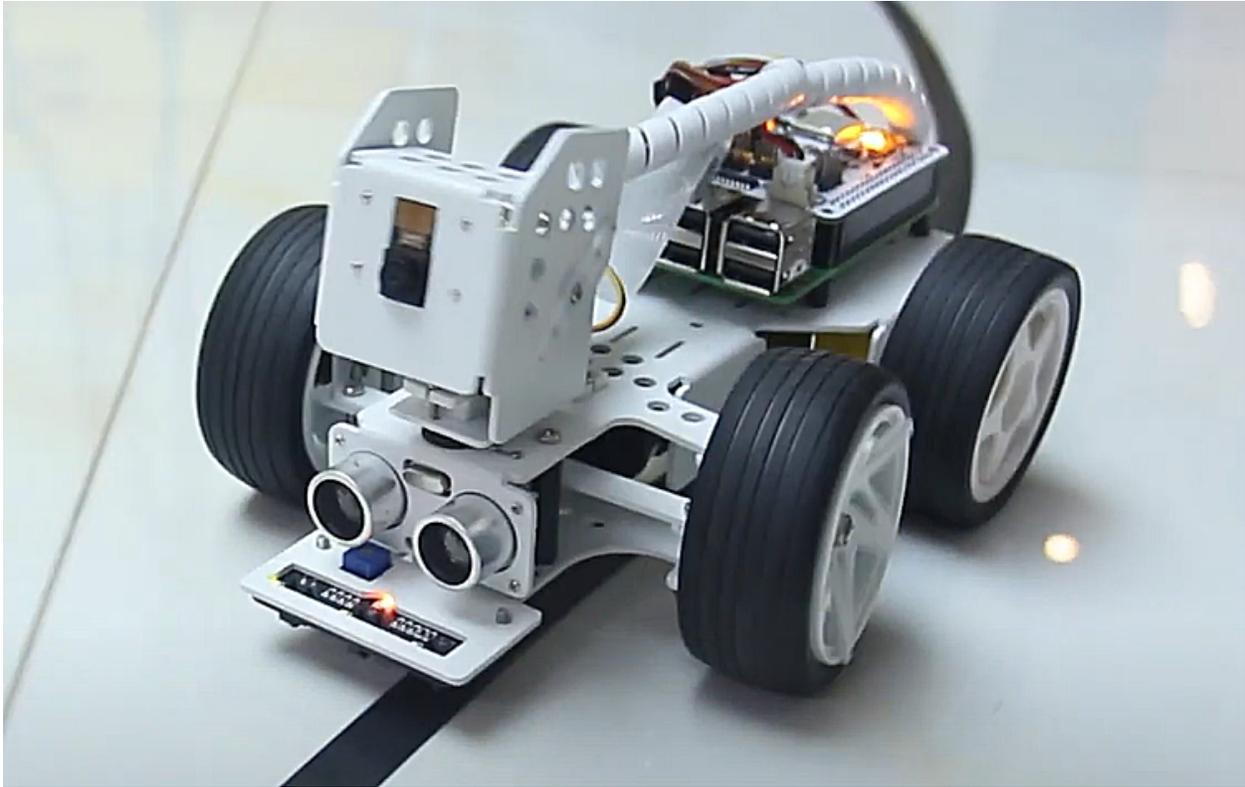






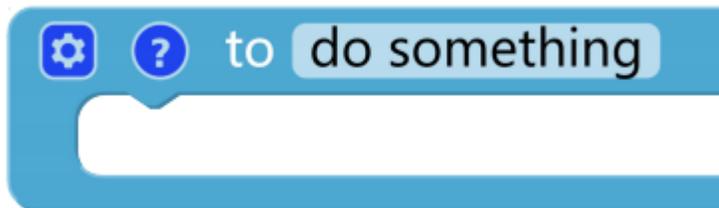
6.16 Minenwagen Plus

In diesem Projekt wurde dem *Minenwagen*-Projekt eine Entgleisungswiederherstellung hinzugefügt, um dem PiCar-X die Anpassung und Wiederherstellung bei stärkeren Kurven zu ermöglichen.



TIPPS

1. Verwende einen weiteren **to do something** -Block, um dem PiCar-X das Zurücksetzen und die Wiederherstellung nach einer scharfen Kurve zu ermöglichen. Beachte, dass die neue **to do something** -Funktion keine Werte zurückgibt, sondern ausschließlich zur Neuausrichtung des PiCar-X verwendet wird.

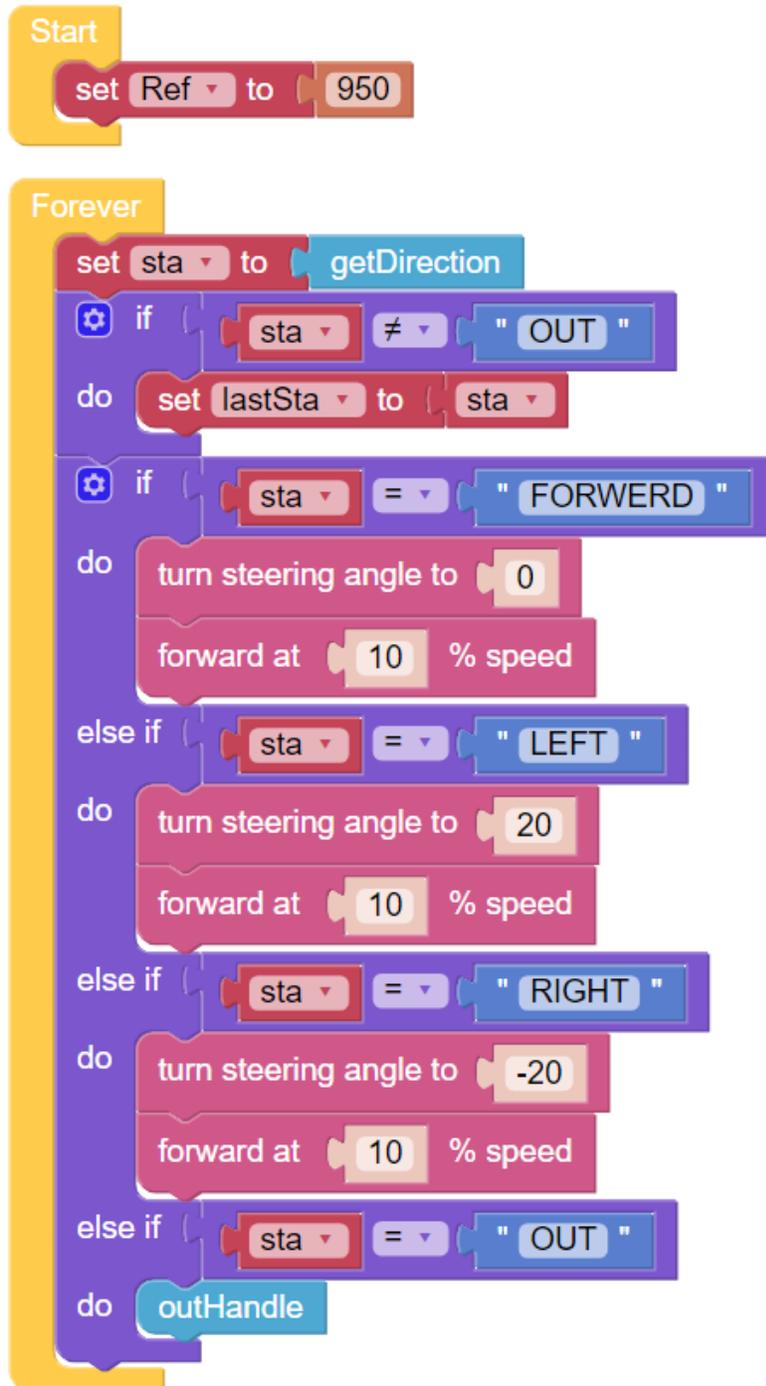


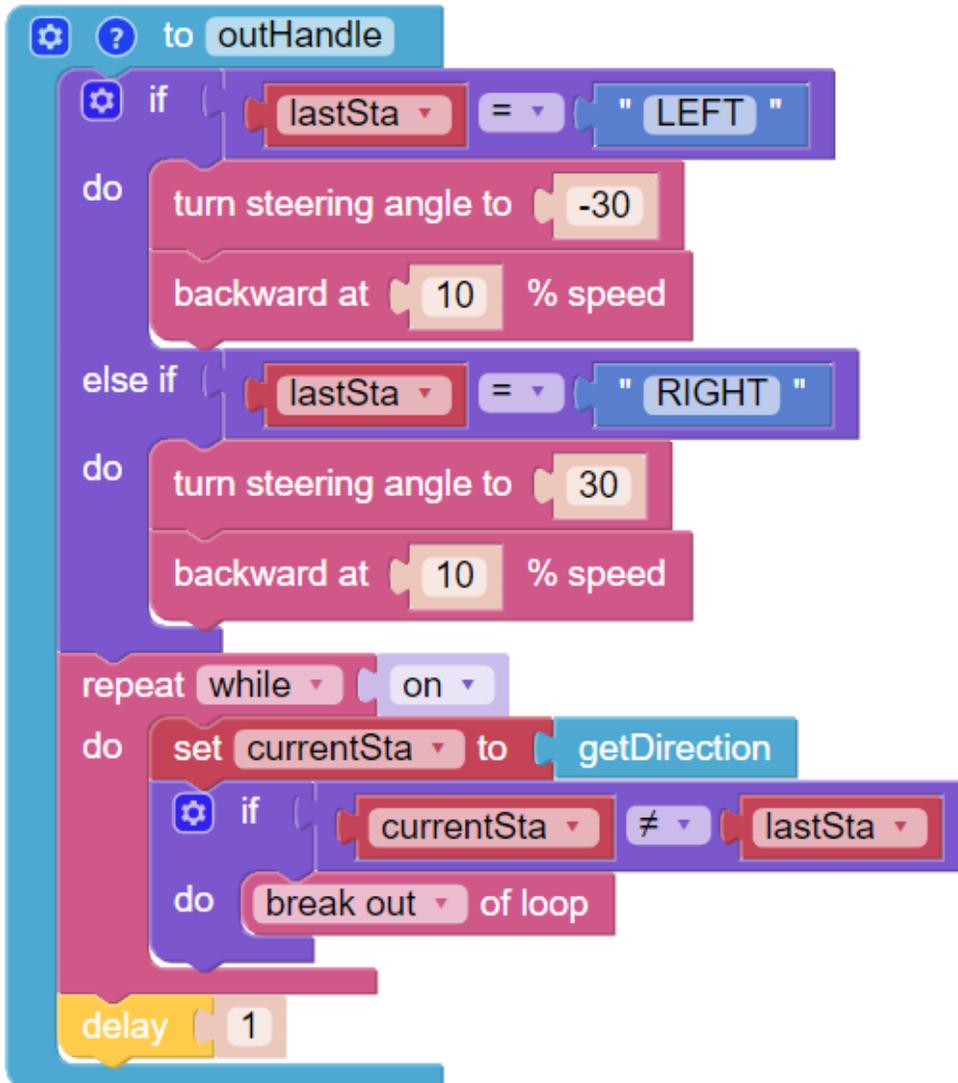
2. Der Block **Set ref to ()** wird verwendet, um den Graustufen-Schwellenwert festzulegen. Du musst ihn entsprechend der tatsächlichen Situation anpassen. Du kannst *Test des Graustufenmoduls* ausführen, um die Werte des Graustufen-Moduls auf weißen und schwarzen Oberflächen zu sehen und ihre mittleren Werte in diesem Block einzutragen.

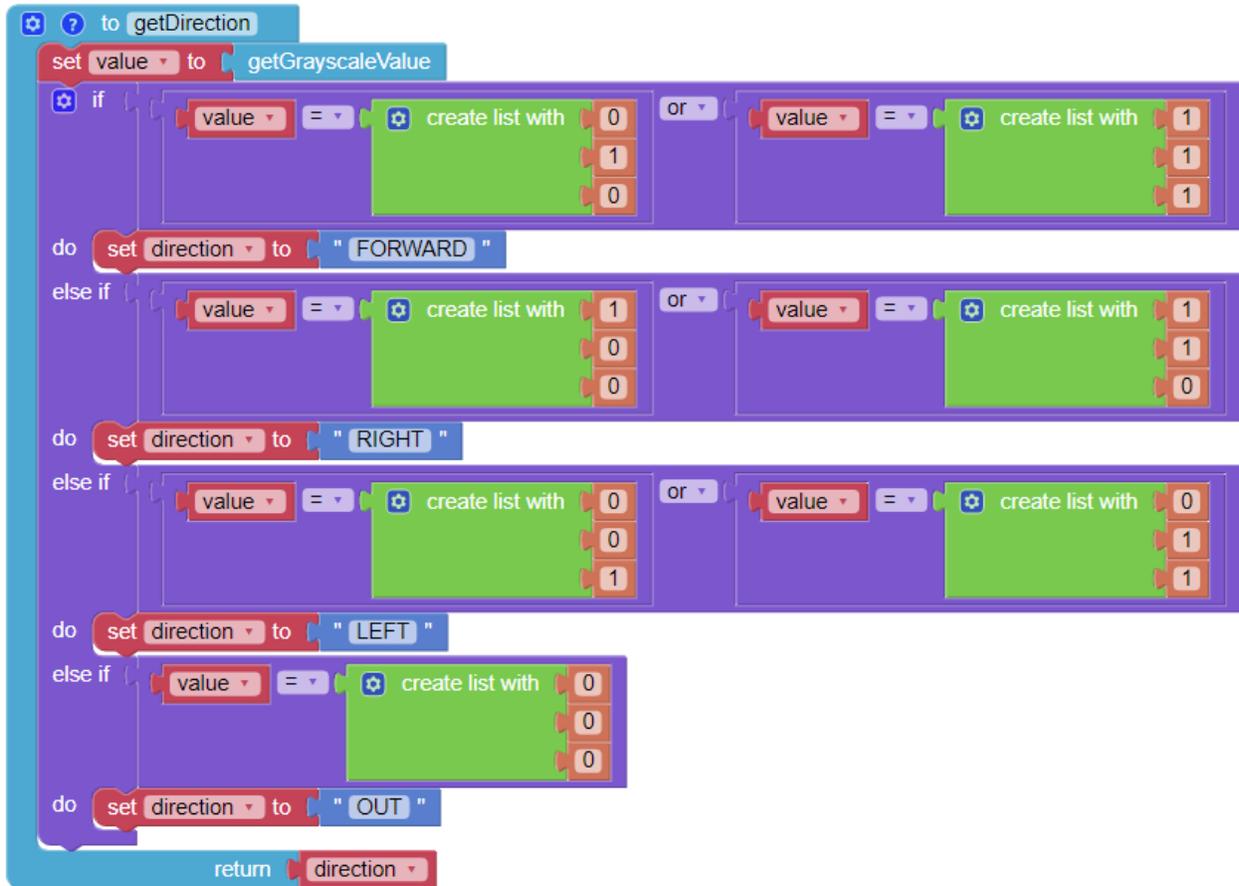
BEISPIEL

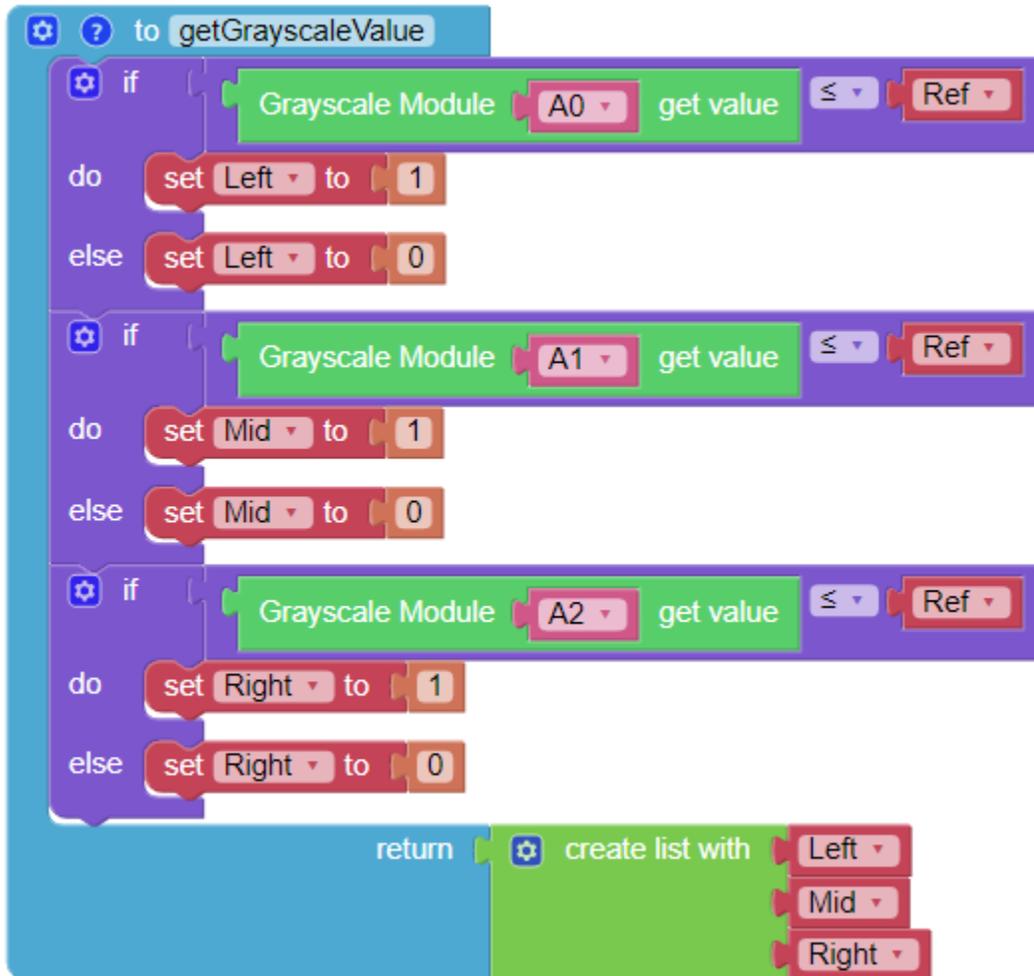
Bemerkung:

- Du kannst das Programm gemäß der folgenden Abbildung schreiben. Bitte beachte das Tutorial: [How to Create a New Project?](#).
- Oder finde den Code mit demselben Namen auf der **Examples**-Seite des EzBlock Studios und klicke direkt auf **Run** oder **Edit**.









6.17 Stierkampf

Verwandeln Sie den PiCar-X in einen wütenden Stier! Bereiten Sie ein rotes Tuch, wie ein Taschentuch, vor und werden Sie ein Stierkämpfer. Achten Sie darauf, nicht getroffen zu werden, wenn der PiCar-X dem roten Tuch hinterherjagt!

Bemerkung: Dieses Projekt ist anspruchsvoller als die vorhergehenden Projekte. Der PiCar-X wird die Farberkennungsfunktion nutzen müssen, um die Kamera auf das rote Tuch auszurichten. Die Körperausrichtung muss sich automatisch an die Richtung anpassen, in die die Kamera zeigt.

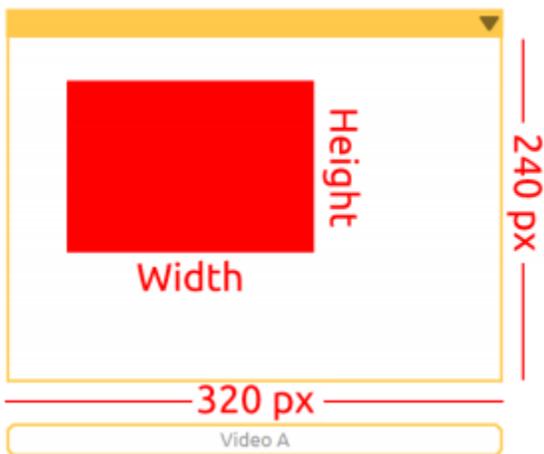
TIPPS



Beginnen Sie damit, den Block **color detection [red]** zum **Start-Widget** hinzuzufügen, um den PiCar-X nach einem roten Objekt suchen zu lassen. In der Dauerschleife fügen Sie den Block **[Breite] der erkannten Farbe** hinzu, um die Eingabe in ein „Objekterkennungs“-Raster zu transformieren.

$(-1,1)$	$(0,1)$	$(1,1)$
$(-1,0)$	$(0,0)$	$(1,0)$
$(-1,-1)$	$(0,-1)$	$(1,-1)$

Die „Objekterkennung“ gibt die erkannten Koordinaten in (x, y) -Werten aus, basierend auf dem Mittelpunkt des Kamerabildes. Der Bildschirm ist in ein 3×3 -Raster unterteilt, wie unten dargestellt. Wenn sich das rote Tuch oben links im Kamerabild befindet, lauten die (x, y) -Koordinaten $(-1, 1)$.

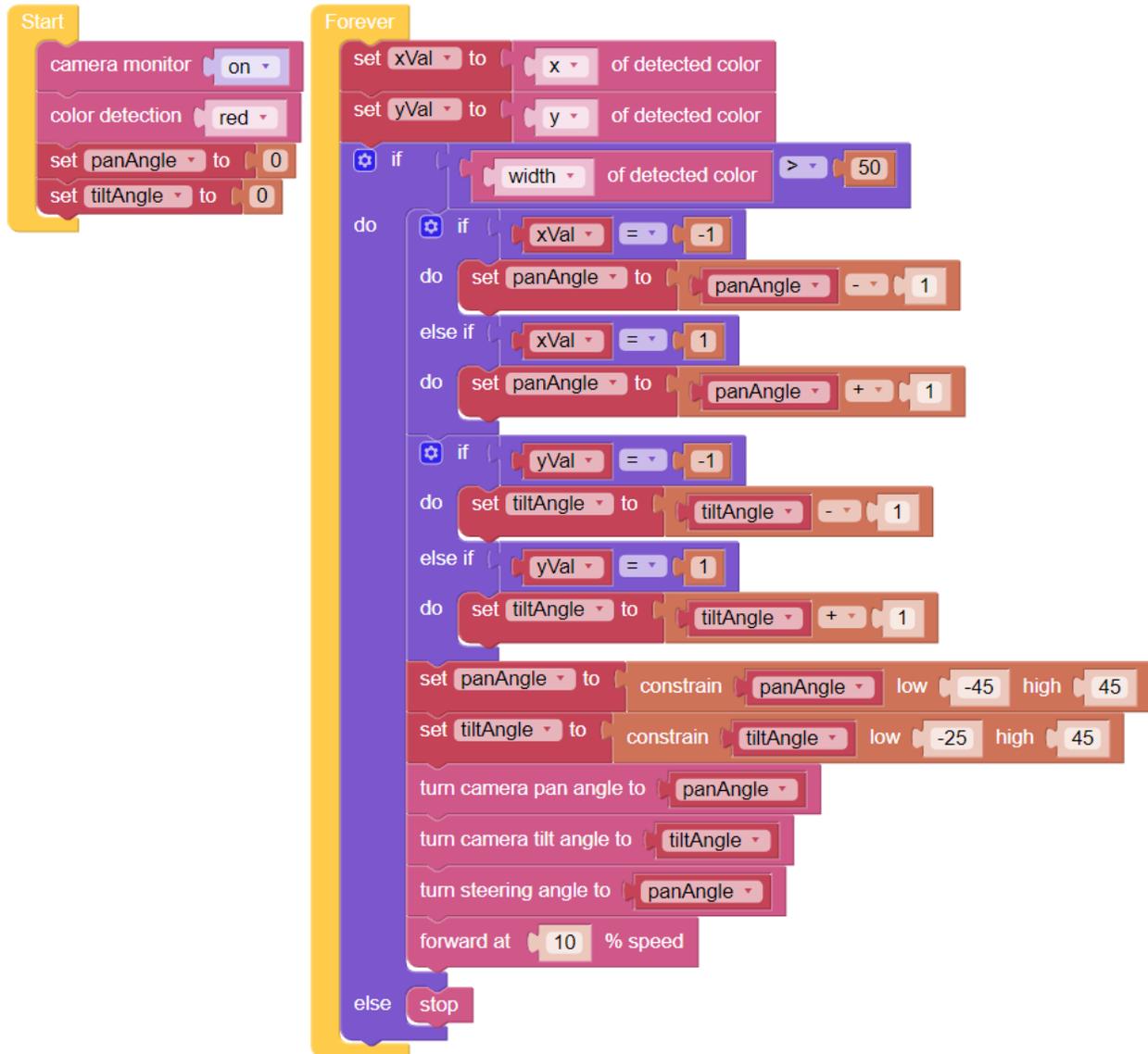


Die „Objekterkennung“ erkennt die Breite und Höhe der Grafik. Wenn mehrere Ziele identifiziert werden, werden die Abmessungen des größten Ziels erfasst.

BEISPIEL

Bemerkung:

- Sie können das Programm gemäß der folgenden Abbildung schreiben. Bitte beachten Sie das Tutorial: [How to Create a New Project?](#).
- Oder finden Sie den Code mit demselben Namen auf der **Examples**-Seite des EzBlock Studios und klicken Sie direkt auf **Run** oder **Edit**.



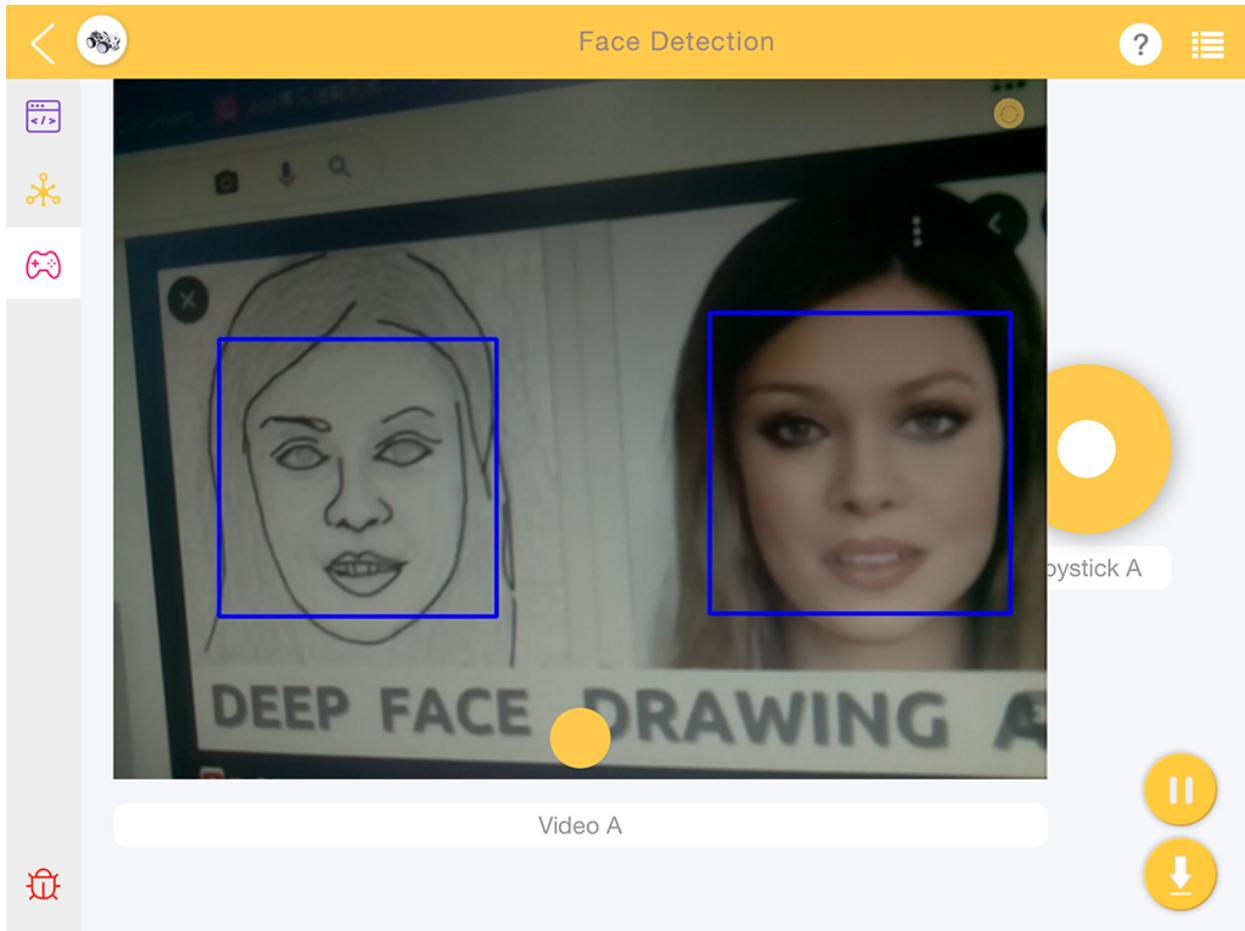
6.18 Achtung Fußgänger

In diesem Projekt wird der PiCar-X entsprechende Maßnahmen basierend auf den Straßenverhältnissen durchführen. Während der Fahrt wird der PiCar-X anhalten, wenn ein Fußgänger auf seinem Weg erkannt wird.

Sobald das Programm läuft, halten Sie ein Foto einer Person vor den PiCar-X. Der Video-Monitor erkennt das Gesicht der Person, und der PiCar-X wird automatisch anhalten.

Um Fahrsicherheitsprotokolle zu simulieren, wird ein Beurteilungsverfahren erstellt, das einen **[Zähler]**-Wert an einen **if do else**-Block sendet. Das Beurteilungsverfahren sucht 10-mal nach einem menschlichen Gesicht, und wenn ein Gesicht erscheint, wird **[Zähler]** um +1 erhöht. Wenn **[Zähler]** größer als 3 ist, wird der PiCar-X anhalten.

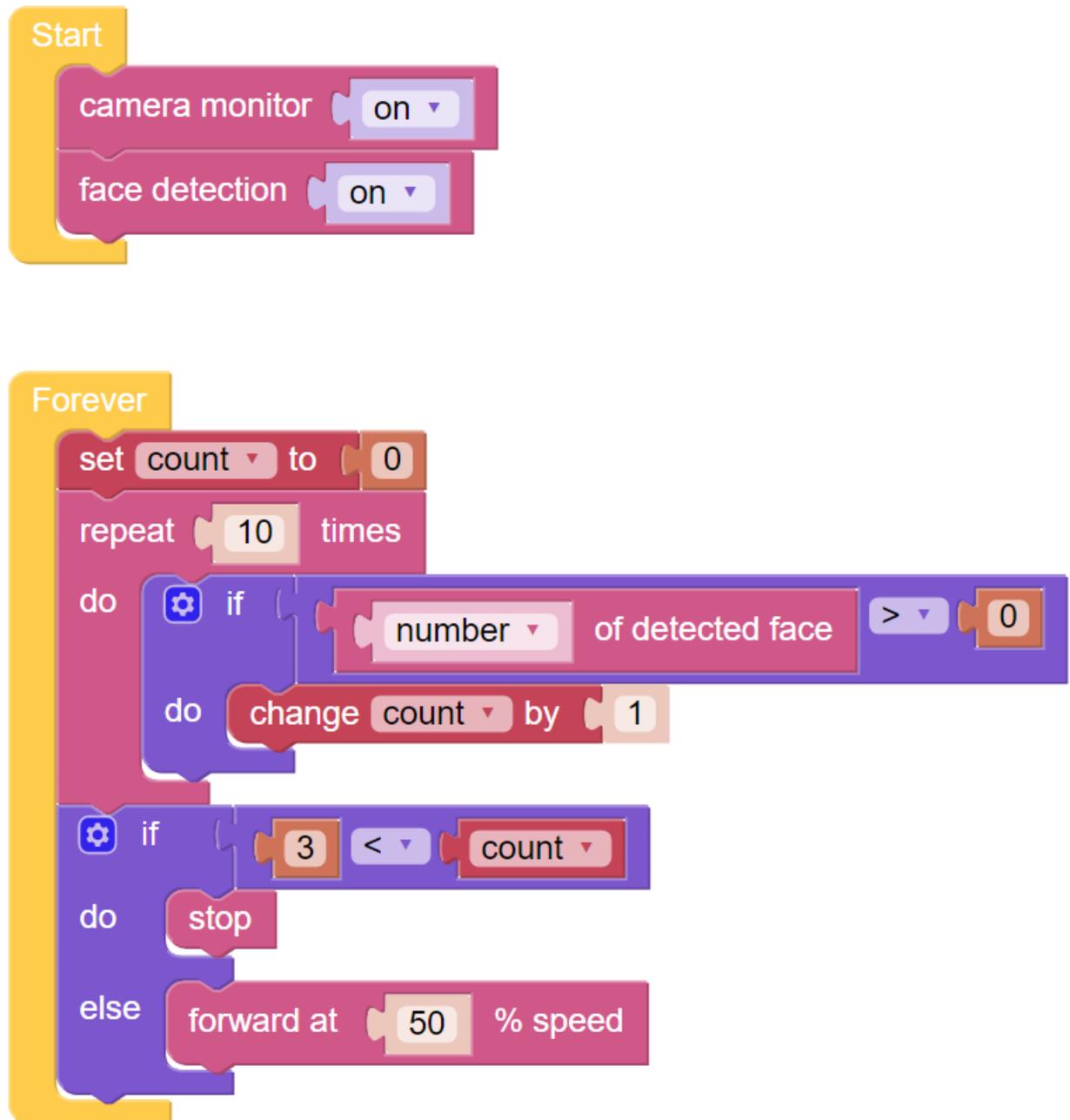
- [How to Use the Remote Control Function?](#)



BEISPIEL

Bemerkung:

- Sie können das Programm gemäß der folgenden Abbildung schreiben. Bitte beachten Sie das Tutorial: [How to Create a New Project?](#).
- Oder finden Sie den Code mit demselben Namen auf der **Examples**-Seite des EzBlock Studios und klicken Sie direkt auf **Run** oder **Edit**.



6.19 Verkehrsschilderkennung

Neben der Farberkennung und der Gesichtserkennung kann der PiCar-X auch Verkehrsschilder erkennen.

Lassen Sie uns nun diese Verkehrsschilderkennung mit der Linienverfolgungsfunktion kombinieren. Der PiCar-X wird der Linie folgen, und wenn Sie das Stoppschild davor platzieren, wird er anhalten. Wenn Sie ein Vorwärtsschild davor platzieren, wird er weiterhin vorwärts fahren.

TIPPS

1. Der PiCar erkennt 4 verschiedene Verkehrsschildmodelle, die im druckbaren PDF unten enthalten sind.



- [PDF]Verkehrsschildkarten

2. Der Block **Set ref to ()** wird verwendet, um den Grauwert-Schwellenwert festzulegen. Sie müssen ihn entsprechend der tatsächlichen Situation anpassen. Sie können den Befehl *Test des Graustufenmoduls* ausführen, um die Werte des Grauwertmoduls auf weißen und schwarzen Oberflächen zu sehen, und deren mittlere Werte in diesem Block eintragen.

BEISPIEL

Bemerkung:

- Sie können das Programm gemäß der folgenden Abbildung schreiben. Bitte beachten Sie das Tutorial: [How to Create a New Project?](#).
 - Oder finden Sie den Code mit demselben Namen auf der **Examples** -Seite des EzBlock Studios und klicken Sie direkt auf **Run** oder **Edit**.
-

```
Start
  set Ref to 700
  camera monitor on
  turn traffic sign detection on
  set Key to 1

Forever
  GetTraffic
  if Key = 0
  do stop
  else if Key = 1
  do
    set status to getDirection
    if status = "FORWARD"
    do
      turn steering angle to 0
      forward at 10 % speed
    else if status = "LEFT"
    do
      turn steering angle to 20
      forward at 10 % speed
    else if status = "RIGHT"
    do
      turn steering angle to -20
      forward at 10 % speed
    else if status = "OUT"
    do stop
```

The image shows a Scratch script for a PiCar-X kit. It starts with a 'Start' block containing four blocks: 'set Ref to 700', 'camera monitor on', 'turn traffic sign detection on', and 'set Key to 1'. Below this is a 'Forever' loop containing a 'GetTraffic' block. Inside the loop, there is an 'if' block for 'Key = 0' leading to a 'stop' block. An 'else if' block for 'Key = 1' leads to a 'do' block. Inside this 'do' block, there is a 'set status to getDirection' block, followed by a series of 'if-else if' blocks. The first 'if' block is for 'status = "FORWARD"', leading to 'turn steering angle to 0' and 'forward at 10 % speed'. The second 'else if' block is for 'status = "LEFT"', leading to 'turn steering angle to 20' and 'forward at 10 % speed'. The third 'else if' block is for 'status = "RIGHT"', leading to 'turn steering angle to -20' and 'forward at 10 % speed'. The final 'else if' block is for 'status = "OUT"', leading to a 'stop' block.

```
to GetTraffic
  set Traffic to type of detected traffic sign
  print Traffic
  if Traffic = "stop"
    do set Key to 0
  else if Traffic = "forward"
    do set Key to 1
end

to getDirection
  set value to getGrayscaleValue
  if value = [create list with 0, 1, 0] or value = [create list with 1, 1, 1]
    do set direction to "FORWARD"
  else if value = [create list with 1, 0, 0] or value = [create list with 1, 1, 0]
    do set direction to "RIGHT"
  else if value = [create list with 0, 0, 1] or value = [create list with 0, 1, 1]
    do set direction to "LEFT"
  else if value = [create list with 0, 0, 0]
    do set direction to "OUT"
  return direction
end
```

The image shows two Scratch code blocks. The first block, 'to GetTraffic', sets a variable 'Traffic' to the 'type' of a detected traffic sign, prints it, and then uses an 'if' statement to set a 'Key' variable to 0 if the traffic sign is 'stop', or to 1 if it is 'forward'. The second block, 'to getDirection', sets a 'value' variable to 'getGrayscaleValue'. It then uses a series of 'if' and 'else if' statements to determine the 'direction' based on the 'value' variable. Each condition involves comparing 'value' to a list of three grayscale values. The directions are 'FORWARD', 'RIGHT', 'LEFT', and 'OUT'. Finally, the 'direction' variable is returned.



6.20 Orientierungslauf

In diesem Projekt wird die Fernsteuerungsfunktion verwendet, um den PiCar-X bei einer Wettbewerbs-Schnitzeljagd zu führen!

Richten Sie zunächst entweder einen Hindernisparcours, ein Labyrinth oder sogar einen leeren Raum ein, durch den der PiCar-X fahren kann. Platzieren Sie dann zufällig sechs Markierungen entlang der Strecke und legen Sie an jeder der sechs Markierungen eine Farbkarte aus, die der PiCar-X finden soll.

Die sechs Farbmodelle für den PiCar-X sind: Rot, Orange, Gelb, Grün, Blau und Lila und können aus dem untenstehenden PDF mit einem Farbdrucker gedruckt werden.

- [PDF]Farbkarten

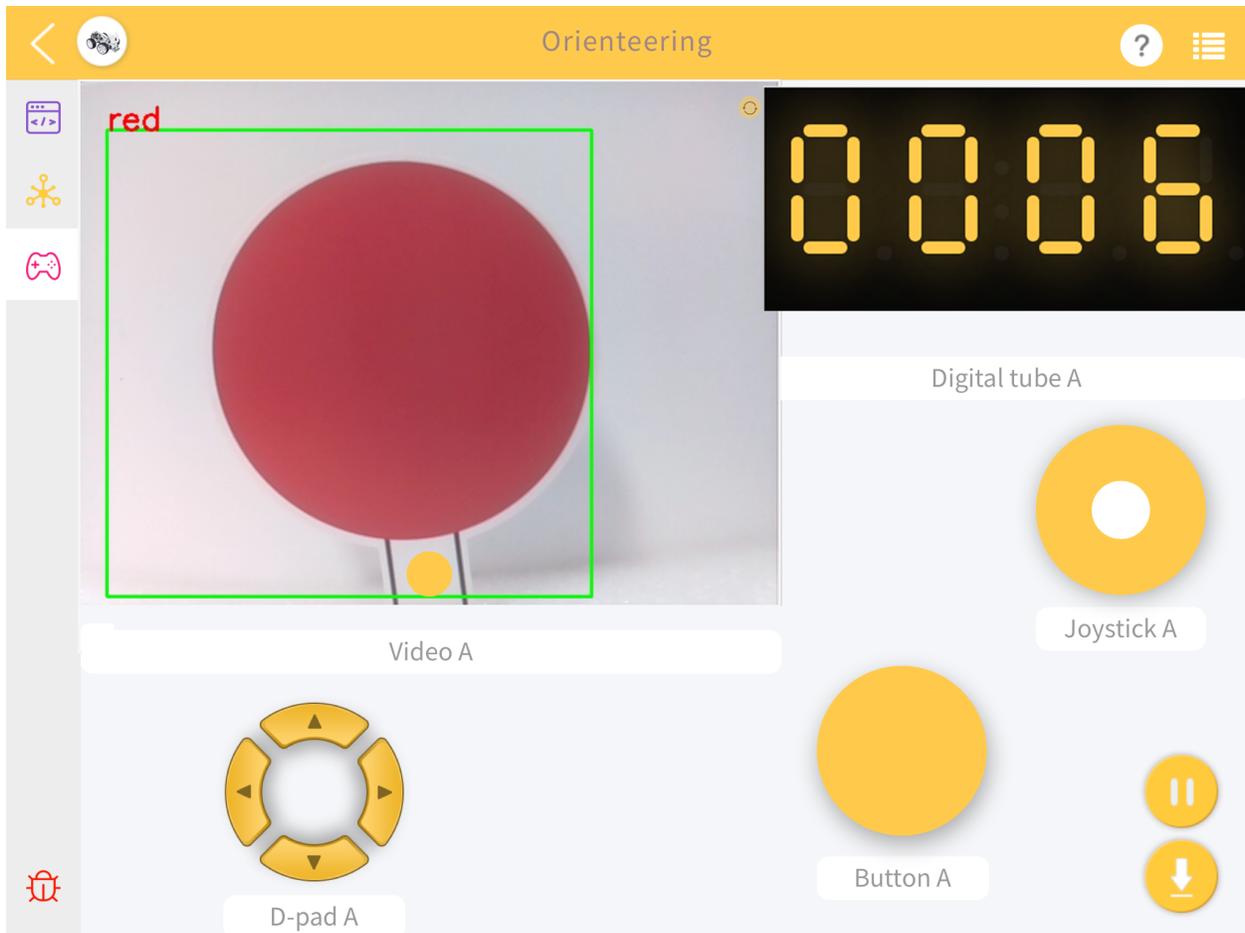


Bemerkung: Die gedruckten Farben können aufgrund von Unterschieden im Druckertoner oder dem gedruckten Medium, wie zum Beispiel einem sandfarbenen Papier, eine leicht unterschiedliche Farbnuance aufweisen. Dies kann zu einer weniger genauen Farberkennung führen.

Der PiCar-X wird programmiert, um drei der sechs Farben in zufälliger Reihenfolge zu finden, und wird die TTS-Funktion verwenden, um anzukündigen, nach welcher Farbe als nächstes gesucht werden soll.

Das Ziel besteht darin, dem PiCar-X dabei zu helfen, jede der drei Farben so schnell wie möglich zu finden.

Platzieren Sie den PiCar-X in der Mitte des Spielfelds und klicken Sie auf die Schaltfläche auf der Fernsteuerungsseite, um das Spiel zu starten.

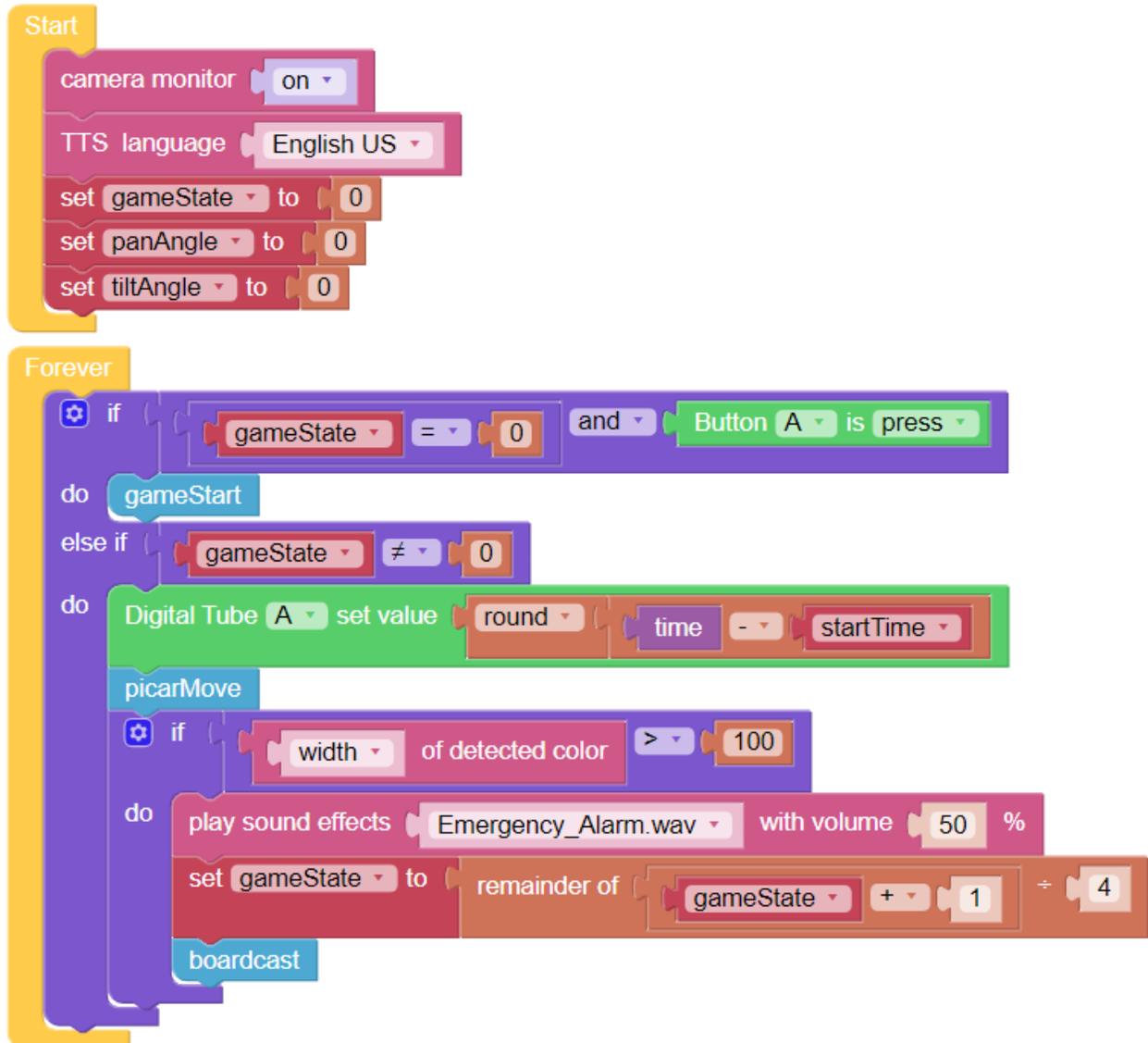


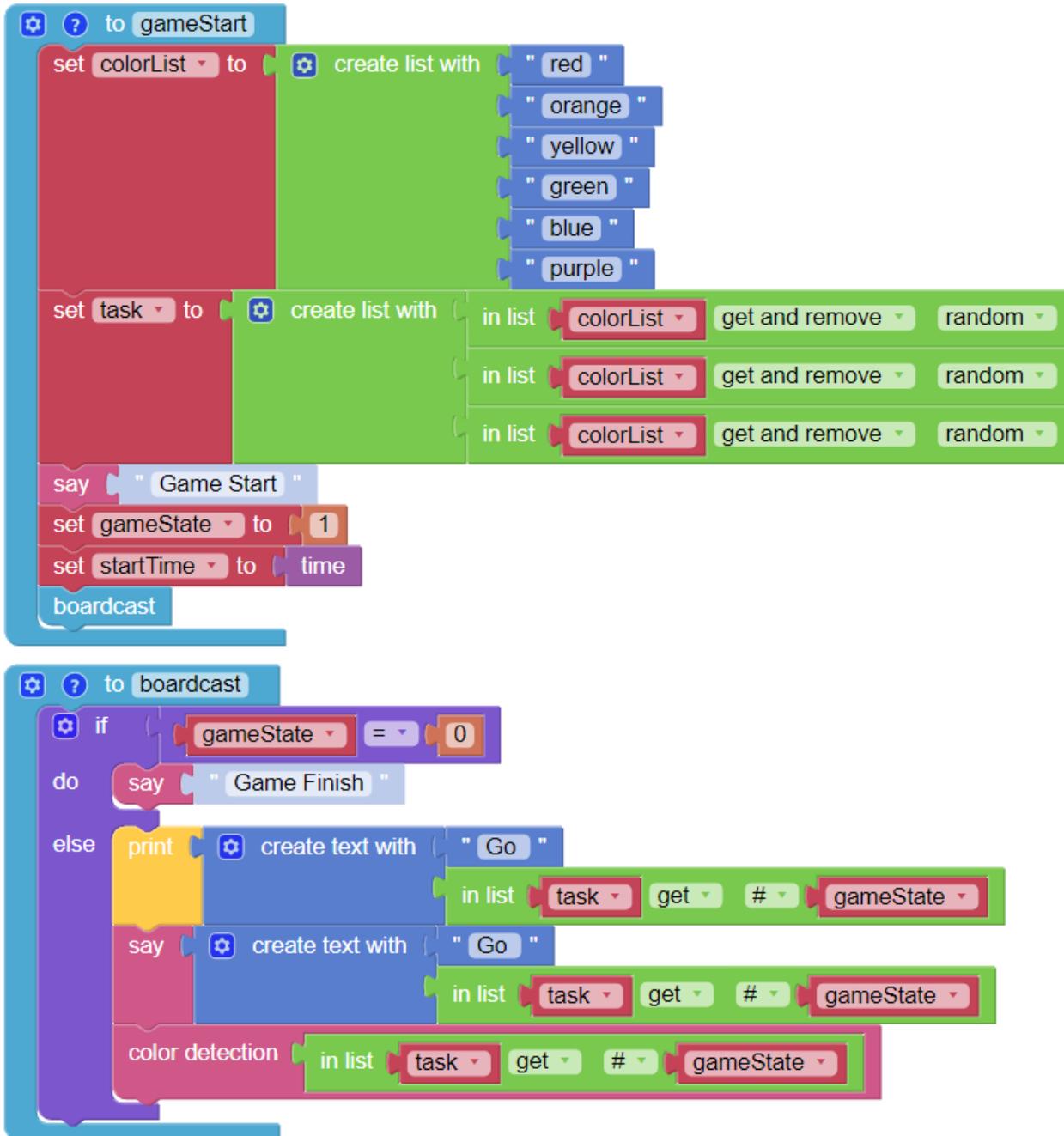
Spielen Sie abwechselnd mit Freunden dieses Spiel, um herauszufinden, wer dem PiCar-X am schnellsten dabei helfen kann, das Ziel zu erreichen!

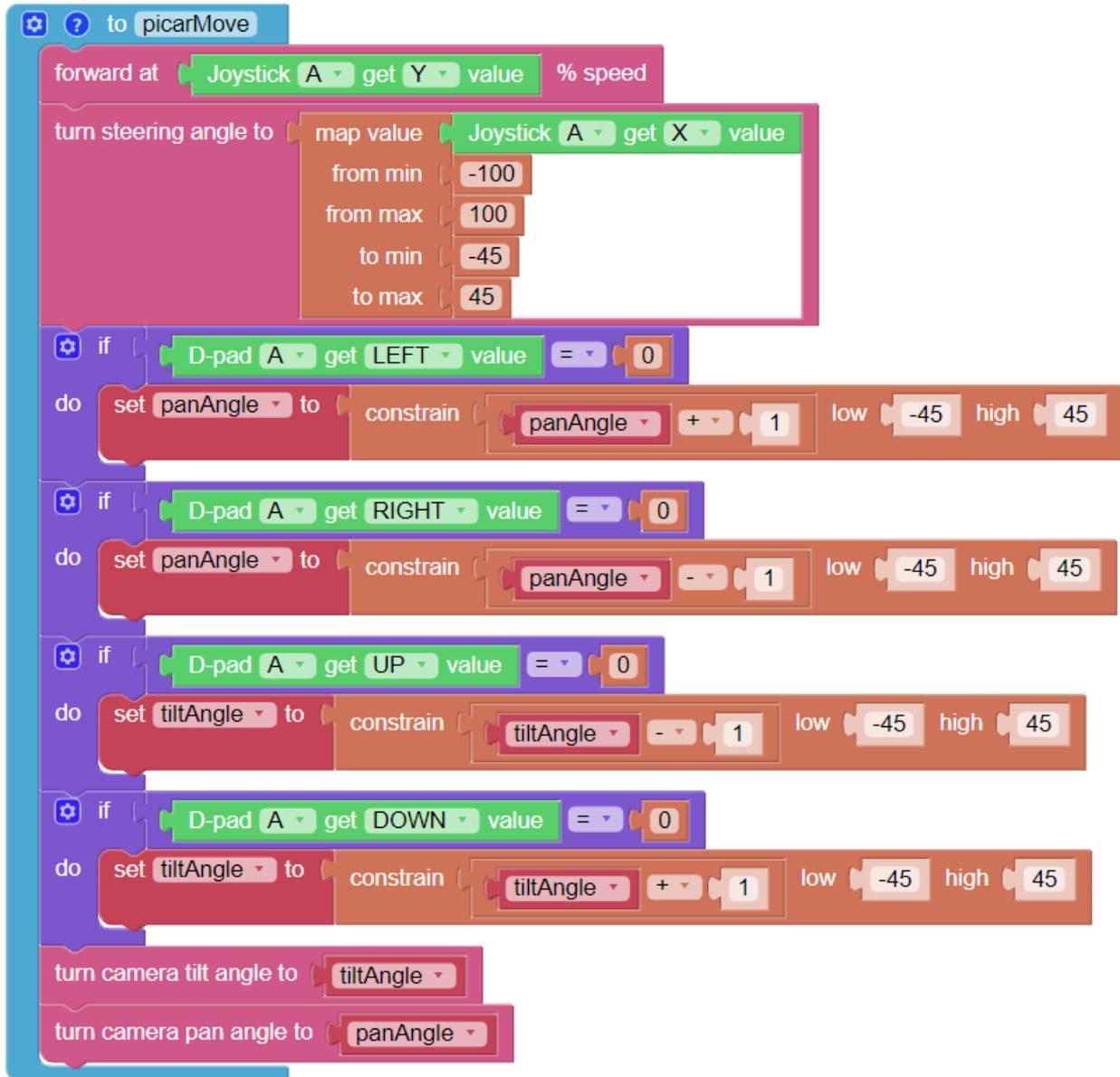
BEISPIEL

Bemerkung:

- Sie können das Programm gemäß der folgenden Abbildung schreiben. Bitte beachten Sie das Tutorial: [How to Create a New Project?](#).
- Oder finden Sie den Code mit demselben Namen auf der **Examples** -Seite des EzBlock Studios und klicken Sie direkt auf **Run** oder **Edit**.







7.1 Filezilla Software



Das File Transfer Protocol (FTP) ist ein standardisiertes Kommunikationsprotokoll, das für die Übertragung von Computerdateien von einem Server auf einen Client in einem Computernetzwerk verwendet wird.

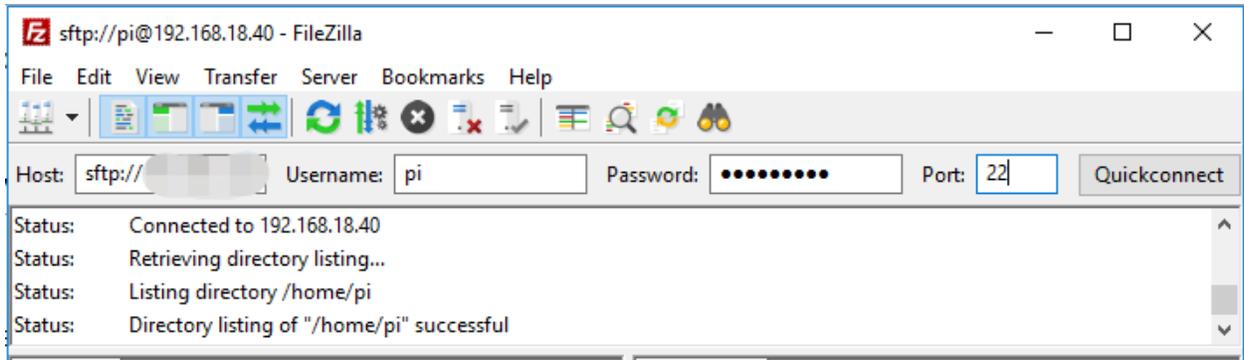
Filezilla ist eine Open-Source-Software, die nicht nur FTP unterstützt, sondern auch FTP über TLS (FTPS) und SFTP. Mit Filezilla können wir lokale Dateien (wie Bilder und Audio usw.) auf den Raspberry Pi hochladen oder Dateien vom Raspberry Pi auf das lokale Gerät herunterladen.

Schritt 1: Laden Sie Filezilla herunter.

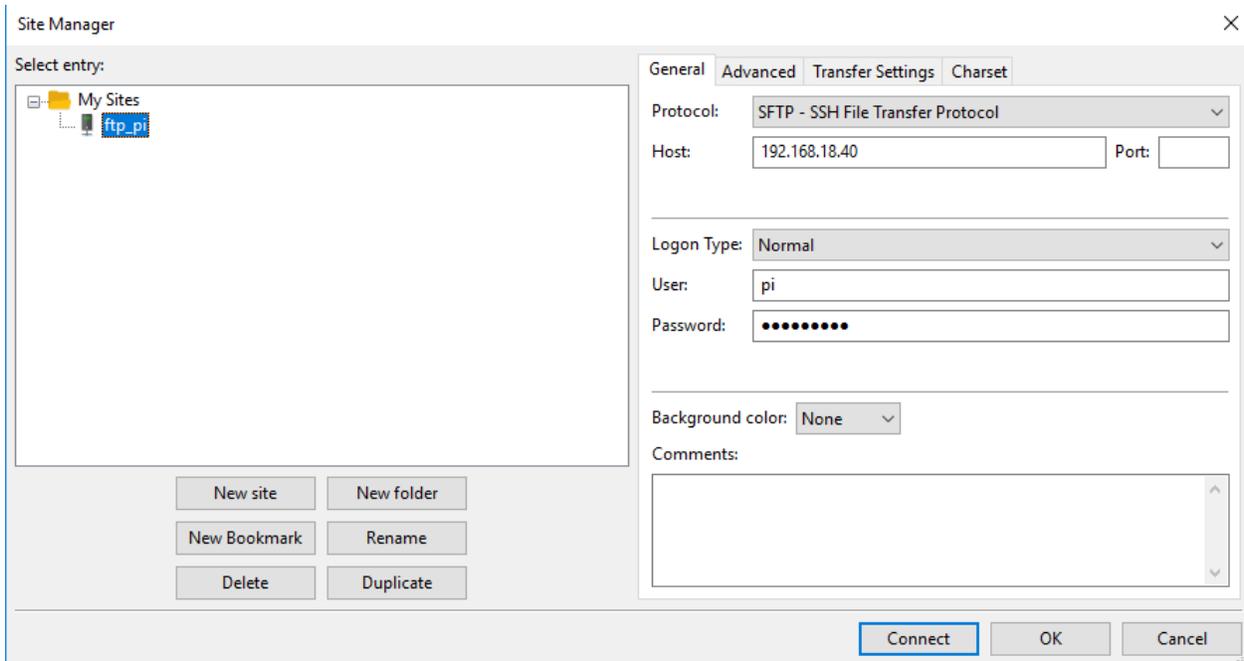
Laden Sie den Client von der offiziellen Website von Filezilla [Filezilla's offizielle Website](#) herunter. Filezilla bietet ein sehr gutes Tutorial, siehe: [Dokumentation - Filezilla](#).

Schritt 2: Verbindung zum Raspberry Pi herstellen

Nach einer schnellen Installation öffnen Sie das Programm und verbinden es nun [mit einem FTP-Server](#). Es gibt 3 Möglichkeiten zur Verbindung, hier verwenden wir die **Quick Connect**. Geben Sie den **hostname/IP**, **username**, **password** und **port (22)** ein, klicken Sie dann auf **Quick Connect** oder drücken Sie **Enter**, um die Verbindung zum Server herzustellen.

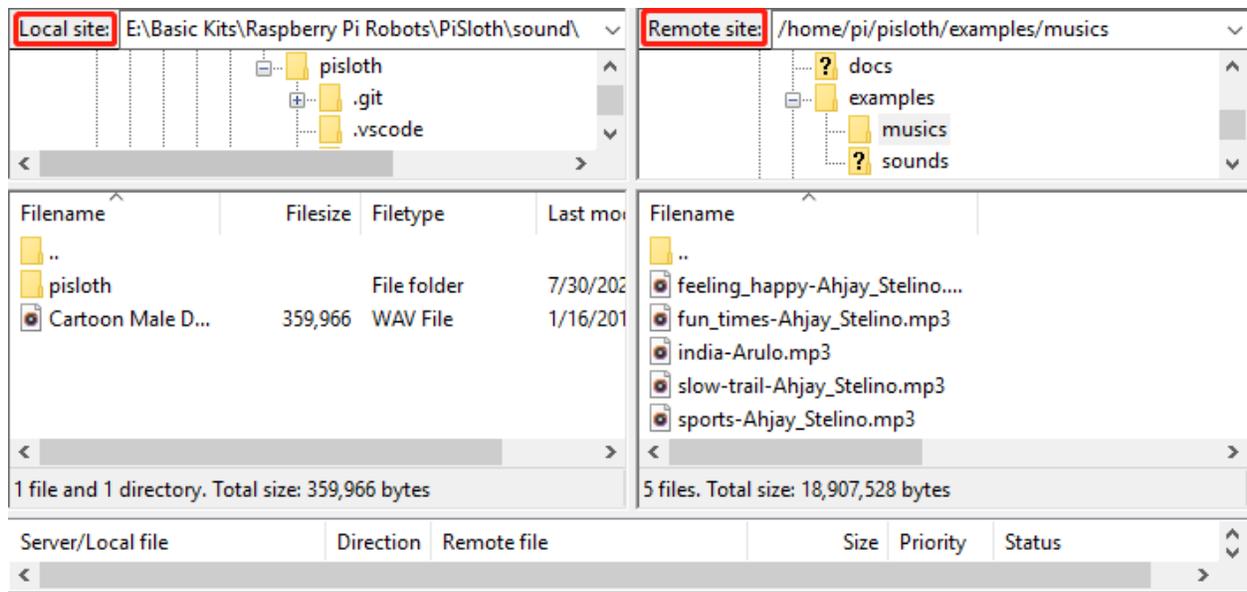


Bemerkung: Schnellverbinden ist eine gute Möglichkeit, Ihre Anmeldeinformationen zu testen. Wenn Sie einen dauerhaften Eintrag erstellen möchten, können Sie nach einer erfolgreichen Schnellverbindung **File -> Copy Current Connection to Site Manager** wählen, den Namen eingeben und auf **OK** klicken. Beim nächsten Mal können Sie die Verbindung herstellen, indem Sie die zuvor gespeicherte Seite unter **File -> Site Manager** auswählen.



Schritt 3: Dateien hoch-/herunterladen.

Sie können lokale Dateien auf den Raspberry Pi übertragen, indem Sie diese per Drag-and-Drop hochladen, oder die auf dem Raspberry Pi befindlichen Dateien lokal herunterladen.



7.2 PuTTY

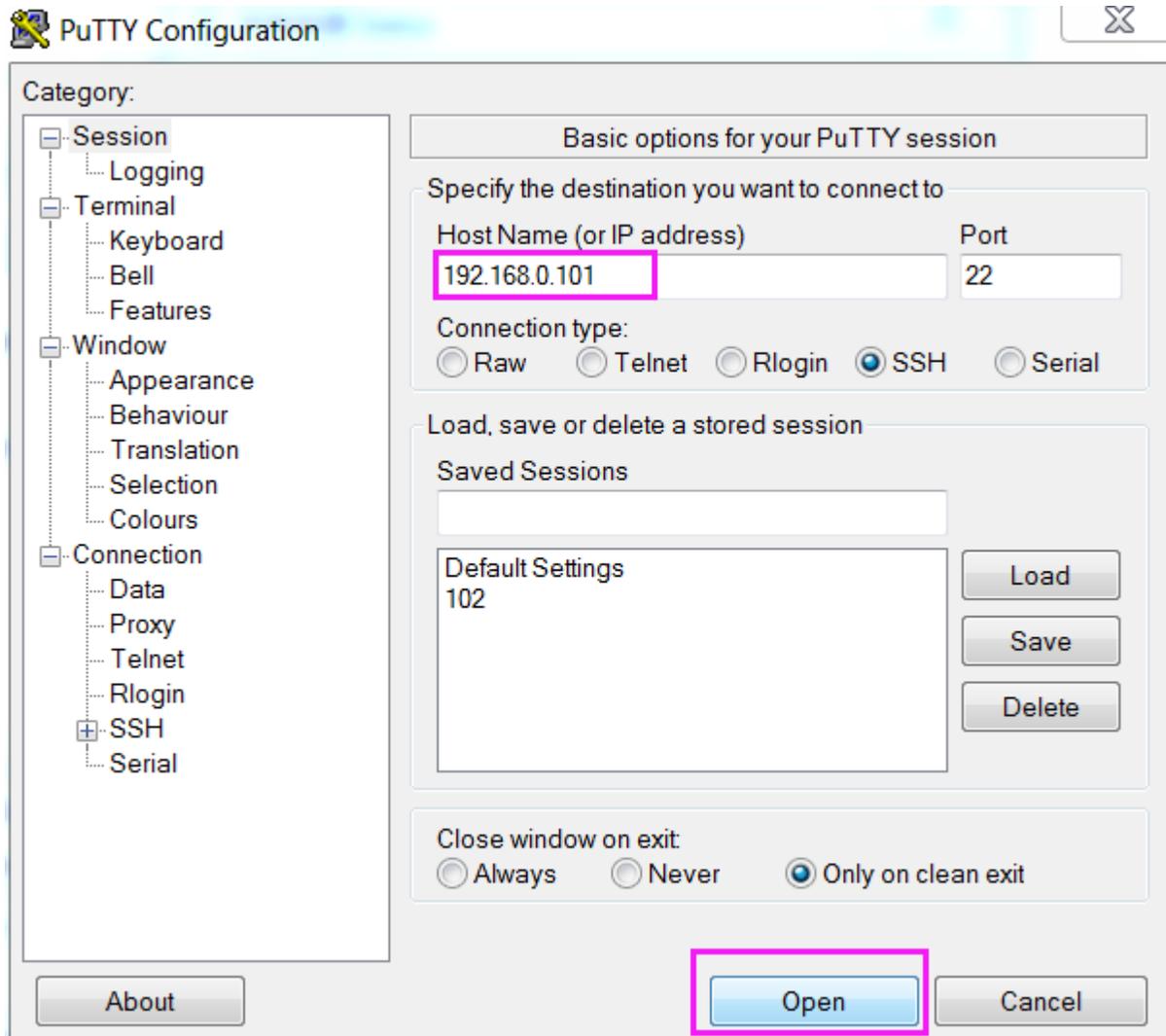
Wenn Sie ein Windows-Benutzer sind, können Sie einige SSH-Anwendungen verwenden. Hier empfehlen wir PuTTY.

Schritt 1

PuTTY herunterladen.

Schritt 2

Öffnen Sie PuTTY und klicken Sie im baumähnlichen Menü links auf **Session**. Geben Sie die IP-Adresse des RPi in das Feld unter **Host Name (oder IP-Adresse)** und **22** unter **Port** ein (standardmäßig ist es 22).



Schritt 3

Klicken Sie auf **Open**. Beachten Sie, dass bei der ersten Anmeldung am Raspberry Pi mit der IP-Adresse eine Sicherheitserinnerung erscheint. Klicken Sie einfach auf **Ja**.

Schritt 4

Wenn im PuTTY-Fenster „**login as:**“ angezeigt wird, geben Sie „**pi**“ ein (der Benutzername des RPi) und als **Passwort:** „**raspberrypi**“ (das Standardpasswort, sofern Sie es nicht geändert haben).

Bemerkung: Wenn Sie das Passwort eingeben, werden die Zeichen nicht im Fenster angezeigt, was normal ist. Wichtig ist nur, dass Sie das richtige Passwort eingeben.

Wenn neben PuTTY „inaktiv“ angezeigt wird, bedeutet dies, dass die Verbindung unterbrochen wurde und erneut hergestellt werden muss.

```

pi@raspberrypi: ~
login as: pi
pi@192.168.0.234's password: raspberry

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Feb 21 02:54:55 2017
pi@raspberrypi:~ $

```

Schritt 5

Jetzt haben wir eine Verbindung zum Raspberry Pi hergestellt und es ist Zeit für die nächsten Schritte.

7.3 Installieren Sie OpenSSH über Powershell

Wenn Sie `ssh <Benutzername>@<Hostname>.local` (oder `ssh <Benutzername>@<IP-Adresse>`) verwenden, um sich mit Ihrem Raspberry Pi zu verbinden, aber die folgende Fehlermeldung angezeigt wird:

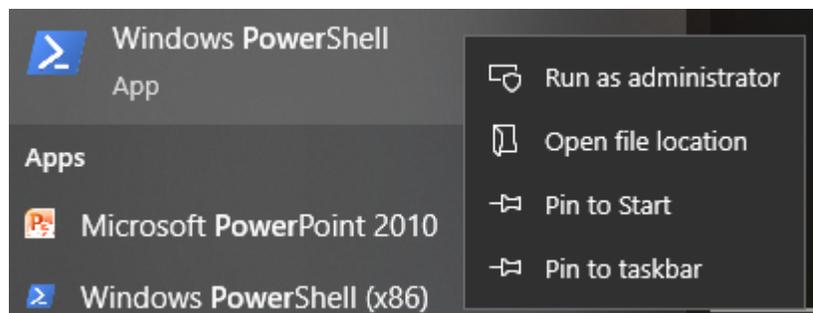
```

ssh: Der Begriff "ssh" wird nicht als Name eines Cmdlets, einer Funktion,
→einer Skriptdatei oder eines ausführbaren Programms erkannt. Überprüfen Sie
→die Rechtschreibung des Namens oder, falls ein Pfad angegeben wurde, stellen
→Sie sicher, dass der Pfad korrekt ist, und versuchen Sie es erneut.

```

Dann bedeutet dies, dass Ihr Computersystem zu alt ist und nicht über [OpenSSH](#) vorinstalliert ist. Sie müssen der folgenden Anleitung folgen, um es manuell zu installieren.

1. Geben Sie `powershell` in die Suchleiste Ihres Windows-Desktops ein, klicken Sie mit der rechten Maustaste auf `Windows PowerShell` und wählen Sie im angezeigten Menü `Als Administrator ausführen`.



2. Verwenden Sie den folgenden Befehl, um `OpenSSH.Client` zu installieren.

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

3. Nach der Installation wird die folgende Ausgabe zurückgegeben.

```

Pfad           :
Online         : True
Neustart erforderlich : False

```

- Überprüfen Sie die Installation mit dem folgenden Befehl.

```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'
```

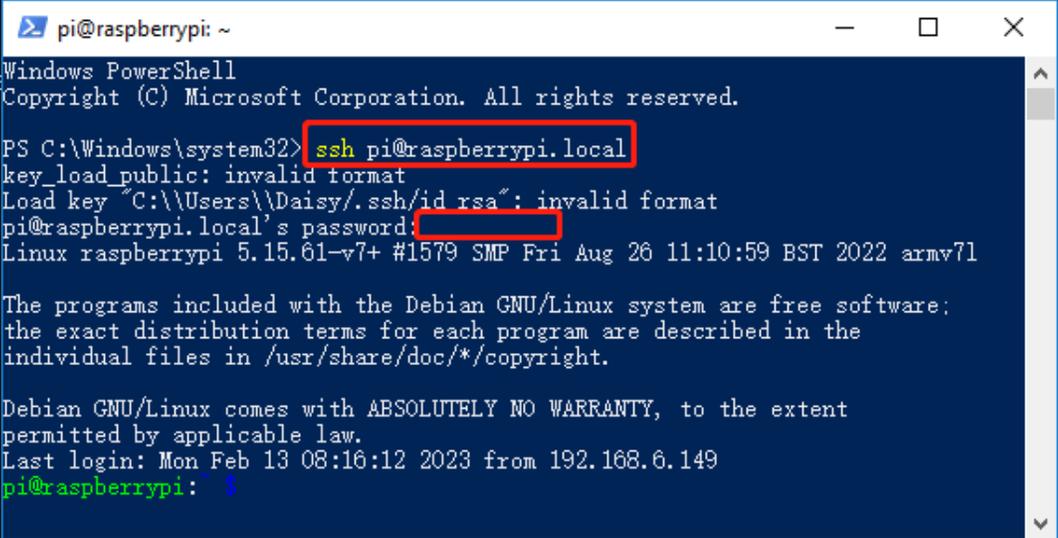
- Es wird Ihnen jetzt mitgeteilt, dass OpenSSH.Client erfolgreich installiert wurde.

```
Name : OpenSSH.Client~~~~0.0.1.0
Zustand : Installiert

Name : OpenSSH.Server~~~~0.0.1.0
Zustand : Nicht vorhanden
```

Warnung: Wenn die obige Aufforderung nicht angezeigt wird, bedeutet dies, dass Ihr Windows-System immer noch zu alt ist, und Sie sollten ein SSH-Tool von Drittanbietern installieren, wie *PuTTY*.

- Starten Sie jetzt PowerShell neu und führen Sie es weiterhin als Administrator aus. An diesem Punkt können Sie sich mit dem Befehl `ssh` auf Ihrem Raspberry Pi anmelden, wo Sie aufgefordert werden, das zuvor eingerichtete Passwort einzugeben.



```
pi@raspberrypi: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

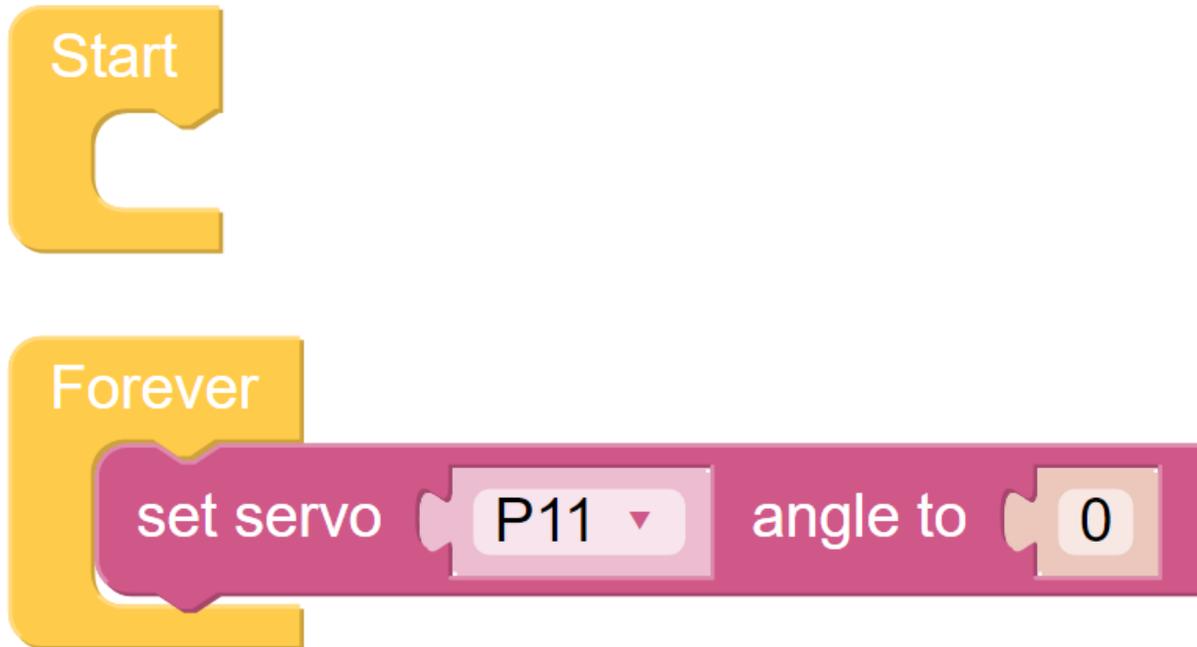
PS C:\Windows\system32> ssh pi@raspberrypi.local
key_load_public: invalid format
Load key "C:\\Users\\Daisy/.ssh/id_rsa": invalid format
pi@raspberrypi.local's password:
Linux raspberrypi 5.15.61-v7+ #1579 SMP Fri Aug 26 11:10:59 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 13 08:16:12 2023 from 192.168.6.149
pi@raspberrypi: ~
```

8.1 Q1: Nach der Installation von Ezblock OS kann der Servo nicht auf 0° drehen?

- 1) Überprüfen Sie, ob das Servokabel richtig angeschlossen ist und ob die Stromversorgung des Robot HAT eingeschaltet ist.
- 2) Drücken Sie den Reset-Knopf.
- 3) Wenn Sie bereits ein Programm im Ezblock Studio ausgeführt haben, ist das benutzerdefinierte Programm für P11 nicht mehr verfügbar. Sie können sich das untenstehende Bild ansehen und im Ezblock Studio ein Programm manuell erstellen, um den Servowinkel auf 0 zu setzen.



8.2 Q2: Bei der Verwendung von VNC werde ich darauf hingewiesen, dass der Desktop momentan nicht angezeigt werden kann?

Geben Sie im Terminal `sudo raspi-config` ein, um die Auflösung zu ändern.

8.3 Q3: Warum kehrt der Servo manchmal ohne Grund in die Mittelposition zurück?

Wenn der Servo durch eine Struktur oder ein anderes Objekt blockiert wird und seine vorgesehene Position nicht erreichen kann, wird der Servo in den Stromabschalt-Schutzmodus versetzt, um zu verhindern, dass er durch zu hohen Stromfluss beschädigt wird.

Nach einer Zeit ohne Stromversorgung wird der Servo, wenn ihm kein PWM-Signal gegeben wird, automatisch in seine Ausgangsposition zurückkehren.

8.4 Q4: Wo finde ich ein detailliertes Tutorial zum Robot HAT?

Hier finden Sie ein umfassendes Tutorial zum Robot HAT, einschließlich Informationen zu seiner Hardware und API.

•

Danke

Vielen Dank an die Gutachter, die unsere Produkte bewertet haben, an die Veteranen, die Vorschläge für das Tutorial gemacht haben, und an die Benutzer, die uns stets begleiten und unterstützen. Ihre wertvollen Vorschläge sind unsere Motivation, bessere Produkte anzubieten!

Besonderer Dank

- Len Davisson
- Kalen Daniel
- Juan Delacosta

Könnten Sie nun etwas Zeit opfern, um diesen Fragebogen auszufüllen?

Bemerkung: Bitte kehren Sie nach dem Ausfüllen des Fragebogens nach oben zurück, um die Ergebnisse anzusehen.

KAPITEL 10

Urheberrechtshinweis

Alle Inhalte dieses Handbuchs, einschließlich aber nicht beschränkt auf Texte, Bilder und Code, sind Eigentum der Sun-Founder Company. Sie dürfen diese nur für persönliches Studium, Forschung, Freude oder andere nicht-kommerzielle oder gemeinnützige Zwecke verwenden, gemäß den einschlägigen Vorschriften und Urheberrechtsgesetzen, ohne die rechtlichen Rechte des Autors und der betreffenden Rechteinhaber zu verletzen. Für jede Einzelperson oder Organisation, die diese ohne Genehmigung für kommerzielle Gewinne verwendet, behält sich das Unternehmen das Recht vor, rechtliche Schritte einzuleiten.