
SunFounder Zeus Robot Car Kit for Arduino

Release 1.0

www.sunfounder.com

18.02.2024

Inhaltsverzeichnis

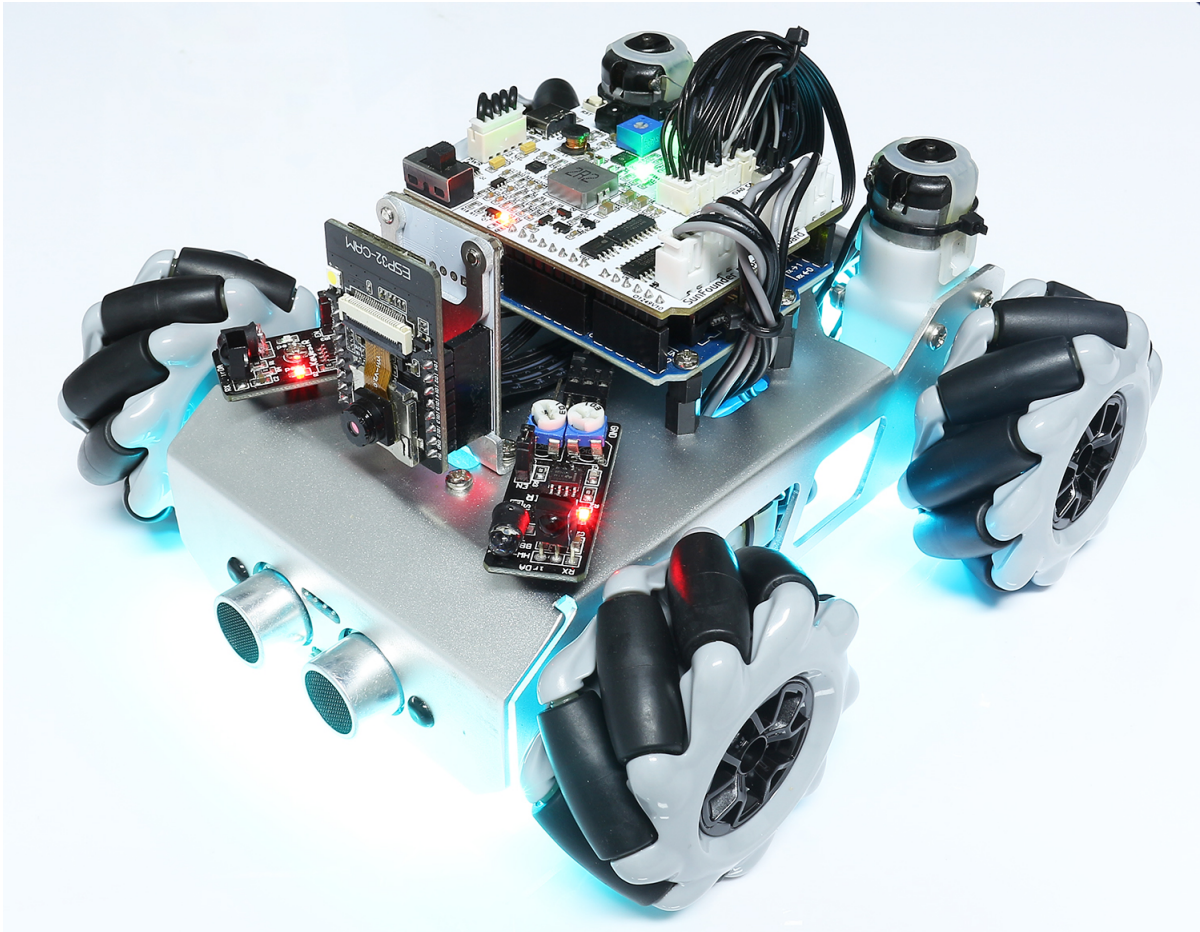
1	Los geht's	3
1.1	Das Auto zusammenbauen	3
1.2	Spielmodus	4
1.3	Programmiermodus	28
2	Hardware	111
2.1	SunFounder R3 Board	112
2.2	Zeus Car Shield	114
2.3	ESP32 CAM	125
2.4	Kamera-Adapterplatine	127
2.5	Omni-Graustufenmodul	133
2.6	Ultraschallmodul	139
2.7	IR Hinderniserkennungsmodul	141
2.8	4 RGB LED-Streifen	144
2.9	18650 Batterie	145
2.10	TT Motor	146
2.11	Mecanum-Rad	147
3	FAQ	151
3.1	Kompilierungsfehler: SoftPWM.h: Datei oder Verzeichnis nicht gefunden	151
3.2	avrdude: stk500_getsync() Versuch 10 von 10: nicht synchronisiert: resp=0x6e?	151
3.3	Q3: Wie kann ich die STT-Funktion auf meinem Android-Gerät nutzen?	152

Danke, dass Sie sich für unser Zeus Car entschieden haben.

Bemerkung: Dieses Dokument ist in den folgenden Sprachen verfügba.

-
-
-

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



Dies ist ein Bildungsbausatz, mit dem Anfänger (Kinder) praktische Erfahrungen in Elektronik, Robotik und Programmierung sammeln können.

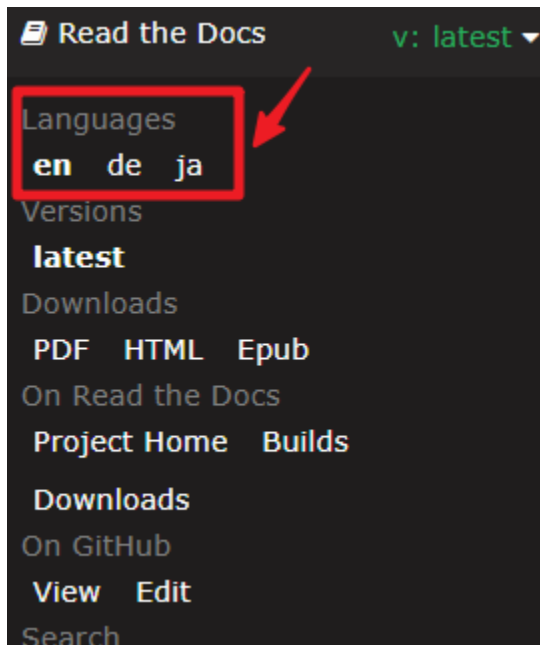
- Mit 6cm Mecanum-Rädern für einfache 360°-Rundumbewegung, einschließlich schwieriger Bewegungen wie seitliches Driften.
- Ausgestattet mit einer ESP32 CAM, ermöglicht erste Erfahrungen aus der Fahrer-Perspektive in der App.
- Mit 2 Hindernisvermeidungsmodulen und einem Ultraschallmodul kann das Auto Hindernissen ausweichen und diesen folgen.
- Am Boden ist ein 8-Kanal-Kreismodul angebracht, mit dem das Auto in jede Richtung einer Linie folgen kann.
- Es gibt auch eine IR-Fernbedienung und coole Lichteffekte.
- Die Modulschnittstellen sind XH1.5, was den Zusammenbau vereinfacht und die Fehlermöglichkeit verringert.

Wir haben einen *Spielmodus* dafür eingerichtet. So können die Benutzer das Auto nach dem Zusammenbau direkt einschalten und mit seinen vielen coolen Funktionen über die eigene Fernbedienung oder Mobil-App spielen. Wenn Sie mehr darüber erfahren möchten, wie Sie seine verschiedenen Funktionen implementieren können, werfen Sie einen Blick auf den *Programmiermodus*.

Wenn Sie Fragen haben, senden Sie bitte eine E-Mail an service@sunfounder.com, und wir werden so schnell wie möglich antworten.

Über die Anzeigesprache

Dieses Dokument ist auch in anderen Sprachen verfügbar. Um die Anzeigesprache zu ändern, klicken Sie bitte auf das **Read the Docs**-Symbol unten links auf der Seite.



Wir haben für das Zeus Auto zwei Modi eingerichtet: den Hands-on-Spielmodus und den Programmiermodus.

- Der Hands-on *Spielmodus* bedeutet, dass Sie das Zeus Auto nach dem Zusammenbau einfach einschalten müssen und es dann direkt mit der Fernbedienung oder der APP steuern können, um verschiedene coole Funktionen auszuführen. Denn wir haben den Code bereits im Werk auf das Arduino-Board hochgeladen.
- Der *Programmiermodus* ist für den Fall, dass Sie nach dem Erleben der verschiedenen coolen Funktionen des Zeus Autos jedes Projekt einzeln nach seinem Prinzip verstehen können und es dann nach Ihren eigenen Vorstellungen ändern, um die gewünschten Funktionen zu erreichen.

1.1 Das Auto zusammenbauen

Hier finden Sie die Montageanleitung für das Zeus Auto in der PDF-Version, die mit dem beiliegenden Faltblatt identisch ist. Wenn Sie der Meinung sind, dass Sie auf dem Faltblatt etwas nicht lesen können, können Sie dieses PDF herunterladen.

(PDF) Montageanleitung Zeus Auto

Montage-Anleitungsvideo

Dieses Video führt Sie durch den Prozess, Ihren Roboter von Grund auf zusammenzubauen.

In diesem Tutorial lernen Sie:

- **Vorbereitung:** Wir stellen Ihnen alle benötigten Werkzeuge und Teile vor, damit Sie gut ausgerüstet sind, bevor Sie mit dem Zusammenbau beginnen.
- **Montageschritte:** Wir demonstrieren jeden Montageschritt in einer systematischen Weise.
- **Tipps und Überlegungen:** Während des gesamten Prozesses teilen wir wesentliche Tipps und Tricks mit Ihnen, um Ihnen zu helfen, häufige Fehler zu vermeiden und sicherzustellen, dass Ihr Auto reibungslos funktioniert.
- **Testen und Kalibrieren:** Nach Abschluss der Montage führen wir Sie durch einen kurzen Test dieses Autos, um sicherzustellen, dass Ihre Montage fehlerfrei ist.

1.2 Spielmodus

Im Spielmodus wird der Code bereits im Werk auf das Arduino-Board hochgeladen, und der Zeus Car kann per APP oder Fernbedienung ferngesteuert werden, sobald er eingeschaltet ist.

- *Mit der APP steuern*: Sie müssen den SunFounder Controller auf Ihrem Mobiltelefon oder Tablet installieren, mit dem Sie den Zeus Car mit einem Joystick-Widget in alle Richtungen steuern können, sowie über Sprachsteuerung. Am wichtigsten ist, dass Sie das Echtzeit-Videostreaming, das vom Zeus Car aufgenommen wird, auf der APP sehen können.
- *Fernsteuerung*: Wir haben eine einfache Fernbedienung beigelegt, schalten Sie einfach den Zeus Car ein und Sie können ihn mit der Fernbedienung steuern. Dies ist eine gute Wahl für Benutzer, die so schnell wie möglich den Spaß am Steuern des Zeus Car erleben möchten.

Bemerkung: Hier ist das Code-Paket für dieses Kit.

- SunFounder Zeus Car Kit für Arduino

Wenn Sie nach dem Hochladen eines anderen Codes auf Ihren Arduino in den **Spielmodus** zurückkehren möchten, müssen Sie zuerst *Bibliotheken installieren (Arduino IDE 2)* durchführen, und dann die Datei Zeus_Car.ino, die unter dem Pfad zeus-car-main\Zeus_Car zu finden ist, auf das Arduino-Board hochladen.

1.2.1 Mit der APP steuern

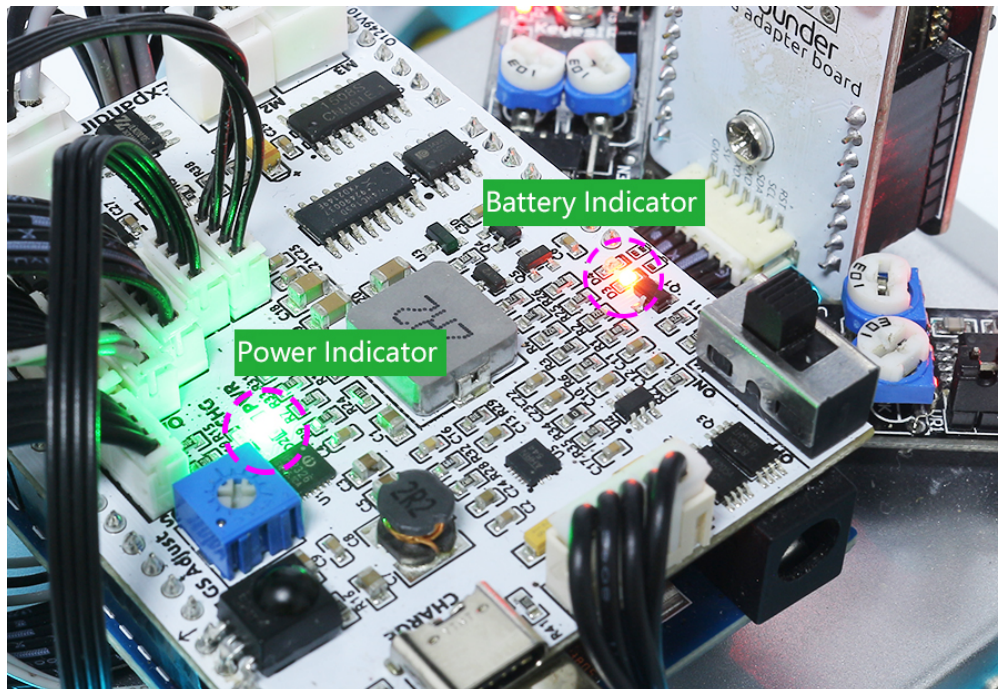
Um das Zeus Car über Ihr Mobilgerät zu steuern, müssen Sie die SunFounder Controller APP herunterladen, sich mit dem Zeus Car LAN verbinden und dann Ihren eigenen Controller in der APP erstellen.

Schnellanleitung

Dies ist ein kurzes Tutorial-Video. Bitte sehen Sie sich zuerst das Video an und befolgen Sie dann die untenstehenden Anweisungen.

1. Beginnen Sie mit dem Zeus Car.

- Bei der ersten Verwendung oder wenn das Batteriekabel abgezogen wird, aktiviert der Zeus Car Shield seine Überentladungsschutzschaltung.
- Daher müssen Sie das Type-C-Kabel für etwa 5 Sekunden anschließen.
- Wenn die Betriebsanzeige leuchtet, bedeutet dies, dass der Schutzstatus freigegeben wurde. Schauen Sie zu diesem Zeitpunkt auf die Batterieanzeigen. Wenn beide Batterieanzeigen aus sind, stecken Sie bitte weiterhin das Type-C-Kabel ein, um die Batterie aufzuladen.

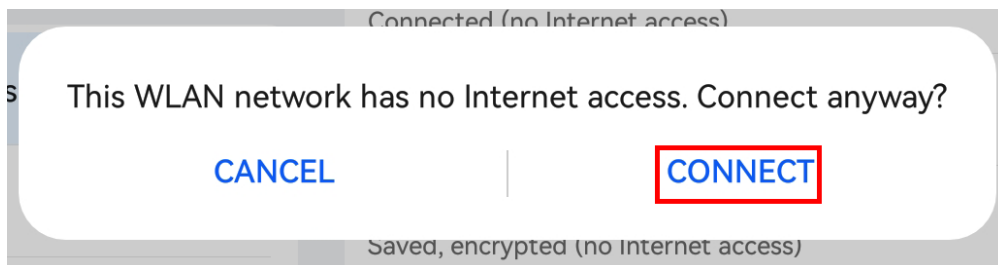


- Nun müssen Sie den kleineren Schalter nach rechts kippen, um die Kommunikation zwischen dem Auto und der ESP32 CAM herzustellen. Drücken Sie danach die Reset-Taste, um den Code neu zu starten. Zu diesem Zeitpunkt werden Sie beobachten, wie die Unterbodenleuchten von Orange zu Hellblau wechseln.

2. Installieren Sie [SunFounder Controller](#) aus dem **APP Store(iOS)** oder **Google Play(Android)**.
3. Verbinden Sie sich mit dem Zeus_Car WLAN.

Verbinden Sie nun Ihr Mobilgerät mit dem lokalen Netzwerk (LAN), das vom Zeus Car bereitgestellt wird. Auf diese Weise befinden sich Ihr Mobilgerät und der Zeus Car im selben Netzwerk, was die Kommunikation zwischen den Anwendungen auf Ihrem Mobilgerät und dem Zeus Car erleichtert.

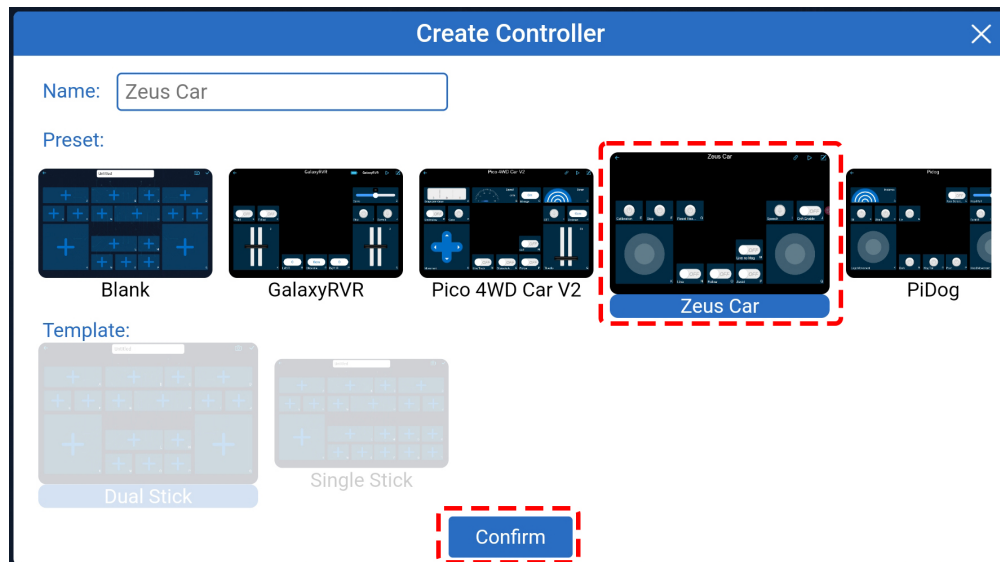
- Finden Sie Zeus_Car im WLAN Ihres Mobiltelefons (Tablet), geben Sie das Passwort 12345678 ein und verbinden Sie sich damit.
- Der Standardverbindungsmodus ist der AP-Modus. Nach der Verbindung erhalten Sie daher eine Meldung, dass dieses WLAN-Netzwerk keinen Internetzugang hat. Bitte wählen Sie „Weiter verbinden“.



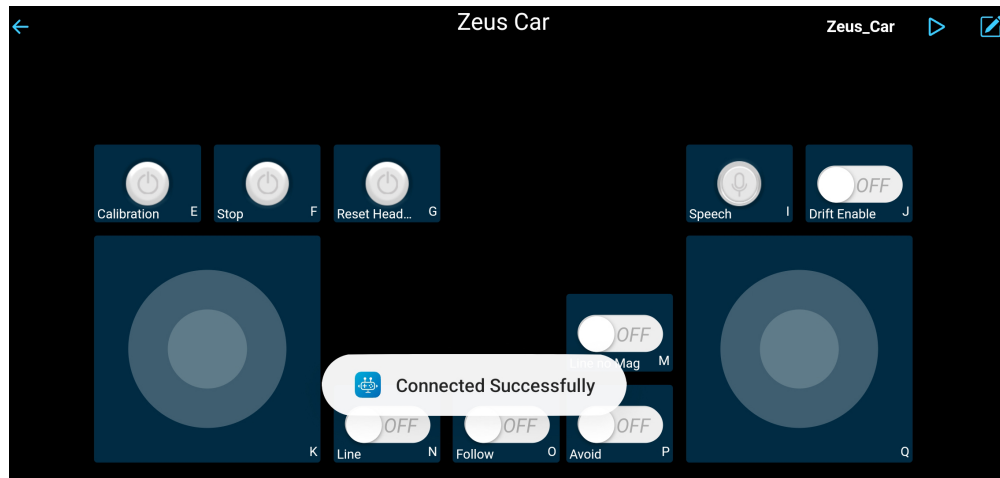
4. Erstellen Sie einen Controller.
 - Um auf dem SunFounder Controller einen Controller hinzuzufügen, klicken Sie auf das + Symbol.




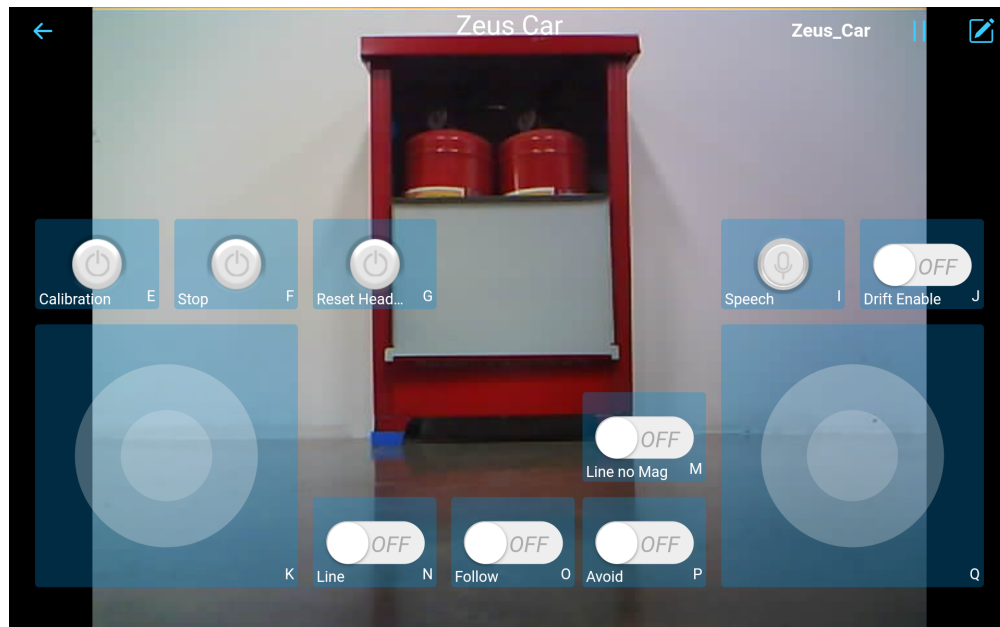
- Voreingestellte Controller sind für einige Produkte verfügbar. Hier wählen wir **Zeus Car**. Geben Sie ihm einen Namen oder tippen Sie einfach auf **Confirm**.



- Einmal drinnen, wird die App automatisch nach dem Zeus Car suchen. Nach einer Weile erscheint eine Meldung, die besagt „Connected Successfully“.



- Tippen Sie nun auf die  Schaltfläche. Dadurch können Sie das Live-Video von der Kamera anzeigen und das Auto mit den bereitgestellten Widgets steuern.

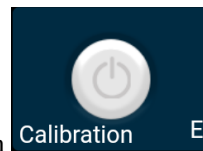


5. Hier sind die Funktionen der Widgets.

- **Calibration(E)**: Kompasskalibrierung einschalten.
- **Stop(F)**: Stoppen Sie alle Bewegungen des Autos.
- **Reset Heading(G)**: Nachdem Sie das Auto manuell in eine Richtung ausgerichtet haben, klicken Sie auf dieses Widget, um diese Richtung als Vorderseite der Autobewegung festzulegen. Damit können Sie schnell eine Richtung angeben, anstatt das Auto langsam mit anderen Widgets in diese Richtung zu drehen.
- **Speech(I)**: Wechseln Sie in den Sprachsteuerungsmodus.
- **Drift Enable(J)**: Aktivieren Sie die Drift-Funktion.
- **Move in All Directions(K)**: Steuern Sie das Auto in alle Richtungen.
- **Line Track**: Die folgenden beiden Widgets können beide in den Linienverfolgungsmodus wechseln.

- **Linie ohne Mag(M)**: Wechseln Sie in den Linienverfolgungsmodus, der nicht vom Magnetfeld beeinflusst wird. Während des Linienverfolgungsvorgangs wird sich die Ausrichtung des Zeus Car ständig ändern.
- **Linie(N)**: Wechseln Sie in den Linienverfolgungsmodus. Aufgrund des Vorhandenseins eines Magnetfelds wird sich die Ausrichtung des Zeus Car während der Linienverfolgung in eine bestimmte Richtung orientieren.
- *Follow(O)*: Wechseln in den Verfolgungsmodus.
- *Avoid(P)*: Wechseln in den Hindernisvermeidungsmodus.
- *Control the Direction(Q)*: Dient zur Steuerung der Kopfrichtung.

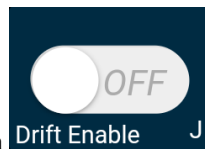
Calibration(E)



Aktivieren Sie die Kompasskalibrierung, indem Sie auf den Calibration E Button klicken.

Stellen Sie das Zeus-Auto auf den Boden. Nach Aktivierung der Kompasskalibrierung wird das Auto gegen den Uhrzeigersinn zu drehen beginnen und nach etwa 1 Minute anhalten. Wenn es länger als 2 Minuten dreht, ist das Magnetfeld hier komplex. Versuchen Sie, den Standort zu wechseln und erneut zu kalibrieren.

Drift Enable(J)




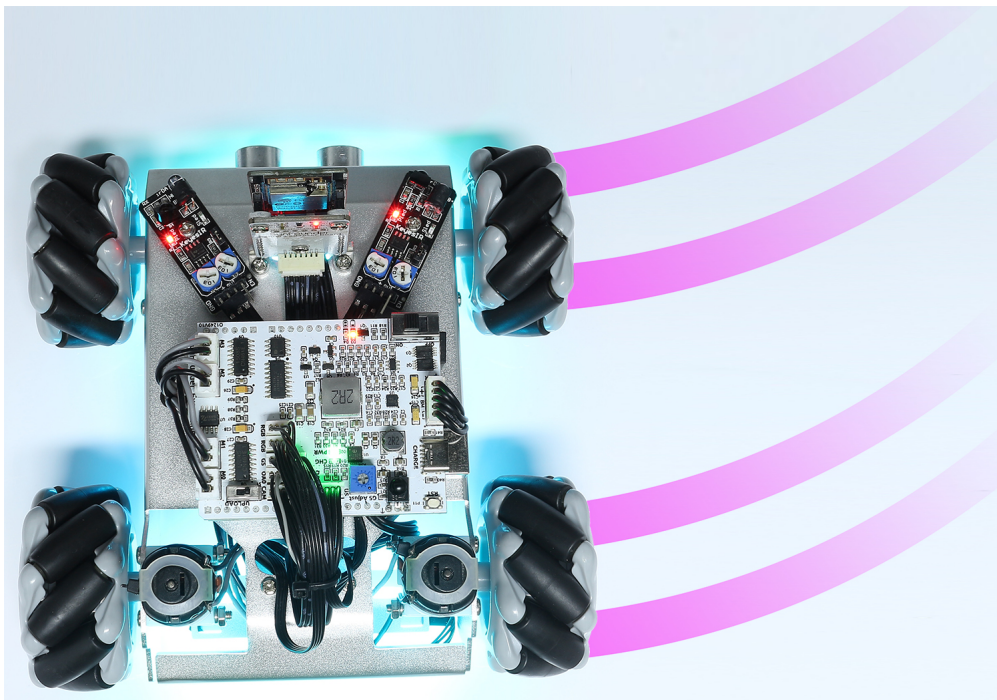
Klicken Sie auf den Drift Enable J Button, um die Drift-Funktion zu aktivieren.



- Wenn Sie das Drift control Q Widget gegen den Uhrzeigersinn schieben, wird das Zeus Auto nach rechts driften. Beim Loslassen der Hand wird das Auto an seiner aktuellen Position anhalten.



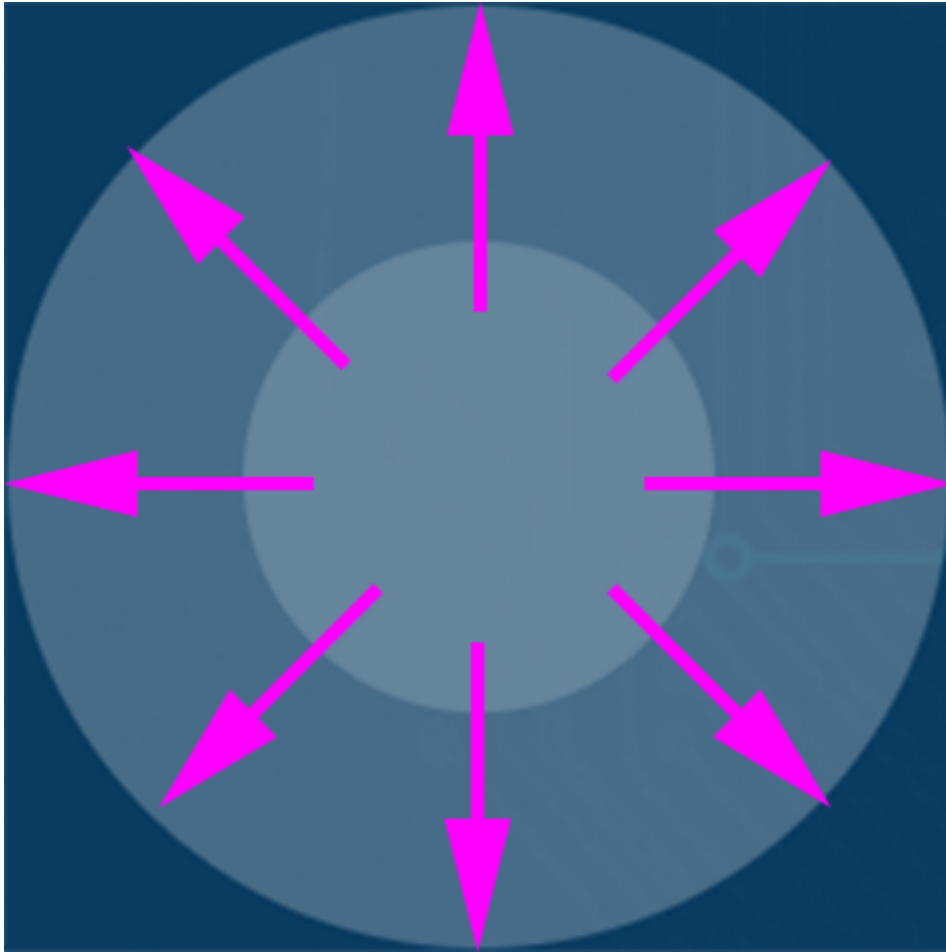
- Entsprechend, wenn Sie das  Widget im Uhrzeigersinn schieben, wird das Zeus Auto nach links driften und an der aktuellen Position anhalten.



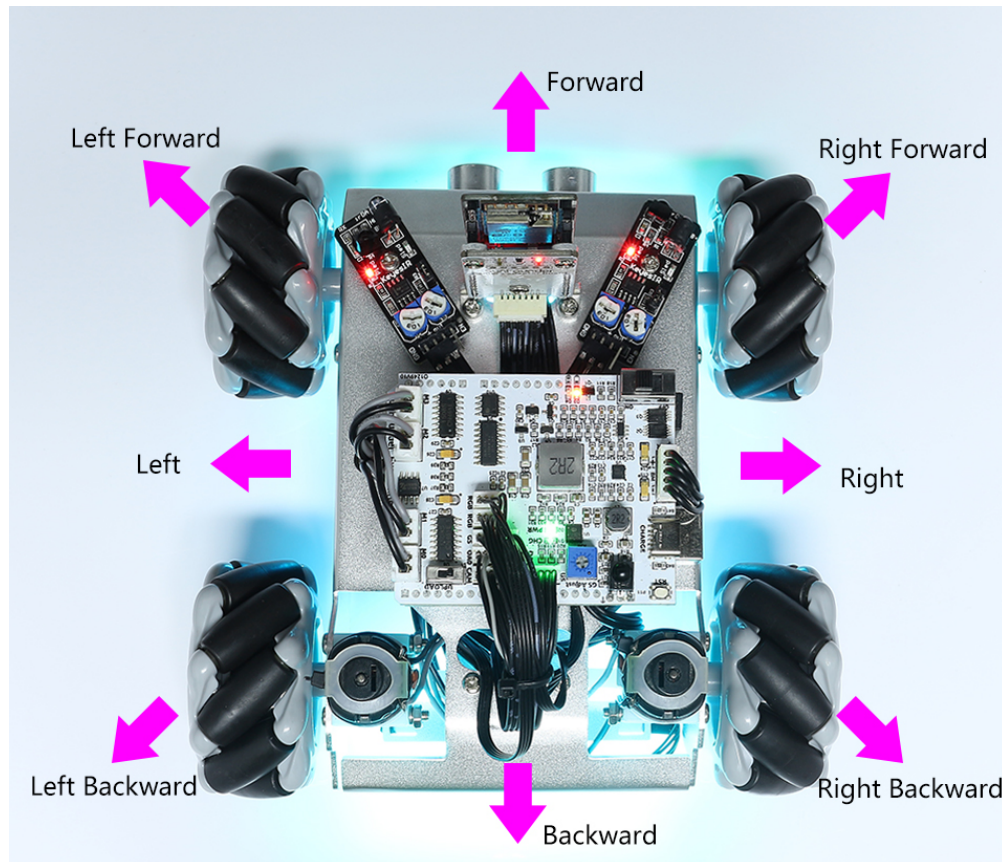
Move in All Directions(K)



Das Zeus Auto bewegt sich in die entsprechende Richtung, wenn Sie das K Widget wischen.



Das Auto bewegt sich jedes Mal, wenn Sie wischen. Wenn Sie Ihre Hand also nicht loslassen, bewegt sich das Auto weiter.



Speech(I)

Warnung: Android-Geräte können die Sprachsteuerungsfunktion nicht nutzen. Die Sprachsteuerung erfordert eine Internetverbindung des Android-Geräts und die Installation der Google Service Komponente.

Während iOS-Geräte einen Offline-Spracherkennungsmotor verwenden, ist keine Netzwerkverbindung erforderlich; AP- und STA-Modusverbindung sind beide verfügbar.

Das Zeus Auto kann auch über Sprache im SunFounder Controller gesteuert werden. Das Zeus Auto führt die eingestellten Aktionen basierend auf den Befehlen aus, die Sie Ihrem Mobilgerät sagen.

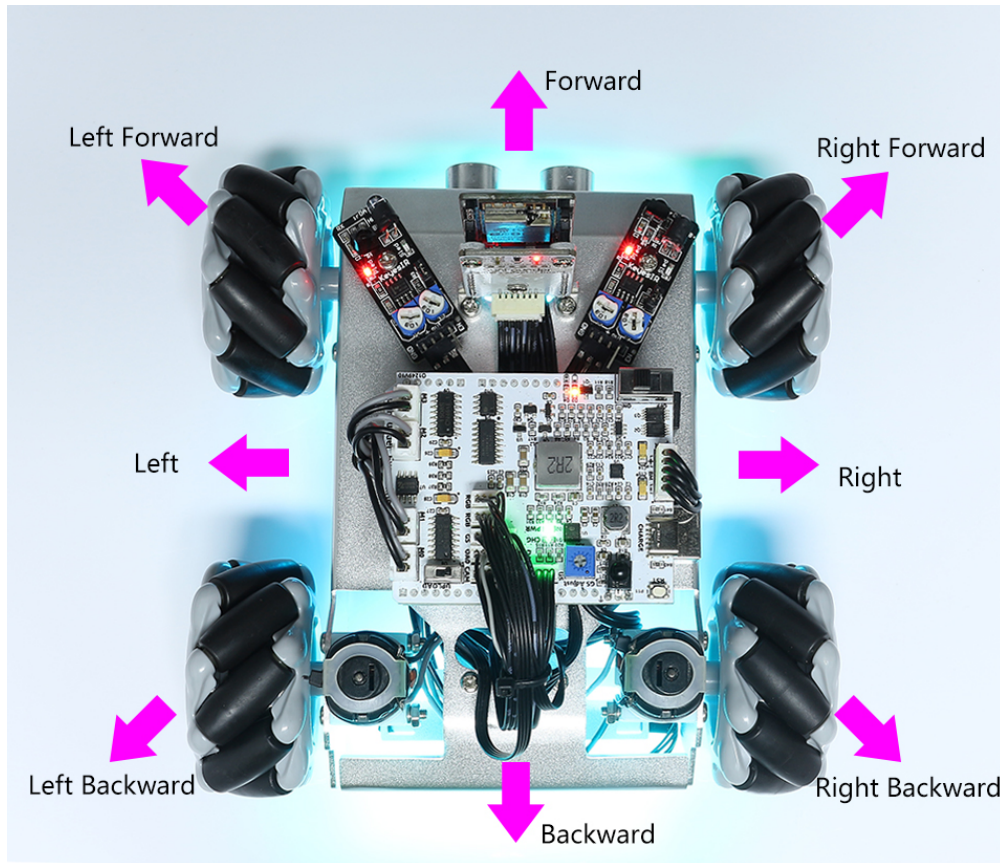


Halten Sie nun das **Speech M** Widget gedrückt und sagen Sie einen der folgenden Befehle, um zu sehen, was passiert.

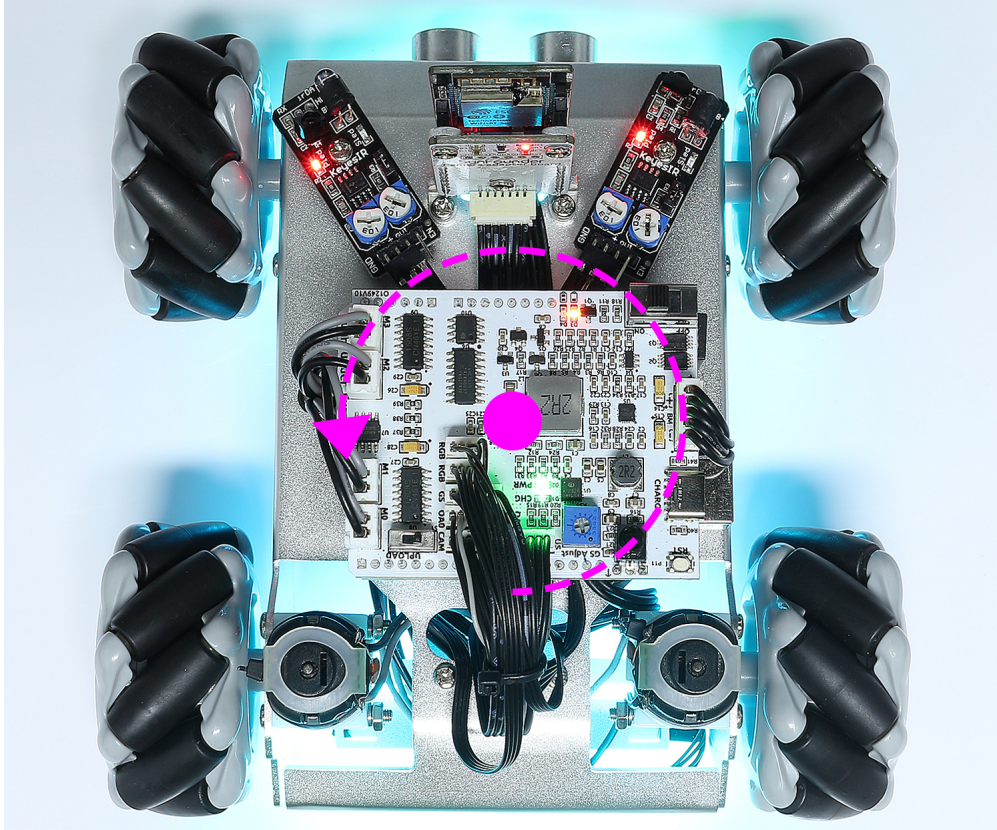
- stop: Alle Bewegungen des Autos können gestoppt werden.
- pause: Die Funktion ähnelt der von Stop, aber wenn die Vorderseite des Autos nicht in die ursprünglich eingestellte Richtung zeigt, wird es sich langsam in die eingestellte Richtung bewegen.
- forward
- backward

- left forward
- left backward
- right forward
- right backward
- move left
- move right

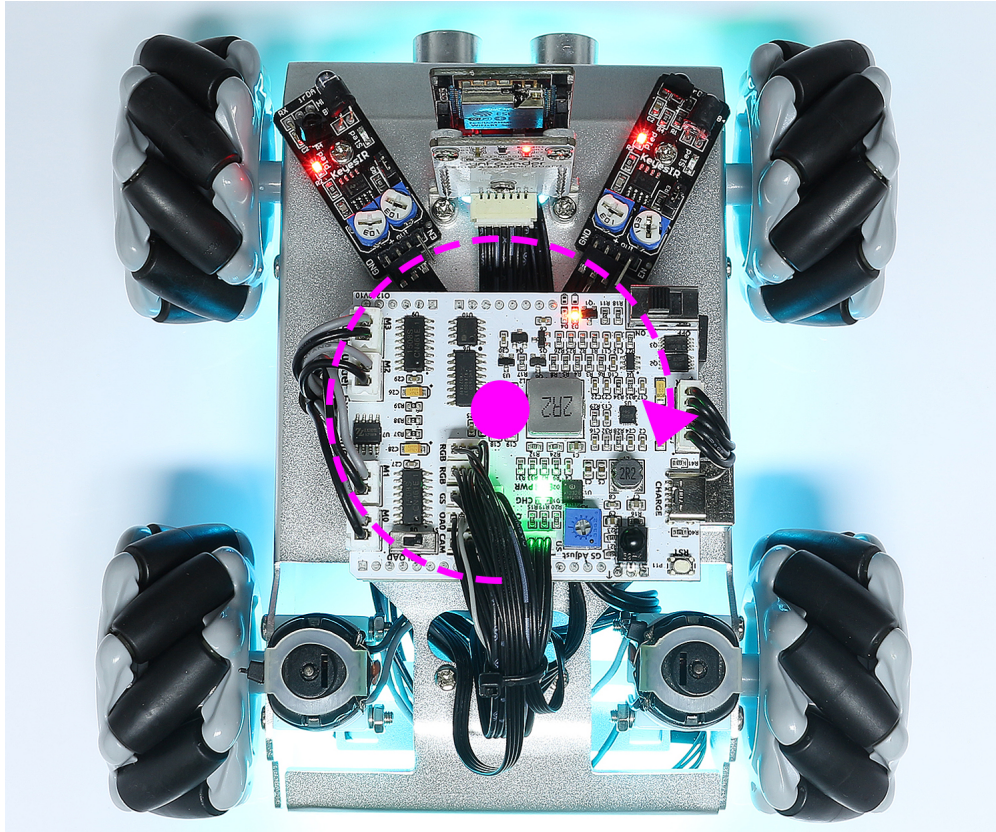
Nachdem das Auto die oben genannten 8 Befehle erhalten hat, wird es sich in die entsprechende Richtung bewegen, bis es die Befehle stop oder pause erhält.



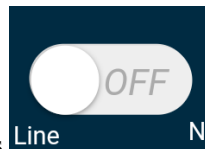
- links drehen: Dieser Befehl lässt das Auto sich um 45° nach links drehen, wobei der Körper als Mittelpunkt dient. Anschließend wird es je nach vorherigem Zustand vorwärts fahren oder anhalten. Wenn der vorherige Zustand „stop“ war, wird es nach 45° Drehung nach links anhalten; wenn es „vorwärts“ war, wird es nach dem Drehen vorwärts fahren.



- **rechts drehen:** Dieser Befehl lässt das Auto sich um 45° nach rechts drehen, wobei der Körper als Mittelpunkt dient, und wird je nach vorherigem Zustand vorwärts fahren oder anhalten.



Line Track

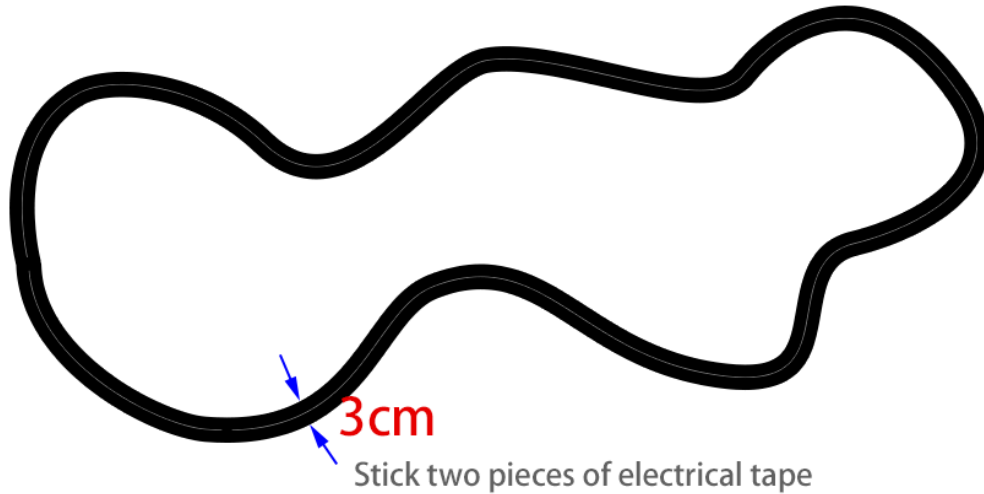


Klicken Sie auf das **Line N** Widget, um in den Linienverfolgungsmodus zu wechseln.

Der Zeus Car bietet zwei Modi der Linienverfolgung an: Einen, bei dem die Vorderseite immer in die Bewegungsrichtung zeigt und einen, bei dem die Vorderseite in eine feste Richtung zeigt. Hier wurde der zweite Modus gewählt.

1. Kleben Sie eine 3 cm breite Linie an

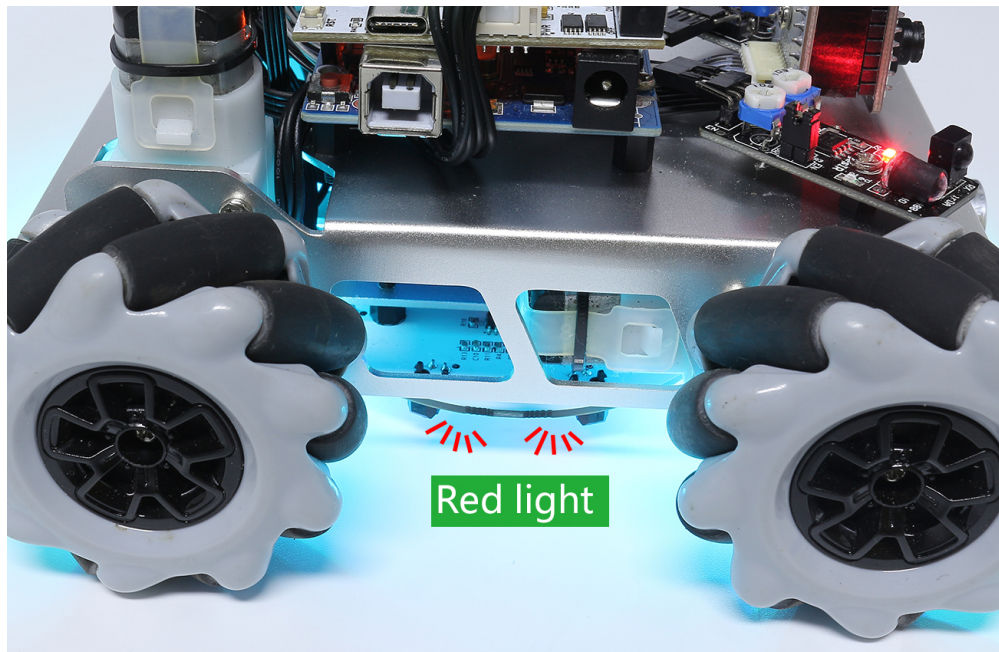
Es gibt acht Sensoren am Omni-Graustufenmodul, und der Abstand zwischen jedem Sensor liegt zwischen 2 und 3 cm. Mindestens zwei Sensoren müssen gleichzeitig die schwarze Linie erkennen. Daher muss die angebrachte Linie mindestens 3 cm breit sein, und der Biegungswinkel sollte nicht kleiner als 90° sein.



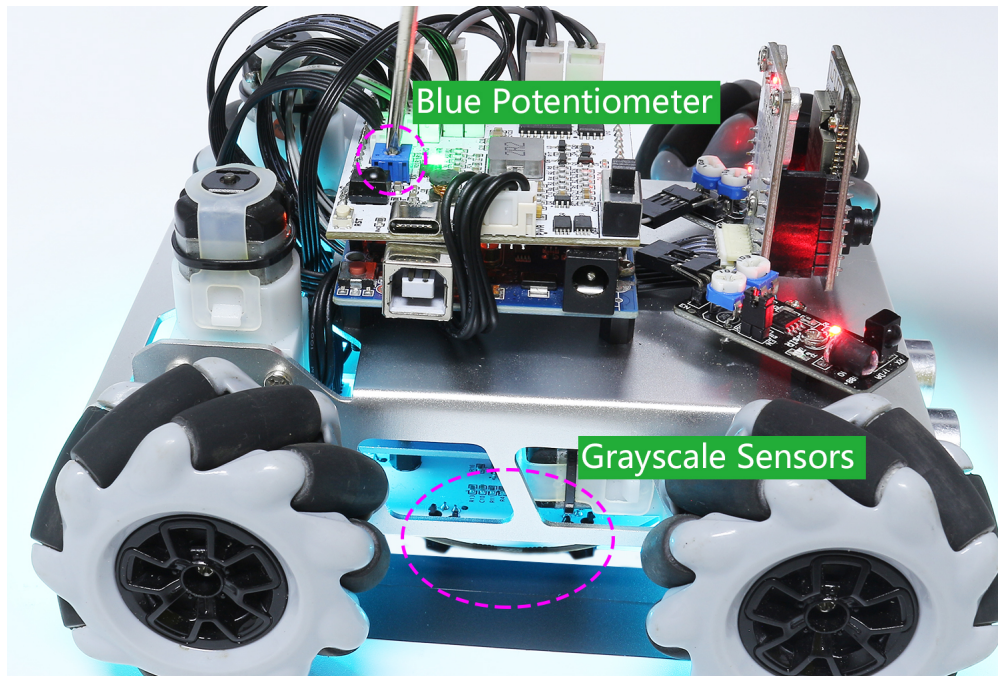
2. Kalibrieren Sie das Omni-Graustufenmodul.

Da jede Untergrundfarbe unterschiedliche Grauwerte aufweist, ist der werkseitig eingestellte Graustufen-Schwellenwert möglicherweise nicht für Ihre aktuelle Umgebung geeignet. Daher müssen Sie dieses Modul vor dem Gebrauch kalibrieren. Es wird empfohlen, die Kalibrierung durchzuführen, wenn sich die Bodenfarbe deutlich ändert.

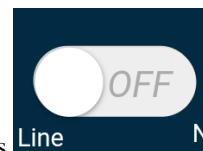
- Stellen Sie den Zeus Car auf eine weiße Oberfläche und drehen Sie das Potentiometer, bis das Grausensorlicht gerade leuchtet.



- Lassen Sie nun die beiden Graustufensensoren an der Seite genau zwischen der schwarzen Linie und der weißen Oberfläche positionieren und drehen Sie das Potentiometer langsam, bis die Signalanzeige gerade erlischt.

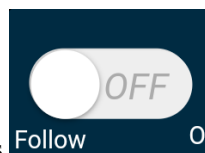


- Sie können mehrfach über die schwarze Linie und die weiße Oberfläche fahren, um sicherzustellen, dass die Lichter des Graustufensensors aus sind, wenn sie zwischen der schwarzen Linie und der weißen Oberfläche und an sind, wenn sie auf der weißen Oberfläche sind. Das zeigt an, dass das Modul erfolgreich kalibriert wurde.



3. Stellen Sie den Zeus Car auf Ihre angebrachte Linie, klicken Sie auf das Line N Widget, und er wird der Linie folgen.
4. Aufgrund der hohen Umgebungsanforderungen des Omni-Graustufenmoduls wird empfohlen, es mehrmals zu kalibrieren, wenn der Verfolgungseffekt nicht zufriedenstellend ist (außerhalb der Spur).

Follow(O)

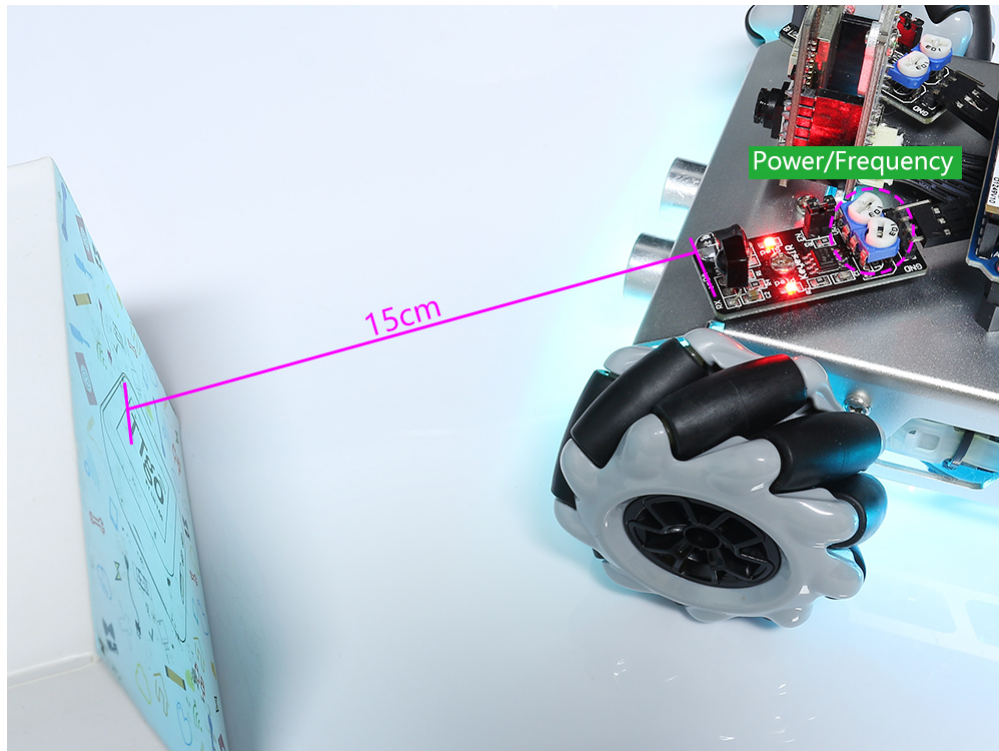


Klicken Sie auf das Follow O Widget, um in den Verfolgungsmodus zu wechseln.

Der Ultraschallsensor erkennt Hindernisse vor sich (20 cm) und folgt ihnen. Diese beiden Hindernisvermeidungsmodule ermöglichen es dem Auto, nach links oder rechts zu folgen, müssen jedoch vor der Verwendung kalibriert werden (15cm).

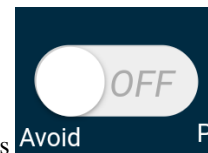
1. Kalibrieren Sie das IR-Hindernisvermeidungsmodul.
 - Platzieren Sie ein Hindernis etwa 15 cm entfernt vom IR-Hindernisvermeidungsmodul.
 - Am Modul befinden sich zwei Potentiometer, eines zur Anpassung der Sendeleistung und eines zur Anpassung der Sendefrequenz. Mit diesen beiden Potentiometern können Sie den Erfassungsabstand einstellen.

- Dann können Sie ein Potentiometer einstellen, und wenn bei 15 cm das Signallight auf dem Modul leuchtet, ist die Einstellung erfolgreich. Wenn nicht, stellen Sie das andere Potentiometer ein.



- Kalibrieren Sie das andere Hindernisvermeidungsmodul auf die gleiche Weise.
2. Stellen Sie das Zeus-Auto auf einen Tisch oder den Boden und lassen Sie es Ihrer Hand oder anderen Hindernissen folgen.



Avoid(P)

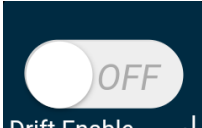



Wenn Sie in den Hindernisvermeidungsmodus wechseln möchten, klicken Sie auf das **Avoid P** Widget, ziehen Sie aber zuerst den *Follow(O)* heran, um die beiden Hindernisvermeidungsmodule zu kalibrieren.

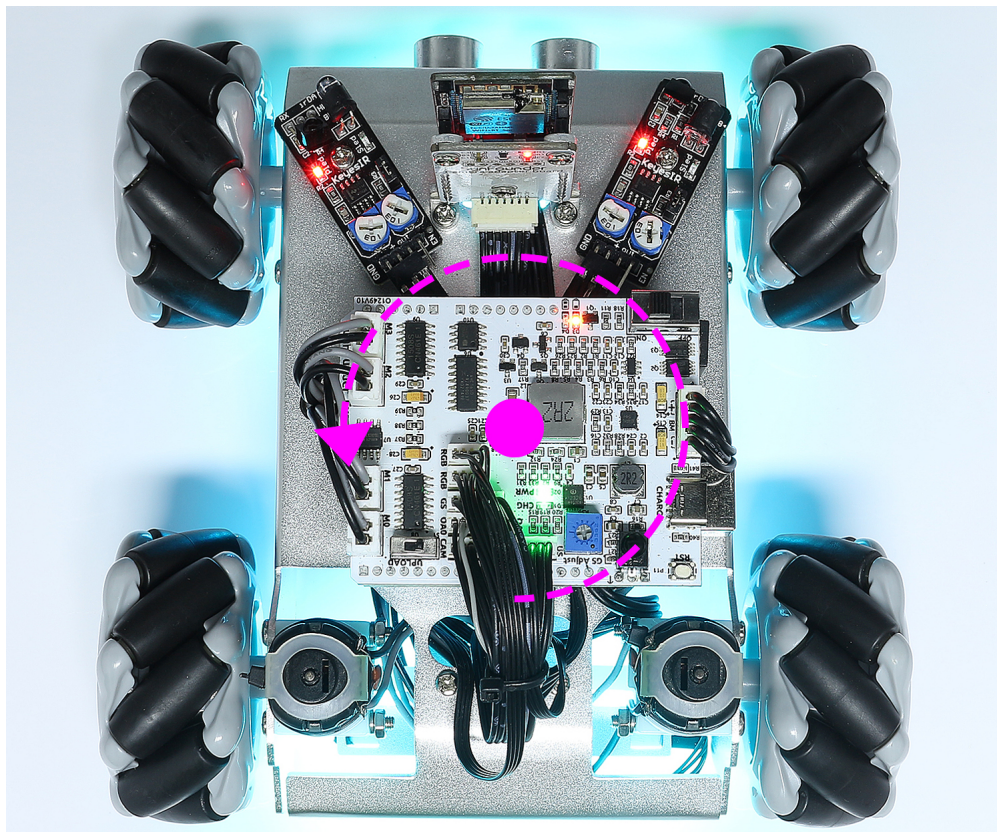
- Das Zeus Auto wird sich vorwärts bewegen.
- Ein Ultraschallmodul erkennt Hindernisse vor sich. Wenn es eines erkennt, dreht sich das Auto nach links.
- Wenn das linke Hindernisvermeidungsmodul ein Hindernis erkennt, dreht sich das Auto nach rechts, und wenn das rechte Hindernisvermeidungsmodul ein Hindernis erkennt, dreht es sich nach links.

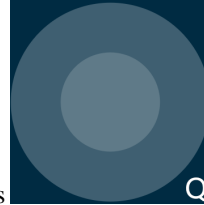
Control the Drection(Q)

- Wenn der  Button aktiviert ist, wird das  Widget verwendet, um das Zeus Auto links und rechts driften zu lassen.

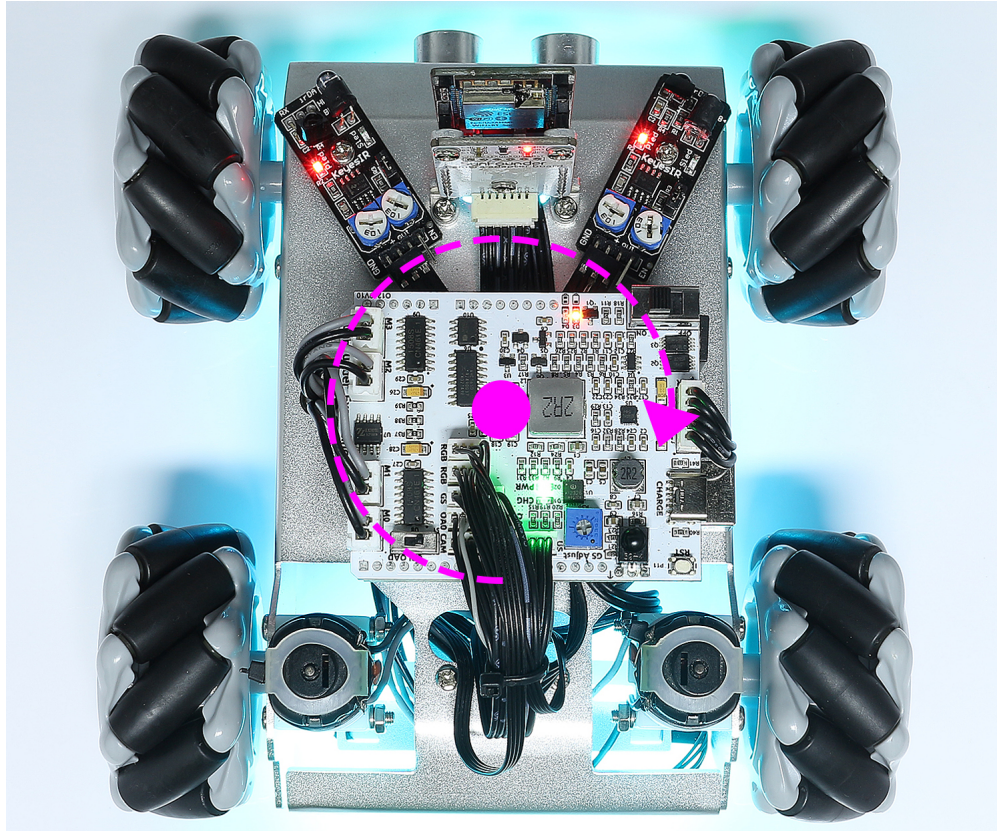
- Wenn das  Widget deaktiviert ist, wird das  Widget verwendet, um die Richtung des Auto-Kopfes zu steuern.

- Durch das Verschieben des  Widgets gegen den Uhrzeigersinn wird auch das Auto gegen den Uhrzeigersinn drehen. Beim Loslassen kehrt der Kopf des Autos in die ursprüngliche Richtung zurück.





- Ebenso wird das Auto im Uhrzeigersinn drehen, wenn das Widget im Uhrzeigersinn verschoben wird und in die ursprüngliche Richtung zurückkehren, wenn es losgelassen wird.



1.2.2 Fernsteuerung

Das Zeus-Auto kann mit einer Fernbedienung gesteuert werden, indem man das Auto einschaltet und dann direkt die Tasten der Fernbedienung drückt. Diese Methode ist stabil und weniger anfällig für äußere Störungen, bietet jedoch möglicherweise nicht so flexible Steuerungsmöglichkeiten.

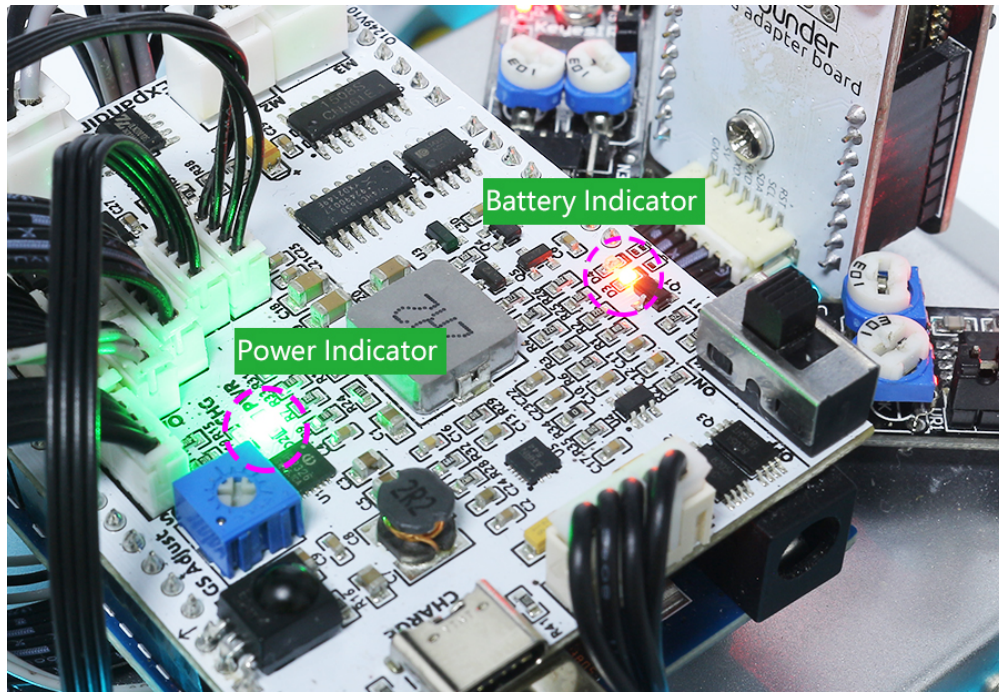
Kurzanleitung

Hier ist ein kurzes Tutorial-Video. Bitte sehen Sie sich zuerst das Video an und folgen Sie dann den untenstehenden Anweisungen.

1. Beginnen wir mit dem Zeus Car.

- Bei der ersten Verwendung oder wenn das Batteriekabel ausgesteckt wird, wird die Überentladungsschutzschaltung des Zeus Car Shield aktiviert.
- Daher müssen Sie das Type-C-Kabel für ungefähr 5 Sekunden einstecken.

- Wenn die Stromanzeige leuchtet, bedeutet dies, dass der Schutzstatus aufgehoben wurde. Schauen Sie zu diesem Zeitpunkt auf die Batterieanzeigen. Wenn beide Batterieanzeigen ausgeschaltet sind, stecken Sie das Type-C-Kabel weiterhin ein, um die Batterie zu laden.



- Nun müssen Sie den kleineren Schalter nach rechts kippen, um die Kommunikation zwischen dem Auto und der ESP32 CAM herzustellen. Drücken Sie anschließend die Reset-Taste, um den Code neu zu starten. Zu diesem Zeitpunkt werden Sie beobachten, wie die Unterbodenleuchten von Orange zu einem hellen Blau wechseln.
2. Drücken Sie die verschiedenen Tasten auf der Fernbedienung, um das Auto zu steuern.



- **Stop:** Stoppen Sie alle Bewegungen des Autos.
- *Compass Calibration:* Kompasskalibrierung aktivieren.
- *Line Track:* Wechseln zum Linienverfolgungsmodus.
- *Follow:* Wechseln zum Verfolgungsmodus.
- *Obstacle Avoidance:* Wechseln zum Hindernisvermeidungsmodus.
- *Drift Left/Right:* Nach links/rechts driften.
- **Set Heading:** Nachdem Sie das Auto mit der Hand in eine Richtung gestellt haben, klicken Sie auf diese Taste, um diese Richtung als Vorderseite der Autobewegung festzulegen. Dies ermöglicht es Ihnen, schnell eine Richtung anzugeben, anstatt das Auto langsam mit anderen Tasten in diese Richtung zu drehen.
- *Rotate Left/Right(Cycle/USD Key):* Nach links/rechts drehen
- *Move in All Directions(1 ~ 9):* Steuern Sie das Auto in alle Richtungen.
- **Pause:** Die Funktion ähnelt Stopp, aber wenn das vordere Ende des Autos nicht in die ursprünglich festgelegte Richtung zeigt, wird es langsam in die festgelegte Richtung bewegt.

Compass Calibration



Schalten Sie die Kompasskalibrierung durch Drücken der -Taste ein.

Stellen Sie das Zeus-Auto auf den Boden. Nach dem Aktivieren der Kompasskalibrierung beginnt das Auto, sich gegen den Uhrzeigersinn zu drehen und stoppt nach ungefähr einer Minute. Wenn es länger als 2 Minuten rotiert, ist das Magnetfeld hier komplex. Versuchen Sie, den Standort zu wechseln und erneut zu kalibrieren.

Line Track



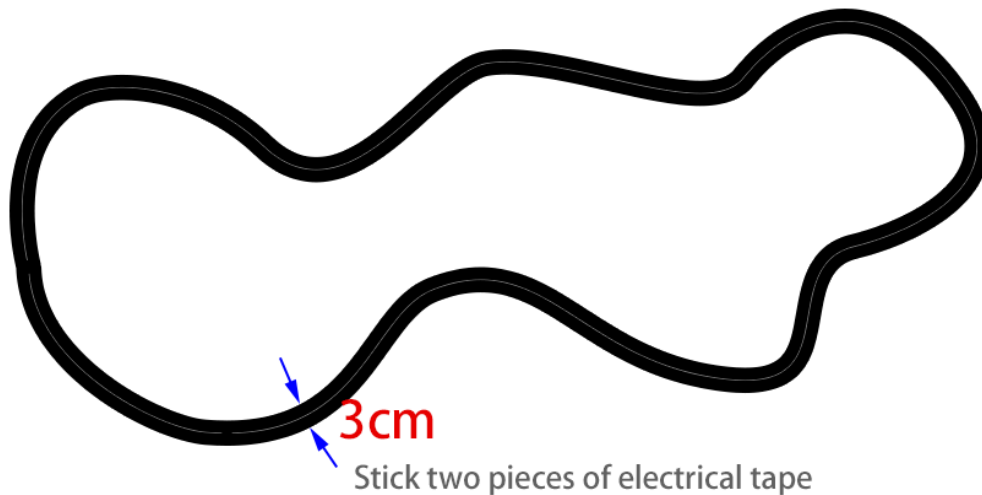
Drücken Sie die -Taste, um in den Linienverfolgungsmodus zu wechseln.

Es gibt zwei Linienverfolgungsmodi im Zeus-Auto: Ein Modus, bei dem der Kopf immer in Bewegungsrichtung zeigt, und ein Modus, bei dem der Kopf in eine feste Richtung zeigt. Hier ist der zweite Modus ausgewählt.

Bevor Sie die Linienverfolgung starten können, müssen Sie das Omni-Graustufenmodul kalibrieren und die Linie befestigen:

1. Kleben Sie eine 3 cm breite Linie:

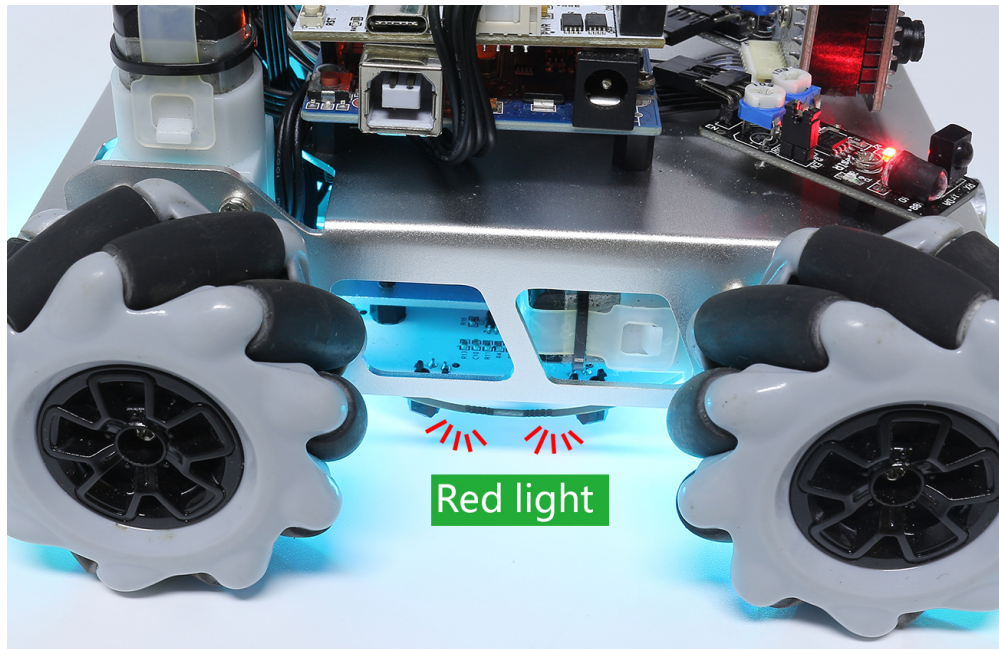
Es gibt acht Sensoren am Omni-Graustufenmodul und der Abstand zwischen den Sensoren liegt zwischen 2 und 3 cm. Es müssen zwei Sensoren gleichzeitig die schwarze Linie erkennen. Daher muss die angebrachte Linie mindestens 3 cm breit sein und der Biege winkel darf nicht weniger als 90° betragen.



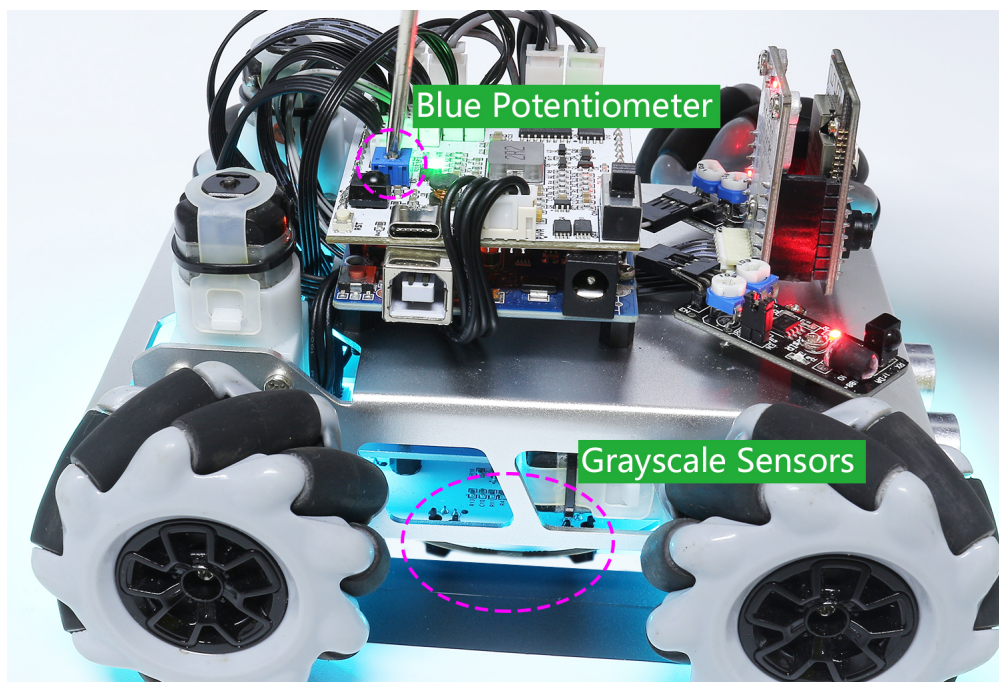
2. Kalibrieren Sie das Omni-Graustufenmodul:

Da jeder Unterboden unterschiedliche Graustufenwerte aufweist, ist der werkseitig eingestellte Graustufenschwellenwert möglicherweise nicht für Ihre aktuelle Umgebung geeignet. Daher müssen Sie dieses Modul vor der Verwendung kalibrieren. Es wird empfohlen, dies jedes Mal zu tun, wenn sich die Bodenfarbe stark ändert.

- Stellen Sie das Zeus-Auto auf eine weiße Fläche und drehen Sie das Potentiometer, bis das Licht des Grausensors gerade leuchtet.



- Positionieren Sie jetzt die beiden Grausensoren so, dass sie genau zwischen der schwarzen Linie und der weißen Oberfläche liegen, und drehen Sie das Potentiometer langsam, bis die Signalanzeige gerade erlischt.



- Bewegen Sie das Auto mehrfach über die schwarze Linie und die weiße Fläche, um sicherzustellen, dass die Lichter des Grausensors aus sind, wenn sie zwischen der schwarzen Linie und der weißen Fläche liegen, und eingeschaltet sind, wenn sie auf der weißen Fläche liegen. Dies zeigt an, dass das Modul erfolgreich kalibriert wurde.

3. Platzieren Sie das Zeus-Auto auf Ihrer markierten Linie, drücken Sie die



-Taste, und es wird der Linie

folgen.

4. Aufgrund der hohen Umgebungsanforderungen des Omni-Graustufenmoduls wird empfohlen, es mehrmals zu kalibrieren, falls der Verfolgungseffekt nicht zufriedenstellend ist (Abweichungen).

Follow

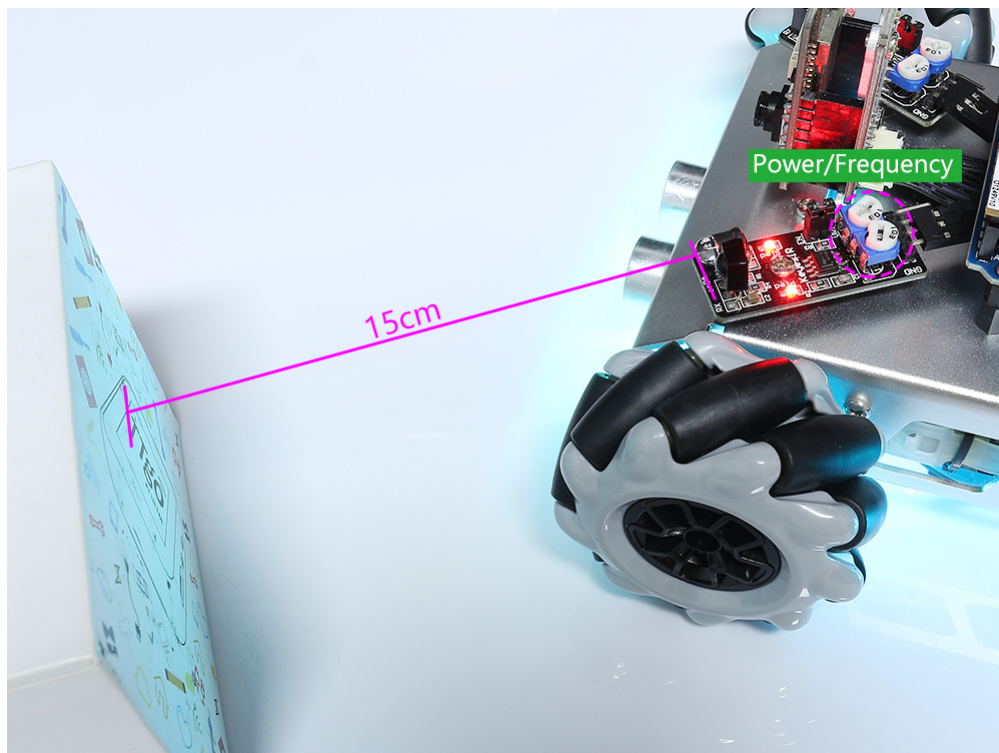


Drücken Sie die -Taste, um in den Folgemodus zu wechseln.

Der Ultraschallsensor erkennt Hindernisse vorne (20 cm) und folgt ihnen. Die zwei Hindernisvermeidungsmodule ermöglichen es dem Auto, nach links oder rechts zu folgen. Sie müssen jedoch vor der Verwendung kalibriert werden (15 cm).

1. Kalibrieren Sie das IR-Hindernisvermeidungsmodul:


- Beginnen Sie mit der Einstellung des rechten Hindernisvermeidungsmoduls. Während des Transports können Kollisionen dazu führen, dass der Sender und Empfänger im Infrarotmodul kippen. Daher müssen Sie sie manuell begradigen.
- Platzieren Sie ein Hindernis etwa 15 cm vom IR-Hindernisvermeidungsmodul entfernt.
- Am Modul befinden sich zwei Potentiometer: eines zur Einstellung der Sendeleistung und eines zur Einstellung der Sendefrequenz. Durch Anpassen dieser beiden Potentiometer können Sie die Erkennungsentfernung einstellen.
- Dann können Sie ein Potentiometer einstellen, und wenn bei 15 cm das Signallämpchen am Modul leuchtet, ist die Einstellung erfolgreich. Wenn nicht, stellen Sie das andere Potentiometer ein.



- Kalibrieren Sie das andere Hindernisvermeidungsmodul auf die gleiche Weise.



2. Stellen Sie das Zeus-Auto auf einen Tisch oder den Boden und lassen Sie es Ihrer Hand oder anderen Hindernissen folgen.


Obstacle Avoidance

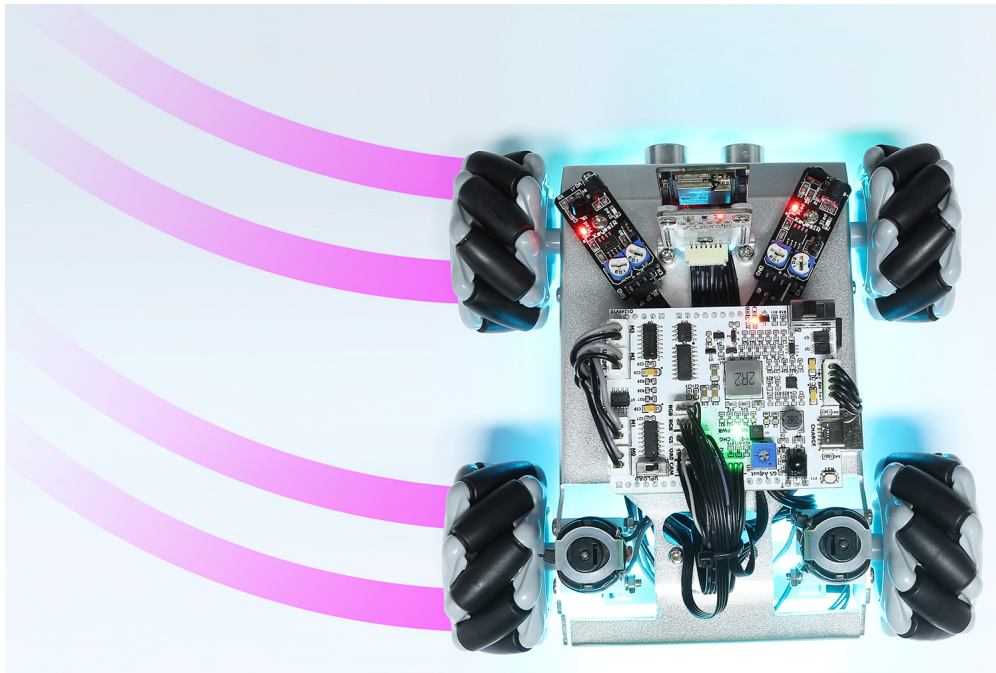
Wenn Sie in den Hindernisvermeidungsmodus wechseln möchten, drücken Sie die -Taste. Konsultieren Sie zuerst den Abschnitt [Follow](#), um die beiden Hindernisvermeidungsmodule zu kalibrieren.


- Das Zeus-Auto wird sich vorwärts bewegen.
- Ein Ultraschallmodul erkennt Hindernisse vorne. Bei Erkennung dreht sich das Auto nach links.
- Wenn das linke Hindernisvermeidungsmodul ein Hindernis erkennt, dreht sich das Auto nach rechts und bei Erkennung durch das rechte Modul nach links.

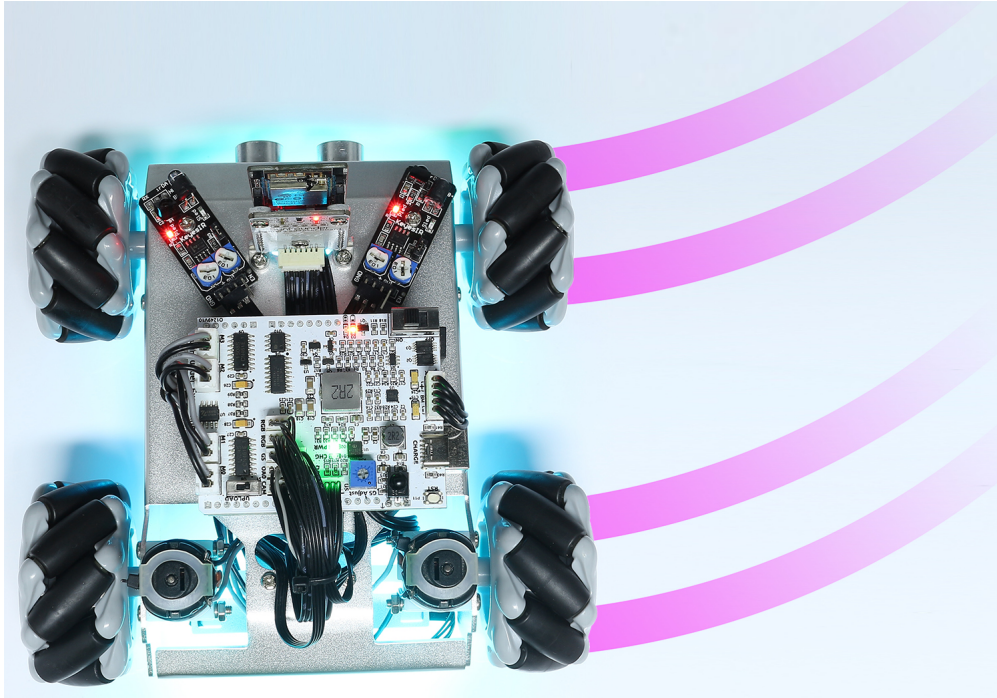
Drift Left/Right



Verwenden Sie die Tasten  und , um das Auto nach links/rechts driften zu lassen.

- Bei Betätigung der -Taste driftet das Auto nach links.



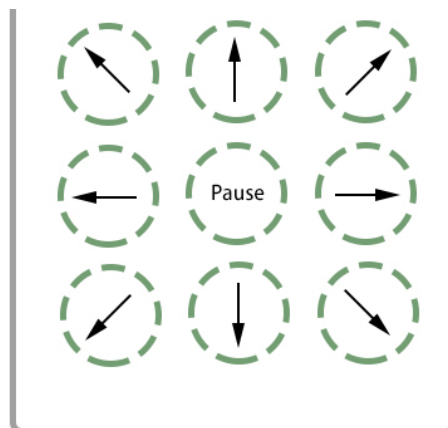
- Bei Betätigung der -Taste driftet das Auto nach rechts.



- Wenn das Auto vor dem Driften gestoppt wurde, bewirkt ein einmaliges Drücken von  oder  einen 90°-Drift des Zeus nach links oder rechts.
- Wenn das Auto vor dem Driften in Bewegung war, driftet es nach Betätigung der Taste 90° nach links oder rechts und fährt dann weiter, bis eine andere Taste gedrückt wird.

Move in All Directions(1 ~ 9)

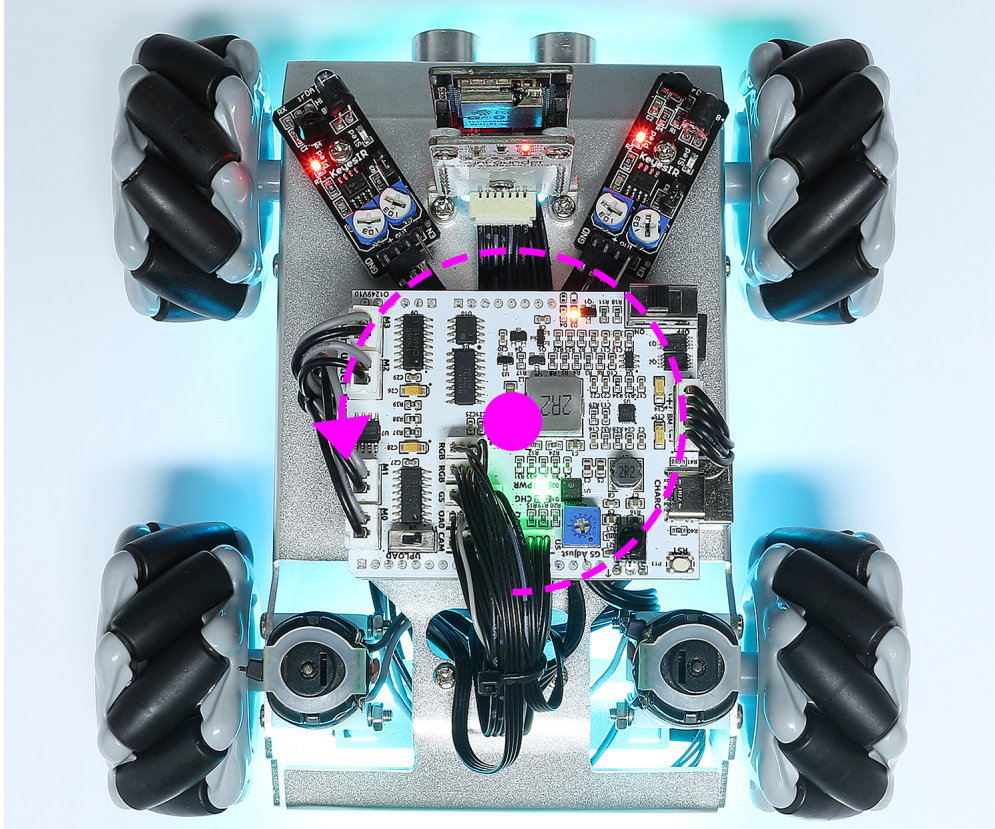
Verwenden Sie die Zahlentasten 1~9, um das Zeus Auto in 8 Richtungen zu steuern.



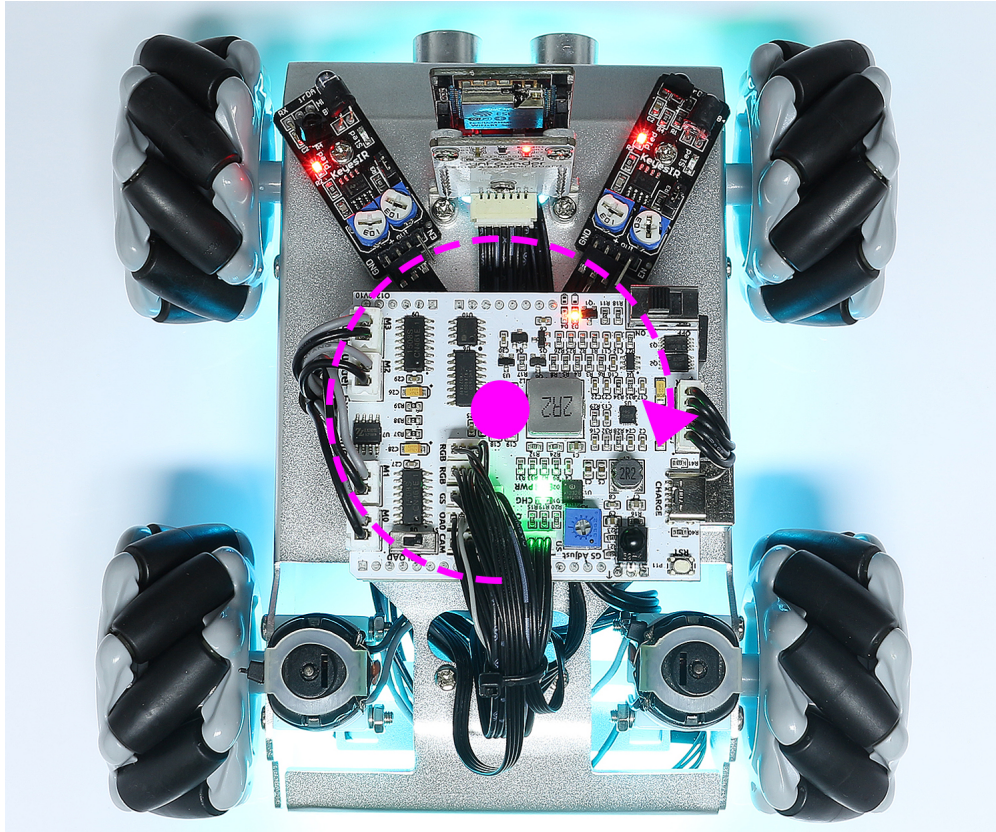
Rotate Left/Right(Cycle/USD Key)



- Wenn Sie die -Taste einmal drücken, dreht sich das Auto mit dem Körper als Zentrum um 45° nach links. Abhängig vom vorherigen Zustand wird es vorwärts fahren oder stoppen. Wenn es zuvor gestoppt wurde, wird es nach einer 45°-Linksdrehung anhalten; wenn es vorwärts gefahren ist, wird es nach der Linksdrehung weiterfahren.



- Ebenso wird das Auto durch einmaliges Drücken von um 45° nach rechts drehen und dann je nach vorherigem Zustand vorwärts fahren oder stoppen.



1.3 Programmiermodus

Nachdem Sie den *Spielmodus* erlebt haben und verstehen möchten, wie jede Funktion implementiert ist und dann den Effekt nach Ihren eigenen Ideen ändern möchten, sollten Sie diesen Programmiermodus sorgfältig durchlesen.

Jede Funktion im Kapitel Programmiermodus wird zu einem separaten Projekt gemacht, es gibt insgesamt 17 davon. Jedes Projekt hat einen entsprechenden Code und eine Erklärung, so dass Sie sehen können, wie jede Funktion funktioniert.

Für dieses Projekt wird die Arduino-Programmiersprache verwendet, daher müssen Sie zuerst die Arduino IDE und die zugehörigen Bibliotheken installieren, bevor Sie beginnen können.

Jetzt beginnen wir die Entdeckungsreise!

1.3.1 Den Code herunterladen

Laden Sie den relevanten Code über den untenstehenden Link herunter.

- [SunFounder Zeus Auto Kit für Arduino](#)

Oder sehen Sie sich den Code unter an.

1.3.2 Download und Installation von Arduino IDE 2.0

Die Arduino IDE, bekannt als Arduino Integrated Development Environment, bietet alle notwendigen Softwareunterstützungen, um ein Arduino-Projekt abzuschließen. Es handelt sich um eine speziell für Arduino entwickelte Programmiersoftware, die vom Arduino-Team bereitgestellt wird und mit der wir Programme schreiben und auf das Arduino-Board hochladen können.

Die Arduino IDE 2.0 ist ein Open-Source-Projekt. Sie stellt einen großen Schritt gegenüber ihrem soliden Vorgänger, der Arduino IDE 1.x, dar und bietet eine überarbeitete Benutzeroberfläche, verbesserten Board- & Library-Manager, Debugger, Autocomplete-Funktion und vieles mehr.

In diesem Tutorial zeigen wir, wie Sie die Arduino IDE 2.0 auf Ihrem Windows-, Mac- oder Linux-Computer herunterladen und installieren können.

Anforderungen

- Windows - Win 10 und neuer, 64 Bit
- Linux - 64 Bit
- Mac OS X - Version 10.14: „Mojave“ oder neuer, 64 Bit

Arduino IDE 2.0 herunterladen

1. Besuchen Sie .
2. Laden Sie die IDE für Ihre Betriebssystemversion herunter.



Arduino IDE 2.0.0

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

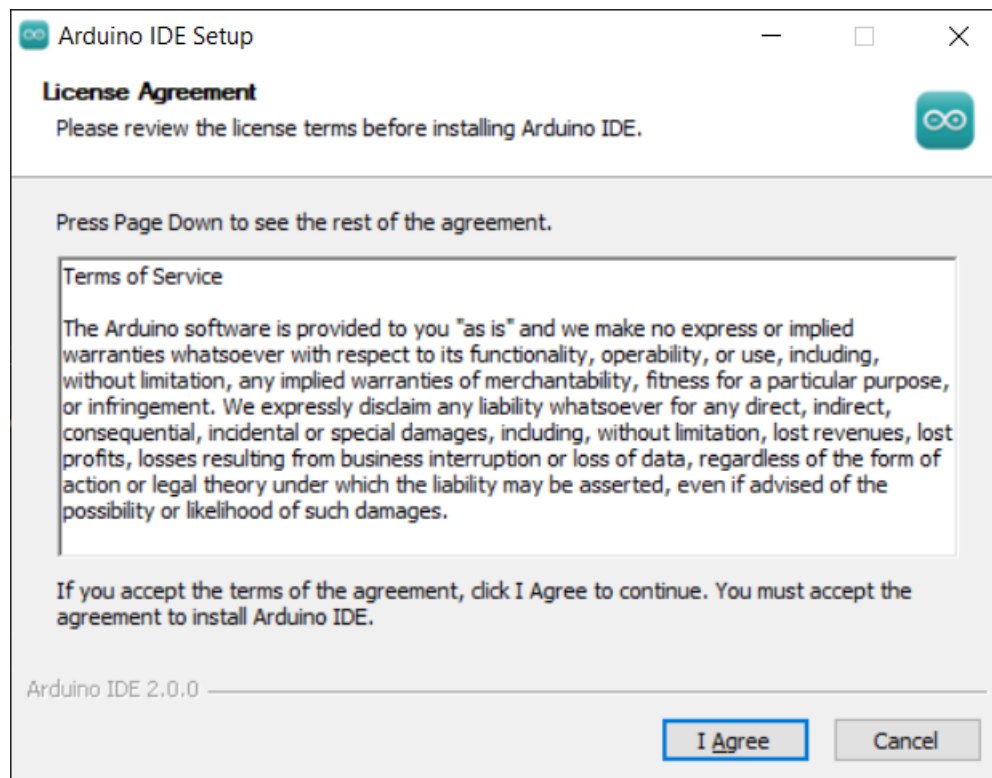
- Windows** Win 10 and newer, 64 bits
- Windows** MSI installer
- Windows** ZIP file
- Linux** AppImage 64 bits (X86-64)
- Linux** ZIP file 64 bits (X86-64)
- macOS** 10.14: "Mojave" or newer, 64 bits

Installation

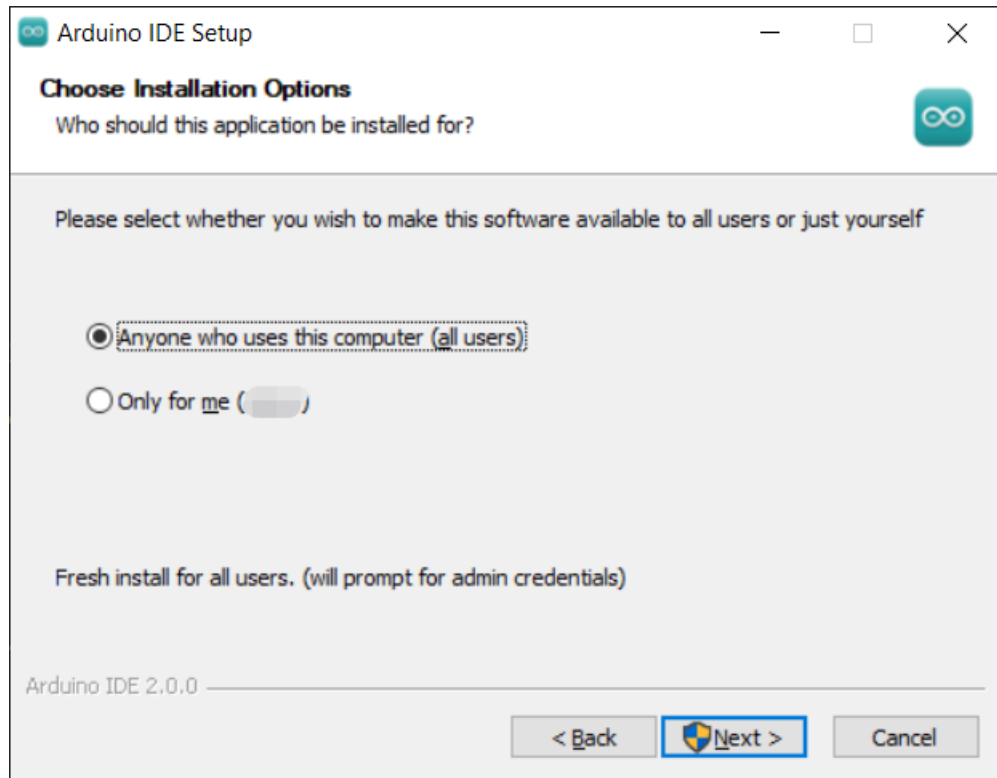
- *Windows*
- *macOS*
- *Linux*

Windows

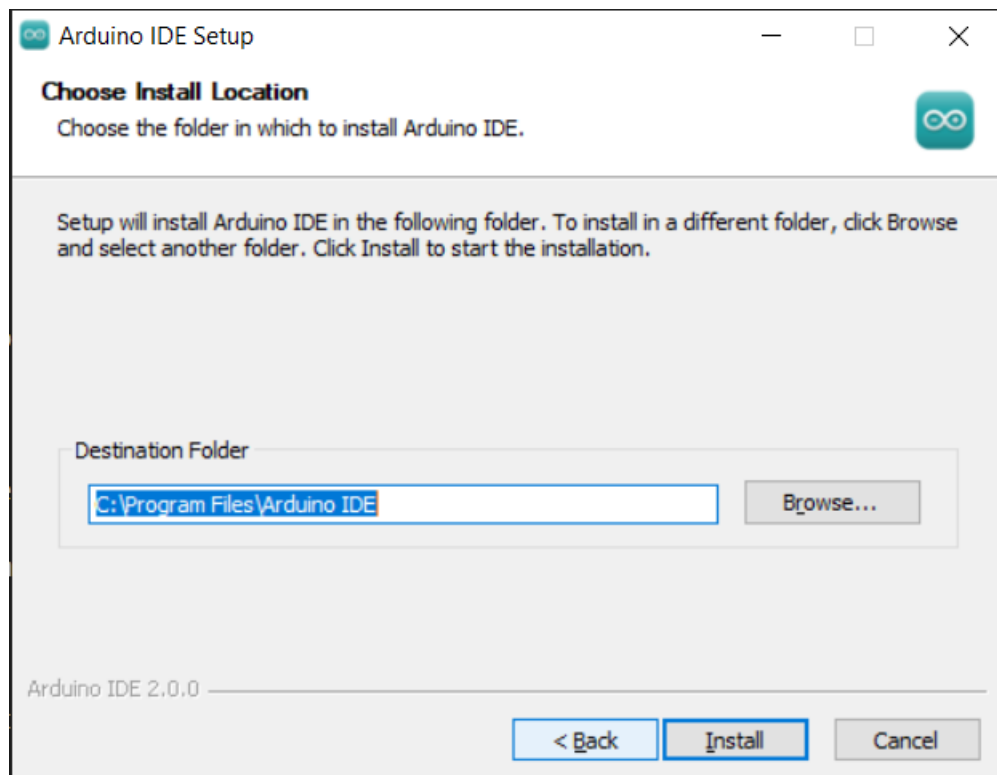
1. Doppelklicken Sie auf die Datei `arduino-ide_xxxx.exe`, um die heruntergeladene Datei auszuführen.
2. Lesen Sie die Lizenzvereinbarung und akzeptieren Sie diese.



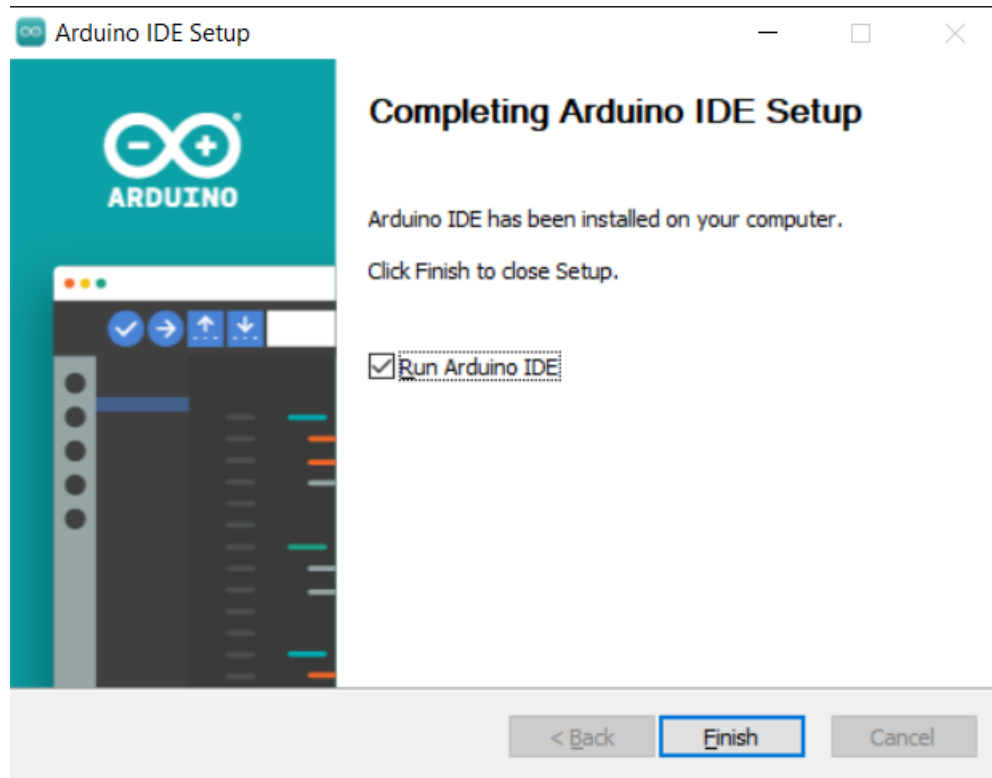
3. Wählen Sie die Installationsoptionen.



4. Wählen Sie den Installationsort. Es wird empfohlen, die Software auf einem anderen Laufwerk als dem Systemlaufwerk zu installieren.

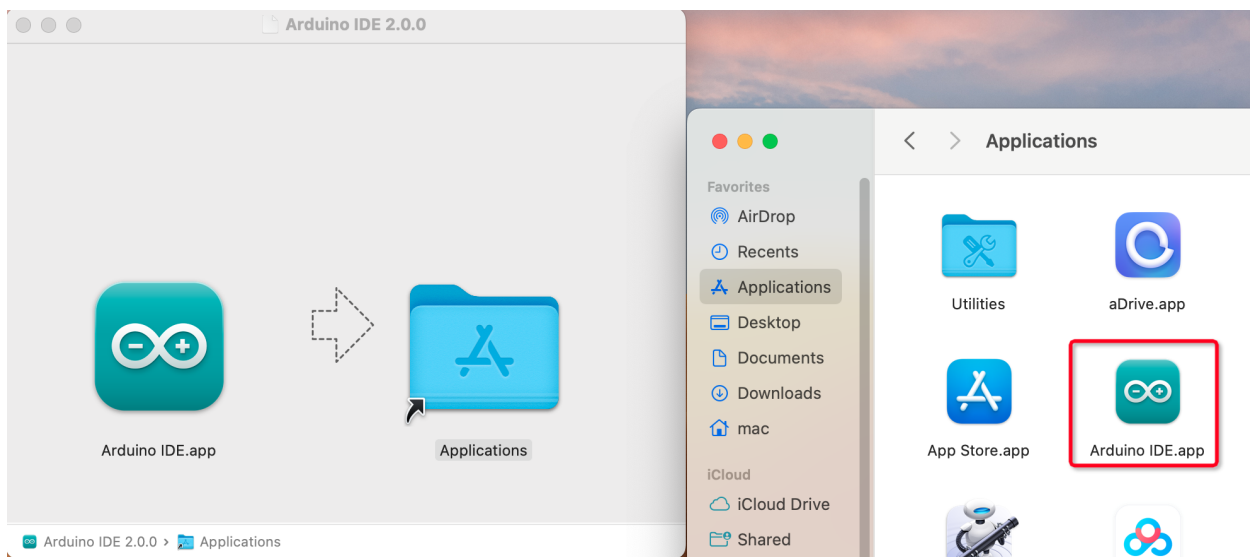


5. Dann beenden.



macOS

Doppelklicken Sie auf die heruntergeladene Datei `arduino_ide_XXXX.dmg` und folgen Sie den Anweisungen, um die **Arduino IDE.app** in den **Anwendungen** Ordner zu kopieren. Nach wenigen Sekunden sehen Sie, dass die Arduino IDE erfolgreich installiert wurde.

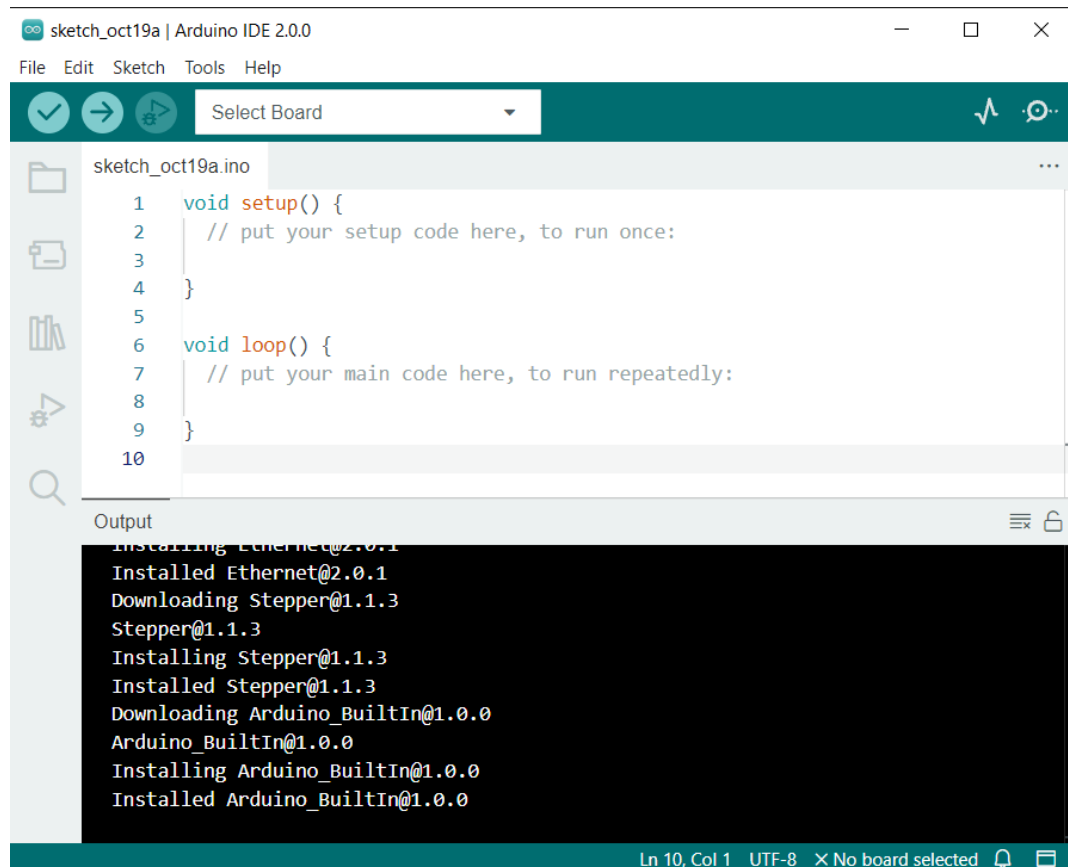


Linux

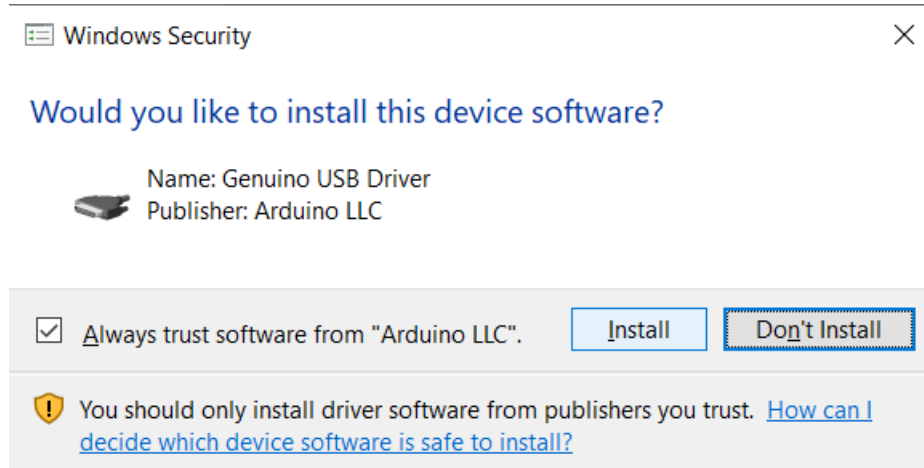
Für das Tutorial zur Installation der Arduino IDE 2.0 auf einem Linux-System besuchen Sie bitte: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing#linux>

Öffnen Sie die IDE

1. Wenn Sie die Arduino IDE 2.0 zum ersten Mal öffnen, installiert sie automatisch die Arduino AVR Boards, eingebaute Bibliotheken und andere erforderliche Dateien.



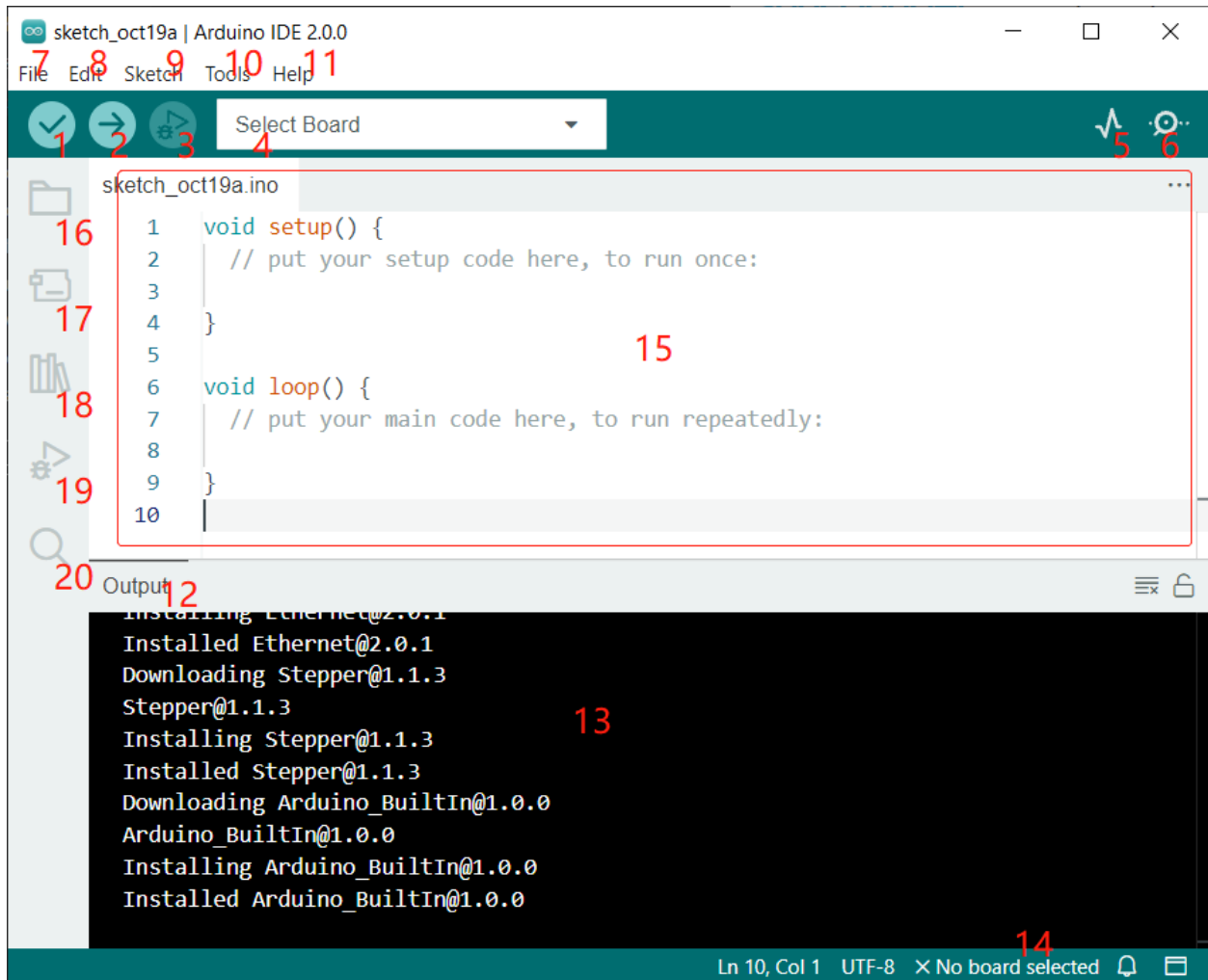
2. Darüber hinaus könnte Ihr Firewall- oder Sicherheitscenter einige Male aufpoppen und Sie fragen, ob Sie einige Gerätetreiber installieren möchten. Bitte installieren Sie alle.



3. Jetzt ist Ihre Arduino IDE einsatzbereit!

Bemerkung: Falls einige Installationen aufgrund von Netzwerkproblemen oder aus anderen Gründen nicht funktioniert haben, können Sie die Arduino IDE erneut öffnen und sie wird den Rest der Installation abschließen. Das Ausgabefenster wird nicht automatisch geöffnet, nachdem alle Installationen abgeschlossen sind, es sei denn, Sie klicken auf Überprüfen oder Hochladen.

1.3.3 Einführung in die Arduino IDE



1. **Überprüfen (Verify):** Kompilieren Sie Ihren Code. Etwaige Syntaxprobleme werden mit Fehlern gemeldet.
2. **Hochladen (Upload):** Laden Sie den Code auf Ihr Board. Wenn Sie die Schaltfläche anklicken, werden die RX- und TX-LEDs auf dem Board schnell blinken und erst aufhören, wenn das Hochladen abgeschlossen ist.
3. **Debuggen (Debug):** Zur zeilenweisen Fehlerprüfung.
4. **Board auswählen (Select Board):** Schnelleinrichtung von Board und Port.
5. **Serial Plotter:** Überprüfen Sie die Änderung des gelesenen Wertes.
6. **Serial Monitor:** Klicken Sie auf die Schaltfläche und ein Fenster erscheint. Es empfängt die Daten, die von Ihrem Steuerungsboard gesendet werden. Es ist sehr nützlich zur Fehlerbehebung.
7. **Datei (File):** Klicken Sie auf das Menü und eine Dropdown-Liste wird angezeigt, einschließlich Dateierstellung, Öffnen, Speichern, Schließen, einigen Parameterkonfigurationen usw.
8. **Bearbeiten (Edit):** Klicken Sie auf das Menü. In der Dropdown-Liste gibt es einige Bearbeitungsvorgänge wie **Ausschneiden (Cut)**, **Kopieren (Copy)**, **Einfügen (Paste)**, **Suchen (Find)** und so weiter, mit ihren entsprechenden Tastenkombinationen.
9. **Skizze (Sketch):** Beinhaltet Vorgänge wie **Überprüfen (Verify)**, **Hochladen (Upload)**, **Dateien hinzufügen**

(Add files) usw. Eine wichtige Funktion ist **Bibliothek einfügen (Include Library)** – hier können Sie Bibliotheken hinzufügen.

10. **Werkzeug (Tool)**: Beinhaltet einige Werkzeuge – am häufigsten verwendet werden Board (das von Ihnen verwendete Board) und Port (der Port, an dem sich Ihr Board befindet). Jedes Mal, wenn Sie den Code hochladen möchten, müssen Sie sie auswählen oder überprüfen.
11. **Hilfe (Help)**: Wenn Sie Anfänger sind, können Sie die Optionen im Menü überprüfen und die benötigte Hilfe erhalten, einschließlich Vorgänge in der IDE, Einführungsinformationen, Fehlerbehebung, Codeerklärung usw.
12. **Ausgabebalken (Output Bar)**: Wechseln Sie hier die Ausgabe-Registerkarte.
13. **Ausgabefenster (Output Window)**: Druckinformationen.
14. **Board und Port**: Hier können Sie das ausgewählte Board und den Port für das Hochladen des Codes anzeigen. Wenn eines davon falsch ist, können Sie es erneut über **Werkzeuge (Tools)** -> **Board / Port** auswählen.
15. Der Bearbeitungsbereich der IDE. Hier können Sie Code schreiben.
16. **Skizzenbuch (Sketchbook)**: Zum Verwalten von Skizzendateien.
17. **Board-Manager (Board Manager)**: Zum Verwalten des Board-Treibers.
18. **Bibliotheksmanager (Library Manager)**: Zum Verwalten Ihrer Bibliotheksdateien.
19. **Debuggen (Debug)**: Hilft beim Debuggen von Code.
20. **Suche (Search)**: Durchsuchen Sie die Codes aus Ihren Skizzen.

1.3.4 Bibliotheken installieren (Arduino IDE 2)

Was ist eine Bibliothek?

Eine Bibliothek, die einige Funktionsdefinitionen und Header-Dateien enthält, besteht normalerweise aus zwei Dateien: .h (Header-Datei, enthält Funktionsdeklarationen, Makrodefinitionen, Konstruktordefinitionen usw.) und .cpp (Ausführungsdatei, mit Funktionsimplementierung, Variablendefinition usw.).

Wenn Sie eine Funktion aus einer bestimmten Bibliothek verwenden möchten, müssen Sie lediglich eine Header-Datei hinzufügen (z.B. `#include <dht.h>`) und dann diese Funktion aufrufen. Dies macht Ihren Code übersichtlicher.

Wenn Sie die Bibliothek nicht verwenden möchten, können Sie diese Funktionsdefinition auch direkt schreiben. Dies führt jedoch dazu, dass der Code lang und schwer lesbar wird.

Einige Bibliotheken sind bereits in der Arduino IDE enthalten, während andere installiert werden müssen. Schauen wir uns jetzt an, wie man eine solche installiert.

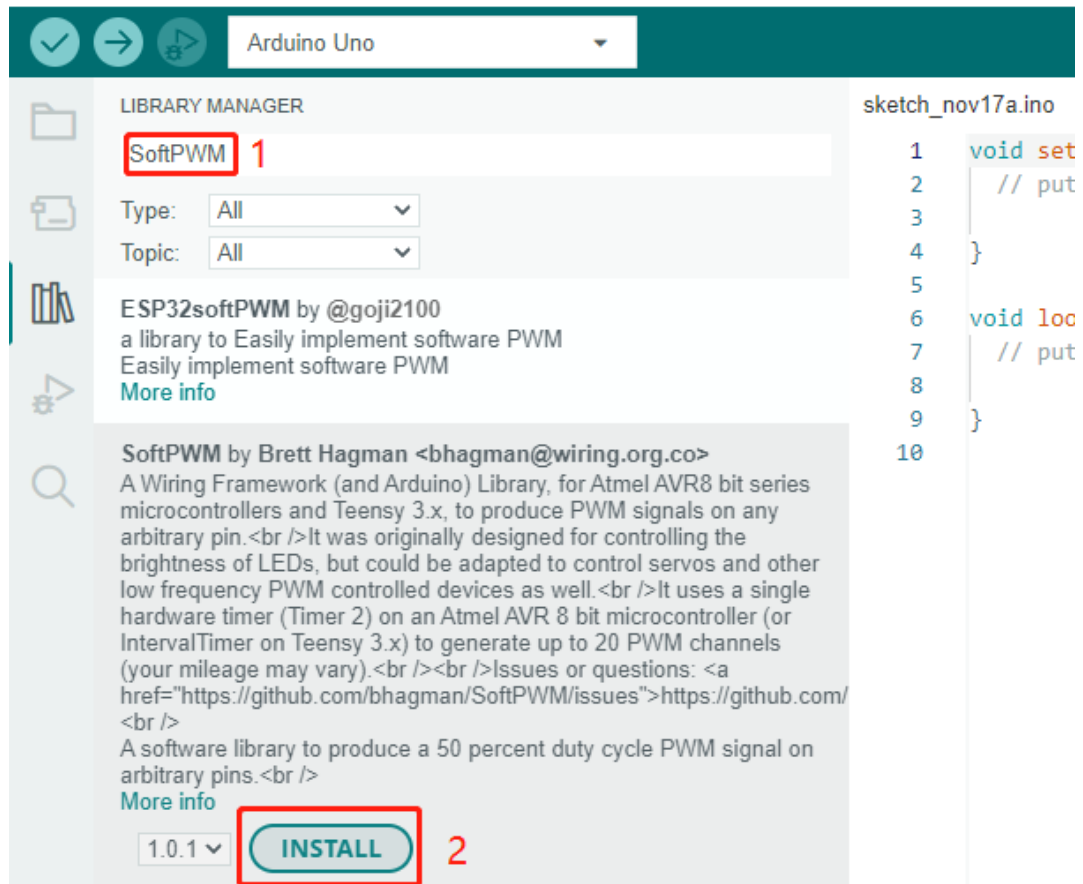
Wie installiert man?

Bemerkung: Ich verwende Arduino IDE 2.0. Wenn Sie Arduino IDE 1.x verwenden, können Sie sich auf beziehen.

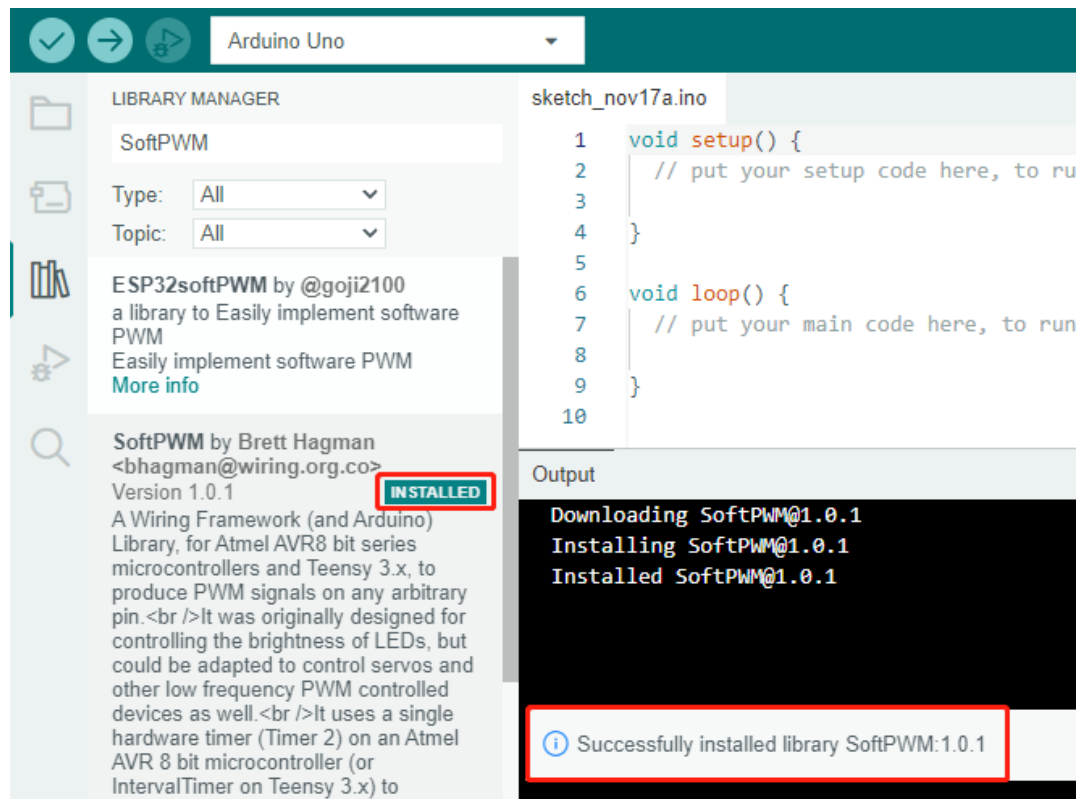
1. Um eine neue Bibliothek in Ihrer Arduino IDE zu installieren, können Sie den **Library Manager** verwenden, den Sie in der linken Spalte öffnen können.



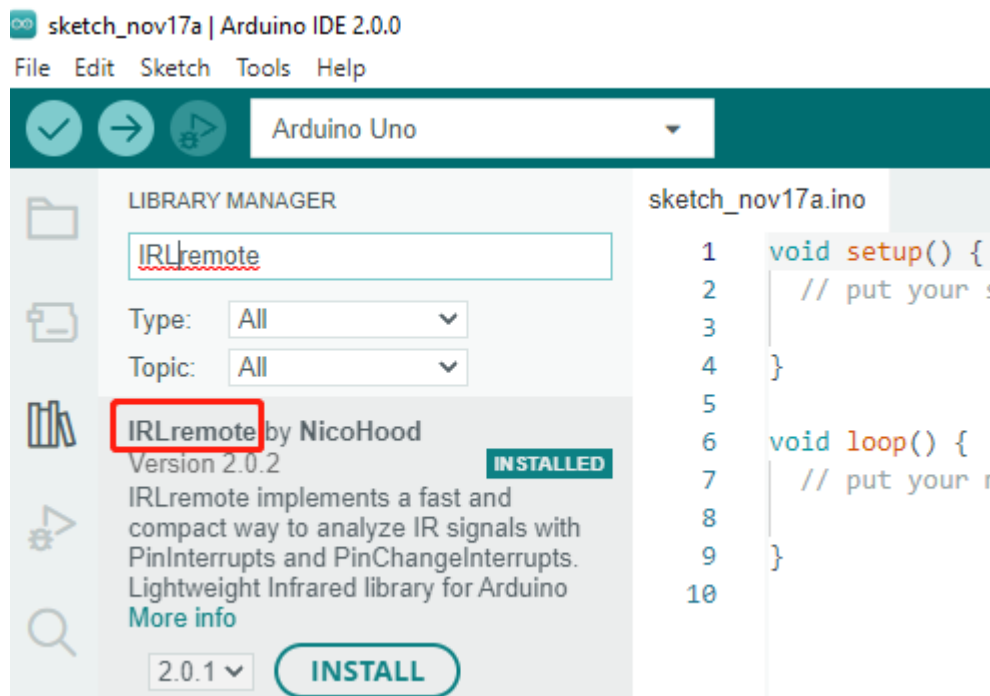
2. Es erscheint nun eine Liste aller verfügbaren Bibliotheken, in der wir auch nach der gewünschten Bibliothek suchen können. Hier werden wir die **SoftPWM** Bibliothek installieren. Klicken Sie auf den **INSTALLIEREN**-Button, um die Bibliothek zu installieren.



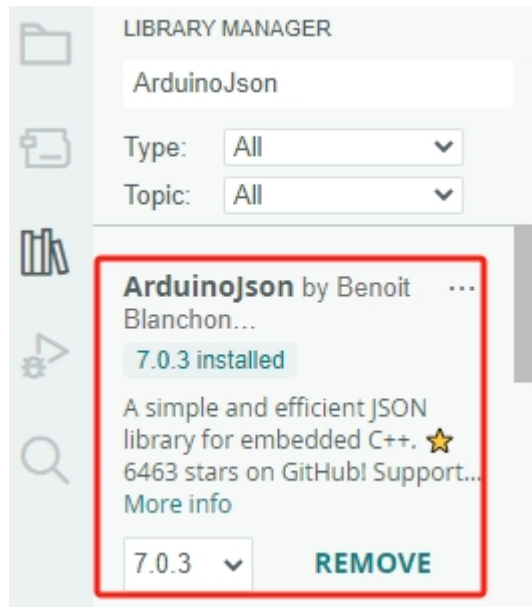
3. Dieser Prozess sollte nicht allzu lange dauern, geben Sie ihm jedoch bis zu einer Minute Zeit für die Installation. Wenn es fertig ist, können wir uns die Bibliothek im Library Manager ansehen, wo **INSTALLIERT** stehen sollte.



4. Installieren Sie die IRLremote Bibliothek auf die gleiche Weise, aber achten Sie darauf, den Namen nicht falsch zu schreiben.



5. Installieren Sie abschließend die ArduinoJson Bibliothek.



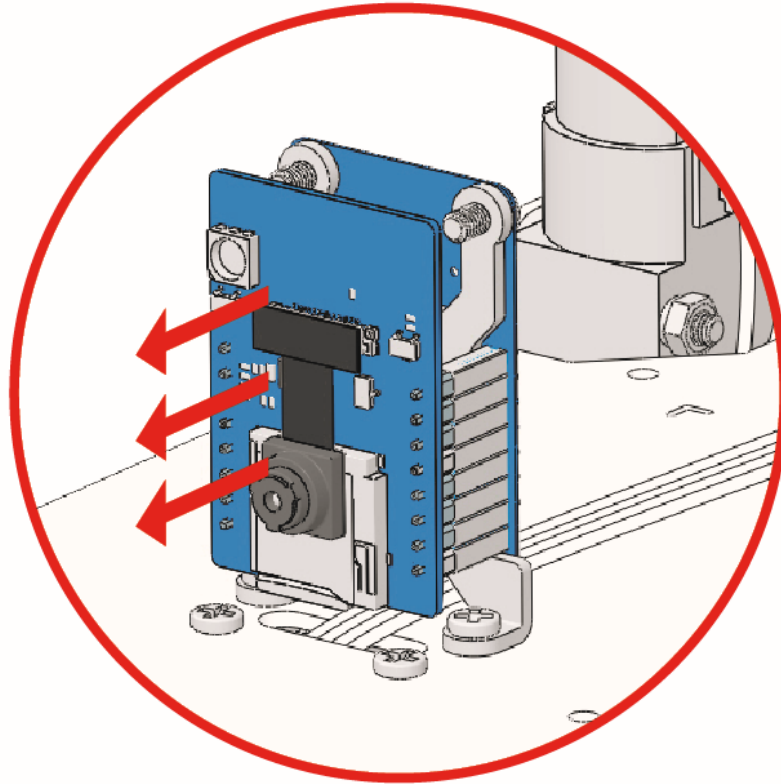
1.3.5 Lustige Projekte

1. Grundbewegungen

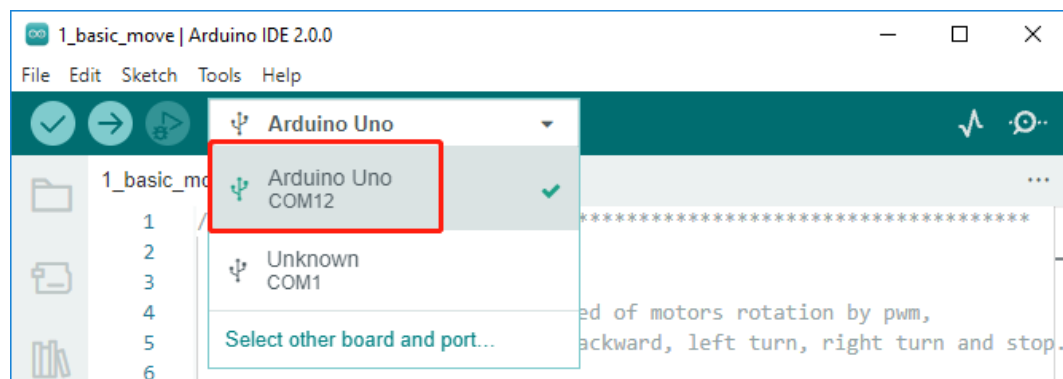
In diesem Projekt lernen Sie, wie Sie das Zeus-Auto in alle Richtungen bewegen können.

Wie geht das?

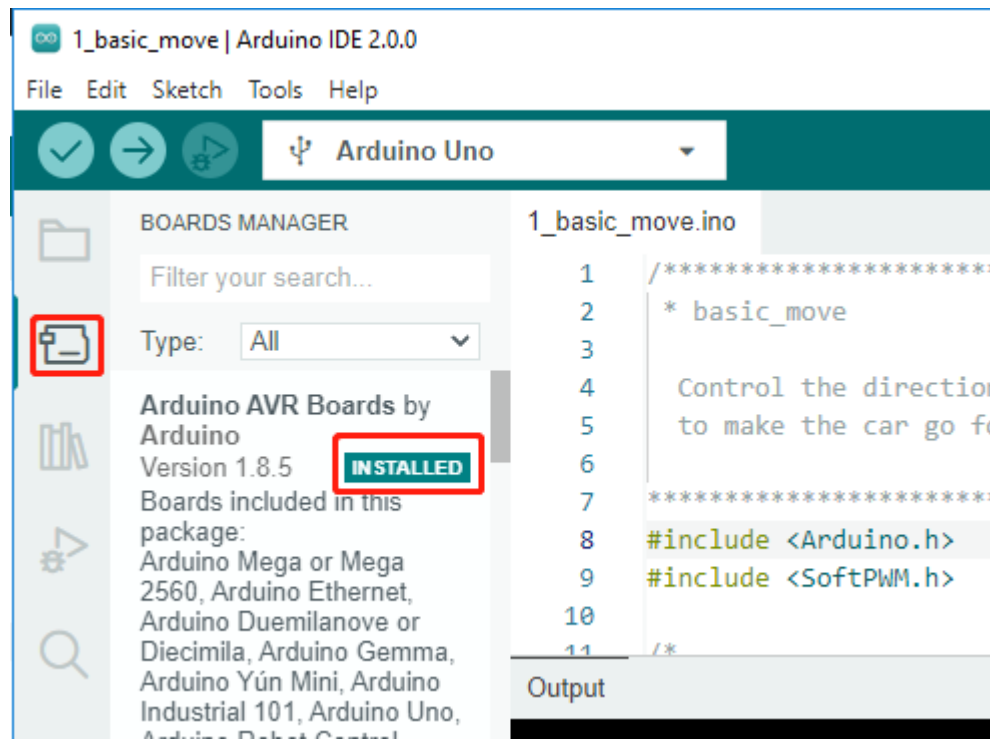
1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



2. Verbinden Sie das Zeus-Auto über das blaue USB-Kabel mit Ihrem Computer. Wenn Sie Ihr Arduino-Board anschließen, erkennt der Computer es automatisch und weist ihm einen COM-Port zu, den Sie im Geräte-Manager einsehen können.
3. Öffnen Sie die Datei `1_basic_move.ino` im Pfad `zeus-car-main\examples\1_basic_move`.
4. Hier können Sie schnell das Board und den Port auswählen.



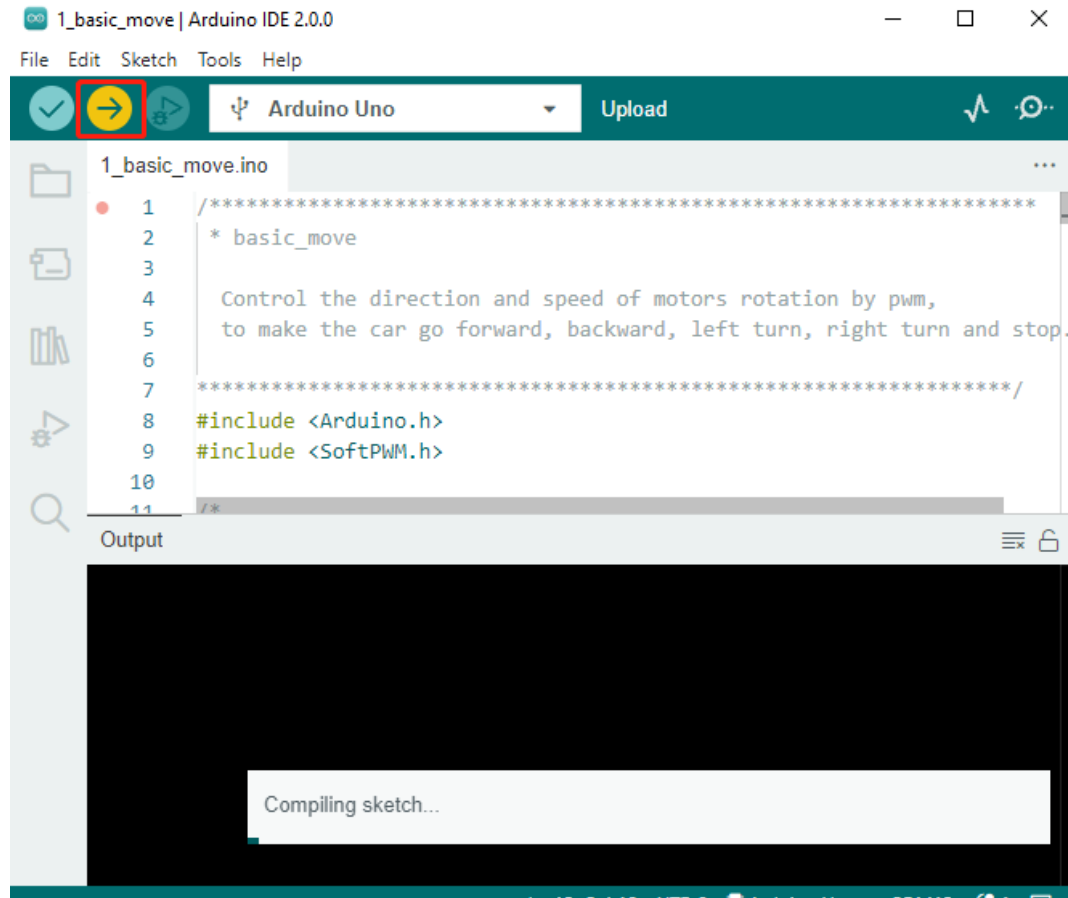
Bemerkung: Falls COMxx nicht erscheint, öffnen Sie den Board Manager links und stellen Sie sicher, dass der Core „Arduino AVR Boards“ installiert ist.



5. Klicken Sie nun auf den **Hochladen**-Button, um den Code auf das Arduino-Board zu übertragen. Wenn dies abgeschlossen ist, wird eine Benachrichtigung im unteren rechten Bereich Ihres IDE-Fensters angezeigt. Natürlich können beim Hochladen manchmal Fehler auftreten, die dann hier ebenfalls aufgelistet werden.

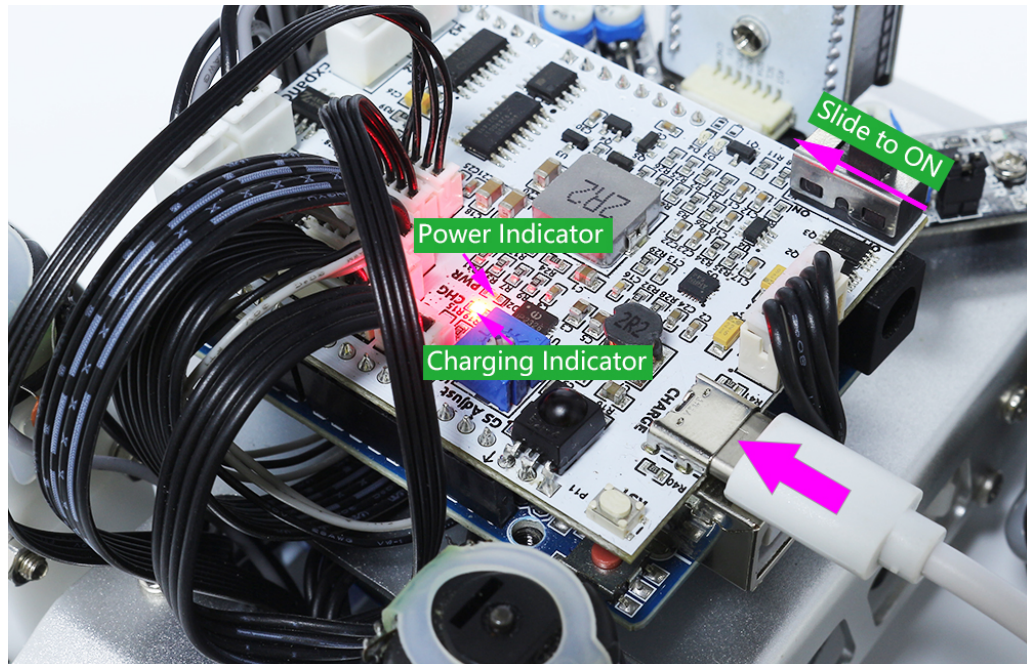
Bemerkung:

- Wenn Sie eine „Compilation error: SoftPWM.h: Datei oder Verzeichnis nicht gefunden“ Meldung erhalten, bedeutet dies, dass Sie die SoftPWM Bibliothek nicht installiert haben.
- Befolgen Sie die Anweisungen unter *Bibliotheken installieren (Arduino IDE 2)*, um die beiden benötigten Bibliotheken SoftPWM und IRLremote zu installieren.

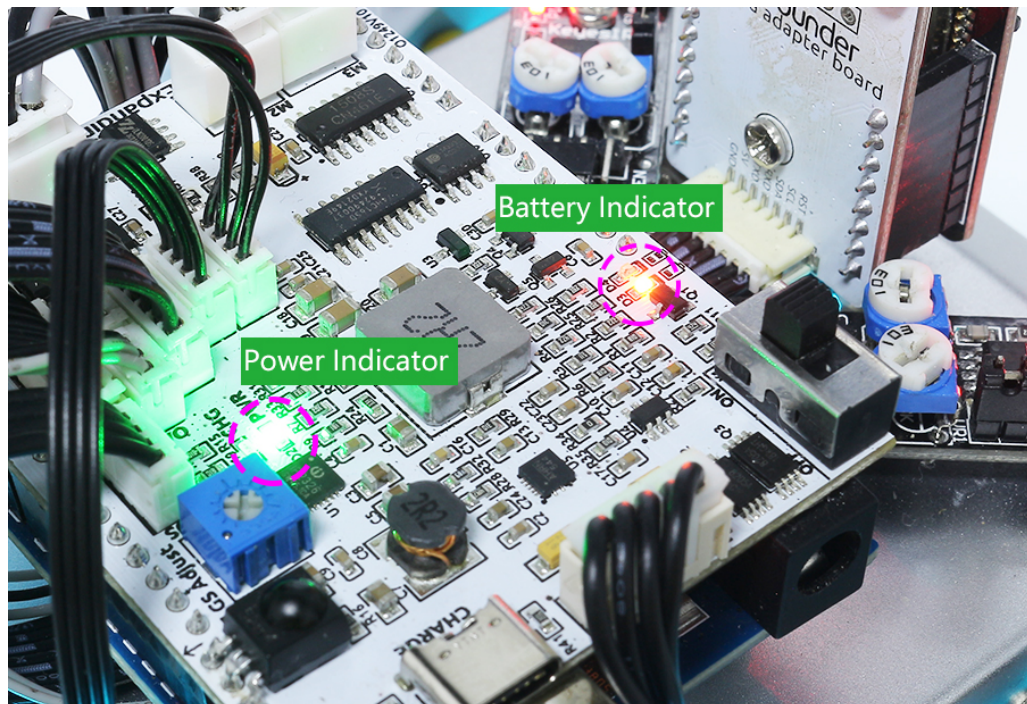


6. Starten wir das Zeus-Auto.

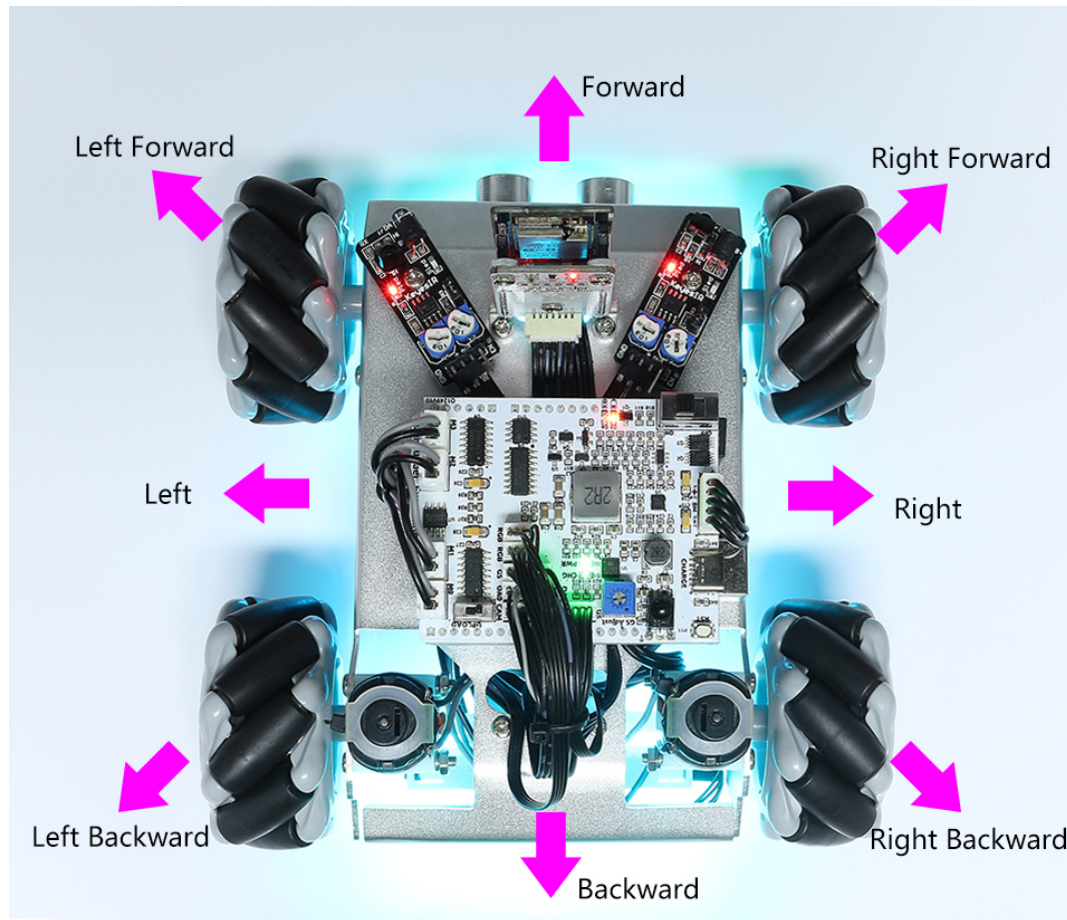
- Bei der ersten Verwendung oder wenn das Batteriekabel getrennt ist, aktiviert das Zeus Car Shield seinen Überentladungsschutz.
- Daher müssen Sie das Type-C-Kabel für etwa 5 Sekunden einstecken.



- Wenn die Stromanzeige leuchtet, bedeutet dies, dass der Schutzstatus aufgehoben wurde. Schauen Sie zu diesem Zeitpunkt auf die Batterieanzeigen; wenn beide Batterieanzeigen aus sind, stecken Sie das Type-C-Kabel weiterhin ein, um die Batterie aufzuladen.



7. Das Zeus-Auto wird nun eine Sekunde lang in jede Richtung fahren.

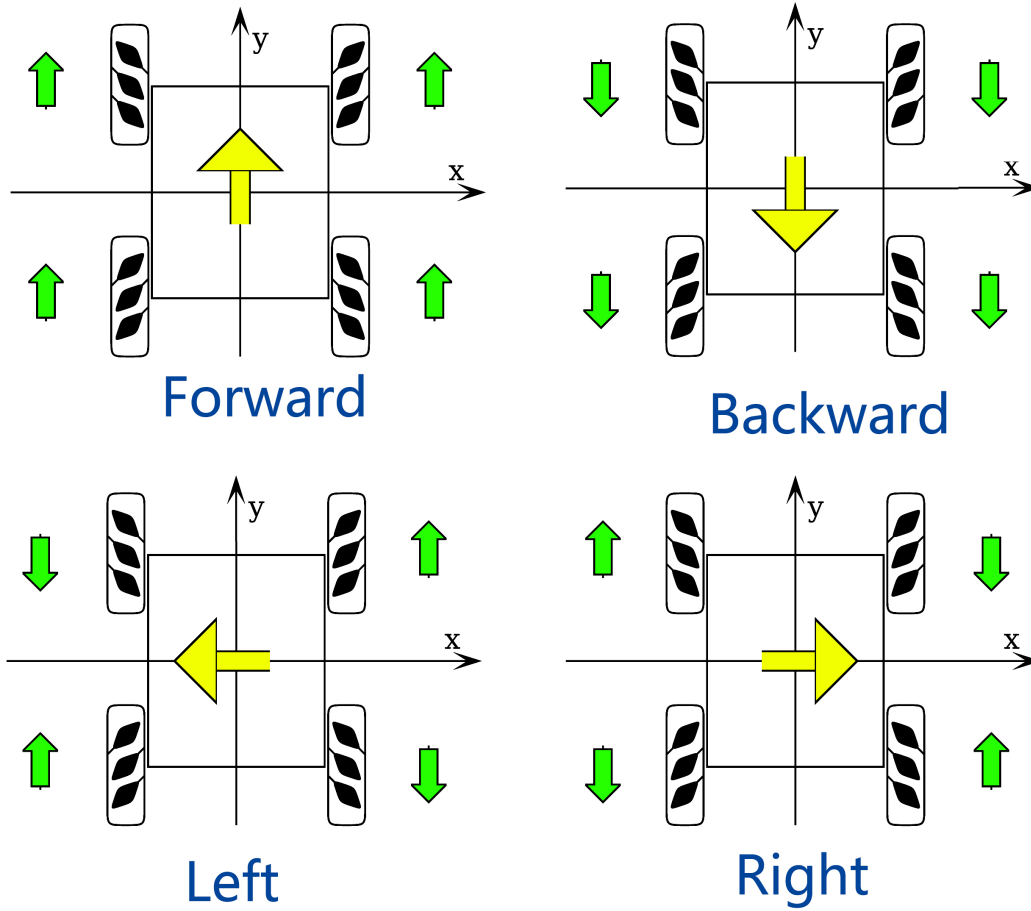


Wie funktioniert es?

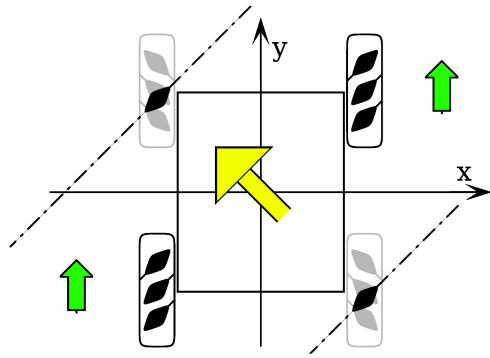
Diese Bewegungen werden durch die Zusammenarbeit der 4 Mecanum-Räder erzielt. Wenn beispielsweise alle vier Räder gleichzeitig vorwärts oder rückwärts fahren, fährt auch das Auto vorwärts oder rückwärts. Zwei Räder vorwärts und zwei Räder rückwärts können das Auto zum Schwenken oder Drehen nach links oder rechts veranlassen.

Schauen wir uns an, wie diese Bewegungen genau erreicht werden.

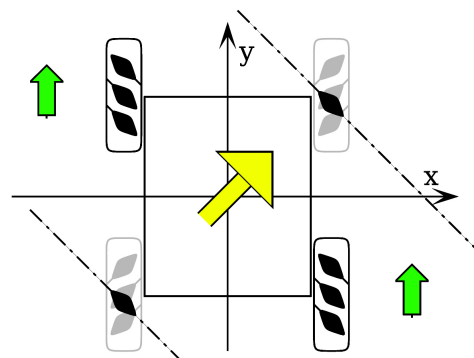
- Vorwärts und rückwärts bewegen, links und rechts schwenken.



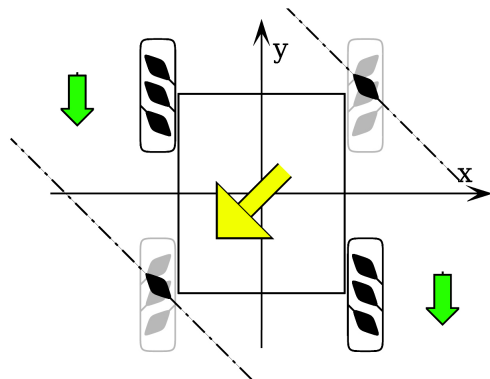
- Diagonal schwenken



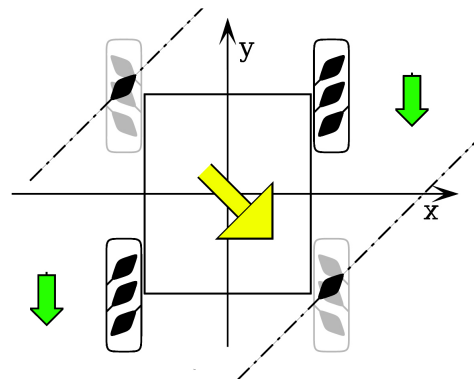
LeftForward



RightForward

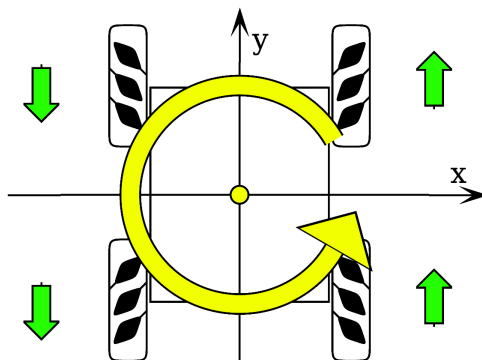


LeftBackward

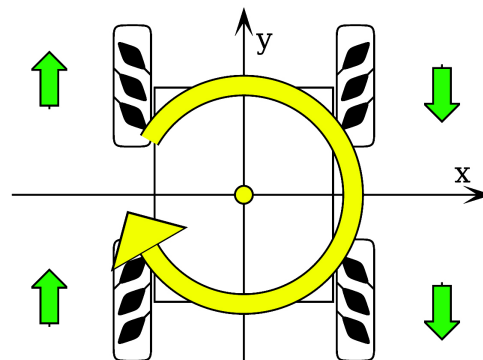


RightBackward

- Links und rechts drehen



Turn Left



TurnRight

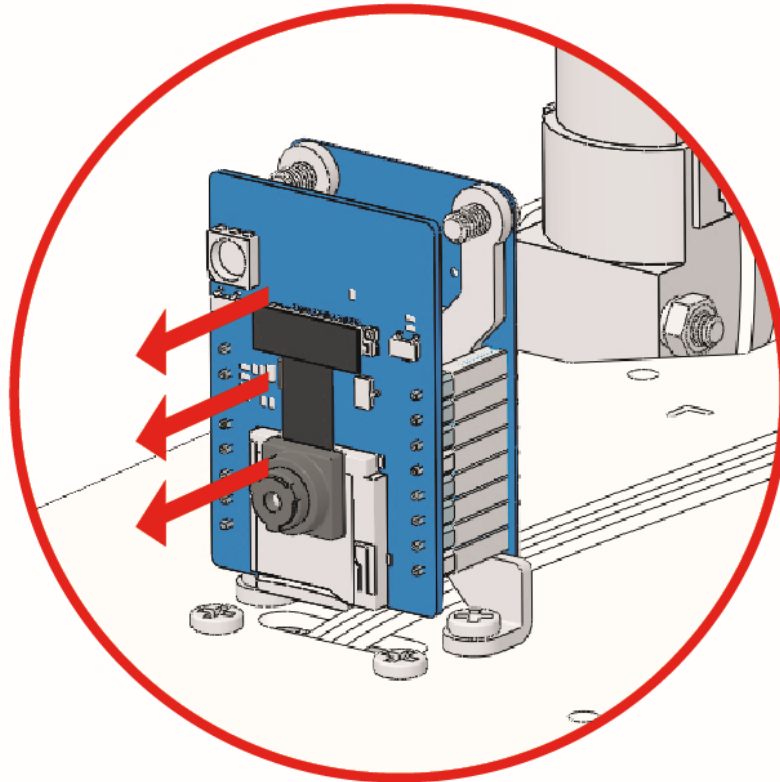
2. Omni-Bewegung

In diesem Projekt lernen Sie, das Zeus-Auto auf 0, 45, 90, 135, 180, 225, 270 und 315 Grad zu bewegen.

Wenn Sie die Reibung am Boden und die strukturellen Toleranzen ignorieren, sollte sein Pfad ein Achteck sein und schließlich zum Ursprung zurückkehren.

Wie geht das?

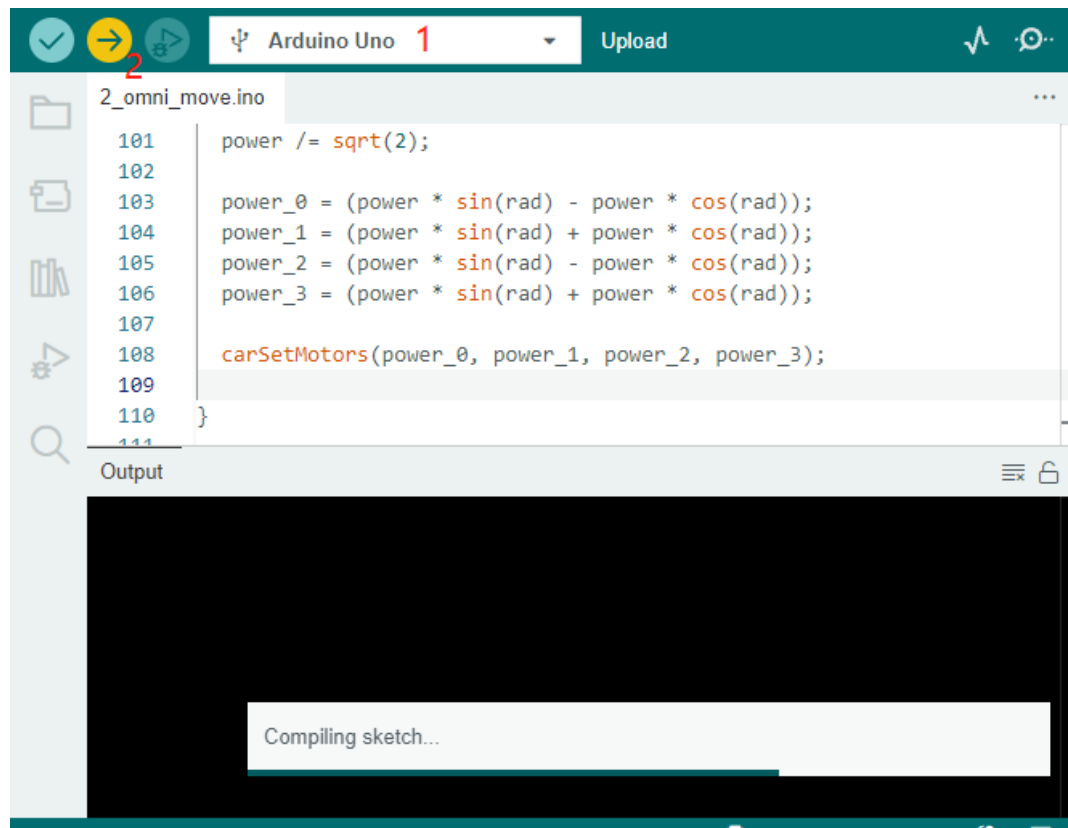
1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



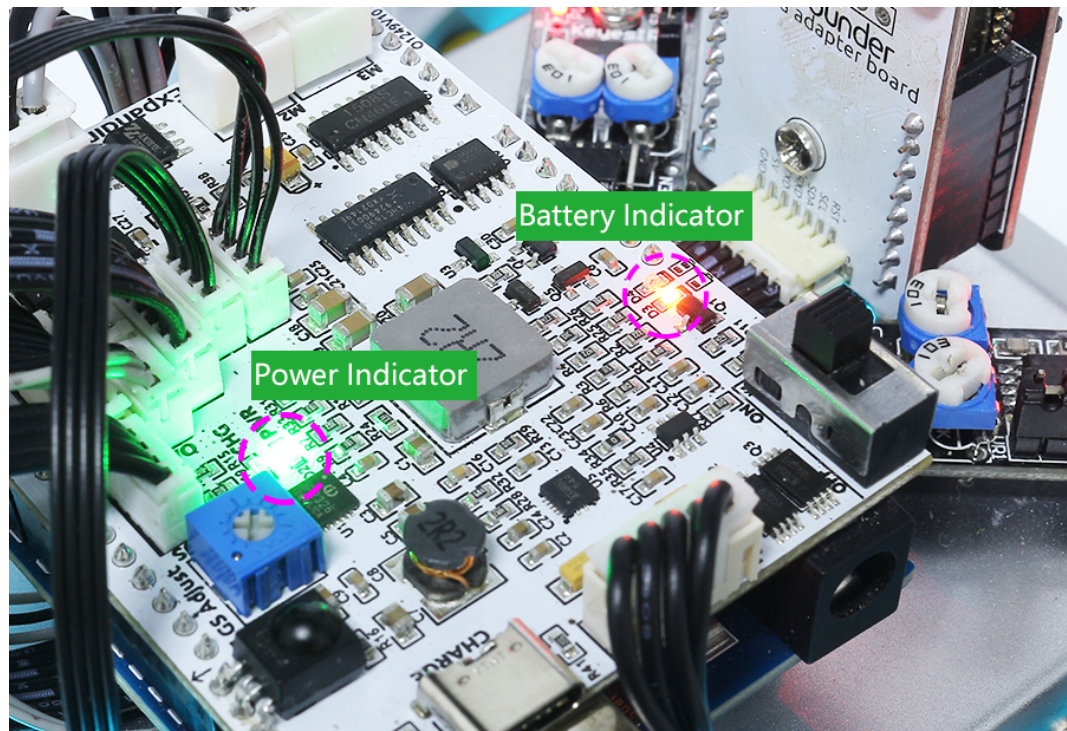
2. Öffnen Sie die Datei `2_omni_move.ino` im Pfad `zeus-car-main\examples\2_omni_move`.
3. Wählen Sie das korrekte Board und den Port aus und klicken Sie dann auf den **Hochladen**-Button.

Bemerkung:

- Wenn Sie eine „Compilation error: SoftPWM.h: Datei oder Verzeichnis nicht gefunden“ Meldung erhalten, bedeutet dies, dass Sie die SoftPWM Bibliothek nicht installiert haben.
 - Befolgen Sie die Anweisungen unter [Bibliotheken installieren \(Arduino IDE 2\)](#), um die beiden benötigten Bibliotheken SoftPWM und IRLremote zu installieren.
-



4. Schalten Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.



5. Zu diesem Zeitpunkt wird das Zeus-Auto auf 0, 45, 90, 135, 180, 225, 270 und 315 Grad fahren.

Wie funktioniert es?

Die Bewegung des Zeus-Autos wird hauptsächlich durch diese Funktion realisiert.

```
void carMove(int16_t angle, int8_t power)
```

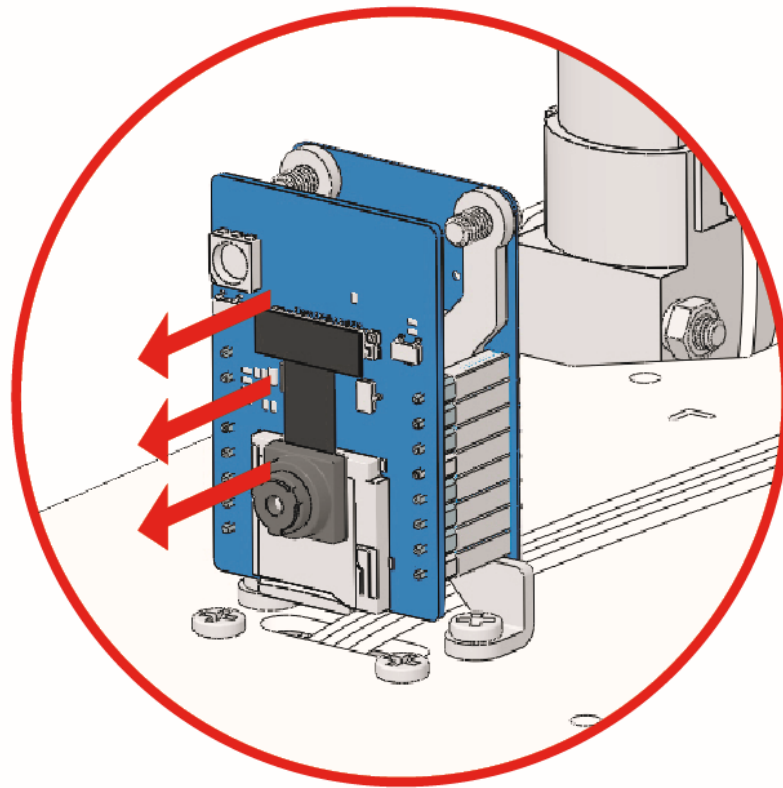
- **angle:** Die Richtung, in die Sie das Auto bewegen möchten. Verwenden Sie die Vorderseite des Autos als den 0-Grad-Winkel und erhöhen Sie den Winkel im Gegenuhrzeigersinn.
- **power:** Die Bewegungskraft, der Bereich liegt zwischen -100% und 100%. Wenn power positiv ist, fährt das Auto vorwärts, andernfalls rückwärts.

3. Bewegen und Rotieren

In diesem Projekt kann das Zeus-Auto die Rotationsparameter erhöhen, sodass es sich an Ort und Stelle drehen oder in einem Bogen fahren kann.

Wie geht das?

1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



2. Öffnen Sie die Datei `3_rotate_and_move.ino` im Pfad `zeus-car-main\examples\3_rotate_and_move`.
3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten. Das Zeus-Auto wird zuerst nach links und dann nach rechts drehen, um in seine Ausgangsposition zurückzukehren. Anschließend wird das Auto langsam in einem Bogen nach außen drehen, wobei der Drehwinkel stetig abnimmt, bis es sich schließlich von selbst dreht.

Wie funktioniert es?

Hier wird ein Rotationskraftparameter `rot` zur Bewegungsfunktion `carMove()` des Zeus-Autos hinzugefügt.

```
void carMove(int16_t angle, int8_t power, int8_t rot)
```

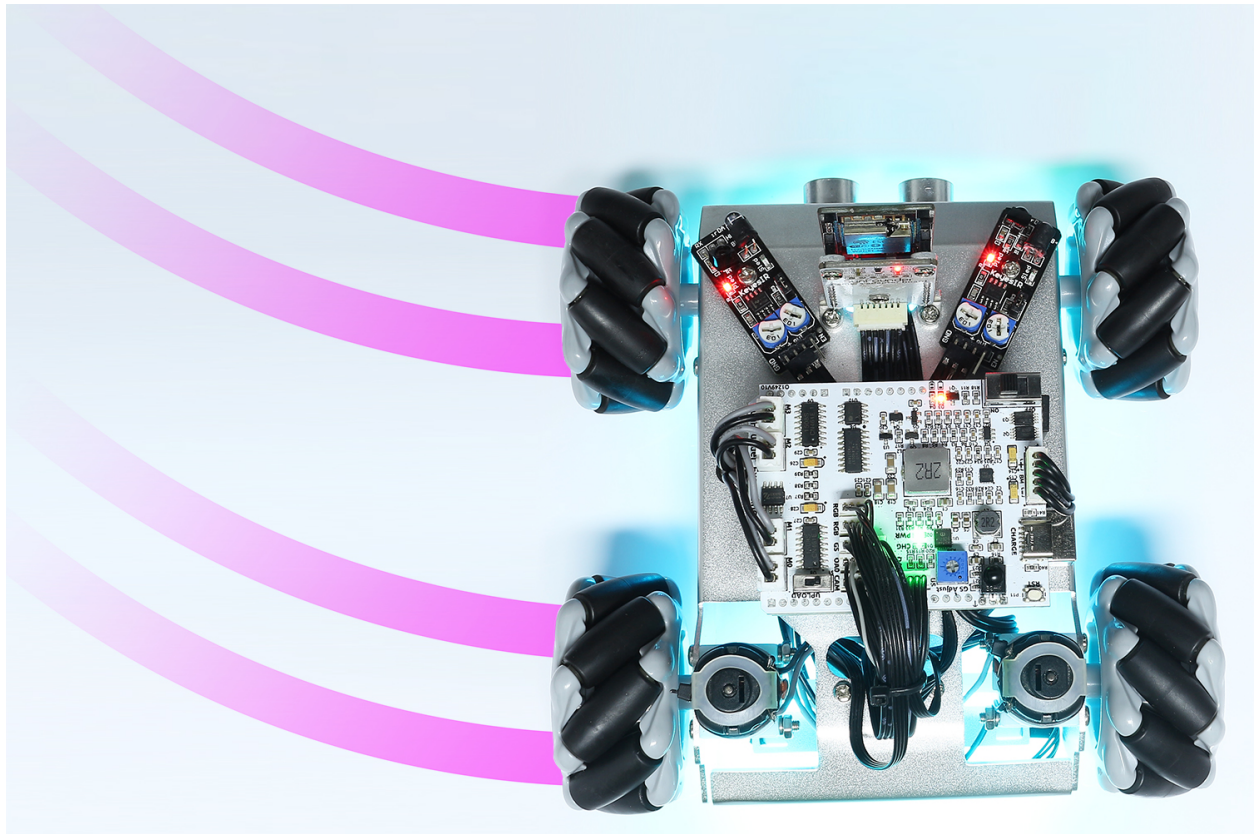
- **angle:** Die gewünschte Bewegungsrichtung des Autos. Verwenden Sie die Vorderseite des Autos als 0 Grad und erhöhen Sie den Winkel in entgegen dem Uhrzeigersinn.
- **power:** Die Bewegungskraft, der Bereich liegt zwischen -100% ~ 100%. Wenn **power** positiv ist, fährt das Auto vorwärts und umgekehrt rückwärts.
- **rot:** Rotationskraft, der Bereich liegt zwischen -100% ~ 100%. Bei positivem **rot** dreht sich das Auto gegen den Uhrzeigersinn und umgekehrt.

Wenn **power** 0 ist und **rot** nicht 0 ist, wird sich das Zeus-Auto an Ort und Stelle drehen. Je höher die Rotationskraft, desto schneller die Drehgeschwindigkeit. Bei positivem **rot** wird das Auto gegen den Uhrzeigersinn rotieren und umgekehrt.

Ist **power** nicht 0, wird das Zeus-Auto in einem Bogen fahren. Der Drehwinkel wird mit zunehmendem **rot** größer, und das Zeus-Auto wird sich drehen, wenn der Wert groß genug ist.

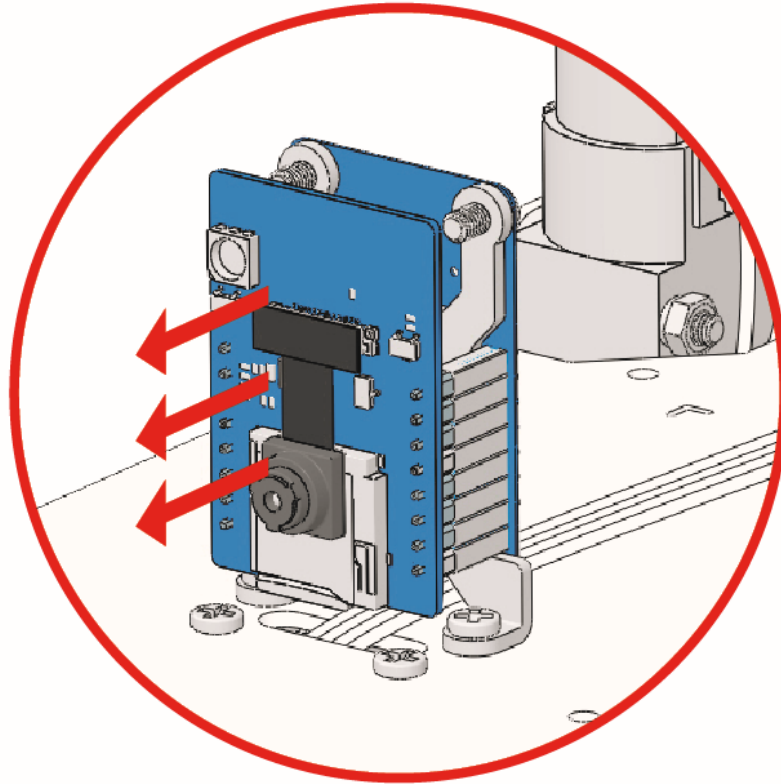
4. Driften

In diesem Projekt lernen Sie, wie Sie das Zeus-Auto driften lassen können.



Wie geht das?

1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



2. Öffnen Sie die Datei `4_drift.ino` im Pfad `zeus-car-main\examples\4_drift`.
3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.
4. Das Zeus-Auto wird jetzt für 0,5 Sekunden vorwärts fahren und dann 90 Grad nach links driften und so weiter. Sie können versuchen, den Wert 45 (rot) durch eine andere Zahl oder eine negative Zahl zu ersetzen, um zu sehen, wie sich der Driftwinkel und die Richtung ändern.

Wie funktioniert es?

Das Driften des Zeus-Autos wird durch Hinzufügen des Drift-Parameters `drift` zur Bewegungsfunktion `carMove()` erreicht.

```
void carMove(int16_t angle, int8_t power, int8_t rot, bool drift)
```

- **angle:** Die gewünschte Bewegungsrichtung des Autos. Nutzen Sie die Vorderseite des Autos als den 0-Grad-Punkt und erhöhen Sie den Winkel im Gegenuhrzeigersinn.
- **power:** Die Bewegungskraft im Bereich von -100% bis 100%. Bei positiver `power` bewegt sich das Auto vorwärts, bei negativer rückwärts.
- **rot:** Drehkraft im Bereich von -100% bis 100%. Bei positiver `rot` dreht sich das Auto im Gegenuhrzeigersinn, und umgekehrt.
- **drift:** Standardmäßig ist dies auf `false` gesetzt. Wenn es auf `true` gesetzt wird, wird der Drift-Modus aktiviert.

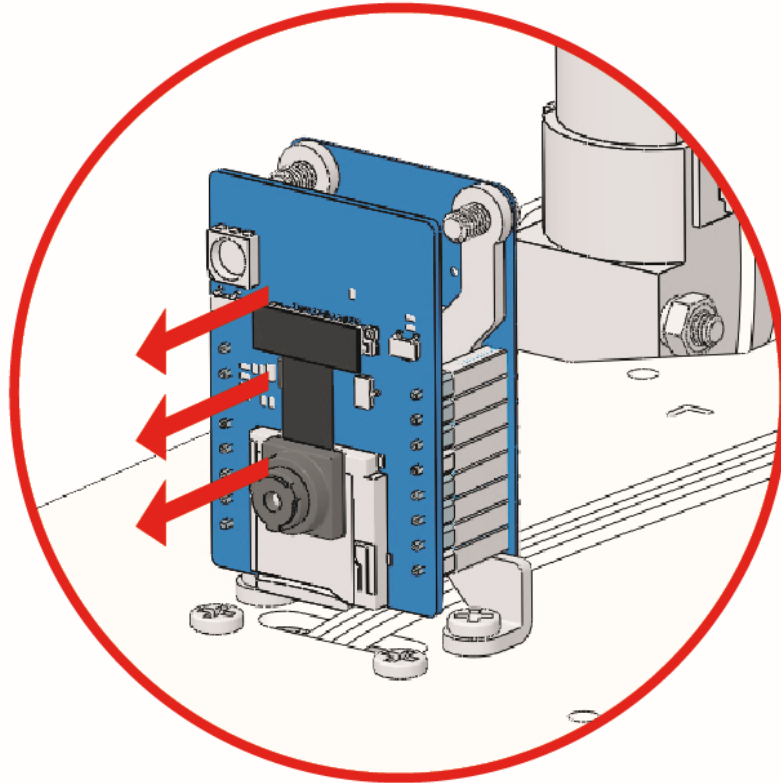
So werden verschiedene `power` und `rot` Werte den Driftwinkel und die Richtung des Zeus-Autos verändern.

5. Fernsteuerung

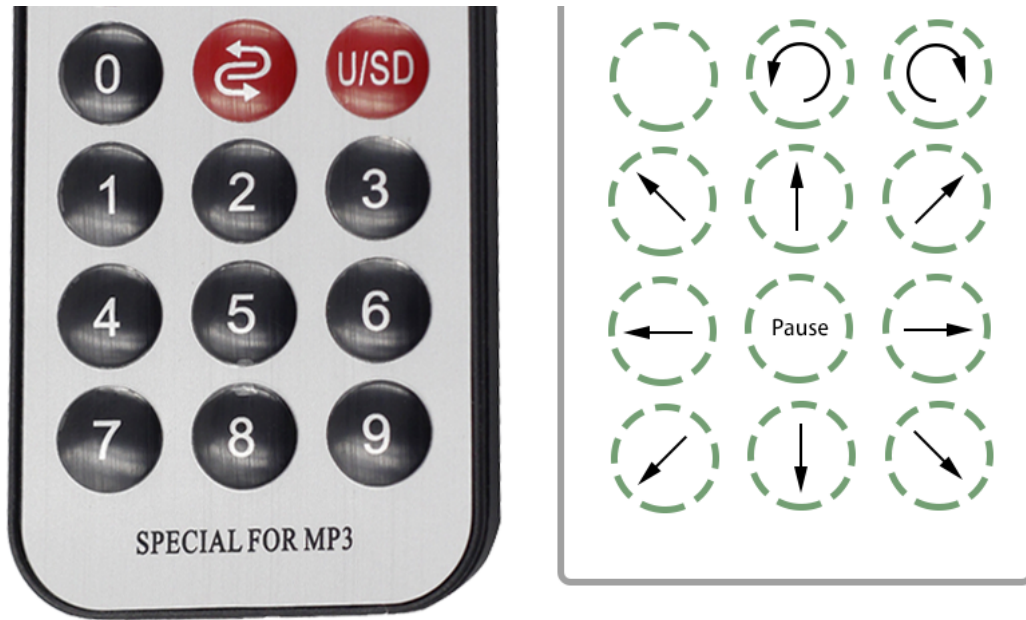
In diesem Projekt lernen Sie, wie Sie das Zeus-Auto mit der Fernbedienung steuern können.


Wie geht das?

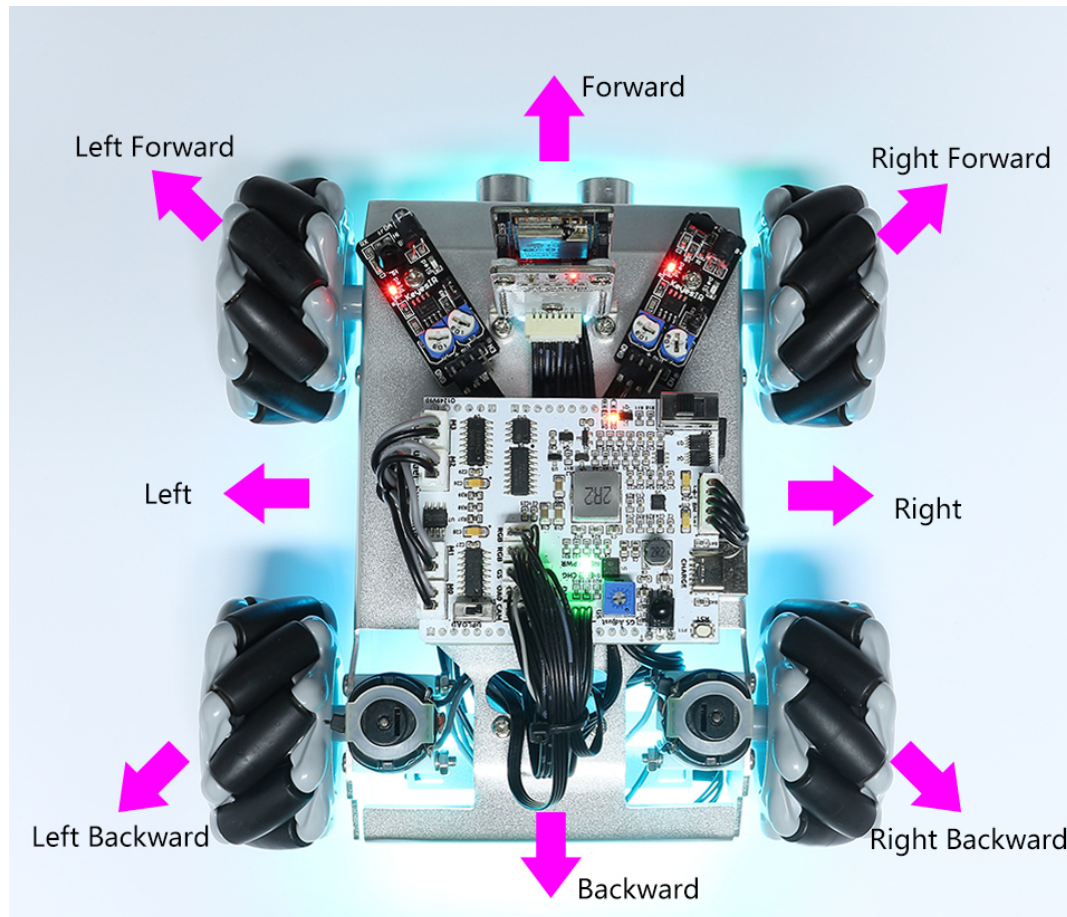
1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.





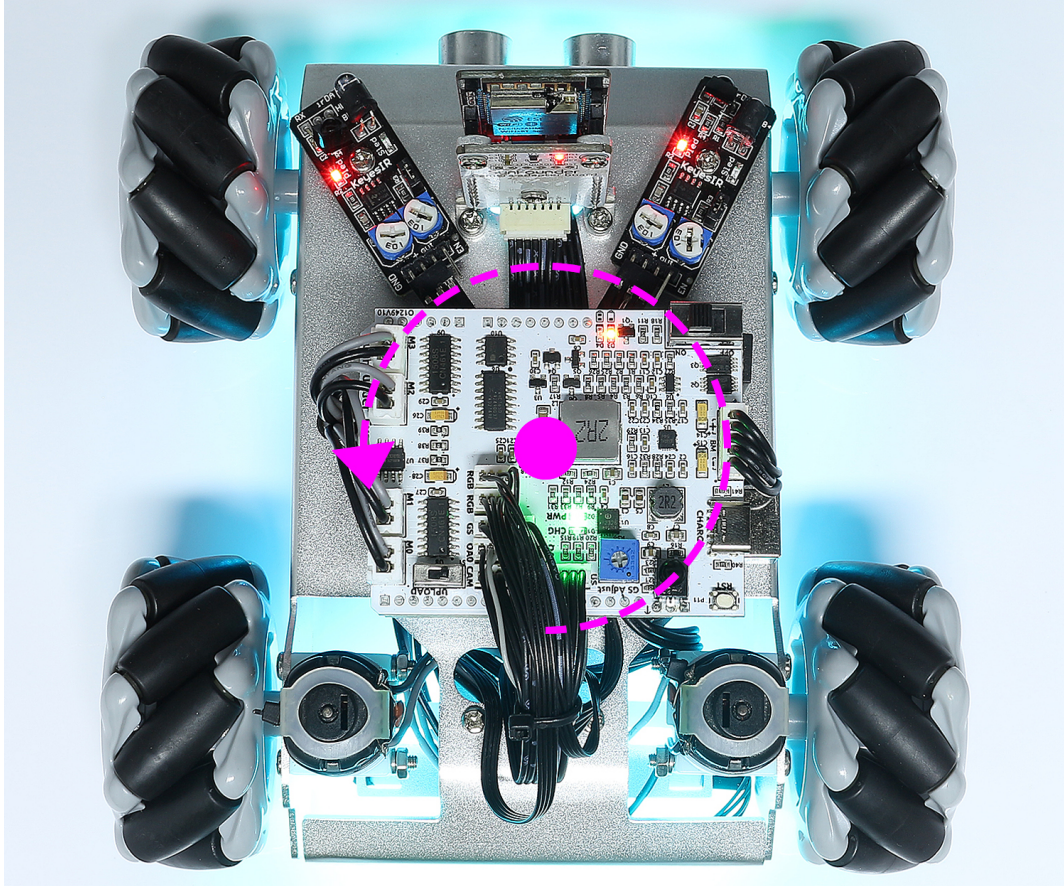
2. Öffnen Sie die Datei `5_remote_control.ino` im Pfad `zeus-car-main\examples\5_remote_control`.
3. Nachdem der Code erfolgreich hochgeladen wurde, schalten Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.
4. Verwenden Sie dann die Zahlen 1~9 auf der Fernbedienung, um das Auto in 8 Richtungen zu steuern.





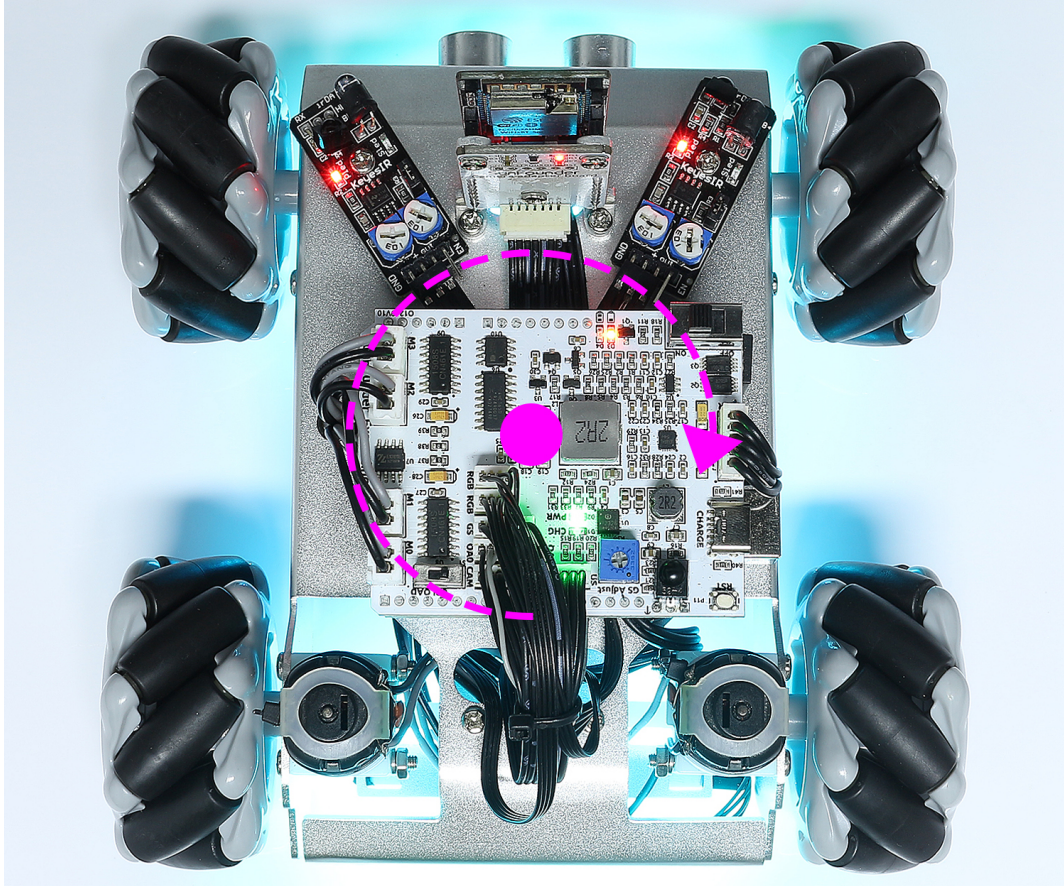
5. Nachdem Sie eine Taste gedrückt haben, wird das Zeus-Auto weiterfahren, bis Sie  oder die Zahlentaste 5 drücken.



6. Wenn Sie einmal  drücken, dreht sich das Auto gegen den Uhrzeigersinn um den Körper als Mittelpunkt und wird anhalten, bis Sie  oder die Zahlentaste 5 drücken.



7. Ebenso wird das Auto im Uhrzeigersinn drehen, wenn Sie einmal  drücken, und es wird anhalten, bis Sie  oder die Zahlentaste 5 drücken.

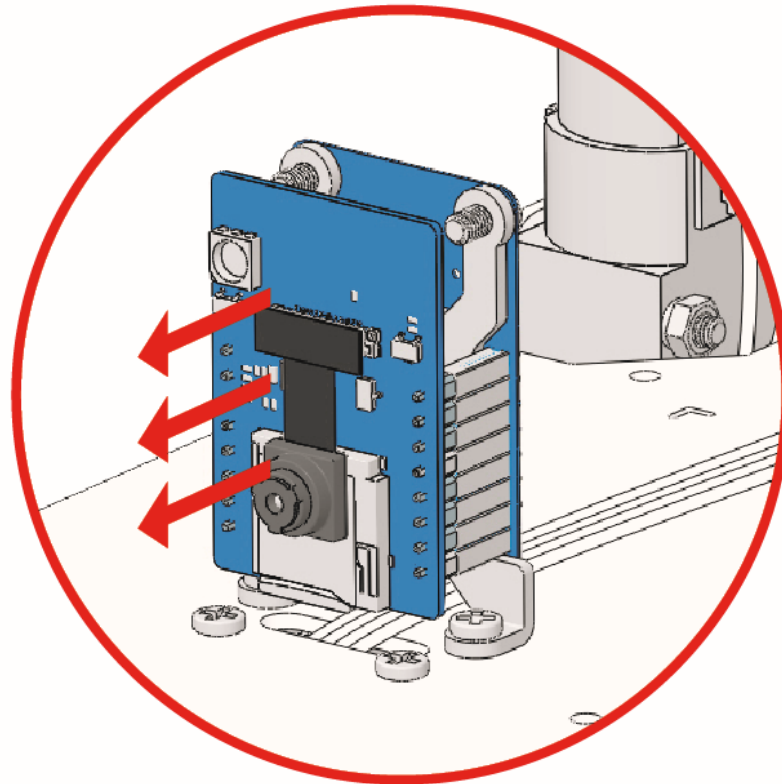



6. Autolicht

In diesem Projekt können Sie die RGB-Lichtleisten am Unterboden des Autos mit der Fernbedienung steuern und ihnen unterschiedliche Farben anzeigen lassen.

Wie geht das?

1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



2. Öffnen Sie die Datei `6_car_light.ino` im Pfad `zeus-car-main\examples\6_car_light`.
3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.
4. Nun können Sie die Zahlen 1 bis 9 auf der Fernbedienung verwenden, um das Auto in verschiedenen Farben leuchten zu lassen. Drücken Sie 0, um das Licht auszuschalten, und drücken Sie , damit das Auto einen Farbzyklus durchführt. Die Tasten 1 bis 9 entsprechen den Farben: Rot, Orange, Gelb, Grün, Türkis, Blau, Violett, Pink, Weiß.

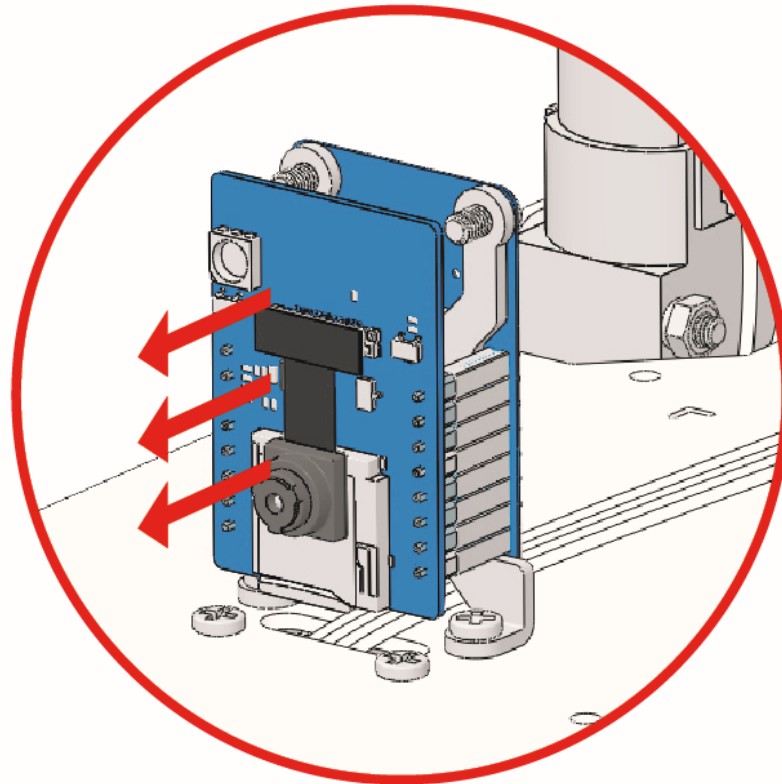
7. Kompass




In diesem Projekt wird das Zeus-Auto zu einem Kompass. Sie müssen ihm eine bestimmte Richtung vorgeben, und egal, wie Sie es drehen, es wird in diese Richtung zeigen.

Bemerkung: Wenn elektrische Leitungen in der Nähe des Zeus-Autos vorhanden sind, kann das elektromagnetische Feld den Betrieb des qmc6310-Moduls stören und somit die vorgegebene Richtung des Autos verändern.

Wie geht das?

1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



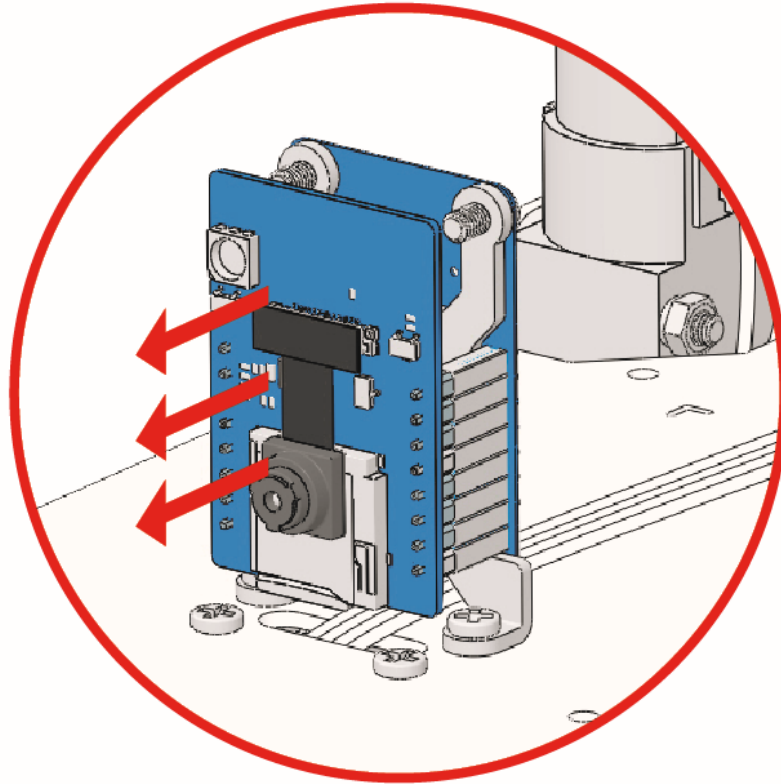
2. Öffnen Sie die Datei `7_compass.ino` im Pfad `zeus-car-main\examples\7_compass`.
3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.
4. Zuerst kalibrieren wir das qmc6310-Modul.
 - Stellen Sie das Auto auf den Boden (achten Sie darauf, nicht in der Nähe von Bereichen mit vielen Kabeln zu sein).
 - Drücken Sie  und das Auto wird sich am Platz drehen.
 - Und stoppen, sobald das Magnetfeld an diesem Ort vollständig erfasst ist.
 - Wenn es nach mehr als 2 Minuten nicht stoppt, können Sie  drücken, um es anzuhalten und dann an einem anderen Ort fortzusetzen.
5. Drehen Sie das Auto in die gewünschte Richtung und drücken Sie . Danach wird das Auto, egal in welche Richtung Sie es drehen, langsam in die voreingestellte Richtung zurückdrehen.
6. Die Magnetfelder am Boden und auf dem Tisch sind nicht identisch und müssen neu kalibriert werden.

8. Feldzentrische Bewegung

Dieses Projekt baut auf dem vorherigen Projekt auf und kombiniert die Kompassfunktion mit der Bewegung des Zeus Autos. Wenn Sie die Bewegung des Zeus Autos mit der Fernbedienung steuern und es durch eine äußere Kraft (z.B. durch einen Hundestoß) abgelenkt wird, wird es automatisch in die ursprüngliche Richtung zurücklenken.

Wie geht das?

1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.

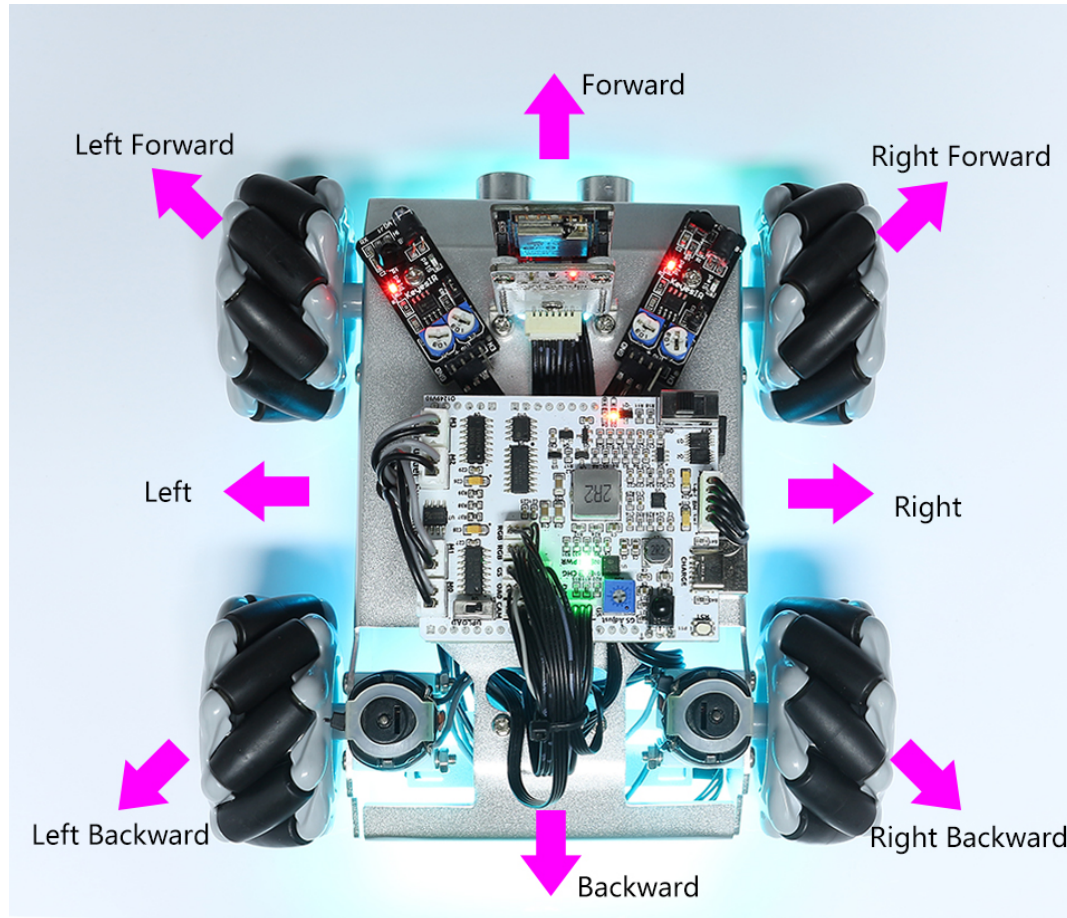




2. Öffnen Sie die Datei `8_move_field_centric.ino` im Pfad `zeus-car-main\examples\8_move_field_centric`.
3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.
4. Verwenden Sie dann die Zahlen 1 ~ 9 auf der Fernbedienung, um das Auto in 8 Richtungen zu steuern. Nach-

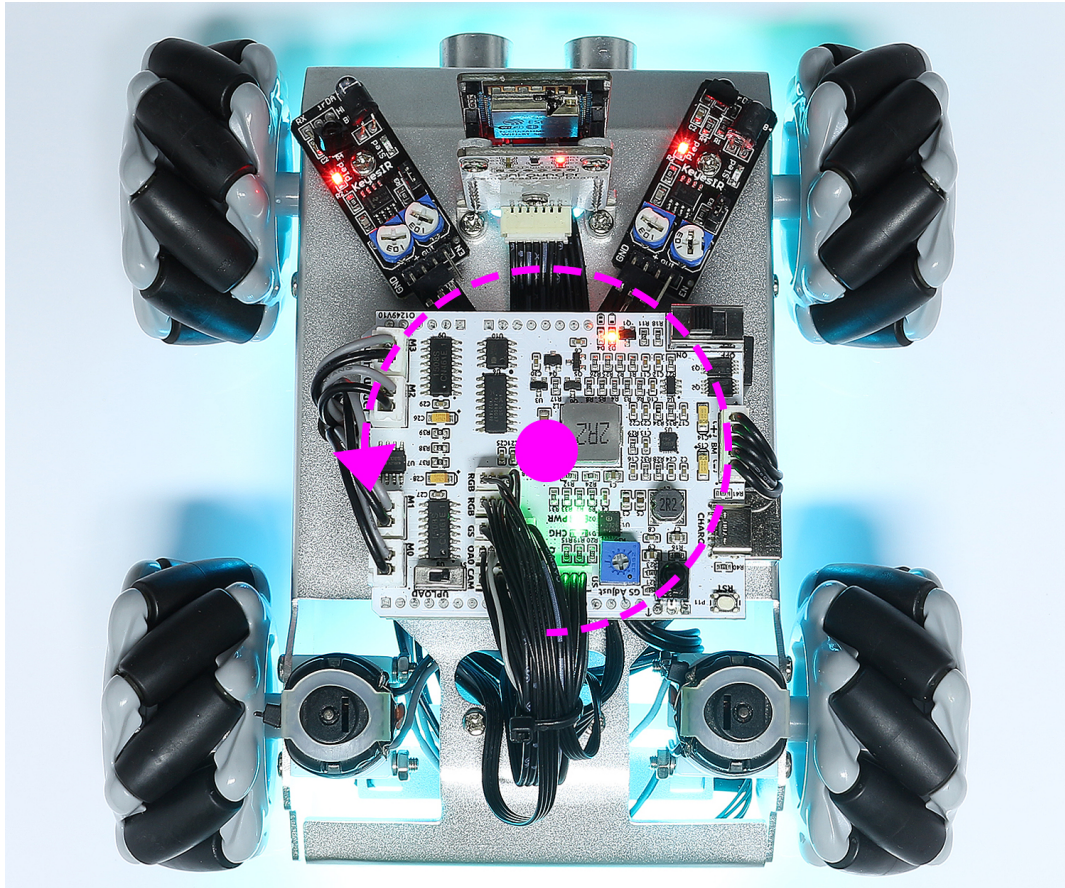
dem Sie eine Taste gedrückt haben, wird das Zeus-Auto weiterfahren, bis Sie




oder die Nummerntaste 5 drücken.



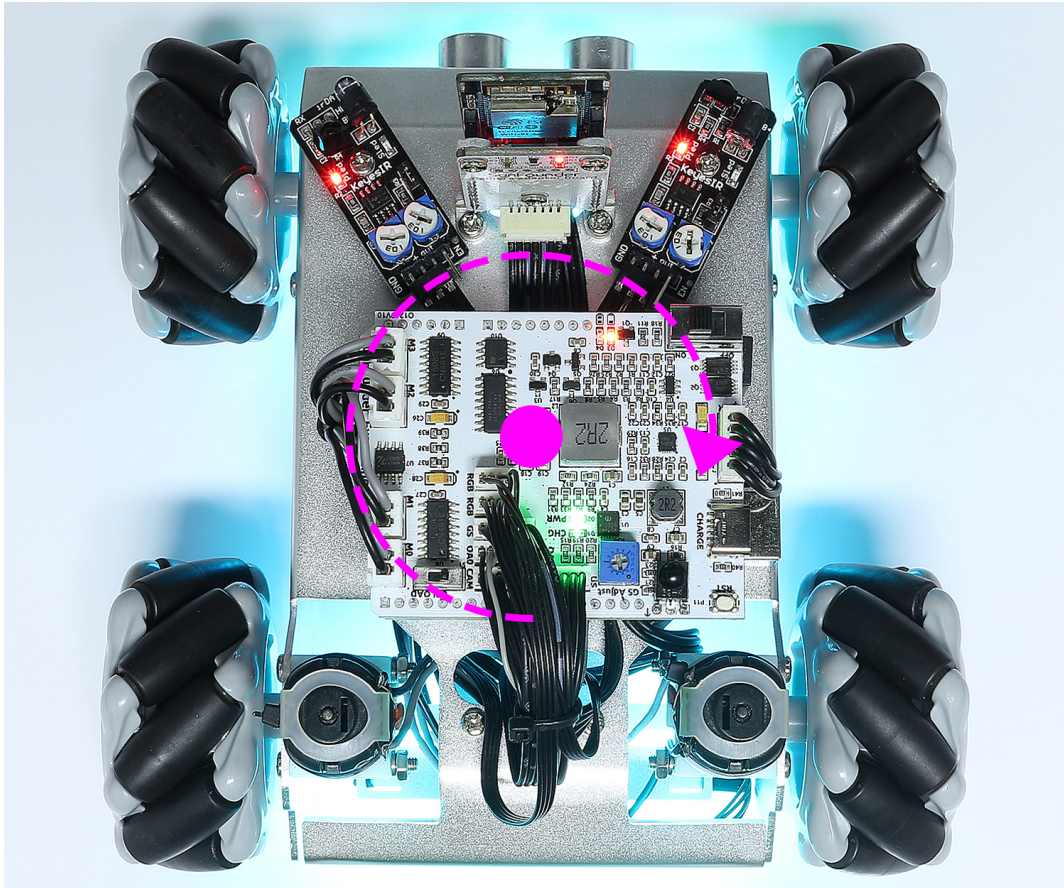
5. Wenn Sie  einmal drücken, wird das Auto gegen den Uhrzeigersinn um den eigenen Körper rotieren und erst stoppen, wenn Sie  oder die Nummerntaste 5 drücken.




6. Ebenso wird das Auto im Uhrzeigersinn rotieren, wenn Sie  einmal drücken und dann stoppen, bis Sie




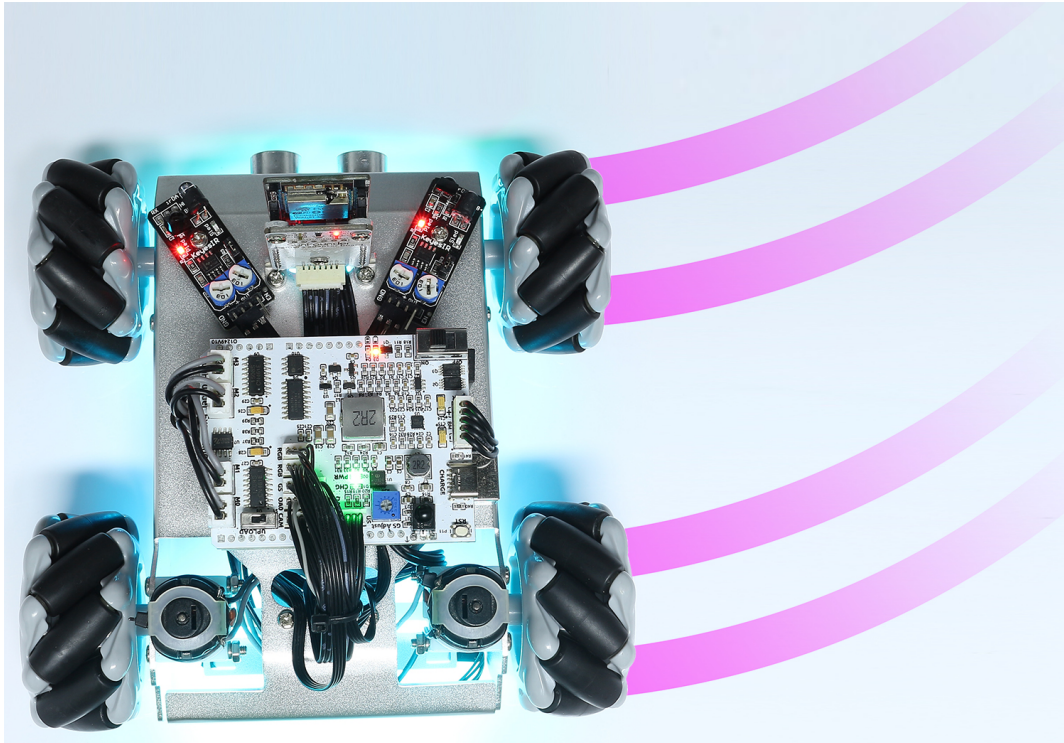
oder die Nummerntaste 5 drücken.




7. Wenn Sie die  Taste drücken, wird das Auto nach links driften.



8. Drücken Sie die  Taste, driftet das Auto nach rechts.



Bemerkung:

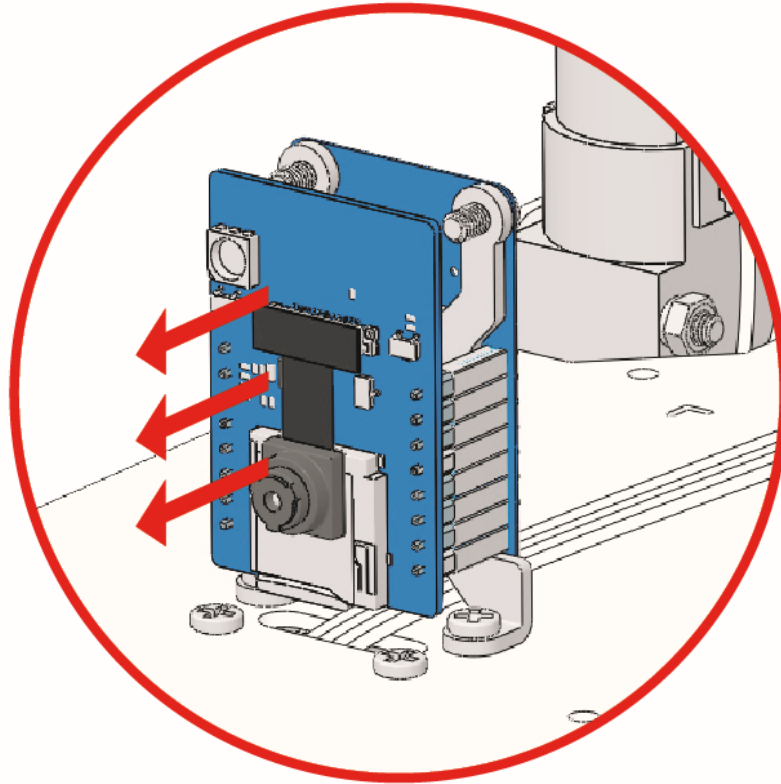
- Falls das Zeus-Auto nicht ordnungsgemäß fährt, ist der Kompass möglicherweise nicht richtig kalibriert. Drücken Sie  zur Neukalibrierung.
 - Wenn Sie das Auto vom Boden auf einen Tisch heben, ändert sich das Magnetfeld und es muss neu kalibriert werden.
-

9. IR Hindernisvermeidung

In diesem Projekt werden Sie lernen, wie Sie die IR-Hindernisvermeidungsmodule an beiden Seiten des Zeus-Autos verwenden können.

Wie geht das?

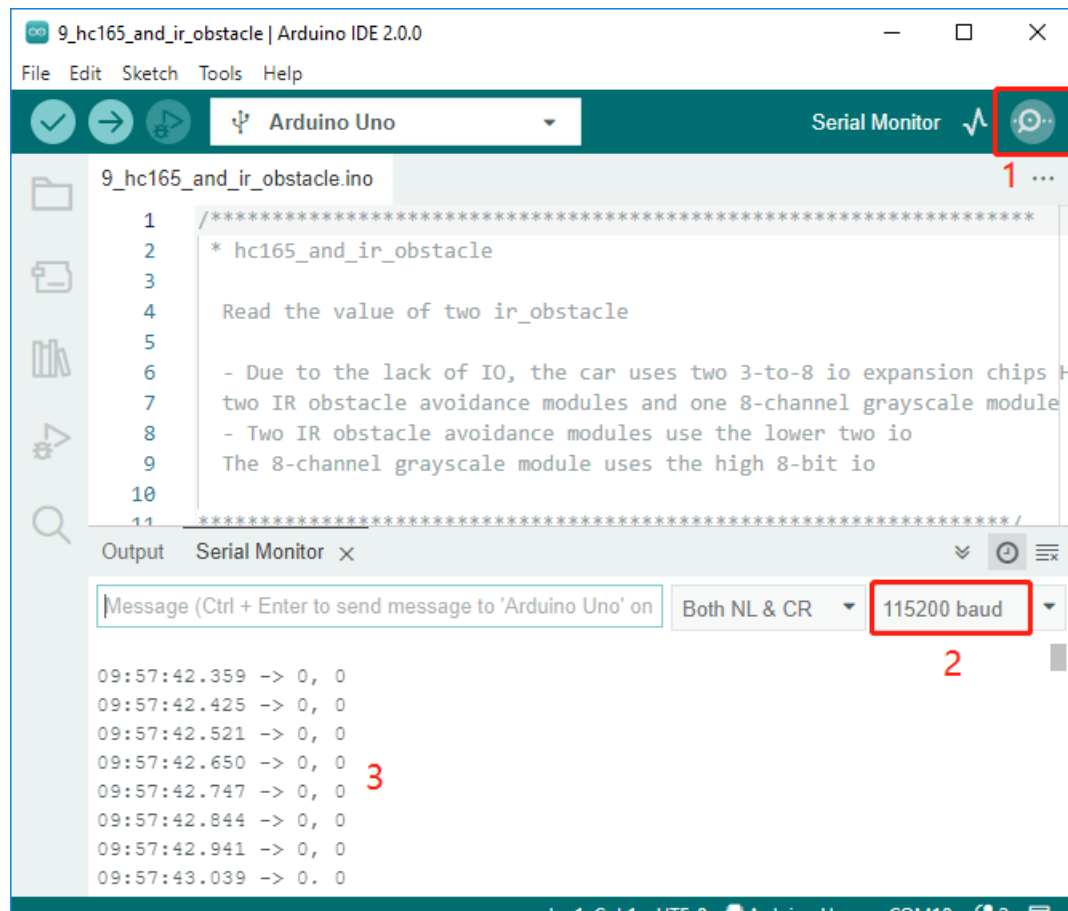
1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



2. Öffnen Sie die Datei `9_hc165_and_ir_obstacle.ino` im Pfad `zeus-car-main\examples\9_hc165_and_ir_obstacle`.
3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.

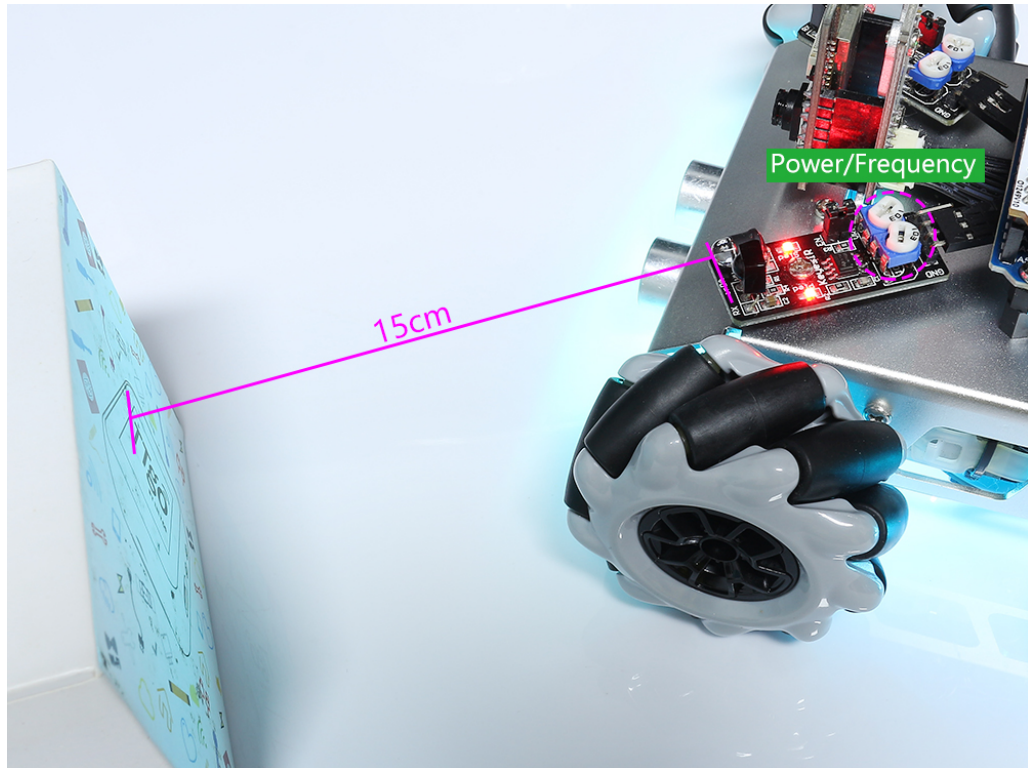
Bemerkung: Ziehen Sie in diesem Schritt nicht das USB-Kabel ab, da Sie die Daten der beiden Hindernisvermeidungsmodule auf Ihrem Computer überprüfen müssen.

4. Öffnen Sie den seriellen Monitor und stellen Sie sicher, dass die aktuelle Baudrate auf 115200 eingestellt ist. Dann können Sie die ausgegebenen Daten betrachten.
 - Wenn beide Hindernisvermeidungsmodule kein Hindernis erkennen, wird der serielle Monitor `0, 0` ausgeben.
 - Halten Sie Ihre Hand vor eines der Hindernisvermeidungsmodule, wird es `1, 0` oder `0, 1` ausgeben.



5. Kalibrieren Sie das IR-Hindernisvermeidungsmodul.

- Beginnen Sie mit der Einstellung des rechten Hindernisvermeidungsmoduls. Während des Transports können Kollisionen dazu führen, dass der Sender und Empfänger im Infrarotmodul kippen. Daher müssen Sie sie manuell begradigen.
- Platzieren Sie ein Hindernis etwa 15cm vom IR-Hindernisvermeidungsmodul entfernt.
- Auf dem Modul befinden sich zwei Potentiometer, eines zur Einstellung der Sendeleistung und eines zur Einstellung der Sendefrequenz. Durch das Verstellen dieser beiden Potentiometer können Sie die Erkennungsdistanz einstellen.
- Danach können Sie ein Potentiometer verstellen, und wenn bei 15cm das Signallicht auf dem Modul leuchtet, ist die Einstellung erfolgreich. Wenn nicht, stellen Sie das andere Potentiometer ein.



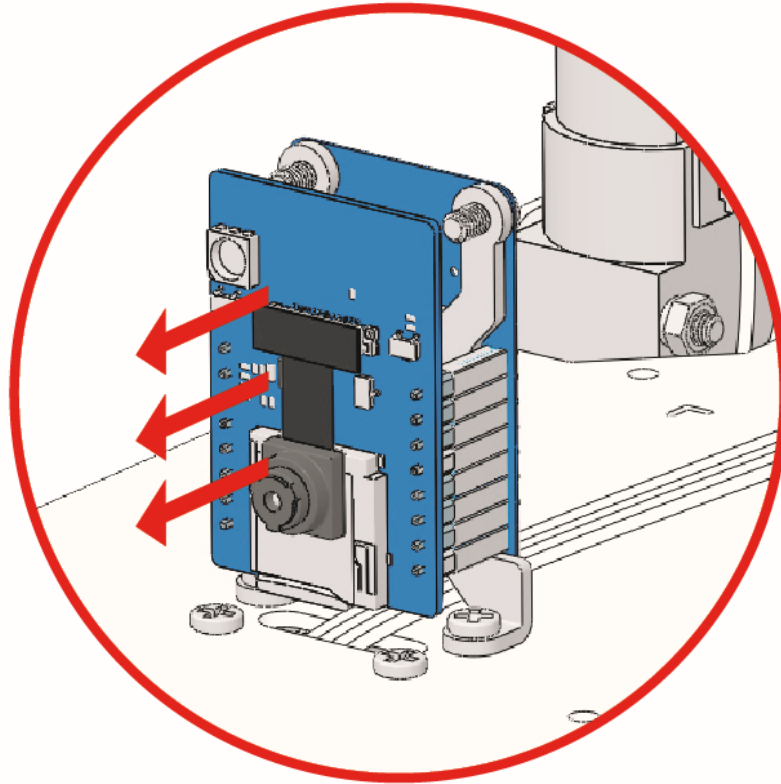
- Kalibrieren Sie das andere Hindernisvermeidungsmodul auf die gleiche Weise.

10. Ultraschall

In diesem Projekt lernen Sie, wie Sie die vom Ultraschallmodul erkannte Entfernung ablesen können.

Wie geht das?

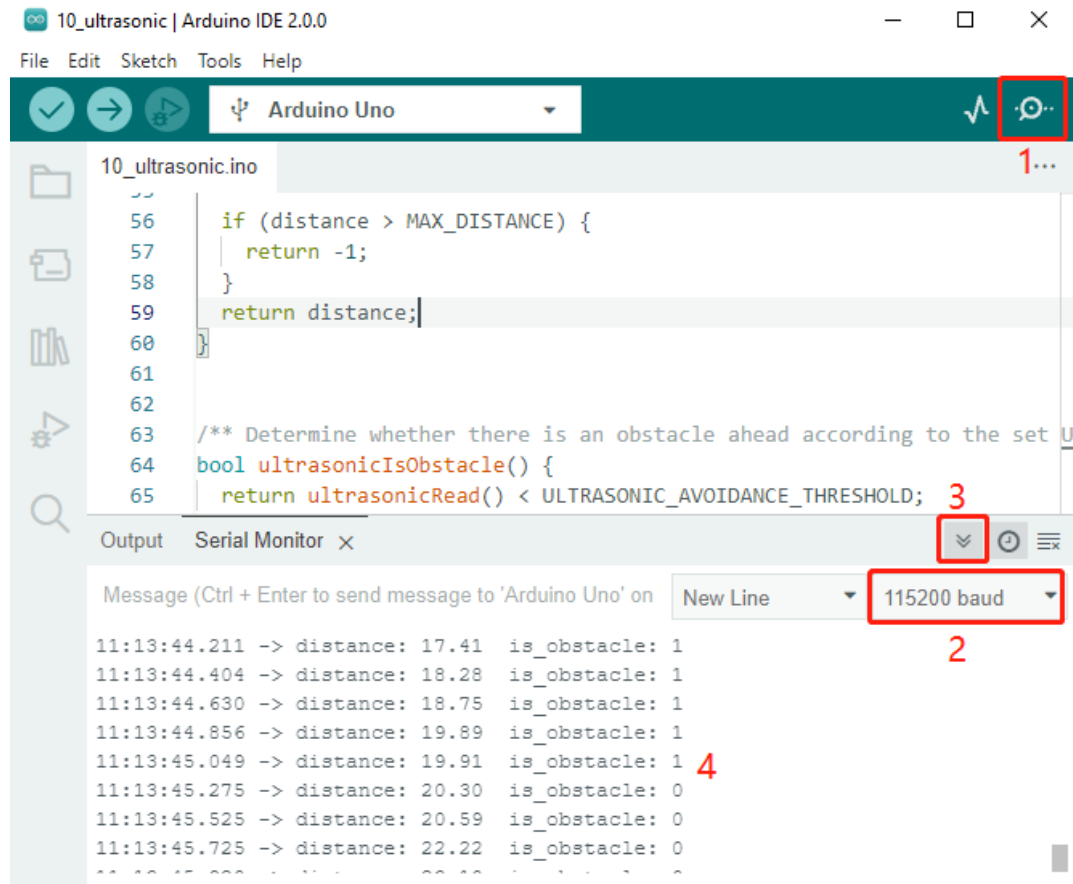
1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



2. Öffnen Sie die Datei `10_ultrasonic.ino` im Pfad `zeus-car-main\examples\10_ultrasonic`.
3. Nachdem der Code erfolgreich hochgeladen wurde, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.

Bemerkung: Trennen Sie in diesem Schritt nicht das USB-Kabel, da Sie die Daten des Ultraschallmoduls auf Ihrem Computer überprüfen müssen.

4. Öffnen Sie den Seriellen Monitor und stellen Sie sicher, dass die aktuelle Baudrate auf 115200 eingestellt ist. Es wird empfohlen, auf das Icon **Automatisches Scrollen umschalten** zu klicken, damit Sie die neuesten gedruckten Daten sehen können.
 - Sie können die gedruckten Daten wie `distance: 21.11 is_obstacle: 0` anzeigen.
 - Wenn der Abstand des Hindernisses vorne innerhalb von 20 cm liegt, wird `is_obstacle: 0` zu `is_obstacle: 1` wechseln.

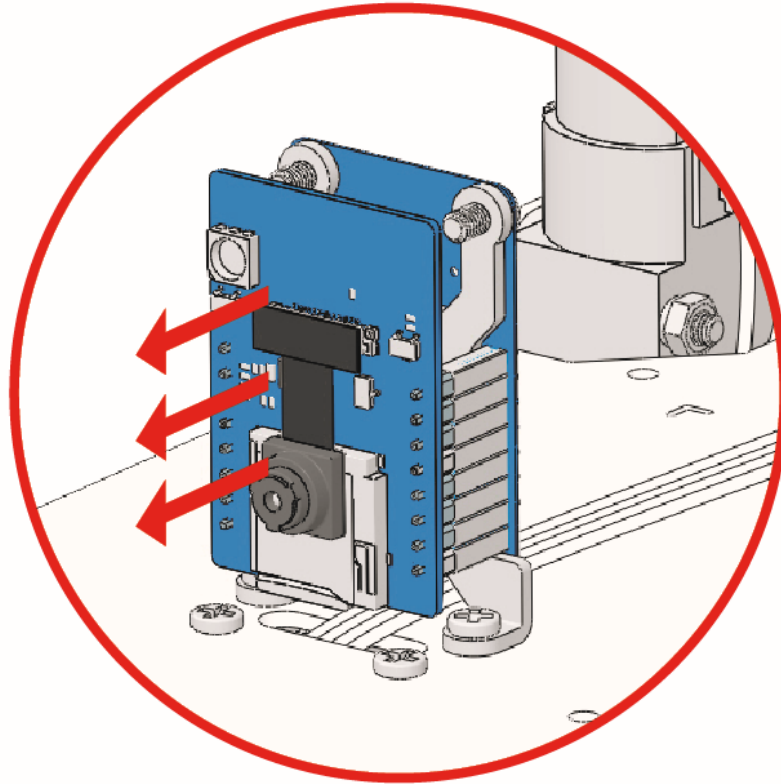


11. Hindernisvermeidung

In diesem Projekt wird das Zeus-Auto automatisch vorwärts fahren. Zwei Hindernisvermeidungsmodule und ein Ultraschallmodul verhindern, dass es gegen Hindernisse stößt.

Wie geht das?

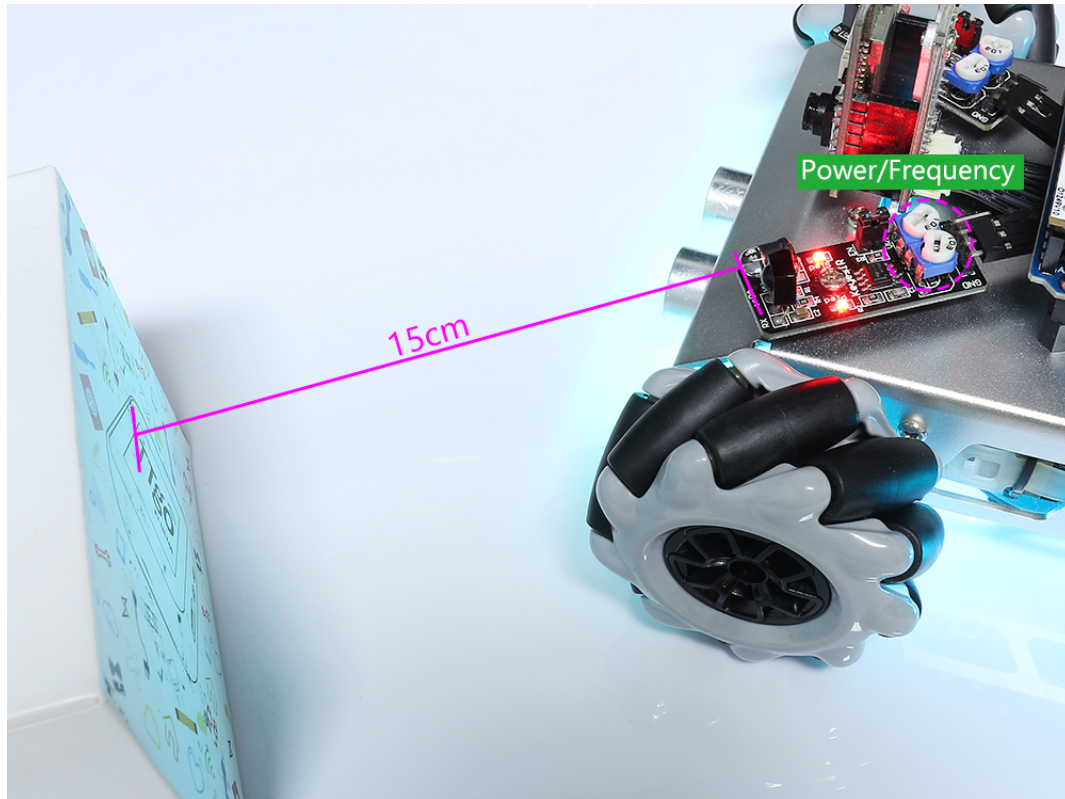
1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



2. Öffnen Sie die Datei `11_obstacle_avoid.ino` im Pfad `zeus-car-main\examples\11_obstacle_avoid`.
3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.
 - Das Zeus-Auto fährt vorwärts.
 - Ein Ultraschallmodul erkennt Hindernisse vorne. Wenn ein Hindernis erkannt wird, dreht das Auto nach links.
 - Wenn das linke Hindernisvermeidungsmodul ein Hindernis erkennt, dreht das Auto nach rechts. Erkennt das rechte Modul ein Hindernis, dreht das Auto nach links.

Bemerkung: Vor der Benutzung müssen Sie den Erkennungsabstand der beiden Hindernisvermeidungsmodule auf 15CM einstellen. Die Schritte sind wie folgt:

- Beginnen Sie mit der Einstellung des rechten Hindernisvermeidungsmoduls. Während des Transports können Kollisionen dazu führen, dass der Sender und Empfänger im Infrarotmodul kippen. Daher müssen Sie sie manuell begradigen.
- Stellen Sie ein Hindernis in etwa 15 cm Entfernung vom IR-Hindernisvermeidungsmodul auf.
- Am Modul gibt es zwei Potentiometer. Eines reguliert die Sendeleistung, das andere die Sendefrequenz. Durch das Einstellen dieser beiden Potentiometer können Sie den Erkennungsabstand anpassen.
- Dann können Sie ein Potentiometer einstellen. Wenn bei 15 cm die Signalleuchte auf dem Modul aufleuchtet, ist die Einstellung erfolgreich. Wenn nicht, stellen Sie das andere Potentiometer ein.



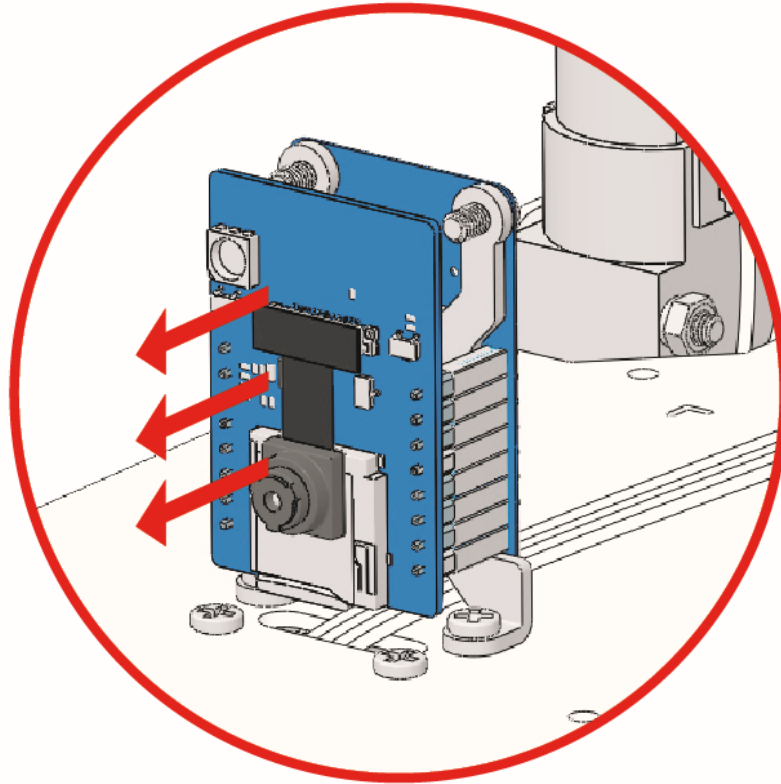
- Kalibrieren Sie das andere Hindernisvermeidungsmodul auf dieselbe Weise.

12. Folgen

In diesem Projekt werden dieselben zwei Hindernisvermeidungsmodule und ein Ultraschallmodul verwendet. Allerdings nicht zur Hindernisvermeidung, sondern um Ihnen zu folgen.

Wie geht das?

1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



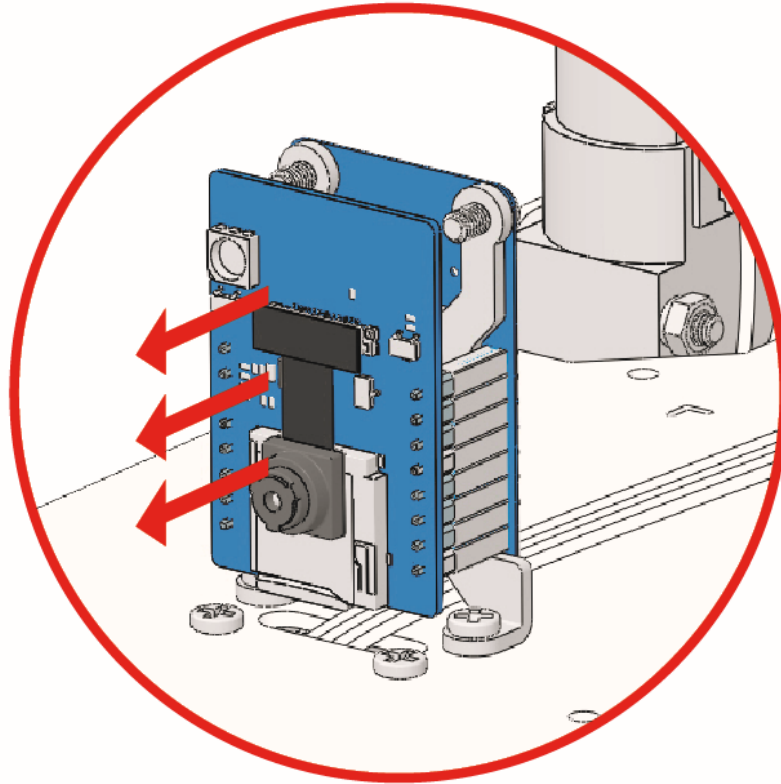
2. Öffnen Sie die Datei `12_follow.ino` im Pfad `zeus-car-main\examples\12_follow`.
3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.
4. In diesem Fall wird das Zeus-Auto nicht fahren, wenn sich nichts vor ihm befindet. Sobald Sie sich nach vorne bewegen und etwa 20 cm von ihm entfernt stehen, wird es Ihnen folgen.

13. Graustufen

In diesem Projekt lernen Sie, wie Sie das Omni Graustufen-Modul auf der Unterseite des Zeus-Autos lesen und kalibrieren.

Wie geht das?

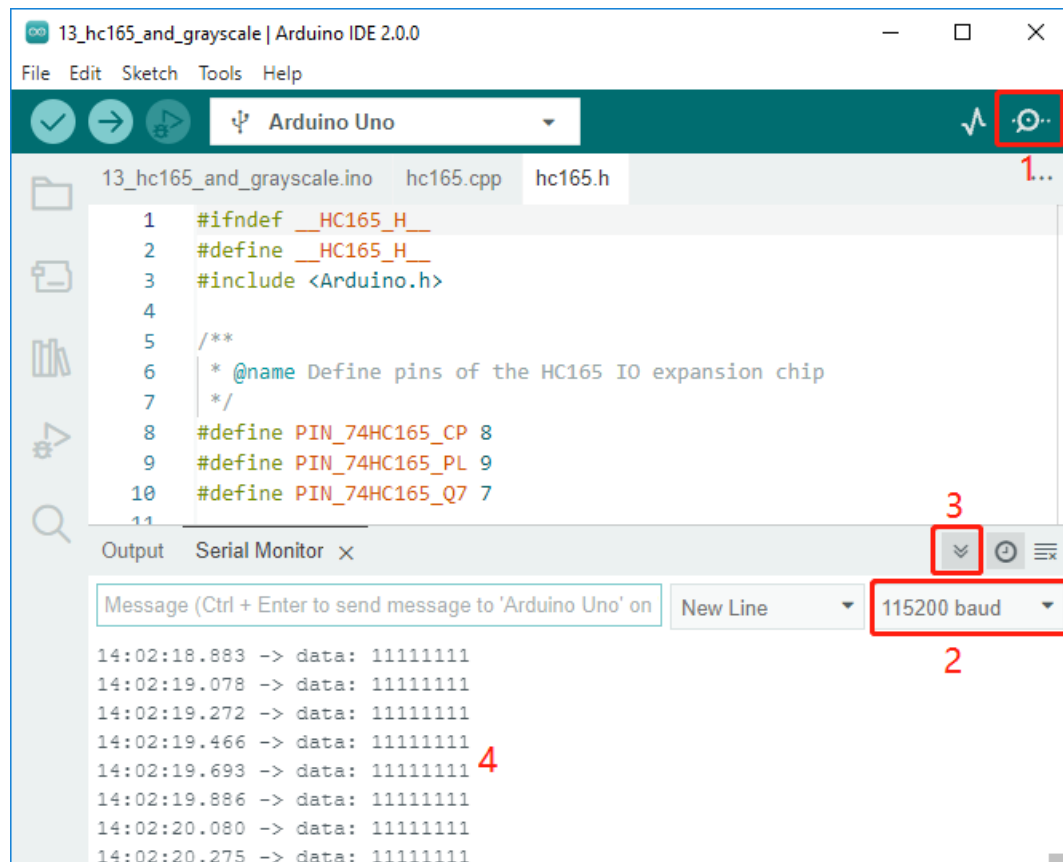
1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



2. Öffnen Sie die Datei `13_hc165_and_grayscale.ino` im Pfad `zeus-car-main\examples\13_hc165_and_grayscale`.
3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.

Bemerkung: Ziehen Sie in diesem Schritt nicht das USB-Kabel ab, da Sie die Daten des Omni Graustufen-Moduls auf Ihrem Computer überprüfen müssen.

4. Öffnen Sie den seriellen Monitor und stellen Sie sicher, dass die aktuelle Baudrate auf 115200 eingestellt ist. Es wird empfohlen, auf das **Automatisches Scrollen umschalten** Symbol zu klicken, damit Sie die neuesten Daten sehen können.

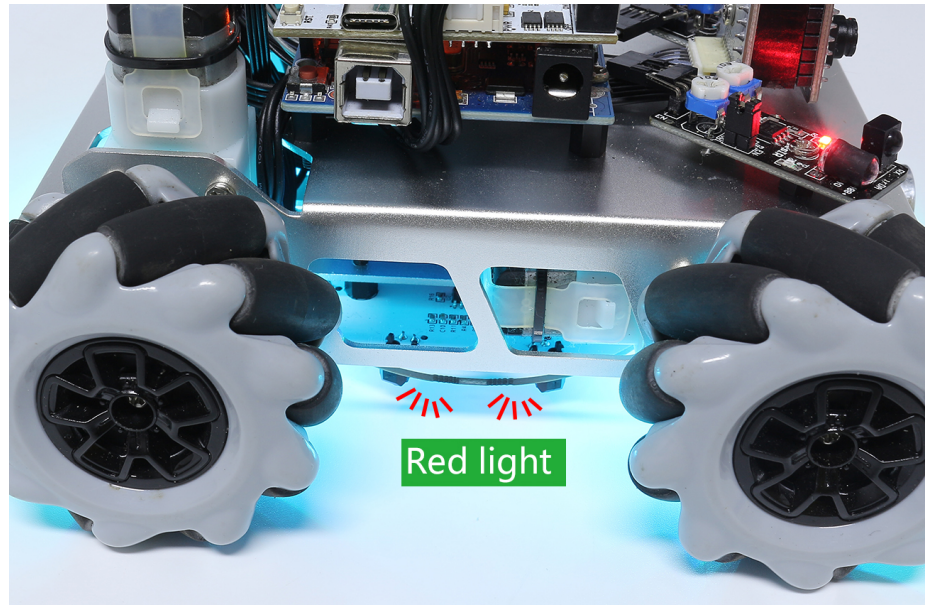


- Drehen Sie das Auto um und im seriellen Monitor wird data: 11111111 angezeigt.
- Wenn Sie einen der Sensoren auf dem Modul mit der Hand abdecken, wird der Wert an der entsprechenden Position zu 0.
- Zum Beispiel, wenn Sie den U11 Sensor abdecken, sehen Sie data: 01111111.

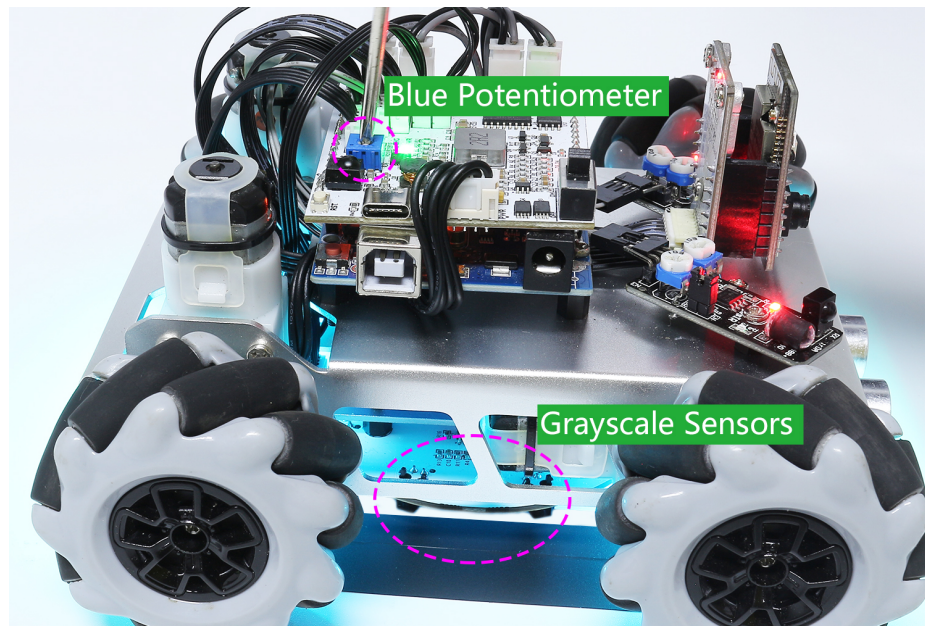
5. Kalibrieren Sie das Omni Graustufen-Modul.

Da jeder Untergrund unterschiedliche Graustufenwerte hat, ist die werkseitig eingestellte Graustufenschwelle möglicherweise nicht für Ihre aktuelle Umgebung geeignet. Daher müssen Sie dieses Modul vor der Verwendung kalibrieren. Es wird empfohlen, die Kalibrierung durchzuführen, wenn sich die Bodenfarbe stark ändert.

- Stellen Sie das Zeus-Auto auf eine weiße Oberfläche und drehen Sie das Potentiometer, bis das Licht des Grausensors gerade leuchtet.



- Lassen Sie nun die beiden Grausensoren an der Seite genau zwischen der schwarzen Linie und der weißen Oberfläche positionieren und drehen Sie das Potentiometer langsam, bis die Signalanzeige gerade erlischt.



- Bewegen Sie das Auto wiederholt über die schwarze Linie und die weiße Oberfläche, um sicherzustellen, dass die Lichter des Grausensors aus sind, wenn sie zwischen der schwarzen Linie und der weißen Oberfläche liegen, und leuchten, wenn sie auf der weißen Oberfläche sind. Das zeigt, dass das Modul erfolgreich kalibriert wurde.

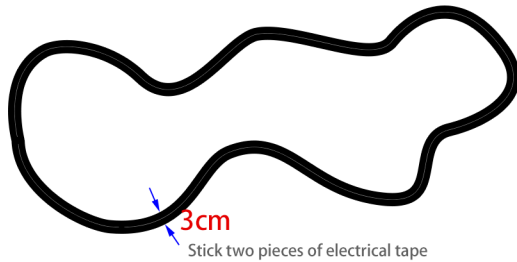
14. Linienverfolgung

In diesem Projekt lernen Sie, das Omni-Grayscale-Modul zur Linienverfolgung zu verwenden.

Bevor Sie mit dem Projekt beginnen, sollten Sie mit schwarzem Isolierband eine Linie aufkleben. Dies kann ein Kreis, eine gerade Linie oder eine unregelmäßige Form sein.

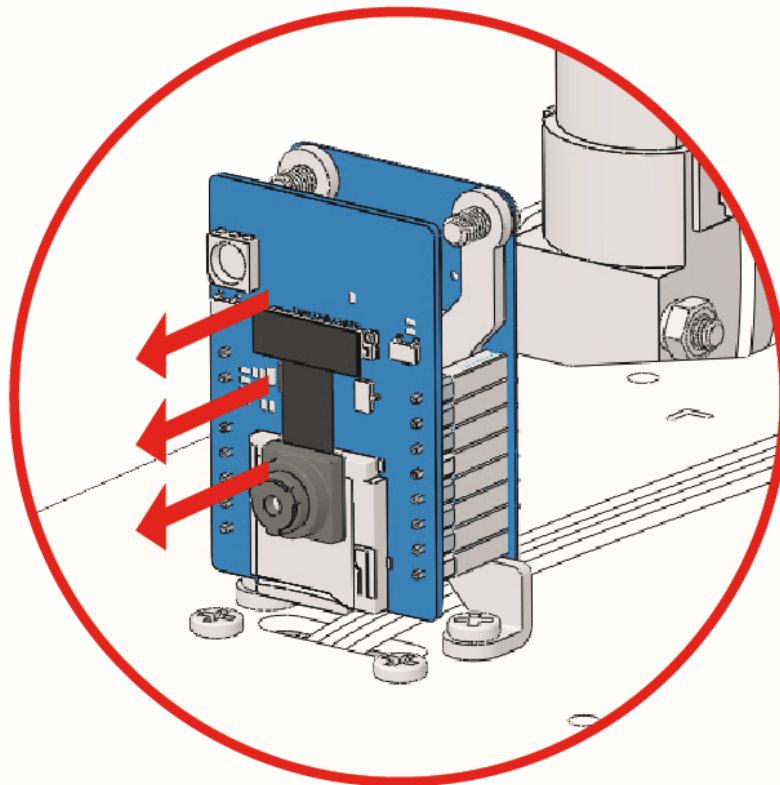
Dabei gibt es jedoch zwei Dinge zu beachten:

1. Diese Linie sollte 3cm breit sein (die Dicke eines Stücks Isolierband beträgt 1,5cm).
2. Der Biege-Winkel sollte nicht kleiner als 90° sein.



Wie geht das?

1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



2. Öffnen Sie die Datei `14_line_track.ino` im Pfad `zeus-car-main\examples\14_line_track`.

3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.
4. Anschließend wird Ihr Zeus-Auto der Linie folgen. Wenn Ihr Zeus-Auto von der Linie abkommt, könnte dies bedeuten, dass Sie das [13. Graustufen](#) neu kalibrieren müssen oder die Geschwindigkeit verringern sollten.

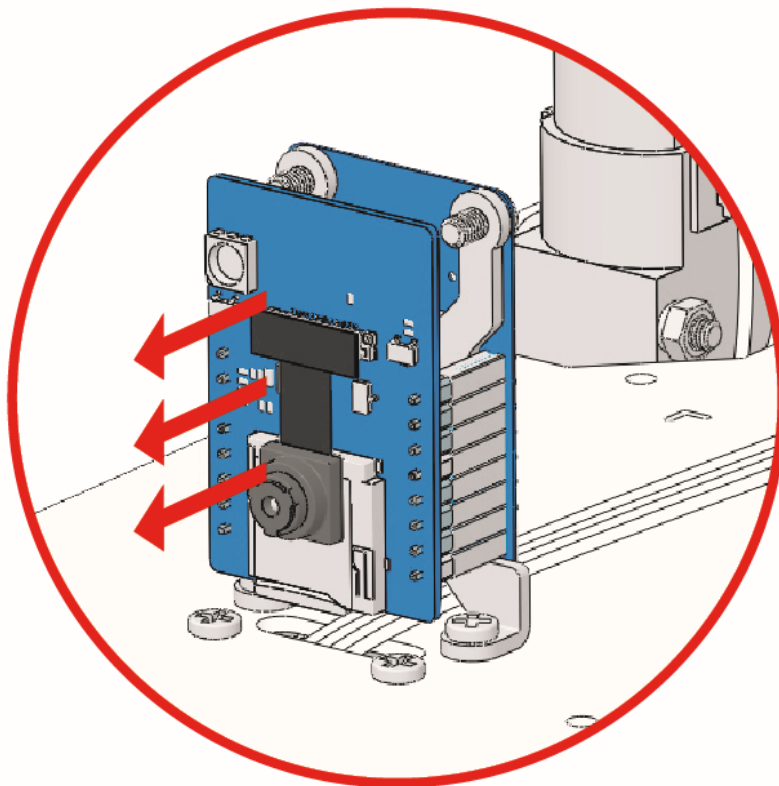
15. Feldzentrisches Linienverfolgen

In diesem Projekt wird eine andere Art des Tracer-Modus untersucht - die Tracer-Bewegung mit Kompass.

Das experimentelle Ergebnis ähnelt fast dem [14. Linienverfolgung](#), jedoch zeigt der Kopf des Zeus-Autos immer in eine feste Richtung; im vorherigen Projekt änderte er sich mit der Linienrichtung.

Wie geht das?

1. ESP32-CAM und das Arduino-Board nutzen dieselben RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes sollten Sie daher zunächst den ESP32-CAM trennen, um mögliche Konflikte oder Probleme zu vermeiden.



2. Öffnen Sie die Datei `15_line_track_field_centric.ino` im Pfad `zeus-car-main\examples\15_line_track_field_centric`.
3. Nach erfolgreichem Hochladen des Codes, schieben Sie den Ein-/Ausschalter auf ON, um das Zeus-Auto zu starten.
4. Als Nächstes wird Ihr Zeus-Auto der Linie folgen, allerdings mit seinem Kopf in einer Richtung ausgerichtet. Wenn Ihr Zeus-Auto von der Linie abkommt, könnte dies bedeuten, dass Sie das [13. Graustufen](#) neu kalibrieren müssen oder die Geschwindigkeit verringern sollten.

16. KI-Erkennung von der APP

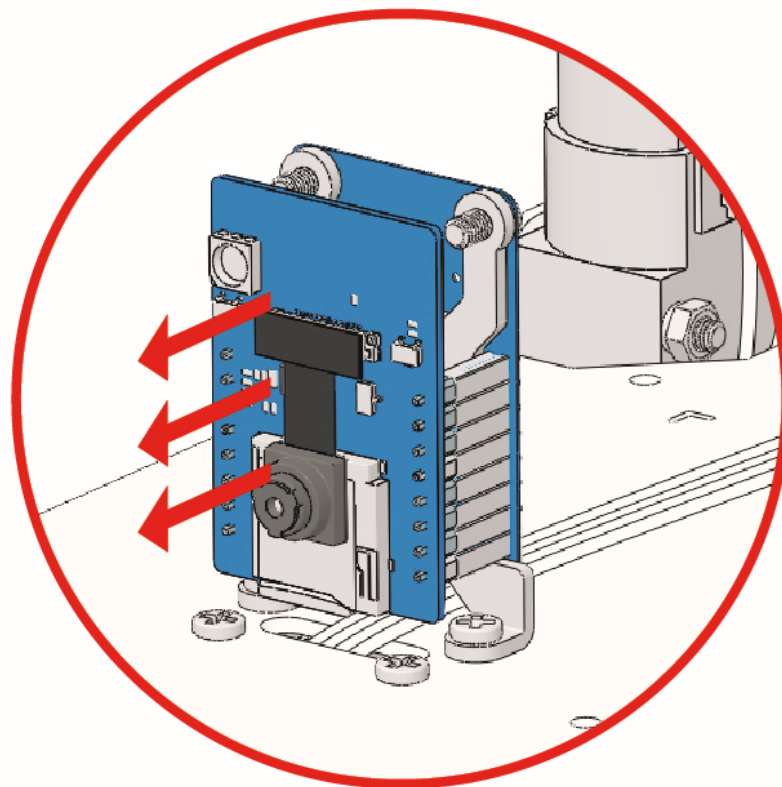
In diesem Projekt verwenden wir eine App - SunFounder Controller - um Echtzeit-Video-Streaming zu betrachten und die KI-Erkennungsfunktionen auf Ihrem mobilen Gerät, wie Gesichts- und Pose-Erkennung, zu nutzen.

Der SunFounder Controller ist eine Anwendung, die es Benutzern ermöglicht, den Controller für die Steuerung ihres Roboters oder als IoT-Plattform anzupassen. Diese App integriert 11 verschiedene Anzeige- und Steuerungs-Widgets, wie Button, Joystick, Anzeigeinstrument und Radar. Die Controller-Seite verfügt über 17 Bereiche von A~Q, in denen Sie unterschiedliche Widgets platzieren können, um Ihren eigenen Controller zu gestalten.

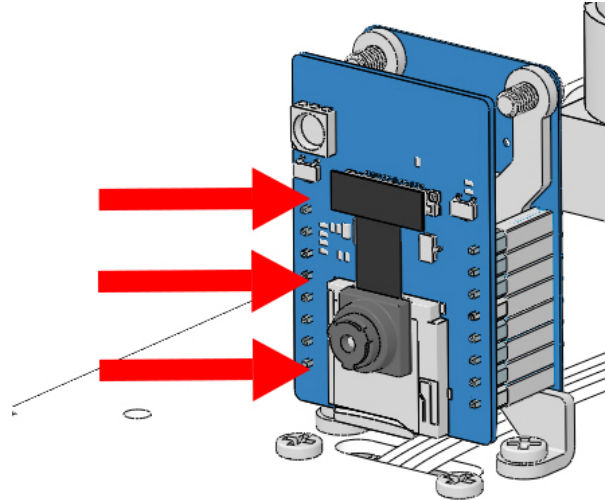
Sehen Sie sich nun an, wie Sie die App auf einfache Weise nutzen können.

Wie geht das?

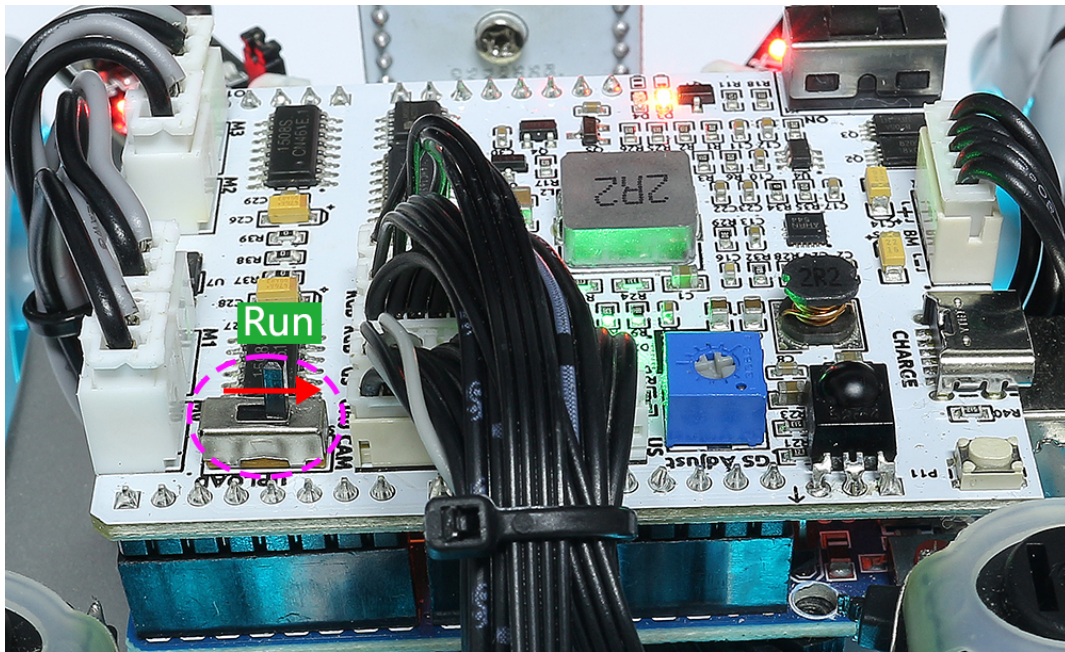
1. Das ESP32-CAM und das Arduino-Board verwenden dieselben RX (Empfang) und TX (Senden) Pins. Wenn Sie also Code hochladen, müssen Sie zuerst das ESP32-CAM trennen, um Konflikte oder mögliche Probleme zu vermeiden.



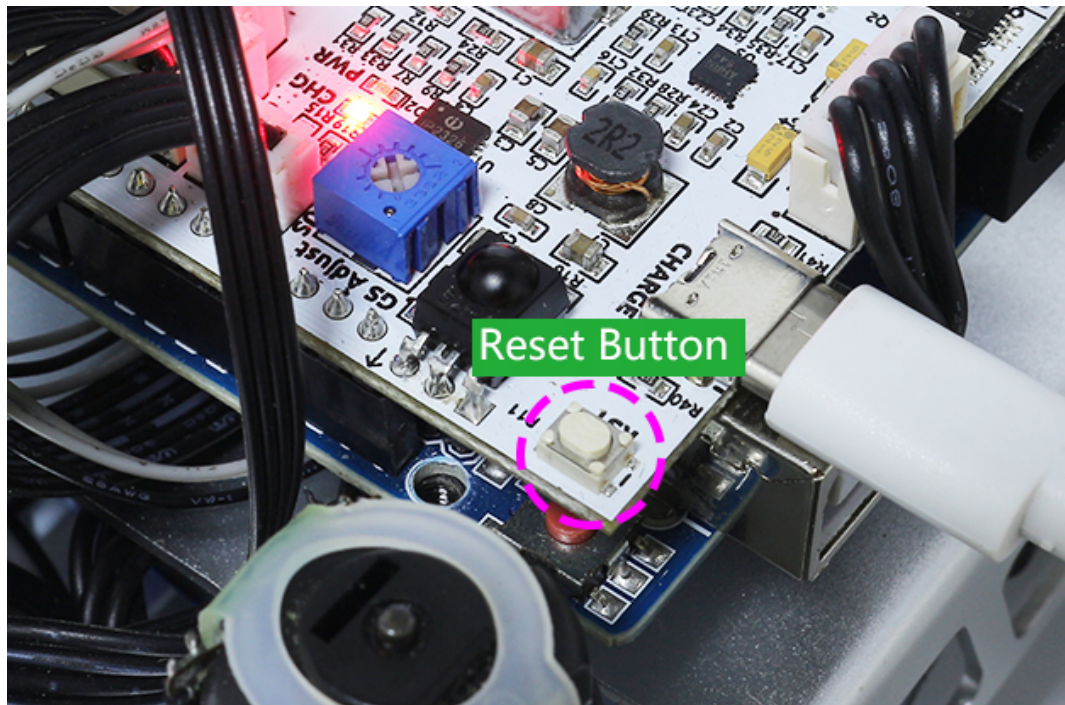
2. Öffnen Sie die Datei `16_ai_detection_from_app.ino` unter dem Pfad `zeus-car-main\examples\16_ai_detection_from_app`.
3. Nachdem der Code erfolgreich hochgeladen wurde, können Sie das ESP32-CAM einstecken und dann den Stromschalter auf EIN schieben, um das Zeus-Auto zu starten.



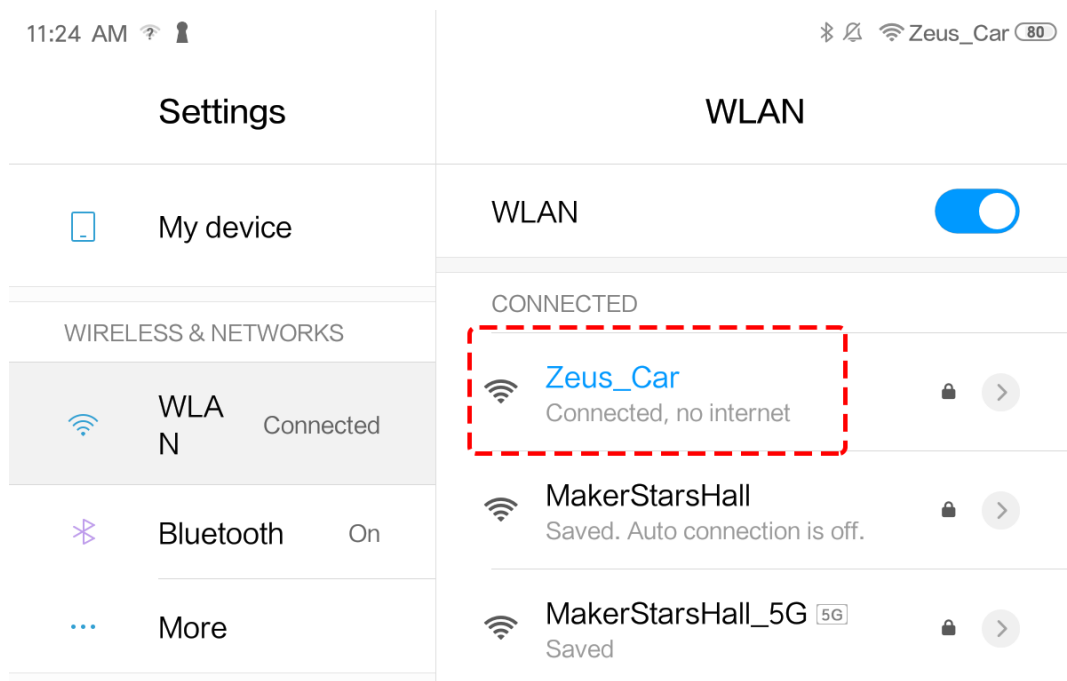
4. Schalten Sie den Upload-Schalter zur Seite „Run“ (auf diesem Diagramm die rechte Seite), um das ESP32 CAM zu starten.



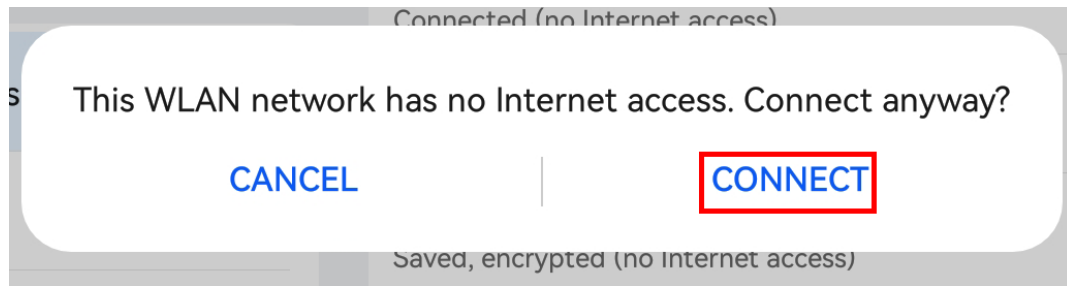
5. Drücken Sie die Reset-Taste, um das Programm des Arduino-Boards erneut zu starten.



6. Installieren Sie nun **SunFounder Controller** aus dem **APP Store(iOS)** oder **Google Play(Android)** auf Ihrem mobilen Gerät.
7. Verbinden Sie sich mit dem WLAN Zeus_Car.
 - Finden Sie Zeus_Car im WLAN Ihres Mobiltelefons (Tablet), geben Sie das Passwort 12345678 ein und verbinden Sie sich damit.



- Der Standardverbindungsmodus ist der AP-Modus. Nach der Verbindung erhalten Sie eine Meldung, dass dieses WLAN-Netzwerk keinen Internetzugang bietet. Bitte wählen Sie „Trotzdem verbinden“.

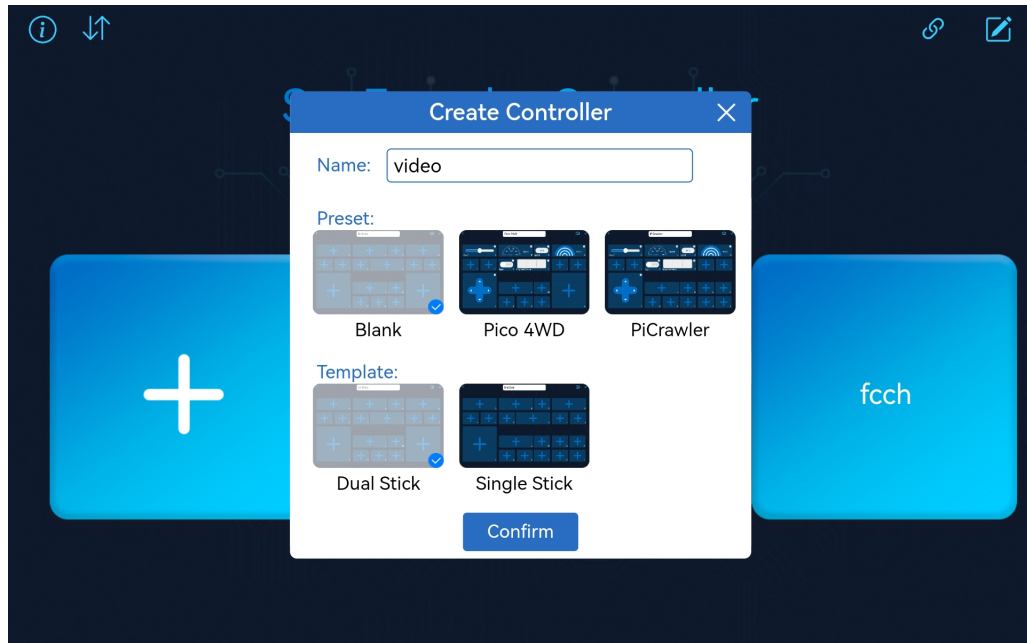



8. Öffnen Sie anschließend die App und erstellen Sie einen Controller.

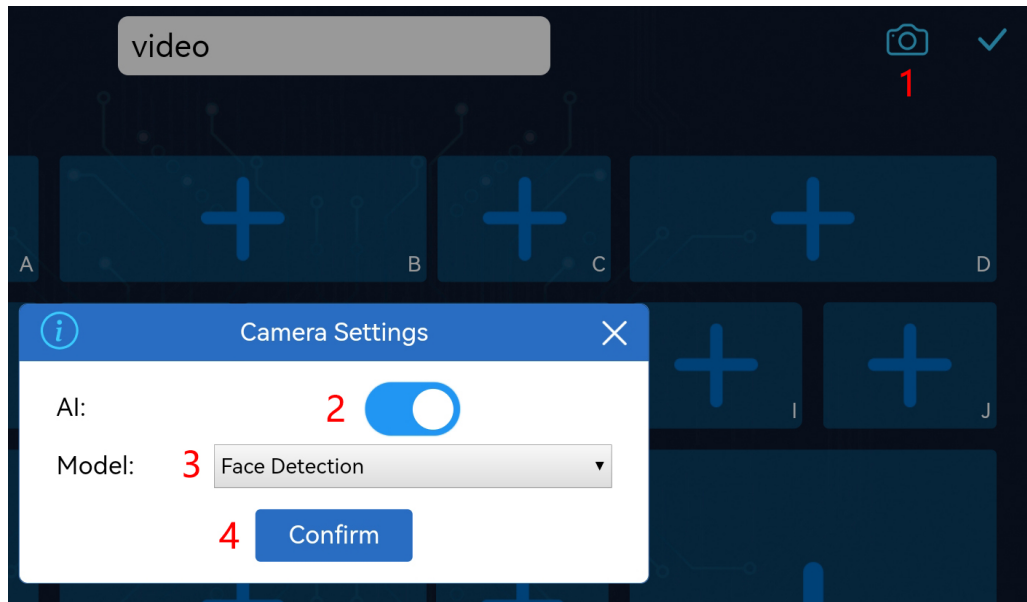
- Um einen Controller im SunFounder Controller hinzuzufügen, klicken Sie auf das + Symbol.




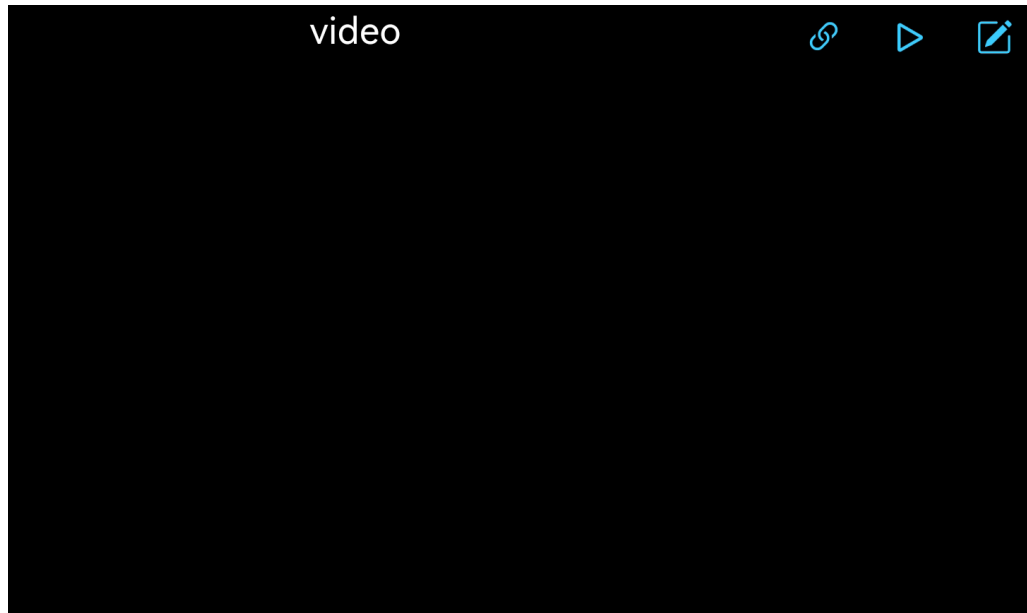
- Wählen Sie das **Blank** und **Dual Stick** Template und geben Sie ihm einen Namen.




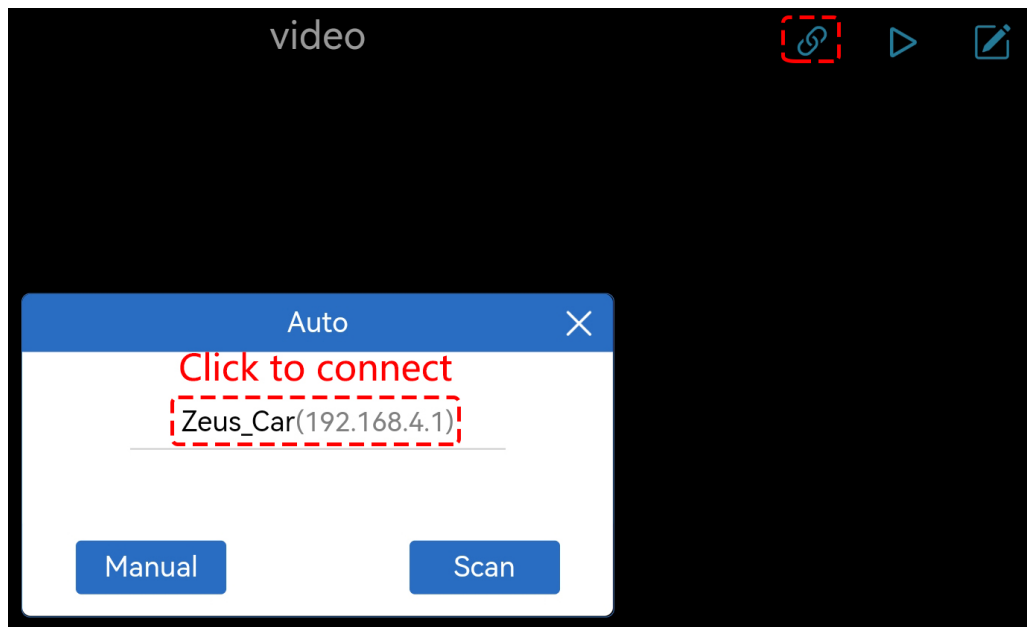
- Wenn Sie sich nun im Controller befinden, klicken Sie auf das  Symbol, um die KI-Erkennungsfunktion zu aktivieren. Sie finden Optionen sowohl für die Gesichtserkennung als auch für die Pose-Erkennung.



- Speichern Sie alle Einstellungen, indem Sie auf den  Button klicken. Da keine Widgets ausgewählt wurden, ist der Bildschirm komplett schwarz.




- Der nächste Schritt ist, das Zeus Car mit Ihrem Gerät über den  Button zu verbinden. Warten Sie ein paar Sekunden und Zeus_Car(IP) wird angezeigt, klicken Sie darauf, um die Verbindung herzustellen.



Bemerkung: Bitte stellen Sie sicher, dass Ihr WLAN mit Zeus_Car verbunden ist, falls Sie die obige Nachricht längere Zeit nicht sehen.

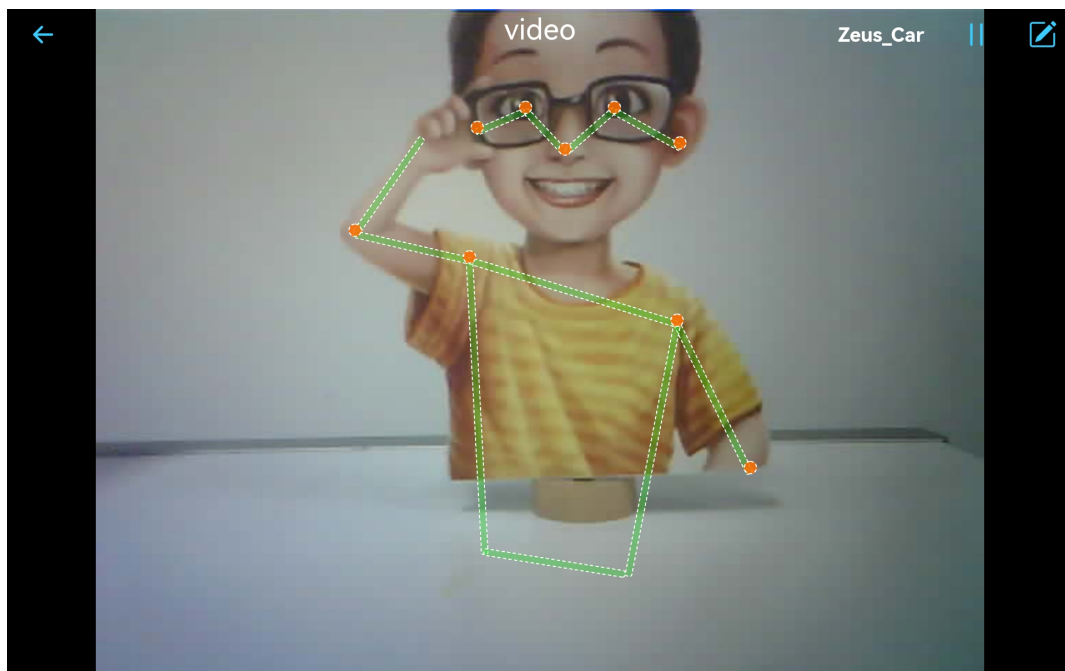
9. Starten Sie den Controller.


Nachdem die Nachricht „Connected Successfully“ erscheint, klicken Sie auf den  Button, dann wird das Kamerabild in der App angezeigt.

Wenn Sie die Gesichtserkennung aktiviert haben, wird das im Bildschirm erscheinende Gesicht umrahmt.



Ist die Pose-Erkennung aktiviert, wird die Pose dargestellt.



Bemerkung: Wenn Sie zu einer anderen KI-Erkennung wechseln möchten, klicken Sie auf das  Symbol und wiederholen Sie die oben genannten Schritte.

17. APP-Steuerung

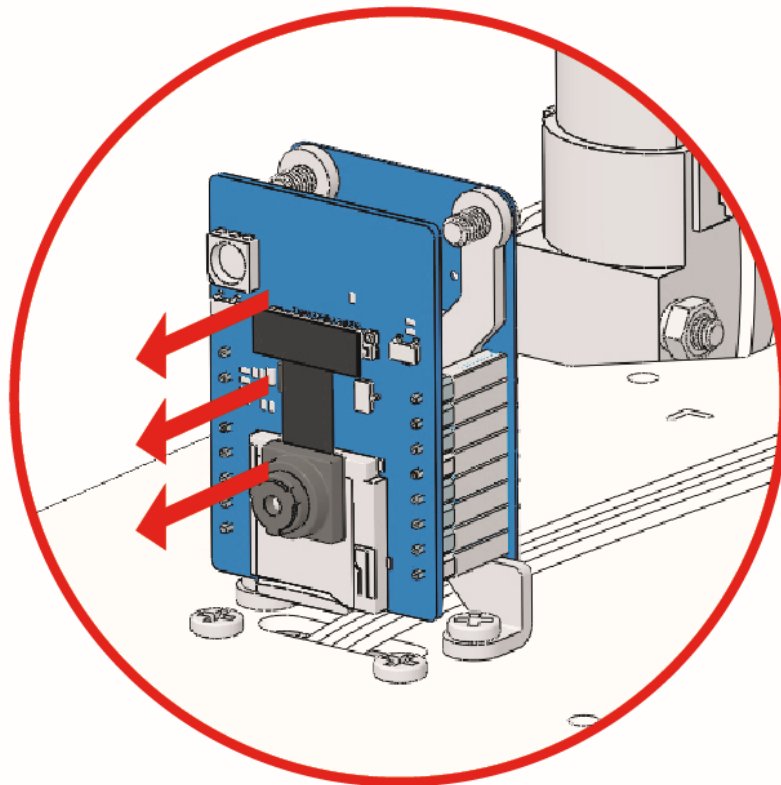
Im vorherigen Projekt haben wir uns einfach die Aufnahmen der Kamera angesehen und die AI-Erkennungsfunktion im SunFounder Controller verwendet.

In diesem Projekt werden wir versuchen, die Bewegung und Richtung des Zeus Autos mit dem Joystick-Widget in der App zu steuern. Schauen wir uns also an, wie das funktioniert.

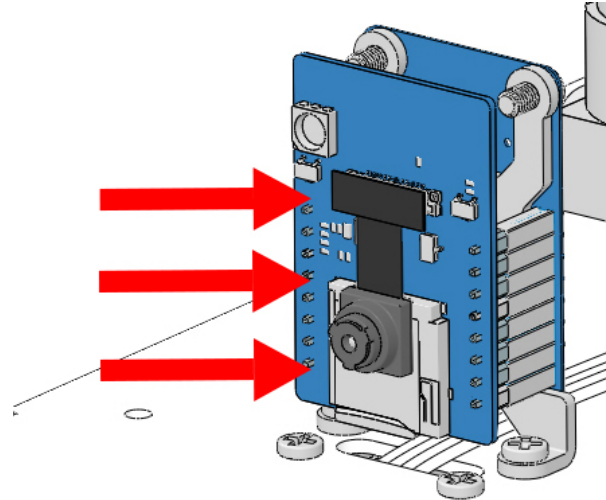
Bemerkung: Bitte installieren Sie den [SunFounder Controller](#) aus dem **APP Store(iOS)** oder **Google Play(Android)**.

Wie geht das?

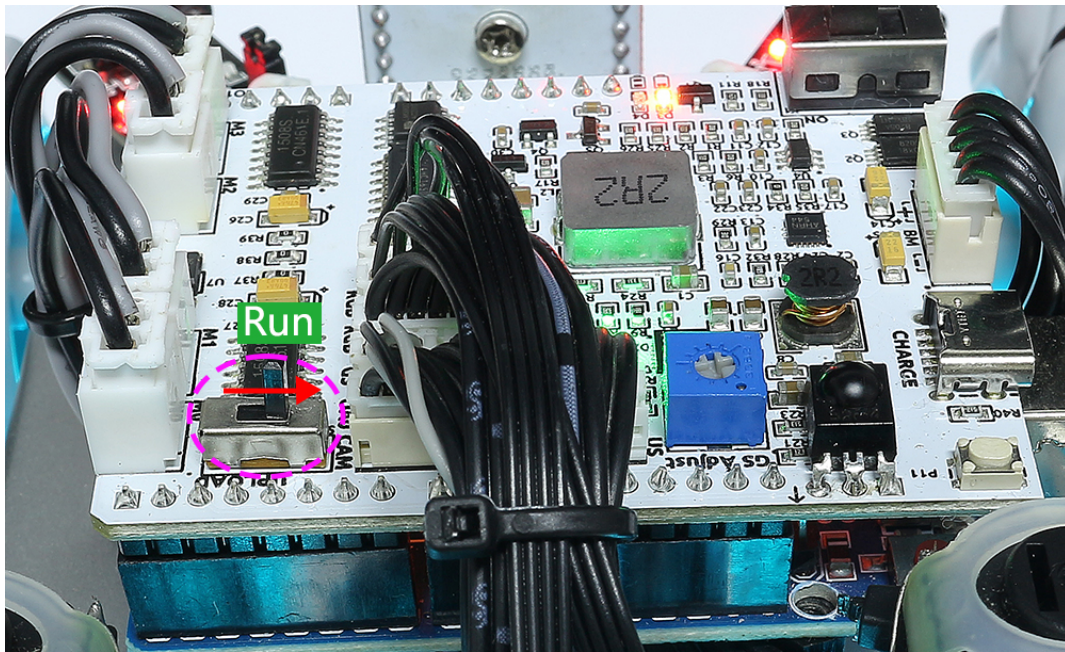
1. Der ESP32-CAM und das Arduino-Board verwenden die gleichen RX (Empfang) und TX (Übertragung) Pins. Beim Hochladen des Codes müssen Sie daher zuerst den ESP32-CAM trennen, um Konflikte oder mögliche Probleme zu vermeiden.



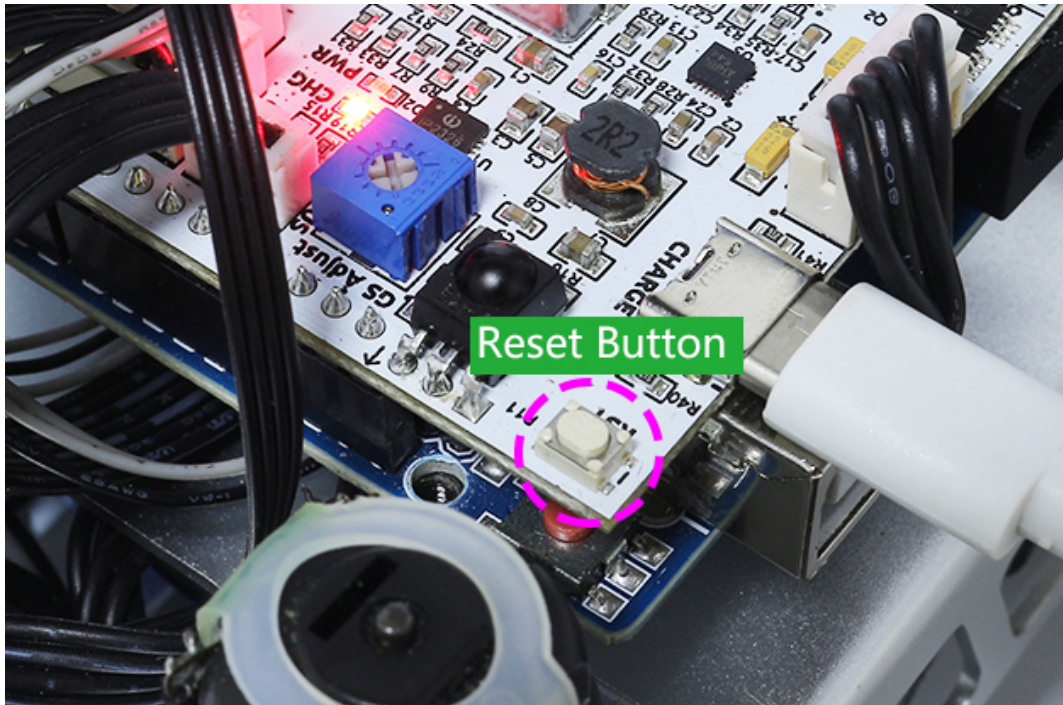
2. Öffnen Sie die Datei `17_app_control.ino` im Pfad `zeus-car-main\examples\17_app_control`.
3. Nachdem der Code erfolgreich hochgeladen wurde, können Sie das ESP32-CAM einstecken und dann den Stromschalter auf EIN schieben, um das Zeus-Auto zu starten.



4. Wechseln Sie den Upload-Schalter auf die Seite „Run“ (rechte Seite auf diesem Diagramm), um die ESP32 CAM zu starten.

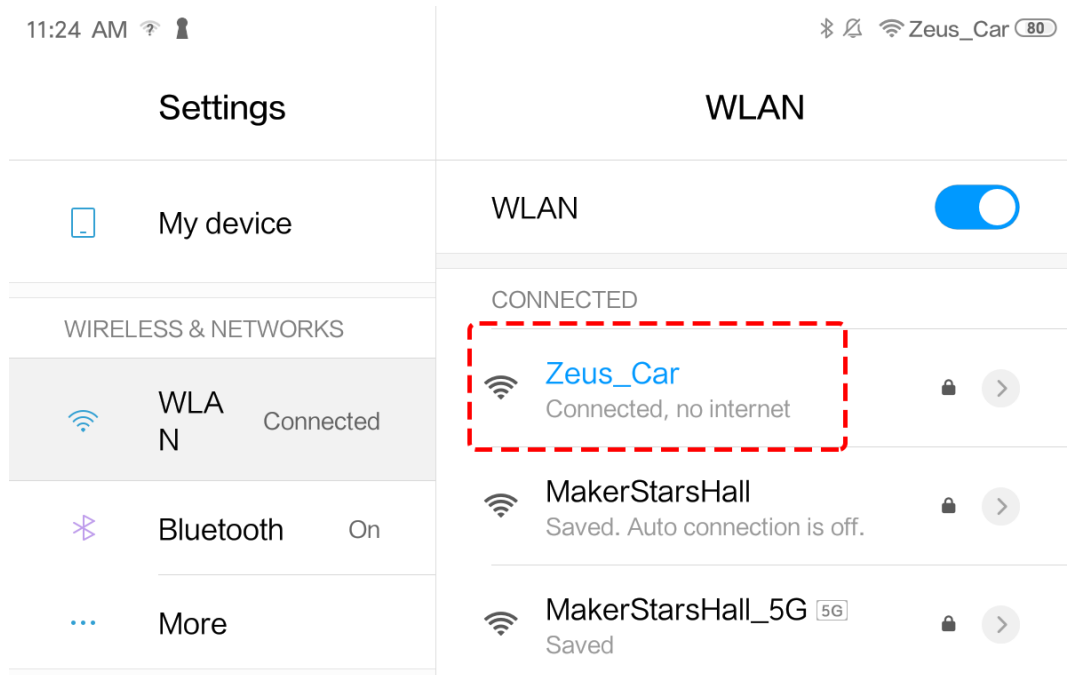


5. Drücken Sie die Reset-Taste, um das Programm des Arduino-Boards erneut zu starten.

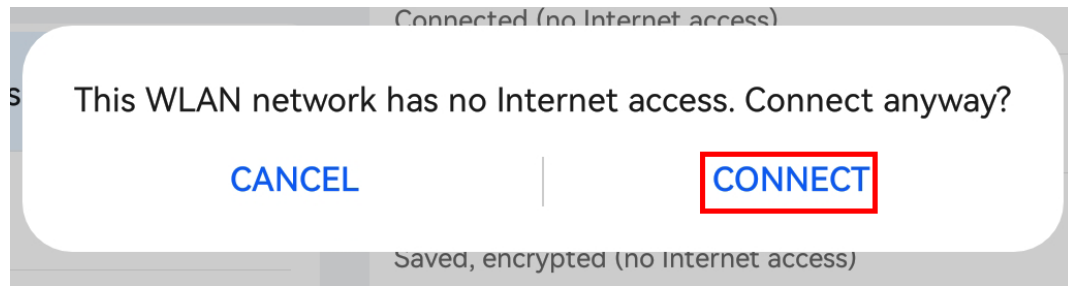


6. Verbinden Sie sich mit dem Zeus_Car WLAN.

- Finden Sie Zeus_Car im WLAN Ihres Mobiltelefons (Tablet), geben Sie das Passwort 12345678 ein und stellen Sie die Verbindung her.



- Der Standardverbindungsmodus ist der AP-Modus. Nach der Verbindung wird angezeigt, dass dieses WLAN-Netzwerk keinen Internetzugang hat. Wählen Sie weiterhin „Verbinden“.



7. Erstellen Sie einen Controller.

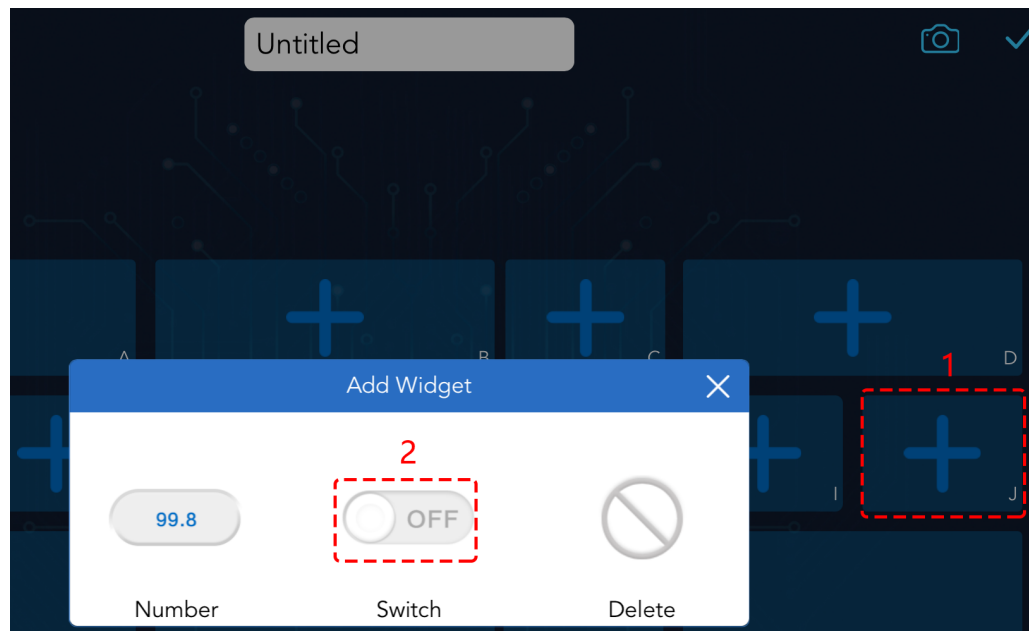
- Um einen Controller im SunFounder Controller hinzuzufügen, klicken Sie auf das + Symbol.



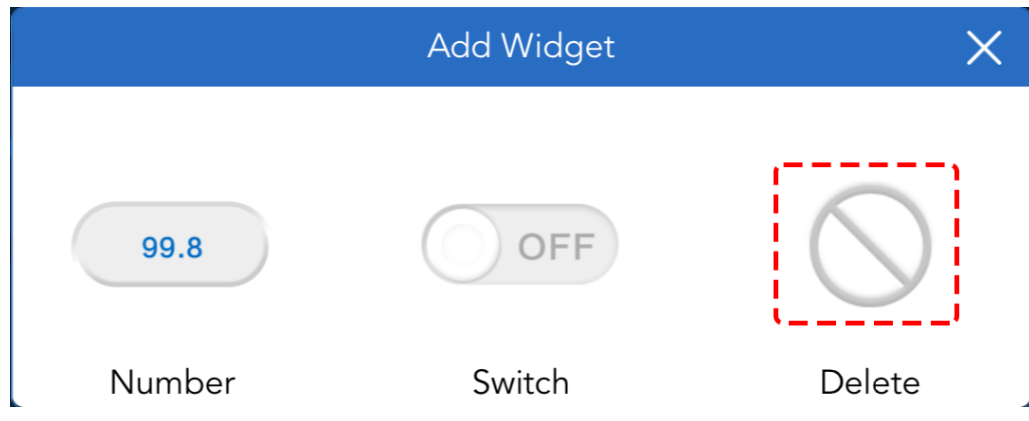
- Einige Produkte haben im Abschnitt „Voreinstellungen“ bereits vorgefertigte Controller. Wir wählen hier jedoch ein **Blanko** und **Dual Stick** Template.



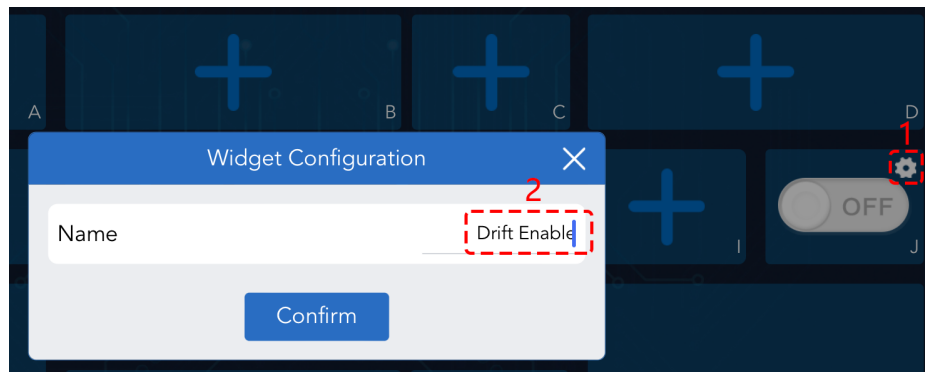
- Klicken Sie im Bereich J auf das + Symbol und wählen Sie ein Schalter-Widget. Möglicherweise müssen Sie nach links wischen, um das Widget zu sehen.




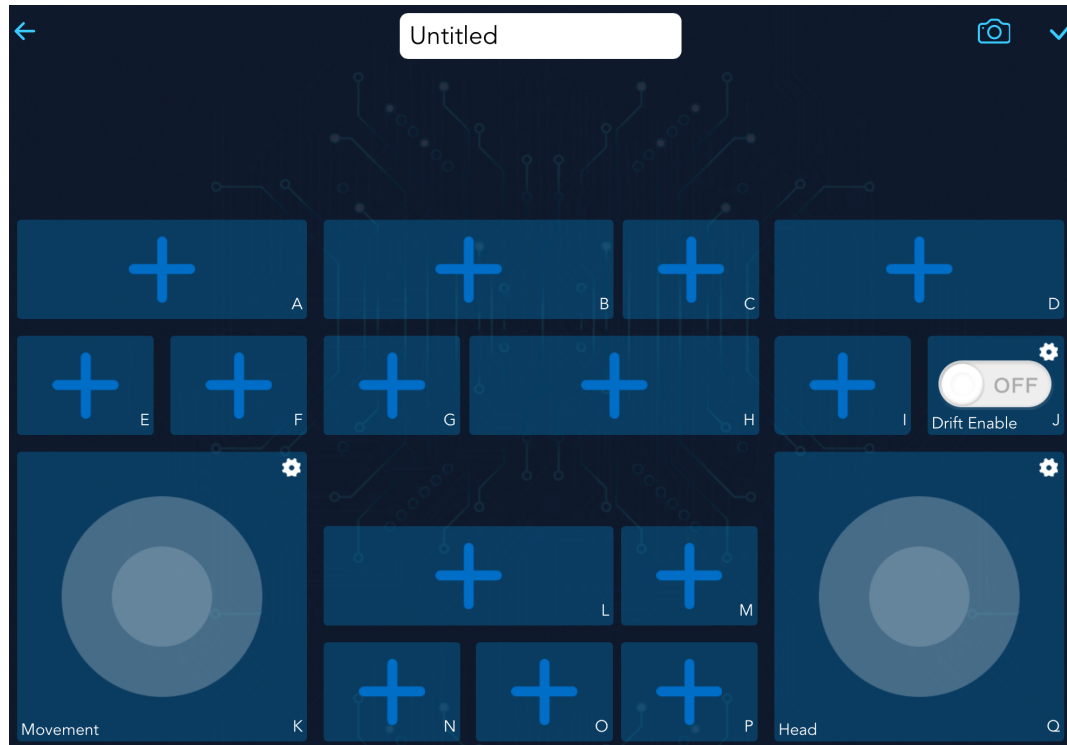
Bemerkung: Wenn Sie denken, dass Sie das falsche Widget ausgewählt haben, können Sie darauf klicken und es mit der Löschtaaste entfernen.




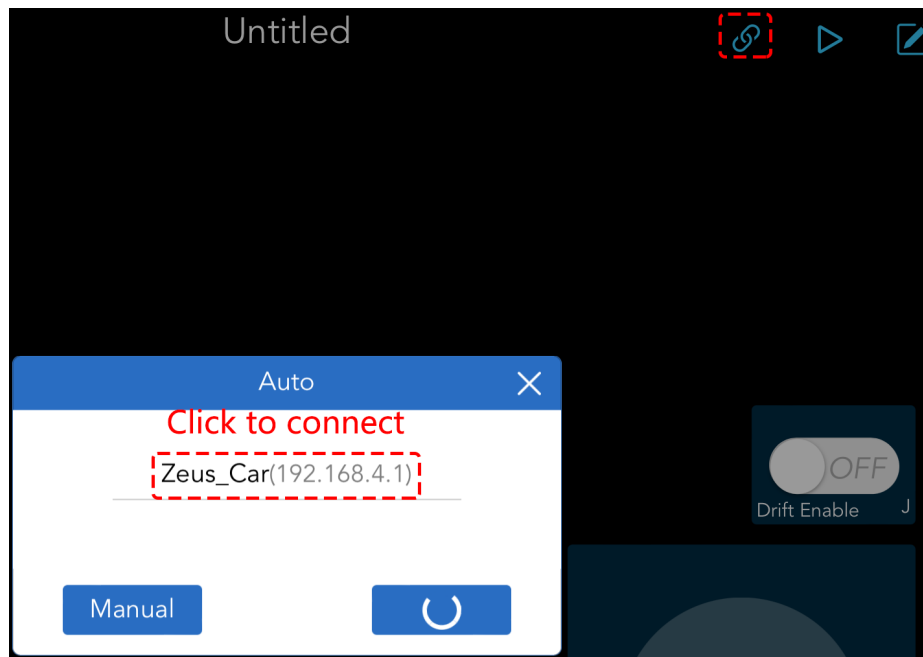
- Klicken Sie auf das Einstellungssymbol in der oberen rechten Ecke, um ihm einen Namen zu geben.



- Wählen Sie im Bereich K und Q ein Joystick-Widget und benennen Sie beide Widgets. Nutzen Sie dann die  Taste, um den Controller zu speichern.




- Der nächste Schritt ist, das Zeus Auto über die  Taste mit Ihrem Gerät zu verbinden. Warten Sie einige Sekunden, und Zeus_Car (IP) wird angezeigt. Klicken Sie darauf, um die Verbindung herzustellen.



Bemerkung: Bitte stellen Sie sicher, dass Ihr WLAN mit Zeus_Car verbunden ist, wenn Sie die oben genannte Nachricht lange nicht sehen.

8. Starten Sie den Controller.

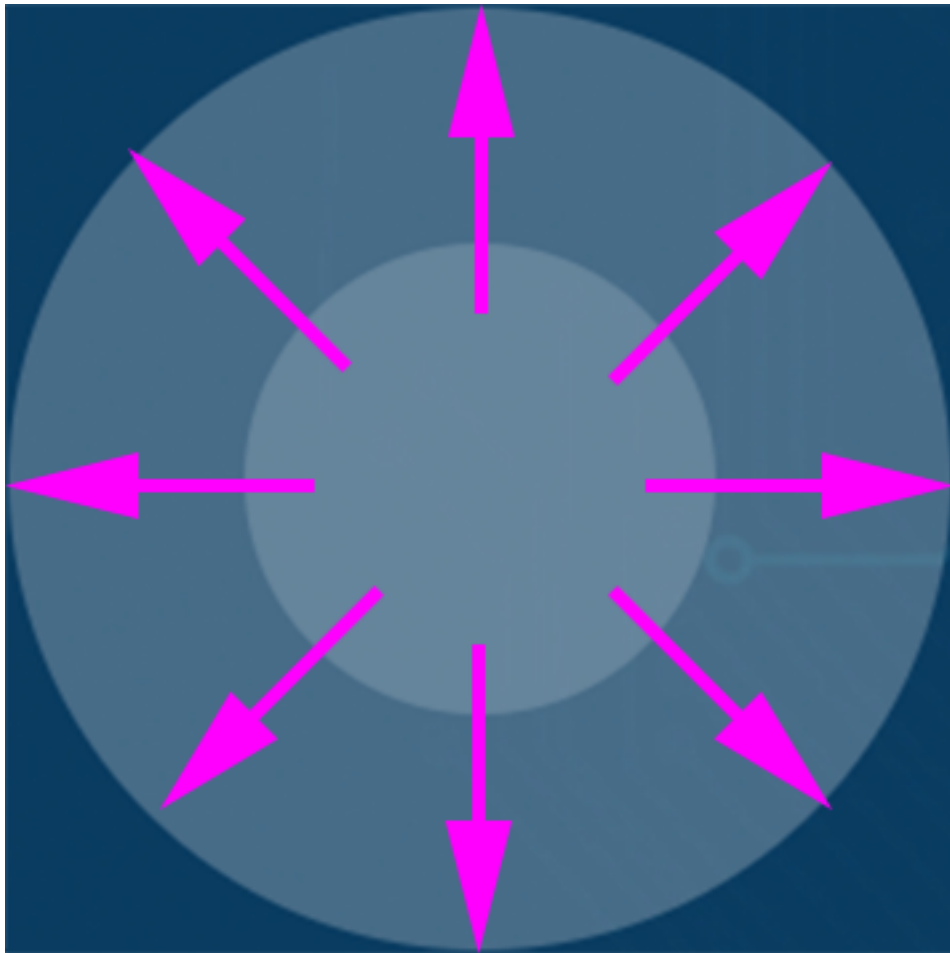
Nachdem die Nachricht „Erfolgreich verbunden“ erscheint, klicken Sie auf die  Taste. Die Kameraaufnahme wird in der App angezeigt und nun können Sie Ihr Zeus Auto mit diesen Widgets steuern.

Die Funktionen der drei Widgets sind wie folgt:

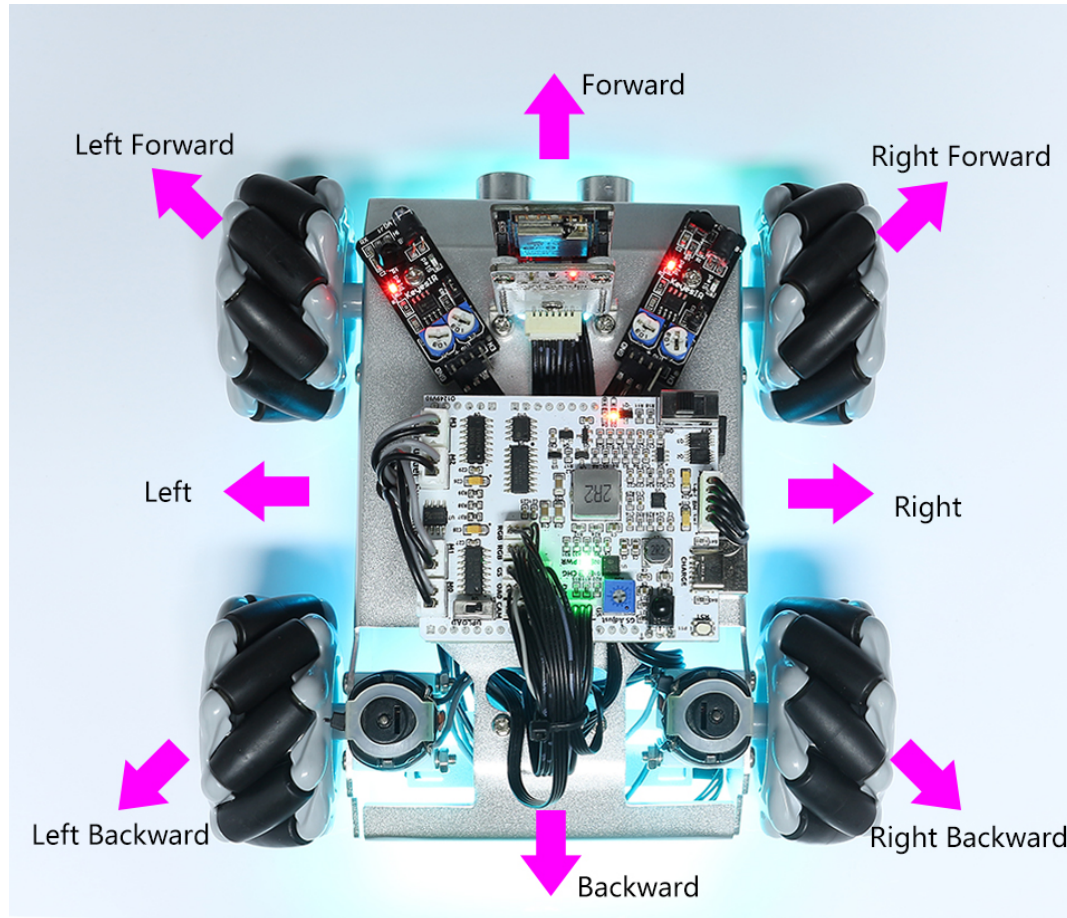
In alle Richtungen bewegen(K)



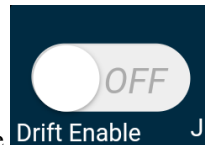
Das Zeus Auto wird sich in die entsprechende Richtung bewegen, wenn Sie das Widget wischen.



Das Auto bewegt sich jedes Mal, wenn Sie wischen. Lassen Sie Ihre Hand nicht los, bewegt sich das Auto weiter.



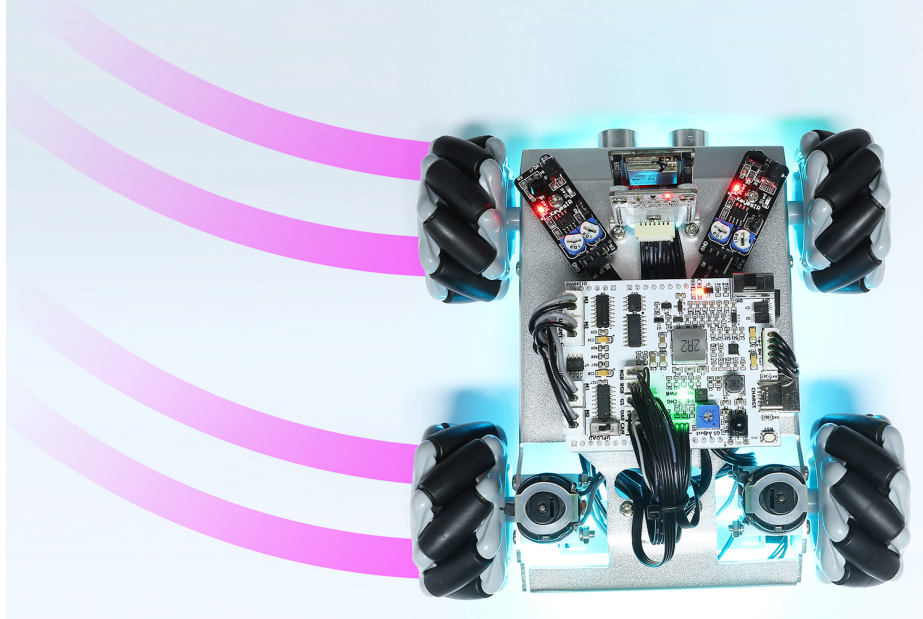
Drift aktivieren(J)



Klicken Sie auf die Drift Enable Taste, um die Drift-Funktion zu aktivieren.

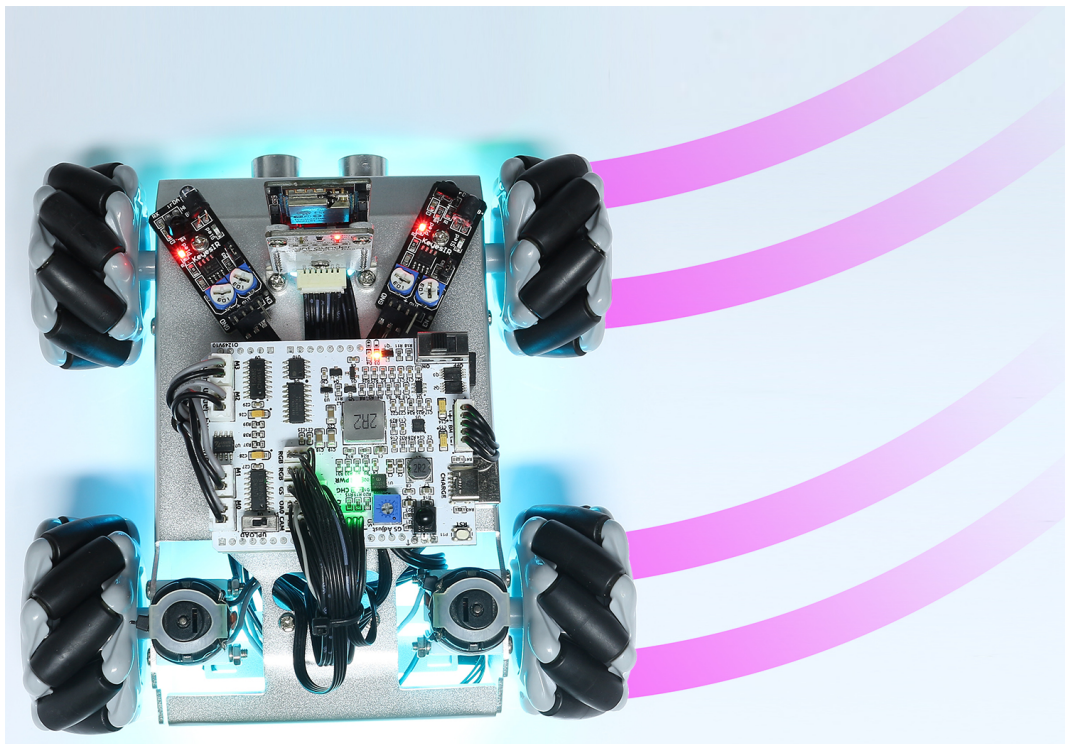


- Wenn Sie das Widget gegen den Uhrzeigersinn schieben, wird das Zeus Auto nach rechts driften. Beim Loslassen stoppt das Auto an seiner aktuellen Position.





Q Widget


- Gleichmaßen wird das Zeus Auto nach links driften, wenn Sie das im Uhrzeigersinn schieben und an der aktuellen Position stoppen.

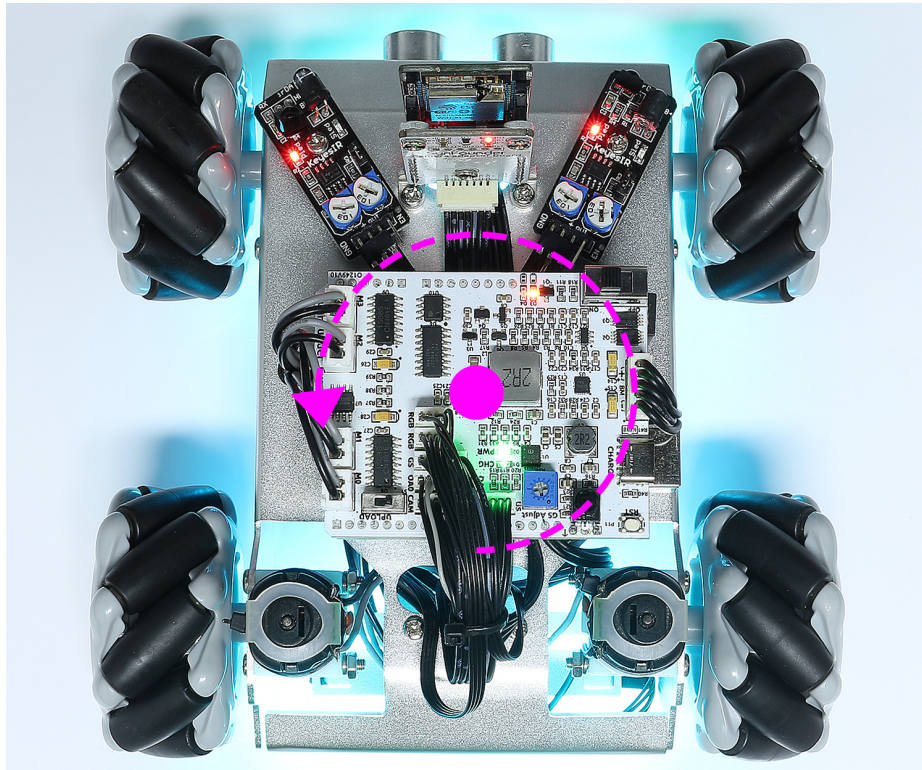


Die Richtung steuern(Q)

- Ist die  Taste aktiviert, wird das  Widget verwendet, um das Zeus Auto links und rechts driften zu lassen.

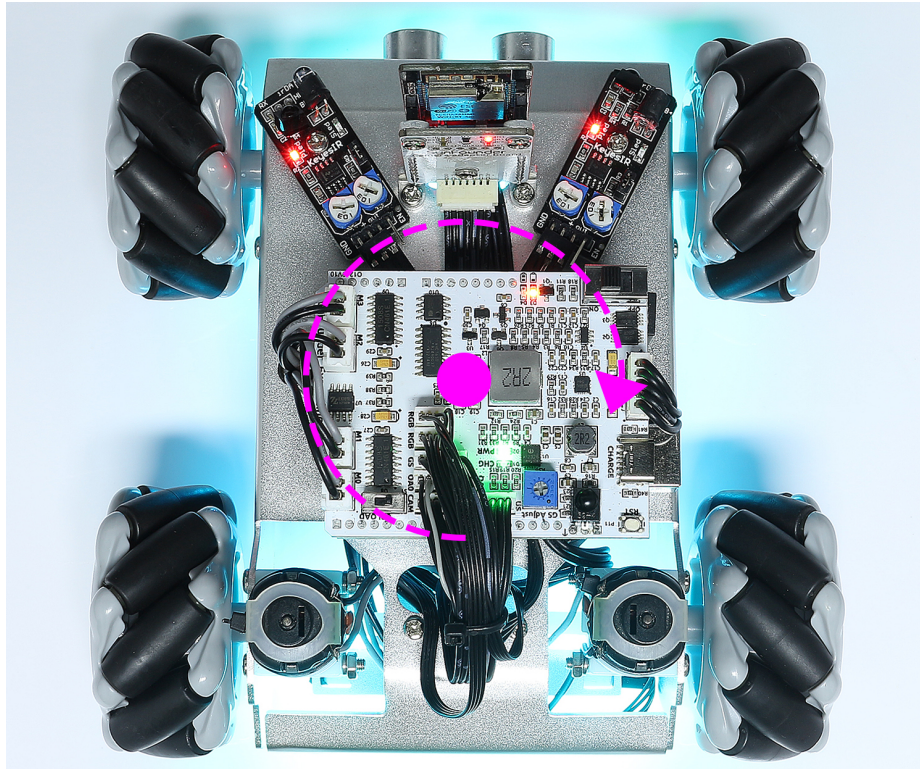
- Ist das  Widget deaktiviert, steuert das  Widget die Richtung des Auto-Kopfes.

- Beim Schieben des  Widgets gegen den Uhrzeigersinn wird das Auto ebenfalls gegen den Uhrzeigersinn drehen. Nach dem Loslassen kehrt der Kopf des Autos in die Ausgangsrichtung zurück.





- Entsprechend dreht sich das Auto im Uhrzeigersinn mit dem Widget und kehrt beim Loslassen in die Ausgangsrichtung zurück.



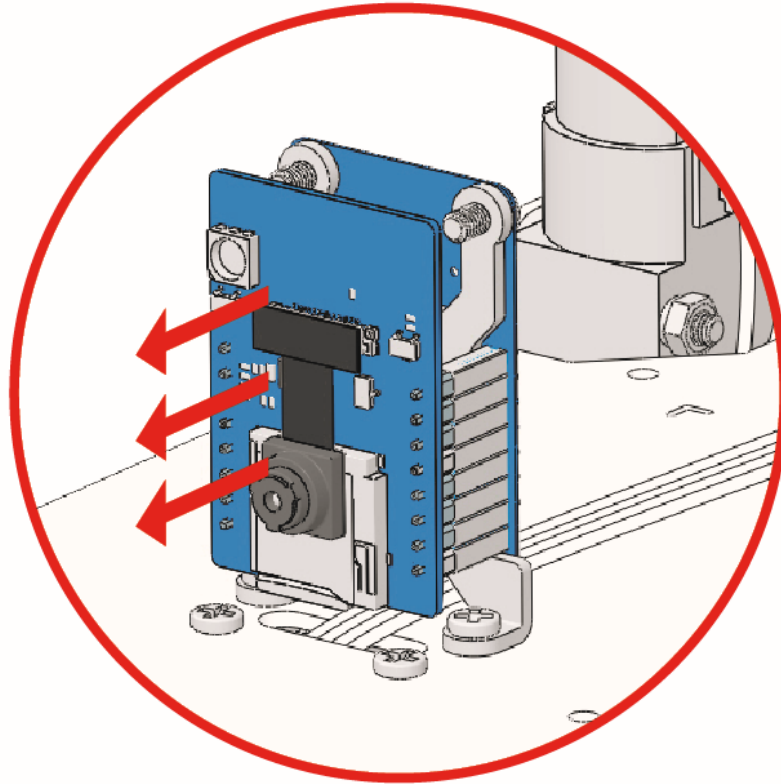
18. APP Control Plus

Dieses Projekt integriert die Funktionen Linienverfolgung, Folgen und Vermeiden basierend auf *17. APP-Steuerung*.

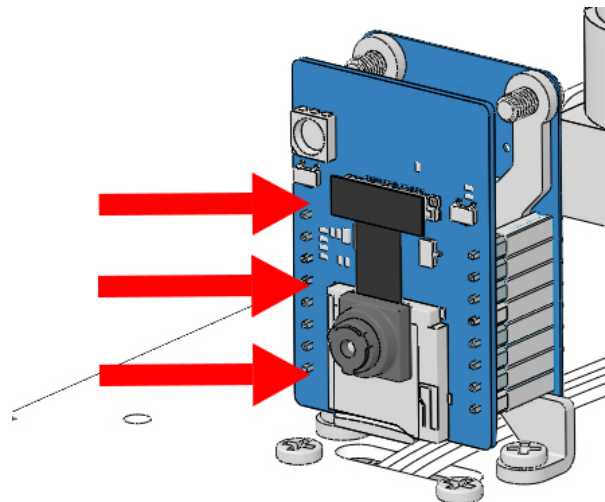
Bemerkung: Bitte installieren Sie den [SunFounder Controller](#) aus dem **APP Store(iOS)** oder **Google Play(Android)**.

Wie funktioniert das?

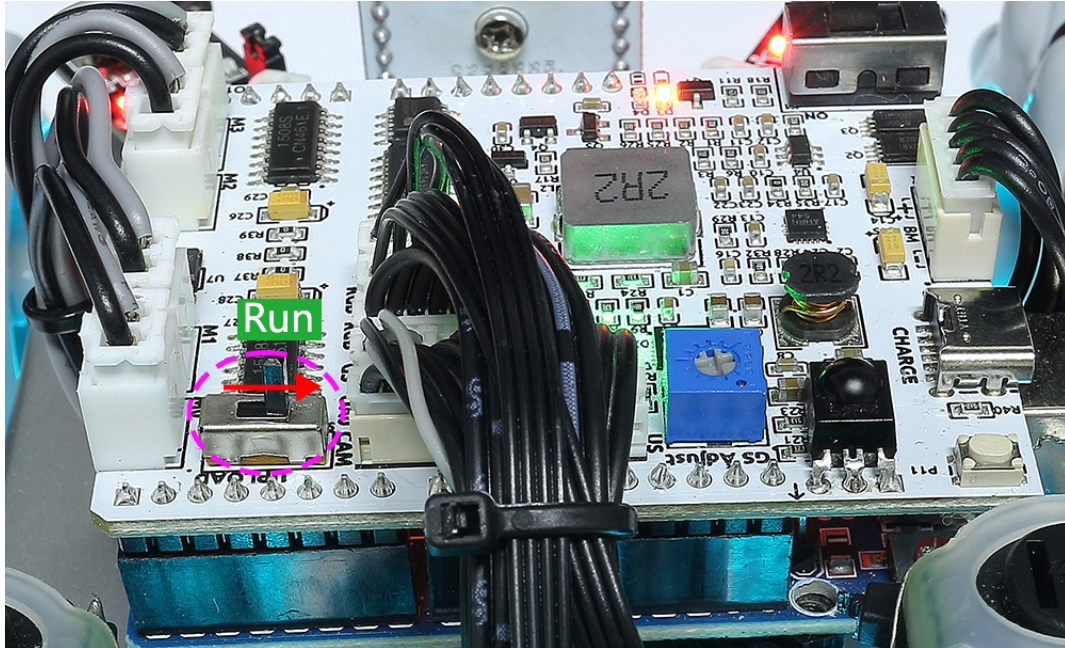
1. Der ESP32-CAM und das Arduino-Board teilen sich die gleichen RX (Empfang) und TX (Senden) Pins. Beim Hochladen des Codes müssen Sie zuerst den ESP32-CAM trennen, um Konflikte oder potenzielle Probleme zu vermeiden.



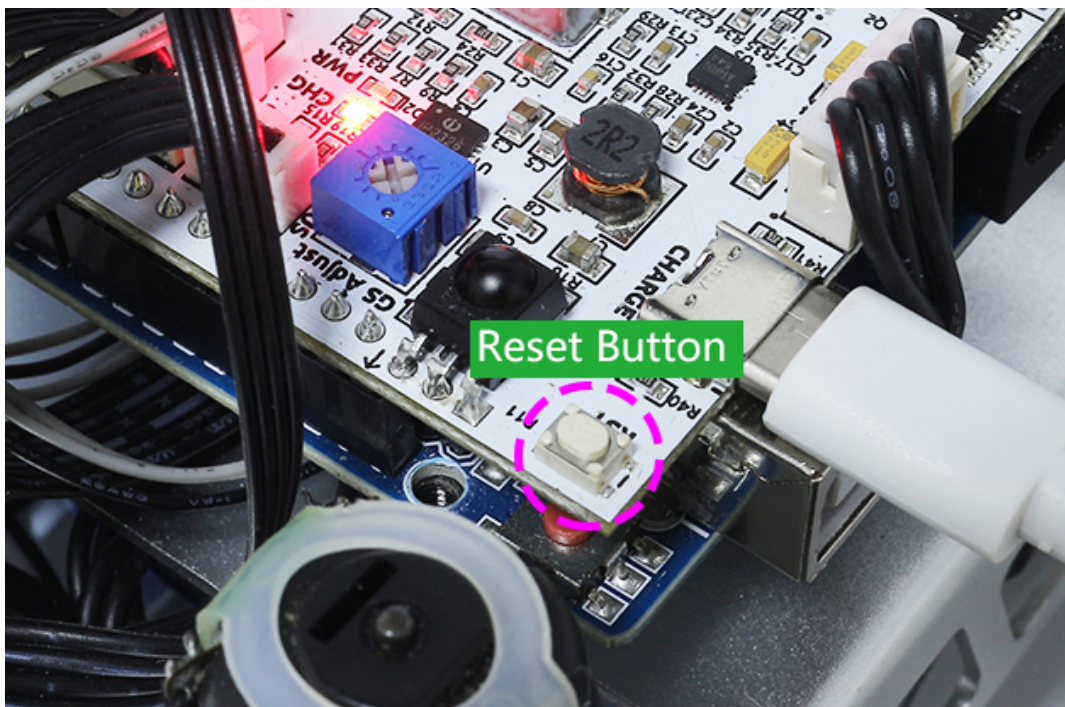
2. Öffnen Sie die Datei `18_app_control_plus.ino` im Pfad `zeus-car-main\examples\18_app_control_plus`.
3. Nachdem der Code erfolgreich hochgeladen wurde, können Sie das ESP32-CAM einstecken und dann den Stromschalter auf EIN schieben, um das Zeus-Auto zu starten.



4. Schalten Sie den Upload-Schalter auf die Seite von „Run“ (rechte Seite in dieser Darstellung), um die ESP32 CAM zu starten.



5. Drücken Sie die Reset-Taste, um das Programm des Arduino-Boards erneut zu starten.

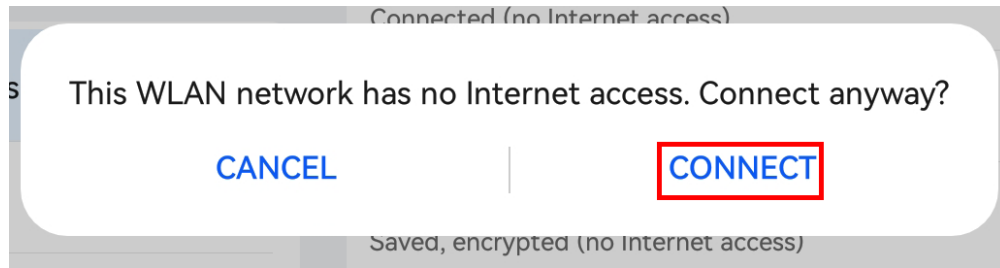


6. Verbinden Sie sich mit dem Zeus_Car WLAN.

Verbinden Sie nun Ihr Mobilgerät mit dem lokalen Netzwerk (LAN), das vom Zeus Car bereitgestellt wird. Auf diese Weise befinden sich Ihr Mobilgerät und der Zeus Car im selben Netzwerk, was die Kommunikation zwischen den Anwendungen auf Ihrem Mobilgerät und dem Zeus Car erleichtert.

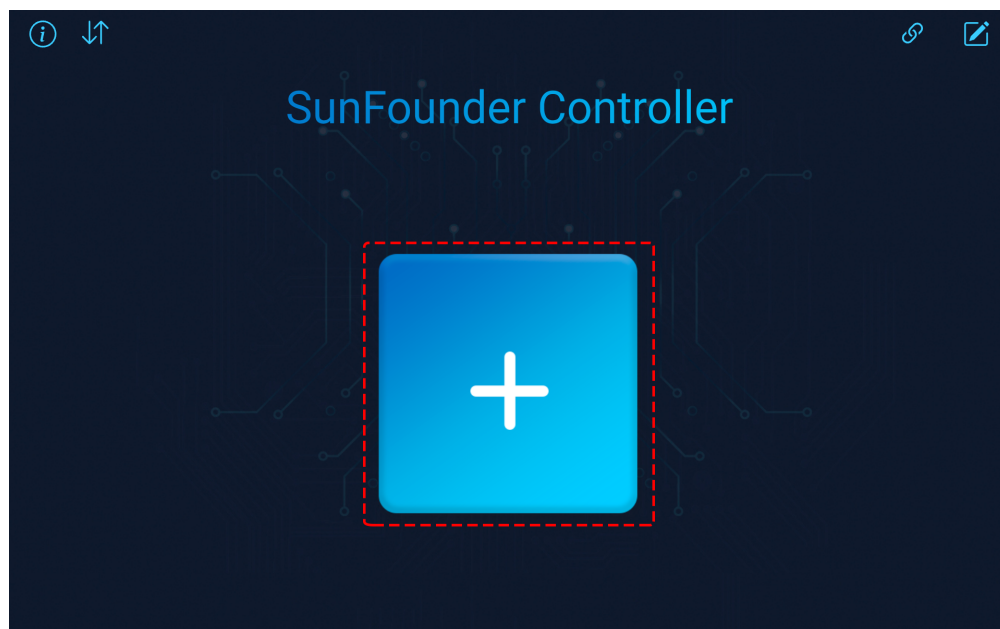
- Finden Sie Zeus_Car im WLAN Ihres Mobiltelefons (Tablet), geben Sie das Passwort 12345678 ein und verbinden Sie sich damit.
- Der Standardverbindungsmodus ist der AP-Modus. Nach der Verbindung erhalten Sie daher ei-

ne Meldung, dass dieses WLAN-Netzwerk keinen Internetzugang hat. Bitte wählen Sie „Weiter verbinden“.

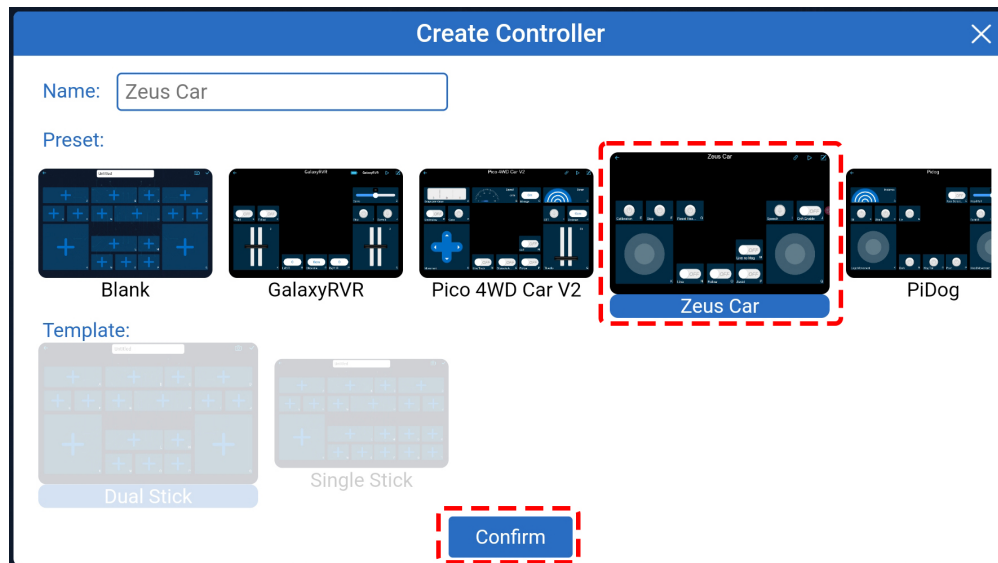


7. Erstellen Sie einen Controller.

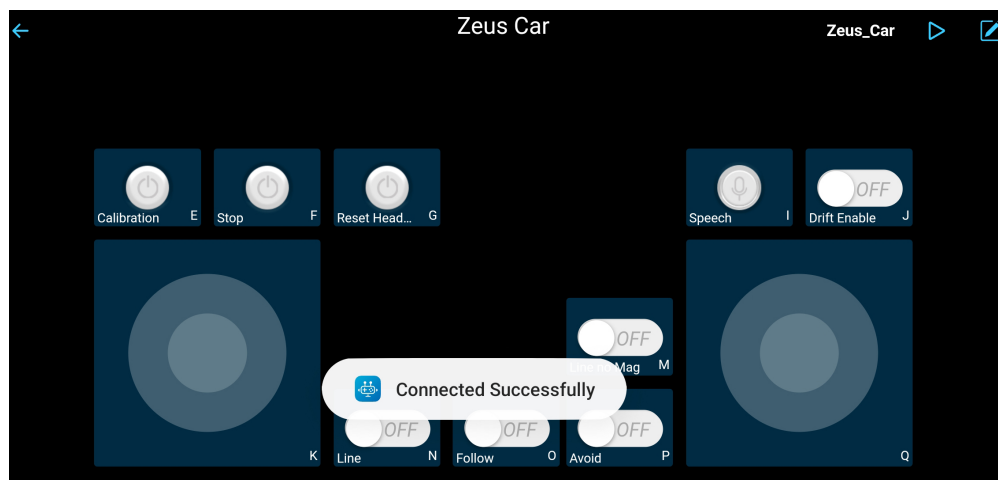
- Um auf dem SunFounder Controller einen Controller hinzuzufügen, klicken Sie auf das + Symbol.




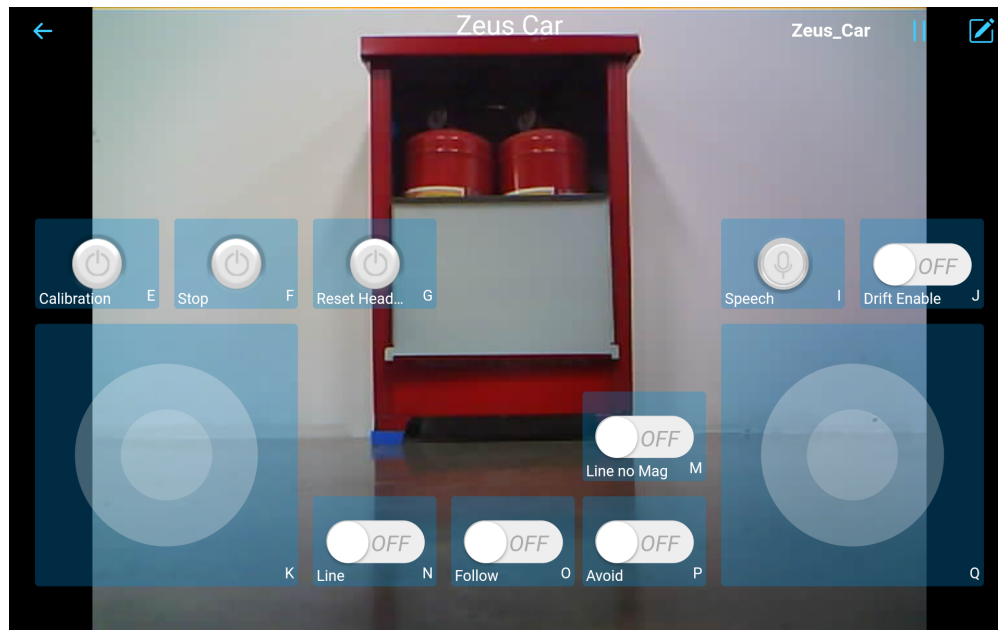
- Voreingestellte Controller sind für einige Produkte verfügbar. Hier wählen wir **Zeus Car**. Geben Sie ihm einen Namen oder tippen Sie einfach auf **Confirm**.



- Einmal drinnen, wird die App automatisch nach dem Zeus Car suchen. Nach einer Weile erscheint eine Meldung, die besagt „Connected Successfully“.



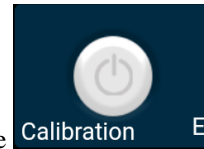
- Tippen Sie nun auf die  Schaltfläche. Dadurch können Sie das Live-Video von der Kamera anzeigen und das Auto mit den bereitgestellten Widgets steuern.



8. Hier sind die Funktionen der Widgets.

- *Calibration(E)*: Kompasskalibrierung einschalten.
- **Stop(F)**: Stoppen Sie alle Bewegungen des Autos.
- **Reset Heading(G)**: Nachdem Sie das Auto manuell in eine Richtung ausgerichtet haben, klicken Sie auf dieses Widget, um diese Richtung als Vorderseite der Autobewegung festzulegen. Damit können Sie schnell eine Richtung angeben, anstatt das Auto langsam mit anderen Widgets in diese Richtung zu drehen.
- *Speech(I)*: Wechseln Sie in den Sprachsteuerungsmodus.
- *Drift Enable(J)*: Aktivieren Sie die Drift-Funktion.
- *Move in All Directions(K)*: Steuern Sie das Auto in alle Richtungen.
- *Line Track*: Die folgenden beiden Widgets können beide in den Linienverfolgungsmodus wechseln.
 - **Linie ohne Mag(M)**: Wechseln Sie in den Linienverfolgungsmodus, der nicht vom Magnetfeld beeinflusst wird. Während des Linienverfolgungsvorgangs wird sich die Ausrichtung des Zeus Car ständig ändern.
 - **Linie(N)**: Wechseln Sie in den Linienverfolgungsmodus. Aufgrund des Vorhandenseins eines Magnetfelds wird sich die Ausrichtung des Zeus Car während der Linienverfolgung in eine bestimmte Richtung orientieren.
- *Follow(O)*: Wechseln in den Verfolgungsmodus.
- *Avoid(P)*: Wechseln in den Hindernisvermeidungsmodus.
- *Control the Direction(Q)*: Dient zur Steuerung der Kopfrichtung.

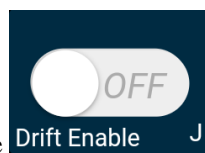
Kalibrierung(E)



Aktivieren Sie die Kompasskalibrierung, indem Sie auf die Schaltfläche Calibration E klicken.

Stellen Sie das Zeus-Auto auf den Boden. Nach dem Aktivieren der Kompasskalibrierung beginnt das Auto, sich gegen den Uhrzeigersinn zu drehen und stoppt nach etwa 1 Minute. Wenn es länger als 2 Minuten dreht, ist das Magnetfeld hier kompliziert. Versuchen Sie, den Standort zu wechseln und erneut zu kalibrieren.

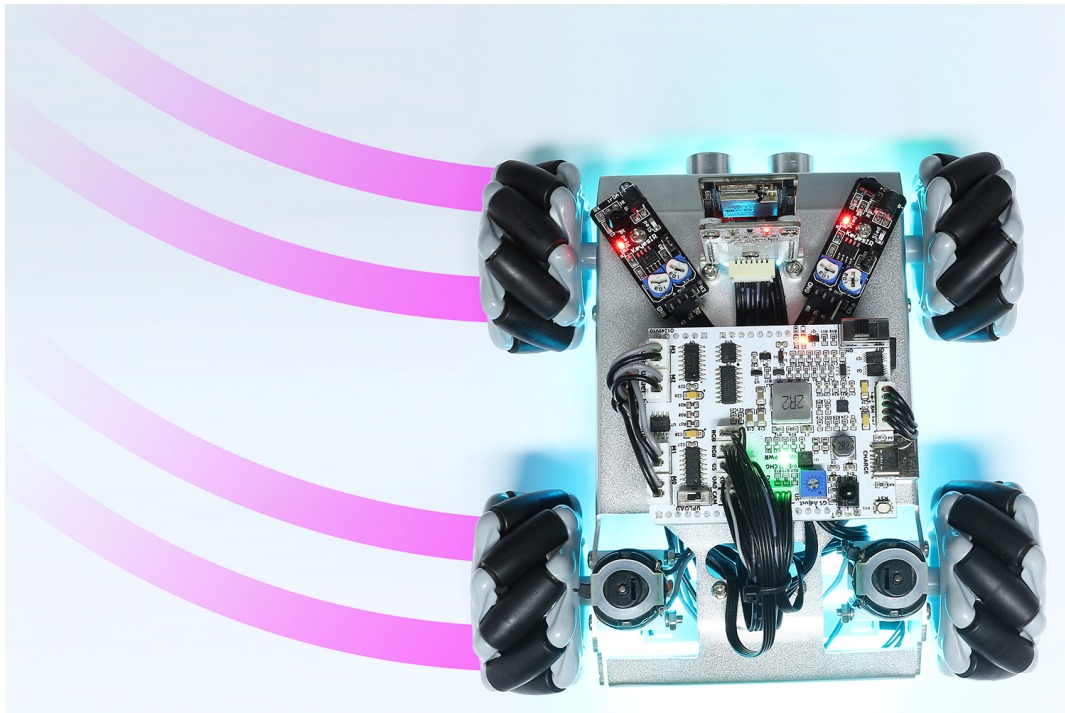
Drift aktivieren(J)

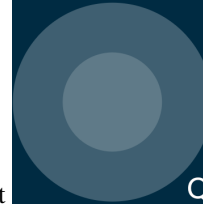


Klicken Sie auf die Schaltfläche Drift Enable J, um die Drift-Funktion zu aktivieren.

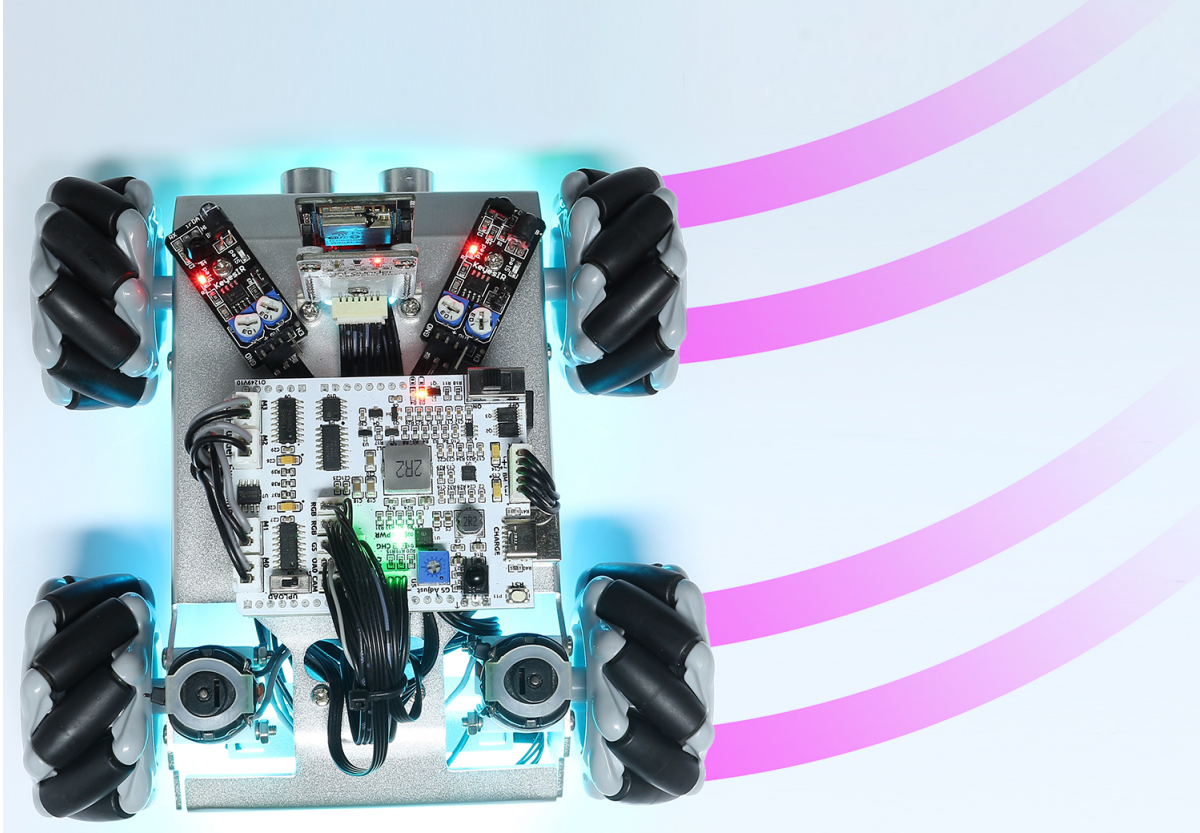


- Wenn Sie das Widget Q gegen den Uhrzeigersinn verschieben, wird das Zeus-Auto nach rechts driften. Beim Loslassen der Hand wird das Auto an seiner aktuellen Position anhalten.





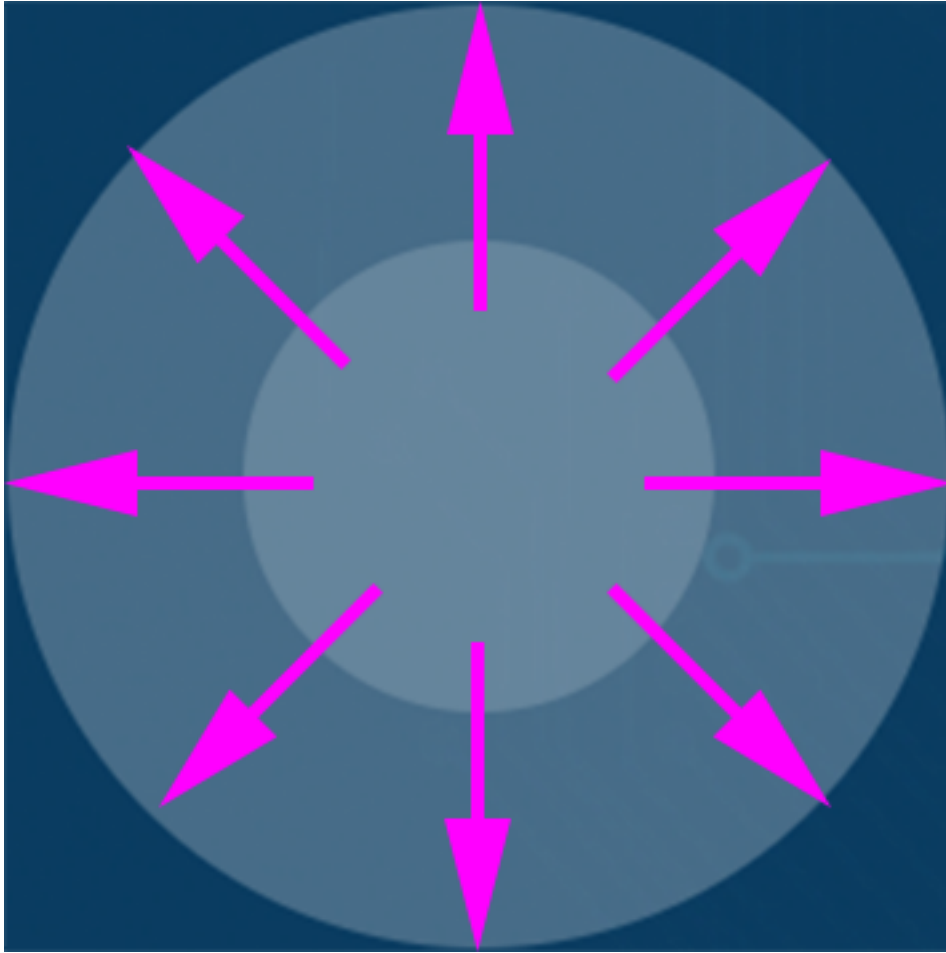
- Ähnlich wird das Zeus-Auto nach links driften, wenn Sie das Widget **Q** im Uhrzeigersinn verschieben und an der aktuellen Position anhalten.



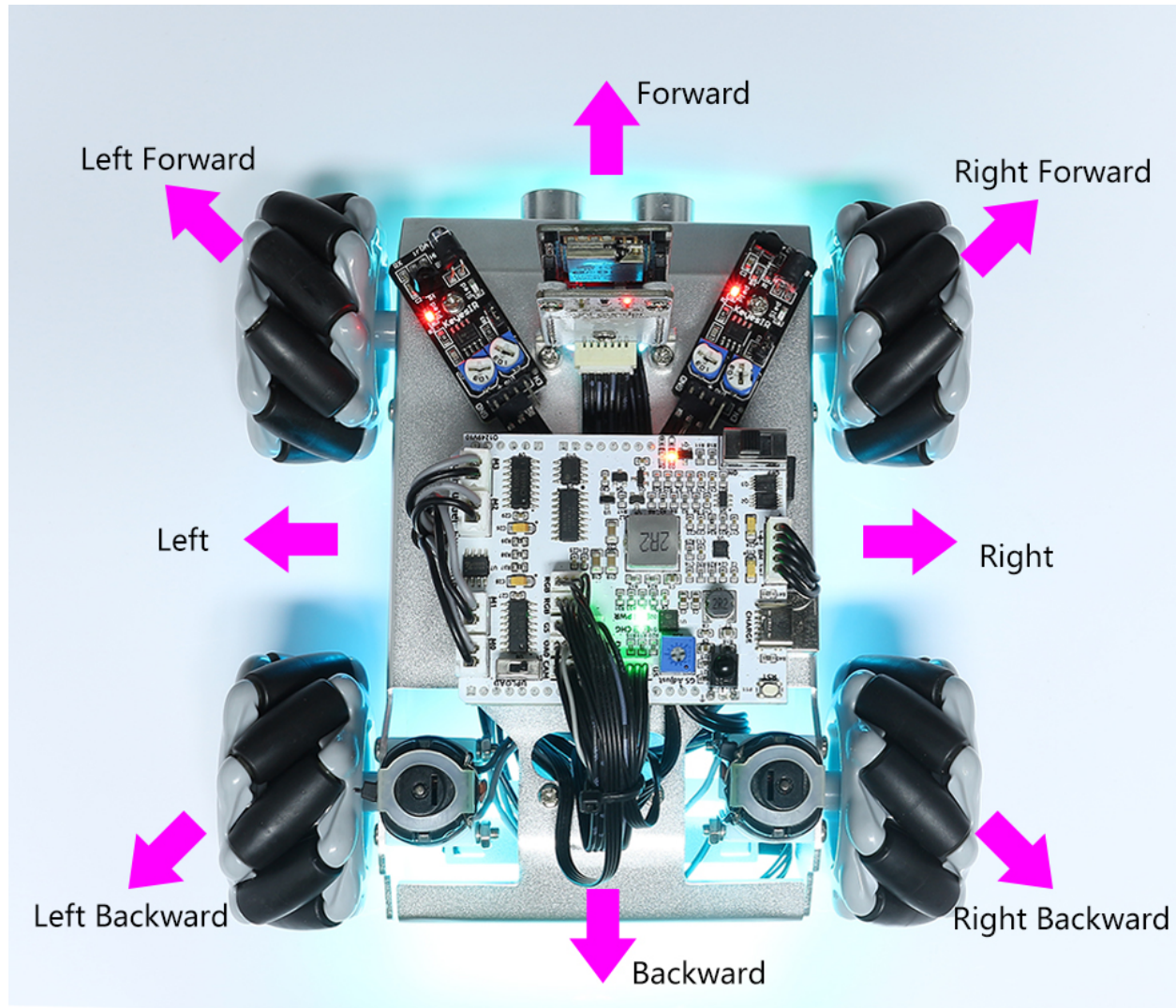
In alle Richtungen bewegen(K)



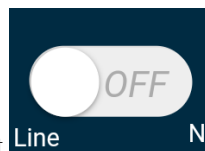
Das Zeus-Auto bewegt sich in die entsprechende Richtung, wenn Sie das Widget **K** wischen.



Das Auto bewegt sich jedes Mal, wenn Sie wischen. Wenn Sie Ihre Hand die ganze Zeit nicht loslassen, bleibt das Auto in Bewegung.



Linie(N)

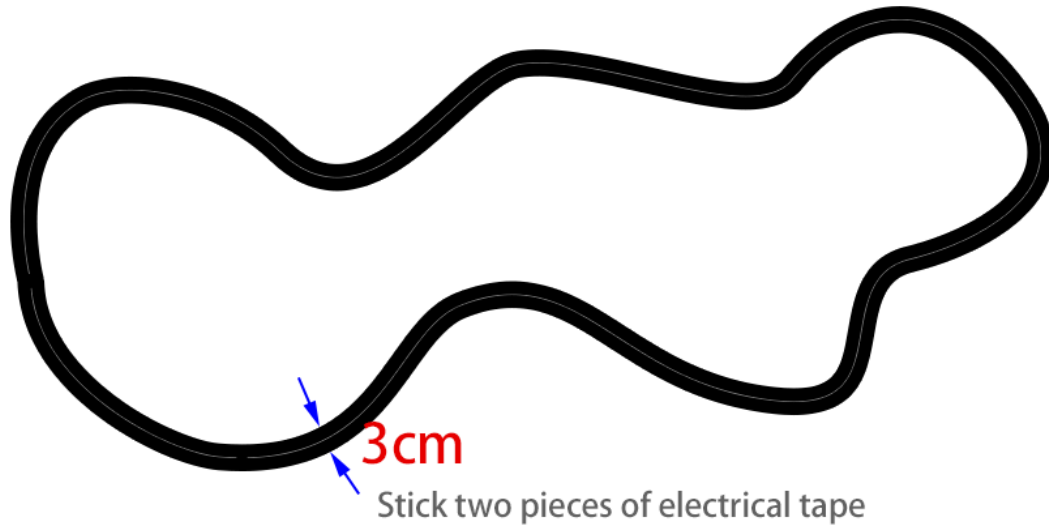


Klicken Sie auf das Widget **Line N**, um in den Linienverfolgungsmodus zu wechseln.

Es stehen zwei Modi der Linienverfolgung im Zeus-Auto zur Verfügung: Ein Modus, bei dem der Kopf stets in Bewegungsrichtung zeigt, und ein Modus, bei dem der Kopf in eine feste Richtung zeigt. Hier wird der zweite Modus ausgewählt.

1. Legen Sie eine 3 cm breite Linie an.

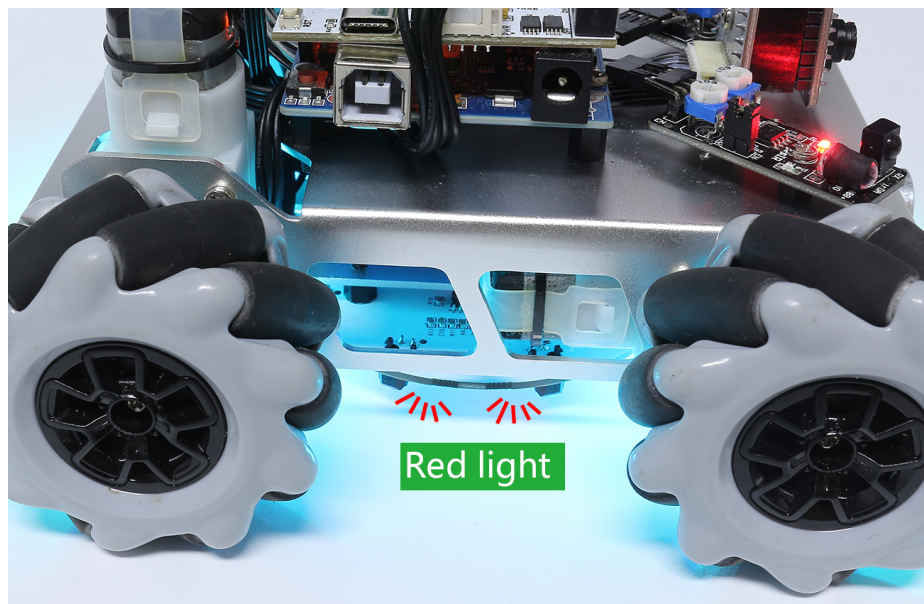
Es gibt acht Sensoren am Omni-Graustufen-Modul und der Abstand zwischen jedem Sensor liegt zwischen 2 und 3 cm. Mindestens zwei Sensoren müssen die schwarze Linie gleichzeitig erkennen. Daher muss die von Ihnen angelegte Linie mindestens 3 cm breit sein und der Biegungswinkel darf nicht kleiner als 90° sein.



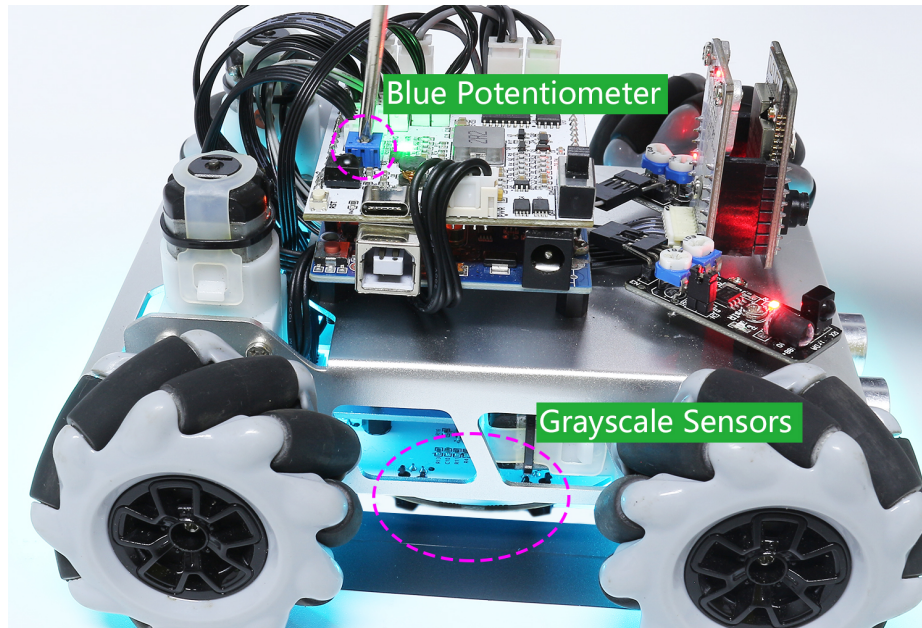
2. Kalibrieren Sie das Omni-Graustufen-Modul.

Da jeder Untergrund unterschiedliche Graustufenwerte hat, ist der werkseitig eingestellte Graustufen-Schwellenwert möglicherweise nicht für Ihre aktuelle Umgebung geeignet. Daher müssen Sie dieses Modul vor der Verwendung kalibrieren. Es wird empfohlen, dies bei jeder wesentlichen Bodenfarbänderung erneut durchzuführen.

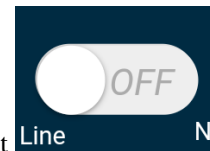
- Stellen Sie das Zeus-Auto auf eine weiße Oberfläche und drehen Sie das Potentiometer so, dass das graue Sensorsignal gerade aufleuchtet.



- Lassen Sie die beiden Graustufensensoren an der Seite genau zwischen der schwarzen Linie und der weißen Oberfläche positionieren und drehen Sie das Potentiometer langsam, bis die Signalanzeige gerade erlischt.

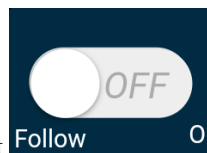


- Sie können mehrfach über die schwarze Linie und die weiße Oberfläche fahren, um sicherzustellen, dass die Lichter des Graustufensensors aus sind, wenn sie sich zwischen der schwarzen Linie und der weißen Oberfläche befinden, und an, wenn sie sich auf der weißen Oberfläche befinden. Dies zeigt an, dass das Modul erfolgreich kalibriert wurde.



- Platzieren Sie das Zeus-Auto auf Ihrer angebrachten Linie, klicken Sie auf das Widget **Line** und es wird der Linie folgen.
- Aufgrund der hohen Umgebungsanforderungen des Omni-Graustufen-Moduls wird empfohlen, es mehrmals zu kalibrieren, wenn der Verfolgungseffekt nicht zufriedenstellend ist (Abweichen von der Linie).

Folgen(O)

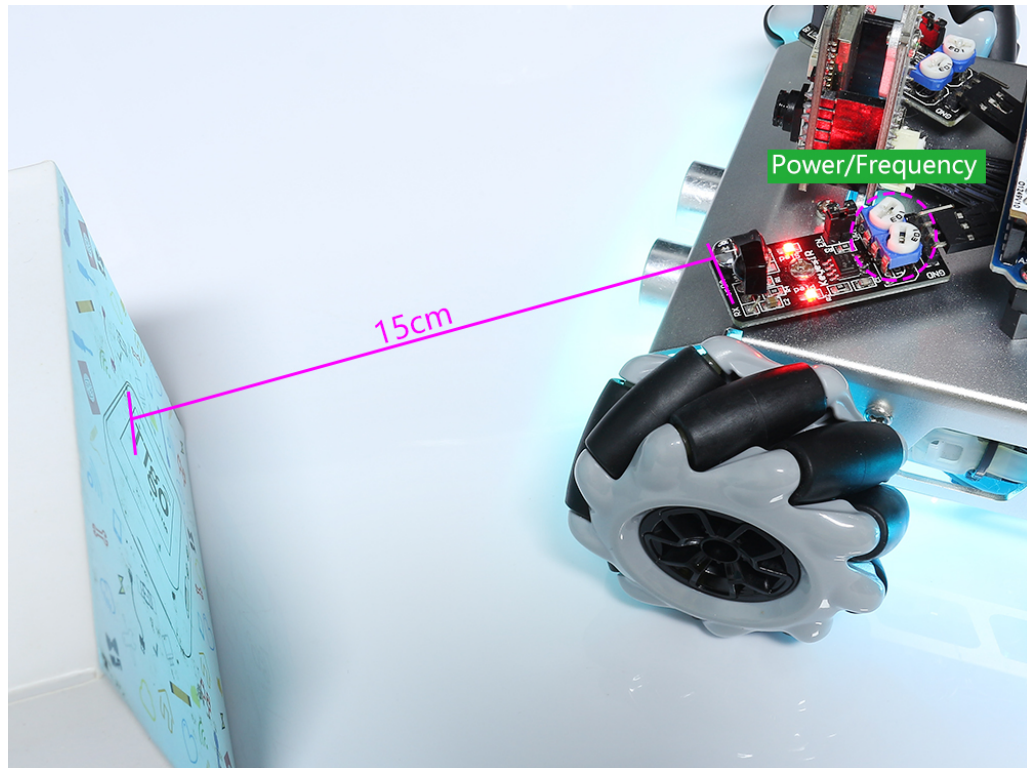


Klicken Sie auf das Widget **Follow**, um in den Folgemodus zu wechseln.

Der Ultraschallsensor erkennt Hindernisse vorne (20 cm) und folgt ihnen. Diese beiden Hindernisvermeidungsmodule ermöglichen es dem Auto, nach links oder rechts zu folgen, müssen jedoch vor der Verwendung kalibriert werden (15cm).

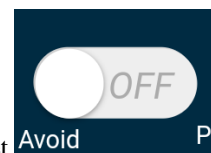
- Kalibrieren Sie das IR-Hindernisvermeidungsmodul.
 - Platzieren Sie ein Hindernis etwa 15 cm vom IR-Hindernisvermeidungsmodul entfernt.
 - Am Modul befinden sich zwei Potentiometer, eines zur Anpassung der Sendeleistung und eines zur Anpassung der Sendefrequenz. Durch Anpassen dieser beiden Potentiometer können Sie den Erfassungsabstand einstellen.

- Dann können Sie ein Potentiometer einstellen und wenn bei 15cm das Signallight am Modul aufleuchtet, ist die Einstellung erfolgreich; wenn nicht, stellen Sie das andere Potentiometer ein.



- Kalibrieren Sie das andere Hindernisvermeidungsmodul auf die gleiche Weise.
2. Stellen Sie das Zeus-Auto auf einen Tisch oder den Boden und lassen Sie es Ihrer Hand oder anderen Hindernissen folgen.

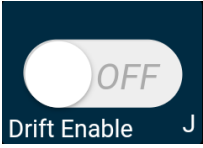

Vermeiden(p)

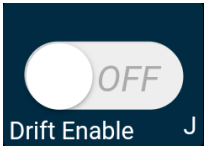




Wenn Sie in den Hindernisvermeidungsmodus wechseln möchten, klicken Sie auf das Widget **Avoid P**. Beachten Sie jedoch zuerst die Referenz *Follow(O)*, um die beiden Hindernisvermeidungsmodule zu kalibrieren.

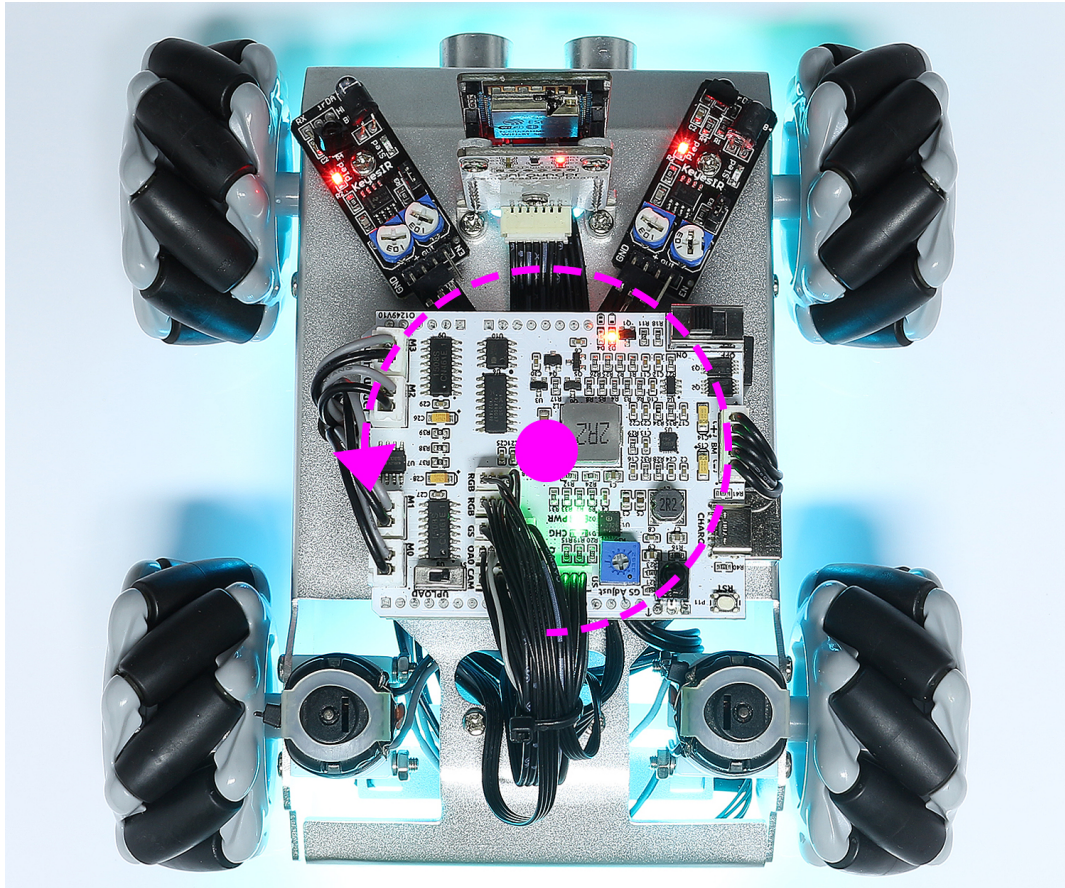
- Das Zeus-Auto fährt vorwärts.
- Ein Ultraschallmodul erkennt Hindernisse vorne. Wenn ein Hindernis erkannt wird, dreht sich das Auto nach links.
- Wenn das linke Hindernisvermeidungsmodul ein Hindernis erkennt, dreht sich das Auto nach rechts und wenn das rechte Hindernisvermeidungsmodul ein Hindernis erkennt, dreht sich das Auto nach links.

Die Richtung steuern(Q)

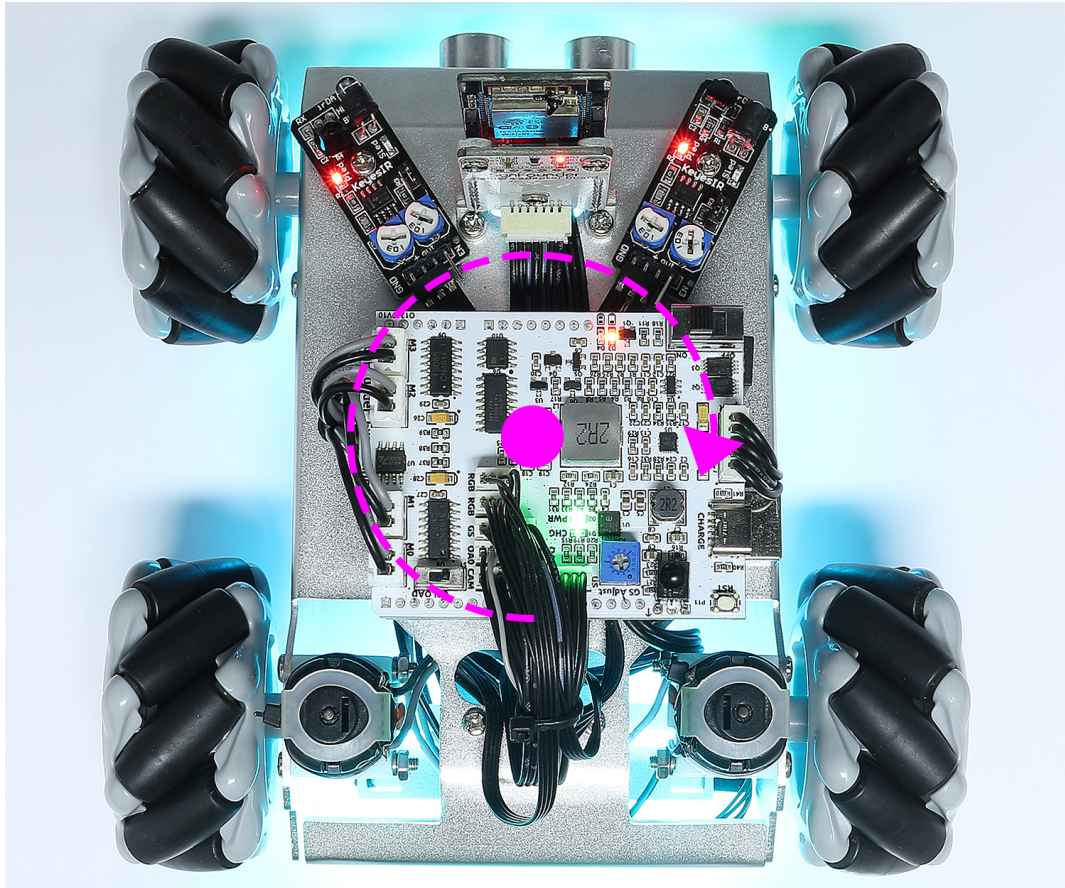
- Wenn die Schaltfläche  aktiviert ist, wird das Widget  verwendet, um das Zeus-Auto links und rechts driften zu lassen.

- Wenn das Widget  deaktiviert ist, wird das Widget  verwendet, um die Richtung des Auto-Kopfes zu steuern.

- Wenn Sie das Widget  gegen den Uhrzeigersinn verschieben, wird das Auto ebenfalls gegen den Uhrzeigersinn drehen. Wenn Sie die Hand loslassen, kehrt der Kopf des Autos in die ursprüngliche Richtung zurück.



- Ebenso wird das Auto mit dem Widget  im Uhrzeigersinn drehen und bei Loslassen in die ursprüngliche Richtung zurückkehren.



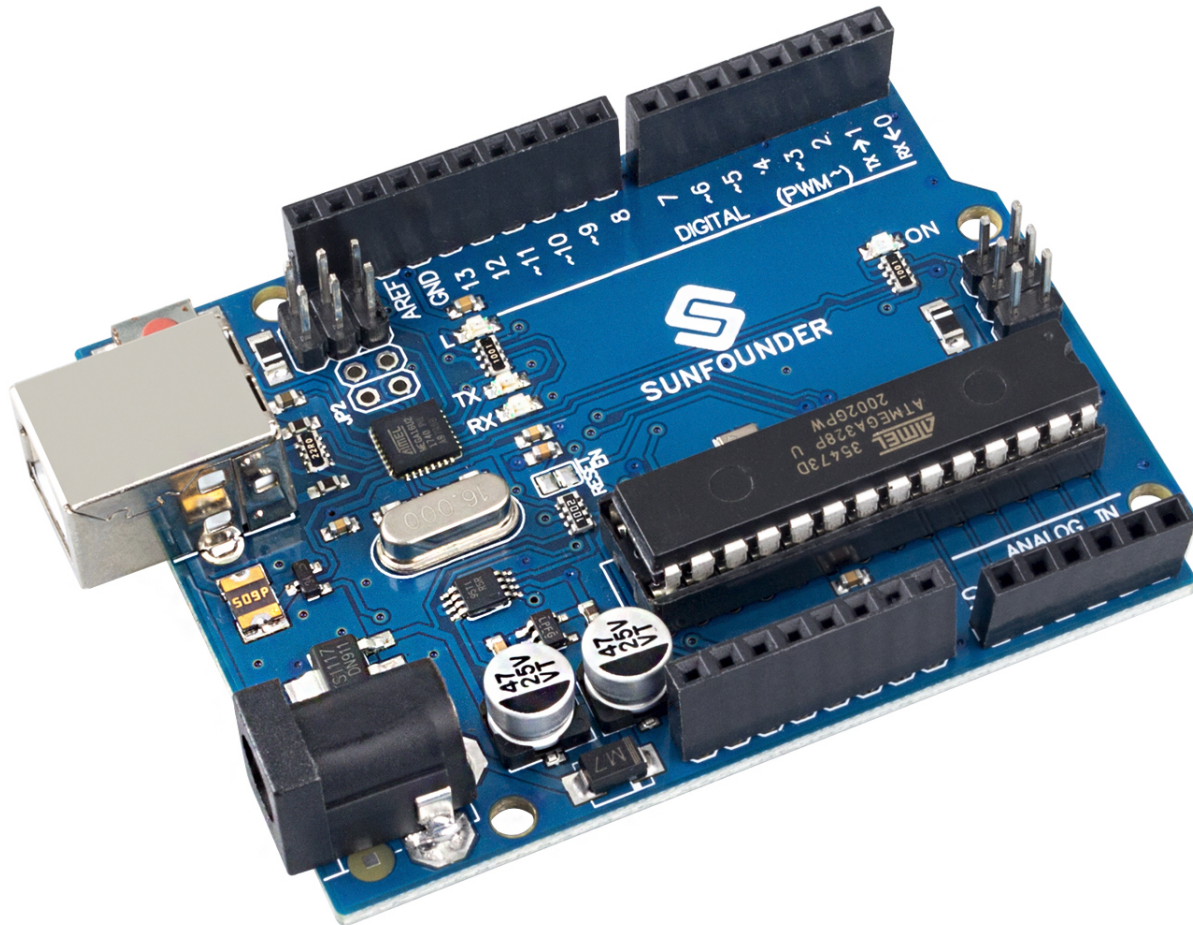
KAPITEL 2

Hardware

Wenn Sie Code schreiben, müssen Sie vielleicht wissen, wie jedes Modul funktioniert oder welche Rolle jeder Pin spielt. Bitte konsultieren Sie dazu dieses Kapitel.

In diesem Kapitel finden Sie eine Beschreibung der Funktion jedes Moduls, technische Parameter und Arbeitsprinzipien.

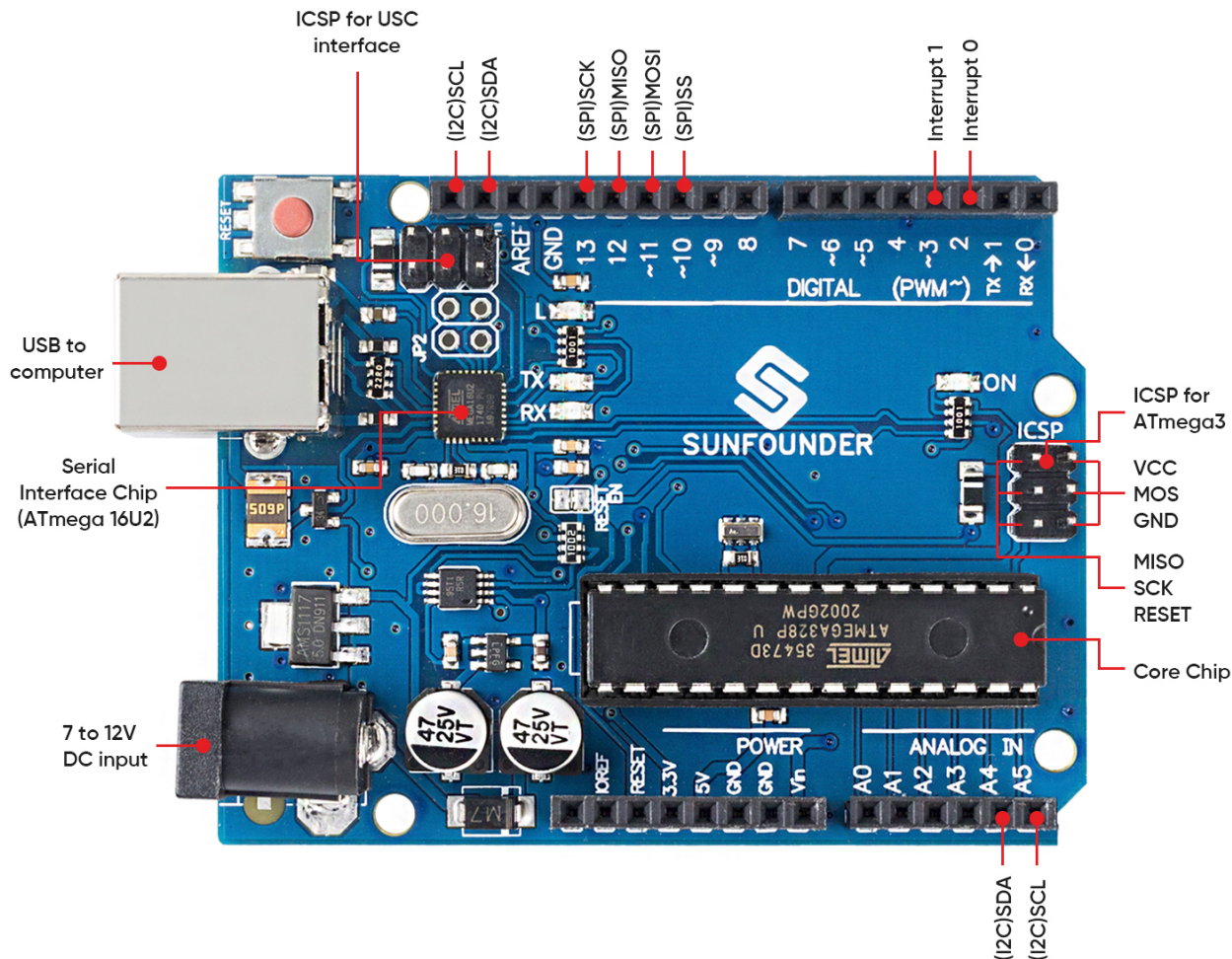
2.1 SunFounder R3 Board



Bemerkung: Das SunFounder R3 Board ist ein Hauptboard, das nahezu die gleichen Funktionen wie das [Arduino Uno](#) bietet. Beide Boards können abwechselnd verwendet werden.

Das SunFounder R3 Board ist ein Mikrocontroller-Board, das auf dem ATmega328P basiert ([Datenblatt](#)). Es verfügt über 14 digitale Ein-/Ausgangspins (von denen 6 als PWM-Ausgänge verwendet werden können), 6 analoge Eingänge, einen 16 MHz Keramikresonator (CSTCE16M0V53-R0), eine USB-Verbindung, eine Strombuchse, einen ICSP-Header und einen Reset-Knopf. Es enthält alles, was zur Unterstützung des Mikrocontrollers benötigt wird; einfach per USB-Kabel an einen Computer anschließen oder mit einem AC-DC-Adapter oder einer Batterie versorgen, um zu starten.

Technische Daten



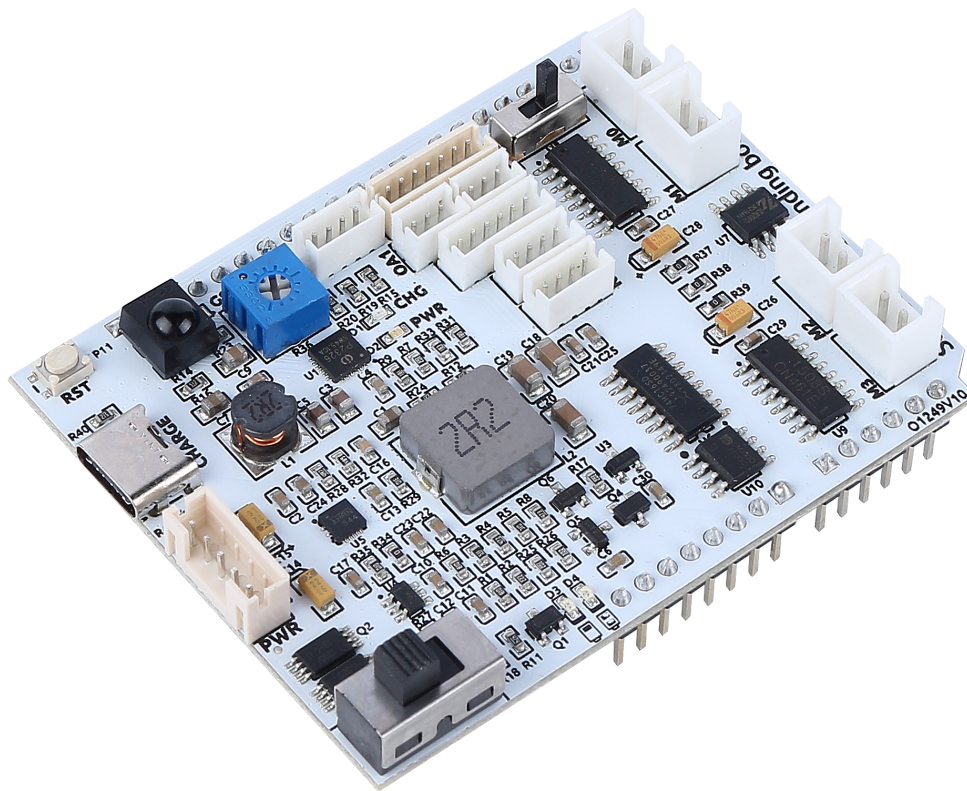
- MIKROCONTROLLER: ATmega328P
- BETRIEBSSPANNUNG: 5V
- EINGANGSSPANNUNG (EMPFOHLEN): 7-12V
- EINGANGSSPANNUNG (GRENZE): 6-20V
- DIGITALE I/O-PINS: 14 (0-13, davon 6 mit PWM-Ausgang(3, 5, 6, 9-11))
- PWM DIGITALE I/O-PINS: 6 (3, 5, 6, 9-11)
- ANALOGE EINGANGSPINS: 6 (A0-A5)
- GLEICHSTROM PRO I/O-PIN: 20 mA
- GLEICHSTROM FÜR 3.3V PIN: 50 mA
- FLASH-SPEICHER: 32 KB (ATmega328P), wovon 0,5 KB vom Bootloader verwendet werden
- SRAM: 2 KB (ATmega328P)
- EEPROM: 1 KB (ATmega328P)
- TAKTFREQUENZ: 16 MHz
- LED_BUILTIN: 13
- LÄNGE: 68,6 mm

- BREITE: 53,4 mm
- GEWICHT: 25 g
- I2C-Anschluss: A4(SDA), A5(SCL)

Weiterführende Informationen

- [Arduino IDE](#)
- [Arduino Programmiersprachen-Referenz](#)
- [Download und Installation von Arduino IDE 2.0](#)
- [ATmega328P Datenblatt](#)

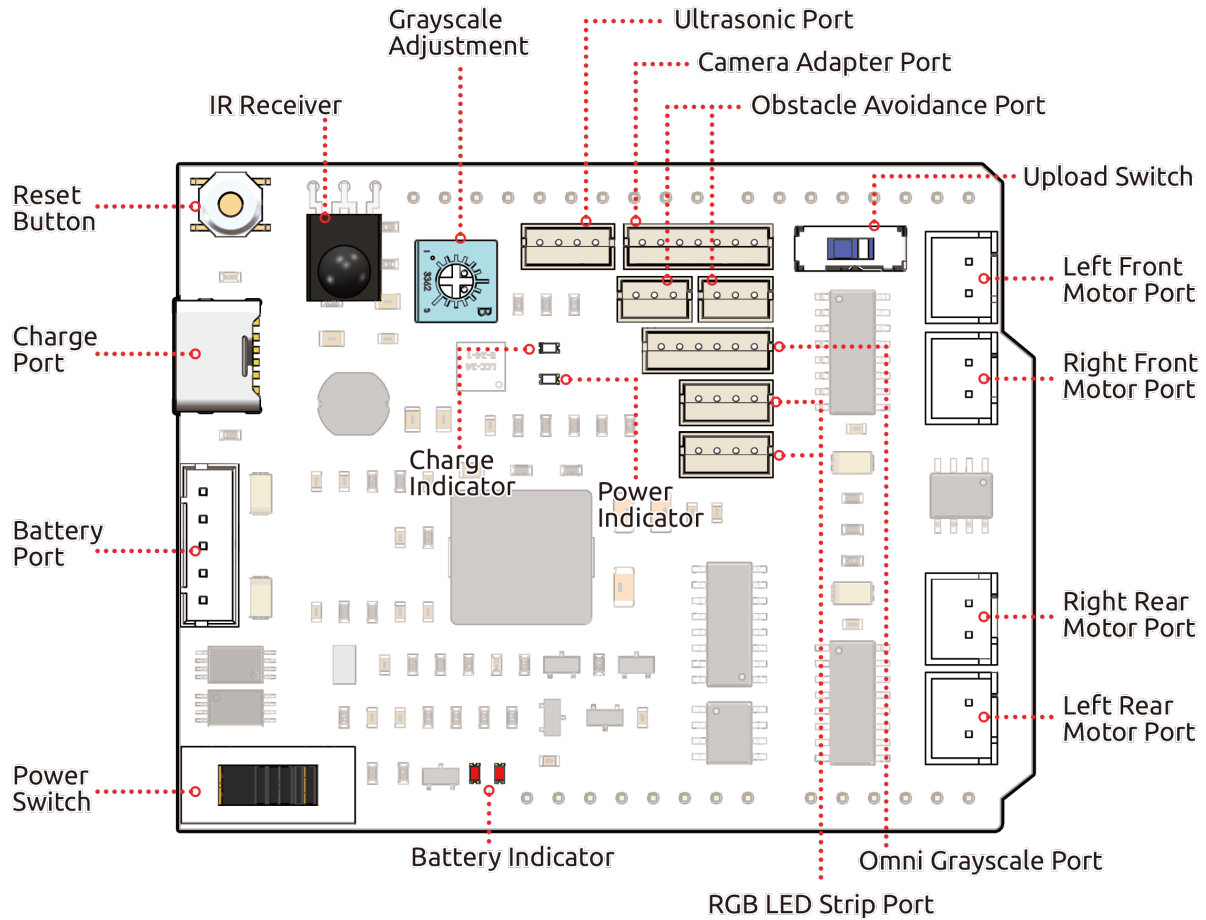
2.2 Zeus Car Shield



Hierbei handelt es sich um eine All-in-One-Erweiterungsplatine, die von SunFounder für Arduino entworfen wurde. Sie enthält verschiedene Modulanschlüsse wie Motor, Lichtleiste, Hindernisvermeidung, Graustufen, ESP32 CAM und Ultraschallmodul. Eingebaut ist auch ein HS0038B IR-Empfänger für die Fernbedienung.

Diese Erweiterungsplatine verfügt auch über einen eingebauten Ladekreis, der den Akku mit PH2.0-5P-Schnittstelle laden kann, und die geschätzte Ladezeit beträgt 130 Minuten.

Zeus Car Shield Anschlüsse

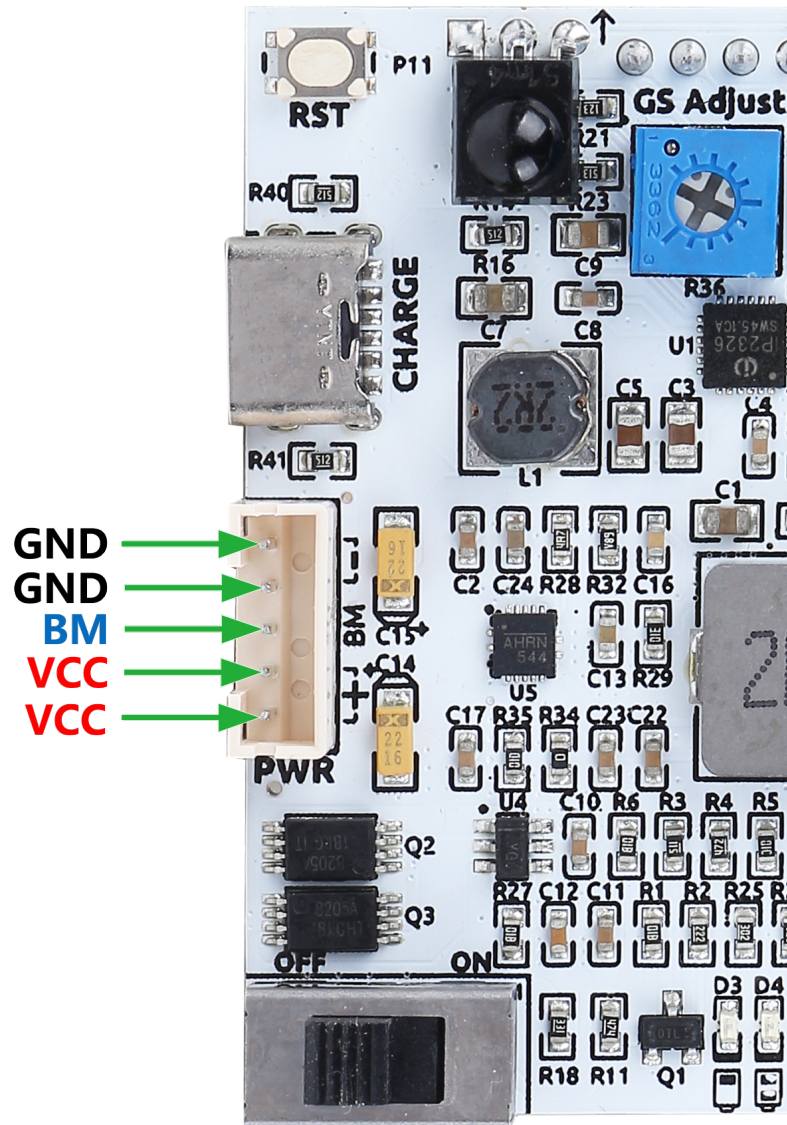


- **Reset-Taste**
 - Drücken Sie diese Taste, um das Programm auf dem Arduino-Board zurückzusetzen.
- **Ladeanschluss**
 - Nachdem Sie in den 5V/2A USB-C-Anschluss eingesteckt haben, kann er verwendet werden, um den Akku in 130 Minuten aufzuladen.
- **Batterieanschluss**
 - 6,6V~8,4V PH2.0-5P Stromversorgung.
 - Versorgt das Zeus Car Shield und das Arduino-Board gleichzeitig mit Strom.
- **Ein-/Ausschalter**
 - Auf ON schieben, um das Zeus Car Shield einzuschalten.
- **IR-Empfänger**
 - Dies ist ein HS0038B IR-Empfänger, dessen Signalpin mit Pin 2 des Arduino-Boards verbunden ist.
- **Zum Graustufenmodul**
 - Graustufen-Justierpotentiometer: Wird verwendet, um die Referenzspannung für das Omni-Graustufen-Modul einzustellen.
 - Graustufenanschluss: Zum Anschluss des Omni-Graustufen-Moduls.
- **Anzeigen**

- **Ladeanzeige:** Leuchtet rot, wenn das Shield über den USB-C-Anschluss geladen wird.
- **Stromanzeige:** Leuchtet grün, wenn der Ein-/Ausschalter auf „ON“ steht.
- **Batterieanzeige:** Zwei orangefarbene Anzeigen repräsentieren unterschiedliche Batteriestände. Sie blinken während des Ladens und schalten sich aus, wenn die Batterie aufgeladen werden muss.
- **Ultraschallanschluss**
 - Zum Anschluss des Ultraschallmoduls sind beide Pins, Trig & Echo, mit Pin 10 des Arduino-Boards verbunden.
- **Kamera-Adapteranschluss**
 - Der Anschluss für das Kamera-Adapter-Board.
- **Anschluss zur Hindernisvermeidung**
 - Wird verwendet, um zwei IR-Hindernisvermeidungsmodule anzuschließen, die Signale der beiden Anschlüsse sind mit Q0 und Q1 des 74HC165 verbunden.
- **RGB LED Streifenanschluss**
 - Zum Anschluss von 2 RGB-LED-Streifen sind die drei Pins des Streifens entsprechend mit 12, 13 und 11 verbunden.
- **Laufschalter**
 - Wenn Sie die Kamera verwenden möchten, müssen Sie diesen Schalter umlegen, damit der ESP32-CAM mit dem Arduino-Board kommunizieren kann.
- **Motoranschluss**
 - 4 Gruppen von Motoranschlüssen.

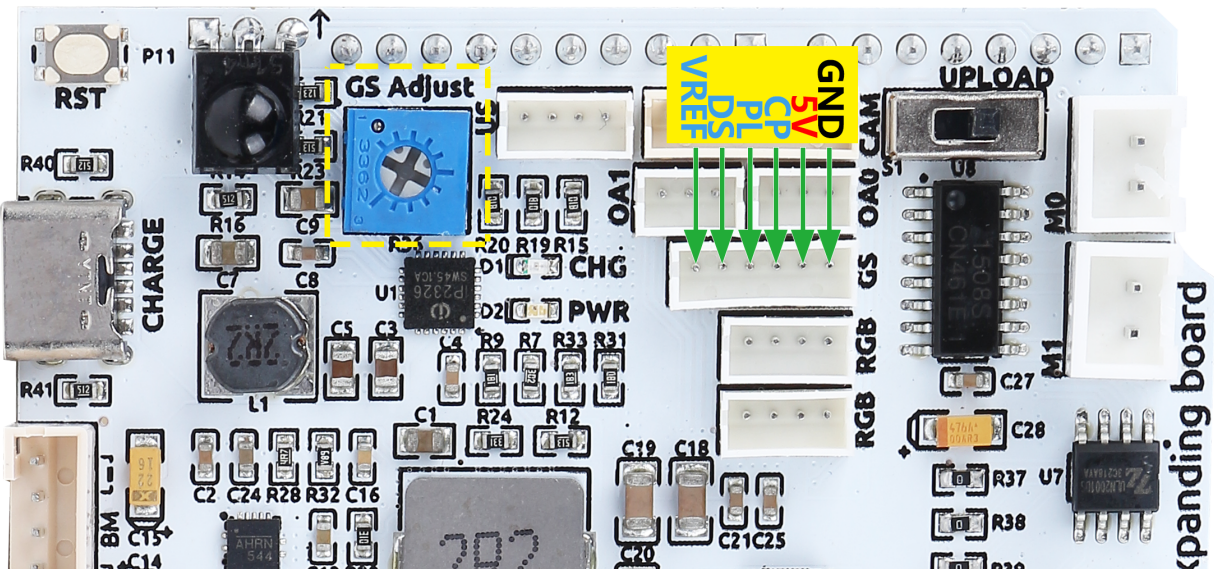
2.2.1 Batterieanschluss

Hier sehen Sie das Pinout-Diagramm für die Batterieschnittstelle. Der Typ ist PH2.0-5P und der Eingangsspannungsbereich liegt zwischen 6.6V und 8.4V.

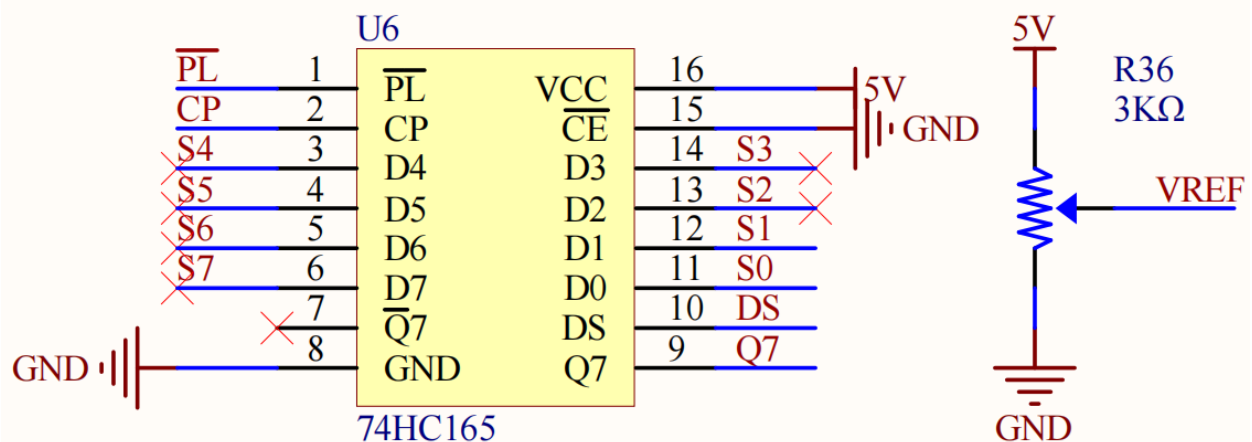


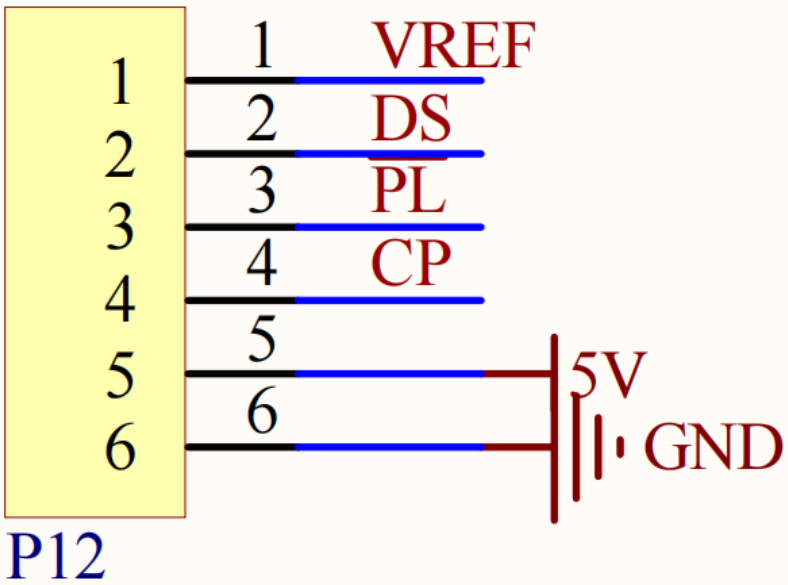
2.2.2 Zum Graustufenmodul

Ein blauer Potentiometer auf dem Zeus Car Shield dient zur Anpassung der Empfindlichkeit des Graustufenmoduls an verschiedene Umgebungen. Dies geschieht durch Einstellen der Referenzspannung für das Modul. Über den VREF-Pin erhält das Graustufenmodul die eingestellte Referenzspannung.



Hier ist das Schaltbild. Die Werte des Graustufenmoduls werden von dem 74HC165-Chip auf das Arduino-Board übertragen. Da das Graustufenmodul selbst einen 74HC165-Chip besitzt, wird durch die Kaskadierung dieser beiden Chips eine 16-Bit-Datenübertragung zum Arduino-Board ermöglicht. Die ersten 8 Bit sind Daten des Graustufensensors und die letzten zwei Bit sind Daten für die IR-Hindernisvermeidung.



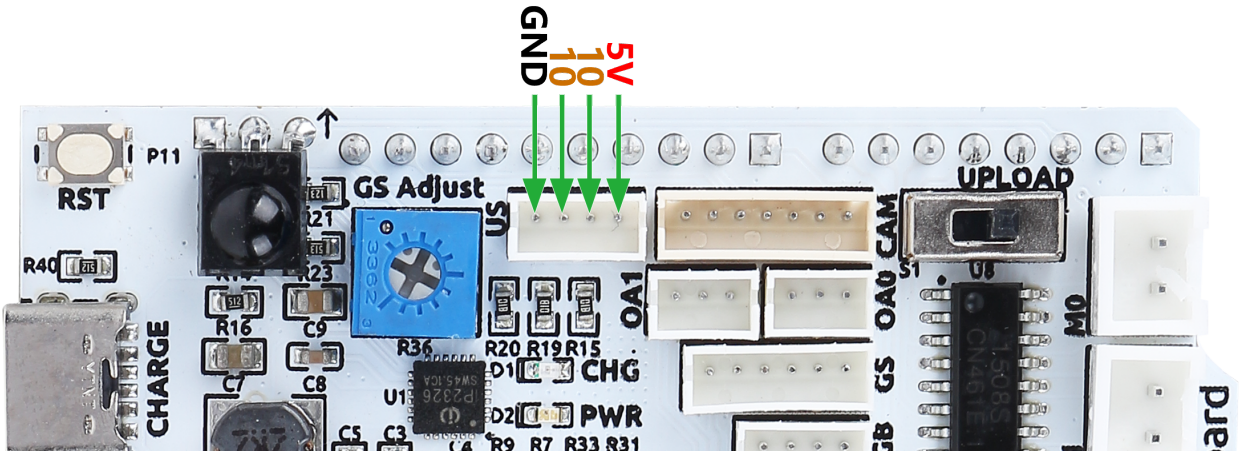


Die Pin-Zuordnung dieses Ports auf dem Arduino-Board ist unten dargestellt.

Arduino Board	Zeus Car Shield
7	74HC165 Q7
8	74HC165 CP
~9	74HC165 PL

2.2.3 Ultraschallanschluss

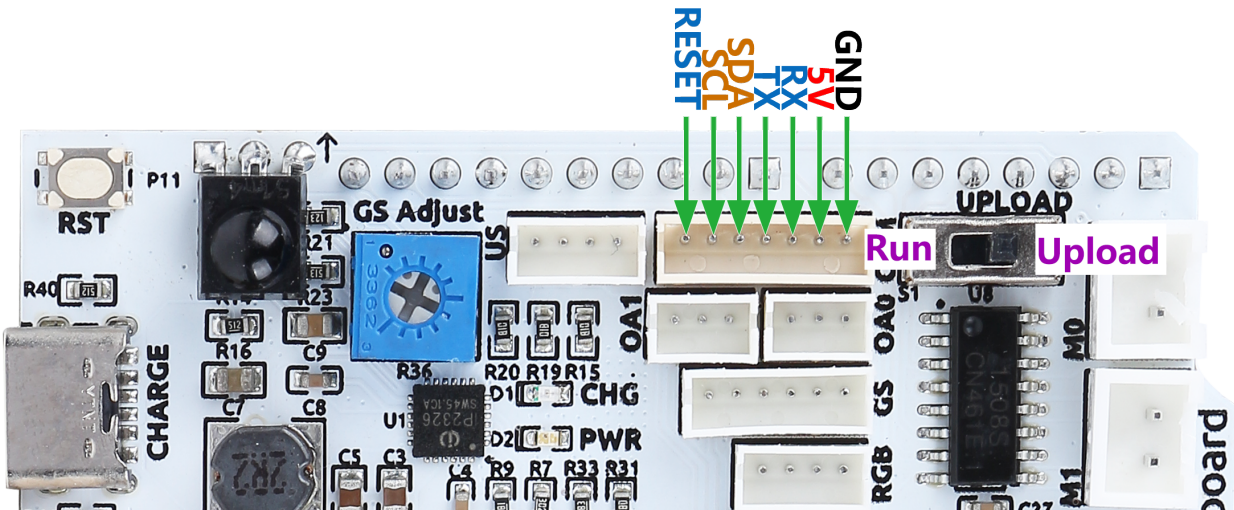
Hier ist das Pinout für den ZH1.5-4P Ultraschallanschluss. Die Trig & Echo Pins sind mit Pin 10 auf dem Arduino Board verbunden.



2.2.4 Kamera-Adapteranschluss

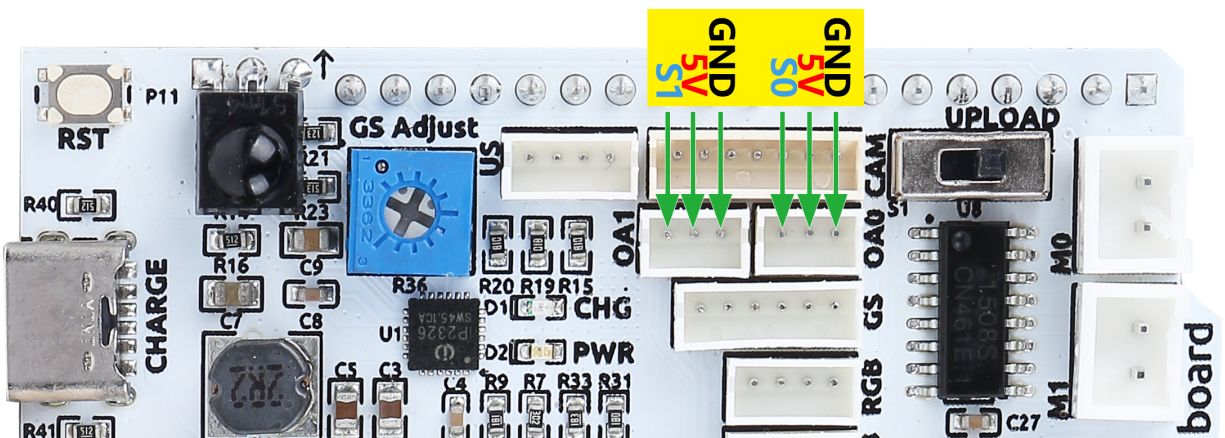
Hier sehen Sie das Pin-Diagramm für die Kamera-Adapter-Schnittstelle. Der Typ ist ZH1.5-7P.

- TX und RX werden für ESP32 CAM verwendet.
- SDA und SCL sind für QMC6310.

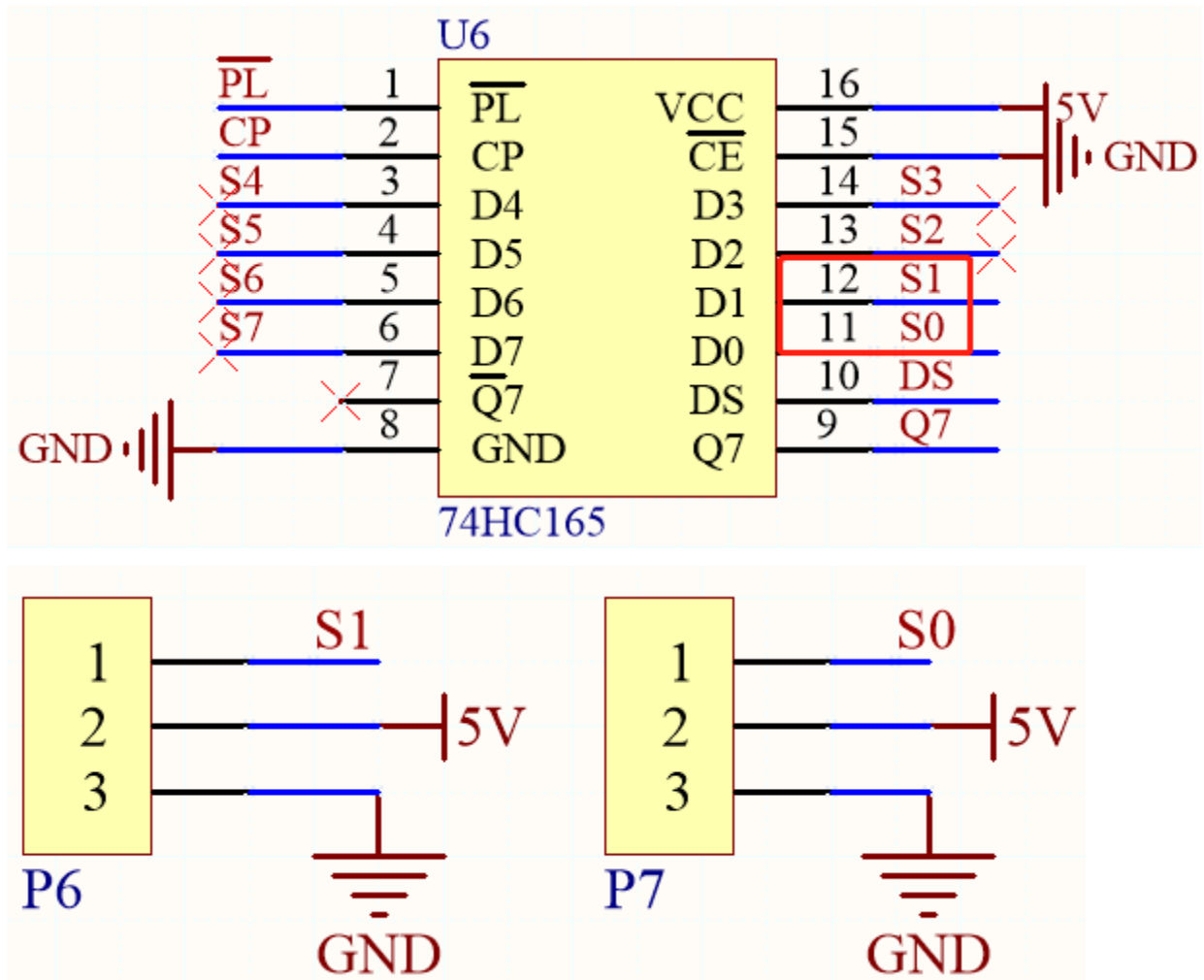


2.2.5 Anschluss zur Hindernisvermeidung

Hier ist das Pinout-Diagramm für die zwei ZH1.5-3P Hindernisvermeidungsanschlüsse.

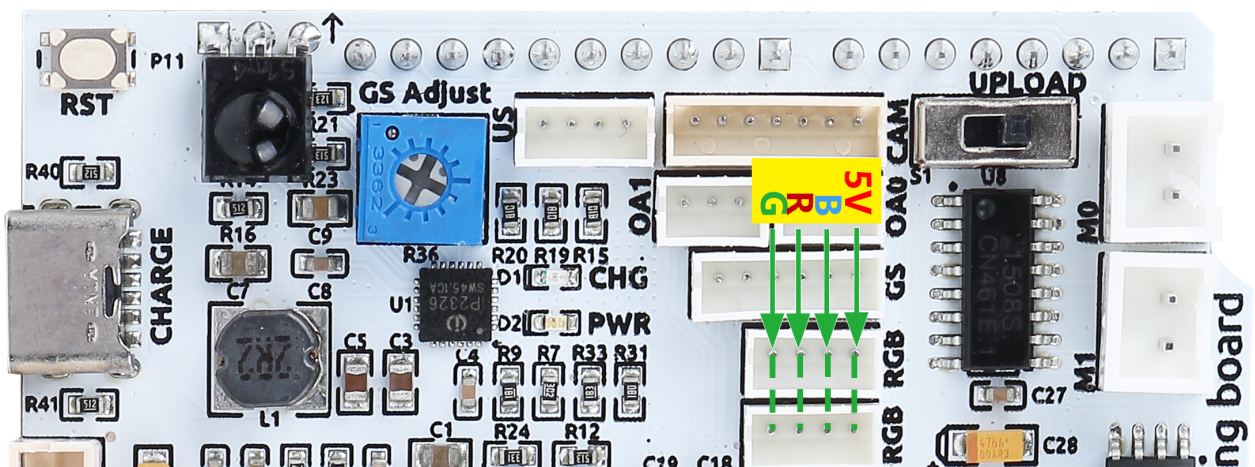


Wie im Schaltplan gezeigt, beziehen sich S1 und S0 auf Q0 und Q1 auf dem 74HC165 Chip. Zwei 74HC165 Kaskaden übertragen die Daten der beiden IR-Hindernisvermeidungsmodule und des Graustufensensors auf das Arduino Board. Die ersten acht Bits sind Daten des Graustufensensors und die letzten zwei Bits stellen Daten für die IR-Hindernisvermeidung dar.

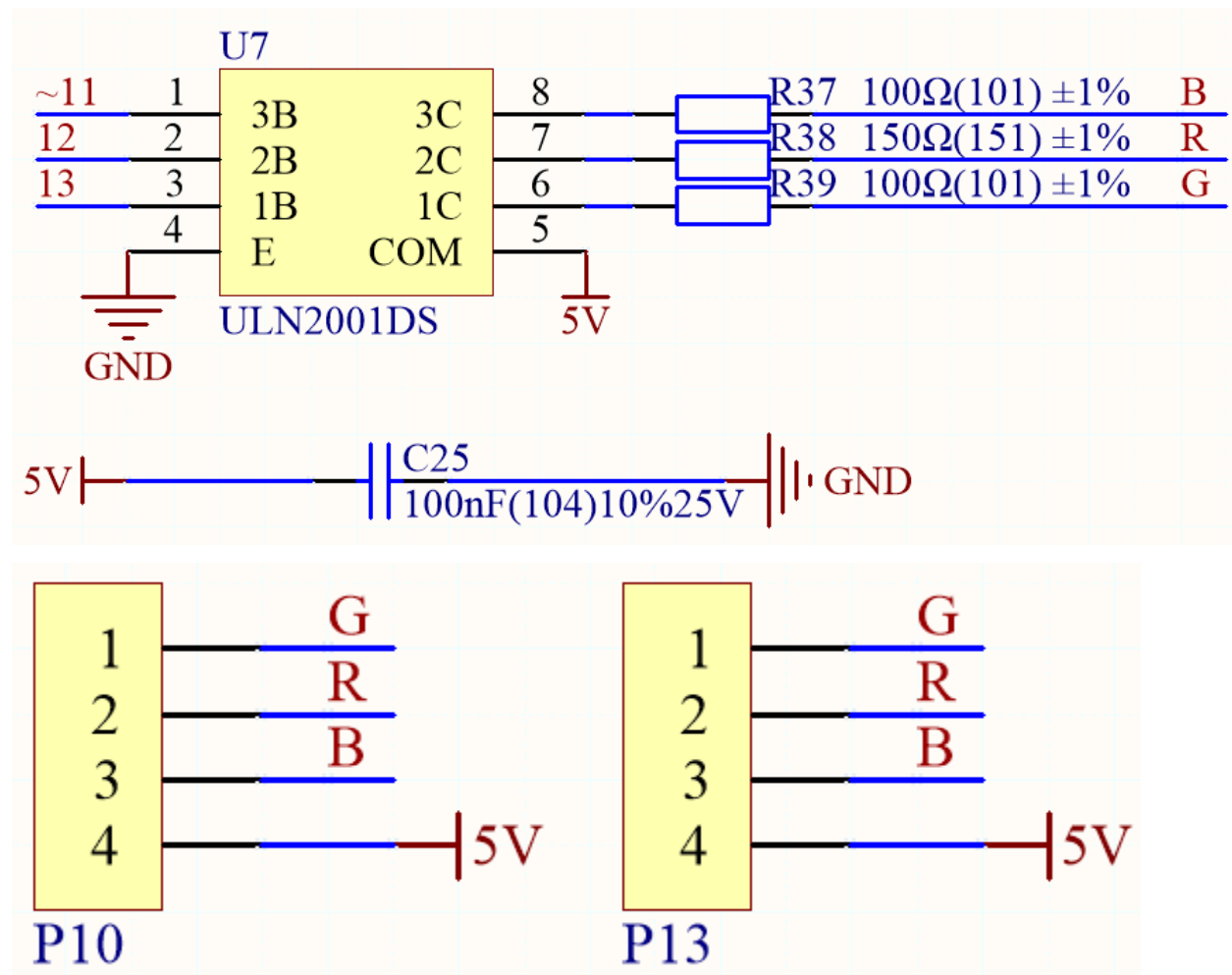


2.2.6 RGB LED Streifenanschluss

Hier ist das Pinout-Diagramm der zwei RGB LED Streifen. Sie sind parallel geschaltet und die Pinbelegungen sind identisch.

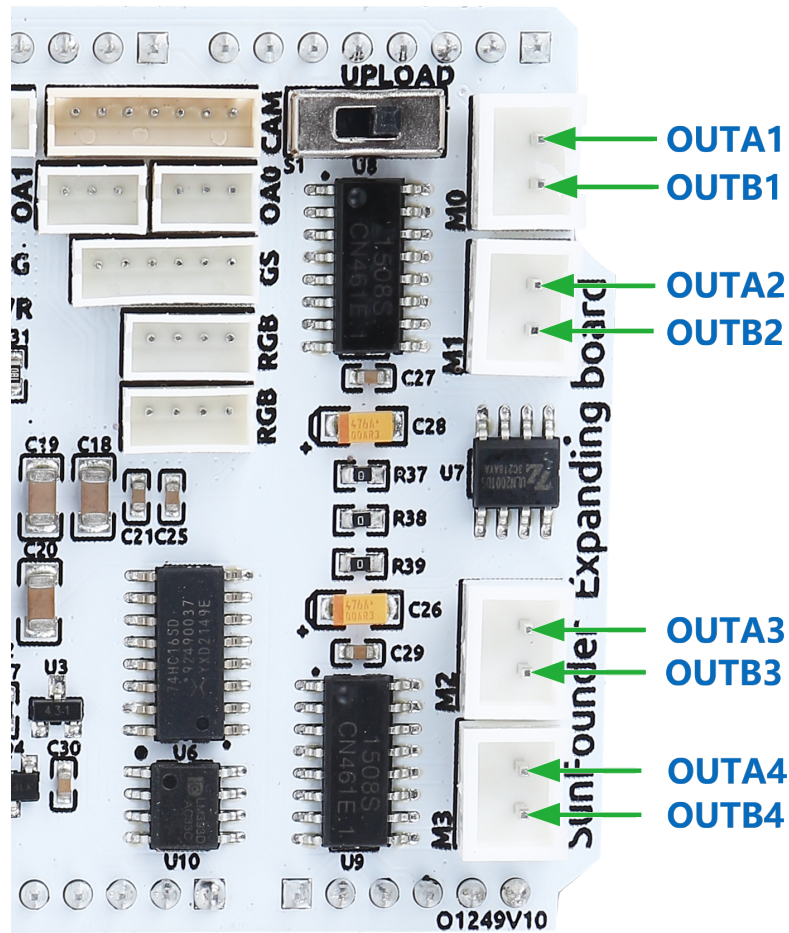


Hier ist das Schaltbild.



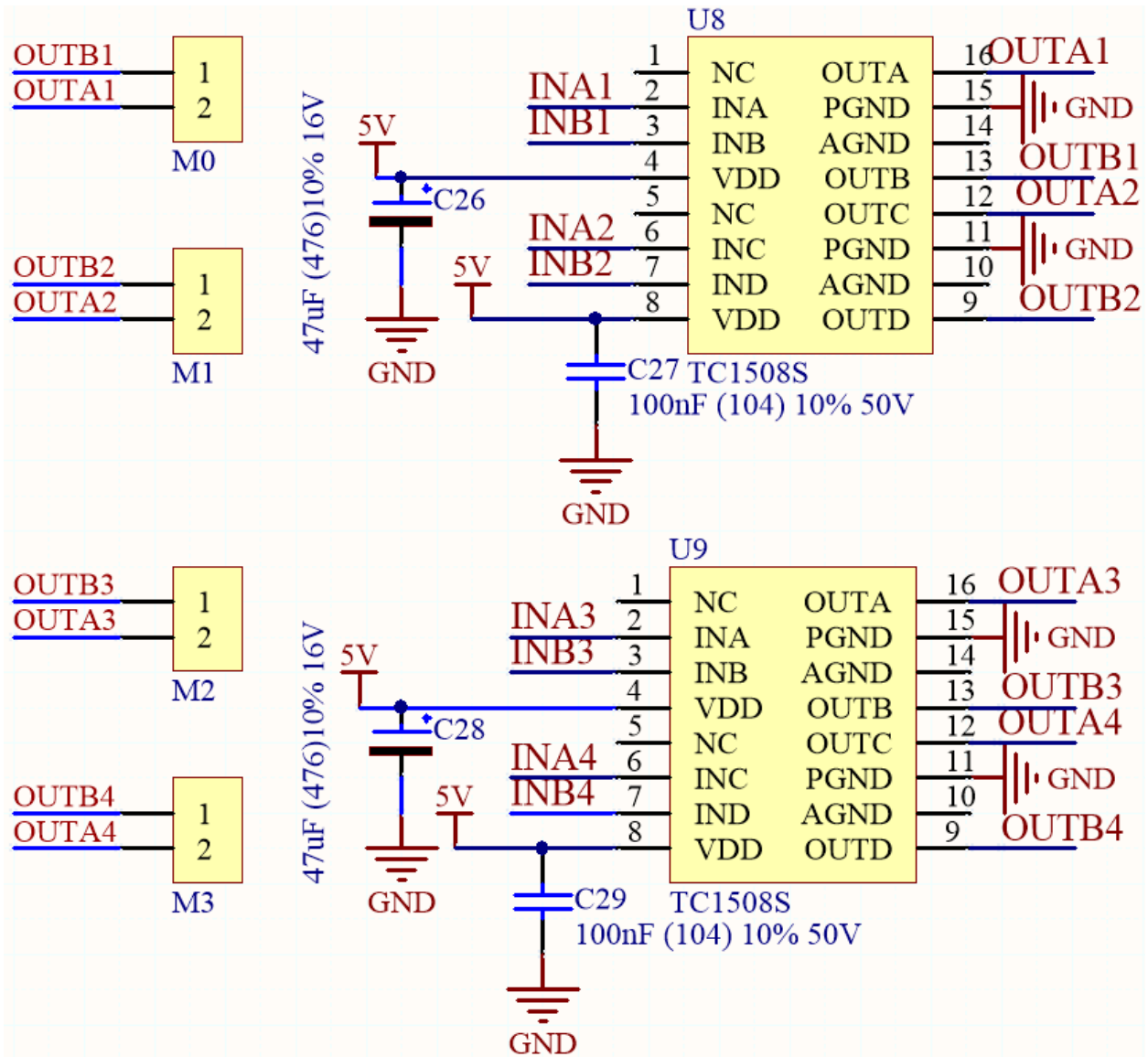
2.2.7 Motoranschluss

Hier ist das Pinout der 4 Motoranschlüsse.



Diese 4 Motorengruppen werden von 2 TC1508S Chips angetrieben. Dies ist ein Dual-Kanal-Motortreiberchip, der einen maximalen Dauerstrom von bis zu 1.8A pro Kanal ausgeben kann.

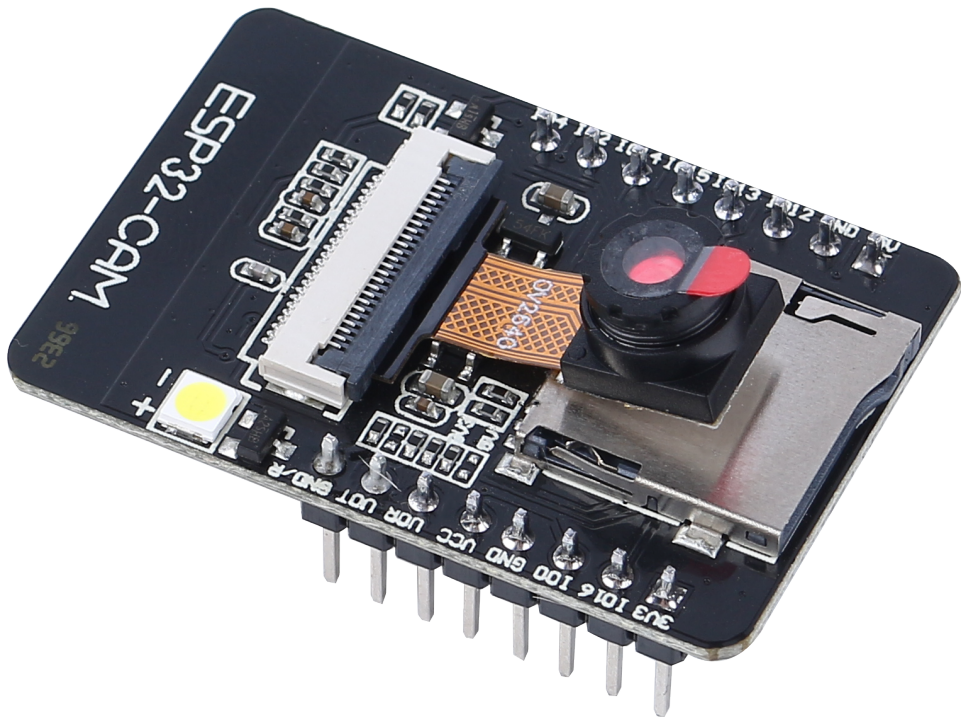
Hier ist das Schaltbild.



Die entsprechenden Steuerepins der 4 Motoranschlüsse sind unten aufgeführt.

Arduino Board	Zeus Car Shield
~3	OUTA1
4	OUTB1
~5	OUTA2
~6	OUTB2
A0	OUTB4
A1	OUTA4
A2	OUTB3
A3	OUTA3

2.3 ESP32 CAM



Das ESP32-CAM ist ein sehr kleines Kameramodul mit dem ESP32-S-Chip, das etwa \$10 kostet. Neben der OV2640-Kamera und mehreren GPIOs zum Anschluss von Peripheriegeräten verfügt es auch über einen microSD-Kartensteckplatz, der nützlich sein kann, um Bilder, die mit der Kamera aufgenommen wurden, zu speichern oder um Dateien für Clients bereitzustellen.

Das Modul kann als kleinstes System selbstständig arbeiten, mit einer Größe von nur 27*40,5*4,5mm und einem Tiefschlafstrom von nur 6mA.

ESP32-CAM kann in verschiedenen IoT-Anwendungen weit verbreitet eingesetzt werden und eignet sich für Heim-Smart-Geräte, industrielle drahtlose Steuerung, drahtlose Überwachung, QR-drahtlose Identifizierung, drahtlose Positionierungssystemsignale und andere IoT-Anwendungen. Es ist eine ideale Lösung für IoT-Anwendungen.

Technische Daten

Modulmodell	ESP32-CAM
Gehäuse	DIP-16
Größe	27*40,5*4,5(±0,2)mm
SPI Flash	standardmäßig 32Mbit
RAM	Intern 520KB + Extern 8MB PSRAM
Bluetooth	Bluetooth 4.2 BR/EDR und BLE-Standards
Wi-Fi	802.11 b/g/n/e/i
Unterstützte Schnittstellen	UART, SPI, I2C, PWM
Unterstützung TF-Karte	bis zu 4G
IO Pins	9
Serielle Ports Geschwindigkeit	standardmäßig 115200 bps
Bildausgabeformat	JPEG(nur OV2640 unterstützt), BMP, GRAUSTUFEN

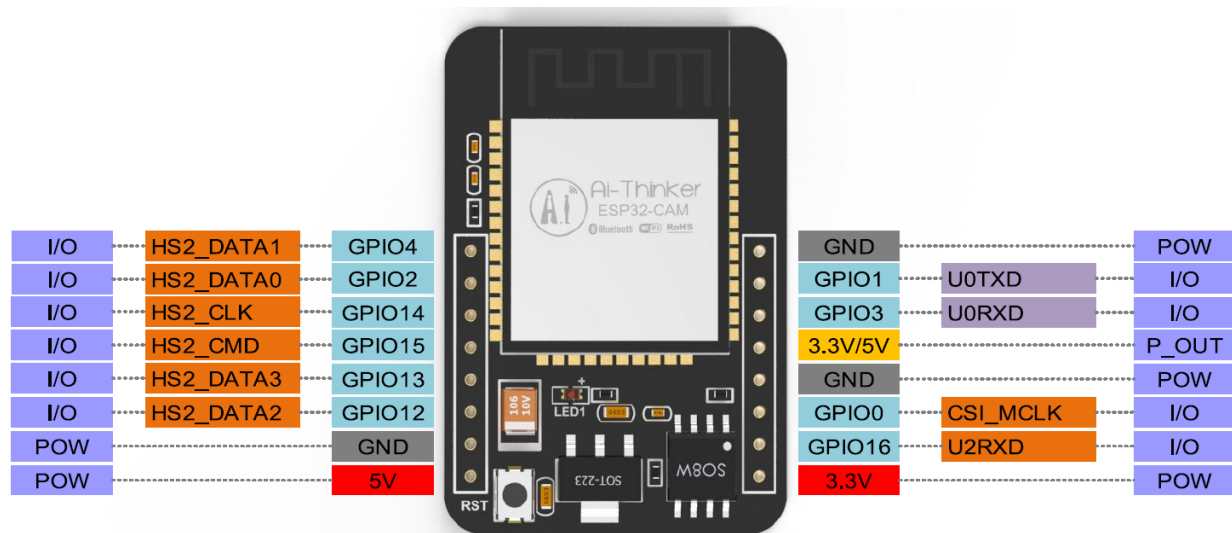
Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Frequenzbereich	2400 ~2483,5MHz
Antennentyp	On-board PCB-Antenne, Gewinn 2dBi
Sendeleistung	802.11b: 17±2 dBm (@11Mbps)
	802.11g: 14±2 dBm (@54Mbps)
	802.11n: 13±2 dBm (@MCS7)
Empfangsempfindlichkeit	CCK, 1 Mbps: -90dBm,
	CCK, 11 Mbps: -85 dBm
	6 Mbps (1/2 BPSK): -88 dBm
	54 Mbps (3/4 64-QAM): -70dBm
	MCS7 (65 Mbps, 72.2 Mbps): -67dBm
Stromverbrauch	Flash aus: 180mA@5V,
	Flash an und Helligkeit auf Maximum: 310mA@5V
	Tiefschlaf: der niedrigste Stromverbrauch kann 6mA@5V erreichen
	Modemschlaf: mindestens 20mA@5V
	Lichtschlaf: mindestens 6,7mA@5V
Sicherheit	WPA/WPA2/WPA2-Enterprise/WPS
Spannungsbereich	4,75-5,25V
Betriebstemperatur	-20 °C ~ 70 °C
Lagerungsumgebung	-40 °C ~ 125 °C, < 90%RH

ESP32-CAM Pinbelegung

Das folgende Bild zeigt die Pinbelegung des ESP32-CAM (AI-Thinker Modul).



- Es gibt drei **GND** Pins und drei Pins für die Stromversorgung: 3,3V, 5V und entweder 3,3V oder 5V.
- **GPIO 1** und **GPIO 3** sind die seriellen Pins. Diese Pins werden benötigt, um Code auf Ihr Board hochzuladen.
- Außerdem spielt **GPIO 0** eine wichtige Rolle, da er bestimmt, ob das ESP32 im Flash-Modus ist oder nicht. Wenn **GPIO 0** mit **GND** verbunden ist, befindet sich das ESP32 im Flash-Modus.
- Die folgenden Pins sind intern mit dem microSD-Kartenleser verbunden:
 - GPIO 14: CLK
 - GPIO 15: CMD
 - GPIO 2: Daten 0

- GPIO 4: Daten 1 (auch mit der Onboard-LED verbunden)
- GPIO 12: Daten 2
- GPIO 13: Daten 3

Hinweis

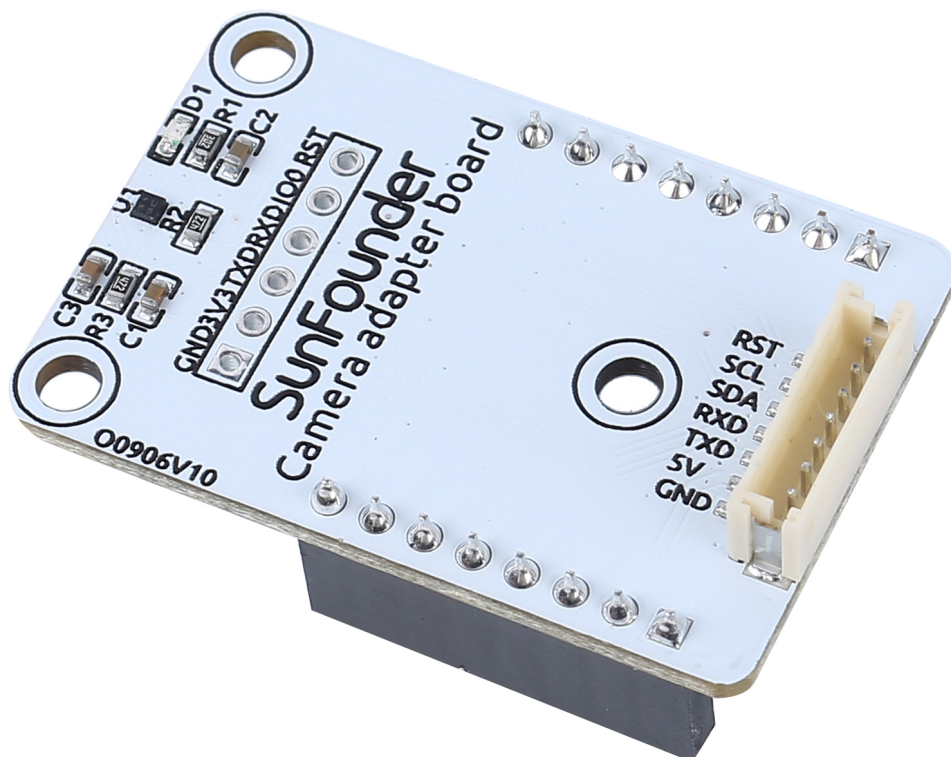
- Bitte stellen Sie sicher, dass die Eingangsleistung des Moduls mindestens 5V 2A beträgt, sonst können Wasserlinien im Bild auftreten.
- Der ESP32 GPIO32 Pin steuert die Kamera Stromversorgung. Wenn die Kamera arbeitet, ziehen Sie GPIO32 auf niedrig.
- Da GPIO0 mit dem Kamera XCLK verbunden ist, lassen Sie GPIO0 in der Luft, wenn Sie ihn verwenden, und verbinden Sie ihn nicht mit hohem oder niedrigem Level.
- Die Standard-Firmware ist bereits ab Werk enthalten und es wird kein zusätzlicher Download bereitgestellt. Bitte seien Sie vorsichtig, wenn Sie andere Firmware neu brennen müssen.

Dokument

- Schaltplan:
- Kamera-Spezifikation (Englische Version):

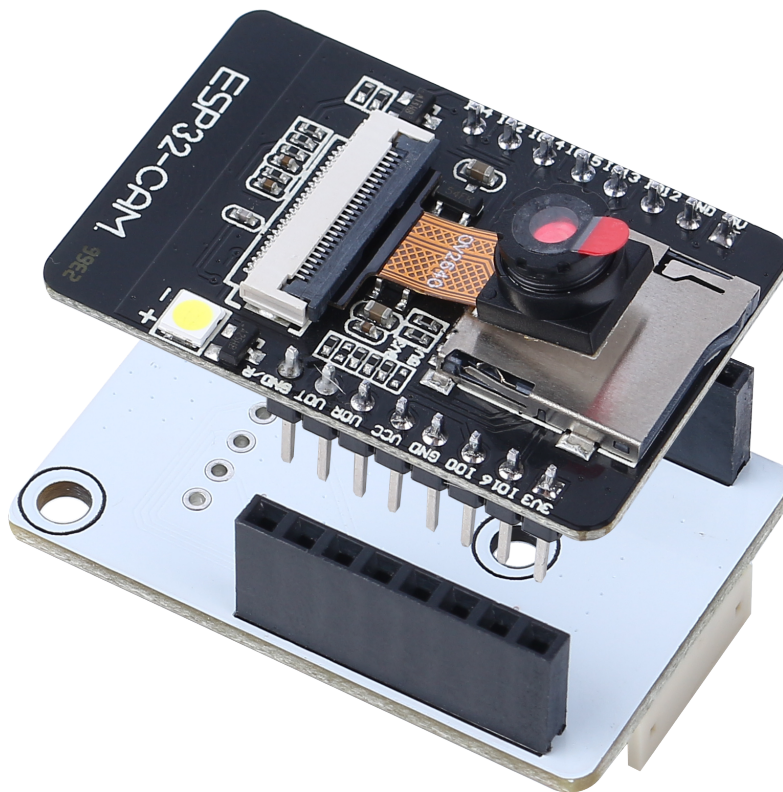
Bemerkung: Alle oben genannten Informationen stammen von

2.4 Kamera-Adapterplatine

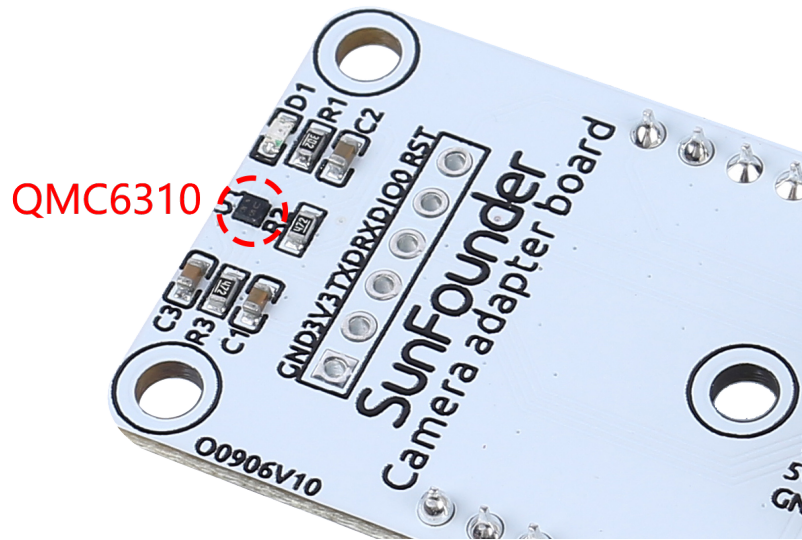


- **RST**: Zum Zurücksetzen des ESP32-CAM verwendet.
- **SCL**: Serieller Daten-Pin für QMC6310.
- **SDA**: Serieller Uhr-Pin des QMC6310.
- **RXD**: Der RXD des ESP32-CAM. Über diese beiden seriellen Pins, RXD und TXD, muss der Code auf den ESP32-CAM hochgeladen werden.
- **TXD**: TXD des ESP32-CAM.
- **5V**: 5V DC Versorgungseingang.
- **GND**: Erdungseingang.

Wie der Name schon sagt, ist die Kamera-Adapterplatine eine Erweiterungsplatine für den ESP-32 CAM. Sie dient dazu, den ESP32-CAM zu erweitern, damit er am Roboter befestigt werden kann und einfach verdrahtet werden kann.



Da der geomagnetische Chip QMC6310 anfällig für Störungen von Motoren ist, haben wir ihn auf diese Kamera-Adapterplatine gelegt, um ihn so weit wie möglich von den Motoren entfernt zu halten.

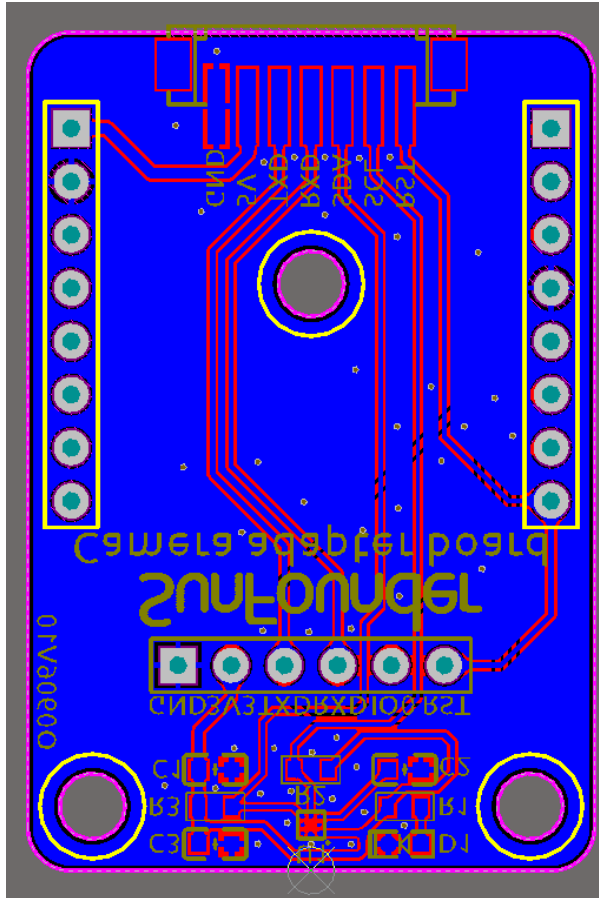


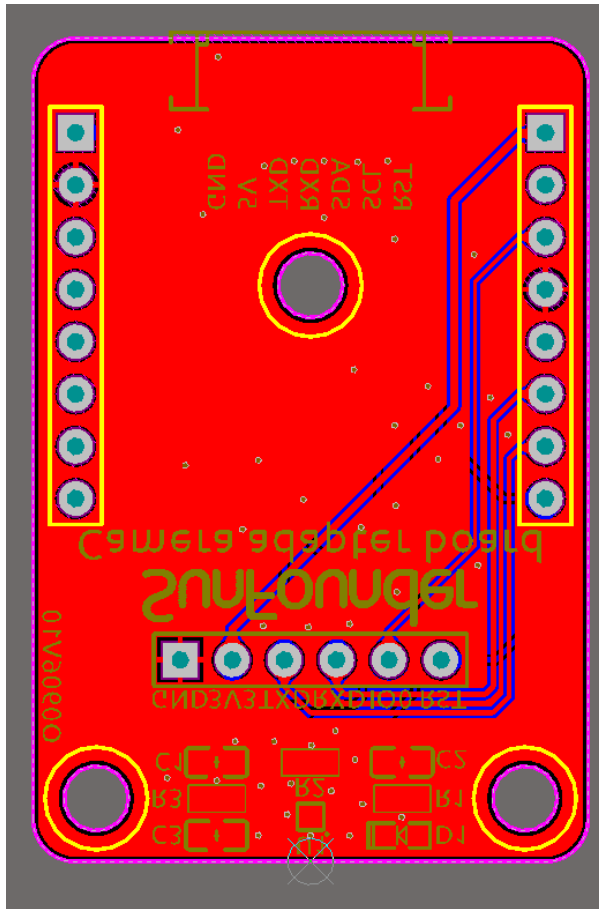
Merkmale

- Betriebsspannung: 5V.
- Schnittstellenmodell: ZH1.5, 7P.
- Abmessungen: 40mm x 27mm x 15mm.
- Kommunikationsprotokoll: UART und I2C.

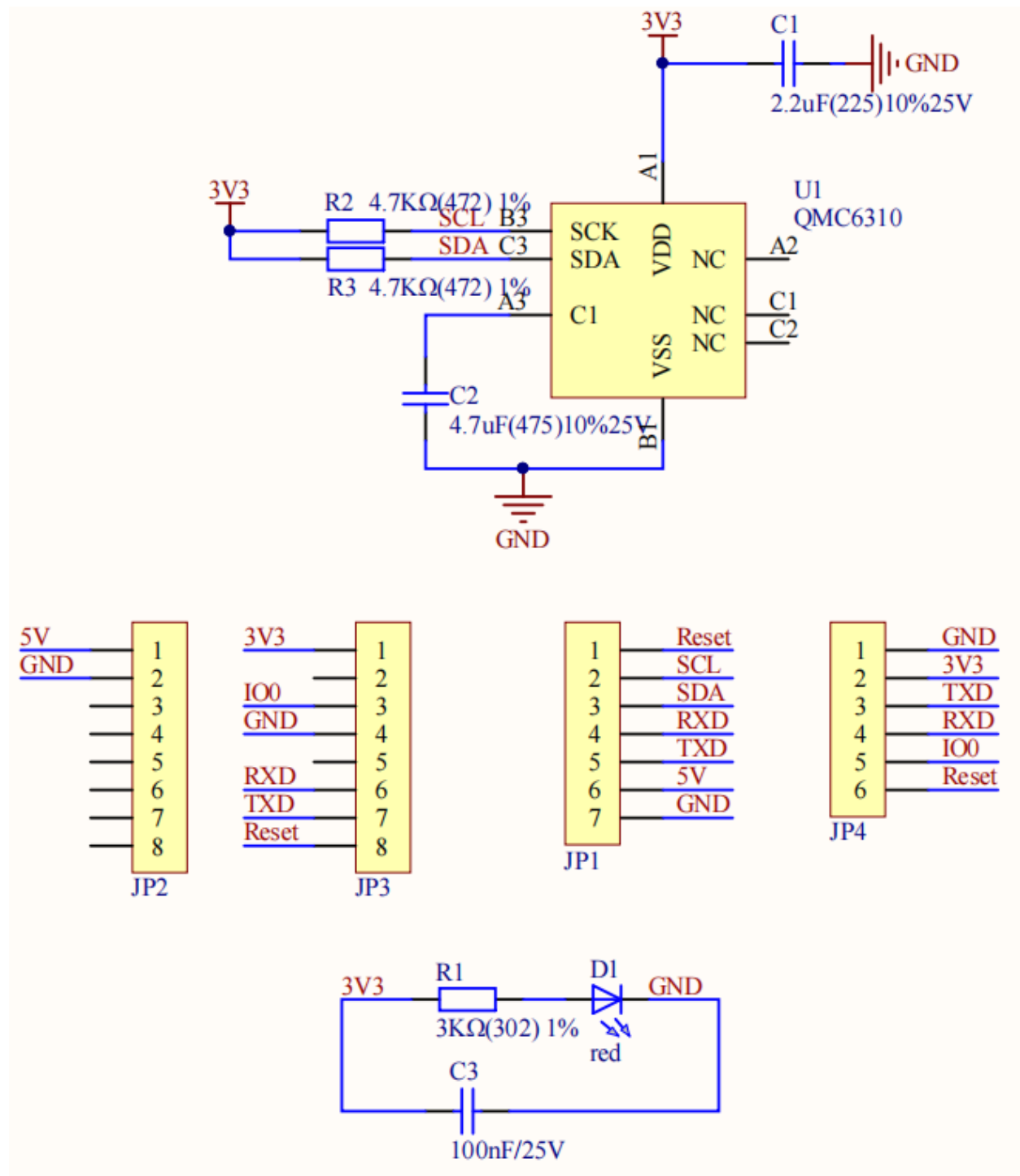
Dokumente

- PCB





- Schaltplan



Über QMC6310

Der QMC6310 ist ein Drei-Achsen-Magnetsensor, der Magnetsensoren und Signalzustands-ASIC in einem Silizium-chip integriert. Dieses Land Grid Array-Paket (LGA) richtet sich an Anwendungen wie E-Kompass, Kartenrotation, Spiele und persönliche Navigation in mobilen und tragbaren Geräten.

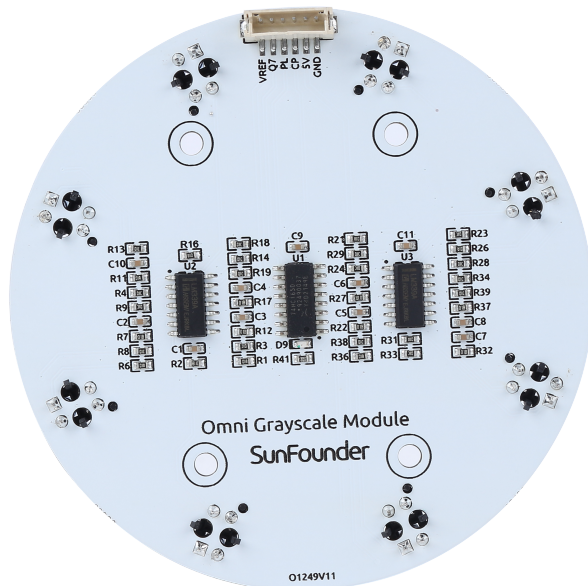
Der QMC6310 basiert auf modernster, hochauflösender, magnetoresistiver Technologie. Zusammen mit dem eigens entwickelten 16-Bit-ADC-ASIC bietet er Vorteile wie geringes Rauschen, hohe Genauigkeit, geringen Stromverbrauch, Offset-Aufhebung und Temperaturkompensationen. QMC6310 ermöglicht eine Kompasskopfgenauigkeit von 1° bis 2°.

Der I²C-Seriellbus ermöglicht eine einfache Schnittstelle.

Der QMC6310 befindet sich in einem 1,2x1,2x0,53mm³ Oberflächenmontage-8-Pin-LGA-Gehäuse.

•

2.5 Omni-Graustufenmodul



- **VREF**: Referenzspannungseingangspin. Der Wert jedes Sensors wird mit dieser Referenzspannung verglichen, um zu bestimmen, ob ein hoher oder niedriger Wert ausgegeben wird.
- **Q7**: Serieller Ausgang aus der letzten Stufe
- **PL**: Asynchroner paralleler Ladeeingang (aktiv LOW)

- **CP**: Clock-Eingang (LOW-to-HIGH Flanken-gesteuert)
- **5V**: 3,3 bis 5V DC Spannungseingang
- **GND**: Masseanschluss

Dies ist ein Omni-Graustufenmodul zur Linienverfolgung und Kantenerkennung. „Omni“ bedeutet omnidirektional, was bedeutet, dass das Modul über 8 TCRT5000 Sendesensoren verfügt, die im Kreis angeordnet sind, um schwarze Linien aus jeder Richtung zu erkennen.

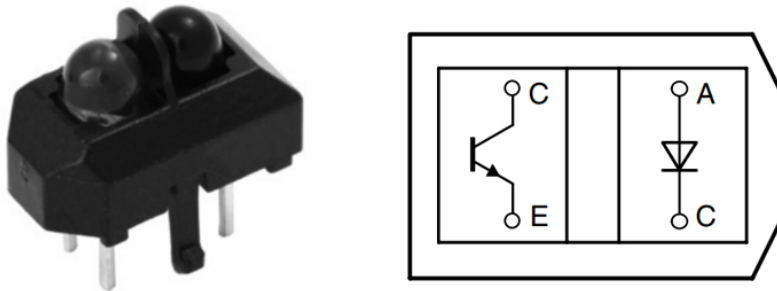
Dies ermöglicht es einem Roboterwagen wie dem Zeus Car mit Mecanum-Rädern, der Linie in verschiedenen Winkeln zu folgen, ohne dass die Vorderseite des Autos nach vorne zeigt.

Die Empfindlichkeit des Moduls in der aktuellen Umgebung kann angepasst werden, indem der VREF-Wert geändert wird. Beim Zeus Car Shield wird das blaue Potentiometer verwendet, um den Wert des VREF-Pins einzustellen.

Funktionsprinzip

•

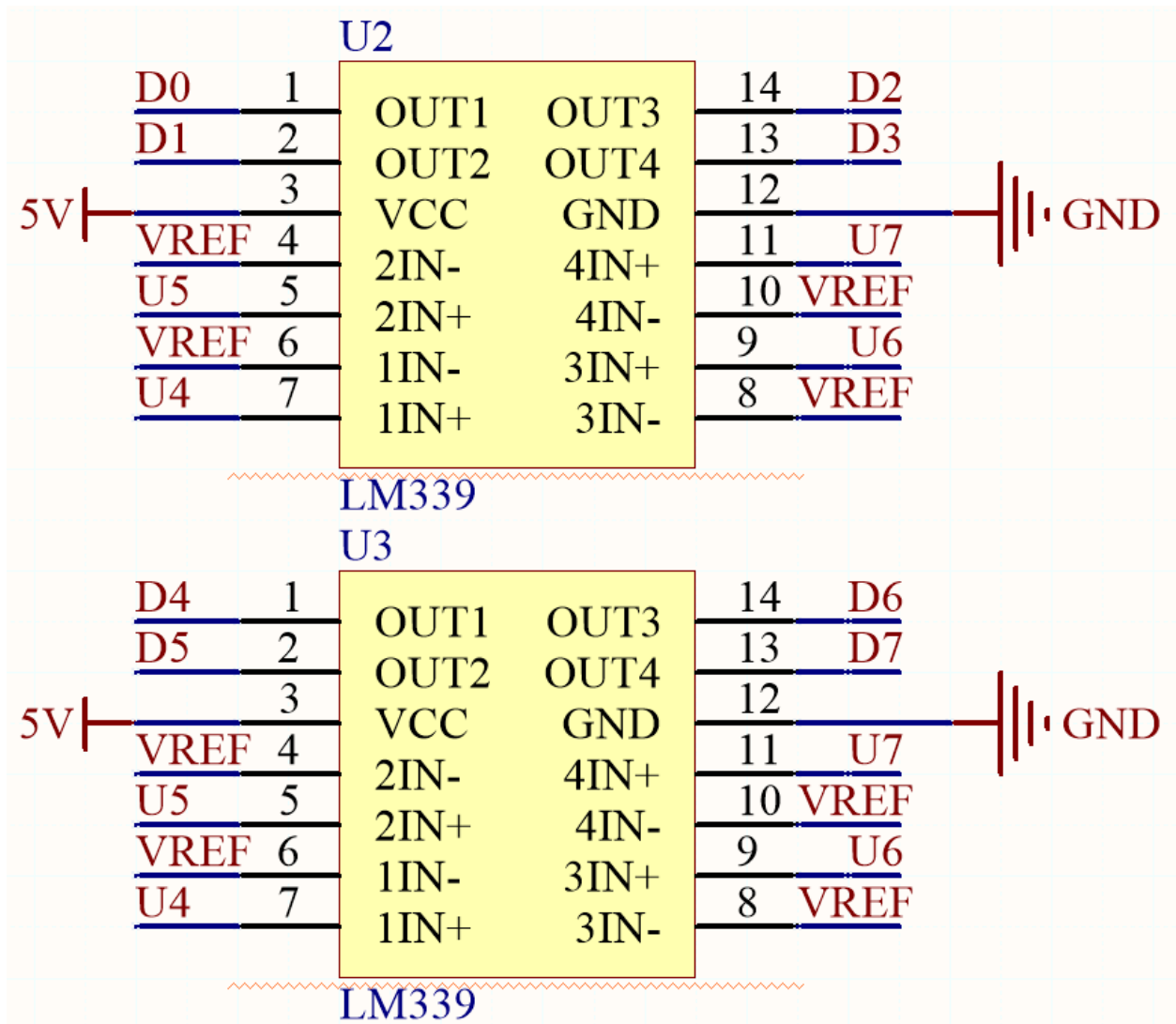
Im Modul sind acht TCRT5000 Sendesensoren integriert, die auf der infraroten optischen Reflexion basieren und eine infrarote LED und einen Fototransistor enthalten, der mit einem Bleimaterial abgedeckt ist, um sichtbares Licht zu blockieren.



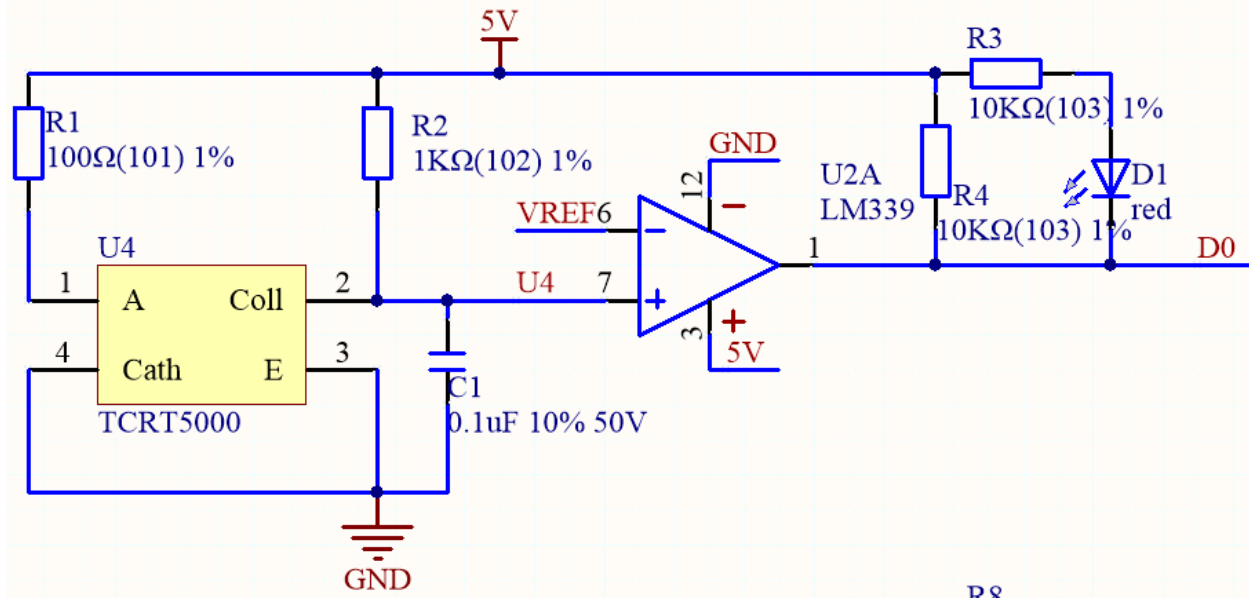
Bei der Arbeit sendet die infrarote LED des TCRT5000 kontinuierlich infrarotes Licht (unsichtbares Licht) mit einer Wellenlänge von 950nm aus. Wenn das emittierte infrarote Licht nicht von einem Hindernis zurück reflektiert wird oder die Reflexionsintensität unzureichend ist, arbeitet der Fototransistor nicht. Wenn das infrarote Licht mit ausreichender Intensität reflektiert und gleichzeitig vom Fototransistor empfangen wird, ist der Fototransistor in Betrieb und liefert einen Ausgang.

•

Auf diesem Omni-Graustufenmodul gibt es 2 LM339-Chips, die insgesamt 8 differentielle Komparatoren enthalten. Diese differentiellen Komparatoren werden verwendet, um den aktuellen Sensorwert mit einem Referenzwert zu vergleichen, um zu bestimmen, ob ein hoher oder niedriger Wert ausgegeben wird. So wissen Sie, ob eine schwarze Linie erkannt wird.



Nachfolgend finden Sie ein Schema eines der Kanäle.



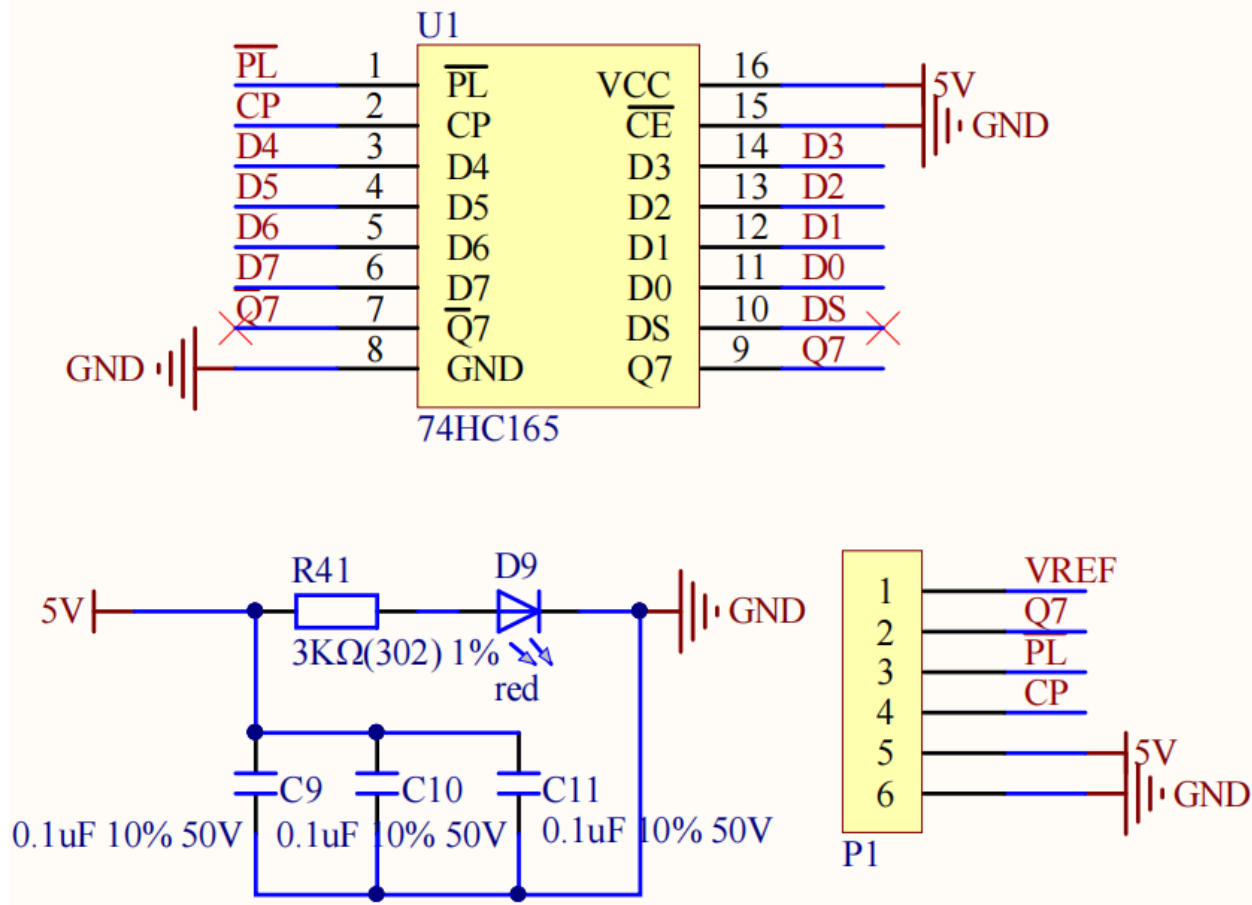
- Legen Sie eine Referenzspannung am VREF-Pin an (diese Referenzspannung wird über ein Potentiometer am Zeus Car Shield eingestellt) und fügen Sie diese Referenzspannung dem invertierenden Eingang (-) des Komparators hinzu.
- Fügen Sie den Kollektor des Fototransistors des TCRT5000-Sensors dem in-Phasen-Eingang (+) des Komparators hinzu.
- Wenn der vom TCRT5000-Sensor ausgesendete Infrarotstrahl nicht zurückreflektiert wird oder die Reflexionsintensität unzureichend ist, funktioniert der lichtempfindliche Transistor nicht, und der Kollektor ist zu diesem Zeitpunkt mit dem Pull-up-Widerstand auf 5V verbunden. Daher ist der in-Phasen-Eingang (+) des Komparators größer als der invertierende Eingang (-).
- Der Komparatorausgang ist hoch und die Anzeige leuchtet nicht auf. Und umgekehrt.
- Da die schwarze Oberfläche Licht absorbiert, reflektiert sie weniger Infrarotlicht. Daher gibt der Komparator auf der schwarzen Oberfläche einen hohen Wert aus und die Anzeige leuchtet nicht auf.
- Die weiße Oberfläche reflektiert mehr Infrarotstrahlen, und der lichtempfindliche Transistor leitet, sodass der Wert des in-Phasen-Eingangs kleiner als der invertierte Eingang ist, und der Komparator gibt einen niedrigen Wert aus, und die Anzeige leuchtet auf.

Diese 8 Sensordaten werden über den 74HC165 (8-Bit-Parallel-Eingabe serieller Ausgangsschieberegister) an das Arduino-Board übertragen.

•

Der 74HC165 ist ein 8-Bit paralleler Eingabe serieller Ausgangsschieberegister, der exklusive serielle Ausgänge (Q0 und Q7) in der Endstufe erhalten kann. Wenn der parallele Lese-Eingang (PL) niedrig ist, werden die parallelen Daten von D0 bis D7 asynchron in das Register eingelesen. Und wenn PL hoch ist, werden die Daten seriell vom DS-Eingang in das Register eingelesen, wobei sich ein Bit bei jeder aufsteigenden Flanke eines Taktimpulses nach rechts verschiebt (Q0 → Q1 → Q2 usw.). Mit dieser Funktion kann die Parallel-zu-Seriell-Erweiterung erreicht werden, indem einfach der Q7-Ausgang an den DS-Eingang der nächsten Stufe gebunden wird.

Der Takteingang des 74HC165 ist eine „gated or“ Struktur, die es ermöglicht, einen der Eingänge als Clock Enable (CE) Eingang mit niedriger Aktivität zu verwenden. Die CP- und CE-Pinzuordnungen sind unabhängig und können bei Bedarf ausgetauscht werden. CE darf nur steigen, wenn CP hoch ist. CP oder CE sollten vor der steigenden Flanke von PL hochgesetzt werden, um eine Datenverschiebung im aktiven Zustand von PL zu verhindern.



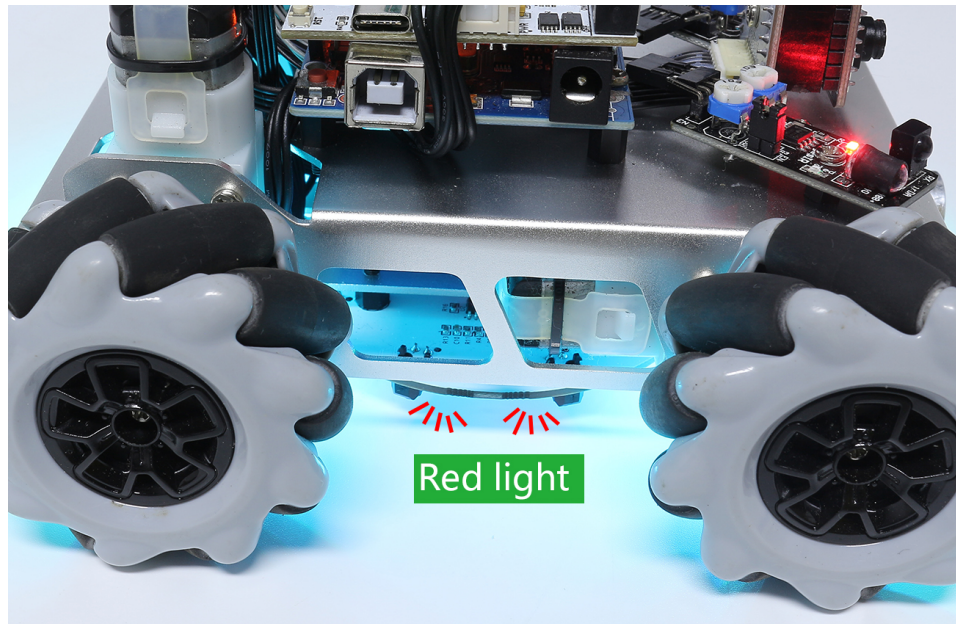
Merkmale

- Betriebsspannung: 3,3 ~ 5V
- Ausgang: digital (an/aus)
- Asynchroner 8-Bit-Parallel-Ladevorgang
- Synchroner serieller Eingang
- Erkennungsschwelle: einstellbar über VREF-Pin
- Sensortyp TCRT5000
- Steckermodell ZH1.5-6P
- Betriebstemperatur: -10 °C bis +50 °C
- Abmessungen: 80mm x 80mm

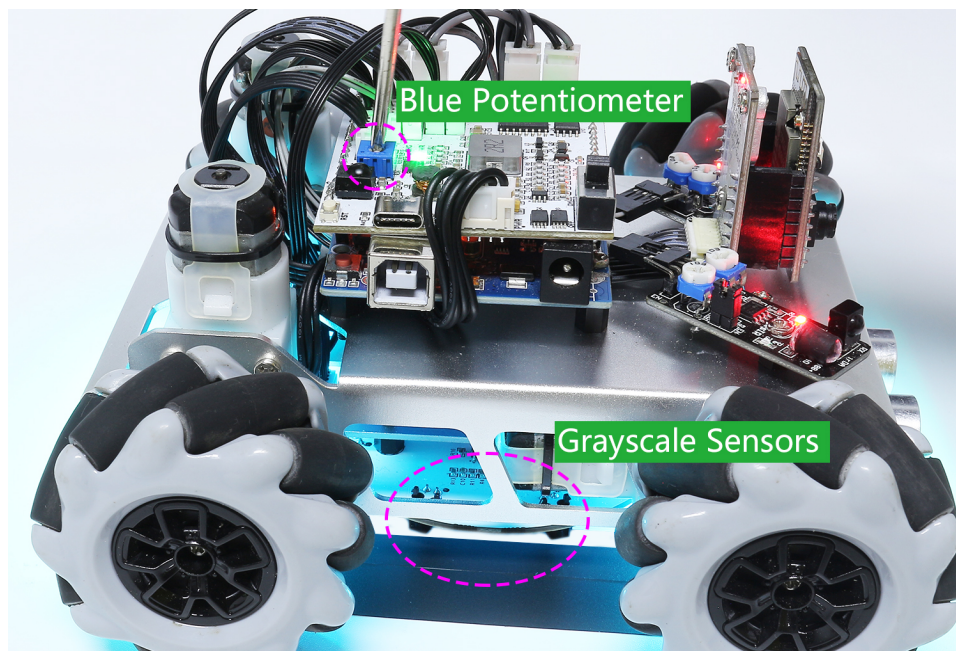
Kalibrierung des Moduls

Da jeder Untergrund unterschiedliche Grauwerte aufweist, ist der werkseitig eingestellte Grauschwellenwert möglicherweise nicht für Ihre aktuelle Umgebung geeignet. Daher müssen Sie dieses Modul vor der Verwendung kalibrieren. Es wird empfohlen, dass Sie es kalibrieren müssen, wann immer sich die Bodenfarbe stark ändert.

- Platzieren Sie den Zeus Car auf einer weißen Fläche und drehen Sie das Potentiometer, bis das Licht des Grausensors gerade leuchtet.



- Lassen Sie nun die beiden Grausensoren an der Seite genau zwischen der schwarzen Linie und der weißen Fläche liegen und drehen Sie das Potentiometer langsam, bis die Signalanzeige gerade erlischt.



- Sie können wiederholt über die schwarze Linie und die weiße Fläche fahren, um sicherzustellen, dass die Lichter des Grausensors aus sind, wenn sie zwischen der schwarzen Linie und der weißen Fläche liegen, und an sind, wenn sie auf der weißen Fläche sind, was darauf hinweist, dass das Modul erfolgreich kalibriert wurde.

2.6 Ultraschallmodul



- **TRIG:** Triggerimpulseingang
- **ECHO:** Echoimpulsausgang
- **GND:** Masse
- **VCC:** 5V Versorgung

Hierbei handelt es sich um den HC-SR04 Ultraschallabstandssensor, der eine berührungslose Messung von 2 cm bis 400 cm mit einer Reichweitengenauigkeit von bis zu 3 mm bietet. Das Modul ist mit einem Ultraschallsender, einem Empfänger und einer Steuerschaltung ausgestattet.

Für Ihre Messprojekte müssen Sie nur 4 Pins anschließen: VCC (Stromversorgung), Trig (Trigger), Echo (Empfang) und GND (Masse).

Eigenschaften

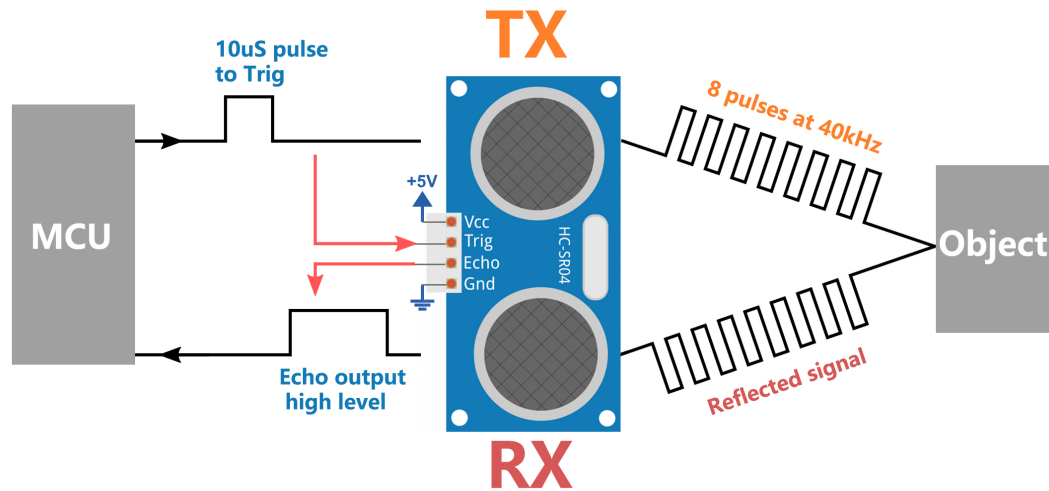
- Betriebsspannung: DC5V
- Arbeitsstrom: 16mA
- Arbeitsfrequenz: 40Hz
- Max. Reichweite: 500cm
- Min. Reichweite: 2cm
- Trigger-Eingangssignal: 10uS TTL-Impuls
- Echo-Ausgangssignal: TTL-Pegel-Eingangssignal und Reichweite proportional
- Steckverbinder: XH2.54-4P
- Abmessungen: 46x20,5x15 mm

Prinzip

Die grundlegenden Prinzipien sind:

- Mindestens 10us hohes Pegelsignal mit IO auslösen.
- Das Modul sendet einen 8-Zyklus-Ultraschallburst mit 40 kHz und erkennt, ob ein Pulssignal empfangen wird.

- Echo gibt ein hohes Pegelsignal aus, wenn ein Signal zurückkommt; die Dauer des hohen Pegels ist die Zeit von der Emission bis zur Rückkehr.
- Entfernung = (High-Level-Zeit x Schallgeschwindigkeit (340M/S)) / 2



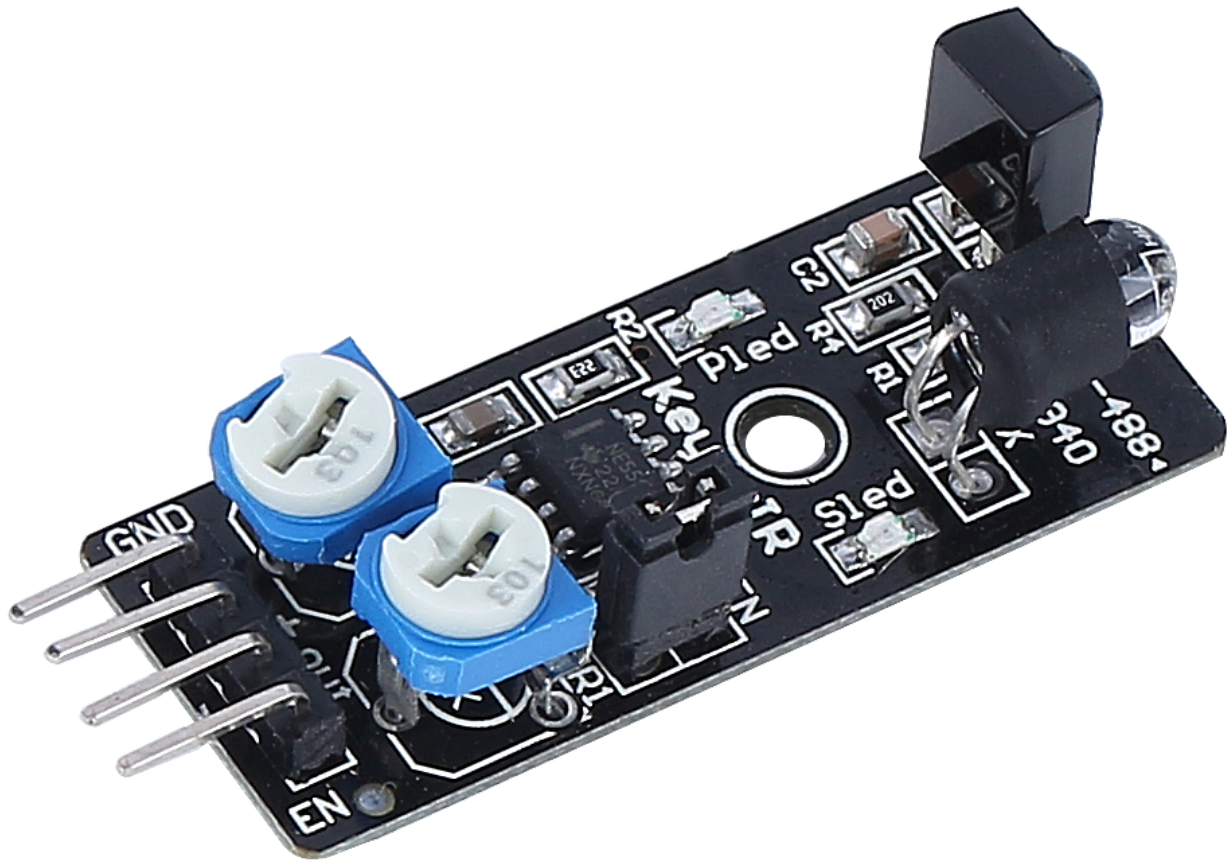
Formel:

- $us / 58 = \text{Entfernung in Zentimetern}$
- $us / 148 = \text{Entfernung in Zoll}$
- $\text{Entfernung} = \text{High-Level-Zeit} \times \text{Geschwindigkeit (340M/S)} / 2$

Anwendungshinweise

- Dieses Modul sollte nicht beim Einschalten verbunden werden. Falls notwendig, sollte zuerst die GND des Moduls verbunden werden. Andernfalls wird es die Arbeit des Moduls beeinflussen.
- Die Fläche des zu messenden Objekts sollte mindestens 0,5 Quadratmeter betragen und möglichst flach sein. Andernfalls beeinflusst dies die Ergebnisse.

2.7 IR Hinderniserkennungsmodul



- **GND:** Erdanschluss
- **+**: 3,3 bis 5V DC Spannungseingang
- **Out:** Signalausgangspin, standardmäßig hoch und gibt ein niedriges Signal aus, wenn ein Hindernis erkannt wird.
- **EN:** Modulaktivierungspin. Wenn dieser niedrig ist, ist das Modul aktiv und standardmäßig mit einem Jumper verbunden.

Dies ist ein übliches IR Hinderniserkennungsmodul, das ein Paar aus IR sendenden und empfangenden Komponenten verwendet. Grundsätzlich sendet der Sender infrarotes Licht aus. Wenn in der Erfassungsrichtung ein Hindernis auftritt, wird das Infrarotlicht zurückgeworfen und vom Empfänger erfasst. Zu diesem Zeitpunkt leuchtet die Anzeige auf. Nach der Schaltkreisverarbeitung wird ein niedriges Signal ausgegeben.

Erfassungsentfernung liegt zwischen 2-40cm und besitzt eine hervorragende Störfestigkeit. Verschiedenfarbige Objekte haben unterschiedliche Reflektivitätsstufen, sodass dunklere Objekte eine kürzere Erkennungsentfernung haben. Dieser Sensor erkennt im Bereich von 2-30cm gegenüber einer weißen Wand.

Wenn der Aktivierungspin auf niedrigem Niveau ist, arbeitet das Modul. Sobald der Jumper gesteckt ist, wird der EN-Pin mit GND verbunden, und das Modul ist ständig in Betrieb. Wenn Sie den EN-Pin per Code steuern möchten, müssen Sie den Jumper entfernen.



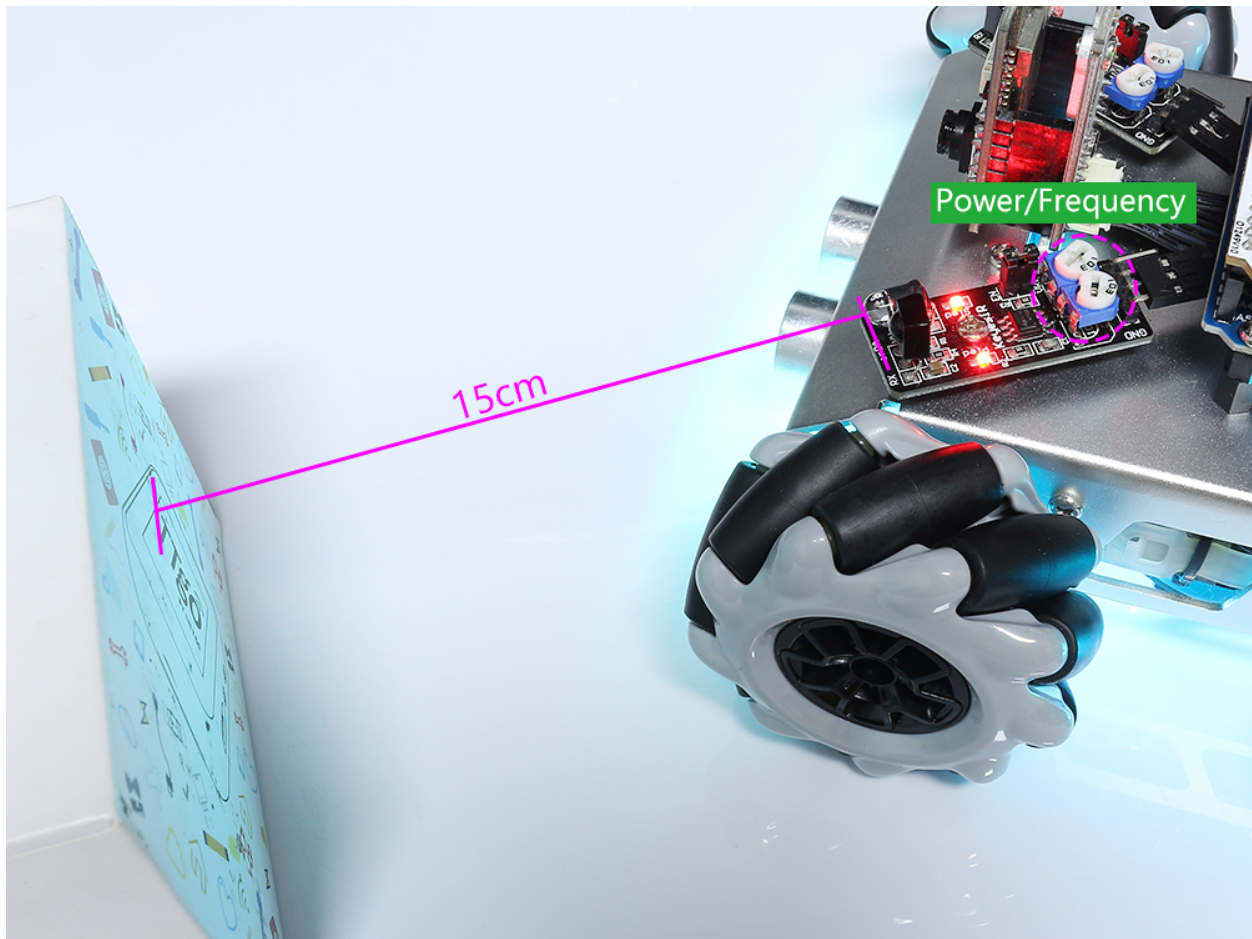
Einstellen der Erkennungsentfernung

Aufgrund verschiedener Lichtverhältnisse kann die werkseitig eingestellte Erkennungsentfernung unpassend sein, daher müssen Sie die tatsächliche Erkennungsentfernung vor Gebrauch anpassen.

Es gibt zwei Potentiometer auf dem Modul, eines zur Anpassung der Sendeleistung und eines zur Anpassung der Sendefrequenz. Durch die Einstellung dieser beiden Potentiometer können Sie die effektive Distanz anpassen.

Sie können ein weißes Hindernis in dem gewünschten Abstand vor das Modul stellen und eines der Potentiometer so einstellen, dass die Anzeigeleuchte gerade aufleuchtet. Bewegen Sie dann das Hindernis mehrmals, um zu sehen, ob die Anzeigeleuchte in dem gewünschten Abstand aufleuchtet.

Wenn die Anzeige an der gewünschten Stelle nicht aufleuchtet oder ständig leuchtet, können Sie das andere Potentiometer anpassen.



Merkmale

- Betriebsspannung: 3,3 V bis 5 V
- Ausgang: digital (ein/aus)
- Erkennungsschwelle: Einstellbar durch 2 Potentiometer
- Erfassungsbereich: 2 bis 40 cm
- Einstellung R5: Frequenzeinstellung 38 kHz (bereits optimiert)
- Einstellung R6: IR LED-Tastverhältniseinstellung (bereits optimiert)
- Betriebstemperatur: -10 °C bis +50 °C
- Effektiver Winkel: 35°
- I/O-Schnittstelle: 4-Draht-Schnittstelle (- / + / S / EN)
- Abmessungen: 45 x 16 x 10 mm
- Gewicht: 9 g

2.8 4 RGB LED-Streifen



- **+5V:** Gemeinsame Anode der drei LEDs und muss mit DC 5V verbunden werden
- **B:** Kathode der blauen LED
- **R:** Kathode der roten LED
- **G:** Kathode der grünen LED

Der RGB-Streifen besteht aus 4 RGB LEDs und kann jeden Farbton jeder Farbe erzeugen. Er stellt eine Mischung aus den drei Grundfarben dar; Rot, Blau und Grün.

Die verwendeten RGB LEDs sind vom Typ 5050 und in einer gemeinsamen Anodenkonfiguration angeschlossen. Jede LED stellt einen unabhängigen Schaltkreis dar und kann nach Belieben entlang des Schalters geschnitten werden, ohne andere Teile zu beschädigen. Hergestellt aus FPC-Platine, mit doppelseitigem Klebeband versehen, kann nach Belieben gebogen und fixiert werden.

Merkmale

- Betriebsspannung: DC5V
- Farbe: Vollfarb RGB
- Arbeitstemperatur: -15-50°C
- RGB Typ5050RGB
- Strom: 150mA (einzelner Schaltkreis)
- Leistung: 1,5W
- Dicke des Lichtstreifens: 2mm
- Breite des Lichtstreifens: 5,5mm
- Kabel: ZH1.5-4P, 25cm, 28AWG, Schwarz

2.9 18650 Batterie



- **VCC:** Batterie-Pluspol, hier gibt es zwei Sets von VCC und GND, um den Strom zu erhöhen und den Widerstand zu verringern.
- **Mittel:** Zum Ausgleichen der Spannung zwischen den beiden Zellen und zum Schutz der Batterie.
- **GND:** Batterie-Minuspol.

Dies ist ein individueller Batteriepack von SunFounder, bestehend aus zwei 18650 Batterien mit einer Kapazität von 2200mAh. Der Anschluss ist PH2.0-5P, welcher direkt geladen werden kann, nachdem er in das Zeus Car Shield eingesteckt wurde.

Merkmale

- Batterieladung: 5V/2A
- Batterieausgang: 5V/5A
- Batteriekapazität: 3,7V 2000mAh x 2
- Batterielaufzeit: 90min
- Ladezeit der Batterie: 130min
- Anschluss: PH2.0, 5P

2.10 TT Motor



Hier handelt es sich um einen TT Gleichstrommotor mit einem Übersetzungsverhältnis von 1:48. Er ist mit 2 x 250mm Kabeln mit XH2.54-2P Steckverbinder ausgestattet.

Sie können diese Motoren mit 3 ~ 6VDC betreiben. Selbstverständlich laufen sie bei höheren Spannungen etwas schneller.

Beachten Sie, dass dies sehr einfache Motoren ohne integrierten Encoder, Geschwindigkeitsregelung oder Positionsfeedback sind. Spannung geht rein, Drehung kommt raus. Es wird Unterschiede von Motor zu Motor geben, daher benötigen Sie für präzise Bewegungen ein separates Feedback-System.

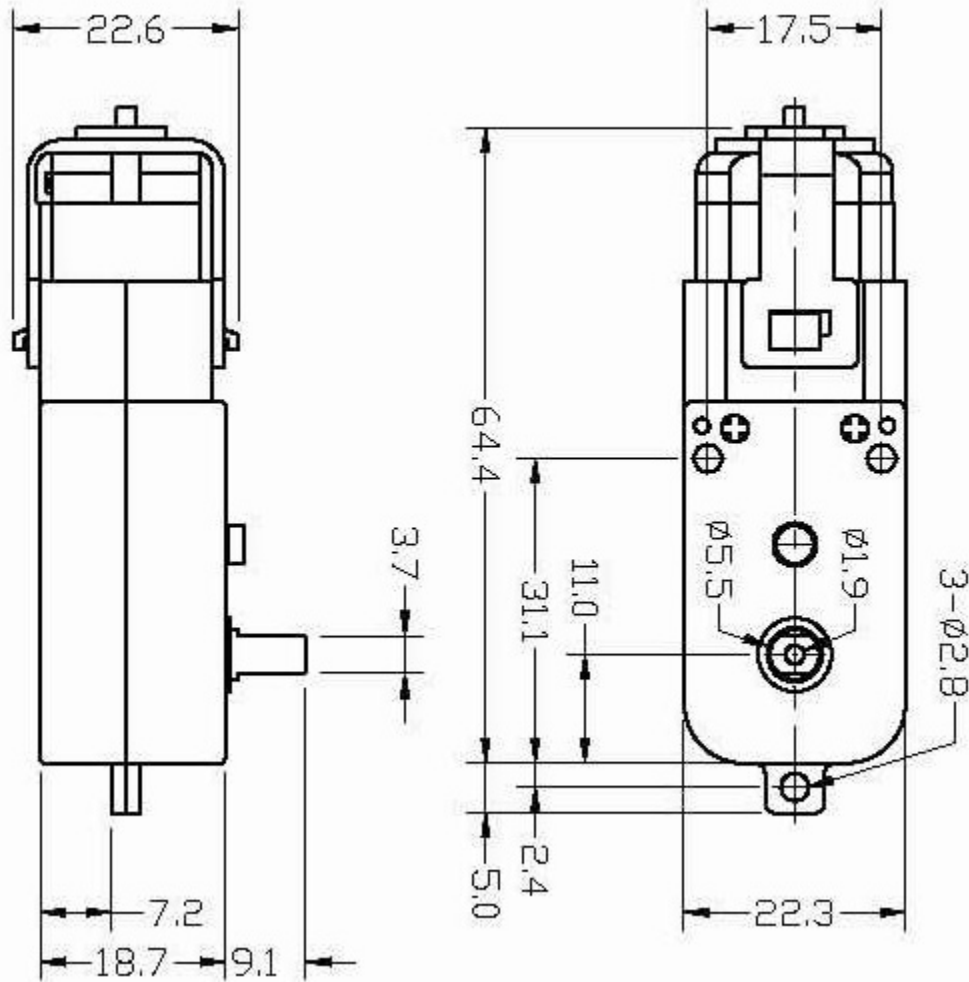
Eigenschaften

- Nennspannung: 3~6V
- Kontinuierlicher Leerlaufstrom: 150mA +/- 10%
- Min. Betriebsgeschwindigkeit (3V): 90+/- 10% RPM
- Min. Betriebsgeschwindigkeit (6V): 200+/- 10% RPM
- Haltemoment (3V): 0,4kg.cm
- Haltemoment (6V): 0,8kg.cm
- Übersetzungsverhältnis: 1:48
- Körperabmessungen: 70 x 22,3 x 36,9mm
- Kabel: Grau und Schwarz, 24AWG, 250mm
- Steckverbinder: XH2.54-2P

- Gewicht: 30,6g

Abmessungszeichnung

Einheit: mm



2.11 Mecanum-Rad

Was ist ein Mecanum-Rad?



Das Mecanum-Rad ist ein omnidirektionales Rad-Design, das es einem landgestützten Fahrzeug ermöglicht, sich in jede Richtung zu bewegen. Es wird manchmal auch als schwedisches Rad oder Ilon-Rad nach seinem Erfinder bezeichnet.

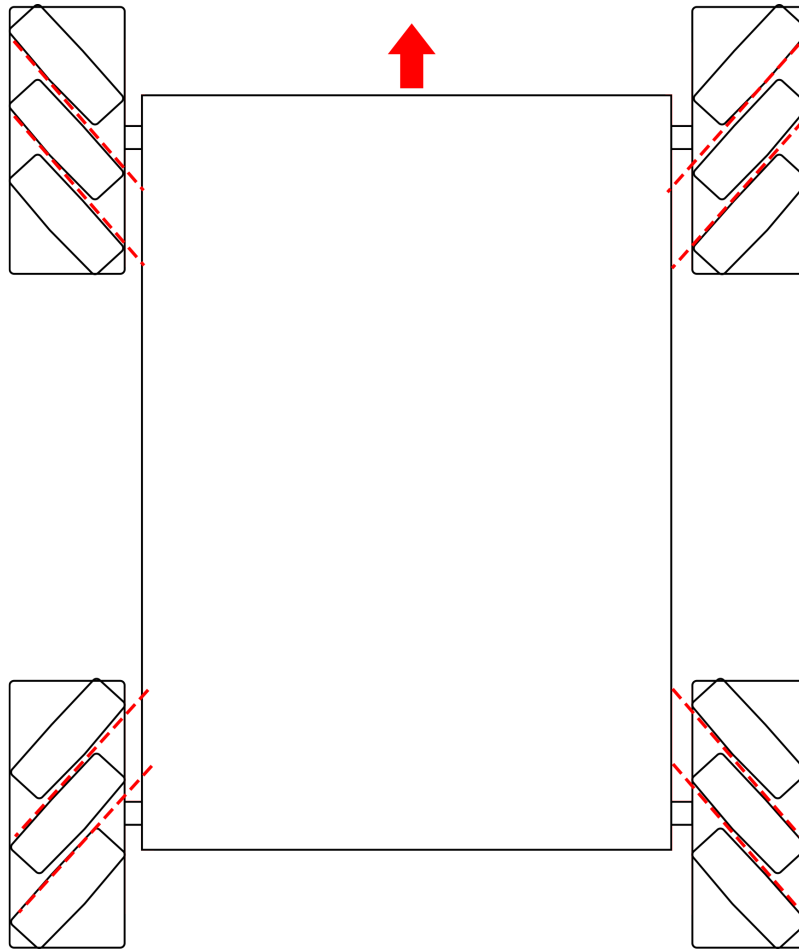
Das Mecanum-Rad ist eine Art von radlosen Rädern, mit einer Reihe von gummierten externen Rollen, die schräg an den gesamten Umfang des Rades angebracht sind. Diese Rollen haben typischerweise eine Rotationsachse im 45° -Winkel zur Radfläche und im 45° -Winkel zur Achslinie.

Jedes Mecanum-Rad ist ein unabhängiges, nicht lenkendes Antriebsrad mit eigenem Antriebsstrang. Beim Drehen erzeugt es eine Vortriebskraft senkrecht zur Rollenachse, die in eine längs- und eine quer gerichtete Komponente im Bezug zum Fahrzeug umgewandelt werden kann.

Das Mecanum-Rad kann je nach 45° -Winkel in linke und rechte Räder unterteilt werden, die Spiegelbilder voneinander sind.



Mecanum-Räder werden normalerweise in Gruppen von vier verwendet, wie in der untenstehenden Abbildung gezeigt.



Merkmale

- Durchmesser: 60mm
- Dicke: 30,62mm
- Rollen: 9 Stück
- Winkel: 45°
- Farbe: Schwarz
- Material: Kunststoff + Gummi

3.1 Kompilierungsfehler: SoftPWM.h: Datei oder Verzeichnis nicht gefunden

Wenn die Meldung „Kompilierungsfehler: SoftPWM.h: Datei oder Verzeichnis nicht gefunden“ erscheint, bedeutet dies, dass die SoftPWM Bibliothek nicht installiert ist.

Bitte beziehen Sie sich auf *Bibliotheken installieren (Arduino IDE 2)*, um die beiden erforderlichen Bibliotheken SoftPWM und IRLremote zu installieren.

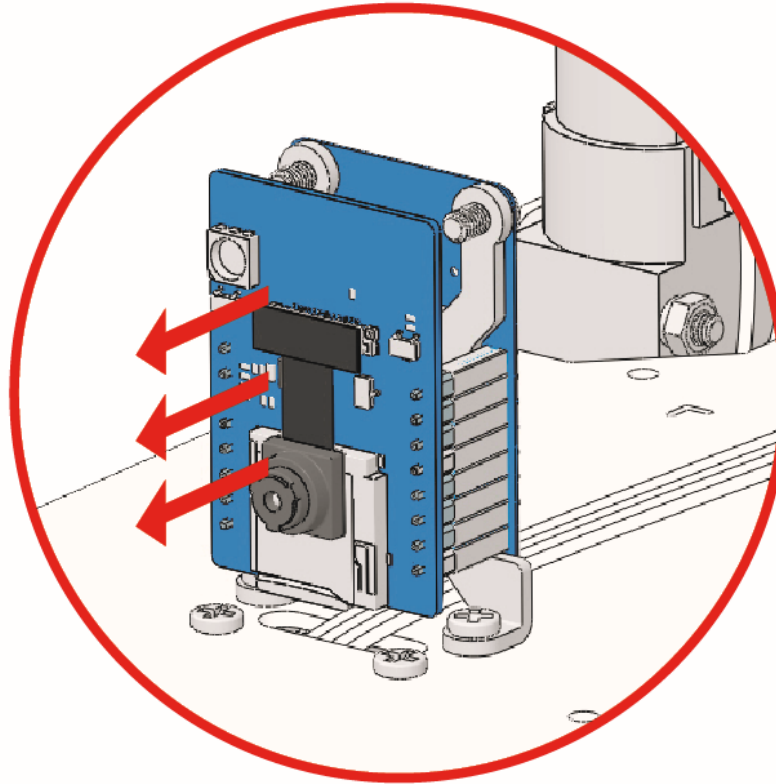
3.2 avrdude: stk500_getsync() Versuch 10 von 10: nicht synchronisiert: resp=0x6e?

Wenn die folgende Meldung immer wieder erscheint, nachdem Sie auf den Hochladen-Button geklickt haben und das Board und den Port korrekt ausgewählt haben:

```
avrdude: stk500_recv(): Programmierer antwortet nicht
avrdude: stk500_getsync() Versuch 1 von 10: nicht synchronisiert: resp=0x00
avrdude: stk500_recv(): Programmierer antwortet nicht
avrdude: stk500_getsync() Versuch 2 von 10: nicht synchronisiert: resp=0x00
avrdude: stk500_recv(): Programmierer antwortet nicht
avrdude: stk500_getsync() Versuch 3 von 10: nicht synchronisiert: resp=0x00
```

Zu diesem Zeitpunkt sollten Sie sicherstellen, dass die ESP32 CAM nicht angeschlossen ist.

Die ESP32-CAM und das Arduino-Board teilen sich die gleichen RX (Empfang) und TX (Senden) Pins. Deshalb sollten Sie die ESP32-CAM vor dem Hochladen des Codes trennen, um mögliche Konflikte oder Probleme zu vermeiden.



3.3 Q3: Wie kann ich die STT-Funktion auf meinem Android-Gerät nutzen?

Für die STT-Funktion muss das Android-Mobilgerät mit dem Internet verbunden sein und die **Google-Dienstkomponente** installiert haben.

Befolgen Sie nun die untenstehenden Schritte.

1. Ändern Sie den AP-Modus der Datei `Zeus_Car.ino` in den STA-Modus.

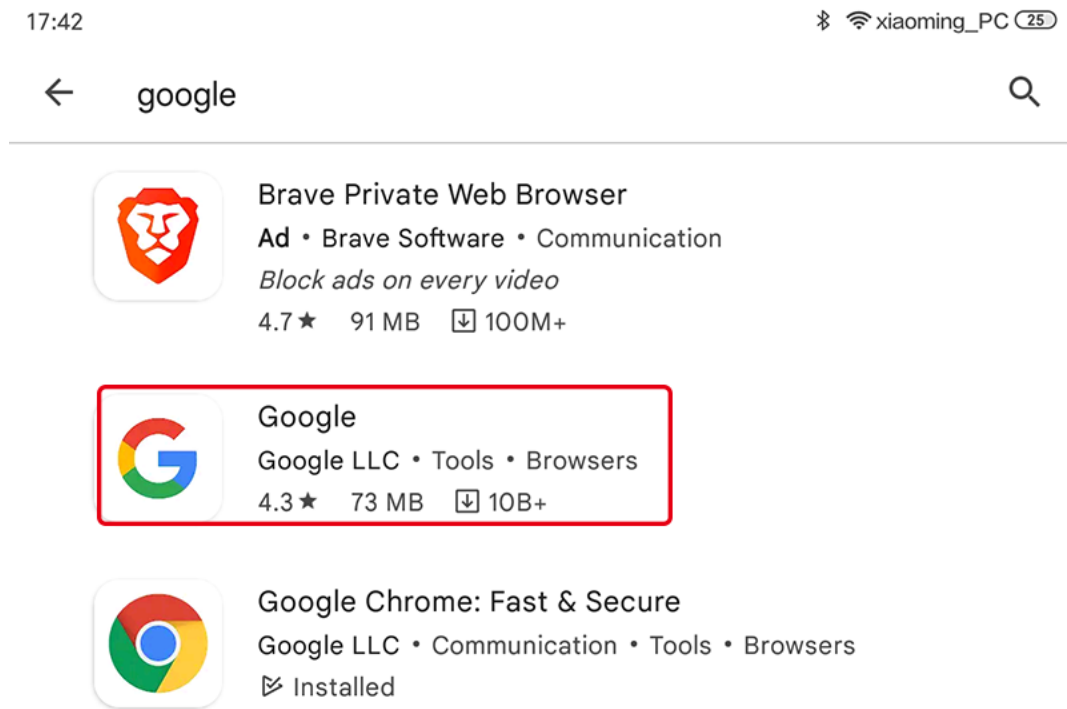
- Öffnen Sie die Datei `Zeus_Car.ino`, die sich im Verzeichnis `zeus-car-main/Zeus_Car` befindet.
- Kommentieren Sie dann den mit dem AP-Modus verbundenen Code aus. Entfernen Sie die Kommentierung für den mit dem STA-Modus verbundenen Code und tragen Sie die SSID und das PASSWORT Ihres Heim-WLANs ein.

```
/** Konfigurieren Sie den WLAN-Modus, SSID, Passwort */
// #define WIFI_MODE WIFI_MODE_AP
// #define SSID "Zeus_Car"
// #define PASSWORD "12345678"

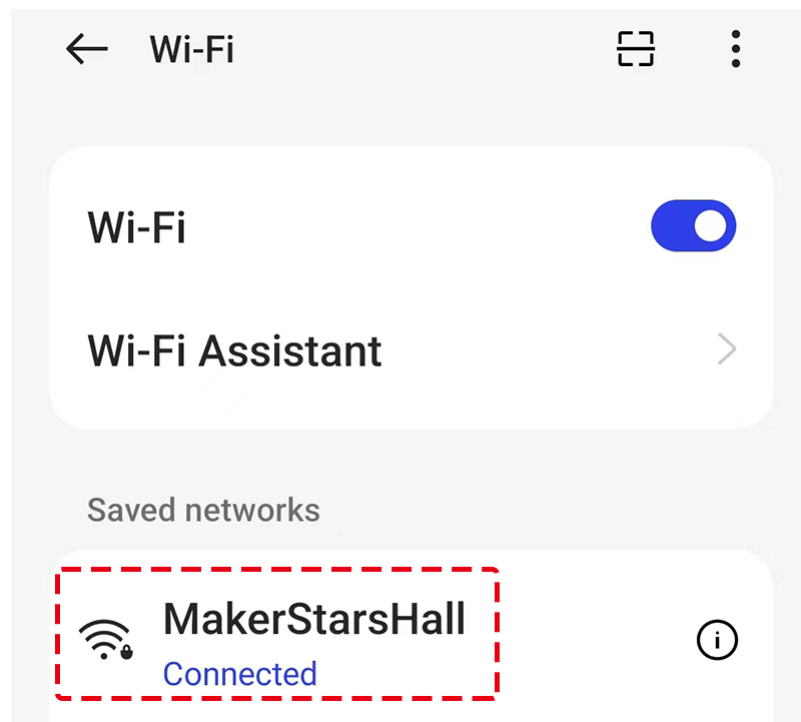
#define WIFI_MODE WIFI_MODE_STA
#define SSID "xxxxxxxxxx"
#define PASSWORD "xxxxxxxxxx"
```

- Speichern Sie diesen Code, wählen Sie das richtige Board (Arduino Uno) und den Port und klicken Sie dann auf die Schaltfläche **Hochladen**, um ihn auf das R3-Board hochzuladen.

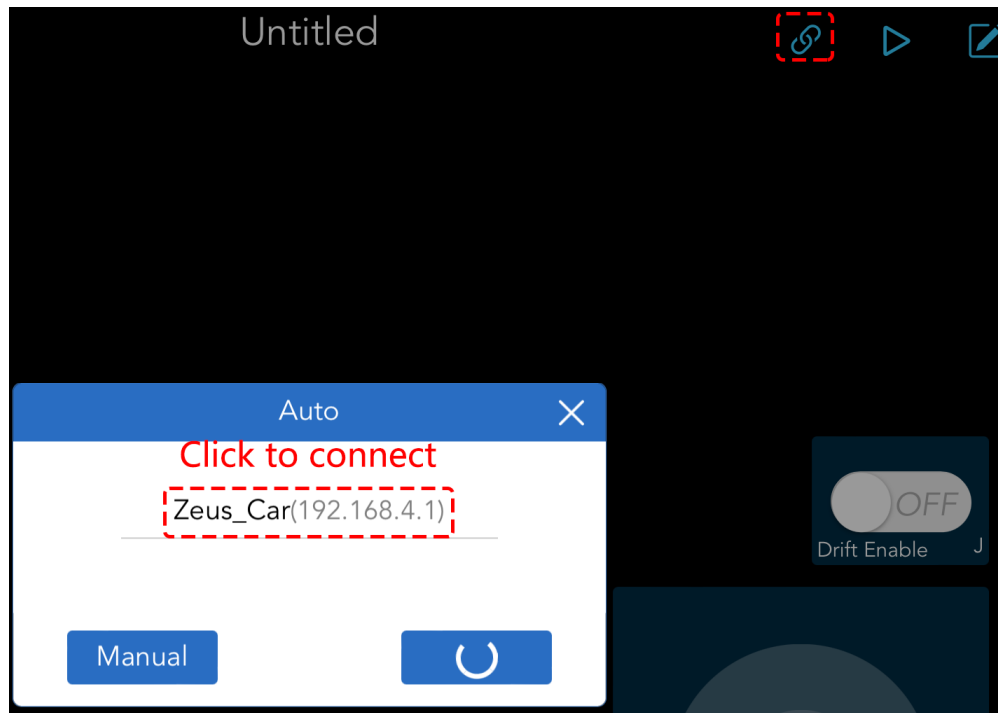
2. Suchen Sie in Google Play nach google und installieren Sie die unten gezeigte App.




3. Verbinden Sie Ihr Mobilgerät mit demselben WLAN, das Sie im Code eingetragen haben.



4. Öffnen Sie den zuvor in SunFounder Controller erstellten Controller und verbinden Sie ihn über die Schaltfläche  mit dem Zeus_Car.



5. Tippen und halten Sie das **Speech** Widget gedrückt, nachdem Sie die Schaltfläche  angeklickt haben. Ein Hinweis wird angezeigt, dass zugehört wird. Sagen Sie den folgenden Befehl, um das Auto zu bewegen.

- stop: Alle Bewegungen des Autos können gestoppt werden.
- pause: Die Funktion ähnelt der von Stop, aber wenn die Vorderseite des Autos nicht in die ursprünglich eingestellte Richtung zeigt, wird es sich langsam in die eingestellte Richtung bewegen.
- forward
- backward
- left forward
- left backward
- right forward
- right backward
- move left
- move right

Urheberrechtshinweis

Alle Inhalte dieses Handbuchs, einschließlich, aber nicht beschränkt auf Texte, Bilder und Code, gehören der SunFounder Company. Sie dürfen es nur für persönliches Lernen, Forschung, Freizeit oder andere nicht-kommerzielle oder gemeinnützige Zwecke verwenden, gemäß den entsprechenden Bestimmungen und Urheberrechtsgesetzen, ohne die gesetzlichen Rechte des Autors und der beteiligten Rechteinhaber zu verletzen. Für jede Person oder Organisation, die dies ohne Erlaubnis zu kommerziellen Zwecken nutzt, behält sich das Unternehmen das Recht vor, rechtliche Schritte einzuleiten.