

---

# **SunFounder GalaxyRVR Kit for Arduino**

***Release 1.0***

**[www.sunfounder.com](http://www.sunfounder.com)**

**27.02.2024**





---

## Inhaltsverzeichnis

---

<b>1</b>	<b>Video-Montage</b>	<b>3</b>
<b>2</b>	<b>Spielmodus</b>	<b>5</b>
2.1	Schnellstartanleitung . . . . .	6
2.2	Avoid(E) . . . . .	10
2.3	Follow(F) . . . . .	11
2.4	STT(J) . . . . .	11
<b>3</b>	<b>Kursmodus</b>	<b>13</b>
3.1	Lektion 1: Enthüllung des Mars-Rovers . . . . .	13
3.2	Lektion 2: Verständnis und Bau des Rocker-Bogie-Systems . . . . .	20
3.3	Lektion 3: Einstieg in die Welt von Arduino und Programmierung . . . . .	23
3.4	Lektion 4: Beherrschung des TT-Motors . . . . .	35
3.5	Lektion 5: Entfesselung der Beweglichkeit des Mars Rovers . . . . .	49
3.6	Lektion 6: Erkundung des Hindernisvermeidungsmoduls . . . . .	54
3.7	Lektion 7: Verbesserung der Rover-Navigation mit Ultraschallmodul . . . . .	62
3.8	Lektion 8 Fortgeschrittene Hindernisvermeidung und intelligentes Folgesystem . . . . .	70
3.9	Lektion 9: Den Weg mit RGB-LED-Streifen beleuchten . . . . .	74
3.10	Lektion 10: Erkundung des visuellen Systems des Mars-Rovers - Servo und Kippmechanismus . . . . .	80
3.11	Lektion 11: Erforschung des visuellen Systems des Mars-Rovers - Kamera und Echtzeitsteuerung . . . . .	87
3.12	Lektion 12: Steuerung des Rovers mit der App . . . . .	100
3.13	Lektion 13: Untersuchung des Energiesystems des Mars-Rovers . . . . .	109
<b>4</b>	<b>Hardware</b>	<b>117</b>
4.1	SunFounder R3 Board . . . . .	118
4.2	GalaxyRVR Shield . . . . .	120
4.3	ESP32 CAM . . . . .	126
4.4	Kamera-Adapterplatine . . . . .	128
4.5	Ultraschallmodul . . . . .	134
4.6	IR-Hindernisvermeidungsmodul . . . . .	135
4.7	4 RGB-LED-Streifen . . . . .	139
4.8	Servo . . . . .	141
4.9	TT-Motor . . . . .	143
4.10	Solarpanel . . . . .	146
4.11	18650 Batterie . . . . .	147
<b>5</b>	<b>FAQ</b>	<b>149</b>

5.1	Q1: Kompilierungsfehler: SoftPWM.h: Datei oder Verzeichnis nicht gefunden? . . . . .	149
5.2	Q2: avrdude: stk500_getsync() Versuch 10 von 10: nicht synchronisiert: resp=0x6e? . . . . .	149
5.3	Q3: Wie kann ich den STT-Modus auf meinem Android-Gerät verwenden? . . . . .	151
5.4	Q4: Über die ESP32 CAM Firmware . . . . .	153
5.5	Q5: Wie wird eine neue Firmware auf einen ESP32 CAM geflasht? . . . . .	153

Danke, dass Sie sich für unser GalaxyRVR entschieden haben.

---

**Bemerkung:** Dieses Dokument ist in den folgenden Sprachen verfügbar.

- 
- 
- 

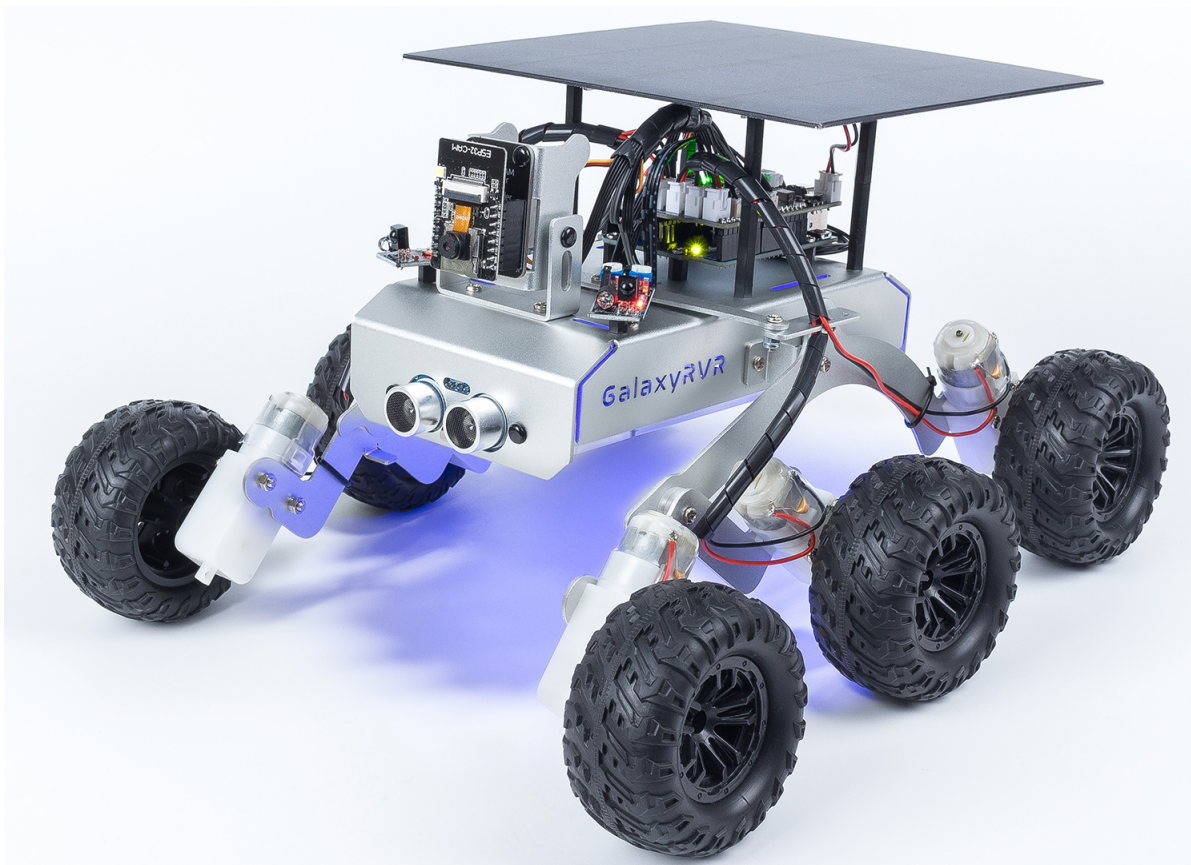
Bitte klicken Sie auf die jeweiligen Links, um das Dokument in Ihrer bevorzugten Sprache aufzurufen.

---

Stellen Sie sich vor, einen Rover auf dem öden Terrain des Mars zu steuern, fremde Landschaften zu erkunden und sich dem Unbekannten zu stellen. Klingt wie ein Traum für NASA-Ingenieure, nicht wahr?

Nicht mehr.

Willkommen in der Welt des GalaxyRVR, einem Mars-Rover-Kit von SunFounder, entworfen, um interplanetare Erkundungen direkt in Ihr Wohnzimmer zu bringen.



Auf dem universellen Rocker-Bogie-System aufgebaut, das von den echten Mars-Rovern der NASA verwendet wird, ist der GalaxyRVR in der Lage, alle Arten von Gelände zu überqueren – seien es felsige Berge, sandige Strände oder grasige Felder. Das Federungssystem sorgt für eine reibungslose Mobilität und macht die fremde Welt des Mars so heimisch wie Ihre eigene.

Ausgestattet mit einem hochauflösenden Kameramodul, bietet der GalaxyRVR eine First-Person-Ansicht und liefert ein immersives Pilotenerlebnis, als ob Sie tatsächlich den Roten Planeten navigieren. Sein fortschrittliches Hindernisvermeidungsmodul und Ultraschallmodul sorgen dafür, dass er Hindernissen ausweichen kann, und verkörpert so einen Geist der Autonomie und Widerstandsfähigkeit.

### Eintauchen in den Kosmos: Die Kurse

- *Spielmodus*: Möchten Sie schnell starten? Der Spielmodus ist für diejenigen gedacht, die es kaum erwarten können, ihre Reise zum Mars zu beginnen. Mit vorinstalliertem Code können Sie einfach Ihren GalaxyRVR zusammenbauen und ihn mit der praktischen App steuern, um seine vielfältigen Funktionen zu erkunden.
- *Kursmodus*: Interessieren Sie sich für die zugrundeliegende Technologie? Der Kursmodus ist für diejenigen, die tiefer eintauchen wollen. Verstehen Sie die Prinzipien hinter dem Design des GalaxyRVR, lernen Sie zu programmieren und bereichern Sie Ihren Rover mit einer Vielzahl von spannenden Funktionen.

Diese Dokumentation ist Ihr Leitfaden, um die Feinheiten des GalaxyRVR zu erforschen. Sie enthält detaillierte Montageanleitungen, Programmieranleitungen und Einblicke in die Arbeitsprinzipien eines Mars-Rovers, alles darauf ausgerichtet, Ihre Neugier zu wecken und Kreativität zu fördern.

Der GalaxyRVR ist nicht nur ein Spielzeug, sondern ein Tor zum Mars, ein Bildungswerkzeug und ein Katalysator für grenzenlose Vorstellungskraft. Beginnen Sie Ihre kosmische Reise heute mit dem GalaxyRVR und lassen Sie Ihre Welt eine Erweiterung des Mars werden.

---

## Video-Montage

---

### Überblick

Wenn Sie beim Zusammenbau der verschiedenen Komponenten lernen möchten, können Sie sich auf *Kursmodus* beziehen. Im Kursmodus kommt jede Lektion mit Montagevideos, Komponentenprinzipien und zugehörigem Testcode, sodass Sie während des Lernens fortfahren können. Daher können Sie dieses Kapitel überspringen.

Wenn Sie schnell zusammenbauen und sofort spielen möchten, können Sie sich zuerst das Montagevideo ansehen. Nach Abschluss der Montage fahren Sie mit *Spielmodus* fort, um mehr über die App-Installation und -Verbindung zu erfahren. Anschließend können Sie die App verwenden, um den GalaxyRVR zu steuern.

### Videos

- Für die Montage des GalaxyRVR bieten wir gedruckte Montageanleitungen im Kit an. Hier ist die PDF-Version davon:
  - Teileliste und Montageanleitungen
- Wenn Sie Schwierigkeiten haben, die gedruckten Montageanleitungen zu verstehen, können Sie stattdessen unserem schrittweisen Montagevideo folgen.

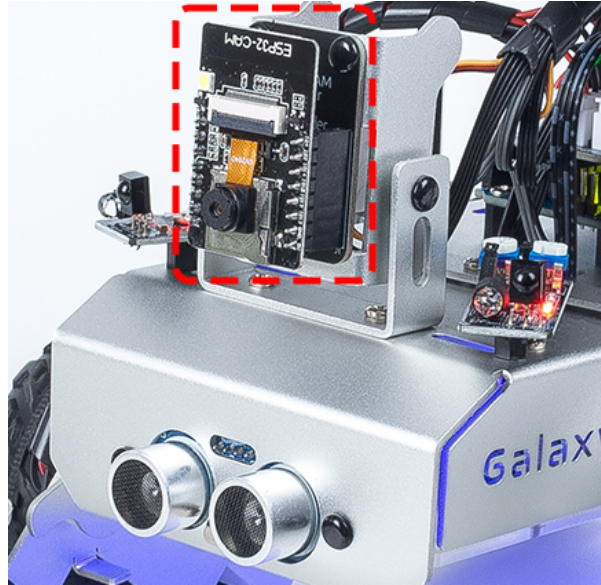
---

#### Bemerkung:

- Die Montagevideos sind eine Serie von 7.
  - Sie können andere Videos aus der Wiedergabeliste in der oberen rechten Ecke überprüfen oder sie automatisch abspielen lassen.
- 

### Tipps

- Achten Sie bei der Einführung der ESP32 CAM in den Kameraadapter auf die Ausrichtung. Sie sollte richtig mit dem ESP32-Adapter ausgerichtet sein.



- Nach der Montage können Sie die Motorleitungen mit Kabelbindern am Teller befestigen, um zu verhindern, dass sie sich in den Rädern verheddern.

## KAPITEL 2

---

### Spielmodus

---

Sind Sie bereit, Ihre Marsreise sofort zu beginnen? Unser Spielmodus ist perfekt für diejenigen unter Ihnen zugeschnitten, die es kaum erwarten können, einzutauchen. Ausgestattet mit werkseitig vorinstalliertem Code, können Sie direkt nach dem Zusammenbau Ihres GalaxyRVR in das Geschehen einsteigen. Nutzen Sie die benutzerfreundliche Sun-Founder Controller-App, um seine zahlreichen Funktionen zu erkunden, einschließlich Fahren aus der Ich-Perspektive, Umschalten zwischen Hindernisvermeidungs- und Folgemodus und sogar Sprachsteuerung.

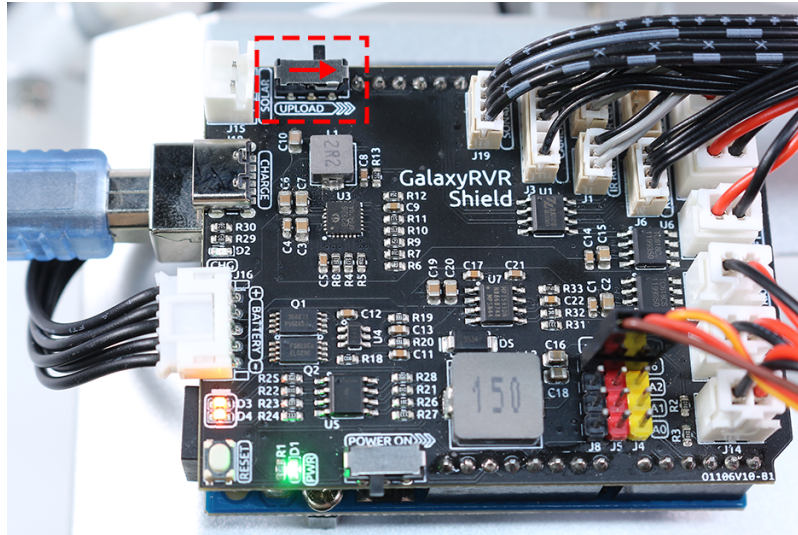
Aber bevor Sie den Roten Planeten erkunden, stellen wir sicher, dass Sie voll ausgestattet und bereit für das Abenteuer sind. Unten finden Sie eine Schnellstartanleitung, die Sie auf dieser spannenden Reise unterstützt!

---

#### **Bemerkung:**

- Wenn Ihr R3-Board bereits mit anderem Code hochgeladen wurde, aber Sie weiterhin den Spielmodus verwenden möchten, müssen Sie den relevanten Code herunterladen.
  - GalaxyRVR Codes
- Öffnen Sie dann die Datei `galaxy-rvr.ino`, die sich im Verzeichnis `galaxy-rvr-main\galaxy-rvr` befindet.
- Bewegen Sie den Schalter nach rechts und klicken Sie dann auf **Upload**.
- *Q1: Kompilierungsfehler: SoftPWM.h: Datei oder Verzeichnis nicht gefunden?*





## 2.1 Schnellstartanleitung

1. Lassen Sie uns den GalaxyRVR starten.

- Beim ersten Gebrauch Ihres GalaxyRVR wird empfohlen, zuerst ein Typ-C-USB-Kabel anzuschließen, um den Akku vollständig aufzuladen. Schalten Sie dann die Stromversorgung ein.
- Um die ESP32 CAM zu aktivieren, bewegen Sie den Modusschalter in die Position **Run**, und drücken Sie die **reset**-Taste, um das R3-Board neu zu starten. Sie werden dann ein zyanfarbenedes Licht am unteren Streifen blinken sehen.

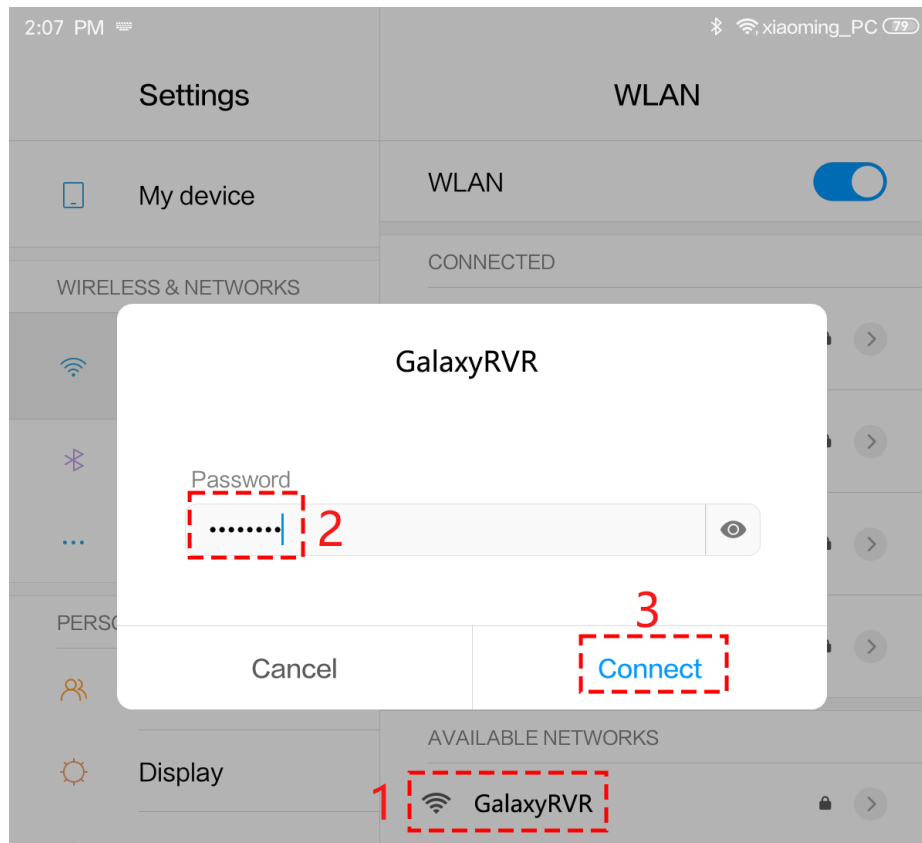
2. Installieren Sie [SunFounder Controller](#) aus dem **APP Store(iOS)** oder **Google Play(Android)**.

3. Verbinden Sie sich mit dem Netzwerk GalaxyRVR.

Für eine optimale Kommunikation zwischen Ihrem mobilen Gerät und dem Rover müssen Sie beide mit demselben lokalen Netzwerk (LAN) verbinden, das von GalaxyRVR bereitgestellt wird.

- Finden Sie GalaxyRVR in der Liste der verfügbaren Netzwerke auf Ihrem mobilen Gerät (Tablet oder Smartphone), geben Sie das Passwort 12345678 ein und verbinden Sie sich damit.





- Der Standardverbindungsmodus ist **AP mode**. Nachdem Sie sich verbunden haben, könnte eine Warnung erscheinen, dass dieses Netzwerk keinen Internetzugang hat. Wenn ja, wählen Sie „Stay connected“.

## Confirm

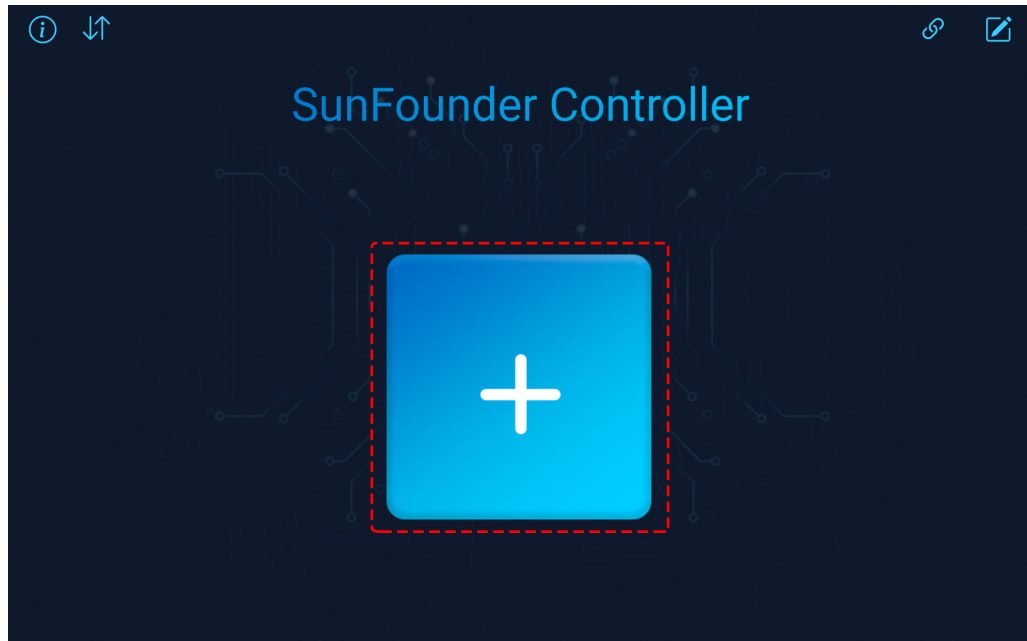
This WLAN network has no access to the internet. Other WLAN networks are available. Switch networks?

Stay connected

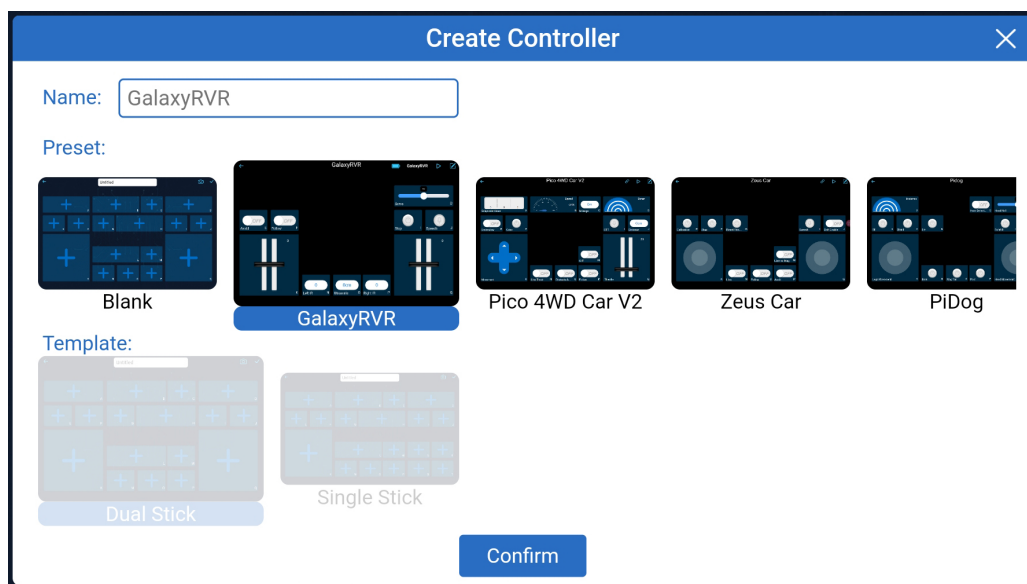
Switch networks

### 4. Einrichtung eines Controllers.

- Um einen Controller in der SunFounder Controller App zu erstellen, tippen Sie auf das + Symbol.




- Voreingestellte Controller sind für einige Produkte verfügbar, hier wählen wir **GalaxyRVR**. Geben Sie ihm einen Namen oder tippen Sie einfach auf **Confirm**.

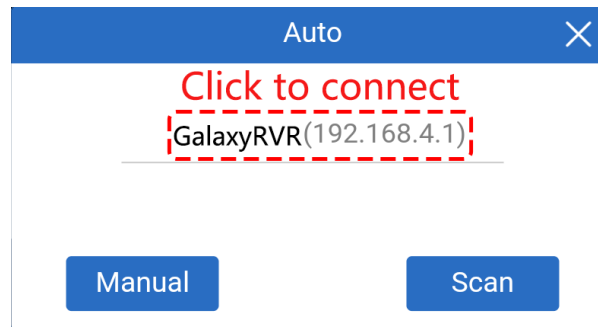


- Sobald Sie in der App sind, sucht diese automatisch nach dem GalaxyRVR. Nach einem Moment erscheint eine Meldung „Connected Successfully“.



**Bemerkung:**

- Wenn keine Verbindung besteht, überprüfen Sie bitte, ob Ihr WLAN mit GalaxyRVR verbunden ist.
- Sie können auch den  Button antippen, um manuell zu verbinden. Nach kurzer Wartezeit sollte GalaxyRVR(IP) erscheinen. Tippen Sie darauf, um eine Verbindung herzustellen.



- Jetzt aktivieren Sie mit dem  Button die Live-Videoübertragung von der Kamera und steuern das Fahrzeug mit den bereitgestellten Widgets.



5. Lassen Sie uns nun die Funktionen jedes Widgets betrachten:

- **Servo(D)**: Steuert den Neigungsmechanismus der Kamera des Rovers, wodurch Sie einen größeren Bereich beobachten können.
- **Avoid(E)**: Schaltet den Rover in den Hindernisvermeidungsmodus. Die werkseitig eingestellten Erkennungsentfernungen für jedes Hindernisvermeidungsmodul können unterschiedlich sein. Eine manuelle Anpassung ist erforderlich.
- **Follow(F)**: Schaltet den Rover in den Verfolgungsmodus. Die werkseitig eingestellten Erkennungsentfernungen für jedes Hindernisvermeidungsmodul können unterschiedlich sein. Eine manuelle Anpassung ist erforderlich.
- **Stop(I)**: Stoppt sofort alle Bewegungen des Rovers.
- **STT(J)**: Drücken Sie dieses Widget, um die Spracherkennung zu starten und den Rover entsprechende Aktionen ausführen zu lassen.
- **HeadLamp(M)**: Um die LED am ESP32 CAM ein-/auszuschalten.
- **Throttle Widgets on K and Q**: Das Gashebel-Widget im Bereich K steuert die linken Motoren des Rovers, während das im Bereich Q die rechten Motoren steuert. Durch die Koordination beider Widgets kann der GalaxyRVR in jede Richtung bewegt werden.
- **Left IR(N)**: Zeigt die Messwerte des linken Hindernisvermeidungsmoduls an.
- **Ultrasonic(O)**: Zeigt die vom Ultraschallmodul gemessene Entfernung an.
- **Right IR(P)**: Zeigt die Messwerte des rechten Hindernisvermeidungsmoduls an.

## 2.2 Avoid(E)

Tippen Sie auf das Widget **Avoid(E)**, um den Hindernisvermeidungsmodus zu aktivieren.

Bevor Sie diesen Modus aktivieren, müssen Sie möglicherweise die Erkennungsbereiche der Sensoren entsprechend Ihrer aktuellen Umgebung anpassen, da die Werkseinstellungen nicht für alle Situationen ideal sind.

Wenn der Erkennungsbereich der beiden Infrarotmodule zu kurz ist, könnte der Mars Rover gegen Hindernisse stoßen. Ist der Bereich hingegen zu lang, könnte der Rover beginnen, zu weit von einem Hindernis wegzuschwenken, was seine Navigation stören könnte.

So können Sie die Einstellungen feinjustieren:

1. Beginnen Sie mit der Anpassung des rechten Hindernisvermeidungsmoduls. Während des Transports können Kollisionen dazu führen, dass der Sender und Empfänger am Infrarotmodul kippen. Daher müssen Sie diese manuell begradigen.
2. Stellen Sie ein Hindernis etwa 20 cm direkt vor dem rechten Modul auf. Die Box, in der unser Rover-Kit geliefert wurde, ist dafür eine gute Wahl! Drehen Sie nun das Potentiometer am Modul, bis die Anzeigeleuchte am Modul gerade aufleuchtet. Bewegen Sie dann das Hindernis vor und zurück, um zu überprüfen, ob die Anzeigeleuchte in der gewünschten Entfernung aufleuchtet. Leuchtet das Licht nicht in der richtigen Entfernung auf oder bleibt es an, ohne auszugehen, müssen Sie das andere Potentiometer einstellen.
3. Wiederholen Sie den gleichen Vorgang für das andere Modul.

## 2.3 Follow(F)

Wenn Sie bereit sind, den Folgemodus zu aktivieren, tippen Sie einfach auf das Widget **Follow(F)**. Wenn Sie zuvor den Erkennungsabstand der Hindernisvermeidungsmodule nicht angepasst haben, müssen Sie zuerst die Schritte in *Avoid(E)* befolgen.

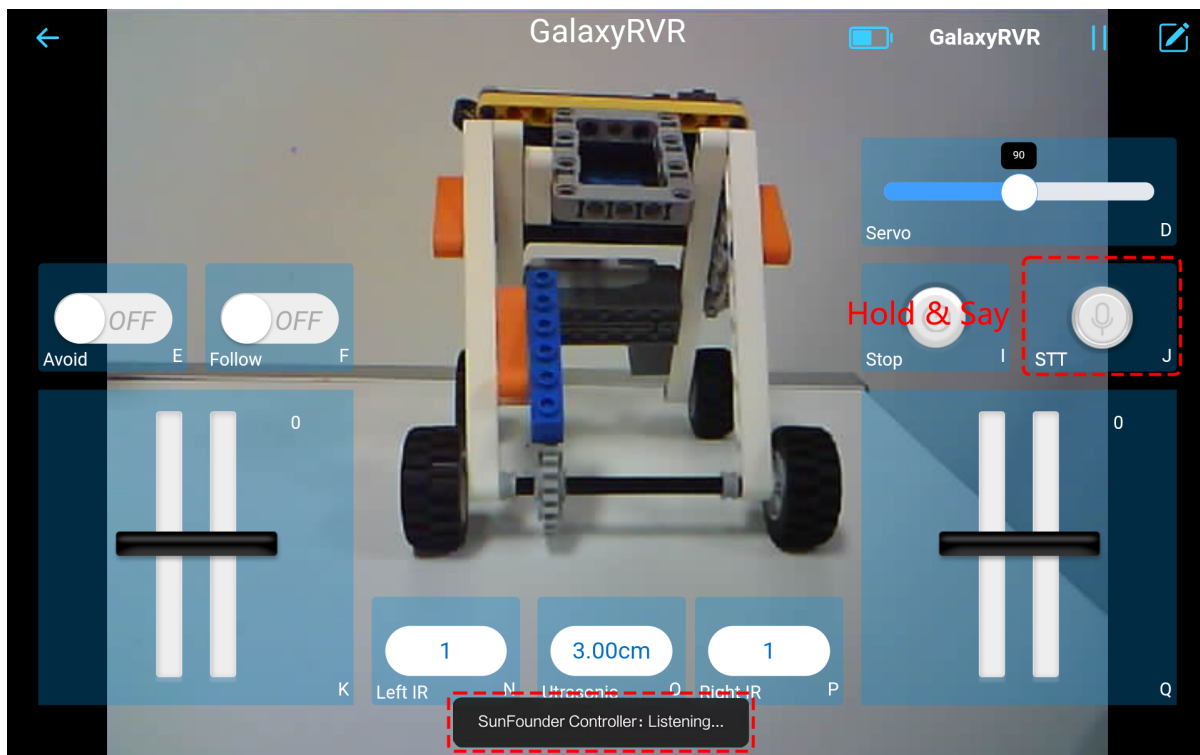
Im Folgemodus bewegt sich der GalaxyRVR auf ein Objekt vor ihm zu oder dreht sich nach links oder rechts, um der Bewegung des Objekts zu folgen.

## 2.4 STT(J)

STT steht für Sprache-zu-Text. Die SunFounder Controller-App integriert sich mit dem Spracherkennungsmotor Ihres mobilen Geräts. Wenn Sie also das Widget **STT(J)** im SunFounder Controller antippen und in Ihr mobiles Gerät sprechen,

wird Ihr Gerät Ihre Sprache erfassen, in Text umwandeln und an den GalaxyRVR senden. Stimmt dieser Text mit den im Code voreingestellten Befehlen überein, führt der Rover die entsprechenden Aktionen aus.

Die folgenden Befehle sind derzeit im Code voreingestellt. Sprechen Sie einen der folgenden Befehle und beobachten Sie, wie der GalaxyRVR reagiert.



- **stop:** Alle Bewegungen des Rovers können gestoppt werden.
- **forward:** Lassen Sie den Rover vorwärts fahren.
- **backward:** Lassen Sie den Rover rückwärts fahren.
- **left:** Lassen Sie den Rover nach links abbiegen.
- **right:** Lassen Sie den Rover nach rechts abbiegen.

---

**Bemerkung:** Die STT (Sprache-zu-Text)-Funktion erfordert eine Internetverbindung und Google-Dienste auf Android-Geräten. Diese funktioniert jedoch nicht mit dem voreingestellten AP-Modus (Access Point) auf dem GalaxyRVR.

Im AP-Modus erstellt der GalaxyRVR ein lokales WLAN-Netzwerk, mit dem sich Ihr mobiles Gerät verbinden kann, aber es bietet keinen Internetzugang.

Um die STT-Funktion auf Android zu nutzen, wechseln Sie den Code des Rovers von AP auf STA-Modus, wie in [Q3: Wie kann ich den STT-Modus auf meinem Android-Gerät verwenden?](#) beschrieben.

---

---

**Bemerkung:** iOS-Geräte, die einen Offline-Spracherkennungsmotor verwenden, funktionieren sowohl im AP- als auch im STA-Modus einwandfrei.

---

Willkommen im Kursmodus, einer strukturierten Reise durch 13 aufschlussreiche Lektionen. Wir beginnen mit einem faszinierenden Einblick in die Geschichte der Mars-Rover und legen damit den Grundstein für unseren eigenen Roverbau.

Ab der zweiten Lektion werden wir praktisch. Jede Lektion konzentriert sich auf eine einzigartige Komponente, die für unseren Mars-Rover entscheidend ist. Du wirst nicht nur ihren Zweck und ihre Bedeutung verstehen, sondern auch deren Montage und Integration in den Rover meistern.

Ob Du ein Anfänger bist, der in die Welt der Mars-Rover eintauchen möchte, oder ein erfahrener Hobbyist, der seine Fähigkeiten verfeinern will, diese Lektionen sind so gestaltet, dass sie ein umfassendes Verständnis der Mechanik, Elektronik und Software vermitteln, die unseren GalaxyRVR zum Leben erwecken. Am Ende des Kurses wirst Du nicht nur deinen eigenen Mars-Rover zusammengebaut haben, sondern auch das Wissen besitzen, um Probleme zu beheben, Änderungen vorzunehmen und weiter mit deinem Rover zu experimentieren.

Also, schnall Dich an für eine spannende Reise in die Welt der Mars-Rover! Tauche ein und beginne, den roten Planeten bequem von zu Hause aus zu erkunden.

### 3.1 Lektion 1: Enthüllung des Mars-Rovers

Willkommen zu Lektion 1: Verständnis des Mars-Rovers. Heute tauchen wir in die aufregende Welt der Mars-Rover ein – unsere entfernten Erforscher des Roten Planeten. Wir werden ihre Entwicklung, ihre Funktionen und die technologischen Wunder, die sie darstellen, kennenlernen. Darüber hinaus wirst du deine Kreativität nutzen, um deinen eigenen Rover zu entwerfen und deine Präsentationsfähigkeiten zu schärfen, indem du dein einzigartiges Design erklärst. Bereite dich darauf vor, den Mars von deinem Klassenzimmer aus zu erkunden!

### 3.1.1 Kursziele

- Ein Verständnis für die Entwicklung und den Zweck von Mars-Rovern gewinnen
- Kreativität ausdrücken, indem du deinen eigenen Mars-Rover entwirfst
- Präsentationsfähigkeiten verbessern, indem du dein Rover-Design teilst und erklärst

### 3.1.2 Kursmaterialien

- Bilder und technische Spezifikationen von Mars-Rovern als Referenz
- Dokumentarfilm über die Geschichte der Mars-Rover
- Computer mit Internetzugang für Recherche und das Ansehen des Dokumentarfilms
- Präsentationsfolien oder interaktives Whiteboard für die Unterrichtsgestaltung
- Zeichenpapier, Bleistifte und Farbmateriale für die Rover-Design-Aktivität
- Arbeitsblätter für angeleitetes Notieren, Reflexion und Designplanung

### 3.1.3 Kursschritte

#### Schritt 1: Was sind Mars-Rover?

Bevor wir in die Welt der Mars-Rover eintauchen, machen wir uns zuerst mit dem Mars selbst vertraut. Wie wir anhand der Bilder und Modelle sehen können, ist die Oberfläche des Mars von Kratern, Bergen, Tälern und Staubstürmen gezeichnet, was ein faszinierendes und herausforderndes Landschaftsbild ergibt.







Kannst du dir vorstellen, wie es wäre, durch ein solch unwegsames Gelände zu navigieren? Nun stell dir vor, du hast die Aufgabe, einen Rover für den Mars zu entwerfen.

- Welche Überlegungen wirst du angesichts des Geländes und der Bedingungen auf dem Mars anstellen?
- Mit welchen Funktionen wirst du ihn ausstatten, um sicherzustellen, dass er seine Aufgaben effektiv erfüllen kann?
- Welche Aufgaben stellst du dir vor, sollte dein Mars-Rover bewältigen?

Denke daran, ein Mars-Rover ist ein Roboter, der dazu bestimmt ist, den Mars zu erforschen, seine Umgebung zu studieren und Daten zurück zur Erde zu senden. Also denke über Aspekte wie Bewegung, Kommunikation, Energieversorgung, wissenschaftliche Forschungsmöglichkeiten und Haltbarkeit unter den extremen Bedingungen des Mars nach.

Lassen wir uns einen Moment Zeit, um zu brainstormen und unsere Ideen zu teilen. Es ist interessant, wie Ingenieure und Wissenschaftler zu denken, nicht wahr? Wir werden uns in den folgenden Schritten näher mit den tatsächlichen Mars-Rover-Designs und ihren Funktionen befassen, also behalte deine kreativen Ideen im Hinterkopf, während wir weitermachen.

## **Schritt 2: Erkundung der Geschichte der Mars-Rover**

Als Nächstes begeben wir uns auf eine Zeitreise, indem wir einen Dokumentarfilm ansehen, der die Geschichte der Mars-Rover detailliert darstellt. Der Film führt uns vom ersten Versuch, einen Rover auf dem Mars zu platzieren, dem sowjetischen Mars 3 Rover, der leider 1971 bei der Landung scheiterte, bis zum ersten erfolgreichen Mars-Rover der NASA, Sojourner, im Jahr 1997.

Unsere Reise endet hier jedoch nicht, denn wir wagen uns weiter, um die Abenteuer der fortschrittlichsten Rover zu verstehen: Spirit, Opportunity, Curiosity und Perseverance.

Dieser Dokumentarfilm bietet nicht nur einen historischen Kontext, sondern vermittelt auch ein umfassendes Verständnis der fortschreitenden wissenschaftlichen und technischen Meilensteine, die zur aktuellen Ära der Mars-Erkundung geführt haben.

## **Schritt 3: Zusammenfassung der Mars-Rover**

Nachdem wir den Dokumentarfilm gesehen haben, fassen wir die verschiedenen Mars-Rover zusammen, die auf den roten Planeten geschickt wurden.

- **Sojourner** (1997)

**Sojourner**, der Pionier der Mars-Rover, begann seine Reise als Teil der Mars Pathfinder-Mission. Er landete erfolgreich in der Region Ares Vallis am 4. Juli 1997. Als erstes Radfahrzeug, das auf einem anderen Planeten als der Erde fuhr, markierte Sojourner einen bedeutenden Meilenstein in der

Erforschung des Mars. Obwohl er nur 92 Marstage, oder Sols, auf dem Mars operierte, legte er den Grundstein für zukünftige Forschungsrover.

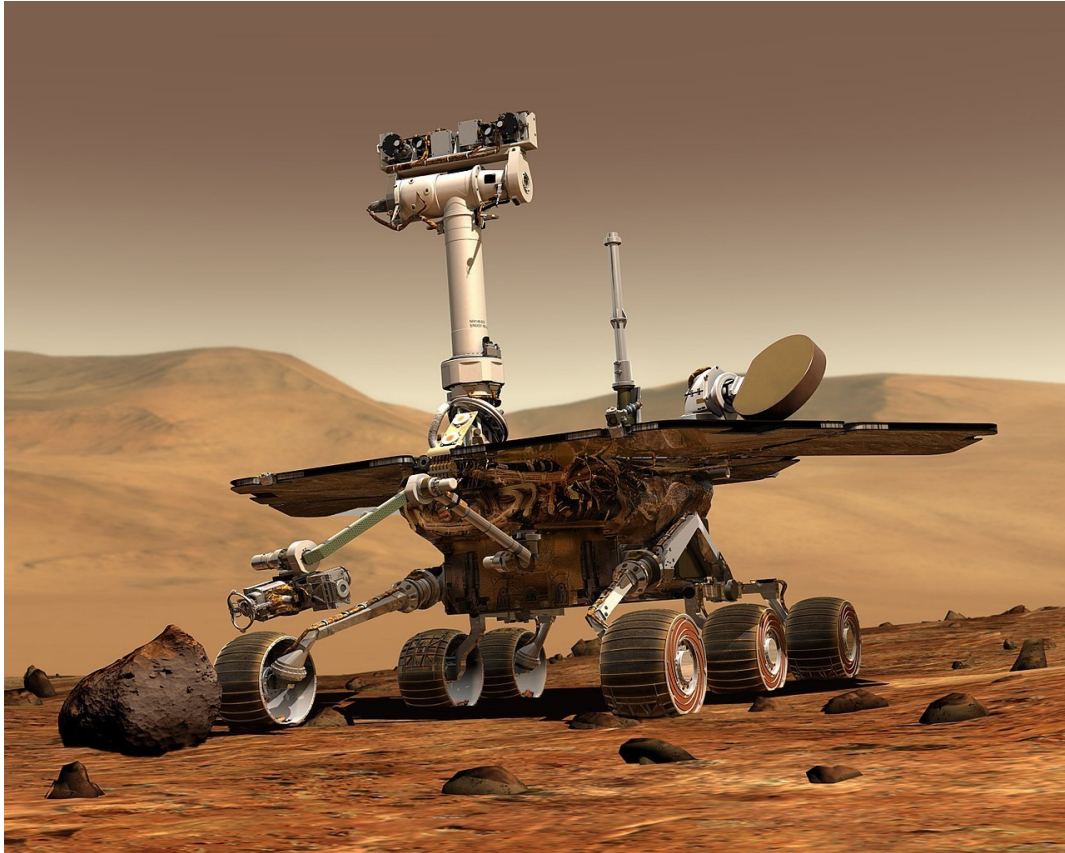


- **Spirit** (2004–2010) und **Opportunity** (2004–2018)

**Spirit** und **Opportunity** sind Zwillingss Rover der NASA-Mars-Explorations-Rover-Mission (MER). **Spirit**, auch bekannt als MER-A, war von 2004 bis 2010 auf dem Mars aktiv.

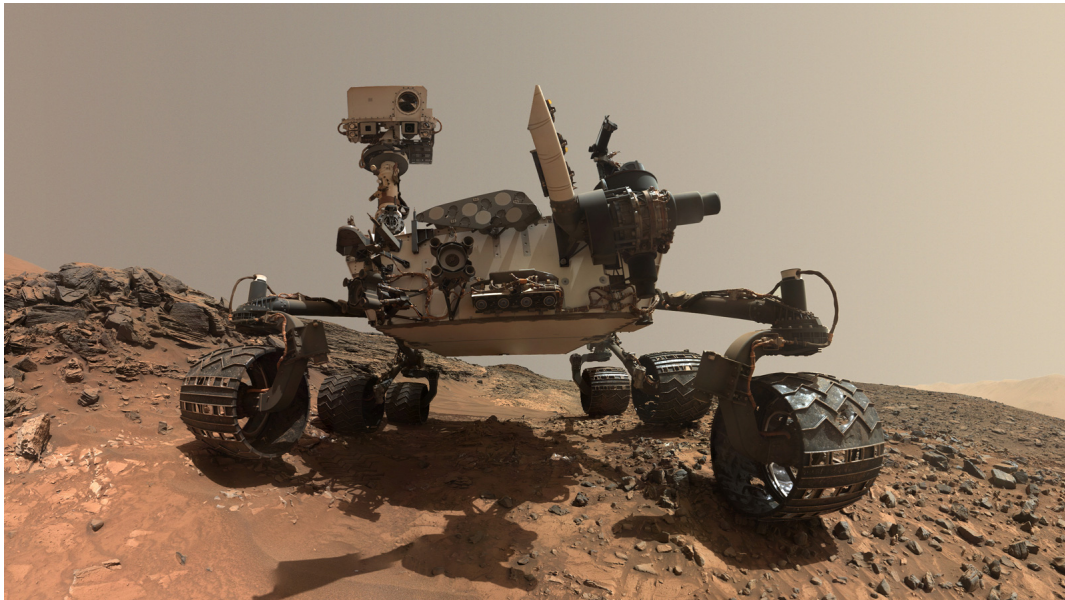
Andererseits hatte **Opportunity**, oder MER-B, einen bemerkenswert langen Lauf von 2004 bis 2018. Zusammen erweiterten sie unser Verständnis der Marsoberfläche und der geologischen Geschichte erheblich.





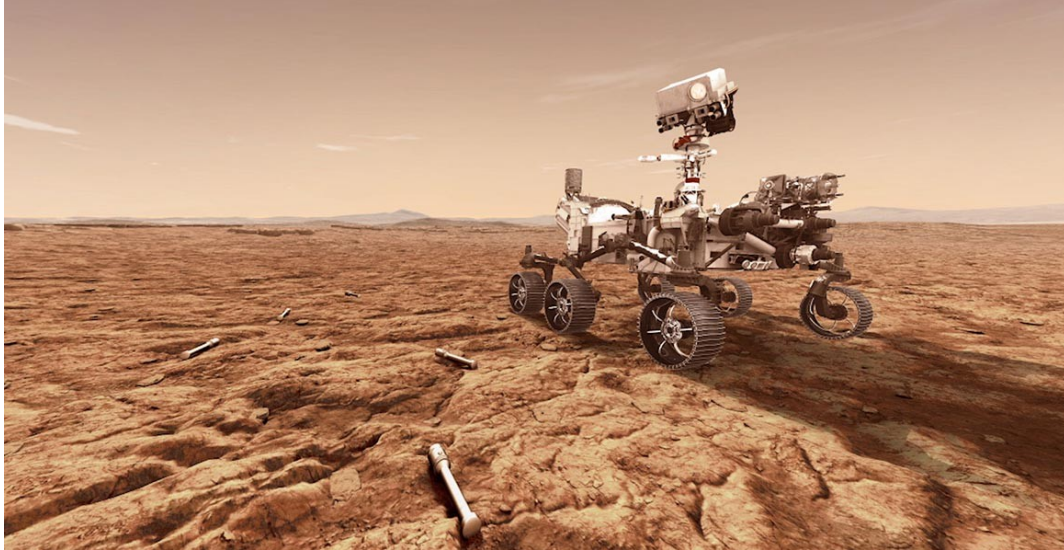
- **Curiosity** (2012–heute):

**Curiosity**, ein auto-großer Mars-Rover, wurde entwickelt, um den Gale-Krater auf dem Mars als Teil der Mars Science Laboratory (MSL)-Mission der NASA zu erkunden. Seit seiner Ankunft im Jahr 2012 hat **Curiosity** zahlreiche bedeutende Entdeckungen gemacht, darunter Hinweise auf früheres flüssiges Wasser auf dem Mars.



- **Perseverance** (2021–heute):

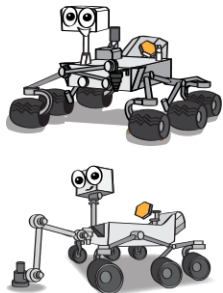
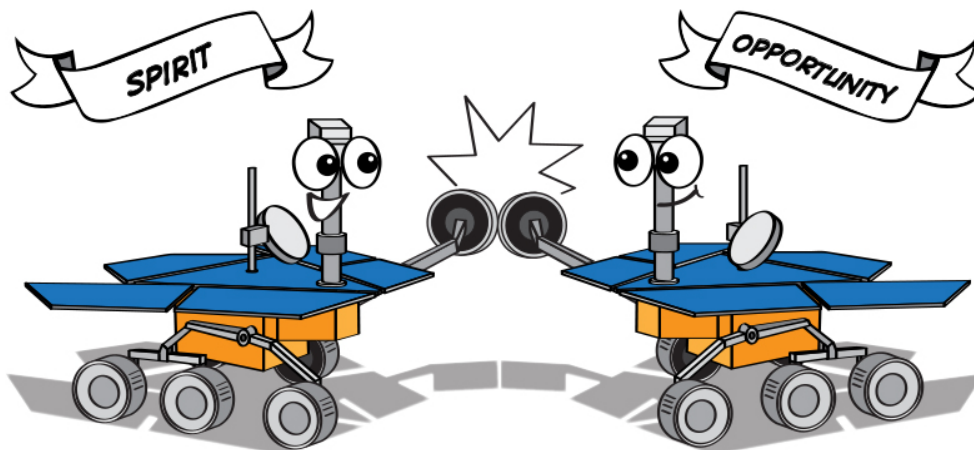
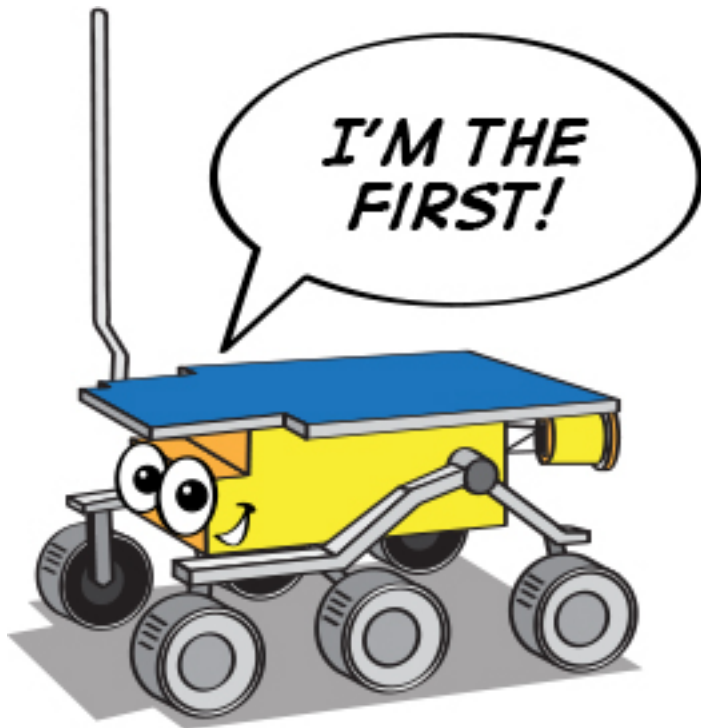
**Perseverance**, auch bekannt als Percy, ist der jüngste Rover, der auf dem Mars angekommen ist. Er wurde entworfen, um den Jezero-Krater als Teil der Mars 2020-Mission der NASA zu erforschen. Neben seinen wissenschaftlichen Instrumenten trägt **Perseverance** auch Ingenuity, einen kleinen experimentellen Mars-Hubschrauber, und markiert damit ein weiteres Novum in der Erforschung des Mars.



Jetzt lasst uns diskutieren. Denkt über die Entwicklung dieser Rover nach.

- Worin unterscheiden sich die Designs dieser Rover? Inwiefern sind sie ähnlich?
- Wie haben die Missionsziele das Design jedes Rovers beeinflusst?
- Welche technologischen Fortschritte könnt ihr zwischen den einzelnen Rovern erkennen?
- Welche Funktionen sollte der nächste Mars-Rover eurer Meinung nach haben?
- Teilt eure Gedanken und Reflexionen sowie etwaige Fragen, die ihr haben könntet!

**Schritt 4: Kunstaktivität: Zeichne deinen eigenen Mars-Rover**



Für unsere nächste Aktivität setzen wir unser Wissen und unsere Kreativität ein, um unseren eigenen Mars-Rover zu entwerfen. Berücksichtige die wichtigsten Merkmale der bisher studierten Rover, denke aber auch über die einzigartigen Eigenschaften nach, die du in deinem Design einbringen möchtest.

Materialien, die du benötigst:

- Zeichenpapier
- Bleistifte und Radiergummis
- Buntstifte, Wachsmalstifte oder Filzstifte

Zeichenanleitung:

1. Beginne mit dem Körper des Rovers. Welche Form wird er haben? Wie groß wird er sein?
2. Betrachte die Räder. Wie viele wird dein Rover haben? Welche Größe und Form werden sie haben?
3. Vergiss nicht die Instrumente. Welche wissenschaftliche Ausrüstung wird dein Rover tragen? Kameras, Bohrer, Spektrometer oder etwas völlig Neues?
4. Überlege dir zuletzt besondere Merkmale. Hat dein Rover Solarpanele oder nutzt er eine andere Energiequelle? Kann er direkt mit der Erde kommunizieren oder benötigt er einen Relaisatelliten?

Sobald alle ihre Zeichnungen fertiggestellt haben, teilen wir sie mit der Klasse. Erkläre deine Designentscheidungen und die Mission, die du dir für deinen Rover vorstellst.

### **Schritt 5: Präsentiere deine Mars-Rover-Entwürfe**

Jetzt, wo jeder seine Mars-Rover-Zeichnungen fertiggestellt hat, ist es an der Zeit, sie zu teilen! Während du präsentierst, besprich den Gedankenprozess hinter deinem Design. Was ist die Mission deines Rovers? Wie unterstützt das Design diese Mission?

Denke daran, in dieser Aktivität gibt es keine falschen Antworten. Der Zweck ist es, deine Kreativität anzuregen und dein Verständnis für die Technologie der Mars-Rover zu vertiefen.

### **Schritt 6: Reflexion und Schlussfolgerung**

Während wir unsere Mars-Rover-Lektion abschließen, nehmen wir uns ein paar Minuten Zeit, um über das Gelernte nachzudenken. Wie spiegeln unsere Rover-Designs die Fortschritte in der Technologie und die wissenschaftlichen Ziele wider? Wie könnten sich die echten Mars-Rover in Zukunft weiterentwickeln?

Denke daran, die Erforschung des Weltraums, wie jedes STEAM-Feld, dreht sich darum, Fragen zu stellen, Probleme zu lösen und Kreativität zu nutzen. Bleibe am Forschen, stelle Fragen und sei neugierig!

## **3.2 Lektion 2: Verständnis und Bau des Rocker-Bogie-Systems**

In unserer vorherigen Lektion haben wir über die Mars-Rover und ihre grundlegende Struktur gelernt. Ein interessanter Aspekt, den wir bei der Betrachtung der Evolution der Mars-Rover feststellen, ist die Konstanz in ihrem Federungssystem. Trotz des technologischen Fortschritts wurden alle Rover von Sojourner bis Perseverance mit einem ähnlichen Typ von Federungssystem entworfen, das als Rocker-Bogie-System bekannt ist.

Aber warum halten sie am Rocker-Bogie-System fest, fragt man sich vielleicht? Welche Vorteile bietet dieses spezielle Design für die Mars-Erkundung?

In der heutigen Lektion werden wir tiefer in die Wissenschaft und Technik hinter dem Rocker-Bogie-System eintauchen und dann eines bauen.

Lasst uns auf diese spannende Ingenieurreise gehen!



### 3.2.1 Kursziele

- Das Designprinzip des Rocker-Bogie-Federungssystems und seine Vorteile verstehen.
- Lernen, wie man ein Grundmodell des Rocker-Bogie-Federungssystems entwirft und baut.
- Grundlegende Physikprinzipien anwenden, um zu erklären, wie das Rocker-Bogie-Federungssystem komplexe Gelände meistert.

### 3.2.2 Kursmaterialien

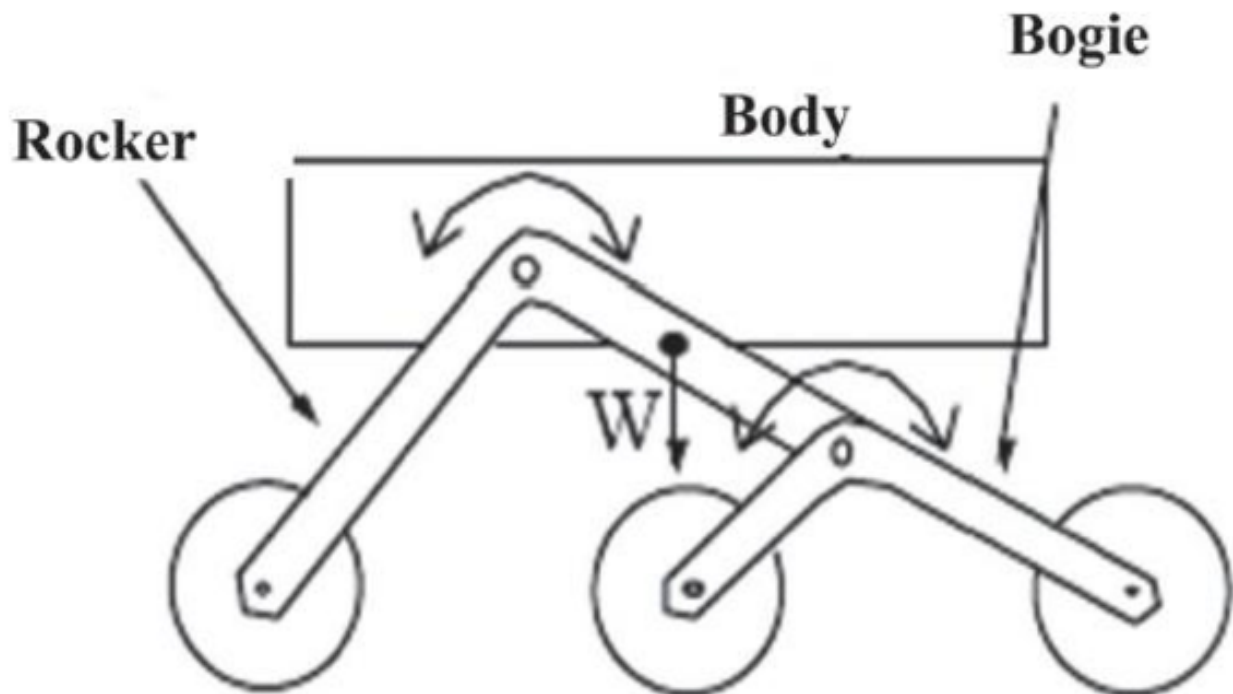
- Blaupausen und Referenzmaterialien (wie NASA Mars Rover-Designzeichnungen und Videos, die zeigen, wie das Rocker-Bogie-Federungssystem funktioniert)
- Mars-Rover-Strukturbausatz
- Grundwerkzeuge und Zubehör (z. B. Schraubendreher, Schrauben usw.)

### 3.2.3 Kursschritte

#### Schritt 1: Entschlüsselung des Rocker-Bogie-Systems

Das Rocker-Bogie-System ist wie eine Gebirgsziege der Mechanik – konzipiert, um alle Räder des Rovers am Boden zu halten, während er über raues und felsiges Gelände navigiert. Es ist speziell für die Bewältigung der unvorhersehbaren Landschaft des Mars gebaut, einschließlich steiler Anstiege und großer Felsbrocken. Dieses System verzichtet auf Federn und nutzt stattdessen die Geometrie seiner sechs Räder und deren Interaktion, um schwieriges Gelände zu bewältigen. Es ist ein glänzendes Beispiel dafür, wie intelligentes mechanisches Design Umwelthindernisse überwinden kann.

Lassen Sie uns in die zwei Hauptteile dieses Systems eintauchen – den „Rocker“ und den „Bogie“.



- Der „Rocker“-Teil des Systems ist wie die zwei großen Gliedmaßen an beiden Seiten des Rovers. Diese Glieder, oder Rocker, verbinden sich miteinander und mit dem Körper des Rovers, oder Chassis, durch einen Mechanismus, der als Differenzial bekannt ist. Ähnlich wie zwei Beine beim Gehen, rotieren die Rocker in entgegengesetzte Richtungen relativ zum Chassis und sorgen dafür, dass die meisten Räder Kontakt zum Boden halten. Der Körper des Rovers behält den durchschnittlichen Winkel beider Rocker bei. Ein Ende eines Rockers verbindet sich mit einem Rad, während das andere Ende mit dem Bogie verbunden ist.
- Der „Bogie“-Teil des Systems ist wie ein kleines, beinähnliches Wesen, das sich an den Rocker anschließt. Es ist ein kleineres Verbindungssystem, das sich in der Mitte zum Rocker hin dreht und an beiden Enden ein Rad hat.

Mit diesem Grundverständnis springen wir zum nächsten Schritt unseres Abenteuers.

### Schritt 2: Das System in Aktion sehen

Unten ist ein GIF, das die einzigartigen Merkmale des Rocker-Bogie-Federungssystems zeigt und veranschaulicht, wie es Mars-Rovern ermöglicht, das herausfordernde Marsgelände zu bewältigen.

Nachdem wir das GIF gesehen haben, lasst uns diskutieren! Denkt über diese Fragen nach:

- Warum denkst du, ist das Rocker-Bogie-Federungssystem für die Mars-Erkundung geeignet?
- Kannst du beschreiben, wie das Rocker-Bogie-System in deinen eigenen Worten funktioniert?
- Was sind die Schlüsselemente des Rocker-Bogie-Systems, die den Rovern helfen, schwieriges Gelände zu bewältigen?

Fühle dich frei, deine Gedanken und Einsichten über das Rocker-Bogie-Federungssystem zu teilen.

### Schritt 3: Bau es

Jetzt, wo wir über das Rocker-Bogie-System gelernt haben, ist es an der Zeit, unser eigenes zu bauen.

Materialien, die du benötigst:

- GalaxyRVR-Kit
- Grundwerkzeuge wie Schraubendreher und Schraubenschlüssel
- Befolge die Schritte in der Montageanleitung des GalaxyRVR-Kits, um das Federungssystem des Rovers zu konstruieren.

Bitte beachte, dass Geduld und Präzision hier wesentlich sind, stelle sicher, dass du jedes Teil korrekt platzierst und fest sicherst.

Unterdessen diskutiere mit deinen Kollegen über das Design und die Funktion jedes Bauteils, das du zusammenbaust. Dies wird nicht nur beim Verständnis des Designs helfen, sondern auch bei seiner praktischen Anwendung in der Mars-Erkundung.

Denke daran, mach dir keine Sorgen, wenn du während des Zusammenbaus oder der Tests auf Probleme stößt. Das ist alles Teil des Ingenieurprozesses! Probleme zu lösen ist, wie wir lernen und Neues entwickeln.

### Schritt 4: Zusammenfassung und Reflexion

Hast du während des Zusammenbaus des Federungssystems bemerkt, dass alle beweglichen Teile selbstsichernde Muttern verwenden? Hast du dich jemals gefragt, warum?

Selbstsichernde Muttern sind eine Art Befestigungselement, das einen Gummiring in einer gewöhnlichen Mutter enthält. Dieses Design stellt sicher, dass die montierten Teile durch Vibrationen während der Bewegung nicht leicht lockern und abfallen.

Darüber hinaus stellt es auch sicher, dass sich die Teile innerhalb eines bestimmten Bereichs drehen können.



Daher musst du während der Montage die Schraube und die selbstsichernde Mutter zuerst mit einer Steckdose und einem Schraubendreher festziehen und dann ein wenig lockern. Dies stellt sicher, dass zwischen den Teilen genügend Raum für freie Rotation besteht, ohne dass sie zu locker sind.

In dieser Lektion haben wir nicht nur über das Rocker-Bogie-System gelernt, sondern auch selbst eines gebaut. Darüber hinaus können wir manuell simulieren, wie es dem Mars-Rover ermöglicht, reibungslos über verschiedene raue Gelände zu fahren.

Mit diesem Wissen und dieser Erfahrung sind wir nun besser gerüstet, um tiefer in die unbekannten Reiche der Mars-Erkundung vorzudringen. Lassen Sie uns weiterhin die Geheimnisse des roten Planeten entschlüsseln.

## **3.3 Lektion 3: Einstieg in die Welt von Arduino und Programmierung**

In unserer vorherigen Lektion haben wir erfolgreich das Rocker-Bogie-Federungssystem gebaut. Um es jedoch funktionsfähig zu machen, müssen wir es mit Strom, einem Steuerungsboard und einer Programmierung versorgen, um seine Bewegungen zu steuern.

In dieser Lektion werden wir uns mit dem Steuerungsboard und der Programmierplattform vertraut machen, die wir verwenden werden.

### **3.3.1 Kursziele**

- Grundkonzepte und Funktionen von Arduino verstehen.
- Über das SunFounder R3 Board lernen.
- Arduino IDE installieren und sich mit der Benutzeroberfläche vertraut machen.
- Grundlegende Syntaxregeln für die Arduino-Programmierung lernen.

### **3.3.2 Kursmaterialien**

- SunFounder R3 Board
- Arduino IDE
- USB-Kabel
- Computer

### **3.3.3 Kursschritte**

#### **Schritt 1: Einführung in Arduino**

Du hast den Begriff „Arduino“ vielleicht schon oft in verschiedenen Zusammenhängen gehört, aber was genau ist es und warum ist es so beliebt geworden?

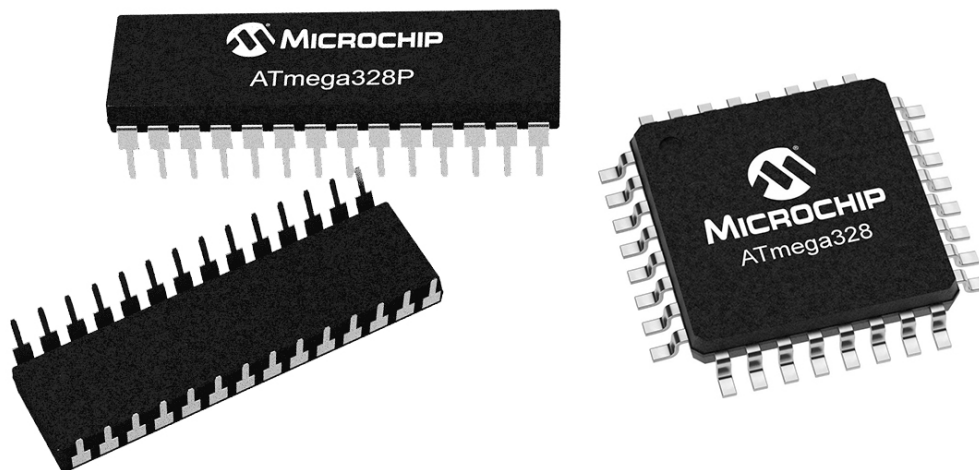
Arduino ist eine benutzerfreundliche Open-Source-Elektronikplattform für sowohl Hardware- als auch Softwareanwendungen. Sie ist darauf ausgelegt, digitale Geräte und interaktive Objekte zu erschaffen, die die physische Welt um sie herum wahrnehmen und steuern können.

Klären wir das genauer:

- **Open-Source:** Denke an Open-Source wie an einen Gemeinschaftsgarten. Jeder kann ihn nutzen, jeder kann dazu beitragen, und jeder kann davon profitieren. Bei Open-Source werden sowohl die Entwürfe der physischen Teile (die Hardware) als auch die Programmieranweisungen (die Software) frei geteilt. Das bedeutet, dass jeder sie nutzen, verbessern oder eigene Versionen erstellen kann. Es geht um Teilen und Kreativität!



- **Mikrocontroller:** Ein Mikrocontroller ist wie das Gehirn des Arduino. Es ist ein kleiner Computer, der einfache Software ausführen kann. Obwohl er nicht so leistungsfähig wie der Computer ist, den du jetzt benutzt, ist er perfekt für einfache Aufgaben wie das Verstehen von Nachrichten von Sensoren oder das Leuchten einer LED (eine kleine Lampe).



- **Entwicklungsboard:** Stell dir das Entwicklungsboard als den Körper vor, der das Gehirn unterstützt. Es ist das Board, auf dem der Mikrocontroller sitzt und es enthält andere Teile, die dem Mikrocontroller helfen, mit der Welt zu interagieren. Diese Teile umfassen Dinge wie Oszillatoren (die bei der Zeitmessung helfen), Spannungsregler (die die Leistungspegel steuern) und Anschlüsse für Strom und Daten (wie die Stecker und Schalter in deinem Haus).

			
Arduino UNO R3	Arduino Mega 2560 Rev3	Arduino Leonardo	Arduino UNO Mini Limited Edition
			
Arduino Due	Arduino Micro	Arduino Zero	Arduino UNO WiFi Rev2

- **Arduino IDE:** Dies ist wie das Klassenzimmer für deinen Arduino. Es ist ein Programm, das auf deinem Computer läuft, wo du die Anweisungen schreiben kannst, die deinem Arduino sagen, was er tun soll. Diese Anweisungen werden in einer auf C++ basierenden Programmiersprache geschrieben. Sobald du deine Anweisungen geschrieben hast, kannst du sie über ein USB-Kabel an das Arduino-Board senden, genau wie wenn du deine Hausaufgaben abgibst!



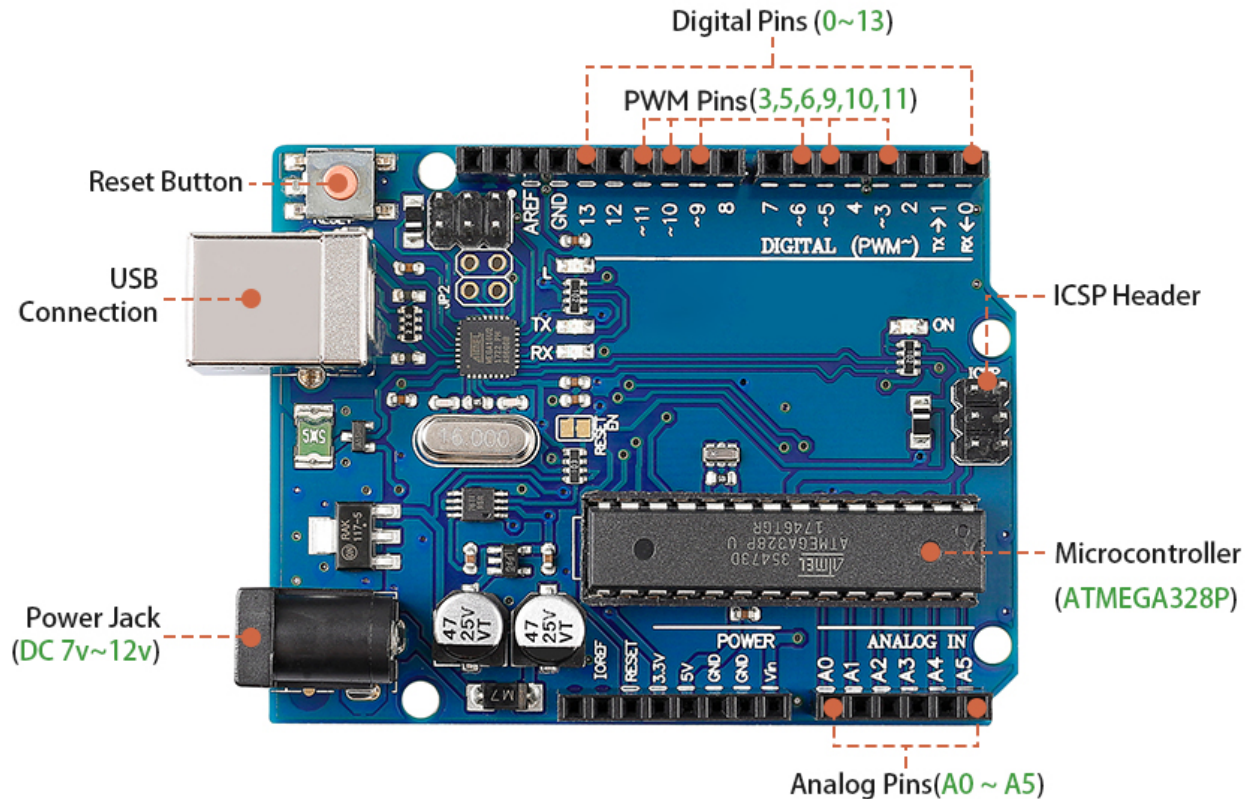
Jetzt, da du diese grundlegenden Ideen verstanden hast, bist du auf dem besten Weg, ein Arduino-Experte zu werden!

Dann werden wir in einige praktische Aktivitäten eintauchen, um dich mit der Arduino-Programmierung und Ingenieursprinzipien vertraut zu machen. Bereite dich auf eine spannende Lernreise vor!

### Schritt 2: Machen Sie sich mit Ihrem SunFounder R3 Board vertraut

In Ihrem Kit finden Sie ein blaues Board, das wie eine kleine Stadt voller kleiner metallischer Türme und Wege erscheint. Aber lassen Sie sich nicht einschüchtern! Dies ist das SunFounder R3 Entwicklungsboard, eine Art Arduino-Board, das verwendet werden kann, um eine Vielzahl von elektronischen Geräten und Projekten zu programmieren und zu steuern.

Lassen Sie uns seine Schlüsselfunktionen in einfachen Begriffen verstehen:



- **14 Digital Pins:** Denken Sie an diese Pins wie an kleine Boten. Sie können programmiert werden, um einfache „Ja“ oder „Nein“ Nachrichten an andere Teile Ihres Mars-Rovers zu senden (Ausgang) oder zu empfangen (Eingang). Diese Nachrichten sind eigentlich „ein“ oder „aus“ Signale, die das Board verwendet, um Dinge wie Lichter oder Motoren zu steuern.
  - Sechs dieser speziellen Pins können sogar Nachrichten in einer Art Geheimcode namens PWM (Pulse Width Modulation) senden. Dieser Code kann verwendet werden, um zu steuern, wie hell ein Licht ist, wie schnell ein Motor sich dreht oder sogar, wo sich ein bewegliches Teil positioniert.
- **6 Analog Pins:** Diese Pins sind wie die sechs besonderen Sinne des Boards. Sie können Signale von verschiedenen Sensortypen (wie einem Temperatursensor) lesen und diese Signale in eine Sprache übersetzen, die das Board verstehen und in seiner Programmierung verwenden kann.
- **USB Connection:** Dies ist wie die Nabelschnur des Boards. Sie können es verwenden, um Ihr Board mit Ihrem Computer zu verbinden. Diese Verbindung ermöglicht es Ihrem Computer, dem Board beizubringen, was es tun soll, indem es ihm ein Programm sendet, das Sie schreiben.
- **Power Jack:** Dies ist die Nahrungsversorgung des Boards. Sie können eine Stromquelle, wie einen Akku oder einen AC/DC-Adapter, an diese Buchse anschließen, um Ihr Board mit dem Strom zu „füttern“, den es benötigt, um zu funktionieren.
- **ICSP Header:** Dies ist wie ein spezieller Eingang zur Programmierung des Boards. Es kann verwendet werden, wenn Sie einen externen Programmierer haben (ein spezielles Gerät zum „Unterrichten“ des Boards).
- **Reset Button:** Wenn Sie sie drücken, ist es, als würden Sie dem Board sagen, dass es vergessen soll, was es gerade getan hat, und sein Programm von vorne beginnen soll.

Mit diesen Grundlagen sind Sie bereit, Ihre Programmierabenteuer mit dem SunFounder R3-Board zu beginnen!

### Schritt 3: Installieren Sie Arduino IDE

Jetzt, da wir wissen, was Arduino und das Arduino-Board sind, ist es an der Zeit, dieses Wissen in die Praxis umzusetzen. Wir werden die Arduino IDE installieren, die Software, die wir verwenden werden, um unser Arduino-Board zu programmieren.

Die neueste Version der Arduino IDE ist Version 2.0. Sie ist vollgepackt mit Funktionen und sehr benutzerfreundlich. Sie sollten jedoch wissen, dass sie einige Systemanforderungen hat:

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

Um zu beginnen, folgen Sie diesen Schritten:

1. Besuchen Sie und 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**

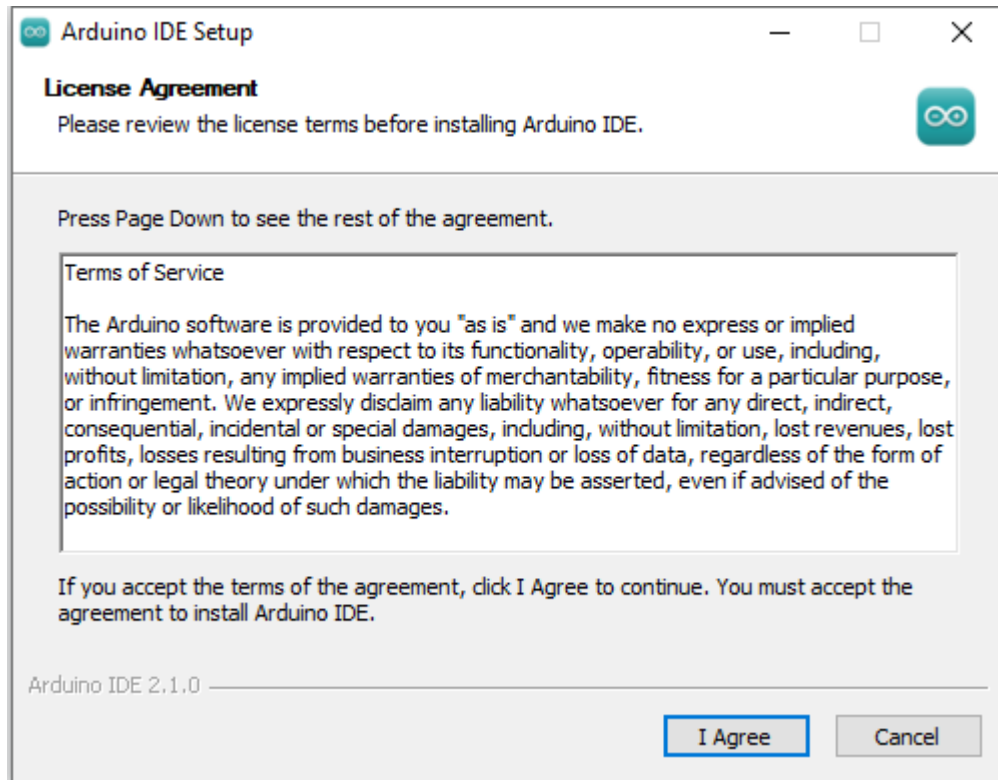
The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

**DOWNLOAD OPTIONS**

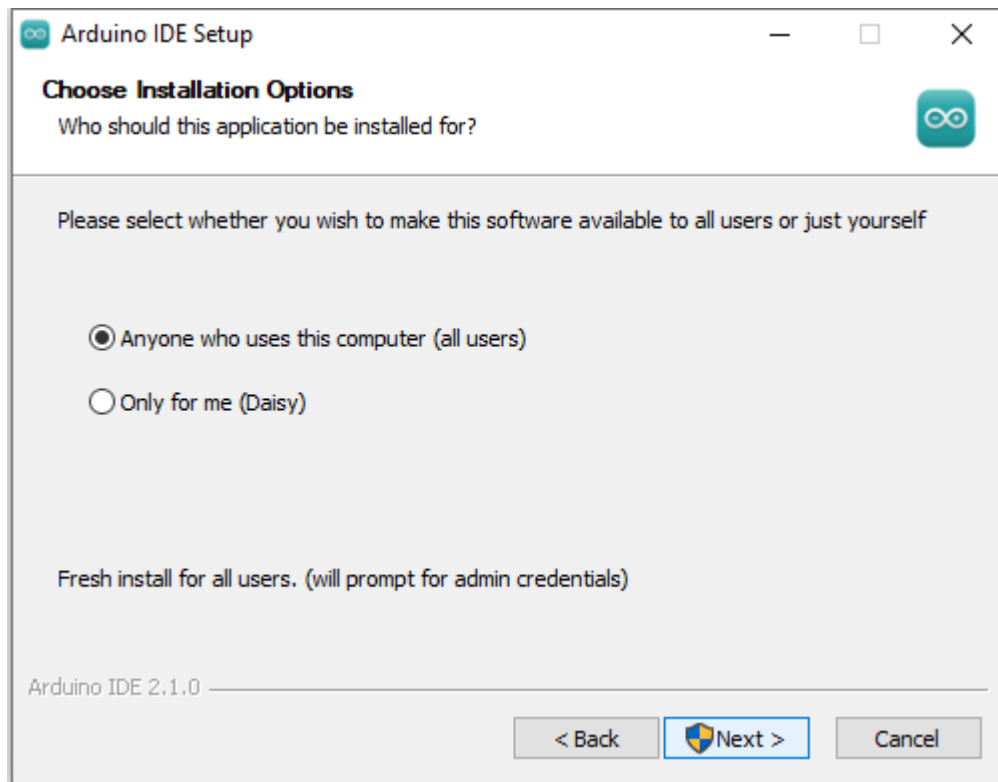
- Windows** Win 10 and newer, 64 bits
- Windows** MSI installer
- Windows** ZIP file
- Linux** AppImage 64 bits (X86-64)
- Linux** ZIP file 64 bits (X86-64)
- macOS** 10.14: "Mojave" or newer, 64 bits

### Für Windows-Benutzer:

1. Sobald Sie die Datei heruntergeladen haben (sie wird etwas wie `arduino-ide_XXXX.exe` heißen), doppelklicken Sie darauf, um den Installationsprozess zu starten.
2. Ihnen wird die **License Agreement** angezeigt. Nehmen Sie sich einen Moment Zeit, diese durchzulesen, und wenn Sie den Bedingungen zustimmen, klicken Sie auf „Ich stimme zu“.



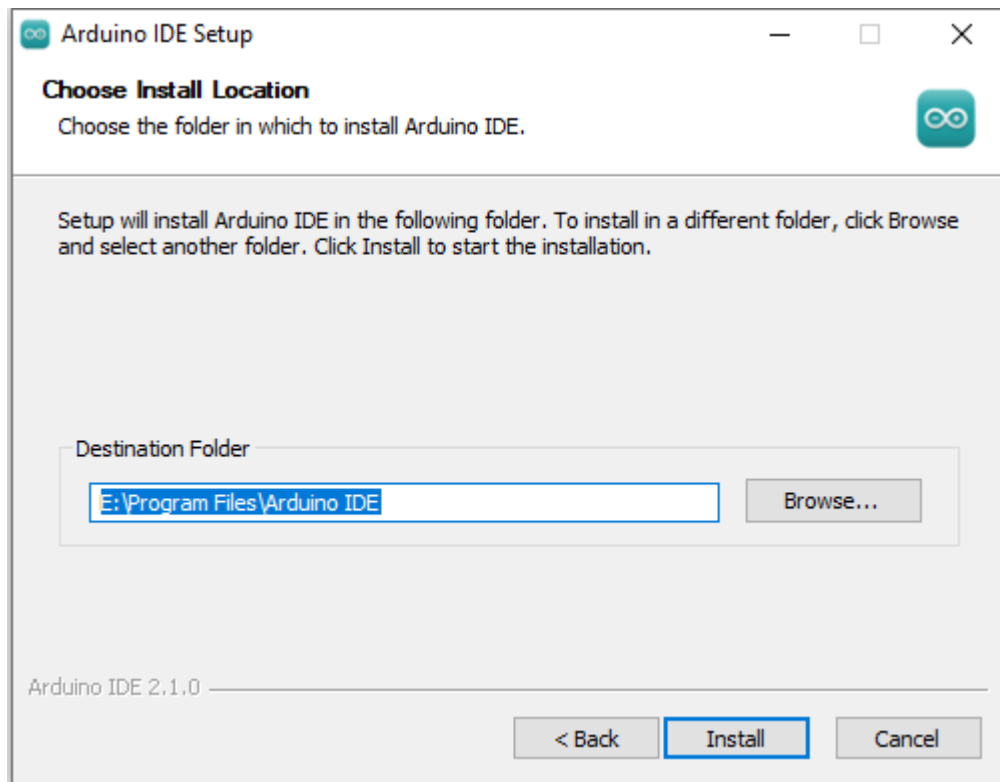
3. Als Nächstes werden Sie gebeten, Installationsoptionen auszuwählen. Lassen Sie diese so, wie sie sind, und klicken Sie auf „Next“.



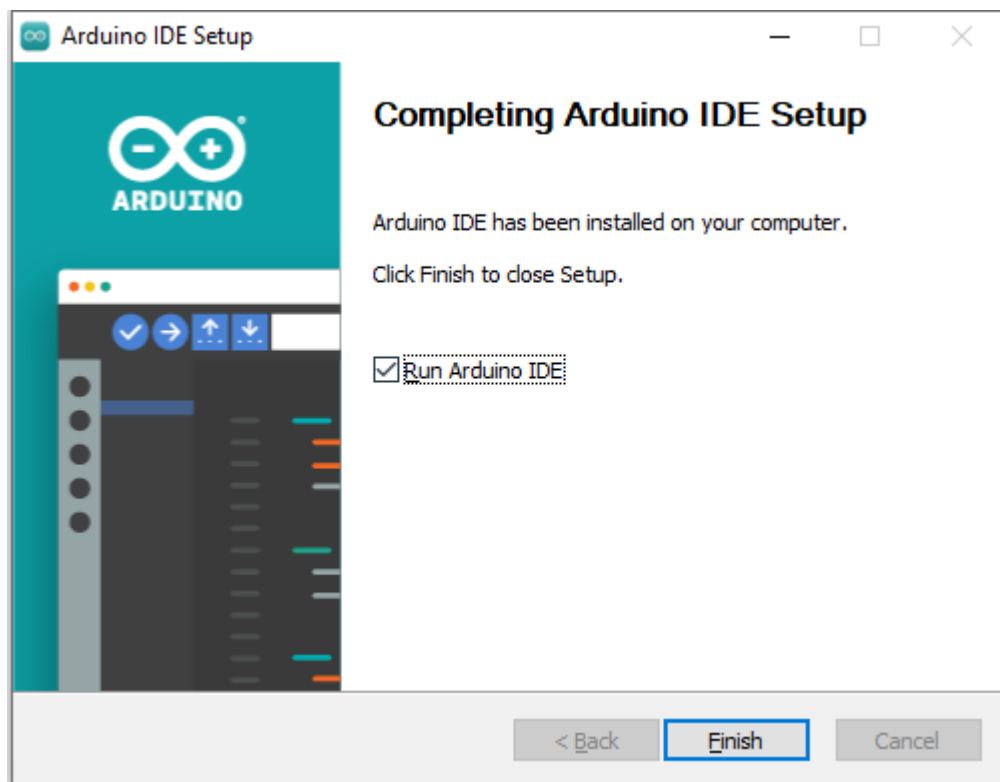
4. Wählen Sie aus, wo Sie die Software installieren möchten. Es ist in der Regel am besten, sie auf einem anderen



Laufwerk als dem zu installieren, das Ihr System verwendet.

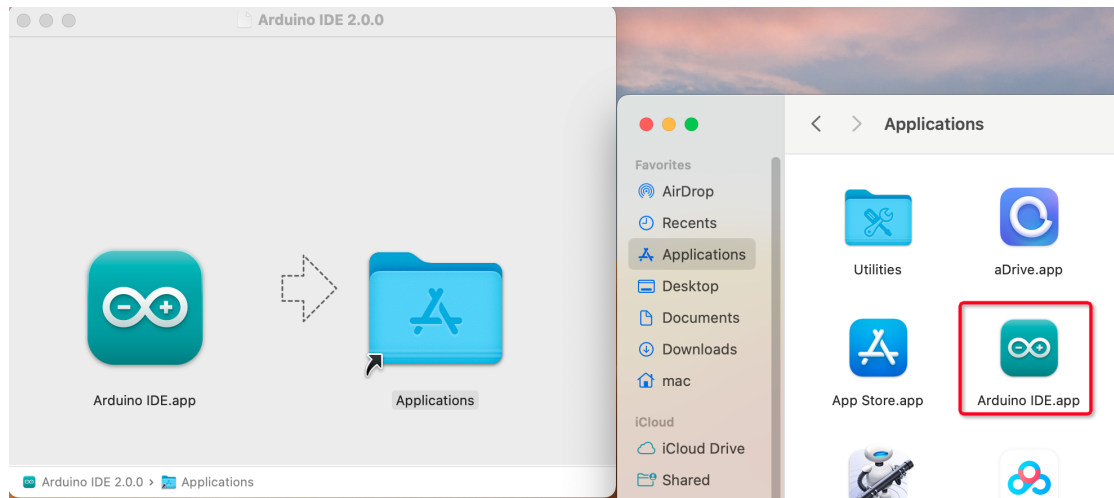


5. Klicken Sie auf „Install“, um die Installation zu starten. Sobald sie abgeschlossen ist, klicken Sie auf „Finish“.



### Für macOS-Benutzer:

Doppelklicken Sie auf die heruntergeladene Datei (die etwas wie `arduino_ide_XXXX.dmg` heißen wird). Befolgen Sie die Anweisungen auf dem Bildschirm, um die **Arduino IDE**-App in den **Applications**-Ordner zu ziehen. Nach einigen Sekunden wird die Arduino IDE erfolgreich installiert sein.

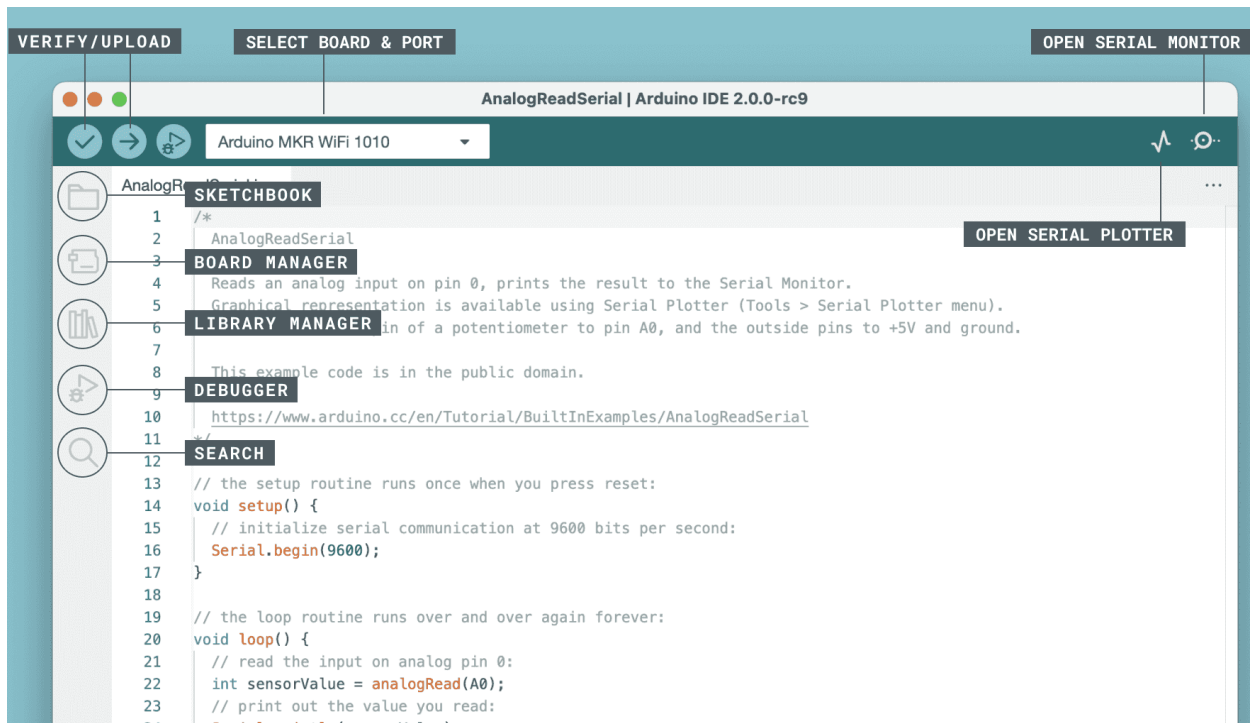


### Für Linux-Benutzer:

Eine detaillierte Anleitung zur Installation der Arduino IDE 2.0 auf einem Linux-System finden Sie hier: .

### Schritt 4: Entdecken des Arduino-Spielplatzes (IDE)

Stellen Sie sich vor, die Arduino IDE sei ein magischer Spielplatz voller Werkzeuge und Gadgets, die darauf warten, von uns erkundet und genutzt zu werden. Als Nächstes werde ich Ihnen helfen, jede Ecke dieses Spielplatzes zu verstehen.



Hier ist, was Sie auf Ihrem Spielplatz finden:



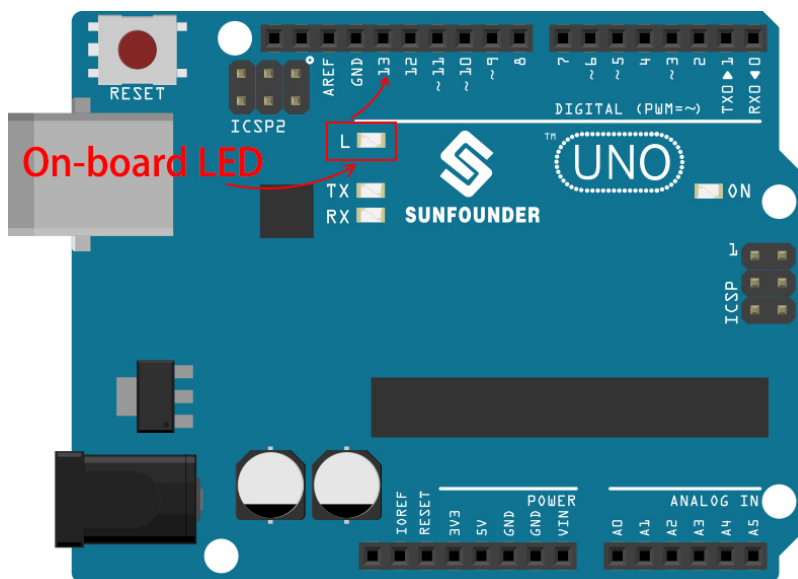
- **Verify / Upload** - Stellen Sie sich das wie Ihren magischen Aufzug vor. Es nimmt den von Ihnen geschriebenen Code und befördert ihn auf Ihr Arduino-Board.
- **Select Board & Port** - Das ist Ihre Schatzkarte. Es zeigt automatisch die Arduino-Boards an, die Sie an Ihren Computer angeschlossen haben, und teilt Ihnen deren Portnummer mit.
- **Sketchbook** - Das ist Ihre persönliche Bibliothek. Hier werden alle Ihre Skizzen (Programme) auf Ihrem Computer gespeichert. Außerdem kann es sich mit der Arduino Cloud verbinden, sodass Sie Ihre Skizzen auch aus der Online-Welt abrufen können.
- **Boards Manager** - Denken Sie daran als Ihr Werkzeugkasten. Hier können Sie verschiedene Pakete für Ihr Arduino finden und installieren.
- **Library Manager** - Das ist Ihre endlose Schatztruhe. Tausende von Bibliotheken, die von Arduino und seiner Community erstellt wurden, warten hier auf Sie. Brauchen Sie ein Werkzeug oder Material für Ihren Code? Tauchen Sie ein und finden Sie es!
- **Debugger** - Stellen Sie sich vor, Sie hätten eine Superkraft, die es Ihnen ermöglicht, Ihren Code in Echtzeit zu testen und zu debuggen, um Probleme zu finden und zu beheben. Das ist es, was dies ist!
- **Search** - Denken Sie daran als Ihre Lupe. Es hilft Ihnen, nach Schlüsselwörtern in Ihrem Code zu suchen.
- **Open Serial Monitor** - Das ist wie Ihr Kommunikationsgerät. Es öffnet einen neuen Tab, der es Ihrem Computer und Ihrem Arduino-Board ermöglicht, Nachrichten hin und her zu senden.

Jetzt, da wir einen Blick auf den Spielplatz geworfen haben, ist es an der Zeit, einzutauchen und zu beginnen, zu erschaffen!

### Schritt 5: Laden Sie Ihre erste Skizze hoch

In Ordnung, es ist Zeit, etwas Spaß zu haben! Wir werden eine LED blinken lassen - es ist wie ein „Hello, World!“ in der Welt von Arduino.

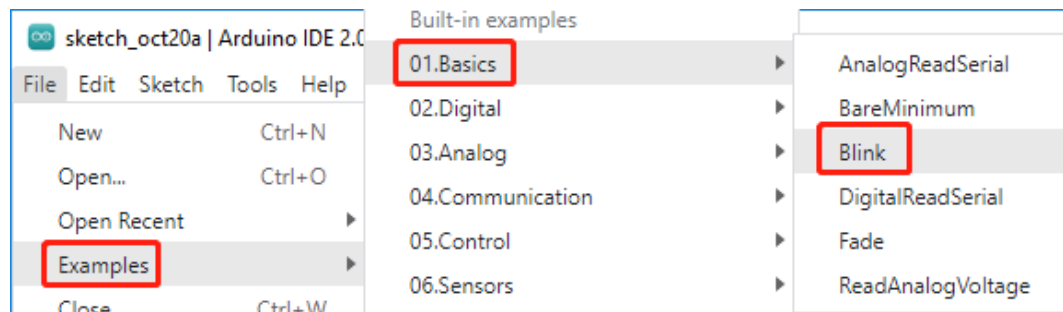
Die meisten Arduino-Boards haben eine eingebaute LED am Pin 13, was dies zu einem guten ersten Experiment macht.



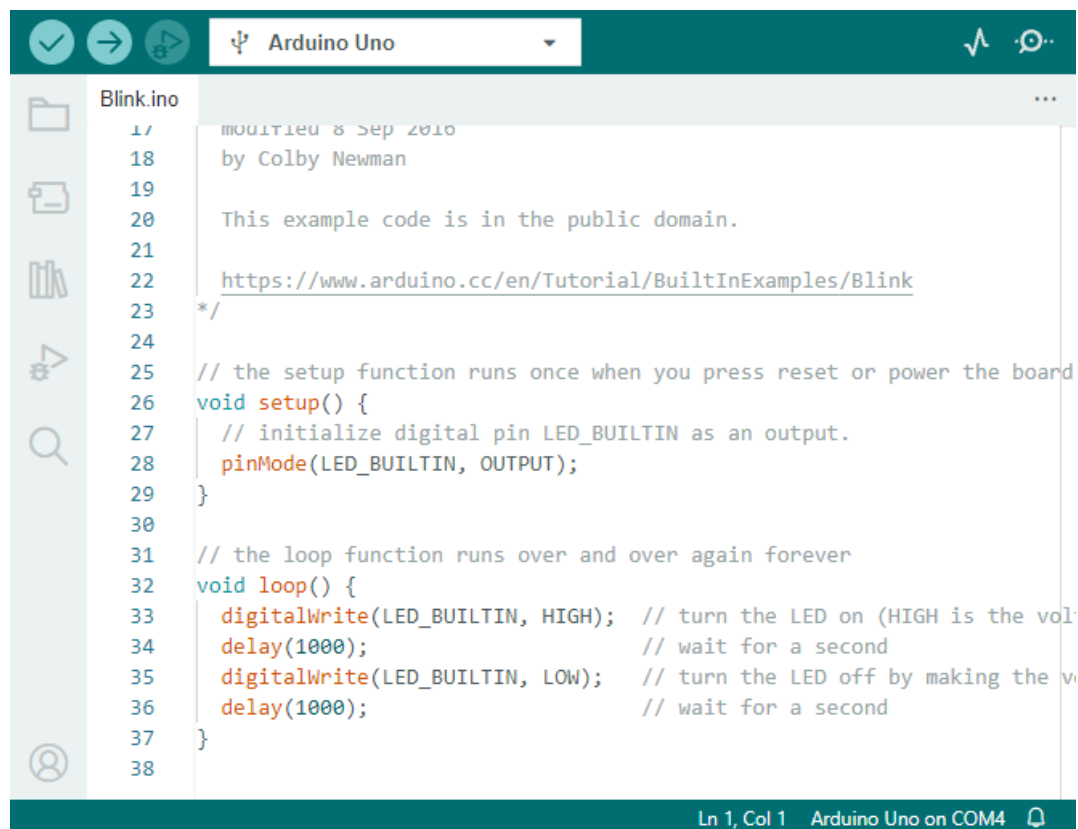
Lassen Sie uns das aufschlüsseln:

1. **Einstecken:** Verbinden Sie Ihr SunFounder R3 Board mit einem USB-Kabel mit Ihrem Computer. So geben wir unserem Board Strom und senden unser Programm (auch „Skizze“ genannt) an das Board. Sie fühlen sich vielleicht so, als würden Sie nur ein Computer-Gadget anschließen, aber glauben Sie mir, Sie verbinden sich mit einer Welt voller Möglichkeiten!

2. **Die Beispiel-Skizze finden:** Gehen Sie in der Arduino IDE zu **File -> Examples -> Basic -> Blink**. Was aufploppt, ist ein fertiges Programm, das wir ändern werden. Es ist wie einen fertigen Kuchen zu bekommen, den wir gleich dekorieren werden!



3. **Verstehe den Entwurf:** Schau dir den Code in diesem neuen Fenster an. Er weist Arduino an, die integrierte LED (die sich an Pin 13 befindet) für eine Sekunde einzuschalten, dann für eine Sekunde auszuschalten und diesen Vorgang zu wiederholen. Es ist wie Morsecode zu senden, aber mit Licht!



4. **Lade den Entwurf hoch:** Sobald du das richtige Board und den richtigen Port ausgewählt hast, klicke einfach auf den Hochlade-Button. Es ist so einfach wie einen Brief zu senden; du übermittelst deine Anweisungen an das Arduino-Board! Meistens erkennt das System automatisch das Board und den Port für dich.
5. **Beobachte, wie es funktioniert:** Wenn alles gut läuft, wirst du sehen, wie die LED auf deinem Arduino-Board anfängt, ein- und auszublinden. Es ist, als würde dein Arduino dir zuzwinkern!

Großartige Arbeit! Du hast gerade dein erstes Arduino-Programm ausgeführt und bist damit ein echter Programmierer! Was kommt als Nächstes? Wir kratzen gerade erst an der Oberfläche dessen, was Arduino leisten kann. Bereit für die nächste Herausforderung?

### Schritt 6: Einige unterhaltsame Fakten zur Arduino-Programmierung

Zeit, einige coole Geheimnisse über die Arduino-Programmierung zu lüften!

- Code-Magie: `setup()` und `loop()`

Ein Arduino-Entwurf, oder ein Stück Code, ist wie ein Theaterstück in zwei Akten:

- `setup()`: Das ist Akt 1, die Eröffnungsszene. Sie findet nur einmal statt, wenn dein Arduino-Board zum ersten Mal erwacht. Sie wird verwendet, um die Bühne vorzubereiten, indem Dinge wie Pin-Modi und Bibliotheken eingerichtet werden.
- `loop()`: Nach Akt 1 gehen wir über zu Akt 2, der sich wiederholt, bis der letzte Vorhang fällt (was nur passiert, wenn wir die Stromversorgung abschalten oder den Reset-Knopf drücken!). Dieser Teil des Codes ist wie der Hauptteil unseres Stücks, wo die eigentliche Aktion stattfindet.

Aber denk daran, selbst wenn es keine Magie (Code) im `setup()` oder `loop()` gibt, müssen wir sie trotzdem behalten. Sie sind wie die Bühne - selbst eine leere Bühne bleibt eine Bühne.

```
void setup() {
    // initialize digital pin LED_BUILTIN as an output.
    pinMode(LED_BUILTIN, OUTPUT);

    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the
    ↪voltage level)
    delay(1000);                     // wait for a second
    digitalWrite(LED_BUILTIN, LOW);  // turn the LED off by making the
    ↪voltage LOW
    delay(1000);                     // wait for a second
}

// the loop function runs over and over again forever
void loop() {
}
```

- Satzzeichen in der Programmierung

Genauso wie in einem Märchenbuch verwendet Arduino spezielle Satzzeichen, um den Code verständlich zu machen:

- Semikolons (;): Diese sind wie Punkte in einer Geschichte. Sie sagen dem Arduino: „Okay, ich bin fertig mit dieser Aktion. Was kommt als Nächstes?“
- Geschweifte Klammern {}: Diese sind wie der Anfang und das Ende eines Kapitels. Sie fassen Code-Teile zusammen und markieren, wo ein Abschnitt beginnt und endet.

Solltest du einige dieser Satzzeichen vergessen, keine Sorge! Der Arduino ist wie ein freundlicher Lehrer, der deine Arbeit überprüft, dir die Fehler zeigt und dir hilft, sie zu beheben. Das gehört alles zum Lernabenteuer!

- Über die Funktionen

Stelle dir diese Funktionen wie magische Zaubersprüche vor. Jeder Spruch hat eine spezifische Wirkung in unserem Arduino-Abenteuer:

- `pinMode()`: Dieser Spruch entscheidet, ob ein Pin ein EINGANG oder ein AUSGANG ist. Es ist, als würde man entscheiden, ob eine Figur in unserer Geschichte spricht (AUSGANG) oder zuhört (EINGANG).
- `digitalWrite()`: Dieser Spruch kann einen Pin HIGH (an) oder LOW (aus) schalten, wie das Ein- und Ausschalten eines magischen Lichts.
- `delay()`: Dieser Spruch lässt den Arduino für eine bestimmte Zeit pausieren, wie eine kurze Pause inmitten unserer Geschichte.

Wie in einem Zauberbuch kannst du all diese und viele weitere Zaubersprüche im finden. Je mehr Sprüche du kennst, desto aufregender können deine Arduino-Abenteuer werden!

- **Kommentare: Unsere Geheimen Nachrichten**

Wir haben auch eine Geheimsprache in der Programmierung, genannt **Kommentare**. Das sind Nachrichten, die wir in unserem Code mit `//` oder `/* */` schreiben können. Das Magische daran? Der Arduino ignoriert sie komplett! Es ist ein großartiger Ort, um Notizen für dich selbst oder andere zu hinterlassen, um zu erklären, was die schwierigen Teile des Codes tun.

- **Lesbarkeit des Codes: Code Benutzerfreundlich Gestalten**

Obwohl du deinen Code schreiben kannst, wie du möchtest (zum Beispiel verursacht das Setzen von Semikolons auf einer separaten Zeile keine Fehler), ist es wichtig, die Lesbarkeit des Codes im Auge zu behalten.

Wie beim Schreiben einer guten Geschichte kann die Art und Weise, wie wir Code schreiben, diesen entweder Spaßig und leicht oder langweilig und schwer lesbar machen. Hier sind einige Möglichkeiten, deinen Code freundlicher zu gestalten:

- Verwende eine angemessene Einrückung, um deine Sätze in ordentliche Absätze zu ordnen. Das hilft dem Leser zu verstehen, wo ein Abschnitt endet und ein anderer beginnt.
- Verwende sinnvolle Variablennamen. Es ist wie, einen Charakter in einer Geschichte mit einem passenden Namen zu nennen.
- Halte deine Funktionen klein und einfach, wie kurze und süße Kapitel in einem Buch.
- Hinterlasse Kommentare für die kniffligen Teile. Es ist wie, eine Fußnote zu hinterlassen, um ein schwieriges Wort zu erklären.

Denk daran, wir programmieren nicht nur für Maschinen, sondern auch für Menschen. Also lass uns dafür sorgen, dass unser Code eine klare und verständliche Geschichte erzählt!

### Schritt 7: Reflektieren und Verbessern

Einen Moment innezuhalten und über unsere Reise nachzudenken, kann uns Einsichten geben, die wir im Trubel der Entdeckung vielleicht übersehen würden. Frag dich selbst:

- Was war der interessanteste Teil dieses Arduino-Abenteuers?
- Gab es Herausforderungen auf dem Weg? Wie hast du sie überwunden?
- Könntest du einem Freund erklären, was Arduino ist, was die Arduino-IDE macht oder wie man Arduino-Code ausführt?
- Wie würdest du deine erste Erfahrung mit der Arduino-Programmierung beschreiben?
- Was möchtest du noch über Arduino lernen?

Indem du über diese Fragen nachdenkst, vertiefst du dein Verständnis und bereitest dich auf zukünftige Erkundungen vor. Denk immer daran, es gibt keine „falsche“ Antwort bei der Reflexion – es ist schließlich deine persönliche Reise!

## 3.4 Lektion 4: Beherrschung des TT-Motors

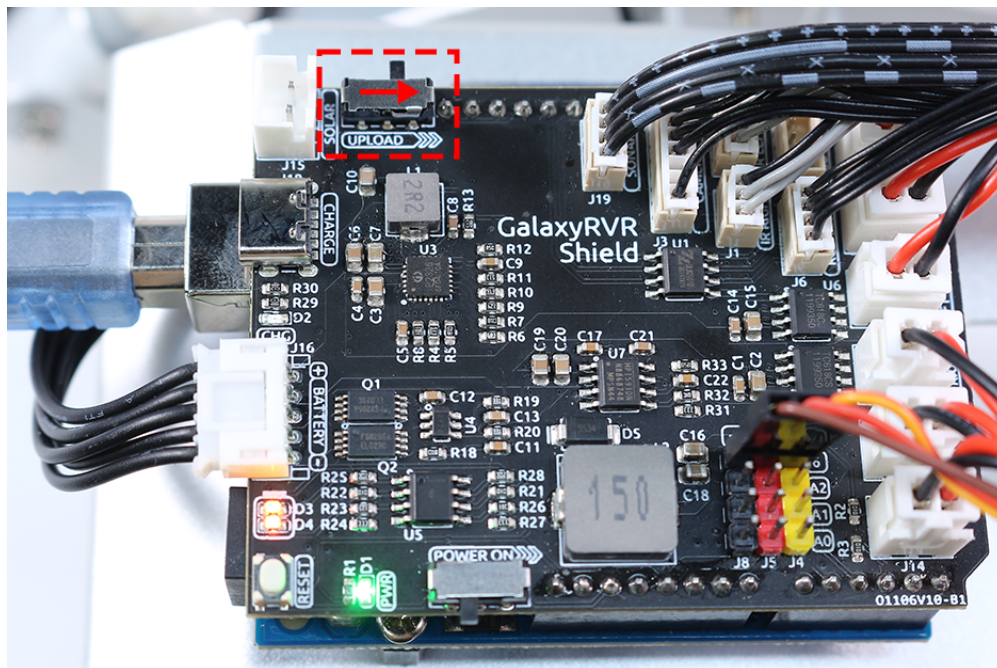
In den vorherigen Lektionen haben wir Mars-Rover, ihre Federungssysteme erkundet und uns in das Wissen über Arduino vertieft.

In diesem spannenden Kurs werden wir die Funktionsweise von Motoren erforschen, einer Schlüsselkomponente, die Mars-Rover antreibt. Wir werden die Prinzipien verstehen, die diese Motoren antreiben, und lernen, sie mit Hilfe eines SunFounder R3-Boards und einem GalaxyRVR-Shield zu steuern.

Am Ende dieses Kurses wirst du ein solides Verständnis für den Betrieb von Motoren und praktische Erfahrung in der Motorsteuerung haben.

Lass uns eintauchen!

**Bemerkung:** Wenn du diesen Kurs nach dem vollständigen Zusammenbau des GalaxyRVR lernst, musst du diesen Schalter vor dem Hochladen des Codes nach rechts bewegen.



### 3.4.1 Kursziele

- Verstehe die grundlegenden Prinzipien von Motoren und die Eigenschaften des TT-Motors.
- Lerne, wie man die Richtung und Geschwindigkeit des TT-Motors steuert.
- Verstehe, wie das GalaxyRVR-Shield sechs Motoren steuert.

### 3.4.2 Kursmaterialien

- SunFounder R3-Board
- TT-Motor
- GalaxyRVR-Shield
- Batterie
- USB-Kabel
- Arduino IDE
- Computer

### 3.4.3 Kursschritte

#### Schritt 1: Was ist ein Motor?

Motoren spielen eine integrale Rolle in unserem täglichen Leben. Sie sind überall! Von den elektrischen Ventilatoren, die uns an heißen Tagen kühlen, den Mixern, die uns helfen, köstliche Kuchen zu backen, bis hin zu den Elektroautos, die auf den Straßen vorbeiflitzen – Motoren bringen Dinge in Bewegung!



Ein Motor ist wie das Herz einer Maschine. Er wandelt elektrische Energie in mechanische Energie um und bringt so unsere Spielzeuge, Haushaltsgeräte und sogar große Fahrzeuge zum Leben!

Die Magie hinter einem Motor ist gar keine Magie – es ist Wissenschaft, genauer gesagt das Prinzip der elektromagnetischen Induktion. So funktioniert es: Wenn Strom einem Motor zugeführt wird, erzeugt er ein Magnetfeld. Dieses Magnetfeld interagiert dann mit anderen Magneten innerhalb des Motors, wodurch der Motor sich zu drehen beginnt. Diese Drehung, wie das Drehen eines Kreisels, kann dann verwendet werden, um Räder, Propeller oder andere bewegliche Teile einer Maschine zu bewegen.

Der Typ des Motors, auf den wir uns in unserem GalaxyRVR konzentrieren, ist eine spezifische Art namens TT-Getriebemotor.





Dies ist im Wesentlichen ein normaler Motor, kombiniert mit einer Reihe von Zahnrädern, die alle in einem Kunststoffgehäuse eingeschlossen sind.

Während sich der Motor dreht, übersetzen die Zahnräder diese Drehung auf die Räder unseres Rovers. Der Einsatz von Zahnrädern bietet einen entscheidenden Vorteil - er erhöht das Drehmoment und ermöglicht es dem Motor, größere, schwerere Lasten zu bewegen.

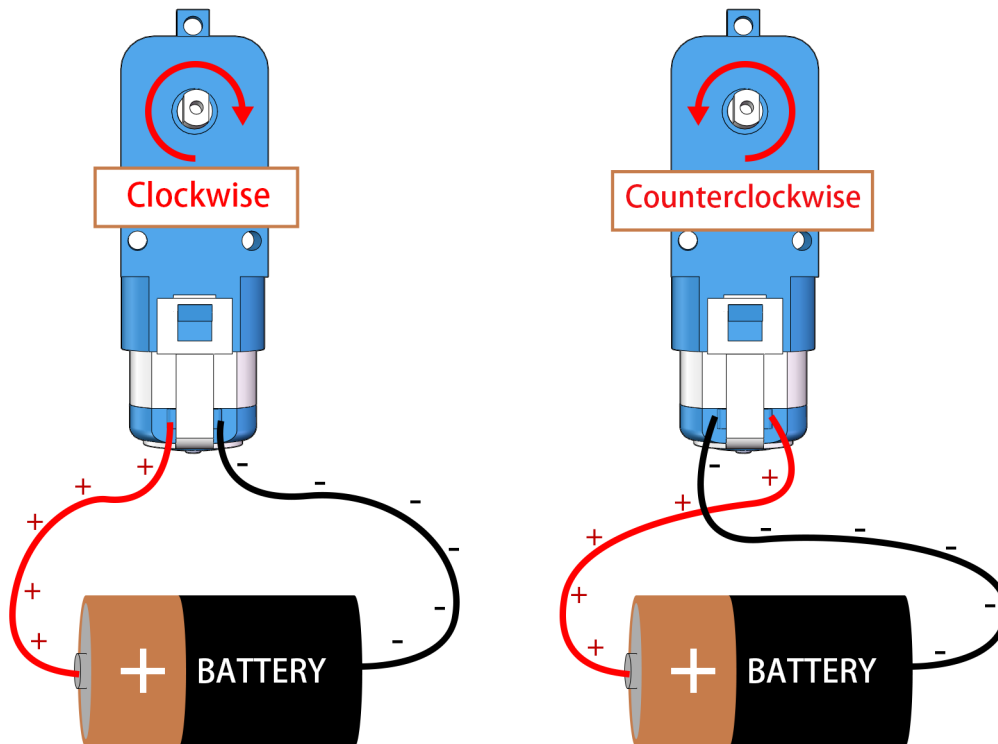
Ist es nicht faszinierend zu sehen, wie Wissenschafts- und Ingenieursprinzipien zum Leben erwachen? Motoren sind ein perfektes Beispiel für diese Prinzipien in Aktion. Indem wir verstehen, wie Motoren funktionieren, können wir uns eine Vielzahl von Maschinen ausdenken und erfinden. Lass uns tiefer in die Welt der Motoren eintauchen und unsere Kreativität entfesseln!

### **Schritt 2: Erforschung der Funktionsweise und des Betriebs von Motoren**

Nachdem wir verstanden haben, was ein Motor ist und sein breites Spektrum an Anwendungen kennen, wird es Zeit, uns in das Herz des Motorbetriebs zu begeben.

Im Wesentlichen funktioniert ein Motor nach dem Prinzip des Elektromagnetismus. Wenn ein elektrischer Strom durch einen Draht fließt, erzeugt er ein Magnetfeld um sich herum. Dieses Magnetfeld kann mit anderen Magnetfeldern interagieren und Bewegung verursachen.

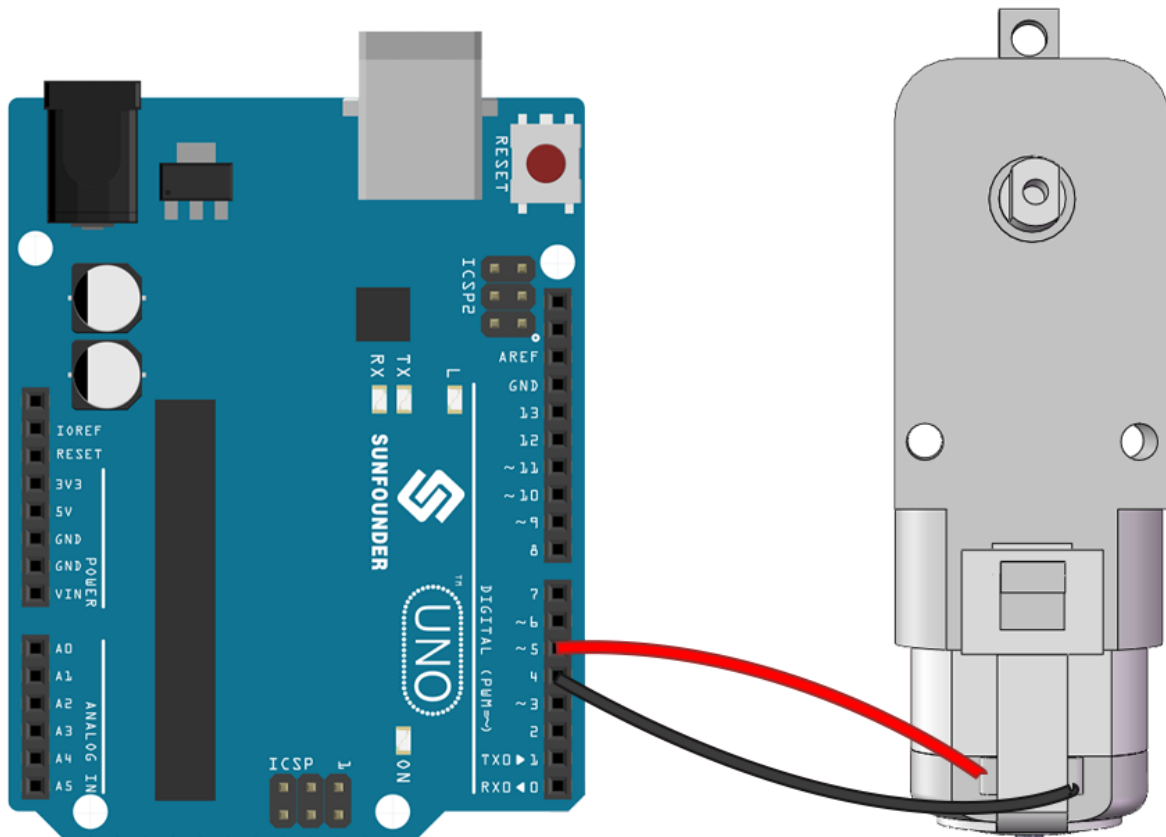
Betrachten wir ein einfaches Experiment, bei dem wir einen Motor direkt an eine Batterie anschließen. Der Strom aus der Batterie fließt in den Motor und löst den internen Mechanismus des Motors aus, der zu drehen beginnt. Diese Drehbewegung ist auf die magnetischen Kräfte im Inneren des Motors zurückzuführen.



Interessanterweise dreht sich der Motor in die entgegengesetzte Richtung, wenn Sie die Anschlüsse an der Batterie umkehren! Dies geschieht, weil sich die Richtung des Stromflusses ändert, was wiederum die Richtung des Magnetfelds und somit die Drehrichtung des Motors ändert.

Nun wissen wir, dass der Anschluss des Motors direkt an eine Batterie ihn zum Drehen bringen kann, aber oft möchten wir seine Bewegung mit Code steuern, also schließen wir ein Arduino-Board dazwischen an. Aber was würde passieren, wenn wir versuchen, den Motor direkt an die Signalpins auf dem Arduino-Board anzuschließen?

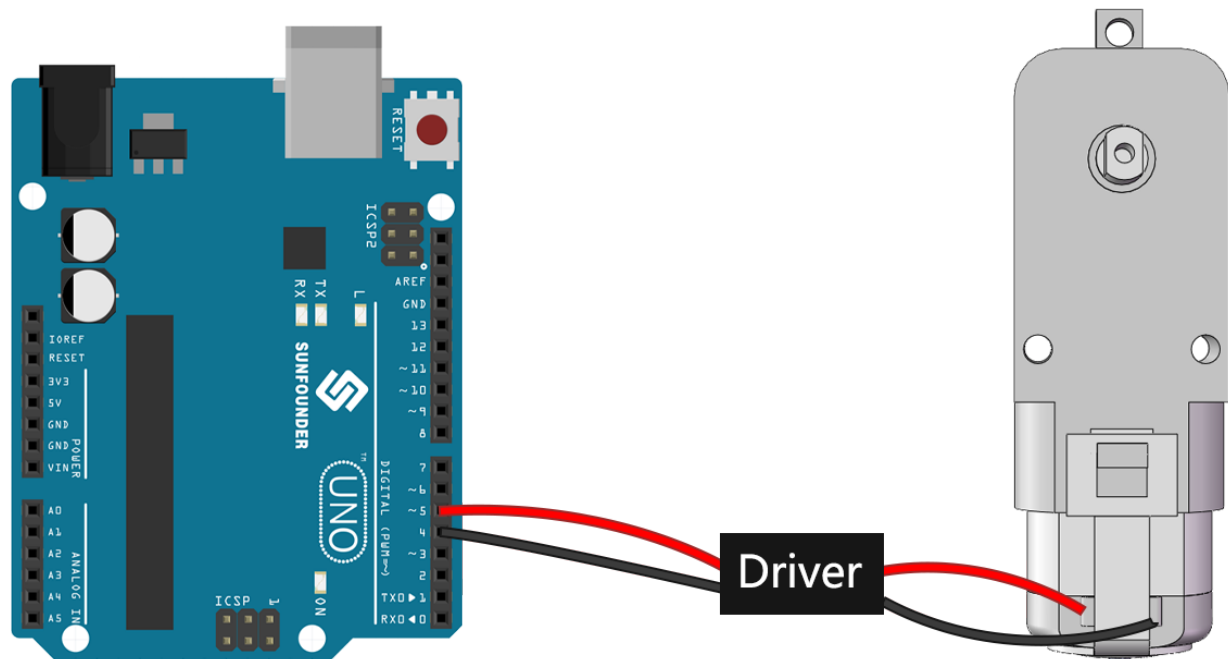




Wenn Sie erraten haben, dass sich der Motor nicht drehen würde, liegen Sie richtig! Aber warum ist das so?

Die Antwort liegt im Stromausgang des Arduino-Boards. Die Signalfpins auf einem typischen Arduino-Board können nur etwa 20 mA Strom ausgeben, was nicht ausreicht, um einen Motor anzutreiben.

Wie können wir also Motoren mit unserem Arduino steuern? Hier kommt eine entscheidende Komponente ins Spiel - ein Motor-Treiber. Denken Sie an einen Motor-Treiber als eine Brücke zwischen dem Arduino und dem Motor. Er nimmt das schwache Steuersignal des Arduino auf, verstärkt es und leitet es an den Motor weiter, wodurch der Motor sich zu drehen beginnt.

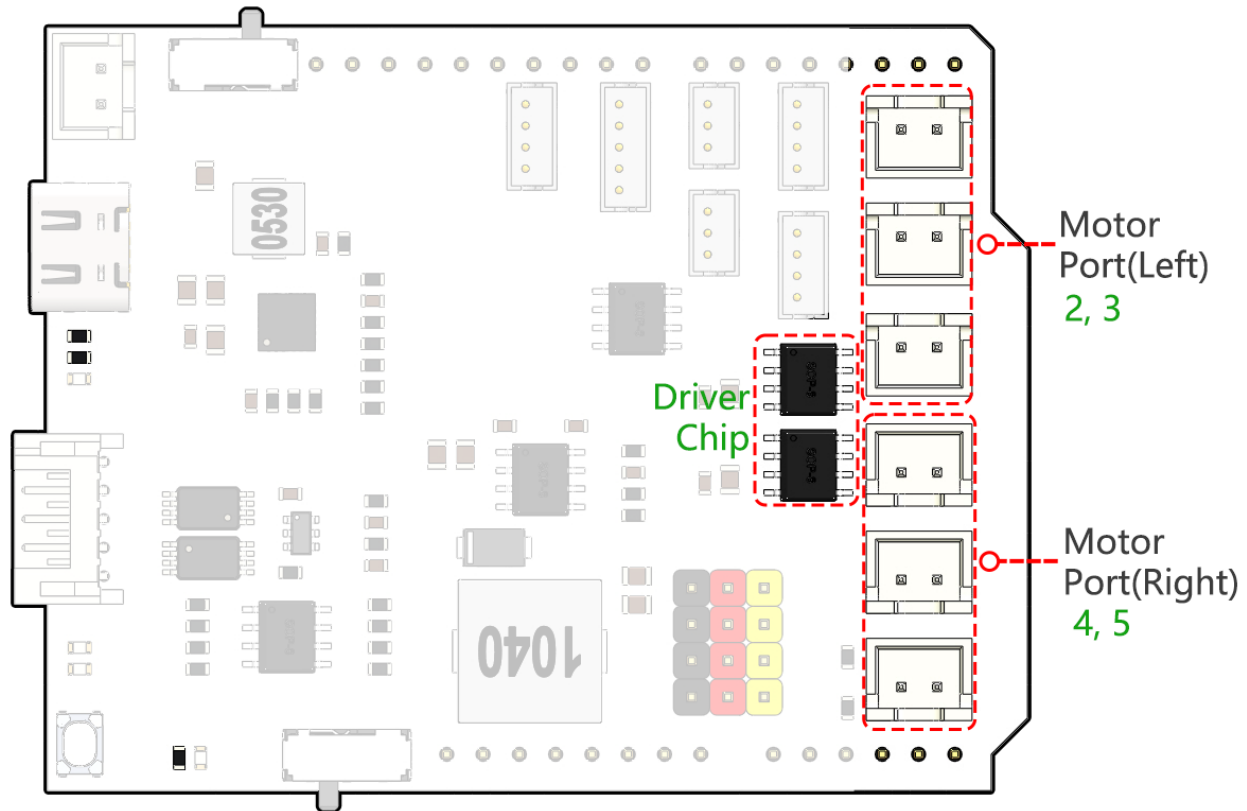


Im nächsten Schritt werden wir uns mit den Einzelheiten des Motor-Treibers befassen und verstehen, wie wir ihn effektiv mit unserem Arduino-Board verwenden können, um einen Motor zu steuern. Bleiben Sie dran für weiteres spannendes Lernen!

### Schritt 3: Wie der Motor durch den Motor-Treiber gesteuert wird

Unser GalaxyRVR-Shield, das im Kit enthalten ist, dient als Kontrollzentrum für unseren Mars-Rover. Es ist der Hub, an den wir alle unsere Sensoren, Motoren und die Stromversorgung anschließen. Es besteht aus mehreren Komponenten, die es uns ermöglichen, unseren Rover effektiv zu steuern und mit Strom zu versorgen.

Auf der rechten Seite des Shields finden Sie sechs Motoranschlüsse. Diese sind jedoch in zwei Gruppen unterteilt, die jeweils von einem separaten Motor-Treiber-Chip gesteuert werden. Drei Anschlüsse, die mit „Links“ gekennzeichnet sind, werden von einem Chip gesteuert, und die anderen drei Anschlüsse, die mit „Rechts“ gekennzeichnet sind, werden von einem anderen gesteuert.



Lernen wir, wie diese beiden Treiber-Chips die sechs Motoren steuern, durch praktische Erfahrung:

- **1. Verbinden des Schaltkreises**

1. Stecken Sie das GalaxyRVR-Shield in das R3-Board, schließen Sie einen Motor an und stecken Sie schließlich den Akku ein, um das Erweiterungsboard mit Strom zu versorgen.
2. Beim ersten Gebrauch wird empfohlen, zuerst ein Type-C USB-Kabel anzuschließen, um den Akku vollständig aufzuladen. Schalten Sie dann den Strom ein.

- **2. Code schreiben und hochladen**

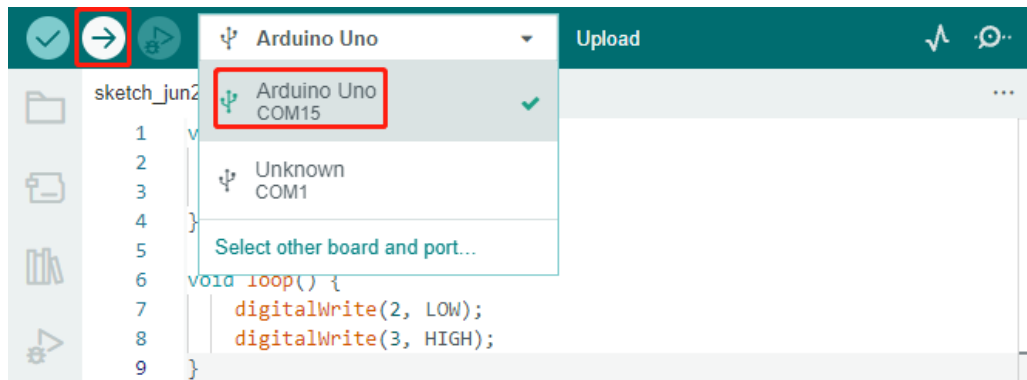
1. Öffne die Arduino IDE und gib den folgenden Code ein:

```
void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
}

void loop() {
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
}
```

- `pinMode()`: Diese Funktion setzt einen Pin als EINGANG oder AUSGANG, vergleichbar mit der Entscheidung, ob eine Figur in unserer Geschichte spricht (OUTPUT) oder zuhört (INPUT).

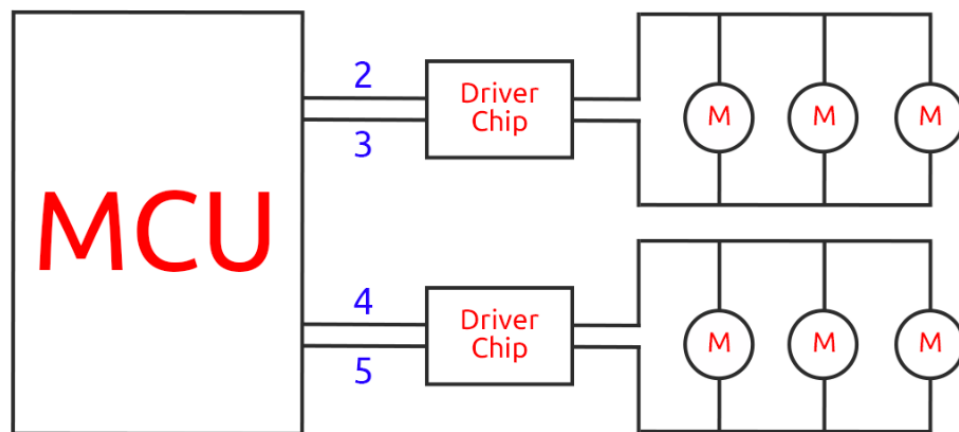
- `digitalWrite()`: Mit dieser Funktion kann ein Pin auf HIGH (on) oder LOW (off) gesetzt werden, ähnlich dem Ein- und Ausschalten eines magischen Lichts.
- 2. Nachdem du das richtige Board (Arduino Uno) und den Port ausgewählt hast, klicke auf den **Upload**-Button. Es ist, als würdest du einen Brief in einen Briefkasten legen - du schickst deine Anweisungen an Arduino!



- 3. Sobald der Code erfolgreich hochgeladen wurde, siehst du, wie sich der Motor im Uhrzeigersinn zu drehen beginnt.

### • 3. Über den internen Schaltungsaufbau

- 1. Du kannst zwei weitere Motoren in die mit „Links“ gekennzeichneten Motoranschlüsse stecken. Du wirst sehen, dass sie sich gleichzeitig drehen.
- 2. Jetzt verstehen wir das einfache Prinzip, wie die beiden Treiber-Chips die sechs Motoren steuern. Pins 2 und 3 auf dem Arduino-Board geben Signale an den Motor-Treiber-Chip aus, und das andere Ende des Chips ist parallel mit drei Motoren verbunden. Ähnlich geben die Pins 4 und 5 Signale an einen anderen Treiber-Chip aus, der wiederum parallel mit drei weiteren Motoren verbunden ist.



- 3. Wenn du einen anderen Treiber-Chip testen möchtest, musst du nur die Pins auf 4 und 5 ändern.

```
const int in3 = 4;
const int in4 = 5;

void setup() {
  pinMode(in3, OUTPUT);
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

    pinMode(in4, OUTPUT);
}

void loop() {
    digitalWrite(in3, LOW);
    digitalWrite(in4, HIGH);
}

```

Hier definieren wir zwei Variablen, um die Pins 4 und 5 zu repräsentieren. Durch die Verwendung von Variablen können wir unsere Pin-Zuweisungen im gesamten Code leicht verwalten und anpassen.

Denk daran, als ob wir jeder Pin-Nummer eine spezifische Rolle oder Aufgabe zuweisen. Wenn wir uns entscheiden, die Rollen neu zuzuweisen, müssen wir anstatt das gesamte Skript zu durchlaufen und jede Instanz zu ändern, lediglich die Zuweisung am Anfang des Skripts aktualisieren (wo die Variable zuerst definiert wird).

#### • 4. Über die Antriebslogik

1. In den vorherigen Tests haben Sie vielleicht bemerkt, dass sich die Motoren alle in eine Richtung drehen. Wie lassen wir sie in die entgegengesetzte Richtung drehen? Jemand könnte vorschlagen, die Zustände HIGH und LOW der beiden Pins zu tauschen. Das ist korrekt.

```

const int in3 = 4;
const int in4 = 5;

void setup() {
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
}

void loop() {
    digitalWrite(in3, HIGH);
    digitalWrite(in4, LOW);
}

```

Sobald Sie Ihren Code geschrieben und auf Ihr Arduino-Board hochgeladen haben, wird sich der Motor entsprechend verhalten.

2. Schauen wir uns jetzt die interne Antriebslogik des Treiberchips an.

INA	INB	Motor
L	L	Standby
L	H	Im Uhrzeigersinn
H	L	Gegen den Uhrzeigersinn
H	H	Bremse

3. Versuchen wir jetzt, den Motor 2 Sekunden lang im Uhrzeigersinn, 2 Sekunden lang gegen den Uhrzeigersinn drehen zu lassen und dann anzuhalten.

```

const int in3 = 4;
const int in4 = 5;

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
void setup() {  
    pinMode(in3, OUTPUT);  
    pinMode(in4, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(in3, LOW);  
    digitalWrite(in4, HIGH);  
    delay(2000);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, LOW);  
    delay(2000);  
    digitalWrite(in3, HIGH);  
    digitalWrite(in4, HIGH);  
    delay(5000);  
}
```

- Hier verwenden wir die Funktion `delay()`, um Arduino eine gewisse Zeit pausieren zu lassen, ähnlich einer kurzen Ruhepause in unserer Geschichte.
- Im Code verwenden wir den Zustand „Brake“ zum Anhalten des Motors, und Sie werden bemerken, dass der Motor abrupt stoppt. Versuchen Sie, beide Pins auf LOW zu setzen, um den Zustand „Standby“ zu testen, und Sie werden feststellen, dass der Motor allmählich langsamer wird, bis er stoppt.

Nun solltest du ein besseres Verständnis davon haben, wie der Motor-Treiber-Chip die Motoren über das GalaxyRVR-Shield steuert und wie wir Arduino-Code verwenden können, um die Bewegungen des Motors zu beeinflussen. Ist es nicht faszinierend, wie ein paar Zeilen Code das Verhalten eines physischen Objekts wie unseres Motors bestimmen können?

Betrachte folgende Fragen, während du weitermachst:

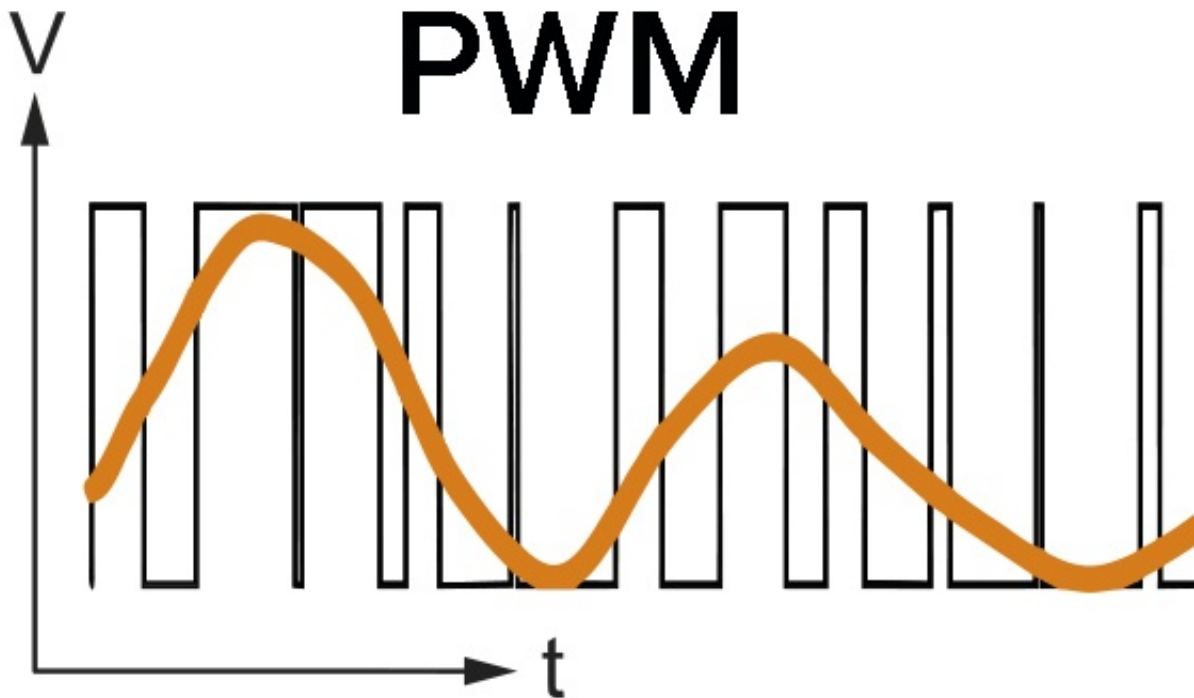
- Wenn wir den gesamten Code aus der Funktion `loop()` in die Funktion `setup()` verschieben, wie würde sich das Verhalten des Motors ändern?
- Wie würdest du den Code ändern, um sechs Motoren gleichzeitig zu steuern?

Denk daran, je mehr du mit deinem Code experimentierst und spielst, desto mehr lernst du. Fühle dich frei, deinen Code nach Belieben anzupassen, zu modifizieren und zu optimieren. Viel Spaß beim Programmieren!

#### Schritt 4: Steuerung der Motorgeschwindigkeit

Im vorherigen Schritt haben wir die Richtung des Motors gesteuert, indem wir seine Pins einfach auf HIGH oder LOW gesetzt haben. Das ist so, als würden wir dem Motor volle Leistung geben, ähnlich wie wenn wir das Gaspedal in einem Auto durchtreten. In vielen Situationen möchten wir jedoch die Motorgeschwindigkeit an unterschiedliche Szenarien anpassen, genau wie wir die Geschwindigkeit eines Autos anpassen, je nachdem, ob wir in der Stadt oder auf der Autobahn fahren. Hier kommt die Pulsweitenmodulation (PWM) ins Spiel.





PWM ist eine Technik, die verwendet wird, um den Effekt einer variablen Spannungsausgabe zu erzeugen, indem der Ausgang schnell zwischen HIGH und LOW geschaltet wird. Mit PWM können wir den Effekt eines analogen Signals simulieren, während wir tatsächlich nur digitale Signale ausgeben.

Du findest das vielleicht schwer zu verstehen, und das ist in Ordnung! Wir werden im Folgenden lernen, wie man die Motorgeschwindigkeit durch Programmierung mit PWM anpasst.

Beachte, dass das SunFounder R3-Board einige Pins mit eingebauter PWM-Funktionalität hat, diese können wir aber für unseren Motor nicht verwenden, da sie bereits andere Funktionen erfüllen. Daher verbinden wir die Treiberchips mit den Pins 2, 3, 4 und 5 und verwenden die SoftPWM-Bibliothek von Arduino, um PWM auf diesen Pins zu ermöglichen.

Das werden wir als Nächstes tun:

1. Öffne die Arduino IDE, suche in der **LIBRARY MANAGER** nach `softpwm` und installiere sie.
2. Gib den folgenden Code in die Arduino IDE ein. Nach dem erfolgreichen Hochladen des Codes wird sich der Motor im Uhrzeigersinn drehen.

```
#include <SoftPWM.h>

const int in1 = 2;
const int in2 = 3;

void setup() {
  SoftPWMBegin();
}
```

(Fortsetzung auf der nächsten Seite)

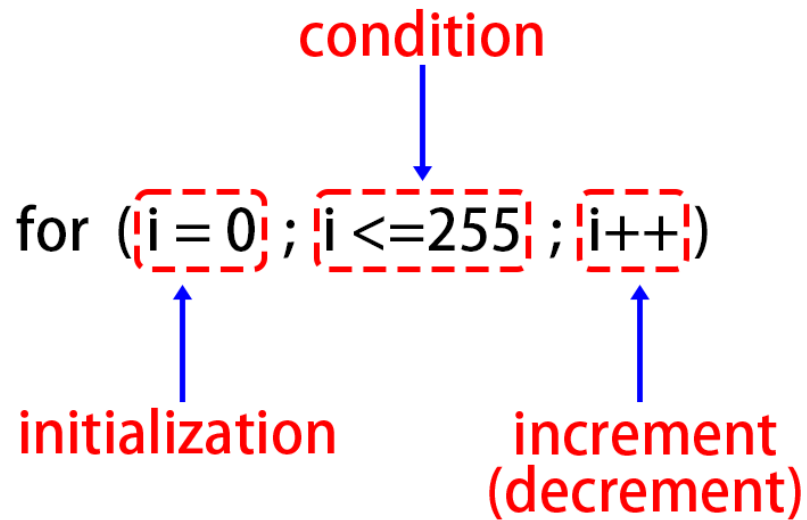
(Fortsetzung der vorherigen Seite)

```
void loop() {  
    SoftPWMSet(in1, 0);  
    SoftPWMSet(in2, 255);  
}
```

- Im obigen Code fügen wir zuerst `SoftPWM.h` oben im Code hinzu, um die Funktionen in der `SoftPWM`-Bibliothek direkt nutzen zu können.
  - Dann initialisieren wir die `SoftPWM`-Bibliothek mit der Funktion `SoftPWMBegin()`.
  - Schließlich verwenden wir in der Funktion `loop()` `SoftPWMSet()`, um verschiedenen Werten an `in1` und `in2` zuzuweisen, was den Motor in Bewegung setzt. Du wirst bemerken, dass der Effekt dem direkten Einsatz von `LOW` und `HIGH` ähnelt, aber hier verwenden wir numerische Werte im Bereich von `0~255`.
  - Denke daran, in der Welt von Arduino wird Geschwindigkeit als ein Wert zwischen 0 (wie ein Auto an einem Stoppschild) und 255 (rasend auf der Autobahn!) ausgedrückt. Wenn wir also sagen `SoftPWMSet(in2, 255)`, sagen wir dem Motor, dass er Vollgas geben soll!
3. Jetzt geben wir andere Werte ein und beobachten, ob sich die Motorgeschwindigkeit ändert.

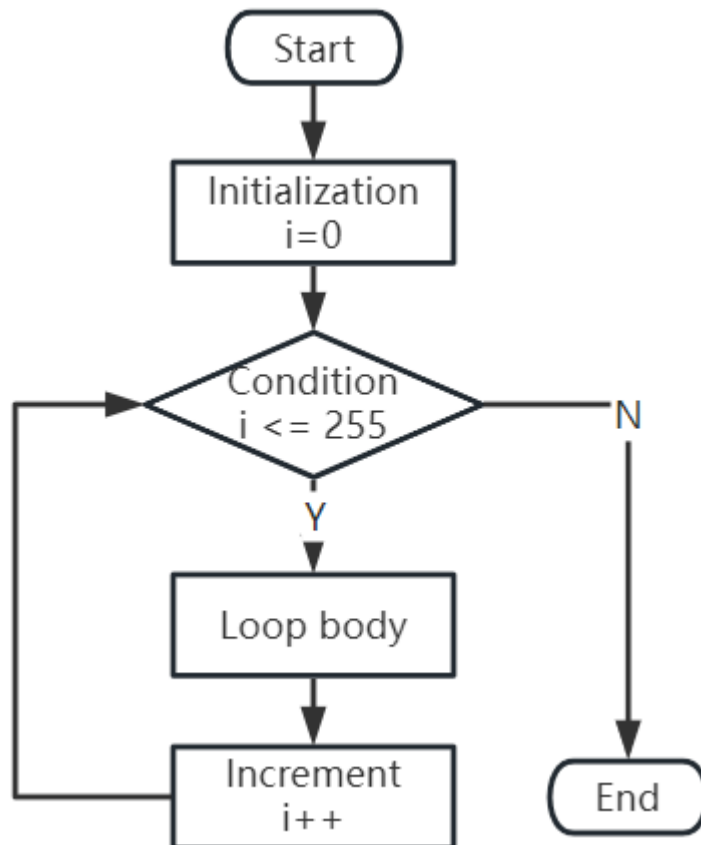
```
#include <SoftPWM.h>  
  
const int in1 = 2;  
const int in2 = 3;  
  
void setup() {  
    SoftPWMBegin();  
}  
  
void loop() {  
    SoftPWMSet(in1, 0);  
    for (int i = 0; i <= 255; i++) {  
        SoftPWMSet(in2, i);  
        delay(100);  
    }  
    delay(1000);  
}
```

Im obigen Code verwenden wir eine `for`-Schleife, um eine Variable `i` bis 255 zu erhöhen. Die `for`-Schleife in der C-Sprache wird verwendet, um einen Teil des Programms mehrmals zu iterieren. Sie besteht aus drei Teilen:



- **Initialization:** Dieser Schritt wird zuerst und nur einmal ausgeführt, wenn wir das erste Mal in die Schleife eintreten. Er ermöglicht es uns, jegliche Schleifensteuerungsvariablen zu deklarieren und zu initialisieren.
- **Condition:** Dies ist der nächste Schritt nach der Initialisierung. Ist sie wahr, wird der Schleifenkörper ausgeführt. Ist sie falsch, wird der Schleifenkörper nicht ausgeführt und die Steuerung des Flusses verlässt die for-Schleife.
- **Increment or Decrement:** Nach Ausführung der Schritte Initialisierung und Bedingung sowie des Schleifenkörper-Codes wird der Schritt Inkrement oder Dekrement ausgeführt. Diese Anweisung ermöglicht es uns, jegliche Schleifensteuerungsvariablen zu aktualisieren.

Das Flussdiagramm für die for-Schleife wird unten gezeigt:



Nach dem Ausführen des obigen Codes wirst du sehen, dass die Motorgeschwindigkeit allmählich zunimmt. Er hält für eine Sekunde an und beginnt dann wieder von 0 und steigt allmählich an.

In diesem Schritt haben wir die Pulsweitenmodulation (PWM) kennengelernt, eine Technik zur Steuerung der Geschwindigkeit unseres Motors. Durch die Verwendung der SoftPWM-Bibliothek von Arduino können wir die Geschwindigkeit des Motors anpassen, was uns ermöglicht, analoge Signale zu simulieren, während wir nur digitale Signale ausgeben. Dies bietet uns eine feinere Kontrolle über die Bewegungen unseres Rovers und bereitet uns auf komplexere Manöver in der Zukunft vor.

### Schritt 5: Reflektieren und Verbessern

Nachdem du diese Lektion abgeschlossen hast, solltest du nun mit den Arbeitsprinzipien von Motoren vertraut sein, sowie damit, wie man ihre Richtung und Geschwindigkeit durch Programmierung steuert.

Teste dein Verständnis mit diesen Herausforderungen:

- Wie würdest du die for-Schleife ändern, um die Motorgeschwindigkeit allmählich zu verringern?
- Wie würdest du den Motor so steuern, dass er beim Drehen gegen den Uhrzeigersinn beschleunigt oder verlangsamt?

Du kannst mit dem bereitgestellten Code experimentieren, um diese Fragen zu beantworten. Fühle dich frei, den Code entsprechend deinen Hypothesen anzupassen und beobachte die Veränderungen im Verhalten des Motors.

Deine praktischen Experimente und Reflexionen über diese Fragen werden dein Verständnis vertiefen und deine Problemlösungsfähigkeiten verbessern. Es ist durch Herausforderungen wie diese, dass echtes Lernen stattfindet. Denke immer daran, es gibt kein „richtig“ oder „falsch“ auf deiner Entdeckungsreise – es geht alles um Lernen und Entdecken!

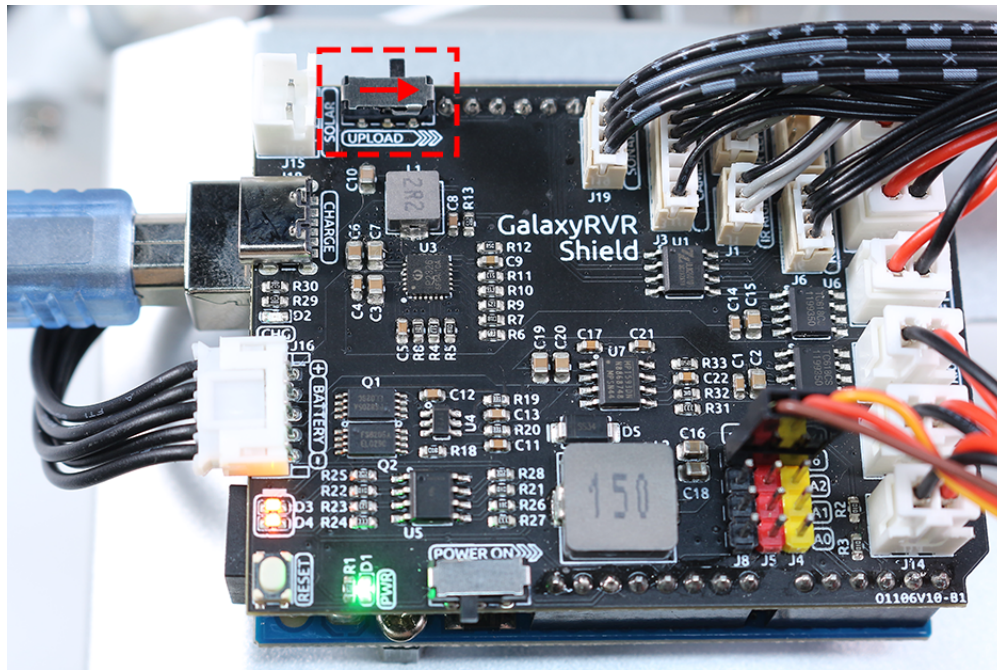
## 3.5 Lektion 5: Entfesselung der Beweglichkeit des Mars Rovers

Macht euch bereit, junge Entdecker! Wir haben gelernt, Motoren zu steuern, und jetzt bringen wir diese Fähigkeiten auf den roten Planeten. In dieser Lektion erwecken wir den Mars Rover zum Leben!

Wir lernen, wie man die Motoren in das Rocker-Bogie-Federungssystem des Rovers einbaut und dann, unter Verwendung unserer Programmierkenntnisse, unseren Rover über imaginäre Marslandschaften steuert.

Es ist ein Marsabenteuer in unserem Klassenzimmer. Lasst uns anfangen!

**Bemerkung:** Wenn du diesen Kurs nach dem vollständigen Zusammenbau des GalaxyRVR lernst, musst du diesen Schalter nach rechts bewegen, bevor du den Code hochlädst.



### 3.5.1 Kursziele

- Verstehe, wie man die Motoren in das Rocker-Bogie-Federungssystem des Mars Rovers einbaut.
- Lerne, wie man Arduino verwendet, um die Bewegung des Mars Rovers zu steuern.
- Übe das Schreiben eines Programms, um die Bewegung des Mars Rovers auf verschiedenen Geländearten zu steuern.

### 3.5.2 Kursmaterialien

- SunFounder R3 Board
- TT-Motoren
- GalaxyRVR-Shield
- Batterie
- Mars Rover Modell (ausgestattet mit Rocker-Bogie-System)
- Grundwerkzeuge und Zubehör (z.B. Schraubendreher, Schrauben, etc.)
- USB-Kabel
- Arduino IDE
- Computer

### 3.5.3 Kursschritte

#### Schritt 1: Zusammenbau der Rover-Komponenten

In diesem Schritt montieren wir die Batterie, das R3-Board, das GalaxyRVR-Shield, die Motoren und die Räder auf das vormontierte Rocker-Bogie-System. Dadurch wird der GalaxyRVR fahrbereit.

Glückwunsch! Wir haben erfolgreich unseren eigenen Rover gebaut, und er ist bereit, die Erkundung zu beginnen. Lasst uns loslegen!

#### Schritt 2: Den Rover in Bewegung setzen

Jetzt ist es an der Zeit, unserem Schöpfung Leben einzuhauchen und sie auf ihre Jungfernfahrt zu schicken. Aber wie kommunizieren wir mit unserem Rover? Wie sagen wir ihm, wohin er gehen und was er tun soll? Da kommen unsere Programmierkenntnisse ins Spiel!

In der realen Welt, wenn wir wollen, dass ein Auto vorwärtsfährt, drücken wir auf das Gaspedal, und beide Räder beginnen sich zu drehen. Die Räder auf der rechten Seite drehen sich im Uhrzeigersinn, während die auf der linken Seite sich gegen den Uhrzeigersinn drehen.

Stell dir vor, du sitzt am Steuer, die Welt rauscht an dir vorbei, während du auf der offenen Straße dahingleitest - genau dieses Erlebnis wollen wir unserem Rover geben.

Jetzt übersetzen wir dieses Erlebnis in die Sprache, die unser Rover versteht - Code!

```
#include <SoftPWM.h>

// Define the pins of motors
const int in1 = 2;
const int in2 = 3;
const int in3 = 4;
const int in4 = 5;

void setup() {
  // Initialize SoftPWM
  SoftPWMBegin();
}
```

(Fortsetzung auf der nächsten Seite)



(Fortsetzung der vorherigen Seite)

```

void loop() {
    // Set the left motors rotate counterclockwise
    SoftPWMSet(in1, 255); // Full speed
    SoftPWMSet(in2, 0);   // Stop

    // Set the right motors rotate clockwise
    SoftPWMSet(in3, 0);   // Stop
    SoftPWMSet(in4, 255); // Full speed
}

```

In diesem Code sprechen wir mit unserem Rover und sagen ihm genau, was er tun soll. Mit der Funktion `SoftPWMSet()` verhalten wir uns wie das Gaspedal und die Bremsen eines Autos, indem wir die Geschwindigkeit und die Richtung jedes Motors steuern. Wir sagen den linken Motoren, sie sollen sich gegen den Uhrzeigersinn drehen, und den rechten Motoren, sie sollen sich im Uhrzeigersinn drehen, und genau so bewegt sich unser Rover vorwärts!

Absolut, das Konzept, den Rover rückwärts zu bewegen, ist unkompliziert, sobald man versteht, wie man ihn vorwärts bewegt. Um den Rover rückwärts fahren zu lassen, müssen wir nur die Drehrichtung jedes Motors umkehren.

So würden wir das im Code umsetzen, wir machen einfach das Gegenteil. Die rechten Räder sollten sich nun gegen den Uhrzeigersinn drehen, und die linken Räder sollten sich im Uhrzeigersinn drehen.

```

#include <SoftPWM.h>

// Define the pins of motors
const int in1 = 2;
const int in2 = 3;
const int in3 = 4;
const int in4 = 5;

void setup() {
    // Initialize SoftPWM
    SoftPWMBegin();
}

void loop() {
    // Set the left motors to rotate clockwise
    SoftPWMSet(in1, 0);   // Stop
    SoftPWMSet(in2, 255); // Full speed

    // Set the right motors to rotate counterclockwise
    SoftPWMSet(in3, 255); // Full speed
    SoftPWMSet(in4, 0);   // Stop
}

```

In diesem Code verwenden wir `SoftPWMSet()`, um den linken Motoren zu sagen, dass sie sich im Uhrzeigersinn drehen sollen und den rechten Motoren, dass sie sich gegen den Uhrzeigersinn drehen sollen.

Ist es nicht faszinierend, dass wir die Reise unseres Rovers einfach mit Code steuern können? Das nächste Mal, wenn du in einem Auto bist, nimm dir einen Moment Zeit, um über die Reise deines Rovers nachzudenken, der die Welt eine Drehung nach der anderen erkundet. Bleib dran, denn die Reise unseres Rovers fängt gerade erst an!

### Schritt 3: Den Rover in andere Richtungen bewegen

Jetzt, wo wir wissen, wie wir unseren Mars Rover vorwärts und rückwärts bewegen können, was ist, wenn wir ihn nach links oder rechts drehen lassen wollen?

Genau wie beim Autofahren im echten Leben gibt es zwei Hauptmethoden, um ein Auto nach links abbiegen zu lassen.

- Die erste Methode besteht darin, die Räder auf der linken Seite langsamer als die auf der rechten Seite rotieren zu lassen. Dieser Geschwindigkeitsunterschied wird den Rover nach links abbiegen lassen.
- Die zweite Methode besteht darin, beide linken und rechten Motoren in dieselbe Richtung rotieren zu lassen (in diesem Fall im Uhrzeigersinn), was den Rover auf seiner Achse nach links drehen lässt.

Lassen Sie uns sehen, wie wir beide Methoden im Code umsetzen können:

### Methode 1: Unterschiedliche Geschwindigkeiten auf jeder Seite

```
#include <SoftPWM.h>

// Define the pins of motors
const int in1 = 2;
const int in2 = 3;
const int in3 = 4;
const int in4 = 5;

void setup() {
    // Initialize SoftPWM
    SoftPWMBegin();
}

void loop() {
    // Set the left motors rotate counterclockwise in low speed
    SoftPWMSet(in1, 40);
    SoftPWMSet(in2, 0);

    // Set the right motors rotate clockwise in higher speed
    SoftPWMSet(in3, 0);
    SoftPWMSet(in4, 200);

    delay(2000); // Last for 2 seconds
}
```

In diesem Code haben wir die Geschwindigkeit der linken Motoren verringert, während wir die rechten Motoren auf einer höheren Geschwindigkeit belassen haben. Dadurch wird der Rover nach links abbiegen.

### Methode 2: Alle Motoren in dieselbe Richtung drehen

```
#include <SoftPWM.h>

// Define the motor pins
const int in1 = 2;
const int in2 = 3;
const int in3 = 4;
const int in4 = 5;

void setup() {
    // Initialize SoftPWM
    SoftPWMBegin();
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
void loop() {  
  // Set all motors to rotate clockwise  
  SoftPWMSet(in1, 0);  
  SoftPWMSet(in2, 255);  
  SoftPWMSet(in3, 0);  
  SoftPWMSet(in4, 255);  
}
```

In diesem Code haben wir alle Motoren so eingestellt, dass sie sich im Uhrzeigersinn drehen. Der Rover wird sich um seine eigene Achse drehen und die Richtung nach links ändern.

Um den Rover nach rechts abbiegen zu lassen, sind die Konzepte dieselben, aber die Richtungen sind umgekehrt. Kannst du herausfinden, wie das geht?

#### Schritt 4: In alle Richtungen bewegen

Wenn wir mehr Funktionen für unseren Mars Rover entwickeln, könnte unser Code ziemlich lang und unübersichtlich werden. In der Programmierung ist es eine gute Praxis, den Code organisiert und wartbar zu halten. Eine Möglichkeit, dies zu erreichen, besteht darin, separate Funktionen für separate Aufgaben zu erstellen.

In diesem Fall können wir separate Funktionen für jede Richtung erstellen, in die sich der Rover bewegen kann. Das macht unseren Code nicht nur leichter verständlich, sondern ermöglicht es uns auch, diese Funktionen überall in unserem Programm wiederverwenden zu können, ohne die gleichen Codezeilen neu schreiben zu müssen.

Lassen Sie uns sehen, wie wir das machen können:

Mit dieser Struktur bleibt unsere Hauptloop einfach und übersichtlich. Wir können deutlich sehen, dass der Rover vorwärts, dann rückwärts fährt, dann nach links und rechts abbiegt und schließlich anhält.

Die Details, wie jede dieser Aktionen erreicht wird, sind in separaten Funktionen versteckt. Dies ist ein wichtiger Aspekt guter Programmierpraktiken, bekannt als Abstraktion.

#### Schritt 5: Teilen und Reflektieren

Herzlichen Glückwunsch zum Abschluss dieser Erkundungsreise mit dem Mars Rover! Das ist eine praktische Anwendung deines Wissens in Wissenschaft und Technik, und du hast fantastische Arbeit geleistet!

Jetzt kannst du teilen, wie dein Rover verschiedene Geländehindernisse überwindet. Du könntest deinen Rover filmen, wie er verschiedene Geländearten navigiert, oder deinen Rover deinen Freunden zeigen.

Das Reflektieren über deinen Lernprozess ist ebenfalls sehr wichtig. Was hast du während dieses Prozesses gelernt? Sind neue Ideen oder kreative Gedanken entstanden? Was hast du im Bereich der Motorensteuerung und Programmierung dazugewonnen?

Du kannst weiterhin erkunden, neues Wissen entdecken und dich ständig verbessern. Mach weiter so, zukünftige Wissenschaftler und Ingenieure!

## 3.6 Lektion 6: Erkundung des Hindernisvermeidungsmoduls

Wir tauchen in die Welt des Infrarot-Hindernisvermeidungsmoduls ein. Versteckt an den Seiten unseres Mars Rovers, agieren diese Sensoren als die „Augen“ des Rovers und helfen ihm, seitliche Hindernisse zu umgehen und sicher durch die Marslandschaft zu navigieren.

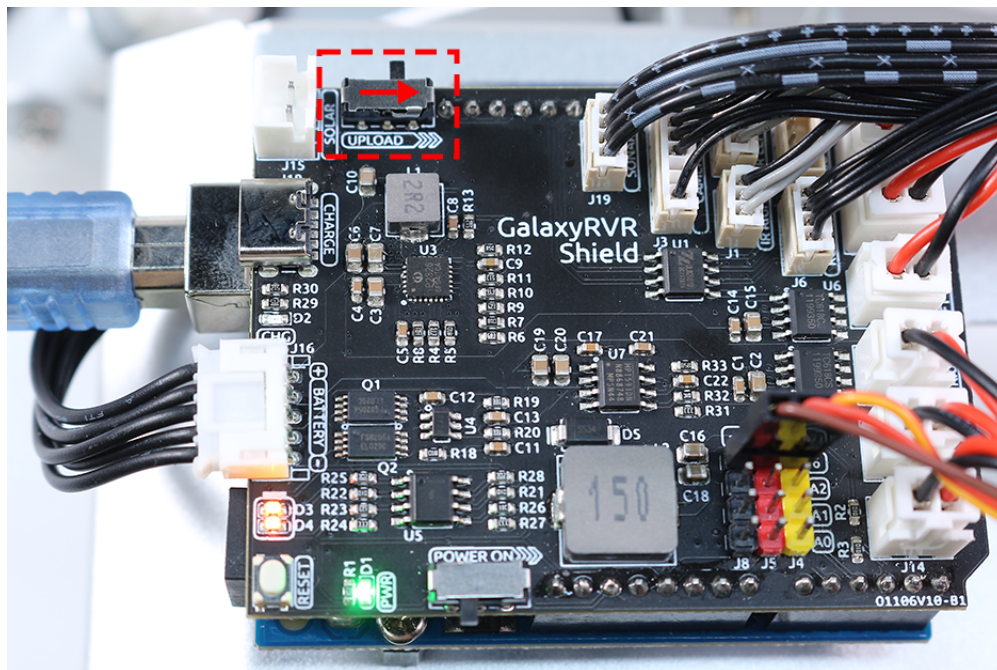
Wir lernen, wie wir diese Module in unseren Rover integrieren, das Geheimnis ihrer Funktion entwirren und Code entwickeln, um unseren Rover geschickt um alle Hindernisse herumzuführen.

Machen wir uns bereit, unseren Rover mit ein wenig marsianischer Hindernisvermeidungsintelligenz auszustatten! Lasst uns loslegen!

---

**Bemerkung:** Wenn du diesen Kurs nach dem vollständigen Zusammenbau des GalaxyRVR lernst, musst du diesen Schalter nach rechts bewegen, bevor du den Code hochlädst.

---



### 3.6.1 Kursziele

- Verstehe das Arbeitsprinzip und die Anwendung des Infrarot-Hindernisvermeidungsmoduls.
- Lerne, wie man Arduino verwendet, um das Infrarot-Hindernisvermeidungsmodul zu steuern.
- Übe das Design und den Bau eines automatischen Hindernisvermeidungssystems basierend auf der Infrarot-Hindernisvermeidung.

### 3.6.2 Kursmaterialien

- Hindernisvermeidungsmodule
- Grundwerkzeuge und Zubehör (z.B. Schraubendreher, Schrauben, Kabel usw.)
- Mars Rover Modell (ausgestattet mit Rocker-Bogie-System, Hauptplatinen, Motoren)
- USB-Kabel
- Arduino IDE
- Computer

### 3.6.3 Kursschritte

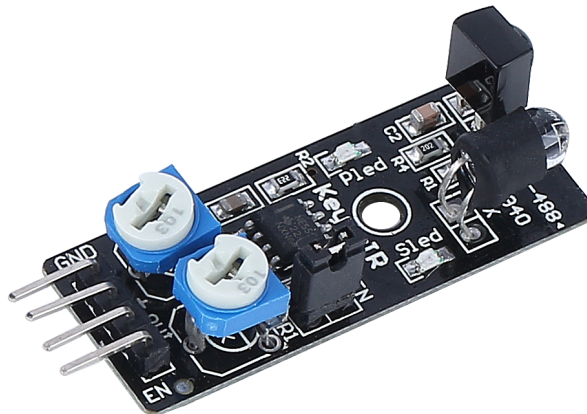
#### Schritt 1: Installiere das Hindernisvermeidungsmodul

Jetzt werden wir die beiden Hindernisvermeidungsmodule am Rover installieren.

Die Montageschritte waren einfach, nicht wahr? In den folgenden Schritten werden wir das Arbeitsprinzip dieser Module kennenlernen und erfahren, wie sie unserem Mars Rover helfen, Hindernisse zu vermeiden. Bleibt dran!

#### Schritt 2: Entmystifizierung des Moduls

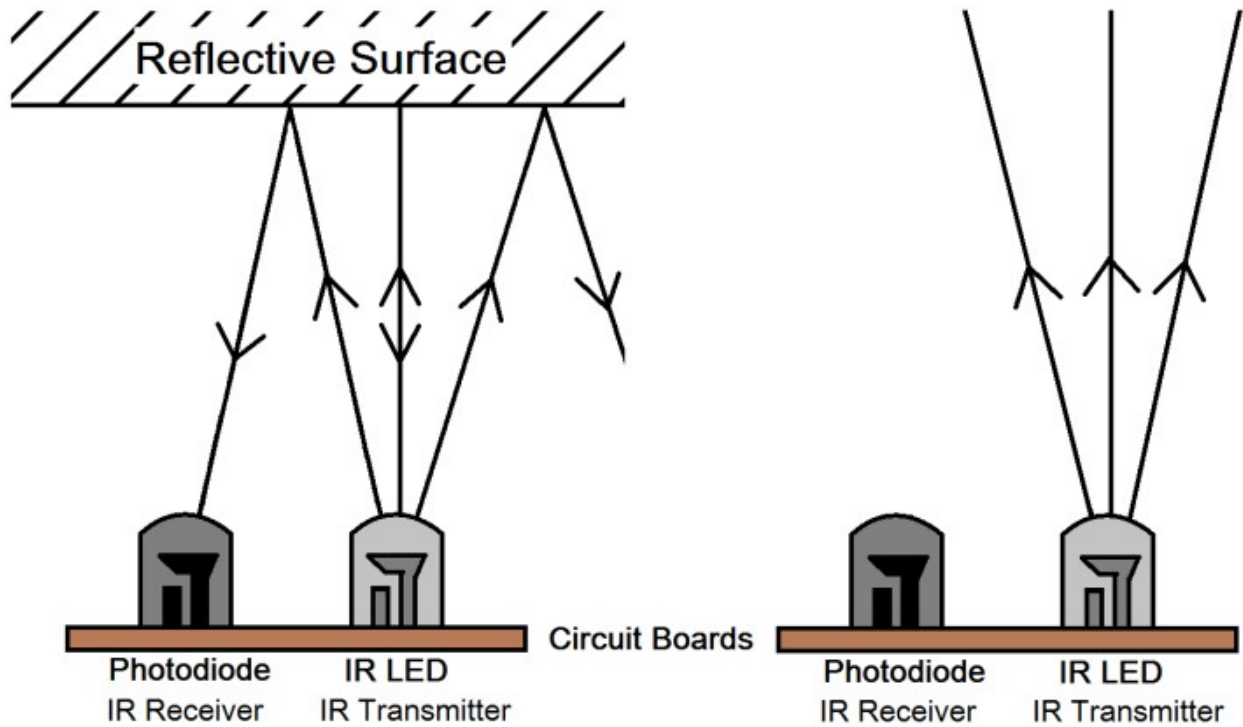
Triff das Infrarot-Hindernisvermeidungsmodul - der schlaue Begleiter unseres Rovers. Dieses kleine Gerät ist ein Wunderwerk. Lass uns eine Tour machen:



Hier sind die Pin-Definitionen:

- **GND:** Das ist wie der Anker des Moduls, das es mit dem Boden oder dem gemeinsamen Punkt im Schaltkreis verbindet.
- **+:** Hier erhält das Modul seine Energie, benötigt eine Stromversorgung von 3,3 bis 5V DC.
- **Out:** Das ist der Kommunikator des Moduls. Standardmäßig bleibt es hoch und wird nur niedrig, wenn es ein Hindernis erkennt.
- **EN:** Treffe den Controller des Moduls. Dieser **Enable**-Pin entscheidet, wann das Modul arbeiten soll. Standardmäßig ist es mit GND verbunden, was bedeutet, dass das Modul immer im Einsatz ist.

Bist du neugierig, wie dieses winzige Modul funktioniert? Es ist ziemlich interessant! Es verwendet ein Paar IR-Komponenten - einen Sender und einen Empfänger. Der Sender ist wie die Taschenlampe des Moduls und emittiert Infrarotlicht. Wenn ein Hindernis auftaucht, wird das Infrarotlicht zurückgeworfen und vom Empfänger aufgefangen. Das Modul gibt dann ein niedriges Signal aus und warnt unseren Rover vor dem Hindernis.



Unser kleines Modul ist ziemlich robust und erkennt Hindernisse in einem Bereich von 2-40 cm und verfügt über ausgezeichnete Störunterdrückungsfähigkeiten. Allerdings wirkt sich die Farbe der Objekte auf die Sensorik aus. Dunklere Objekte, insbesondere schwarze, werden auf kürzere Distanz erkannt. Gegen eine weiße Wand ist der Sensor am effizientesten und erkennt innerhalb eines Bereichs von 2-30 cm.

Der niedrige Zustand des **EN**-Pins aktiviert das Modul, wobei die Jumper-Kappe den **EN**-Pin mit dem GND verbindet. Wenn du den **EN**-Pin über Code steuern möchtest, muss die Jumper-Kappe entfernt werden.



Es gibt zwei Potentiometer auf dem Modul, eines zur Einstellung der Sendeleistung und eines zur Einstellung der Sendefrequenz, und durch Anpassen dieser beiden Potentiometer kannst du seine effektive Distanz anpassen.



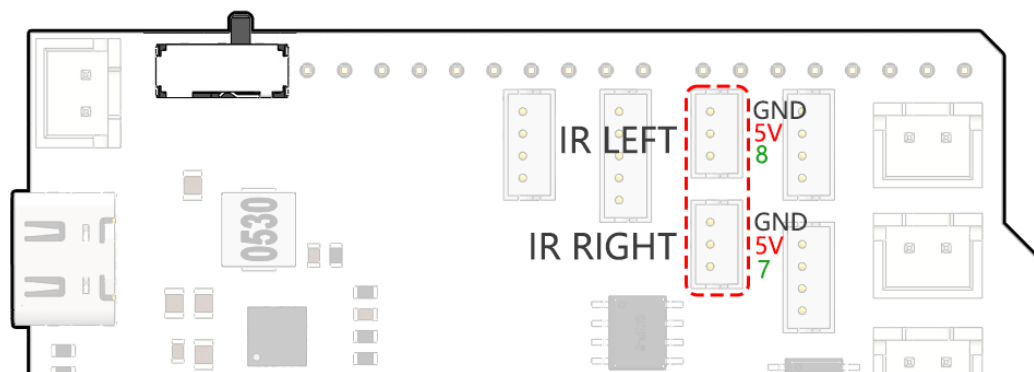


Das war einiges über unser kleines Modul. Im nächsten Schritt lernen wir, wie wir es in unseren Rover integrieren und mit dem Arduino steuern können. Bleibt dran!

### Schritt 3: Auslesen der 2 Module

Wie neugierige Weltraumforscher tauchen wir in das Universum von Codes und Sensoren ein!

1. Unser Mars Rover ist mit zwei speziellen „Alien-Auge“-Sensoren ausgestattet, die ordentlich auf den Pins 7 (right) und 8 (left) angebracht sind. Diese „Alien-Auge“-Sensoren sind tatsächlich unsere Infrarot-Hindernisvermeidungsmodule, immer wachsam, um jegliche „Weltraumfelsen“ (Hindernisse) auf der interstellaren Reise unseres Rovers zu umgehen!



2. Als Nächstes müssen wir mit unserem Rover in der universellen Sprache des Arduino-Codes kommunizieren.

Zuerst geben wir jedem Auge des Rovers einen einzigartigen Namen. Wir nennen sie IR\_RIGHT und IR\_LEFT, so verwechseln wir sie nicht.

```
#define IR_RIGHT 7
#define IR_LEFT 8
```

Jetzt lassen wir unseren Rover wissen, dass dies seine speziellen Augen sind - sie werden Informationen aus der Außenwelt in das elektronische Gehirn des Rovers übermitteln.

```
pinMode(IR_RIGHT, INPUT);
pinMode(IR_LEFT, INPUT);
```

Um sicherzustellen, dass unser Rover seine Erkenntnisse mit uns teilt, richten wir eine geheime Kommunikationslinie ein, wie Spione in einem Sci-Fi-Film. Diese nächste Zeile startet ein seriell Gespräch mit einer Geschwindigkeit von 9600 Bits pro Sekunde - das ist blitzschneller Plauderei!

```
Serial.begin(9600);
```

Nun scannt unser Rover seine Umgebung mit seinen „Alien-Augen“ und leitet die Befunde an uns zurück. Wenn er ein Hindernis entdeckt, ist der Wert 0; wenn der Weg frei ist, ist der Wert 1. Er sendet uns diese Nachrichten ständig, um uns auf dem Laufenden zu halten.

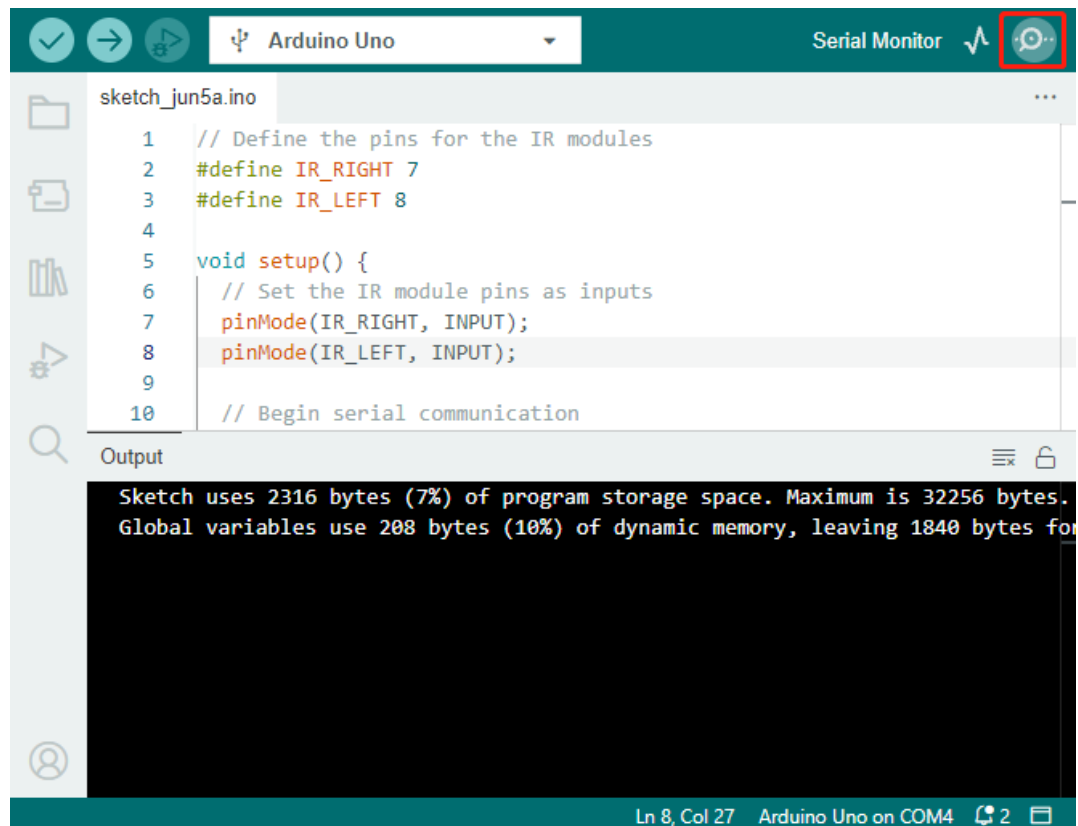
```
int rightValue = digitalRead(IR_RIGHT);  
int leftValue = digitalRead(IR_LEFT);  
Serial.print("Right IR: ");  
Serial.println(rightValue);  
Serial.print("Left IR: ");  
Serial.println(leftValue);
```

Schließlich pausiert der Rover für einen Moment (etwa 200 Millisekunden) nach jeder Übertragung. Diese kleine Pause gibt uns die Chance, die Nachricht des Rovers zu interpretieren, bevor er eine weitere sendet.

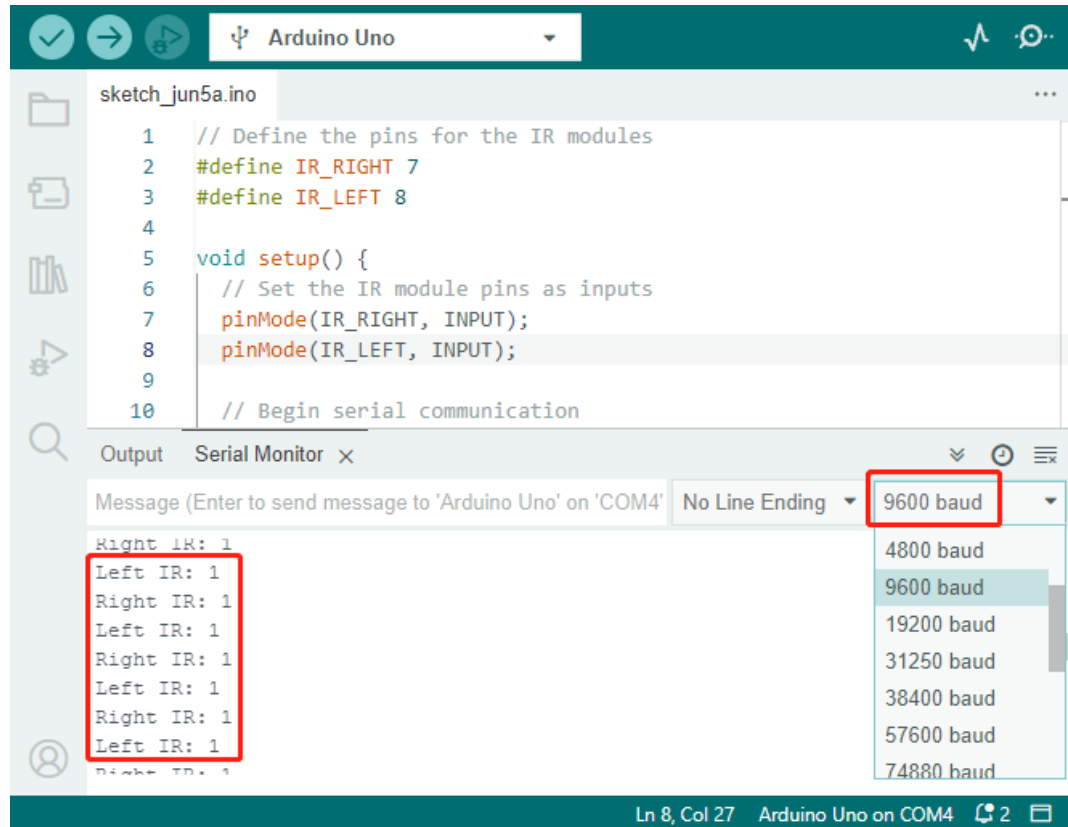
```
delay(200);
```

Hier ist der vollständige Code:

3. Sobald dein Code fertig ist, wähle das richtige Board und den Port aus und sende den Code an deinen Mars Rover. Dann stimme dich in unsere geheime Kommunikationslinie (den Seriellen Monitor) ein, indem du auf sein Symbol in der oberen rechten Ecke klickst.



4. Bevor du die Nachrichten des Rovers empfangst, stelle sicher, dass deine geheime Kommunikationslinie mit derselben Geschwindigkeit (9600 Baud) wie dein Rover abgestimmt ist. Und da hast du es - Live-Updates von deinem Mars Rover!



5. Um unser System auf die Probe zu stellen, winke mit einem „Weltraumfelsen“ (deiner Hand) vor einem der Sensoren. Du wirst sehen, wie der Wert auf 0 umschaltet, und die entsprechende LED auf dem Modul leuchtet auf. Das ist der Rover, der sagt: „Achtung, Weltraumfelsen auf meiner rechten Seite!“

```

Right IR: 0
Left IR: 1
Right IR: 0
Left IR: 1
Right IR: 0
Left IR: 1

```

Bis jetzt bist du nicht nur durch den Weltraum gereist, sondern hast auch Martianisch entschlüsselt! Ich kann es kaum erwarten zu sehen, welche interstellaren Geheimnisse wir in unserer nächsten Mission enthüllen!

#### Schritt 4: Anpassen der Erkennungsdistanz

Wir sind nun bei einem wesentlichen Schritt angelangt: der Anpassung der Erkennungsdistanzen unserer Sensoren an unsere aktuelle Umgebung. Die Werkseinstellungen sind möglicherweise nicht optimal.

Ist die Erkennungsdistanz der beiden Infrarotmodule zu kurz, könnte der Mars Rover mit Hindernissen kollidieren. Ist sie zu weit, könnte der Rover anfangen, sich zu drehen, obwohl er noch weit von einem Hindernis entfernt ist, was seine Bewegung beeinträchtigen könnte.

So kannst du Anpassungen vornehmen:

1. Beginne mit der Anpassung des rechten Hindernisvermeidungsmoduls. Während des Transports können Stöße dazu führen, dass Sender und Empfänger auf dem Infrarotmodul kippen. Daher musst du sie manuell gerade

richten.

2. Platziere ein Hindernis etwa 20 cm direkt vor dem rechten Modul. Die Box, in der unser Rover-Kit kam, ist eine gute Wahl dafür! Drehe nun das Potentiometer am Modul, bis die Anzeigeleuchte am Modul gerade aufleuchtet. Bewege dann das Hindernis hin und her, um zu überprüfen, ob die Anzeigeleuchte in der gewünschten Entfernung aufleuchtet. Leuchtet das Licht nicht in der richtigen Entfernung auf oder bleibt es ohne auszugehen an, musst du das andere Potentiometer einstellen.

3. Wiederhole den gleichen Prozess für das andere Modul.

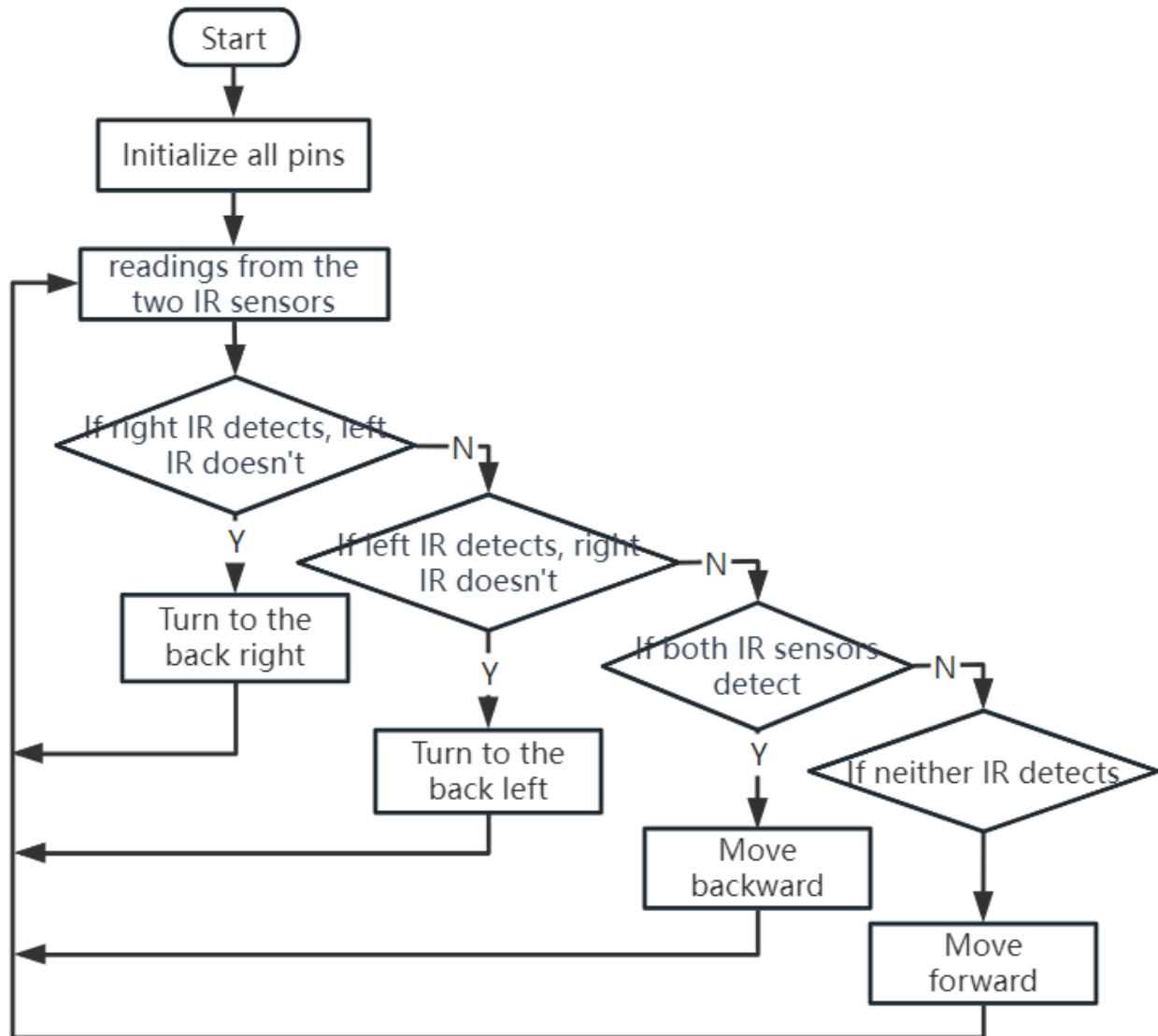
Jetzt, da unsere Sensoren vollständig vorbereitet sind, lasst uns die nächste Reise beginnen!

### **Schritt 5: Entwurf eines automatischen Hindernisvermeidungssystems**

Nun machen wir einen großen Sprung in unserer Weltraumforschung und nutzen diese Nachrichten vom Rover. Wir werden ein automatisches Hindernisvermeidungssystem erstellen!

Hier ist unser Plan: Wenn der rechte Sensor ein Hindernis erkennt, wird der Rover nach hinten rechts abbiegen. Wenn der linke Sensor ein Hindernis erkennt, wird der Rover nach hinten links abbiegen. Wenn beide Sensoren ein Hindernis erkennen, wird der Rover rückwärts fahren. Werden keine Hindernisse erkannt, fährt der Rover geradeaus weiter.

Visualisieren wir dies in einem Flussdiagramm, um es noch klarer zu machen. Flussdiagramme sind eine großartige Möglichkeit, einen Plan logisch zu skizzieren, besonders wenn es um Programmierung geht!



Lassen Sie uns diesen Plan unserem Rover in seiner Sprache (Arduino-Code) zuflüstern:

In diesem Code verwenden wir die `if...else`-Anweisung in der Funktion `loop()`.

Die `if...else`-Anweisung wird verwendet, um einen Block von Code unter zwei Alternativen auszuführen. Wenn wir jedoch unter mehr als zwei Alternativen wählen müssen, verwenden wir die `if...else if...else`-Anweisung.

Die Syntax der `if...else if...else`-Anweisung lautet:

```

if (condition1) {
  // code block 1
}
else if (condition2){
  // code block 2
}
else if (condition3){
  // code block 3
}

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
else {  
  // code block 4  
}
```

Hier,

- Wenn Bedingung1 wahr ist, wird Codeblock 1 ausgeführt.
- Wenn Bedingung1 falsch ist, wird Bedingung2 ausgewertet.
- Wenn Bedingung2 wahr ist, wird Codeblock 2 ausgeführt.
- Wenn Bedingung2 falsch ist, wird Bedingung3 ausgewertet.
- Wenn Bedingung3 wahr ist, wird Codeblock 3 ausgeführt.
- Wenn Bedingung3 falsch ist, wird Codeblock 4 ausgeführt.

Nun, da wir unser automatisches Hindernisvermeidungssystem entworfen haben, kommt der spannende Teil - wir setzen es auf die Probe!

- Beobachten Sie, ob sich der Rover so bewegt, wie Sie es erwartet haben.
- Oder setzen Sie ihn verschiedenen Lichtverhältnissen aus, um zu sehen, wie sich seine Bewegungen ändern.

Indem wir Wissenschaft in unser Ingenieurprojekt integrieren, werden wir zu Weltraumdetektiven, die die Geheimnisse des Verhaltens unseres Rovers lösen. Es geht nicht nur darum, Fehler zu korrigieren, sondern die Leistung zu optimieren, um unseren Rover bestmöglich zu machen! Macht weiter so, Weltraumdetektive!

### Schritt 6: Reflexion und Zusammenfassung

In der Testphase haben Sie vielleicht ein interessantes Verhalten unseres Mars-Rovers bemerkt: Während er geschickt Hindernissen zu seiner Linken und Rechten ausweicht, könnte er Schwierigkeiten haben, kleinere Hindernisse direkt vor ihm zu erkennen.

Wie können wir diese Herausforderung meistern?

Bleiben Sie dran für die nächste Lektion, in der wir unsere Entdeckungsreise in die faszinierende Welt der Programmierung, Sensoren und Hinderniserkennung fortsetzen.

Denken Sie daran, jede Herausforderung ist eine Gelegenheit zum Lernen und zur Innovation. Und während wir unsere Weltraumforschungsreise fortsetzen, gibt es noch so viel mehr zu entdecken und zu lernen!

## 3.7 Lektion 7: Verbesserung der Rover-Navigation mit Ultraschallmodul

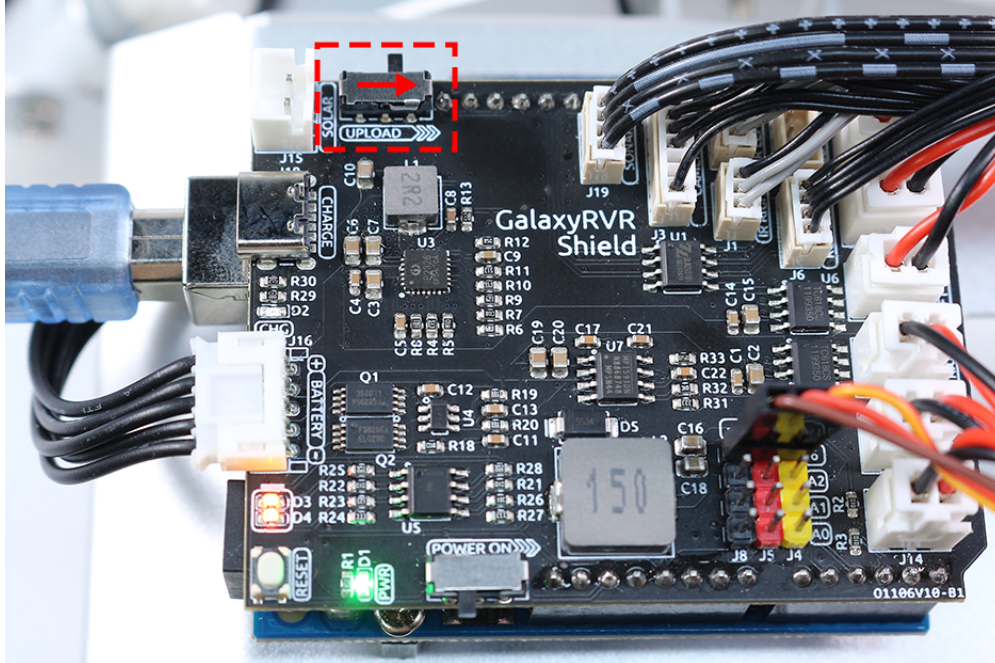
In unserem letzten Abenteuer haben wir unseren Mars-Rover mit „Augen“ an seinen Seiten ausgestattet, wodurch wir ein grundlegendes Hindernisvermeidungssystem geschaffen haben. Doch gibt es einen toten Winkel direkt vorne – eine Herausforderung, die wir bereit sind zu meistern!

Heute, in dieser Lektion, werden wir unserem Rover einen neuen Sinn für „Sehen“ geben. Wir installieren ein Ultraschallsensormodul, das als ein Paar zentraler Augen fungiert, um unserem Rover zu helfen, Hindernisse direkt vor ihm zu erkennen.

Wir werden in die faszinierende Mechanik von Ultraschallwellen eintauchen und erforschen, wie sie die Fähigkeit unseres Rovers verbessern, komplexe Gelände zu navigieren. Mit dieser Ergänzung wird unser Mars-Rover wahrnehmungsfähiger und agiler sein, bereit für ambitioniertere Erkundungsmissionen.

Begleiten Sie uns, wenn wir einen Schritt weiter in diese aufregende STEAM-Reise machen, und unseren Mars-Rover noch geschickter bei der Erkundung unbekannter Gebiete machen!

**Bemerkung:** Wenn Sie diesen Kurs nach der vollständigen Montage des GalaxyRVR durchführen, müssen Sie diesen Schalter vor dem Hochladen des Codes nach rechts bewegen.



### 3.7.1 Kursziele

- Verstehen Sie das Prinzip der Ultraschall-Entfernungsmessung.
- Lernen Sie, wie man Arduino und Ultraschallmodul zur Entfernungsmessung verwendet.
- Üben Sie die Anwendung des Ultraschallmoduls für die Hindernisvermeidung am Mars-Rover-Modell.

### 3.7.2 Benötigte Materialien

- Ultraschallmodul
- Grundwerkzeuge und Zubehör (z. B. Schraubendreher, Schrauben, Kabel usw.)
- Mars-Rover-Modell (ausgestattet mit Schaukel-Bogie-System, Hauptplatinen, Motoren, Hindernisvermeidungsmodul)
- USB-Kabel
- Arduino IDE
- Computer



### 3.7.3 Kursschritte

#### Schritt 1: Montieren Sie das Ultraschallsensormodul

Nun, da wir uns darauf vorbereiten, unseren Mars-Rover mit einem neuen, leistungsstarken „Seh“-Sinn auszustatten, ist es an der Zeit, das Ultraschallsensormodul zusammenzusetzen.

Und da haben wir es! Unser Mars-Rover hat nun ein vollständig montiertes Ultraschallsensormodul, bereit, ihm wie nie zuvor bei der Navigation zu helfen. Sind Sie gespannt, wie es die Hinderniserkennungsfähigkeiten unseres Rovers verändert? Lassen Sie uns direkt eintauchen!

#### Schritt 2: Erkundung des Ultraschallmoduls

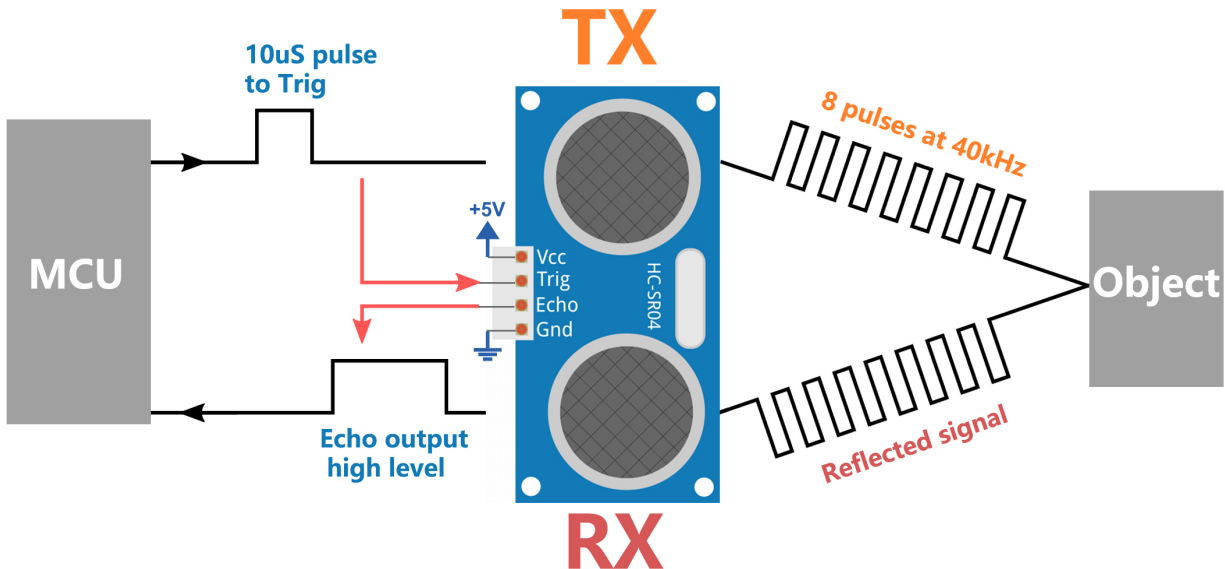
Lernen wir den HC-SR04 kennen, einen leistungsstarken Ultraschall-Entfernungssensor. Dieses winzige Gerät kann Entfernungen von 2 cm bis zu 400 cm genau messen, und das alles ohne Berührung! Erstaunlich, nicht wahr? Es ist wie eine Superheldenkraft! Es kann die Entfernungen nur mithilfe von Schallwellen „sehen“, ähnlich wie eine Fledermaus bei Nacht navigiert.

Es verwendet vier Superkräfte oder besser gesagt, vier Pins, um seine Magie zu wirken:



- **TRIG (Trigger-Puls-Eingang)** - Es ist der Startknopf für unseren Superhelden. Es sagt unserem Superhelden: „Hey, es ist Zeit, eine supersonische Welle auszusenden!“
- **ECHO (Echo-Puls-Ausgang)** - So hört unser Superheld das Echo der Schallwelle, die er ausgesendet hat.
- **VCC** - Auch Superhelden brauchen Energie. Wir verbinden es mit einer 5V-Stromversorgung.
- **GND** - Es ist die Erdverbindung. So wie Superhelden mit der Realität verbunden bleiben müssen!

Stellen Sie sich unseren Superhelden, den HC-SR04 Ultraschallsensor, vor, wie er in den Bergen ein Echo-Spiel spielt.



- Zunächst sagt das Gehirn unseres Superhelden, der MCU, „Ready, Set, Go!“, indem es ein Hochpegelsignal für mindestens 10 Mikrosekunden an unseren Superhelden sendet. Das ist so, als würden wir unsere Energie sammeln, bevor wir in ein Tal hineinrufen.
- Auf das „Go!“ hin ruft unser Superheld achtmal sehr schnell laut heraus. Dieser supersonische Ruf wird mit einer Geschwindigkeit von 40 kHz ausgesendet. Der Superheld startet auch eine Stoppuhr und hält Ausschau nach zurückkehrenden Rufen.
- Gibt es ein Hindernis vorne, trifft der Ruf darauf und hallt zurück. Beim Hören des Echos stoppt unser Superheld die Stoppuhr und notiert die Zeit. Er sendet auch ein Hochpegelsignal aus, um dem MCU mitzuteilen, dass er ein Echo gehört hat.
- Zuletzt, um herauszufinden, wie weit das Hindernis entfernt ist, verwendet unser Superheld eine einfache Formel. Er nimmt die auf der Stoppuhr aufgezeichnete Zeit, teilt sie durch 2 und multipliziert sie mit der Schallgeschwindigkeit (340 m/s). Das Ergebnis ist die Entfernung zum Hindernis!

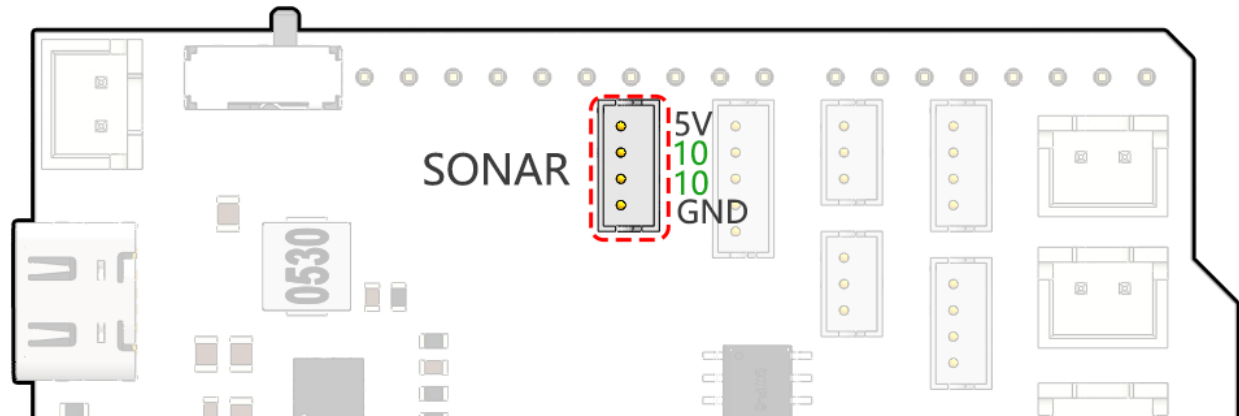
So kann unser Superhelden-Sensor herausfinden, ob ein Hindernis auf seinem Weg ist und wie weit es entfernt ist. Erstaunlich, nicht wahr? Als Nächstes lernen wir, wie wir diese Superheldenkraft in unserem Mars-Rover einsetzen können!

### Schritt 3: Programmierung unseres Superhelden-Sensors

Nachdem wir unseren Superhelden-Sensor zusammengesetzt und verstanden haben, wie er seine Superkräfte einsetzt, ist es an der Zeit, diese Kräfte in Aktion zu setzen! Lassen wir einen Arduino-Sketch schreiben, der es unserem Ultraschallsensor ermöglicht, Entfernungen zu messen und uns diese Messungen anzuzeigen.

Hier sind die Schlüsselschritte, die unser Superhelden-Sensor befolgen wird:

- Wir haben sowohl den TRIG- als auch den ECHO-Pin an Pin 10 auf dem GalaxyRVR Shield angeschlossen. Dies ermöglicht es uns, sowohl die Signalübertragung als auch den Empfang des Ultraschallmoduls mit einem einzigen Arduino-Pin zu steuern.



```
// Define the pin for the ultrasonic module
#define ULTRASONIC_PIN 10
```

- **Vorbereitung:** Zuerst beginnen wir die serielle Kommunikation mit einer Baudrate von 9600, damit wir die Messungen unseres Superhelden in unserem Seriellen Monitor sehen können.

```
void setup() {
  // Start the serial communication
  Serial.begin(9600);
}
```

- **Ruf raus!:** Wir setzen den ULTRASONIC\_PIN als OUTPUT-Pin, um einen 10-Mikrosekunden-Impuls zu senden, was so ist, als würden wir unseren Ultraschall-Superhelden befehlen, einen kraftvollen Ruf (Ultraschall-Schallwellen) auszustoßen!

```
// A 4ms delay is required, otherwise the reading may be 0
delay(4);

//Set to OUTPUT to send signal
pinMode(ULTRASONIC_PIN, OUTPUT);

// Clear the trigger pin
digitalWrite(ULTRASONIC_PIN, LOW);
delayMicroseconds(2);

// Trigger the sensor by sending a high pulse for 10us
digitalWrite(ULTRASONIC_PIN, HIGH);
delayMicroseconds(10);

// Set the trigger pin back to low
digitalWrite(ULTRASONIC_PIN, LOW);
```

- **Warten und Lauschen:** Setzen Sie den ULTRASONIC\_PIN als INPUT. Auf diese Weise ist unser Superhelden-Sensor nun bereit, auf das Echo seines Rufs zu lauschen. Wenn sich ein Hindernis vorne befindet, trifft der Ruf darauf und hallt zurück. Beim Hören des Echos stoppt unser Superheld die Stoppuhr und notiert die Zeit. Er sendet auch ein Hochpegelsignal aus, um dem MCU mitzuteilen, dass er ein Echo gehört hat.

```
pinMode(ULTRASONIC_PIN, INPUT);
float duration = pulseIn(ULTRASONIC_PIN, HIGH);
```

- **Super-Mathematik:** Mit dem zurückgekehrten Echo verwendet unser Sensor die Schallgeschwindigkeit, um die

Entfernung zum Objekt zu berechnen. Wir teilen die Gesamt-Echodauer durch 2, da die Zeit sowohl den Ruf als auch das Warten auf das Echo umfasst.

```
float distance = duration * 0.034 / 2;
```

- **Ergebnisse melden:** Unser Superhelden-Sensor offenbart dann das Ergebnis seiner Mission und zeigt die Entfernung auf dem Seriellen Monitor an, damit wir sie sehen können.

```
// Print the distance to the serial monitor
Serial.print("The distance is: ");
Serial.print(distance);
Serial.println(" cm");
```

- **Ruhe & Bereitschaft:** Jeder Superheld braucht eine Pause, also macht unser Sensor eine kurze Pause vor der nächsten Mission. Dies ermöglicht es dem Sensor, sich „zurückzusetzen“, bevor wir ihn bitten, eine weitere Messung zu starten.

```
delay(200);
```

Hier ist der vollständige Code, der unseren Sensor in einen Superhelden verwandelt:

#### Schritt 4: Programmierung des Ultraschallmoduls zur Steuerung des Mars-Rovers

Jetzt, da wir unseren Mars-Rover mit einem Ultraschallsensormodul ausgestattet haben, ist es an der Zeit, ihn so zu programmieren, dass er basierend auf den Messungen des Sensors reagiert.

- Zur einfacheren Lesbarkeit haben wir eine Funktion namens `readSensorData()` erstellt. Diese Funktion umfasst den gesamten Code, der erforderlich ist, um die Entfernung vom Ultraschallsensor zu lesen.

```
float readSensorData() {
    // A 4ms delay is required, otherwise the reading may be 0
    delay(4);

    //Set to OUTPUT to send signal
    pinMode(ULTRASONIC_PIN, OUTPUT);

    // Clear the trigger pin
    digitalWrite(ULTRASONIC_PIN, LOW);
    delayMicroseconds(2);

    // Trigger the sensor by sending a high pulse for 10us
    digitalWrite(ULTRASONIC_PIN, HIGH);
    delayMicroseconds(10);

    // Set the trigger pin back to low
    digitalWrite(ULTRASONIC_PIN, LOW);

    //Set to INPUT to read
    pinMode(ULTRASONIC_PIN, INPUT);

    // pulseIn returns the duration of the pulse on the pin
    float duration = pulseIn(ULTRASONIC_PIN, HIGH);

    // Calculate the distance (in cm) based on the speed of sound (340 m/s or 0.034
    ↪ cm/us)
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
float distance = duration * 0.034 / 2;

return distance;
}
```

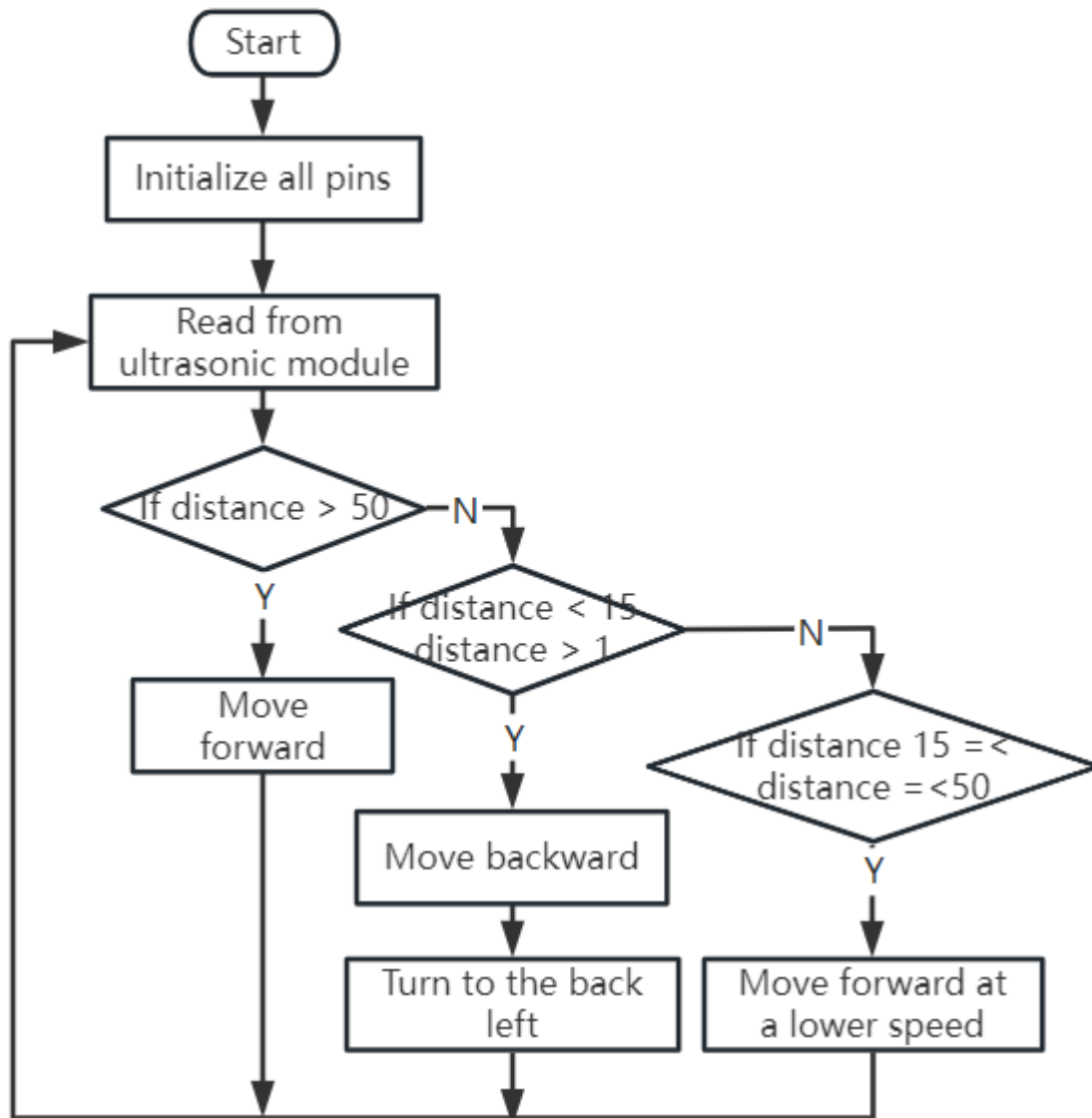
- In der loop()-Funktion rufen wir readSensorData() auf und speichern den zurückgegebenen Wert in der Variable distance.

```
float distance = readSensorData();
```

- Abhängig von dieser Distanz bewegt sich der Rover vorwärts, rückwärts oder hält an.

```
// Control rover based on distance reading
if (distance > 50) { // If it's safe to move forward
  moveForward(200);
} else if (distance < 15) { // If there's an obstacle close
  moveBackward(200);
  delay(500); // Wait for a while before attempting to turn
  backLeft(150);
  delay(1000);
} else { // For distances in between, proceed with caution
  moveForward(150);
}
```

- Wenn der Weg frei ist (das Hindernis mehr als 50 cm entfernt ist), bewegt sich unser Rover mutig vorwärts.
- Und wenn ein Hindernis näher kommt (weniger als 50 cm, aber mehr als 15 cm entfernt), bewegt sich unser Rover mit geringerer Geschwindigkeit vorwärts.
- Wenn ein Hindernis zu nah für Komfort ist (weniger als 15 cm entfernt), bewegt sich der Mars-Rover rückwärts und dann nach links.



Unten finden Sie den kompletten Code. Sie können diesen Code auf das R3-Board hochladen und sehen, ob er den gewünschten Effekt erzielt. Sie können auch die Erkennungsdistanz basierend auf der tatsächlichen Umgebung anpassen, um dieses Hindernisvermeidungssystem zu perfektionieren.

Indem wir diese verbesserten Fähigkeiten nutzen, wäre der Mars-Rover besser in der Lage, potenzielle Hindernisse auf seinem Weg zu identifizieren, Distanzen genau zu messen und fundierte Entscheidungen zu treffen, um sie zu umfahren. Dies würde die Wahrscheinlichkeit von Kollisionen oder anderen Gefahren, die die Erkundungsmission des Rovers behindern könnten, erheblich verringern.

Mit seinen Super-Sinnen kann der Mars-Rover mit größerem Vertrauen und Effizienz operieren, was ihm ermöglicht, tiefer in die Geheimnisse des Mars einzudringen und wertvolle wissenschaftliche Daten für die Forscher auf der Erde zu sammeln.

### Schritt 5: Zusammenfassung und Reflexion

In dieser Lektion haben wir uns mit der Funktionsweise von Ultraschallwellen befasst und wie ihre Rückkehrzeit zum Sensor in messbare Entfernung durch Programmierung übersetzt wird.

Anschließend nutzten wir Ultraschallwellen, um ein Hindernisvermeidungssystem zu entwickeln. Dieses spezielle System variiert seine Reaktionen basierend auf der Entfernung zum bevorstehenden Hindernis.



Nun lassen Sie uns einige Fragen zur Reflexion über diese Lektion stellen:

- Wie erkennt ein Ultraschallmodul Entfernungen? Können Sie das zugrunde liegende Konzept erläutern?
- Wie unterscheidet sich das Hindernisvermeidungssystem dieser Lektion von dem der vorherigen? Was sind ihre jeweiligen Vor- und Nachteile?
- Ist es machbar, diese beiden Hindernisvermeidungssysteme zu kombinieren?

Sich mit diesen Fragen auseinanderzusetzen wird helfen, unser Verständnis zu festigen und uns dazu anregen, über die Anwendung dieser Konzepte auf andere Projekte nachzudenken. Ich freue mich auf unser nächstes Abenteuer!

### 3.8 Lektion 8 Fortgeschrittene Hindernisvermeidung und intelligentes Folgesystem

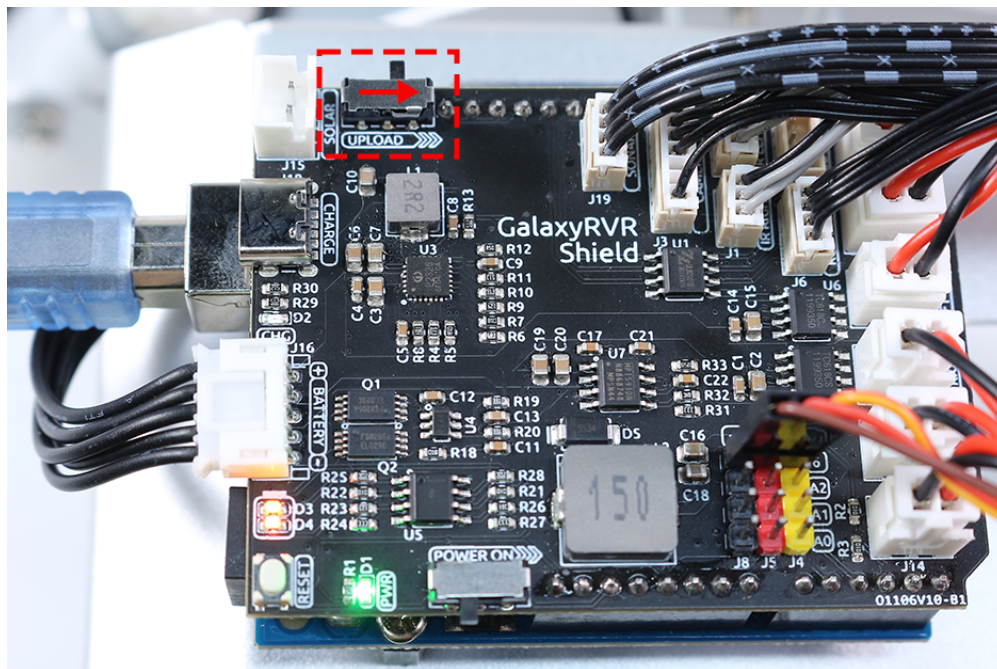
In der heutigen Lektion werden wir unsere STEAM-Fähigkeiten weiterentwickeln. Wir kombinieren ein Hindernisvermeidungsmodul mit einem Ultraschallsensor, um ein fortgeschrittenes Hindernisvermeidungssystem zu erstellen. Außerdem implementieren wir ein intelligentes Folgesystem in unseren Rover.

Am Ende dieser Lektion wird unser Mars-Rover nicht nur in der Lage sein, Hindernisse auf seinem Weg zu vermeiden, sondern auch beweglichen Objekten zu folgen. Stellen Sie sich vor, Sie hätten ein kleines robotisches Haustier, das Ihnen folgt! Aufregend, nicht wahr? Also, lassen Sie uns beginnen.

---

**Bemerkung:** Wenn Sie diesen Kurs nach der vollständigen Montage des GalaxyRVR absolvieren, müssen Sie diesen Schalter nach rechts bewegen, bevor Sie den Code hochladen.

---





### 3.8.1 Kursziele

- Lernen, wie man Hindernisvermeidungsmodule mit einem Ultraschallmodul für verbesserte Navigation kombiniert.
- Verstehen der Prinzipien und Funktionalitäten hinter einem fortgeschrittenen Hindernisvermeidungssystem.
- Erlernen der Implementierung eines intelligenten Folgesystems im Mars-Rover.

### 3.8.2 Kursmaterialien

- Mars-Rover-Modell (das wir in den vorherigen Lektionen gebaut haben)
- USB-Kabel
- Arduino IDE
- Computer
- Und natürlich Ihren kreativen Geist!

### 3.8.3 Kursschritte

#### Schritt 1: Verständnis des Konzepts

Das Hindernisvermeidungsmodul hilft unserem Rover, wie der Name schon sagt, Hindernisse zu vermeiden. Es erkennt Hindernisse, indem es ein Infrarotsignal aussendet und dann das vom Objekt reflektierte Signal empfängt. Ist ein Hindernis vor dem Modul, wird das Infrarotsignal zurückreflektiert, und das Modul erkennt es.

Wenn wir nun einen Ultraschallsensor hinzufügen, verbessert sich dieses System. Ultraschallsensoren messen Entfernungen, indem sie eine Schallwelle mit einer bestimmten Frequenz aussenden und auf das Zurückprallen dieser Schallwelle hören. Indem man die vergangene Zeit zwischen der Erzeugung der Schallwelle und dem Zurückprallen der Schallwelle aufzeichnet, kann man die Entfernung zwischen dem Sensor und dem Objekt berechnen.

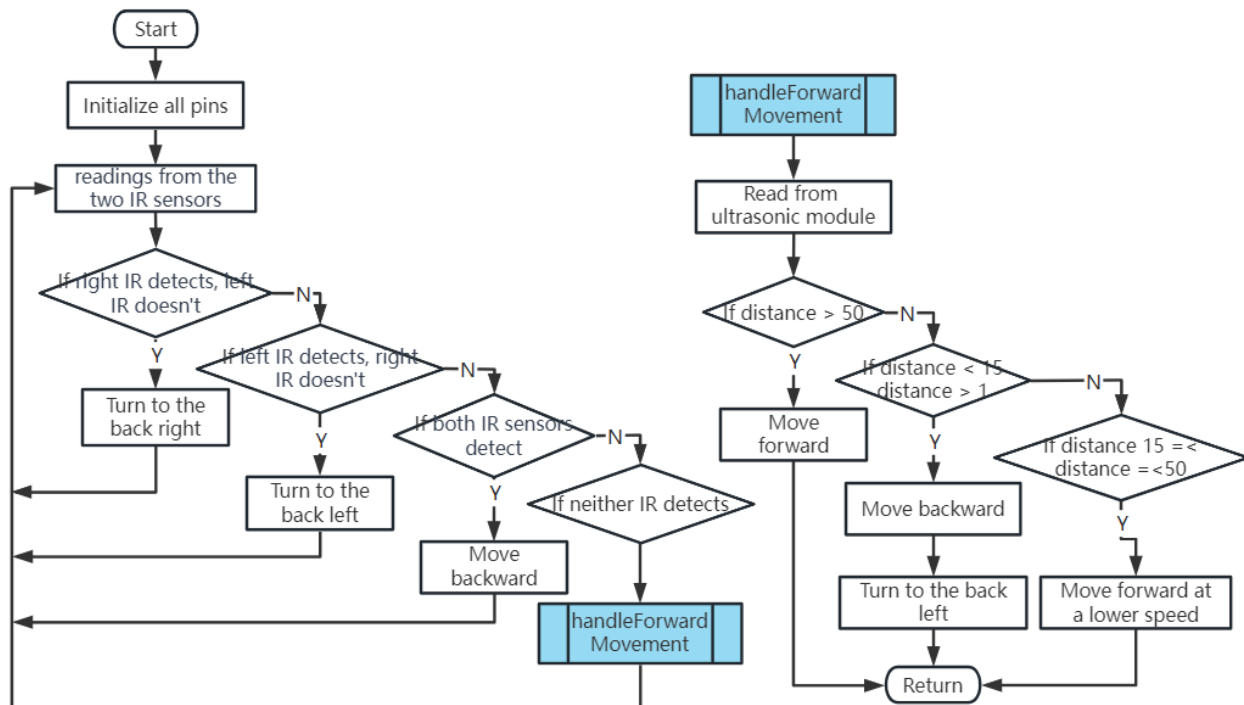
Die Kombination dieser beiden Systeme ergibt ein zuverlässiges, effizientes und vielseitiges Hindernisvermeidungssystem!

#### Schritt 2: Konstruktion fortgeschrittener Hindernisvermeidungssysteme

In unseren vorherigen Lektionen haben wir die Grundlagen der Hindernisvermeidung mit Infrarotsensoren gelernt. Wir haben auch erforscht, wie ein Ultraschallmodul funktioniert. Jetzt werden wir all diese Teile zusammenbringen und ein fortgeschrittenes Hindernisvermeidungssystem bauen!

Unser verbesserter Mars-Rover wird nun sowohl Ultraschall- als auch Infrarotsensoren nutzen, um seine Umgebung zu navigieren.

Lassen Sie uns überlegen, wie die Infrarot- und Ultraschallmodule zusammenarbeiten sollten. Um unsere Logik zu verdeutlichen, verwenden wir ein Flussdiagramm. Das Erlernen der Erstellung von Flussdiagrammen ist ein wertvoller Schritt auf unserer Programmierreise, da es Ihnen helfen kann, Ihre Gedanken zu klären und Ihren Plan systematisch zu skizzieren.



Nun lassen Sie uns dieses Flussdiagramm in tatsächlichen Code umsetzen, um unseren Rover zum Leben zu erwecken.

Beachten Sie, dass die Funktion `handleForwardMovement()` der Ort ist, an dem wir das Verhalten des Ultraschallsensors integriert haben. Wir lesen die Entfernungsdaten vom Sensor und entscheiden auf Basis dieser Daten über die Bewegung des Rovers.

Nachdem Sie den Code auf Ihr R3-Board hochgeladen haben, ist es Zeit, das System zu testen. Stellen Sie sicher, dass der Rover Hindernisse effizient erkennen und vermeiden kann. Denken Sie daran, dass Sie die Erkennungsdistanz im Code basierend auf Ihrer tatsächlichen Umgebung anpassen müssen, um das System zu perfektionieren.

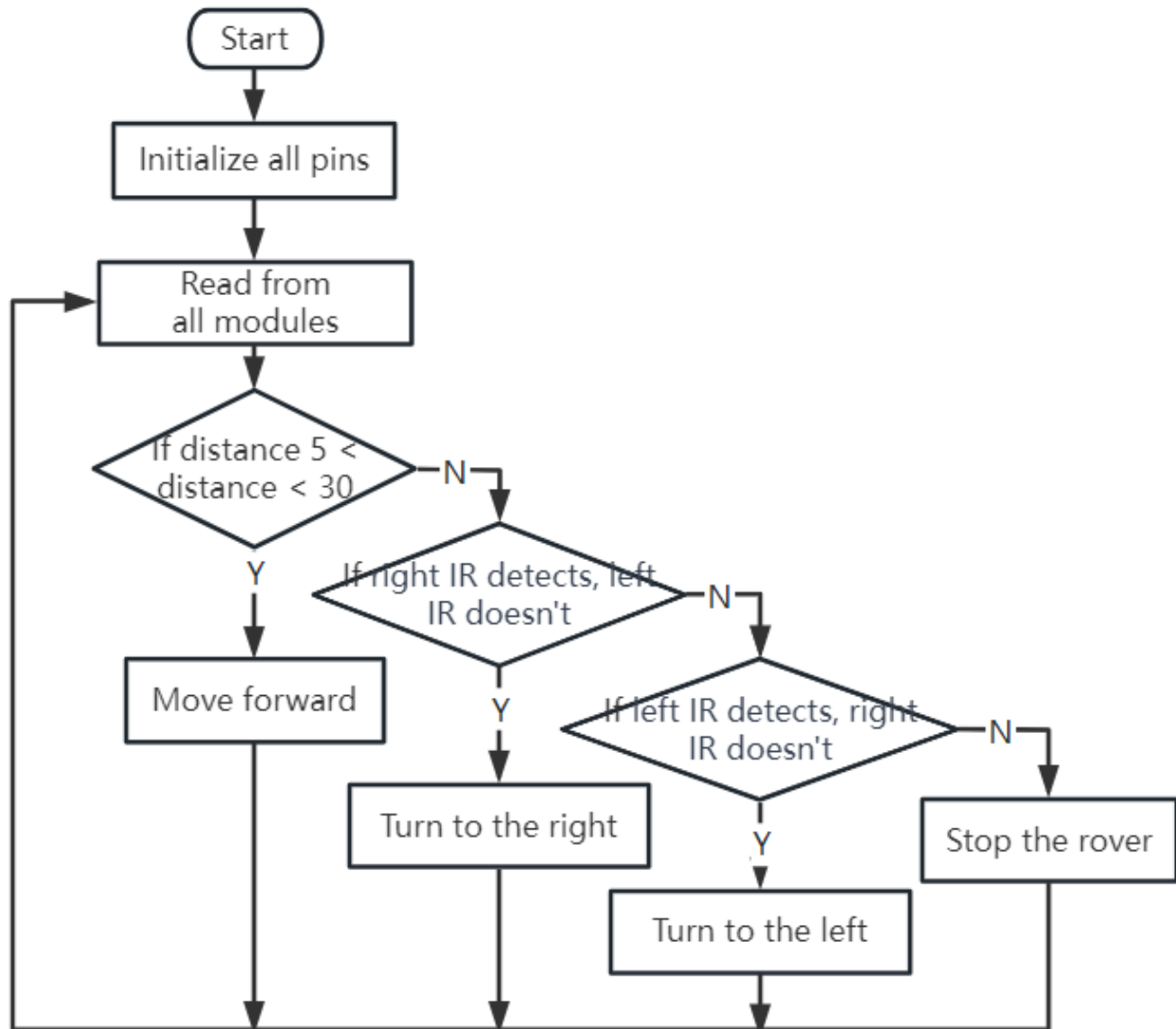
### Schritt 3: Programmierung des intelligenten Folgesystems

Jetzt, da unser Rover in der Lage ist, Hindernissen auszuweichen, wollen wir ihn weiter verbessern, indem wir ihn Objekte verfolgen lassen. Unser Ziel ist es, unseren bestehenden Code zu modifizieren, damit der Rover sich auf ein sich bewegendes Objekt zubewegt.

Haben Sie sich jemals über die Unterschiede zwischen einem Folgesystem und einem Hindernisvermeidungssystem gewundert?

Der Schlüssel liegt darin, dass wir in einem Folgesystem möchten, dass unser Rover auf erkannte Objekte reagiert, während wir in einem Hindernisvermeidungssystem versuchen, erkannten Objekten auszuweichen.

Lassen Sie uns den gewünschten Arbeitsablauf visualisieren:



- Wenn der Ultraschallsensor ein Objekt im Bereich von 5-30 cm erkennt, sollte unser Rover sich darauf zubewegen.
- Wenn der linke IR-Sensor ein Objekt erkennt, sollte unser Rover nach links abbiegen.
- Wenn der rechte IR-Sensor ein Objekt erkennt, sollte unser Rover nach rechts abbiegen.
- In allen anderen Fällen sollte unser Rover stillstehen.

Jetzt ist es an der Zeit, den Code zu vervollständigen.

Sobald der Code fertiggestellt ist, testen Sie, ob der Rover Ihren Bewegungen folgt.

Wie bei dem Hindernisvermeidungssystem wird es entscheidend sein, unser Folgesystem zu testen und eventuell auftretende Probleme zu beheben. Sind Sie bereit zu beginnen?

#### Schritt 4: Zusammenfassung und Reflexion

Heute haben Sie etwas Erstaunliches erreicht. Sie haben verschiedene Module und Konzepte kombiniert, um ein fortgeschrittenes Hindernisvermeidungs- und Folgesystem für Ihren Mars-Rover zu erstellen. Denken Sie daran, das Lernen endet hier nicht - erkunden Sie weiter, innovieren Sie und wenden Sie Ihre neu erworbenen Fähigkeiten auf andere Projekte an.

Denken Sie immer daran, über Ihren Lernprozess zu reflektieren. Überlegen Sie folgendes:

- Warum denken Sie, haben wir im Hindernisvermeidungssystem das Hindernisvermeidungsmodul vor dem Ultraschallsensor priorisiert und umgekehrt im Folgesystem?
- Wie würde sich das Ergebnis ändern, wenn wir die Reihenfolge, in der diese Module im Code überprüft werden, tauschen würden?

Herausforderungen und Probleme sind ein integraler Bestandteil des STEAM-Lernprozesses und bieten wertvolle Möglichkeiten zur Verbesserung. Scheuen Sie sich nicht vor der Fehlersuche – sie ist an sich ein mächtiges Lernwerkzeug!

Auf Ihrem Weg wissen Sie, dass jede überwundene Hürde Sie einen Schritt näher an die Meisterung Ihrer STEAM-Fähigkeiten bringt. Machen Sie weiter und genießen Sie die Reise!

### 3.9 Lektion 9: Den Weg mit RGB-LED-Streifen beleuchten

Auf unserer bisherigen Reise haben wir unseren Mars-Rover in einen intelligenten Entdecker verwandelt, der geschickt um Hindernisse manövriert. Er ist mittlerweile recht geschickt darin, sich auf den Mars-ähnlichen Geländen, die wir für ihn vorbereitet haben, zurechtzufinden.

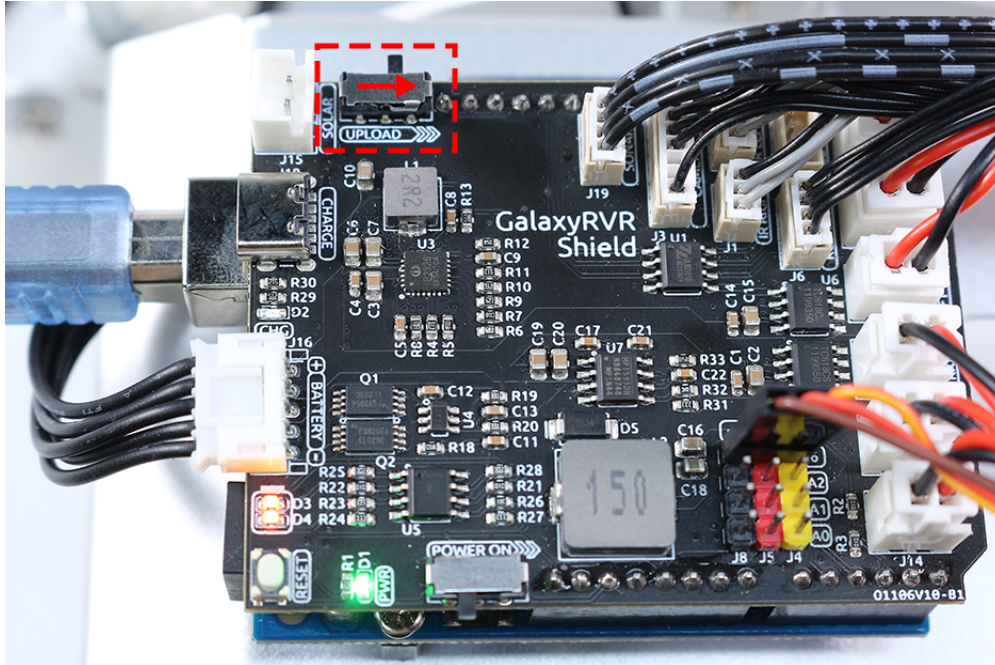
Aber, was wäre, wenn wir seinem praktischen Nutzen ein wenig Flair hinzufügen könnten? Lassen Sie uns unserem Rover die Fähigkeit geben, sich durch ein Spektakel aus Farben und Licht auszudrücken. Wir sprechen davon, RGB-LED-Streifen zu integrieren - ein cooles Feature, das es unserem Rover ermöglichen würde, seinen Weg zu erhellen, selbst unter dunkelsten Bedingungen.

Stellen Sie sich das vor - der Rover hinterlässt eine Spur aus farbcodierten Signalen, die es uns erleichtert, seine Bewegungen zu verstehen. Ein grünes Leuchten, wenn er unterwegs ist, ein strenges Rot, wenn er hält, oder ein auffälliges Gelb bei diesen schnellen Wendungen. Er könnte sogar nur zum Spaß in einem Farbenmeer aufleuchten!

Unser Ziel in dieser Lektion ist es, die Prinzipien von RGB-LED-Streifen zu verstehen, zu lernen, wie man ihre Farbe und Helligkeit steuert und dies dann mit den Bewegungen des Rovers zu synchronisieren. Am Ende wird unser Mars-Rover mehr als eine Maschine sein. Er wird eine leuchtende, farbwechselnde Entität sein, die den Weg in der weiten Marslandschaft weist!

---

**Bemerkung:** Wenn Sie diesen Kurs nach der vollständigen Montage des GalaxyRVR absolvieren, müssen Sie diesen Schalter nach rechts bewegen, bevor Sie den Code hochladen.



### 3.9.1 Kursziele

- Verstehen der Arbeitsprinzipien und Anwendungen von RGB-LED-Streifen.
- Lernen, wie man Arduino-Programmierung verwendet, um die Farbe und Helligkeit von RGB-LED-Streifen zu steuern.
- Übung im Installieren und Verwenden von RGB-LED-Streifen am Mars-Rover-Modell als Anzeiger.

### 3.9.2 Kursmaterialien

- RGB-LED-Streifen (jeder Streifen hat 8 RGB-LEDs, insgesamt zwei Streifen)
- Grundwerkzeuge und Zubehör (z. B. Schraubendreher, Schrauben, Kabel usw.)
- Mars-Rover-Modell (ausgestattet mit Schaukel-Bogie-System, Hauptplatten, Motoren, Hindernisvermeidungsmodul, Ultraschallmodul)
- USB-Kabel
- Arduino IDE
- Computer

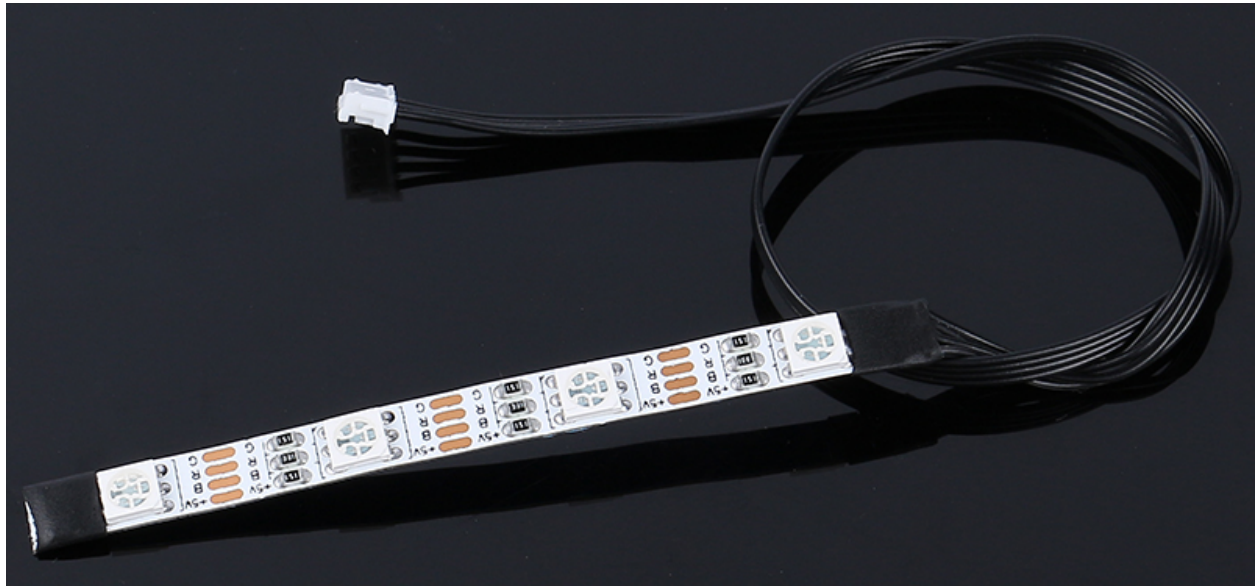
### 3.9.3 Kursschritte

#### Schritt 1: Installieren der RGB-LED-Streifen am Mars-Rover

Fixieren Sie nun die beiden RGB-Lichtstreifen an den unteren Seiten des Wagens. Sie werden von einem einzigen Satz Pins gesteuert, daher ist während des Verdrahtungsprozesses keine Unterscheidung notwendig.

#### Schritt 2: Erforschen Sie die Magie des Lichts mit RGB-LED-Streifen

Erinnern Sie sich an das letzte Mal, als Sie einen Regenbogen gesehen haben? Wie er den Himmel mit sieben lebendigen Farben bunt gemacht hat? Wie wäre es, wenn Sie hier bei unserem kleinen Mars-Rover Ihren eigenen Regenbogen kreieren könnten? Lassen Sie uns in die Magie des Lichts mit RGB-LED-Streifen eintauchen!



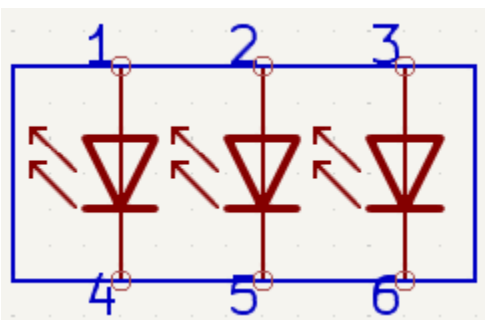
Sie werden bemerken, dass unser RGB-LED-Streifen vier Pins wie folgt beschriftet hat:

- **+5V**: Dies ist das gemeinsame „positive“ Ende oder die „Anode“ der drei kleinen Glühbirnen (LEDs) in unserem Streifen. Es muss mit DC 5V verbunden werden, einer Art elektrischem Saft, der unsere winzigen Glühbirnen antreibt!
- **B**: Dies ist das „negative“ Ende oder die „Kathode“ der blauen LED.
- **R**: Dies ist die „Kathode“ der roten LED.
- **G**: Dies ist die „Kathode“ der grünen LED.

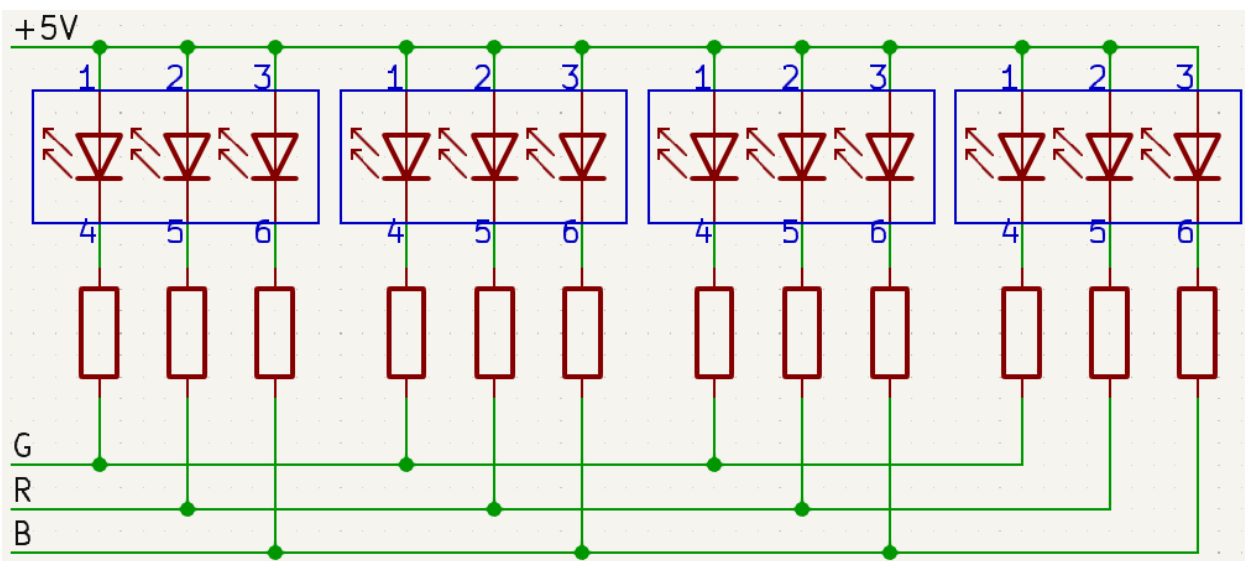




Erinnern Sie sich an die drei Grundfarben - Rot, Blau und Grün - die wir im Kunstunterricht gelernt haben? Genau wie ein Künstler diese Farben auf seiner Palette mischt, um neue Schattierungen zu kreieren, enthält unser Streifen 4 „5050“ LEDs, die diese Grundfarben mischen können, um praktisch jede Farbe zu erzeugen! Jede „5050“ LED ist wie ein winziges Kunststudio, das diese drei farbigen Glühbirnen beherbergt.



Diese winzigen Kunststudios sind dann auf einer flexiblen Platine clever verbunden - quasi wie eine Mini-Elektroautobahn! Die „positiven“ Enden aller LEDs (Anoden) sind miteinander verbunden, während die „negativen“ Enden (Kathoden) mit ihren entsprechenden Farbspuren verbunden sind (G zu G, R zu R, B zu B).





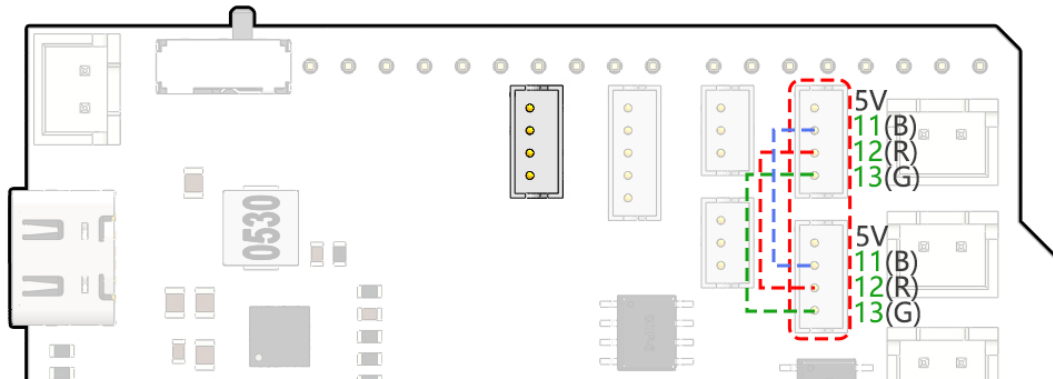
Und das coolste daran? Auf unseren Befehl hin können alle LEDs auf diesem Streifen gleichzeitig ihre Farben ändern! Es ist, als hätten wir unser eigenes Lichtorchester an der Spitze unserer Finger!

Also bereiten wir uns darauf vor, etwas Lichtmusik zu spielen! Im nächsten Schritt lernen wir, wie wir diese LEDs steuern können, um die gewünschten Farben anzuzeigen. Es wird sein, als würden wir eine Symphonie des Lichts dirigieren!

### Schritt 3: Die Show erleuchten - Programmierung zur Steuerung der RGB-LED-Streifen

Wir sind in das Reich der Farben eingetreten, es ist Zeit, unseren Mars-Rover zum Leben zu erwecken. Machen Sie sich bereit, die Dunkelheit mit einem Farbspektrum zu bemalen, das mit RGB-LED-Streifen erzeugt wird. Denken Sie daran, dies als Chance zu sehen, Ihren Mars-Rover in eine mobile Diskoparty zu verwandeln!

- Bevor wir in den spaßigen Teil eintauchen, sollten wir verstehen, dass wir zwar zwei LED-Streifen haben, diese jedoch beide von denselben Pins gesteuert werden. Denken Sie daran, als hätten Sie zwei umwerfende Tänzer, die sich in perfekter Synchronisation bewegen!



- Jetzt ist es an der Zeit, unsere Programmiermagie zu beschwören. Wir werden unsere Pins mit dem Arduino-Code initialisieren.

```
#include <SoftPWM.h>

// Define the pin numbers for the RGB strips
const int bluePin = 11;
const int redPin = 12;
const int greenPin = 13;
```

- Mit unseren Pins an ihrem Platz werden wir jetzt die Funktion `SoftPWMSet()` verwenden, um diese Pins zu steuern. Um den RGB-Streifen rot anzeigen zu lassen, schalten wir die rote LED ein und die anderen aus.

```
void setup() {
    // Initialize software-based PWM on all pins
    SoftPWMBegin();
}

void loop() {
    // Set the color to red by turning the red LED on and the others off
    SoftPWMSet(redPin, 255); // 255 is the maximum brightness
    SoftPWMSet(greenPin, 0); // 0 is off
    SoftPWMSet(bluePin, 0); // 0 is off
    delay(1000); // Wait for 1 second
}
```

Im obigen Code haben wir nur demonstriert, wie man eine einzelne Farbe anzeigt.

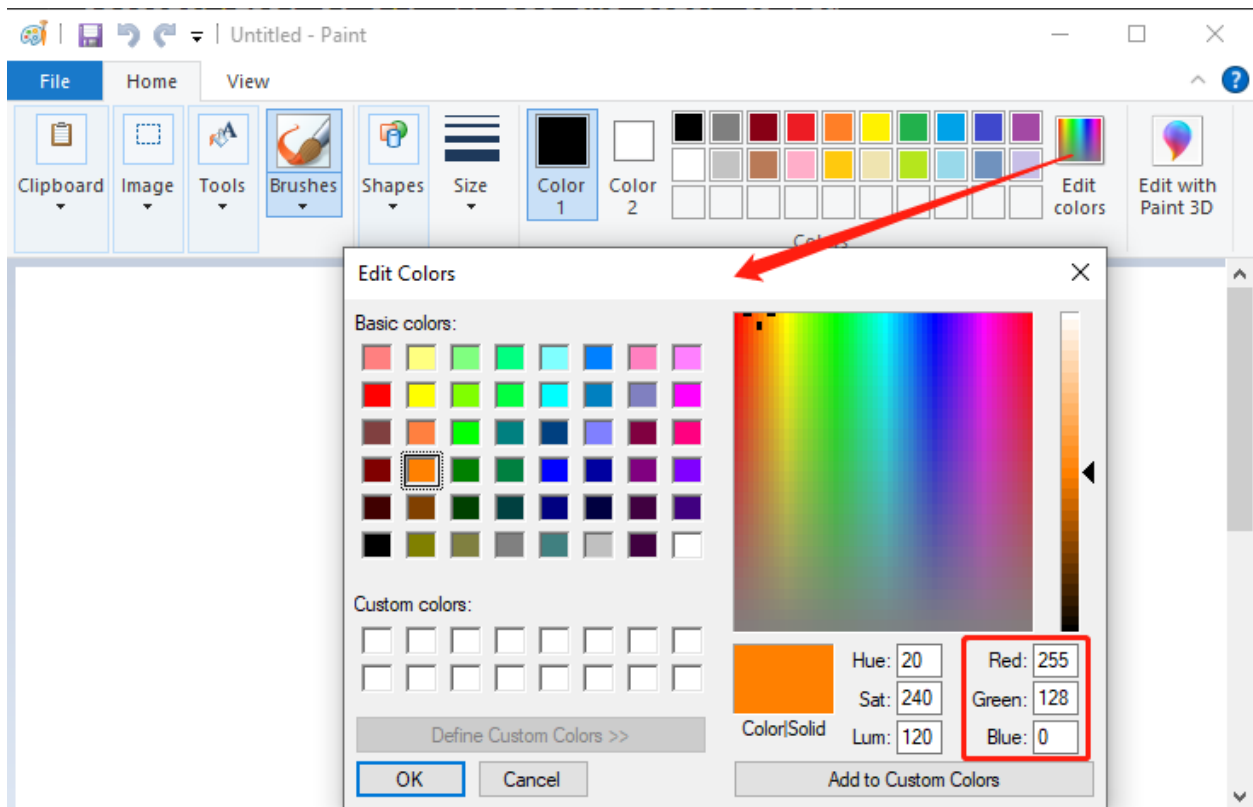
Wenn wir eine Vielzahl von Farben mit dieser Methode präsentieren würden, könnte der Code recht umständlich werden. Daher, um unseren Code prägnanter und wartbarer zu machen, können wir eine Funktion erstellen, um PWM-Werte den drei Pins zuzuweisen. Dann können wir innerhalb der `loop()`-Funktion ganz einfach eine Vielzahl von Farben einstellen.

Nachdem Sie den Code auf Ihr R3-Board hochgeladen haben, stellen Sie möglicherweise fest, dass die Farben Orange und Gelb etwas abweichen. Dies liegt daran, dass die rote LED auf dem Streifen im Vergleich zu den anderen beiden LEDs relativ schwach leuchtet. Daher müssen Sie in Ihrem Code Offset-Werte einführen, um diese Farbabweichung zu korrigieren.

Nun sollte der RGB-LED-Streifen die richtigen Farben anzeigen können. Wenn Sie immer noch Abweichungen bemerken, können Sie die Werte von `R_OFFSET`, `G_OFFSET` und `B_OFFSET` manuell anpassen.

Fühlen Sie sich frei zu experimentieren und jede beliebige Farbe Ihrer Wahl auf dem LED-Streifen anzuzeigen. Alles, was Sie tun müssen, ist, die RGB-Werte für die gewünschte Farbe einzugeben.

Hier ein Tipp: Sie können das Malprogramm auf Ihrem Computer verwenden, um die RGB-Werte Ihrer gewünschten Farbe zu ermitteln.



Jetzt, wo wir die Kunst des Farbsettings gemeistert haben, werden wir im nächsten Schritt diese schillernden Displays mit den Bewegungen des Mars-Rovers integrieren. Aufregende Zeiten stehen bevor!

#### Schritt 4: Bewegung des Rovers mit Farbanzeige

Nun werden wir Farbanzeigen zu den Bewegungen des Mars-Rovers hinzufügen. Beispielsweise können wir Grün für Vorwärtsbewegungen, Rot für Rückwärtsbewegungen und Gelb für das Abbiegen nach links oder rechts verwenden.

Um dies zu tun, werden wir einen Steuermechanismus in unserem Code hinzufügen, der die Farbe des LED-Streifens basierend auf der Bewegung des Rovers einstellt. Dies wird beinhalten, unseren Rover-Steuerungscode zu modifizieren, um unsere Farbsteuerungsfunktionen einzubeziehen.

Lassen Sie uns ein Beispiel sehen, wie wir das machen können:

Innerhalb der `loop()`-Funktion haben wir den Rover beauftragt, eine Reihe von Aktionen durchzuführen, indem wir verschiedene Funktionen aufrufen. Jede Aktion hatte ihre entsprechende Farbanzeige - Grün für Vorwärtsbewegungen, Rot für Rückwärtsbewegungen und Gelb für das Abbiegen. Diese Farbanzeigefunktion wurde mit der Funktion `setColor()` zum Leben erweckt, welche die Helligkeit jedes RGB-Farbkanals steuerte.

Für die Stop-Aktion haben wir ein fesselndes Element eingeführt - einen Atmungseffekt mit rotem und blauem Licht. Dies wurde erreicht, indem die Helligkeit der roten und blauen Kanäle in der Funktion `stopMove()` zyklisch angepasst wurde. So wechselte der LED-Streifen beim Anhalten die Farben zwischen Rot und Blau und schuf einen dynamischen visuellen Effekt.

Nun besitzt unser Mars-Rover seine eigenen lebhaften Farbeffekte und hinterlässt eine Spur aus farbcodierten Signalen, von denen jedes eine einzigartige Bewegung darstellt.

Durch dieses Projekt haben wir entdeckt, wie STEAM-Fächer verschmelzen können, um einem ansonsten gewöhnlichen Gerät Leben einzuhauchen, und es in ein lebendiges, interaktives und unterhaltsames Lernwerkzeug zu verwandeln.

### Schritt 5: Zusammenfassung und Reflexion

In der heutigen Lektion haben wir uns in die Welt der RGB-LED-Streifen vertieft und erforscht, wie man sie manipuliert, um eine lebendige Palette von Farben anzuzeigen. Diese brillanten Farbtöne hauchten unserem Mars-Rover neues Leben ein und verwandelten ihn von einer bloßen Maschine in ein lebendiges Spektakel.

Nun lade ich Sie ein, nachzudenken - Wenn Sie am Steuer säßen, wie würden Sie diese Farben nutzen, um Ihren Mars-Rover zu verbessern? Welche einzigartigen Effekte möchten Sie, dass er zeigt?

Darüber hinaus hoffe ich, dass Sie durch den Prozess ein praktisches Verständnis dafür gewonnen haben, wie verschiedene STEAM-Konzepte in einem ansprechenden Projekt verwoben werden können, und Ihnen eine breitere Perspektive ihrer praktischen Anwendungen bieten.

Wir sehen uns in unserem nächsten spannenden Abenteuer!

## 3.10 Lektion 10: Erkundung des visuellen Systems des Mars-Rovers - Servo und Kippmechanismus

Willkommen zurück, junge Entdecker! In unserem heutigen Abenteuer werden wir in die faszinierende Welt des visuellen Systems des Mars-Rovers eintauchen. Genau wie unsere Augen und unser Hals zusammenarbeiten, um uns zu helfen, unsere Umgebung zu sehen und zu navigieren, benötigt unser Rover ein ähnliches System, um die tückische Marslandschaft zu durchqueren. Und genau das werden wir heute bauen!

Das visuelle System unseres Rovers besteht aus zwei Hauptteilen: einer Kamera, die als seine „Augen“ dient, und einem Kippmechanismus, der wie ein „Hals“ fungiert, der es ihm ermöglicht, nach oben und unten zu schauen. Bis zum Ende dieser Lektion werden wir unserem Rover die Fähigkeit geben, zu „sehen“ und zu „nicken“!

Zuerst werden wir den Kippmechanismus bauen - ein Gerät, das die Kamera unseres Rovers hält und es ihr ermöglicht, sich vertikal zu drehen. Es ist, als würden wir unserem Rover einen Hals geben, damit er seinen „Kopf“ oder die Kamera nach oben und unten nicken kann!

Als Nächstes werden wir den Servo kennenlernen, den winzigen, aber leistungsstarken „Muskel“, der unseren Kippmechanismus bewegt. Wir werden verstehen, wie er funktioniert und wie wir ihn mit Arduino-Programmierung steuern können.

Genau wie unsere Halsmuskeln unseren Kopf bewegen, damit unsere Augen eine bessere Sicht haben, wird der Servo den Kippmechanismus bewegen, damit die Kamera des Rovers die Marslandschaft besser überblicken kann.

Also, schnallt euch an, Entdecker, lasst uns unsere Mission beginnen, unseren Rover mit seinem eigenen visuellen System auszustatten!

### 3.10.1 Kursziele

- Üben Sie die Installation und Bedienung des Kippmechanismus am Mars-Rover-Modell.
- Verstehen Sie die Funktionsprinzipien und Anwendungsmöglichkeiten eines Servos.
- Erlernen Sie, wie man die Bewegung des Servos durch Arduino-Programmierung steuert.

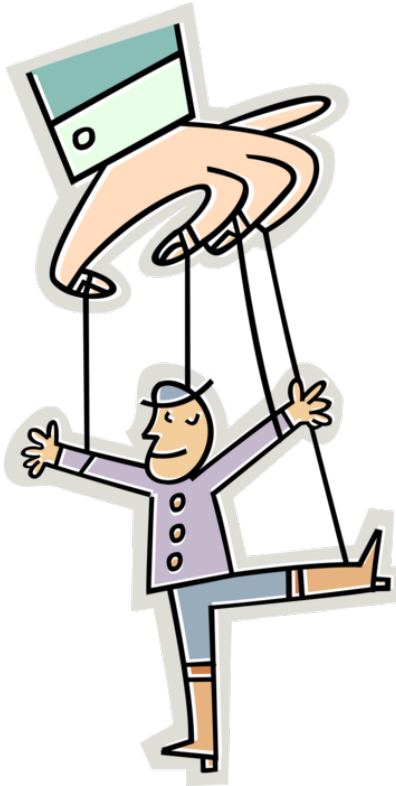
### 3.10.2 Kursmaterialien

- Arduino UNO Entwicklungsboard
- Servo
- Gimbal und Kamera
- Mars-Rover-Modell (bereits ausgestattet mit TT-Motor, Federungssystem, Ultraschall- und Infrarot-Hindernisvermeidungsmodulen, RGB-LED-Streifen)
- Arduino IDE
- Computer

### 3.10.3 Kursschritte

#### Schritt 1: Was ist ein Servo?

Haben Sie schon einmal eine Puppenshow gesehen? Wenn ja, haben Sie sich vielleicht darüber gewundert, wie der Puppenspieler die Arme, Beine und den Kopf der Puppe so geschmeidig bewegen kann, nur indem er an einigen Fäden zieht. In gewisser Weise sind Servomotoren wie unsere Puppenspieler.



Servomotoren sind eine besondere Art von Motoren, die sich nicht einfach nur drehen wie ein Rad. Stattdessen können sie sich auf eine bestimmte Position bewegen und diese Position halten. Stellen Sie sich vor, Sie spielen „Simon sagt“, und Simon sagt: „Heben Sie Ihren Arm in einem 90-Grad-Winkel!“ Das können Sie, oder? Das liegt daran, dass Sie, genau wie ein Servo, genau steuern können, wie weit Sie Ihren Arm bewegen.



- Braune Leitung: GND
- Orange Leitung: Signalleitung, verbinden mit dem PWM-Pin des Hauptboards.
- Rotes Kabel: VCC

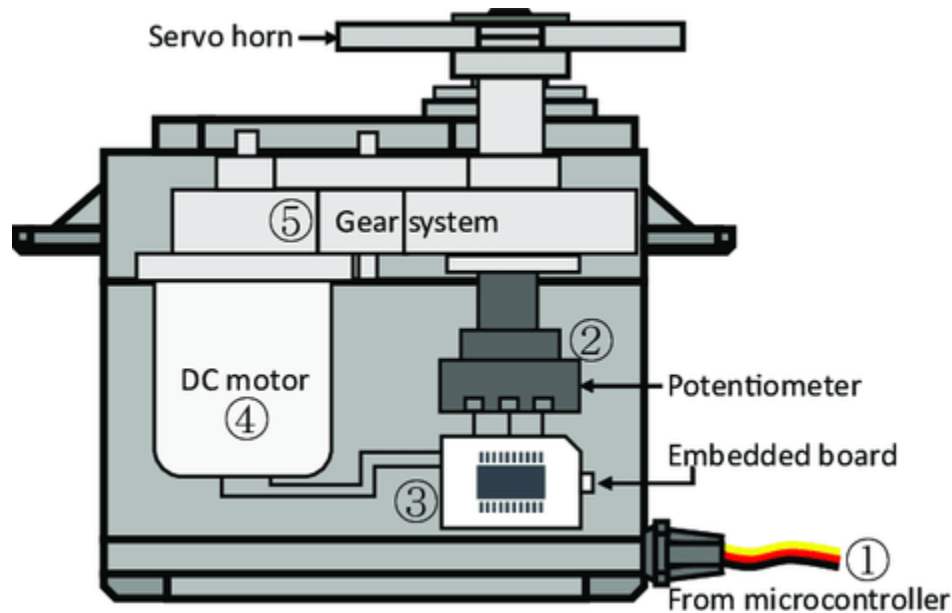
Genau wie Sie Ihre Arme kontrollieren können, um sich in bestimmte Positionen zu bewegen, können wir Servomotoren verwenden, um die genaue Position von Objekten in unseren Projekten zu steuern. In unserem Mars-Rover werden wir einen Servo verwenden, um die Auf- und Abbewegung unseres Kippmechanismus zu steuern, genau wie Sie mit Ihrem Kopf nicken können.

Im nächsten Schritt werden wir eine faszinierende Reise ins Innere eines Servomotors unternehmen, um zu verstehen, wie er funktioniert. Gespannt? Los geht's!

### **Schritt 2: Wie funktioniert ein Servo?**

Wie genau funktioniert nun ein Servo? Lassen Sie uns eine spannende Reise ins Innere eines Servos unternehmen!

Wenn wir in einen Servo hineinschauen würden, würden wir einige Teile sehen. Im Herzen eines Servos befindet sich ein normaler Motor, ganz ähnlich den Motoren, die die Räder unseres Mars-Rovers drehen. Um den Motor herum befindet sich ein großes Zahnrad, das mit einem kleineren Zahnrad auf der Motorwelle verbunden ist. So wird die schnelle, kreisförmige Bewegung des Motors in eine langsamere, aber stärkere Bewegung umgewandelt.



Aber das ist nicht das, was einen Servo besonders macht. Die Magie passiert in einem winzigen Stück Elektronik, genannt „potentiometer“, und der „control circuitry“. So funktioniert es: Wenn sich der Servo bewegt, dreht sich das Potentiometer und ändert seinen Widerstand. Die Steuerschaltung misst diese Widerstandsänderung und weiß genau, in welcher Position sich der Servo befindet. Clever, oder?

Um einen Servo zu steuern, senden wir ihm ein spezielles Signal, genannt „Pulsbreitenmodulations“-Signal oder PWM. Indem wir die Breite dieser Pulse ändern, können wir genau steuern, wie weit sich der Servo bewegt und ihn in dieser Position halten.

Im nächsten Schritt lernen wir, wie man einen Servo mit einem Arduino steuert. Bereit für ein paar magische Sprüche in Form von Code? Los geht's!

### Schritt 3: Steuerung eines Servos mit Arduino

In Ordnung, Entdecker, jetzt, wo wir wissen, wie ein Servo funktioniert, lernen wir, wie man ihn mit unserem Zauberstab, dem Arduino, steuert!

Einen Servo zu steuern ist wie ihm Anweisungen zu geben. Erinnern Sie sich an die zuvor erwähnten Pulsbreitenmodulations- (PWM) Signale? Diese werden wir verwenden, um dem Servo zu sagen, wohin er sich bewegen soll.

Glücklicherweise macht Arduino uns diese Aufgabe leicht mit einer eingebauten Bibliothek namens `Servo`. Mit dieser Bibliothek können wir ein `Servo`-Objekt erstellen, einen Pin damit verbinden (den Pin, an den unser Servo angeschlossen ist) und dann einen einfachen Befehl, `write()`, verwenden, um den Winkel einzustellen.

Hier ist ein Ausschnitt, wie der Code aussieht:

```
#include <Servo.h>

Servo myServo; // create a servo object

void setup() {
  myServo.attach(6); // attaches the servo on pin 6
}

void loop() {
  myServo.write(90); // tell servo to go to 90 degrees
}
```

In diesem Code ist `myServo` unser Servo-Objekt, `attach(6)` sagt dem Arduino, dass unser Servo an Pin 6 angeschlossen ist, und `write(90)` sagt dem Servo, er soll sich auf 90 Grad bewegen.

Großartige Arbeit, Entdecker! Sie haben gerade gelernt, wie man einen Servomotor mit Arduino steuert. Sie können auch mit verschiedenen Winkeln experimentieren!

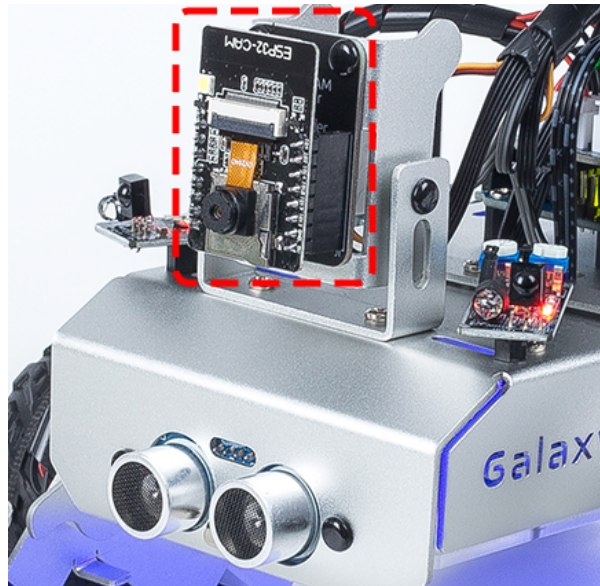
### Schritt 4: Montage des visuellen Systems

Jetzt sind Sie bereit, das visuelle System unseres Rovers zusammenzubauen.

---

#### Bemerkung:

- Achten Sie beim Einsetzen der ESP32 CAM in den Kameraadapter auf die Ausrichtung. Sie sollte richtig mit dem ESP32-Adapter ausgerichtet sein.



---

### Schritt 5: Die Grenzen des Kippmechanismus verstehen

Obwohl der Servo dafür ausgelegt ist, sich zwischen 0 und 180 Grad zu drehen, haben Sie vielleicht bemerkt, dass er jenseits eines bestimmten Punktes (sagen wir nach 150 Grad) nicht mehr reagiert. Haben Sie sich jemals gefragt, warum das passiert? Lassen Sie uns dieses Geheimnis in unserem nächsten Abenteuer zusammen erkunden!

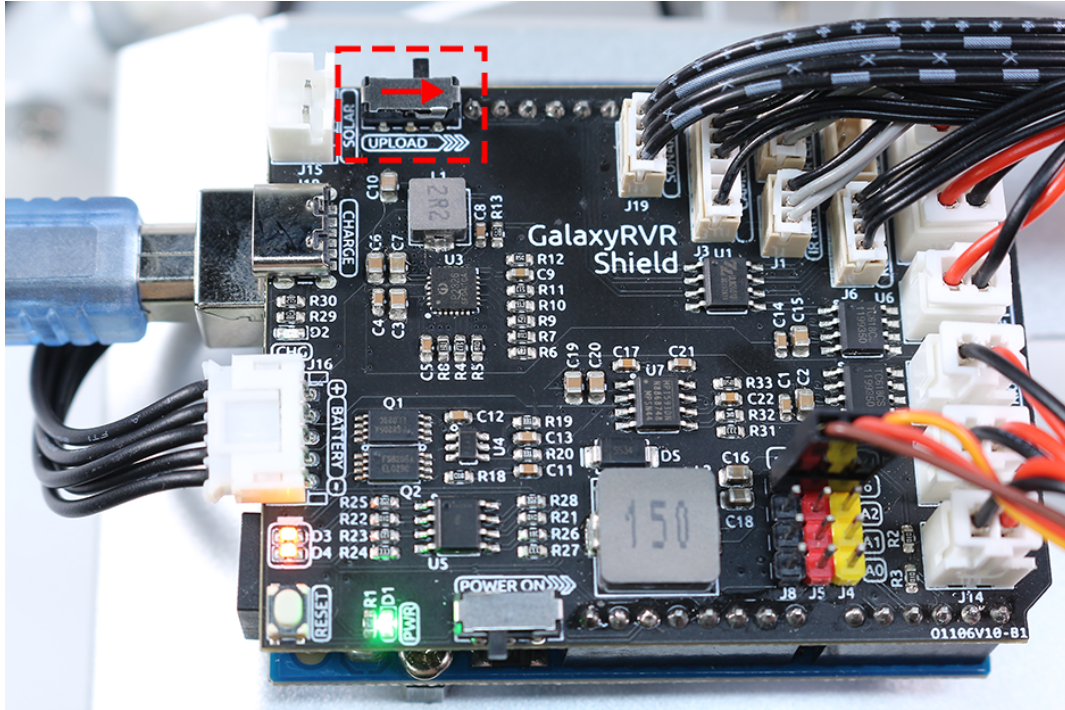
Können Sie sich einen Vogel vorstellen, der versucht, seinen Hals zu weit zu beugen, sodass er seinen eigenen Körper berührt und sich nicht weiterbewegen kann? Unser Rover-Kippmechanismus steht vor einer ähnlichen Situation. Wenn sich der Servo nach unten bewegt, kann er gegen den Körper unseres Rovers stoßen und kann nicht über einen bestimmten Winkel hinausgehen.

Wenn wir versuchen, ihn durch Schreiben eines unerreichbaren Winkels in unserem Code zu zwingen, sich über diesen Punkt hinaus zu bewegen, kann unser kleiner Servo-Vogel stecken bleiben und sich sogar beschädigen! Das wollen wir nicht, oder? Daher wollen wir seine Bewegungsgrenzen mit einem kleinen Experiment verstehen.

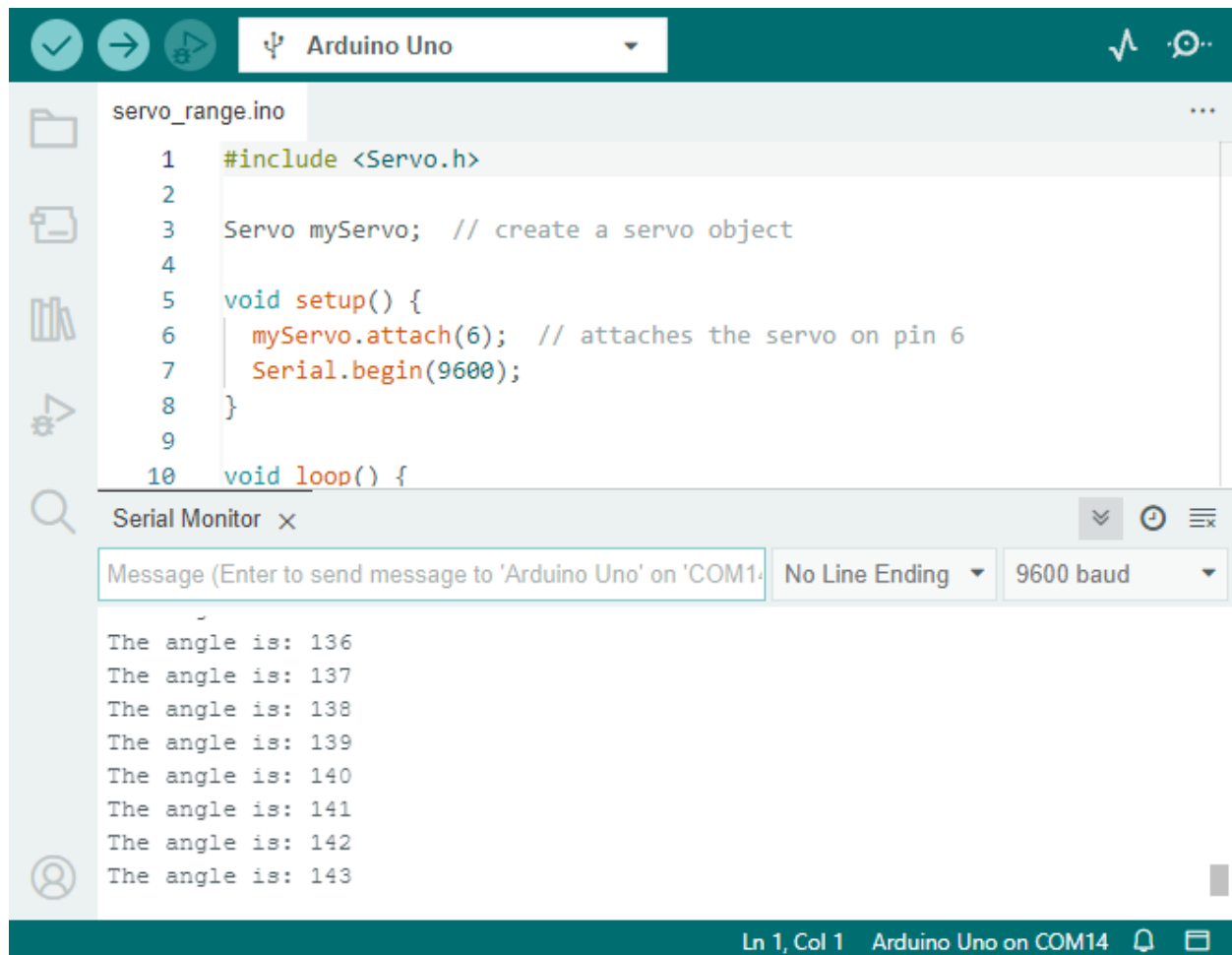
Wir verwenden eine `for`-Schleife, um den Servo von 0 bis 180 Grad zu drehen, während wir den Winkel im Seriellen Monitor notieren.

- Der ESP32-CAM und das Arduino-Board teilen sich dieselben RX (Empfang) und TX (Senden) Pins. Daher müssen Sie vor dem Hochladen des Codes zuerst den ESP32-CAM durch Verschieben dieses Schalters nach rechts freigeben, um Konflikte oder potenzielle Probleme zu vermeiden.





- Nachdem Sie diesen Code hochgeladen haben, öffnen Sie den **Serial Monitor**. Wenn keine Informationen erscheinen, drücken Sie die **Reset button** auf dem GalaxyRVR-Shield, um den Code erneut auszuführen.
- Sie werden sehen, wie sich der Servo dreht, und der Serielle Monitor zeigt den Winkel an.



Bei meinem Rover konnte der Kippmechanismus bis zu etwa  $140^\circ$  hochgehen, bevor er den Körper des Rovers berührte und nicht weiter gehen konnte.

Also, Entdecker, denken Sie immer daran, die Grenzen Ihres Rovers zu respektieren, um ihn sicher und funktionsfähig zu halten!

### Schritt 6: Teilen und Reflektieren

Gut gemacht, Entdecker! Heute haben Sie nicht nur einen Kippmechanismus für Ihren Rover gebaut, sondern auch verstanden, wie man einen Servo steuert, um ihn zu bewegen. Das ist ein großer Schritt vorwärts in unserer Mars-Rover-Mission.

Jetzt wollen wir unsere Erfahrungen teilen und über das, was wir gelernt haben, nachdenken.

Sind Ihnen beim Aufbau des Kippmechanismus oder bei der Programmierung des Servos Herausforderungen begegnet? Wie haben Sie diese überwunden?

Denken Sie daran, jede Herausforderung, die wir überwinden, macht uns klüger und unseren Rover besser. Zögern Sie also nicht, Ihre Geschichten, Ideen und Lösungen zu teilen. Wer weiß, vielleicht hilft Ihre innovative Lösung einem anderen Entdecker auf seiner Reise!

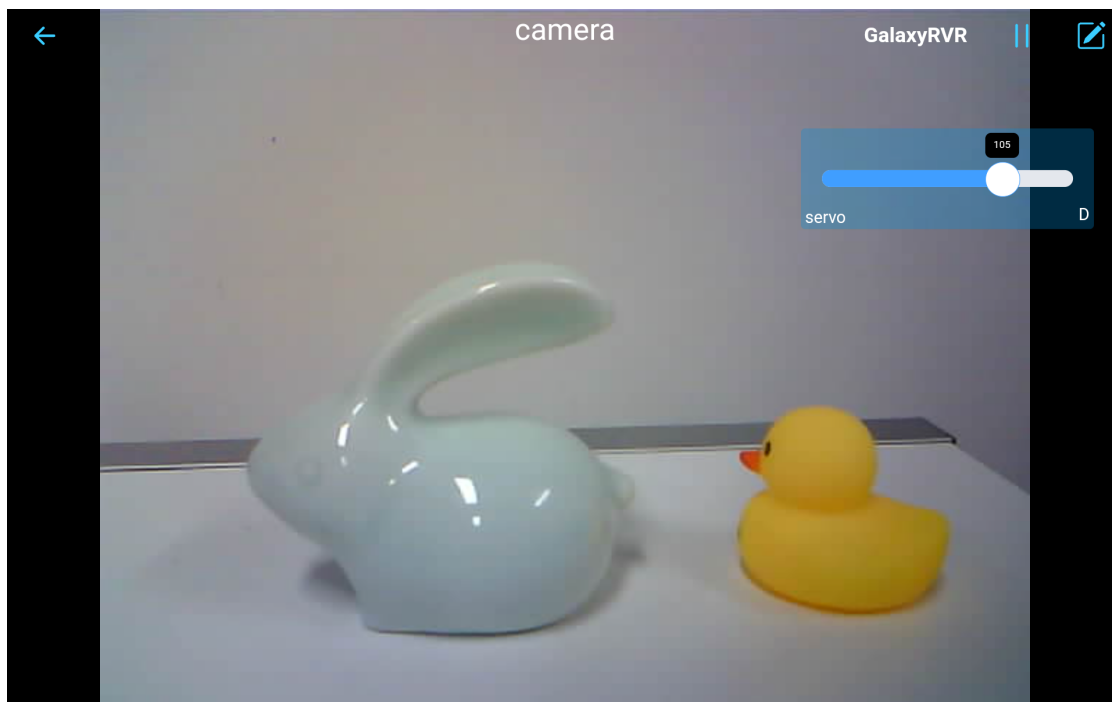
## 3.11 Lektion 11: Erforschung des visuellen Systems des Mars-Rovers - Kamera und Echtzeitsteuerung

Willkommen zurück, junge Entdecker! In der letzten Lektion haben wir unseren Mars-Rover mit der Fähigkeit ausgestattet, mit einem Kippmechanismus zu „nicken“. Jetzt ist es an der Zeit, unserem Rover „Augen“ zu geben - die Kamera!

In dieser aufregenden Reise tauchen wir in die Einrichtung des Kamera-Systems des Rovers ein. Ihr werdet lernen, wie man die vom Rover eingefangenen Bilder auf eine Webseite überträgt, sodass ihr genau sehen könnt, was der Rover sieht, und das in Echtzeit. Stellt euch die Begeisterung vor, die Marslandschaft aus der Perspektive des Rovers zu erleben!

Die Begeisterung setzt sich fort, da wir auch die SunFounder Controller-App vorstellen. Diese Anwendung ermöglicht es uns, ein Live-Feed der Sicht des Rovers zu erhalten, während er sich bewegt, und wir können den Kippmechanismus direkt von unseren Smartphones oder Tablets aus steuern. Es ist, als hätte man eine Fernbedienung mit eingebautem Bildschirm!

Dies bietet ein noch interaktiveres und spannenderes Erlebnis mit unserem Rover. Bleibt dran für mehr Abenteuer!



### 3.11.1 Kursziele

- Verstehen, wie man eine WLAN-Verbindung mit dem ESP32 CAM herstellt.
- Lernen, genau zu sehen, was der Rover sieht, und das in Echtzeit.
- Lernen, wie man die SunFounder Controller-App verwendet, um eine virtuelle Fernbedienung zu erstellen und den Mars-Rover zu steuern.

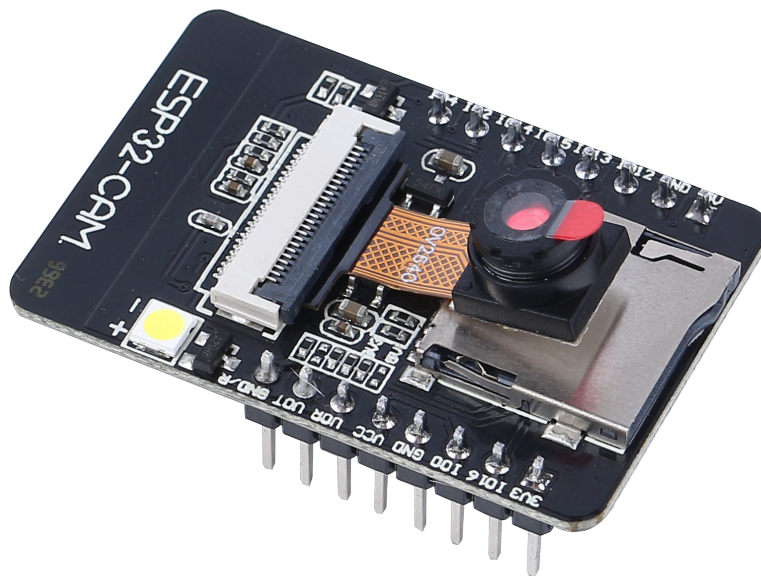
### 3.11.2 Kursmaterialien

- Mars-Rover-Modell (mit allen Komponenten ausgestattet)
- Arduino IDE
- Computer
- Tablet oder Smartphone mit installierter SunFounder Controller-App

### 3.11.3 Kursschritte

#### Schritt 1: Einführung in ESP32 CAM

In unserem vorherigen Abenteuer haben wir unseren Mars-Rover mit einem Paar „Augen“ ausgestattet, indem wir den ESP32 CAM integriert haben. Heute werden wir mehr darüber lernen und ihn tatsächlich „sehen“ lassen.



Der ESP32 CAM, der als die Augen unseres Rovers fungiert, ist ein kleines, aber leistungsstarkes Modul. Es integriert nicht nur WLAN- und Bluetooth-Funktionalitäten, sondern verfügt auch über eine kompakte Kamera. Diese Kamera hilft unserem Rover, Bilder seiner Umgebung aufzunehmen.

Genau wie wir unsere Augen verwenden, um unsere Umgebung zu beobachten, kann der ESP32 CAM „sehen“, was vor dem Rover liegt, und diese visuellen Daten an unser Smartphone oder Computer senden. Dies ermöglicht es uns, alles zu sehen, was der Rover in Echtzeit sieht!

Es ist, als würden wir den Rover direkt steuern, nicht nur den Rover selbst beobachten, sondern auch die Welt, die er erkundet! Unglaublich, oder? Also, lasst uns tiefer eintauchen...

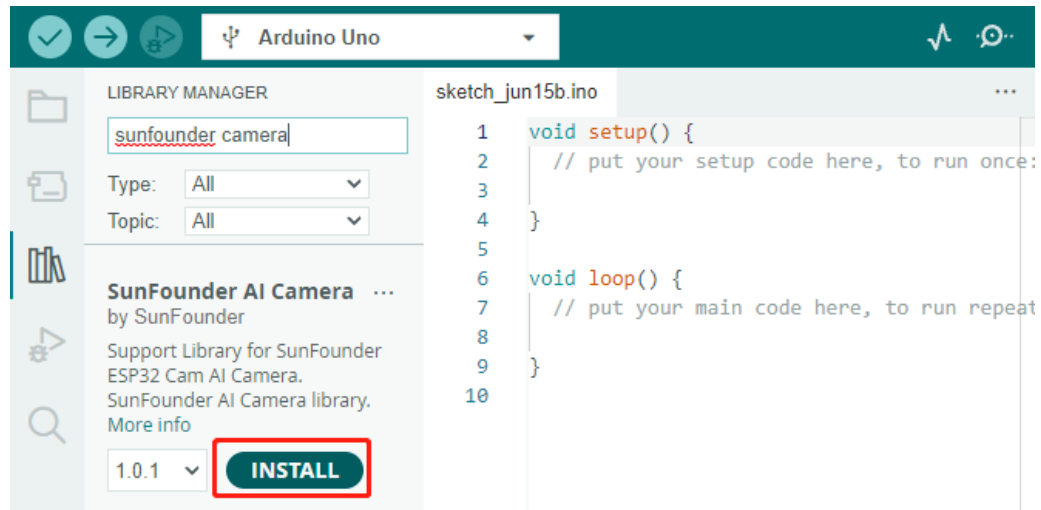
#### Schritt 2: Programmierung der Kamera des Rovers und Betrachtung des Feeds

Nachdem wir die ESP32-CAM an unseren Rover angebracht haben, müssen wir sie nun zum Leben erwecken. Dazu verwenden wir die Arduino IDE, um ein Programm zu schreiben, das die Kamera steuert, es ihr ermöglicht, sich mit WLAN zu verbinden, und die eingefangenen Bilder zu streamen.

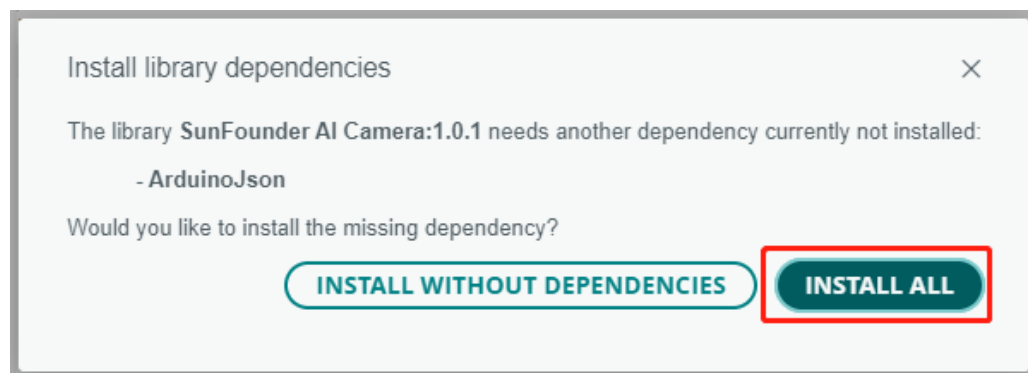
So können wir das tun:

1. Installieren Sie die SunFounder AI Camera-Bibliothek.

- Öffnen Sie den **Library Manager** der Arduino IDE, suchen Sie nach „SunFounder Camera“ und klicken Sie auf **INSTALL**.



- Ein Popup-Fenster erscheint für die Installation der Bibliotheksabhängigkeiten. Klicken Sie auf **INSTALL ALL** und warten Sie, bis der Vorgang abgeschlossen ist.



- Geben Sie in der Arduino IDE den folgenden Code ein.

Bezüglich der Variablen **NAME**, **TYPE** und **PORT** im Code werden wir uns an dieser Stelle nicht vertiefen. Sie kommen im nächsten Schritt zum Einsatz. Behalten Sie lediglich im Hinterkopf, dass diese Variablen in unserer bevorstehenden Reise wichtig sein werden, um einen Echtzeit-Video-Feed von unserem Mars-Rover zu etablieren.

Beachten Sie, dass wir im Code zwei Verbindungsmodi haben - **AP-Modus** und **STA-Modus**. Sie können entscheiden, welchen Sie je nach Ihren spezifischen Bedürfnissen verwenden möchten.

- AP Mode:** In diesem Modus erstellt der Rover einen Hotspot (in unserem Code als GalaxyRVR benannt). Dies ermöglicht es jedem Gerät wie einem Mobiltelefon, Tablet oder Laptop, sich mit diesem Netzwerk zu verbinden. Dies ist besonders nützlich, wenn Sie den Rover unter allen Umständen fernsteuern möchten. Beachten Sie jedoch, dass dies Ihr Gerät vorübergehend daran hindert, eine Internetverbindung herzustellen.

```
// AP Mode #define WIFI_MODE WIFI_MODE_AP #define SSID „GalaxyRVR“ #define
PASSWORD „12345678“
```

- STA Mode:** In diesem Modus verbindet sich der Rover mit Ihrem Heim-WLAN-Netzwerk. Den-

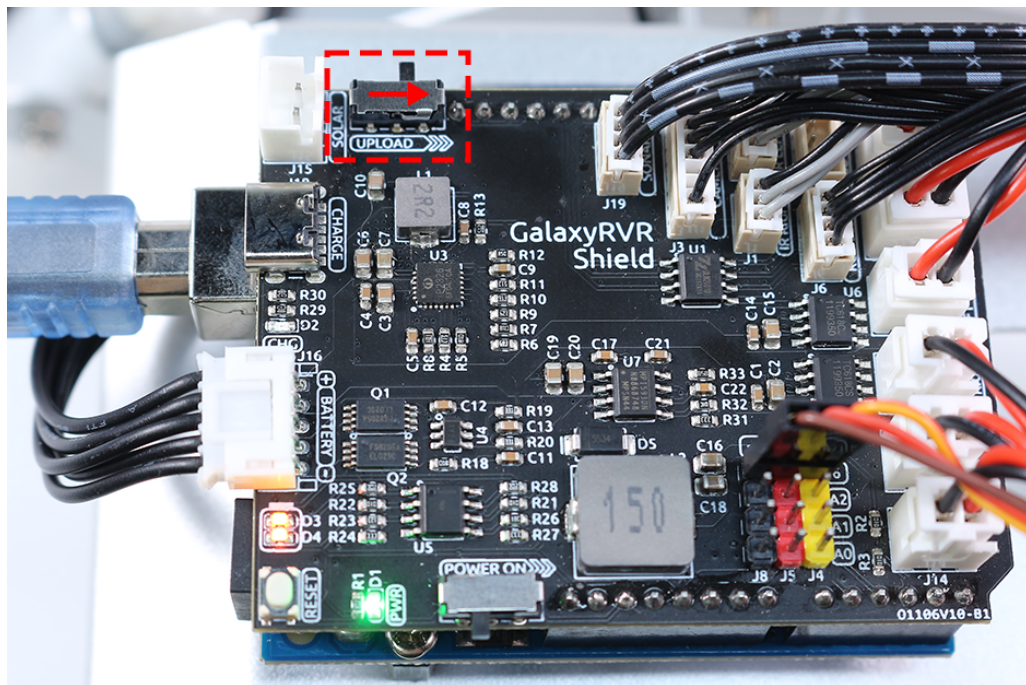


ken Sie daran, dass Ihr Steuergerät (wie ein Mobiltelefon oder Tablet) ebenfalls mit demselben WLAN-Netzwerk verbunden sein sollte. Dieser Modus ermöglicht es Ihrem Gerät, seine reguläre Internetverbindung beizubehalten, während es den Rover steuert, beschränkt aber den Einsatzbereich des Rovers auf Ihre WLAN-Abdeckungsfläche.

```
// STA Mode
#define WIFI_MODE WIFI_MODE_STA
#define SSID "YOUR SSID"
#define PASSWORD "YOUR PASSWORD"
```

3. Laden Sie den Code auf unseren Rover und erwecken Sie unsere ESP32-CAM zum Leben!

- Der ESP32-CAM und das Arduino-Board teilen sich die gleichen RX (Empfang) und TX (Senden) Pins. Daher müssen Sie vor dem Hochladen des Codes zuerst den ESP32-CAM durch Verschieben dieses Schalters nach rechts freigeben, um Konflikte oder mögliche Probleme zu vermeiden.

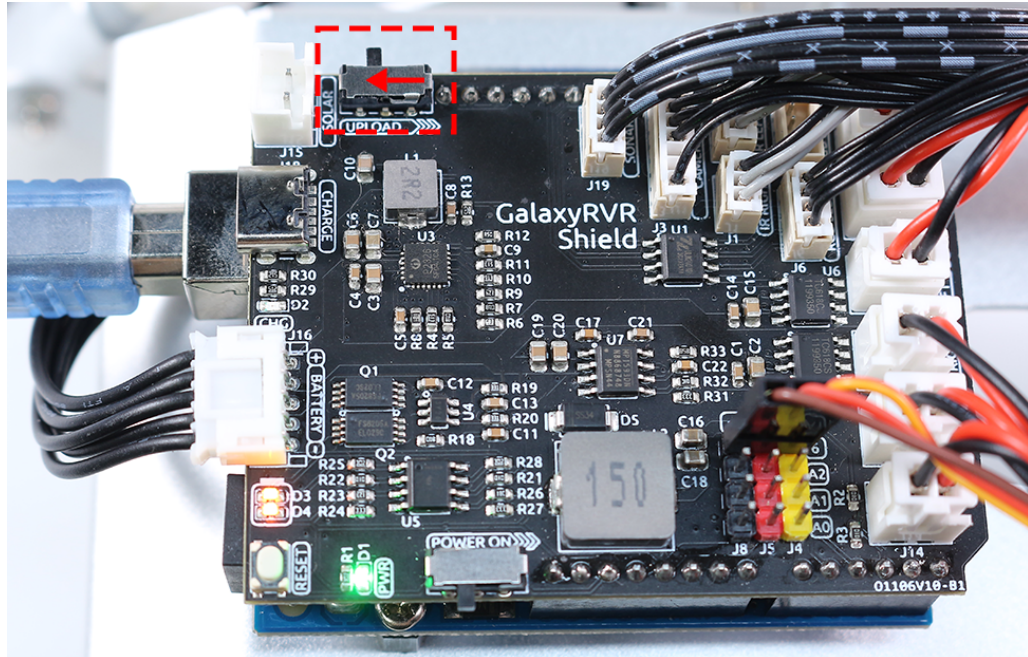


- Sobald der Code erfolgreich hochgeladen wurde, schalten Sie ihn zurück auf die linke Seite, um die ESP32-CAM zu starten.

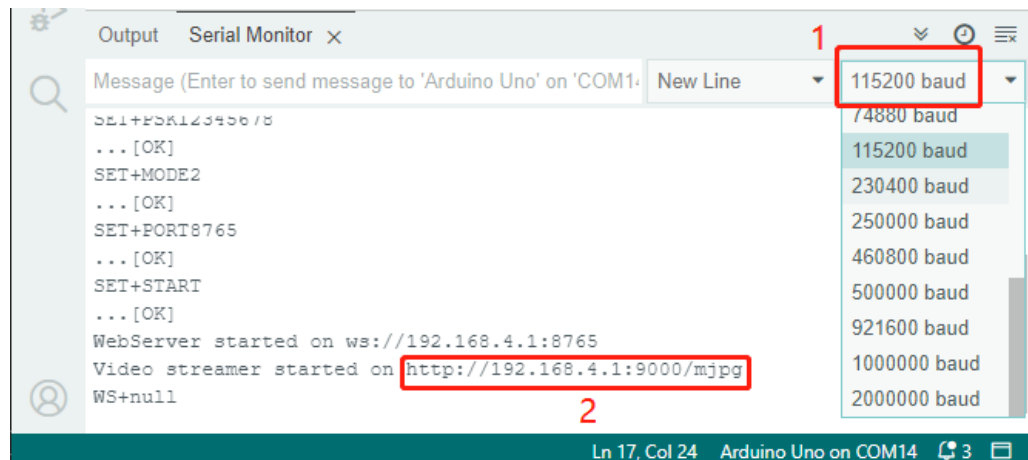
---

**Bemerkung:** Dieser Schritt und der vorherige sind jedes Mal erforderlich, wenn Sie den Code neu hochladen.

---



- Öffnen Sie den **Serial Monitor** und stellen Sie die Baudrate auf 115200 ein. Wenn keine Informationen erscheinen, drücken Sie die **Reset button** auf dem GalaxyRVR-Shield, um den Code erneut auszuführen. Sie sollten eine IP-Adresse im Ausgabebereich des seriellen Monitors sehen. Dies ist die Adresse, an die die Kamera Ihres Rovers sendet.



- Jetzt ist es an der Zeit, tatsächlich zu sehen, was unser Rover sieht! Öffnen Sie einen Webbrowser - wir empfehlen Google Chrome - und geben Sie die URL ein, die Sie im Seriellen Monitor sehen, im Format `http://ip:9000/mjpg`.





Und voilà! Jetzt sollten Sie den Live-Feed von der Kamera Ihres Rovers sehen können. Ist es nicht erstaunlich zu denken, dass Sie den Mars (oder vielleicht nur Ihr Wohnzimmer) aus der Perspektive des Rovers betrachten? Ganz wie ein echter Mars-Rover-Wissenschaftler!

Denken Sie daran, dies ist erst der Anfang. Es gibt noch so viel mehr zu entdecken und zu lernen. Im nächsten Schritt werden wir erforschen, wie wir unseren Rover steuern können, während wir den Live-Kamera-Feed betrachten. Aufregend, nicht wahr? Weiter so, Entdecker!

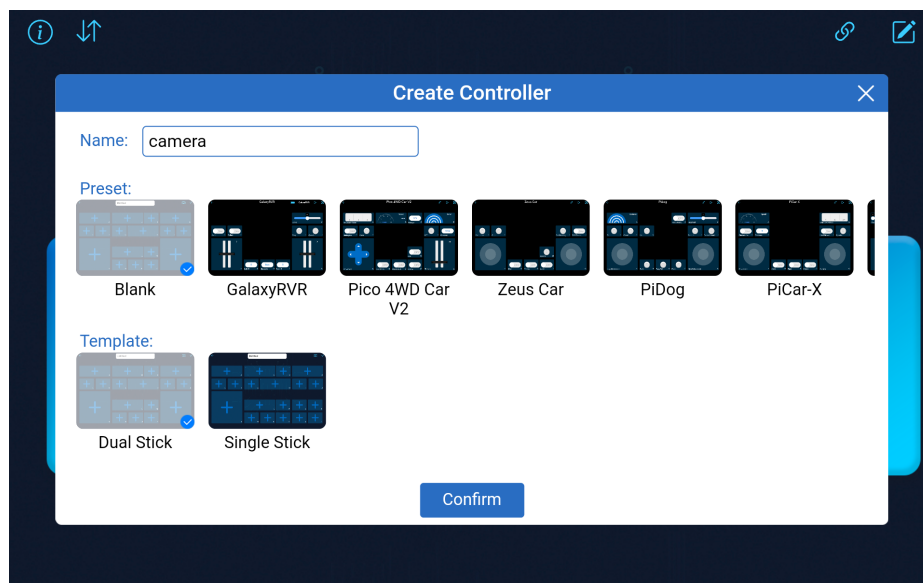
### Schritt 3: Steuern und Anzeigen des Kamera-Feeds mit der App

Haben Sie sich schon einmal gewünscht, den visuellen Feed des Mars-Rovers direkt auf Ihrem Smartphone ansehen zu können, während Sie gleichzeitig seinen Kippmechanismus steuern? Jetzt können Sie das! Mit Hilfe der SunFounder Controller-App können Sie genau das tun. Folgen Sie den unten stehenden Schritten:

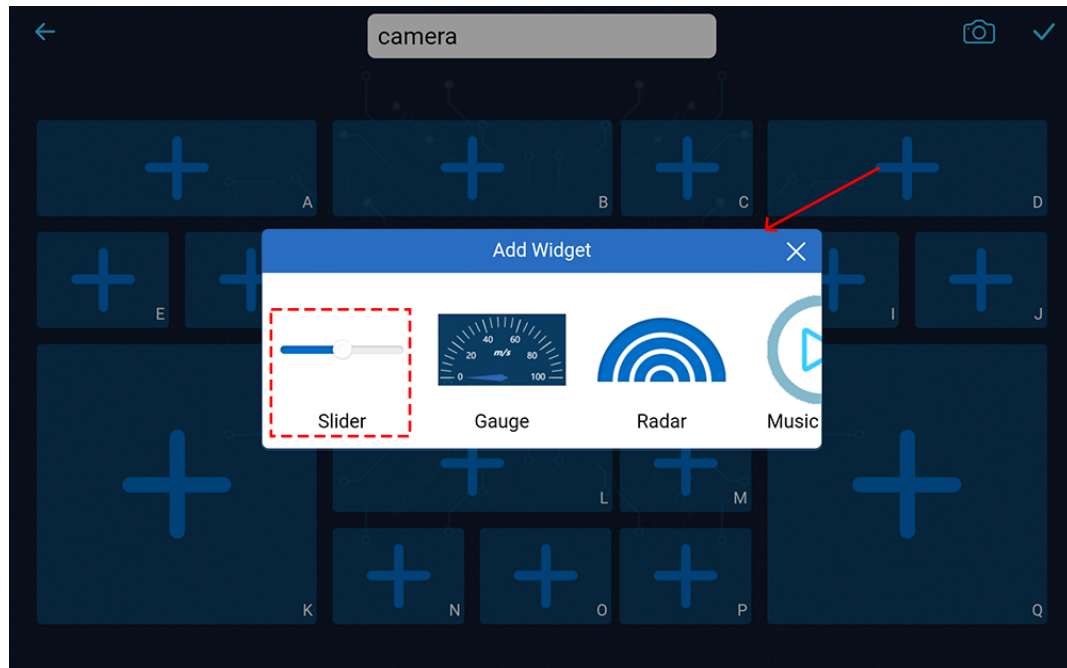
1. Installieren Sie aus dem **APP Store(iOS)** oder **Google Play(Android)**.
2. Erstellen Sie einen Controller.
  - Um einen Controller in der SunFounder Controller-App hinzuzufügen, klicken Sie auf das +-Symbol.



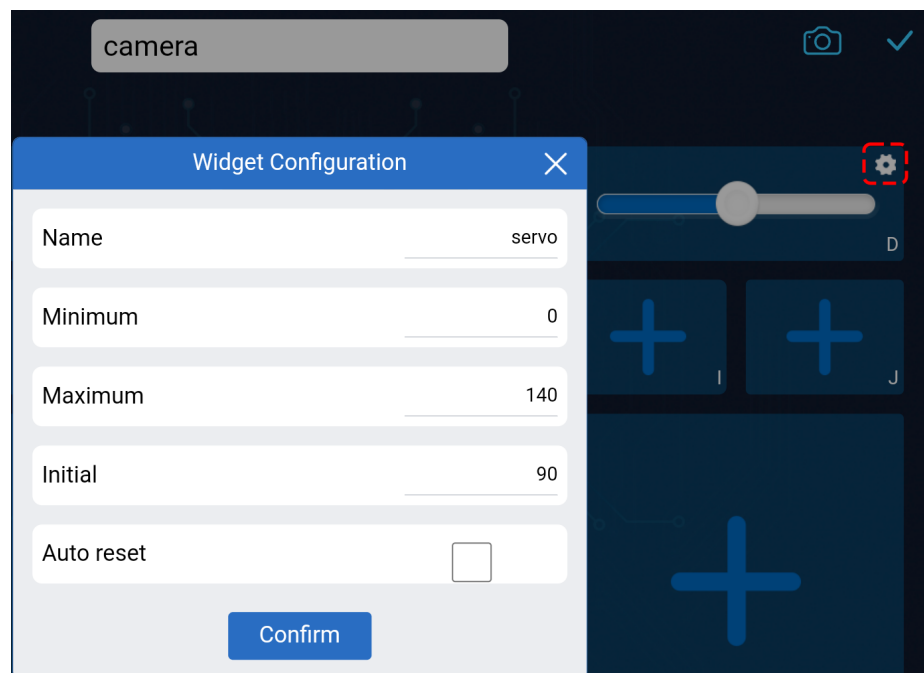
- Wählen Sie das **Blank**-Voreinstellung, wählen Sie entweder **Dual** oder **Single Stick** je nach Ihrer Präferenz. Geben Sie Ihrem neuen Controller einen Namen und klicken Sie auf **Confirm**.




- Sie befinden sich jetzt im Controller. Klicken Sie im D-Bereich auf das +-Symbol und wählen Sie Slider aus dem Popup-Menü.



- Dieses Slider-Widget ist dazu gedacht, den Kippmechanismus zu steuern. Wie wir in der vorherigen Lektion gelernt haben, liegt sein Bereich zwischen 0 und 140. Daher werden wir diese Werte als Mindest- und Höchstwerte für unser Slider-Widget festlegen.



- Klicken Sie auf den -Button in der oberen rechten Ecke, um diesen Controller zu speichern.

3. Schreiben wir einen Code, um den Wert des Sliders zu erfassen:

- Basierend auf dem vorherigen Code wechseln wir in den AP-Modus, in dem Sie das SSID und PASSWORD nach Ihren Wünschen festlegen können.

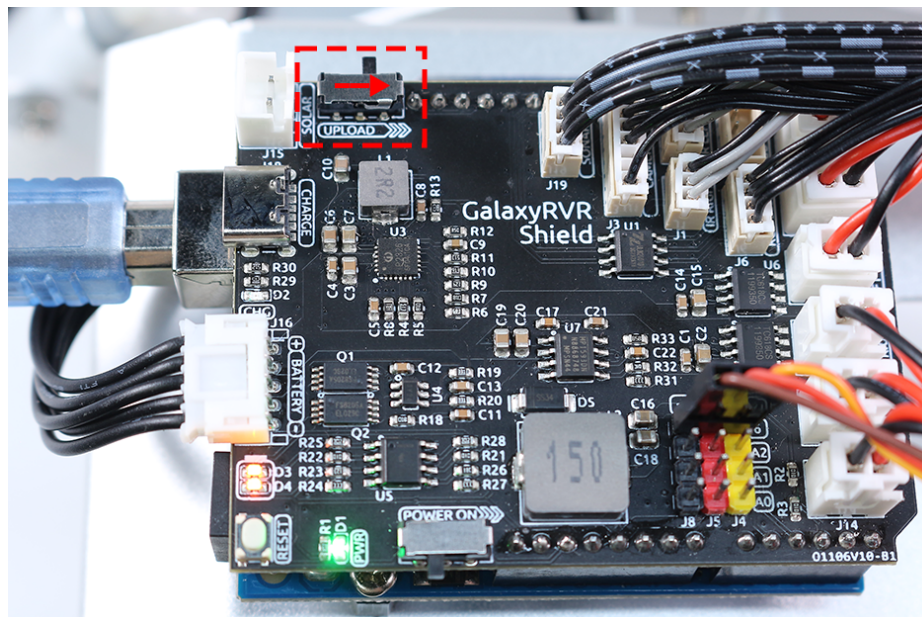
```
// AP Mode
#define WIFI_MODE WIFI_MODE_AP
#define SSID "GalaxyRVR"
#define PASSWORD "12345678"
```

- Als Nächstes fügen wir eine `onReceive()`-Funktion hinzu, um Werte vom SunFounder Controller zu empfangen und diese Werte im Seriellen Monitor anzuzeigen. Wir verwenden die Funktion `getSlider()`, um den Wert des **slider**-Widgets zu erhalten. Ich habe ein **slider**-Widget in Region D hinzugefügt, aber wenn Sie es in einer anderen Region hinzugefügt haben, müssen Sie `REGION_D` in Ihre Region ändern.

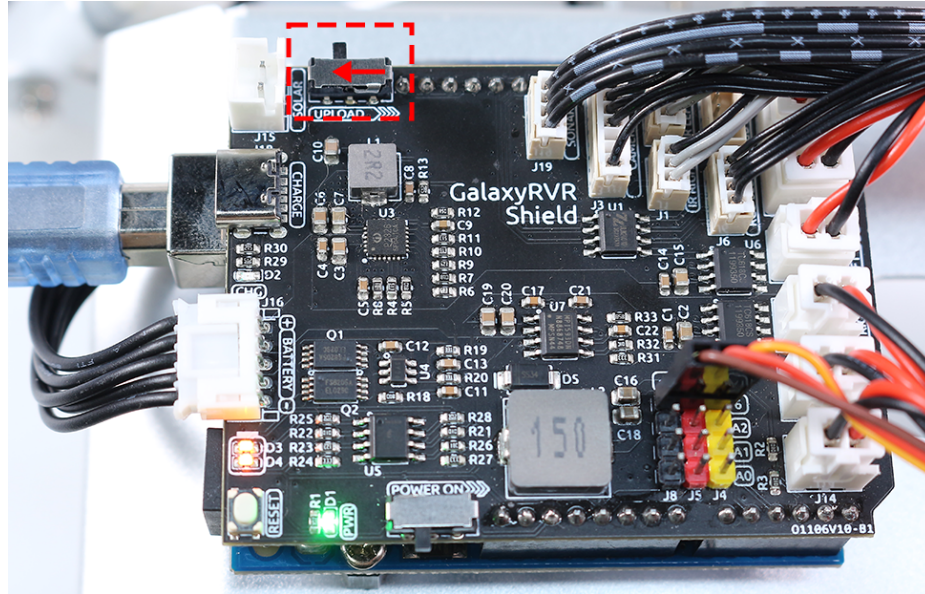
```
void onReceive() {
    int16_t sliderD = aiCam.getSlider(REGION_D);
    Serial.print("Slider D: ");
    Serial.println(sliderD);
}

void setup() {
    ...
    // Set the function to execute when data is received
    aiCam.setOnReceived(onReceive);
    ...
}
```

- Hier ist der komplette Code:
- Stellen Sie vor dem Hochladen des Codes sicher, dass der Schalter nach rechts gedreht ist.



- Nachdem der Code erfolgreich hochgeladen wurde, bewegen Sie den Schalter nach links, um die ESP32-CAM zu starten.



- Wenn Sie die folgenden Informationen im Seriellen Monitor sehen, können Sie mit dem nächsten Schritt fortfahren.

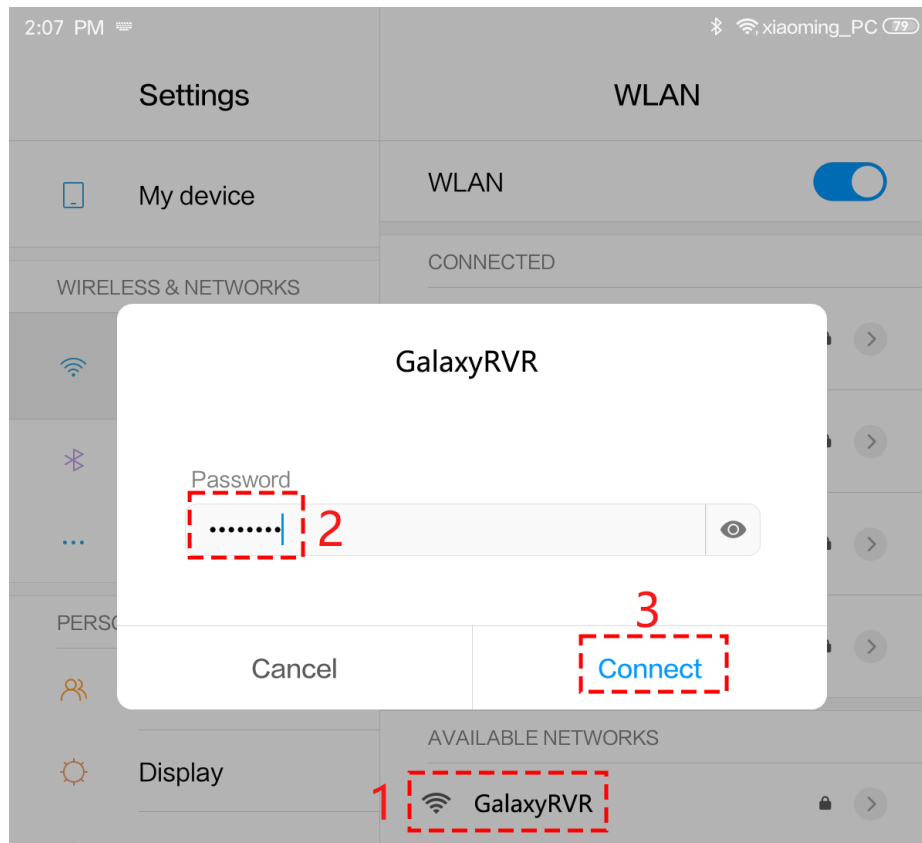
```
... [OK]  
SET+PORT8765  
... [OK]  
SET+START  
... [OK]  
WebServer started on ws://192.168.4.1:8765  
Video streamer started on http://192.168.4.1:9000/mjpg  
WS+null
```

#### 4. Verbinden Sie sich mit dem Netzwerk GalaxyRVR.

An diesem Punkt sollten Sie Ihr mobiles Gerät mit dem lokalen Netzwerk (LAN) verbinden, das von GalaxyRVR bereitgestellt wird. Dadurch befinden sich sowohl Ihr mobiles Gerät als auch der Rover im selben Netzwerk, was eine reibungslose Kommunikation zwischen den Anwendungen auf Ihrem mobilen Gerät und dem Rover ermöglicht.

- Finden Sie GalaxyRVR in der Liste der verfügbaren Netzwerke auf Ihrem mobilen Gerät (Tablet oder Smartphone), geben Sie das Passwort 12345678 ein und verbinden Sie sich damit.





- Der Standardverbindungsmodus ist **AP mode**. Nachdem Sie die Verbindung hergestellt haben, kann eine Warnmeldung erscheinen, dass es keinen Internetzugang in diesem WLAN-Netzwerk gibt, bitte wählen Sie, die Verbindung fortzusetzen.


### Confirm

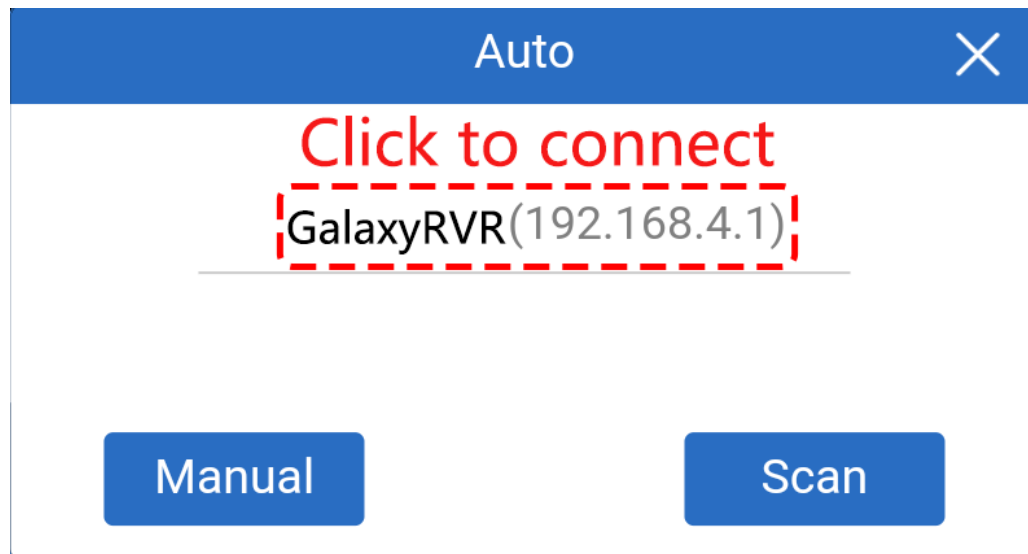
This WLAN network has no access to the internet. Other WLAN networks are available. Switch networks?

Stay connected


Switch networks

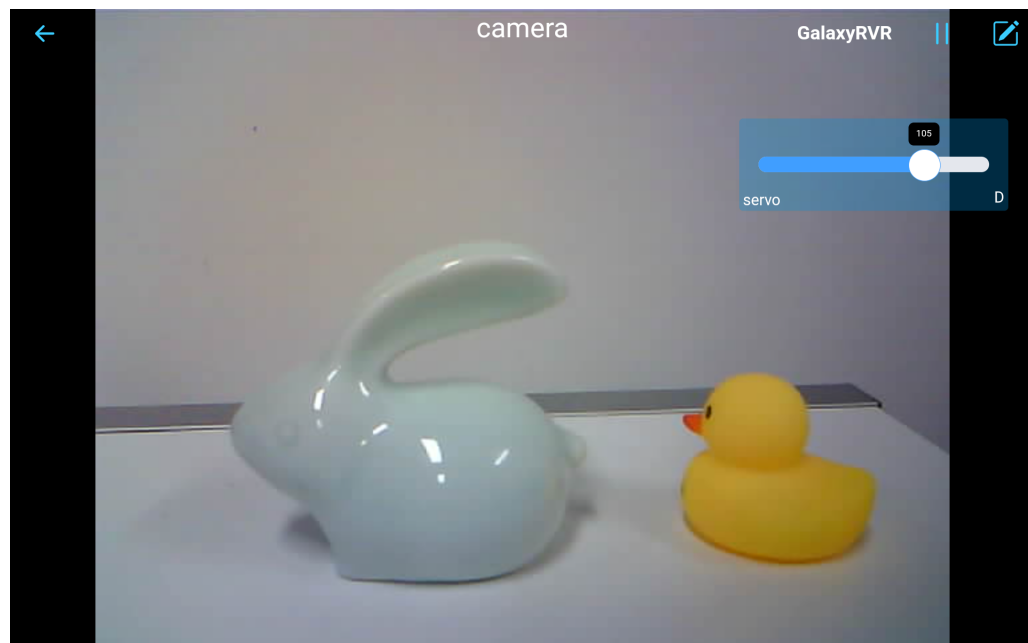
#### 5. Verbinden und Aktivieren des Controllers.

- Kehren Sie nun zu dem Controller zurück, den Sie zuvor erstellt haben (in meinem Fall heißt er „camera“). Verwenden Sie den -Button, um den SunFounder Controller mit dem Rover zu verbinden und eine Kommunikationsverbindung herzustellen. Nach einer kurzen Wartezeit erscheint „GalaxyRVR(IP)“ (der Name, den Sie im Code mit `#define NAME "GalaxyRVR"` zugewiesen haben). Klicken Sie darauf, um eine Verbindung herzustellen.



**Bemerkung:** Bitte überprüfen Sie, ob Ihr WLAN mit „GalaxyRVR“ verbunden ist, falls Sie die obige Nachricht nach einiger Zeit nicht sehen.

- Sobald Sie die Nachricht „Connected Successfully“ sehen, drücken Sie den -Button. Dies wird das Live-Footage der Kamera in der App anzeigen.



- Bewegen Sie jetzt den Schieberegler und öffnen Sie gleichzeitig den seriellen Monitor der Arduino IDE. Sie sollten ähnliche Daten wie unten sehen.

```
Slider D: 105  
WS+null  
Slider D: 105  
WS+null
```

(Fortsetzung auf der nächsten Seite)



(Fortsetzung der vorherigen Seite)

```
Slider D: 105
WS=null
```

6. Lassen Sie den Schieberegler den Kippmechanismus steuern.

**Jetzt, wo wir die vom Slider-Widget übertragenen Werte kennen, können wir diese Werte direkt verwenden, um den Servo zu drehen.**

Fügen Sie daher basierend auf dem vorherigen Code die folgenden Zeilen hinzu, um den Servo zu initialisieren und den Wert des Schiebereglers auf den Servo zu schreiben.

```
...
#include <Servo.h>

Servo myServo; // create a servo object
myServo.write(int(slidebarD)); // control the servo to move to the current
↪angle

...

void onReceive() {
    ...
    myServo.write(int(slidebarD)); // control the servo to move to the
↪current angle
}

void setup() {
    ...
    myServo.attach(6); // attaches the servo on pin 6
    ...
}
```

Hier ist der vollständige Code:

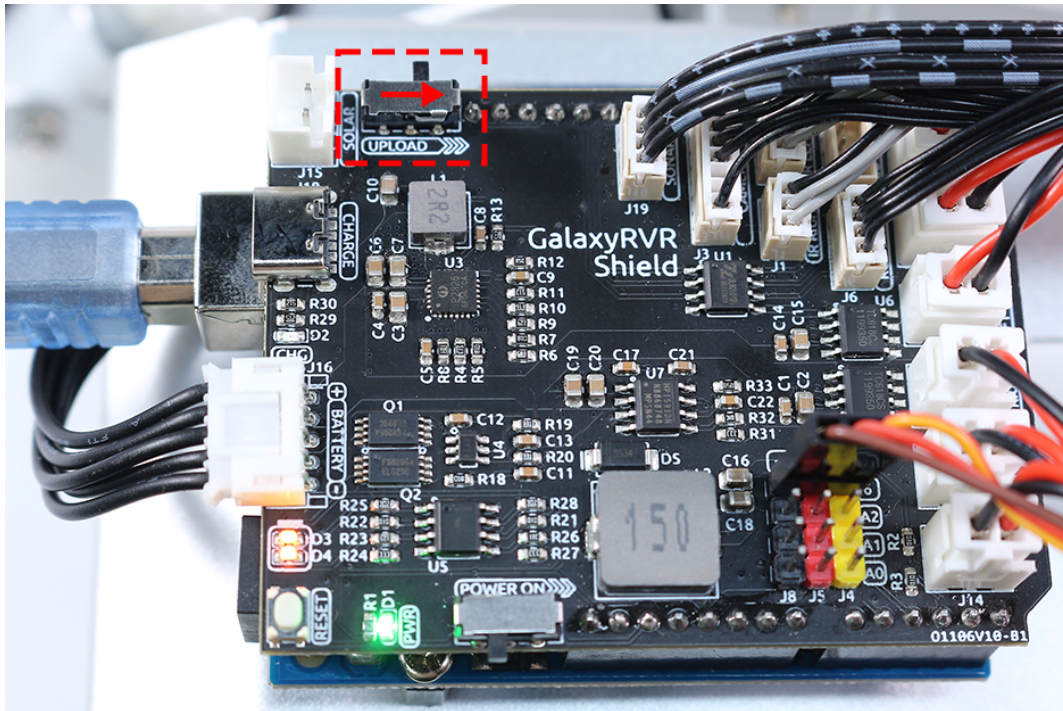
Laden Sie den oben stehenden Code auf den GalaxyRVR, wiederholen Sie die Schritte 4 und 5 von oben, verbinden Sie sich erneut mit dem LAN GalaxyRVR und führen Sie den Vorgang im SunFounder Controller erneut aus, dann können Sie den Schieberegler verwenden, um den Kippmechanismus des Rovers zu steuern.

Jetzt haben Sie erfolgreich gelernt, den SunFounder Controller zu implementieren und wie Sie das Schieberegler-Widget verwenden, um Servobewegungen zu steuern. Dieser Prozess ermöglicht es Ihnen, auf intuitive und direkte Weise mit Ihrem GalaxyRVR zu interagieren.

#### Schritt 4: Reflexion und Zusammenfassung

Die Verwendung des SunFounder Controllers zur Bedienung Ihres Mars-Rovers mag anfangs etwas kompliziert erscheinen. Jedes Mal, wenn Sie Ihren Code ändern, müssen Sie die folgenden Schritte wiederholen:

- Stellen Sie vor dem Hochladen des Codes sicher, dass der Schalter nach rechts gedreht ist.



- Nachdem der Code erfolgreich hochgeladen wurde, schalten Sie auf die linke Seite, um die ESP32-CAM zu starten.
- Verbinden Sie sich mit dem Netzwerk GalaxyRVR.
- Verbinden und starten Sie den Controller.

Obwohl diese Schritte mühsam erscheinen mögen, sind sie entscheidend für den Prozess. Nachdem Sie sie einige Male wiederholt haben, werden Sie mit dem Verfahren vertrauter und wohler fühlen.

Jetzt, da wir diese Lektion abgeschlossen haben, lassen Sie uns über das Nachdenken, was wir gelernt haben, durch einige Fragen reflektieren:

- Bei der Erstellung eines neuen Controllers sind Sie auf viele verschiedene Arten von Bausteinen gestoßen. Haben Sie sich überlegt, welche individuellen Funktionen diese haben könnten?
- Ist es möglich, andere Widgets zu verwenden, um den Kippmechanismus zu steuern?
- Oder sogar direkt die Bewegungen des Mars-Rovers zu steuern?

Lassen Sie uns unsere Erkundung dieser Fragen in der nächsten Lektion mit Spannung erwarten!

## 3.12 Lektion 12: Steuerung des Rovers mit der App

In unserem letzten Abenteuer haben wir gelernt, mit dem SunFounder Controller nicht nur die Welt durch die Augen des Rovers zu erkunden, sondern auch seinen Blick durch Steuerung des Neigungsmechanismus anzupassen. Es war, als hätten wir unserem Mars-Rover das Sehen beigebracht!

Stellen Sie sich nun vor, Sie könnten diesen Mars-Erkunder auch steuern und seinen Weg nach Belieben lenken. In dieser Lektion werden wir genau das tun! Wir vertiefen unser Verständnis des SunFounder Controllers und lernen, unseren Rover zu manövrieren, was uns das aufregende Erlebnis eines Mars-Rover-Fahrers bietet!

Mit unserem lebendigen und anschaulichen STEAM-Kurs werden Kinder nicht nur lernen; sie erleben die Spannung und das Wunder der Raumfahrt! Schnallen Sie sich an und bereiten Sie sich auf eine spannende Reise vor!

### 3.12.1 Kursziele

- Vertiefung des Verständnisses des SunFounder Controllers.
- Erlernen der Steuerung des Mars-Rovers mittels der mobilen App.

### 3.12.2 Kursmaterialien

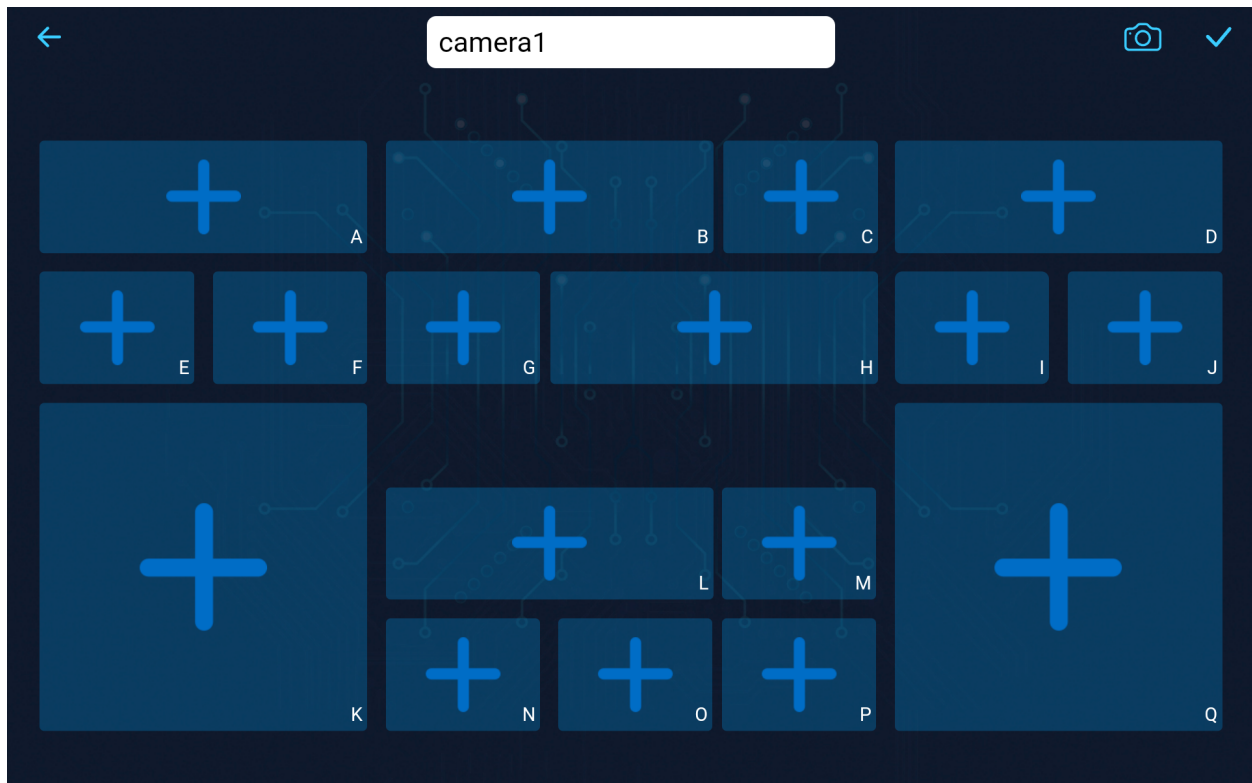
- Mars-Rover-Modell (mit allen Komponenten, außer Solarpanel und Bodenplatte)
- Arduino IDE
- Computer
- Tablet oder Smartphone mit installierter SunFounder Controller-App

### 3.12.3 Kursschritte

#### Schritt 1: Vertiefung in den SunFounder Controller

In unserer letzten Lektion haben wir erste Einblicke in den SunFounder Controller, seine grundlegenden Funktionen und Anwendungen erhalten. Aber sicherlich haben Sie noch einige brennende Fragen, richtig? Zeit, diese Neugier zu stillen und tiefer in dieses technologische Wunder einzutauchen.

Beim Erstellen eines neuen Controllers werden Sie zunächst mit einem Bildschirm konfrontiert, der anfangs wie ein Rätsel erscheinen mag.



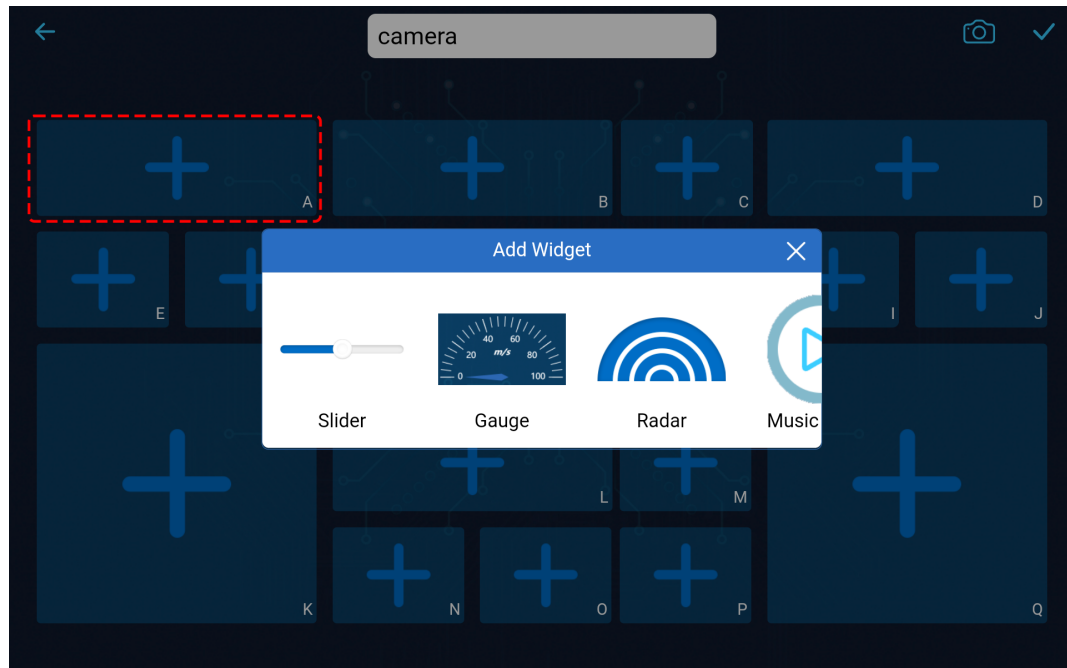
Sie sehen eine Vielfalt von Formen: lange Rechtecke, kurze Rechtecke und große Quadrate, jedes einzigartig mit Kennzeichnungen von A bis Q versehen.

Haben Sie sich jemals gefragt, warum diese Vielfalt? Was bedeuten die Kennzeichnungen von A bis Q?

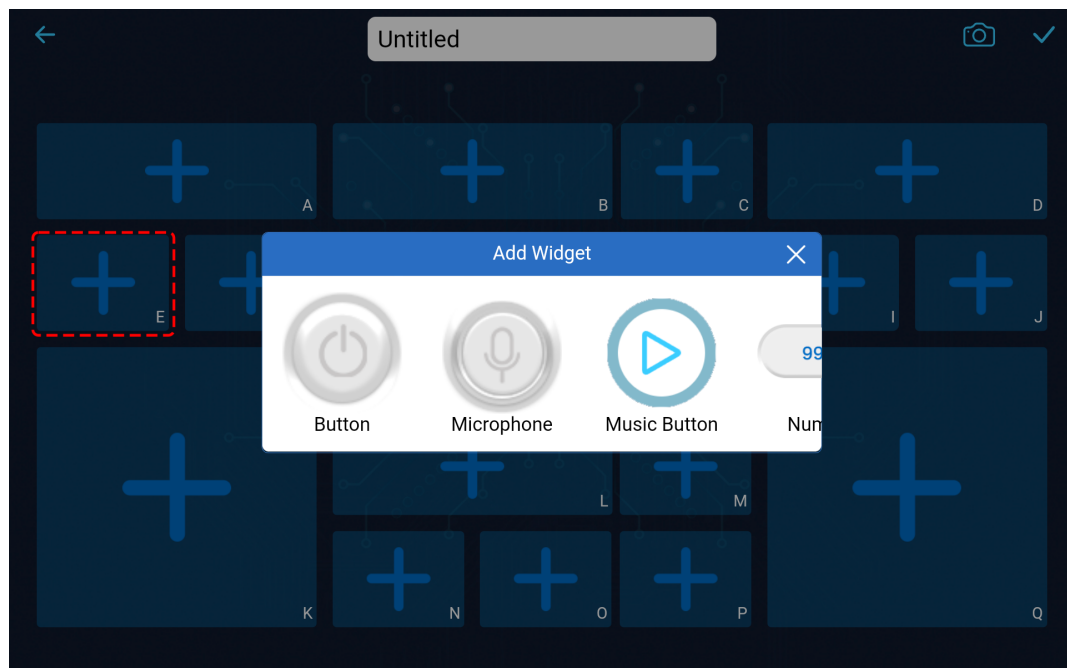
Begeben wir uns auf diese Entdeckungsreise!

- **Entschlüsselung der Formen**

Tippen Sie beispielsweise auf ein langes Rechteck. Wie das Öffnen einer Schatztruhe werden mehrere Widgets enthüllt. Erinnern Sie sich an den **Slider** aus unserem letzten Unterricht? Durch Wischen nach links oder rechts wurde die Neigung der Mars-Rover-Kamera gesteuert. Dann gibt es das **Gauge**, das als Tachometer Ihres Rovers dienen kann. Und eine Vielzahl anderer Widgets!

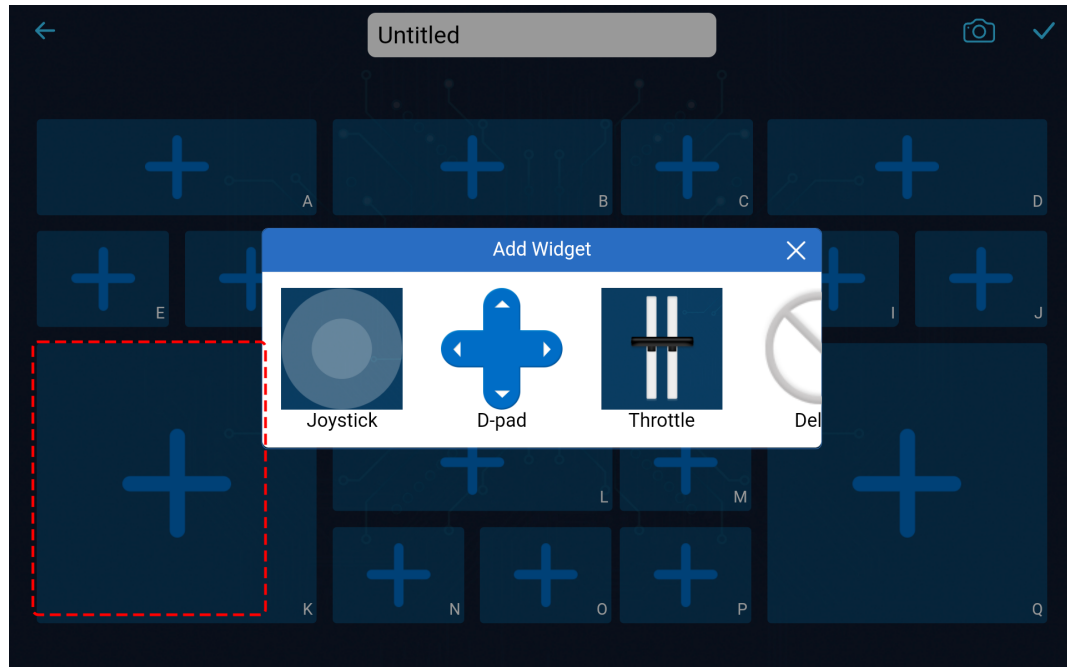


Tippen Sie auf das kürzere Rechteck, und es zeigt eine andere Gruppe von Widgets. Das **Button**-Widget, ähnlich einem binären Schalter, gibt einzelne Befehle aus, während das **Microphone**-Widget Ihre Sprachbefehle zum Steuern des Rovers hören kann, unter anderen Widgets.





Und der quadratische Block? Er beherbergt das **Joystick**-Widget, das D-Pad-Widget und das Gashebel-Widget. Und ja, es gibt noch mehr!



Keine Sorge um die Funktion oder Anwendung jedes Widgets zu diesem Zeitpunkt. Machen Sie sich erst einmal mit den Widgets vertraut, die in jeder Form enthalten sind.

- **Nutzung der Widgets**

Durch den oben genannten Inhalt wissen wir, dass es viele verschiedene Widgets gibt. Wofür können diese Widgets also verwendet werden?

Wir kategorisieren diese Widgets in zwei Typen: Steuerungs-Widgets und Anzeige-Widgets.

- **Steuerungs-Widgets** ermöglichen es Ihnen, die App zu manipulieren, woraufhin der Mars-Rover diese Steuerbefehle empfängt und entsprechende Aktionen ausführt.
- **Anzeige-Widgets** erlauben es Ihnen, bestimmte Werte in der App anzuzeigen, wie zum Beispiel Sensordaten.

Für eine detaillierte Beschreibung dieser Widgets, siehe bitte: .

Für Funktionen, Parameter usw., die mit diesen Widgets zusammenhängen, siehe bitte: .

- **Die Rolle der Kennzeichner**

Jede Form trägt einen Kennzeichner in der unteren rechten Ecke. Warum, fragen Sie? Verstehen wir das anhand eines Beispiels.

In unserer letzten Klasse haben wir ein **Slider**-Widget im Bereich **D** hinzugefügt. Der Code, um dessen Wert zu erhalten, sah in etwa so aus:

```
int16_t sliderD = aiCam.getSlider(REGION_D);
```

Was, wenn wir ein **Slider**-Widget im Bereich **B** hinzugefügt hätten? Wie würde sich der Code ändern? So einfach, wie REGION\_D durch REGION\_B zu ersetzen.

```
int16_t sliderD = aiCam.getSlider(REGION_B);
```



Einfach, richtig? Diese Kennzeichner helfen dabei, zu erkennen, welches Widget Sie in welchem Bereich hinzugefügt haben.

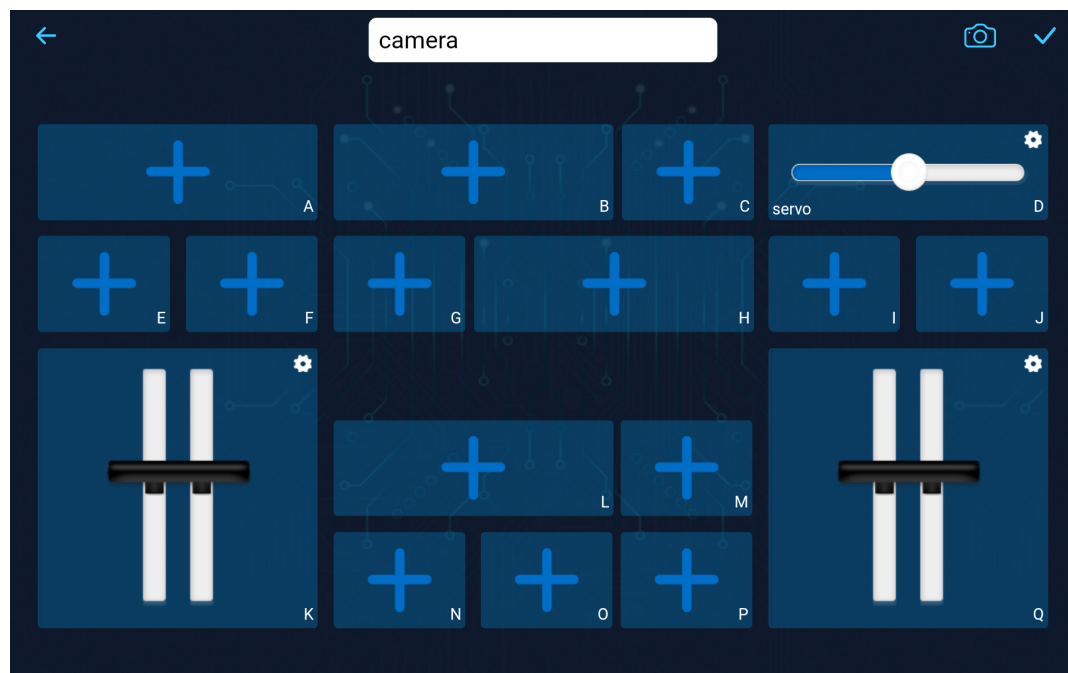
### Bemerkung:

- Detaillierte Anweisungen zum Erstellen und Hinzufügen von Widgets sowie zum Verbinden und Bedienen des SunFounder Controllers finden Sie in **Schritt 3 von Lektion 11**.
- Für tiefgreifendere Operationen besuchen Sie bitte .

### Schritt 2: Steuerung des Mars-Rovers mit Dual-Throttles

Jetzt, da wir uns eingehend mit den SunFounder Controller Widgets beschäftigt haben, steigern wir unser Spiel, indem wir zwei Drosselklappen-Widgets verwenden, um die Bewegung des Mars-Rovers zu steuern.

1. Platzieren Sie je ein **Throttle**-Widget in den Bereichen **K** und **Q**. Sie müssen oben rechts auf die Schaltfläche  klicken, um in den Bearbeitungsmodus zu wechseln, und nachdem Sie alles eingerichtet haben, klicken Sie auf , um Ihre Änderungen zu speichern.



2. Da wir zwei **Throttle**-Widgets zur Steuerung der Mobilität des Rovers verwenden wollen, passen wir die Funktion, die die Bewegungen des Rovers steuert, entsprechend an:

```
// Function to set the power of the motors
void carSetMotors(int8_t power_L, int8_t power_R) {
    // Set power for the left motor
    if (power_L >= 0) {
        SoftPWMSet(in1, map(power_L, 0, 100, 0, 255));
        SoftPWMSet(in2, 0);
    } else {
        SoftPWMSet(in1, 0);
        SoftPWMSet(in2, map(power_L, 0, -100, 0, 255));
    }
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

// Set power for the right motor
if (power_R >= 0) {
    SoftPWMSet(in3, 0);
    SoftPWMSet(in4, map(power_R, 0, 100, 0, 255));
} else {
    SoftPWMSet(in3, map(power_R, 0, -100, 0, 255));
    SoftPWMSet(in4, 0);
}
}

```

Natürlich, lassen Sie uns die Funktion `carSetMotors()` genauer betrachten. Diese Funktion nimmt zwei Argumente entgegen, `power_L` und `power_R`, die wahrscheinlich die Leistungseinstellungen für die linken und rechten Motoren sind. Die Werte dieser Argumente liegen vermutlich zwischen -100 und 100, wobei negative Werte Rückwärtsbewegung anzeigen, 0 einen Stopp bedeutet und positive Werte Vorwärtsbewegung signalisieren.

- Leistungseinstellung für den linken Motor:
  - Wenn `power_L` größer oder gleich 0 ist, wird der linke Motor auf Vorwärtsbewegung eingestellt.
  - `SoftPWMSet(in1, map(power_L, 0, 100, 0, 255))` verwendet die Arduino-Funktion `map`, um den Eingabebereich (0 bis 100) auf den Ausgabebereich (0 bis 255) abzubilden - dies liegt daran, dass die PWM-Werte in Arduino zwischen 0 (0 % Tastverhältnis) und 255 (100 % Tastverhältnis) liegen. Dieser abgebildete Wert wird dann zusammen mit `in1` an die Funktion `SoftPWMSet` übergeben.
  - Wenn `power_L` kleiner als 0 ist, wird der linke Motor auf Rückwärtsbewegung eingestellt, und der Eingabebereich für die Funktion `map` ist jetzt 0 bis -100.
- Leistungseinstellung für den rechten Motor:
  - Dies folgt der gleichen Logik wie das Einstellen der Leistung für den linken Motor, aber benutzt stattdessen `in3` und `in4`, und die Werte für die Eingangsleistung sind `power_R` anstelle von `power_L`.

Insgesamt nimmt diese Funktion zwei Motorleistungswerte, wandelt sie in die entsprechenden PWM-Werte um und setzt die PWM-Werte auf den richtigen Motorsteuerungspins, um die gewünschte Bewegung zu erreichen.

3. Innerhalb der Funktion `onReceive()` holen Sie die Werte von den beiden **Throttle**-Widgets ab und verwenden sie als Leistung für die linken und rechten Motoren des Mars-Rovers.

```

void onReceive() {
    // Get the value of the slider in region D
    int16_t sliderD = aiCam.getSlider(REGION_D);

    // Move the servo to the angle indicated by the slider
    myServo.write(int(sliderD));

    // Get the throttle values for the left and right
    int throttle_L = aiCam.getThrottle(REGION_K);
    int throttle_R = aiCam.getThrottle(REGION_Q);

    // Set the power for the motors

```

(Fortsetzung auf der nächsten Seite)



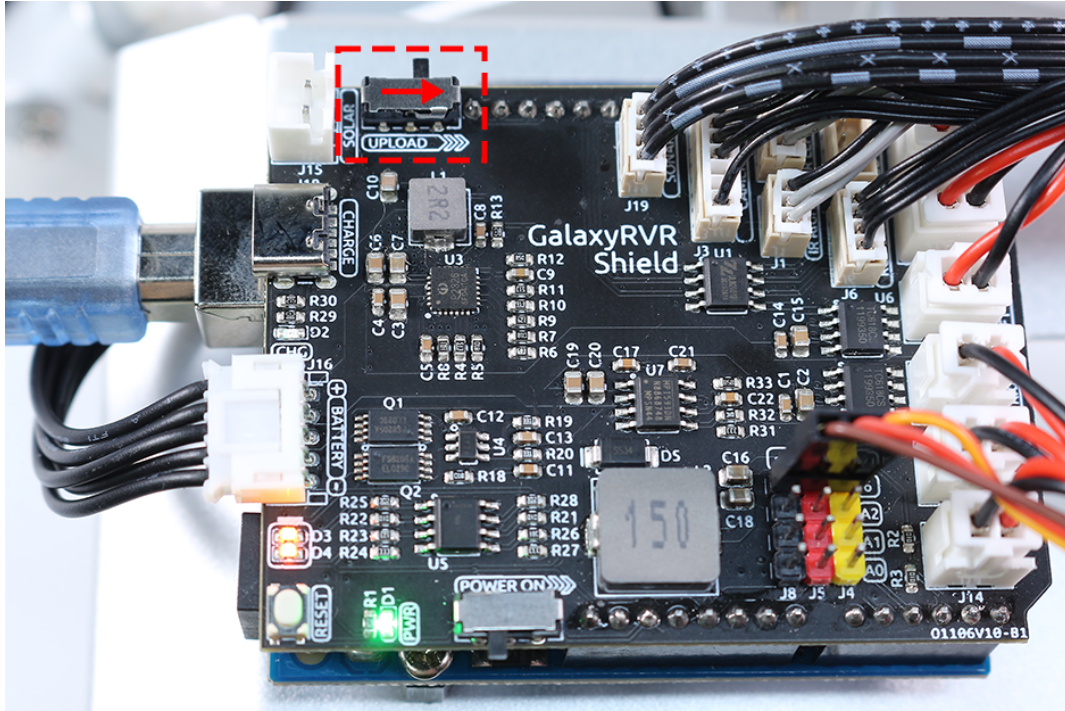
(Fortsetzung der vorherigen Seite)

```
carSetMotors(throttle_L, throttle_R);  
}
```

Hier ist der vollständige Code:

Jedes Mal, wenn Sie den Code erneut ausführen, müssen Sie die folgenden vier Schritte wiederholen:

- Vor dem Hochladen des Codes stellen Sie sicher, dass der Schalter nach rechts gedreht ist.



- Nachdem der Code erfolgreich hochgeladen wurde, schalten Sie auf die linke Seite, um die ESP32 CAM zu starten.
- Verbinden Sie sich mit dem Netzwerk GalaxyRVR.
- Verbinden Sie sich und führen Sie den Controller aus.

Nun können Sie mit einer einfachen Bewegung Ihres Daumens auf den Throttle-Widgets den Mars-Rover in Aktion sehen, wie er mit erneuertem Geist an Kraft gewinnt und sich dreht. Lasst die Erkundung beginnen!

### Schritt 3: Visualisierung von Sensorwerten

Auf unserer Reise mit dem SunFounder Controller haben wir aktiv mit unserem Mars-Rover interagiert, unter anderem durch Steuerungs-Widgets wie Schieberegler zur Neigungsanpassung und Drosselklappen zur Bewegungssteuerung des Rovers. Aber wie steht es mit der Nutzung von Anzeige-Widgets, um ein lebendiges Bild von der Umgebung unseres Rovers zu zeichnen?

Lassen Sie uns sehen, wie wir dies umsetzen können, indem wir die Werte von den linken und rechten Infrarot (IR)-Vermeidungsmodulen und die vom Ultraschallmodul erfasste Entfernung visualisieren. Diese Echtzeitdaten geben uns ein klares Bild von der Betriebsumgebung des Rovers.

So können wir das erreichen:

1. Beginnen Sie damit, drei Nummern-Widgets zu Ihrem SunFounder Controller hinzuzufügen. Vergessen Sie nicht, Sie können deren Namen und Einheiten mit dem Einstellungsbutton personalisieren.



2. Als Nächstes tauchen wir in den Code ein. Starten Sie, indem Sie die Code-Schnipsel für die IR-Vermeidungsmodule und das Ultraschallmodul aus unseren vorherigen Lektionen übernehmen.

```
...
// Define the pin for the ultrasonic module
#define ULTRASONIC_PIN 10

// Define the pins for the IR modules
#define IR_RIGHT 7
#define IR_LEFT 8

void setup() {
    ...

    // Set the IR module pins as inputs
    pinMode(IR_RIGHT, INPUT);
    pinMode(IR_LEFT, INPUT);
}

float readSensorData() {
    // A 4ms delay is required, otherwise the reading may be 0
    delay(4);

    //Set to OUTPUT to send signal
    pinMode(ULTRASONIC_PIN, OUTPUT);

    ...
}
```

3. In der Funktion `onReceive()` extrahieren Sie die Werte aus den Vermeidungsmodulen und dem Ultraschallsensor. Aktualisieren Sie anschließend diese Werte im Wörterbuch `sendDoc[]`. Die Schlüssel `N`, `P` und `O` entsprechen den Regionscodes der drei hinzugefügten Nummern-Widgets.

```
// Function to execute when data is received from the Controller
```

(Fortsetzung auf der nächsten Seite)



(Fortsetzung der vorherigen Seite)

```

void onReceive() {

    ...

    // Read values from IR sensors
    int leftValue = digitalRead(IR_LEFT);
    int rightValue = digitalRead(IR_RIGHT);
    aiCam.sendDoc["N"] = leftValue;
    aiCam.sendDoc["P"] = rightValue;

    // ultrasonic
    float distance = readSensorData();
    aiCam.sendDoc["0"] = distance;
}

```

Hier ist der vollständige Code:

Nachdem der Code erfolgreich hochgeladen wurde, starten Sie Ihren SunFounder Controller. Sie werden mit den Echtzeitwerten der Vermeidungsmodule und der vom Ultraschallsensor erfassten Entfernung begrüßt, was ein klares Bild von der unmittelbaren Umgebung des Rovers zeichnet.



Mit diesem Schritt haben Sie die Welt der Anzeige-Widgets erfolgreich erkundet. Experimentieren Sie ruhig mit ver-

schiedenen Widgets, um die Informationen anzuzeigen, die Sie interessant finden. Viel Spaß beim Erforschen!

#### **Schritt 4: Reflexion und Schlussfolgerung**

In dieser Lektion haben wir ein tieferes Verständnis für den SunFounder Controller entwickelt und verstanden, wie wir seine Widgets nicht nur zur Steuerung unseres Mars-Rovers, sondern auch zur Überwachung seiner Umweltdaten in Echtzeit nutzen können.

Nun, hier ist eine Herausforderung für Sie:

Wie wäre es, einige Schalter-Widgets zu Ihrem SunFounder Controller hinzuzufügen? Mit diesen aktivierten Schaltern könnte der Mars-Rover zwischen Vermeidungs- und Folgemodi umschalten. Oder nutzen Sie die Schalter zur Steuerung des Lichtstreifens – ihn ein- oder ausschalten oder sogar seine Farbe ändern?

Haben Sie das Vertrauen, diese Herausforderung anzunehmen?

Wir freuen uns darauf, Sie diese Herausforderung meistern zu sehen!

### **3.13 Lektion 13: Untersuchung des Energiesystems des Mars-Rovers**

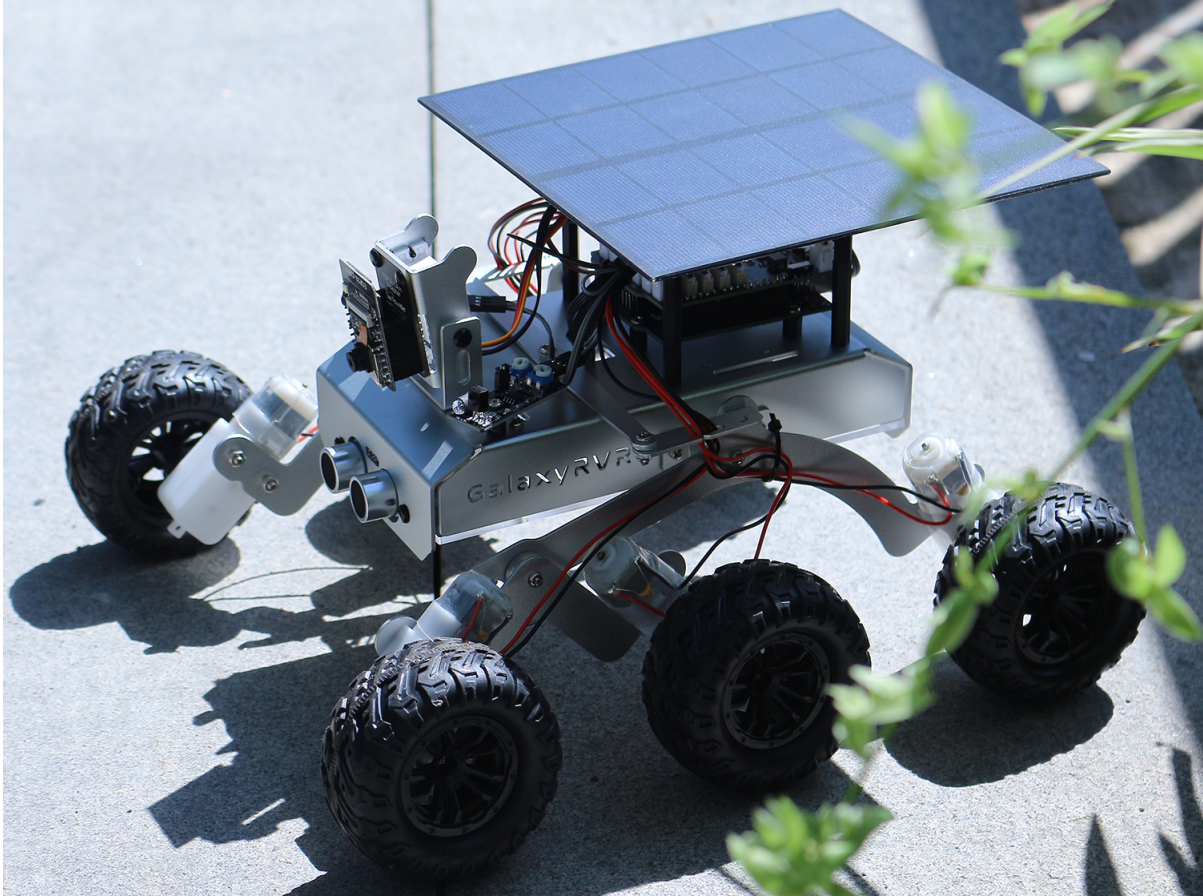
Willkommen zur letzten Lektion unserer Reise zur Erforschung des Mars-Rovers. Diesmal tauchen wir in das Herz des Rovers ein - sein Energiesystem.

Wenn wir über die Erforschung ferner Planeten wie Mars nachdenken, ist einer der wichtigsten Aspekte die Energie. Wie versorgen sich diese Rover in solch rauen und abgelegenen Umgebungen selbst mit Strom? In dieser Lektion werden wir dieses faszinierende Thema erkunden und erfahren, wie Rover, wie unser Mars-Rover-Modell, Energie nutzen und verwalten.

Wir werden die Arbeitsprinzipien von Batterie und Solarpanel untersuchen und sogar praktische Erfahrungen im Installieren und Nutzen dieser Energiequellen an unserem Rover-Modell sammeln. Darüber hinaus werden wir unsere Fähigkeiten mit Arduino verbessern, um den Batteriestand zu überwachen.

Am Ende der Lektion werden wir in der Lage sein, diese vitalen Energiestatistiken direkt in unserer App anzuzeigen, was uns ein Echtzeitverständnis des Energiestatus unseres Rovers ermöglicht. Bereit, unsere Erkundung mit Energie zu starten? Lass uns beginnen!





### 3.13.1 Kursziele

- Verstehen der Arbeitsprinzipien von Batterie und Solarpanel.
- Praktische Erfahrungen im Installieren des Solarpanels am Mars-Rover-Modell.
- Erlernen der Überwachung des Batteriestandes und des Ladezustandes des Solarpanels mit Arduino.
- Anzeige des Batteriestandes in der App.

### 3.13.2 Kursmaterialien

- Mars-Rover-Modell (mit allen Komponenten, außer Solarpanel und Bodenplatte)
- Solarpanel und Bodenplatte
- Arduino IDE
- Computer
- Tablet oder Smartphone mit installierter SunFounder Controller-App

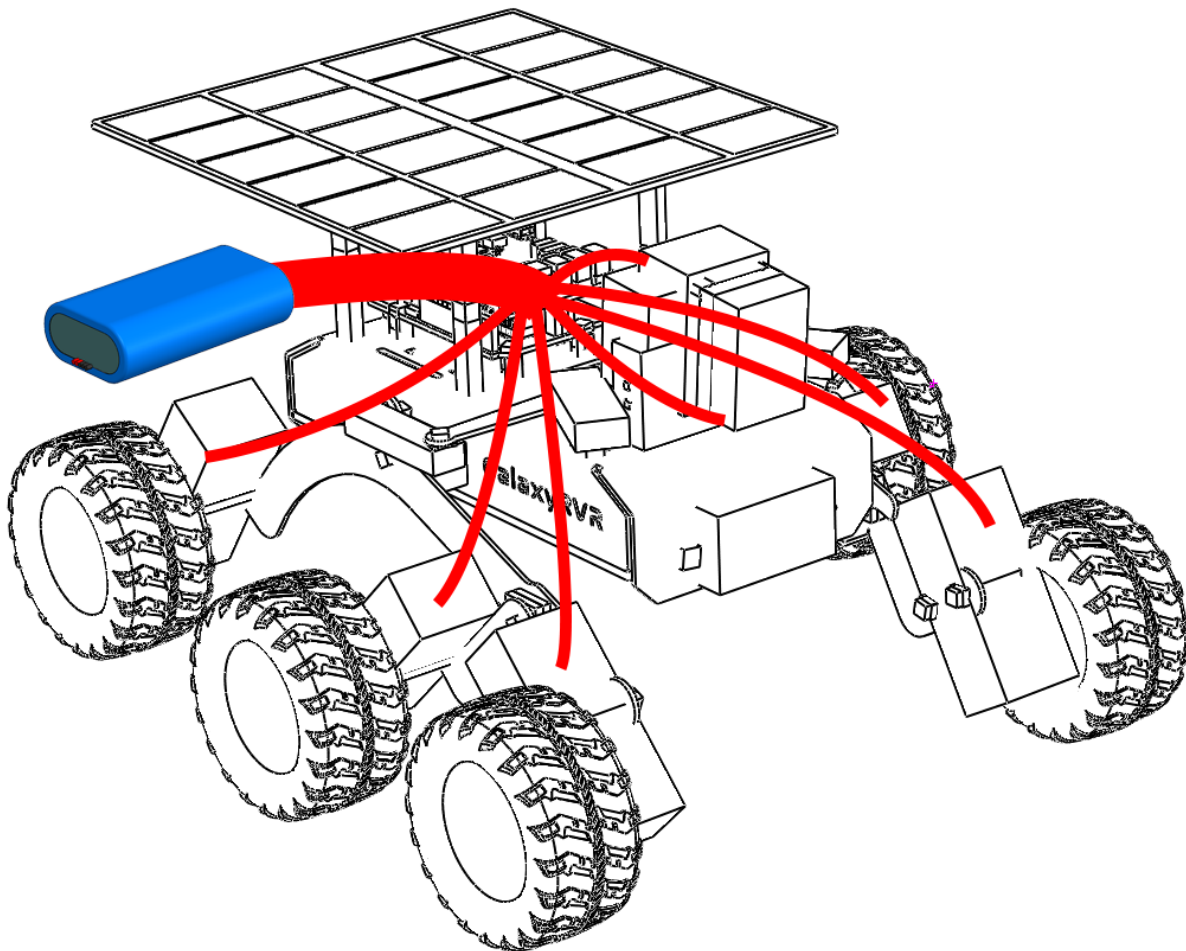
### 3.13.3 Kursschritte

#### Schritt 1: Einführung in das Energiesystem des Mars-Rovers

Genau wie unser Körper eine ständige Energiezufuhr benötigt, braucht unser Mars-Rover eine Möglichkeit, Energie für seine Erkundungsmissionen zu speichern und zu erzeugen. Stellen Sie sich das Energiesystem des Rovers vor wie das Herz in unserem Körper. Genau wie unser Herz Blut in alle Teile unseres Körpers pumpt und notwendigen Sauerstoff und Nährstoffe liefert, hält das Energiesystem des Rovers die Energie in allen Teilen des Rovers fließend, um sicherzustellen, dass er seine Aufgaben reibungslos erfüllen kann.

Die Hauptkomponenten dieses Energiesystems sind die Batterien und die Solarpanels, die zusammenarbeiten, um sicherzustellen, dass der Rover jederzeit, Tag und Nacht, betriebsbereit ist.

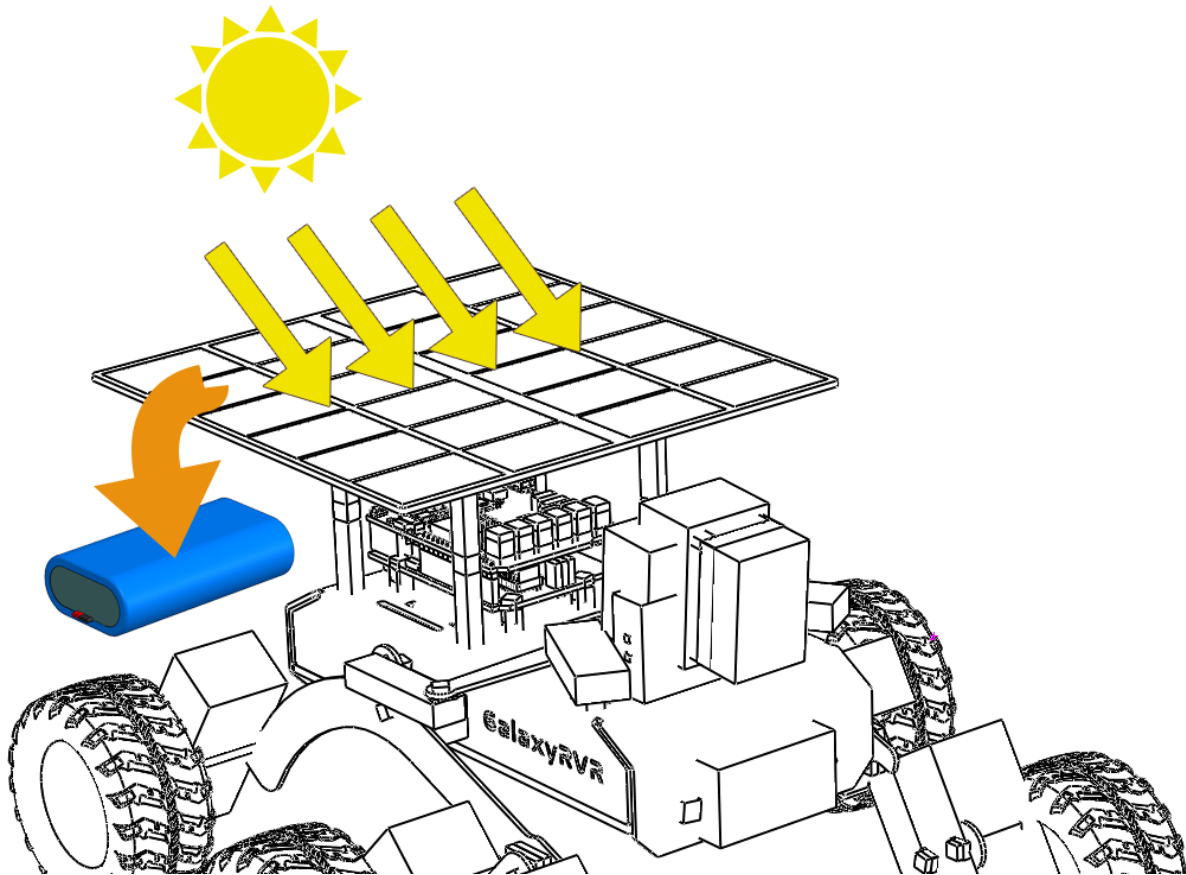
Die Rolle der Batterien im Energiesystem des Rovers ist ähnlich der Rolle der Energiespeicherung in unserem Körper. Genau wie wir Energie speichern müssen, um sie bei Aktivität zu nutzen, braucht der Rover eine Möglichkeit, Energie für seine Erkundungsmissionen zu speichern. Die in den Batterien gespeicherte Energie wird kontinuierlich an verschiedene Teile des Rovers abgegeben, damit er seine Aufgaben systematisch durchführen kann.



Aber was passiert, wenn die Energie in den Batterien zur Neige geht? Wie wird seine Energie wieder aufgefüllt? Hier kommen die Solarpanels ins Spiel.

Ähnlich wie Bäume Sonnenlicht für die Photosynthese aufnehmen, um Nahrung zu erzeugen, nutzt unser Mars-Rover Solarpanels, um Energie von der Sonne zu nutzen und sie in Elektrizität umzuwandeln, die in den Batterien gespeichert wird. Jedes Solarpanel besteht aus vielen kleineren Solarzellen. Diese Zellen bestehen aus einem Material, das Licht in Elektrizität umwandeln kann – ein Prozess, der als photovoltaischer Effekt bezeichnet wird. Wenn Sonnenlicht auf die

Zellen trifft, erzeugen sie einen elektrischen Strom, der sofort genutzt oder in den Batterien des Rovers für die spätere Nutzung gespeichert werden kann.



Jedoch ist das Nutzen von Solarenergie auf dem Mars nicht so einfach, wie es klingt. Staubstürme können die Menge des Sonnenlichts, die die Panele erreicht, verringern, und das schwächere Mars-Sonnenlicht (im Vergleich zur Erde) bedeutet, dass die Panele weniger Strom erzeugen, als sie es hier zu Hause würden. Trotz dieser Herausforderungen ist Solarenergie immer noch die praktischste und effizienteste Art, unseren Mars-Rover mit Strom zu versorgen.

Aber wie wissen wir, wann die Solarpanele ihre Arbeit verrichten und wann die Batterien an Leistung verlieren? Hier kommt unser Arduino ins Spiel. Im nächsten Abschnitt lernen wir, wie wir Arduino nutzen können, um das Laden und Entladen der Rover-Batterien zu überwachen.

### Schritt 2: Montage des Solarpanels am Mars-Rover

Bevor wir mit diesem Schritt beginnen, benötigen wir unser Mars-Rover-Modell, ein Solarpanel und die notwendigen Kabel, um das Solarpanel mit dem Stromsystem des Rovers zu verbinden.

Dies ist ein Prozess, der es uns ermöglicht, Theorie in die Praxis umzusetzen und die Faszination der Ausbildung in Wissenschaft, Technik, Ingenieurwesen und Mathematik (STEM) wirklich zu schätzen. Lass uns beginnen!

### Schritt 3: Programmierung zur Überwachung der Batteriespannung und -ladung

Jetzt, da wir die Solarpanels an unserem Mars-Rover-Modell installiert haben, besteht der nächste Schritt darin, die Spannung und Ladung der Batterie durch Programmierung zu überwachen.

Dieser Code erstellt effektiv einen einfachen Batteriemonitor, der besonders nützlich in Anwendungen wie dem Mars-Rover ist, wo das Energiemanagement entscheidend ist. Er ermöglicht es Ihnen, den Zustand der Batterie zu überwachen und zu verstehen, wann der Rover aufgeladen werden muss oder wann energieintensive Aufgaben geplant werden sollten.



Sicher, lassen Sie uns die verschiedenen Teile dieses Codes erläutern:

- Diese Zeile definiert BATTERY\_PIN als den analogen Pin A3, von dem die Batteriespannung abgelesen wird.

```
#define BATTERY_PIN A3
```

- Diese Funktion berechnet die Spannung der Batterie. Zuerst liest sie den analogen Wert von BATTERY\_PIN, dann wandelt sie ihn in Spannung um. Da der analoge-digital-Wandler (ADC) des Arduino auf einer Skala von 0-1023 arbeitet, teilen wir den Rohwert durch 1023. Wir multiplizieren dann mit 5 (die Referenzspannung des Arduino) und mit 2 (unter der Annahme eines Spannungsteilers von 2), um dies in eine Spannungsanzeige umzuwandeln.

```
float batteryGetVoltage() {
    // Reads the analog value from the battery pin
    int adcValue = analogRead(BATTERY_PIN);
    // Converts the analog value to voltage
    float adcVoltage = adcValue / 1023.0 * 5 * 2;
    // Rounds the voltage to two decimal places
    float batteryVoltage = int(adcVoltage * 100) / 100.0;
    return batteryVoltage;
}
```

Der rohe ADC-Wert des Arduino-Analog-Digital-Umsetzers wird durch 1023 geteilt, um ihn in einen Bruchteil umzuwandeln, und dann mit 5 multipliziert, um ihn in Spannung umzusetzen, da Arduino eine Referenzspannung von 5 Volt verwendet.

Da jedoch die Batteriespannung höher als die maximale Eingangsspannung des Arduino ist, wird ein Widerstand verwendet, um den Arduino zu schützen. Daher multiplizieren wir die ADC-Spannung mit 2, um den Effekt des Widerstands auszugleichen und die korrekte Batteriespannung zu erhalten.

- Diese Funktion berechnet den Ladungsprozentsatz der Batterie anhand ihrer Spannung. Sie verwendet die Funktion map, um den Spannungswert (der zwischen 6,6 und 8,4 Volt liegt) in einen Prozentsatz (der zwischen 0 und 100 liegt) umzuwandeln.

```
uint8_t batteryGetPercentage() {
    float voltage = batteryGetVoltage(); // Gets the battery voltage
    // Maps the voltage to a percentage.
    int16_t temp = map(voltage, 6.6, 8.4, 0, 100);
    // Ensures the percentage is between 0 and 100
    uint8_t percentage = max(min(temp, 100), 0);
    return percentage;
}
```

#### Schritt 4: Test des Energiesystems des Mars-Rovers: Innen- und Außeneinsätze

Nachdem wir unser Batterieüberwachungssystem programmiert haben, ist es nun an der Zeit, den Mars-Rover in Aktion zu setzen. Beginnen Sie damit, den Rover vollständig aufzuladen, und planen Sie zwei 30-minütige Erkundungsmissionen - eine in Innenräumen, und eine weitere im Sonnenlicht im Freien. Zeichnen Sie den anfänglichen Batteriestand vor jeder Mission auf und vergleichen Sie ihn mit dem Batterieprozentsatz am Ende jedes Tests. Die folgende Tabelle dient als nützliche Vorlage, um Ihre Erkenntnisse festzuhalten:

Tab. 1: Leistungstest

	Sonnenschein	Innenraum
Anfangs Batterieprozentsatz		
Endgültiger Batterieprozentsatz		

Beobachten Sie den Unterschied in den Batterieständen nach jedem Test. Hält die Batterie des Rovers länger, als er im

Freien im Sonnenlicht stand? Welche Schlussfolgerungen können wir über die Wirksamkeit des Solarpanels aus dieser Beobachtung ziehen?

Das Verständnis dieser Unterschiede wird uns helfen, besser zu verstehen, wie Solarenergie einen Mars-Rover effektiv antreiben kann, selbst in abgelegenen, harschen Umgebungen wie denen auf der Marsoberfläche.

### Schritt 5: Anzeige des Batteriestands in der App

Natürlich ist es unpraktisch, wiederholt neuen Code auf den Mars-Rover hochzuladen, nur um den verbleibenden Batteriestand zu überprüfen. Das könnte ziemlich umständlich sein.

Stattdessen können wir den Batteriestand an eine App senden, die es uns ermöglicht, einfach zu überwachen, wie viel Spielzeit noch verbleibt, während wir Spaß haben!

Aus früheren Lektionen haben wir gelernt, dass wir, wenn wir Daten auf dem SunFounder Controller anzeigen oder den Mars-Rover mit Widgets steuern wollen, diese Widgets zuerst hinzufügen müssen.

Die Anzeige des Batteriestands ist jedoch ein spezielles Widget. Es hat einen eigenen Schlüssel (BV), und seine Anzeige befindet sich nicht in den von A bis Q gekennzeichneten Bereichen. Stattdessen wird es durch ein Batteriesymbol in der oberen rechten Ecke dargestellt.

So fügen wir es in unseren Code ein:

```
...
// This pin reads the voltage of the battery
#define BATTERY_PIN A3
...

void setup() {
    ...
    // Sets the battery pin as an input
    pinMode(BATTERY_PIN, INPUT);
}

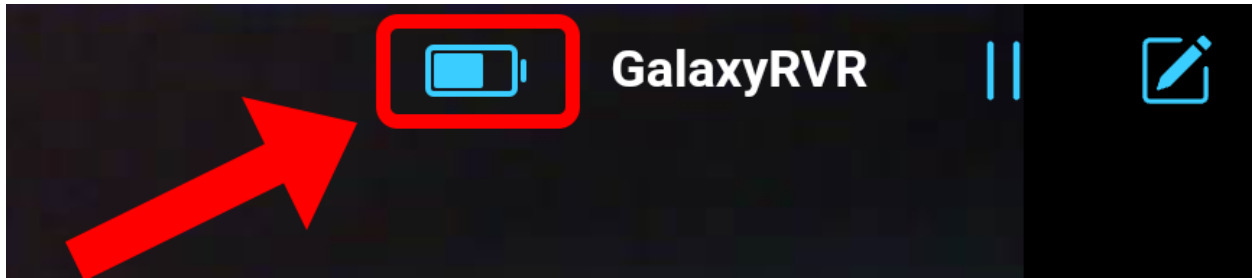
// Function to execute when data is received from the Controller
void onReceive() {
    ...
    //show battery voltage
    aiCam.sendDoc["BV"] = batteryGetVoltage();
}

...

// This function reads the battery voltage
float batteryGetVoltage() {
    // Reads the analog value from the battery pin
    int adcValue = analogRead(BATTERY_PIN);
    // Converts the analog value to voltage
    float adcVoltage = adcValue / 1023.0 * 5 * 2;
    // Rounds the voltage to two decimal places
    float batteryVoltage = int(adcVoltage * 100) / 100.0;
    return batteryVoltage;
}
```

Bitte finden Sie den vollständigen Code im Anhang:

Nachdem Sie den Code erfolgreich hochgeladen haben, starten Sie Ihren SunFounder Controller. Sie werden den Batteriestand in der oberen rechten Ecke sehen.



Mit diesem Schritt haben wir das Energiesystem des Mars-Rovers gemeistert und die Fähigkeit erlangt, seine Energielevels in Echtzeit zu überwachen.

Jetzt, da wir gelernt haben, wie wir die Kraft der Sonne nutzen können, um den Mars-Rover effektiv zu betreiben, können wir mit der Planung umfangreicherer Erkundungen unseres Hinterhofs oder sogar den Aufbruch in herausforderndere Gelände beginnen!

### **Schritt 6: Reflexion**

In dieser Lektion konzentrierten wir uns darauf, die entscheidende Rolle des Energiesystems im Mars-Rover zu verstehen und die Mechanismen zur Überwachung der verbleibenden Energie des Rovers. Das auf Solarmodulen basierende Energiesystem versorgt nicht nur den Rover mit Strom, sondern unterstreicht auch die Bedeutung erneuerbarer Energiequellen in der Raumfahrt.

Mit dem Wissen, das Sie jetzt haben, denken Sie über die realen Auswirkungen dieses Systems nach. Überlegen Sie, welche Herausforderungen ein Solarenergiesystem auf dem Mars antreffen könnte. Wie könnten extreme Temperaturen, Staubstürme oder lange Dunkelphasen die Energieversorgung beeinflussen? Welche Lösungen könnten Sie vorschlagen, um diese Hindernisse zu überwinden?

### **Schritt 7: Ausblick**

Jetzt, da wir unserem Mars-Rover die Fähigkeit zur Bewegung gegeben haben, ist es an der Zeit, ihn seine Erkundungsreise beginnen zu lassen! Sie können ihn in verschiedenen Geländearten, die die Mars-Umgebung nachahmen, wandern lassen.

Zum Beispiel können Sie ihn über einen Haufen Steine klettern lassen.

Oder lassen Sie ihn durch einen dichten Grasfleck navigieren.

Oder setzen Sie ihn auf einen Kurs in einem Kiesgelände voller Steine.

Bitte beachten Sie jedoch, dass der Rover möglicherweise nicht in der Lage ist, über zu hohe Hindernisse zu klettern.

Diese unterschiedlichen Geländearten stellen einzigartige Herausforderungen für den Rover dar, genauso wie sie es für einen echten Mars-Rover tun würden. Während Sie zusehen, wie Ihr Rover versucht, diese Hindernisse zu überwinden, erleben Sie einen kleinen Teil dessen, was Wissenschaftler und Ingenieure bei der NASA tun, wenn sie Rover zum Mars schicken!

Während wir unsere Mars-Rover-Lektionen abschließen, ist es wichtig, über das Gelernte nachzudenken. Wir hoffen, dass diese Reise nicht nur Ihr Wissen und Ihre Fähigkeiten erweitert hat, sondern auch Neugier und den Wunsch zu erkunden geweckt hat. Ob Ihr Rover in Ihrem Hinterhof oder in der weiten Ausdehnung Ihrer Fantasie umherstreift, die Entdeckungen, die Sie auf dem Weg machen, werden sicherlich außergewöhnlich sein.



## KAPITEL 4

---

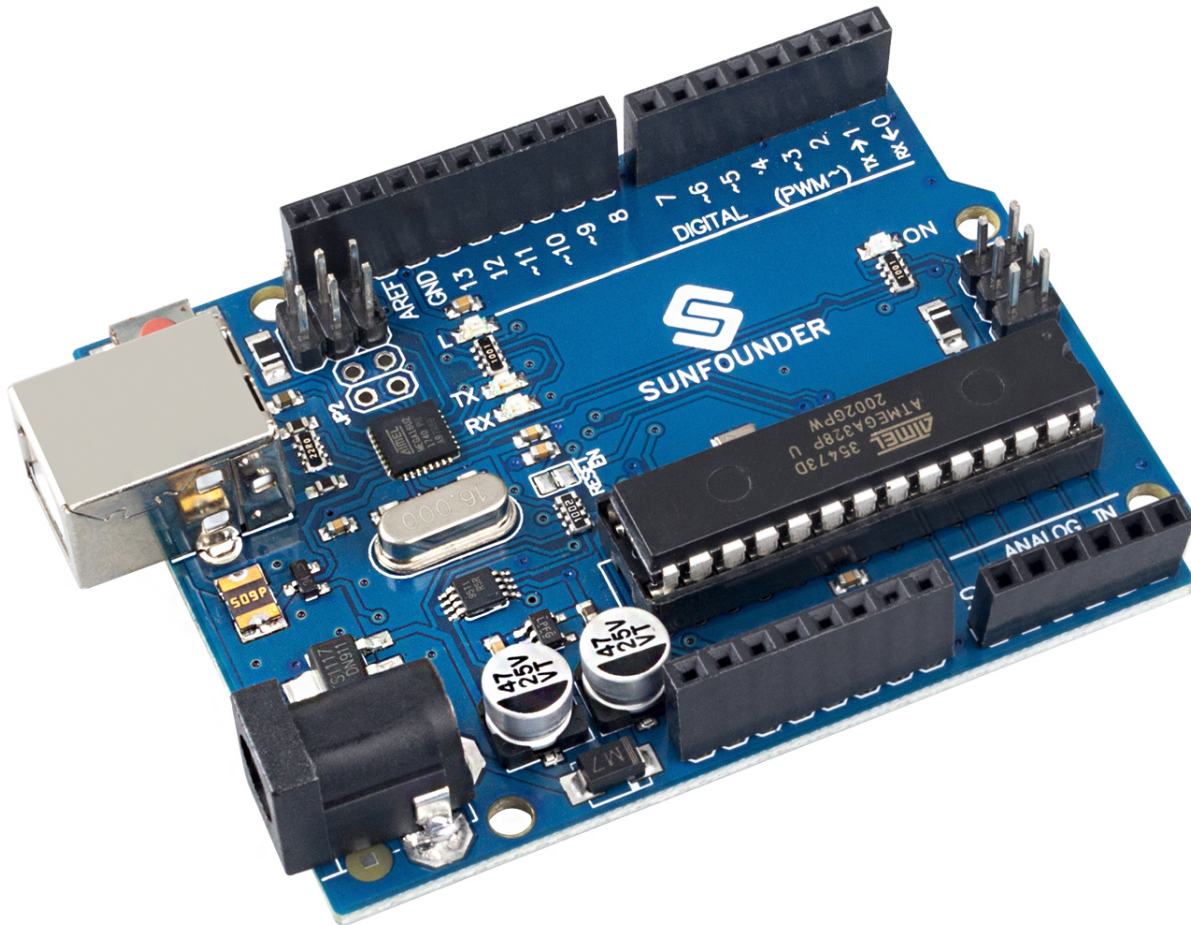
### Hardware

---

Wenn Sie Code schreiben, müssen Sie vielleicht wissen, wie jedes Modul funktioniert oder welche Rolle jeder Pin spielt, dann schauen Sie bitte in dieses Kapitel.

In diesem Kapitel finden Sie eine Beschreibung der Funktion jedes Moduls, der technischen Parameter und des Arbeitsprinzips.

## 4.1 SunFounder R3 Board



---

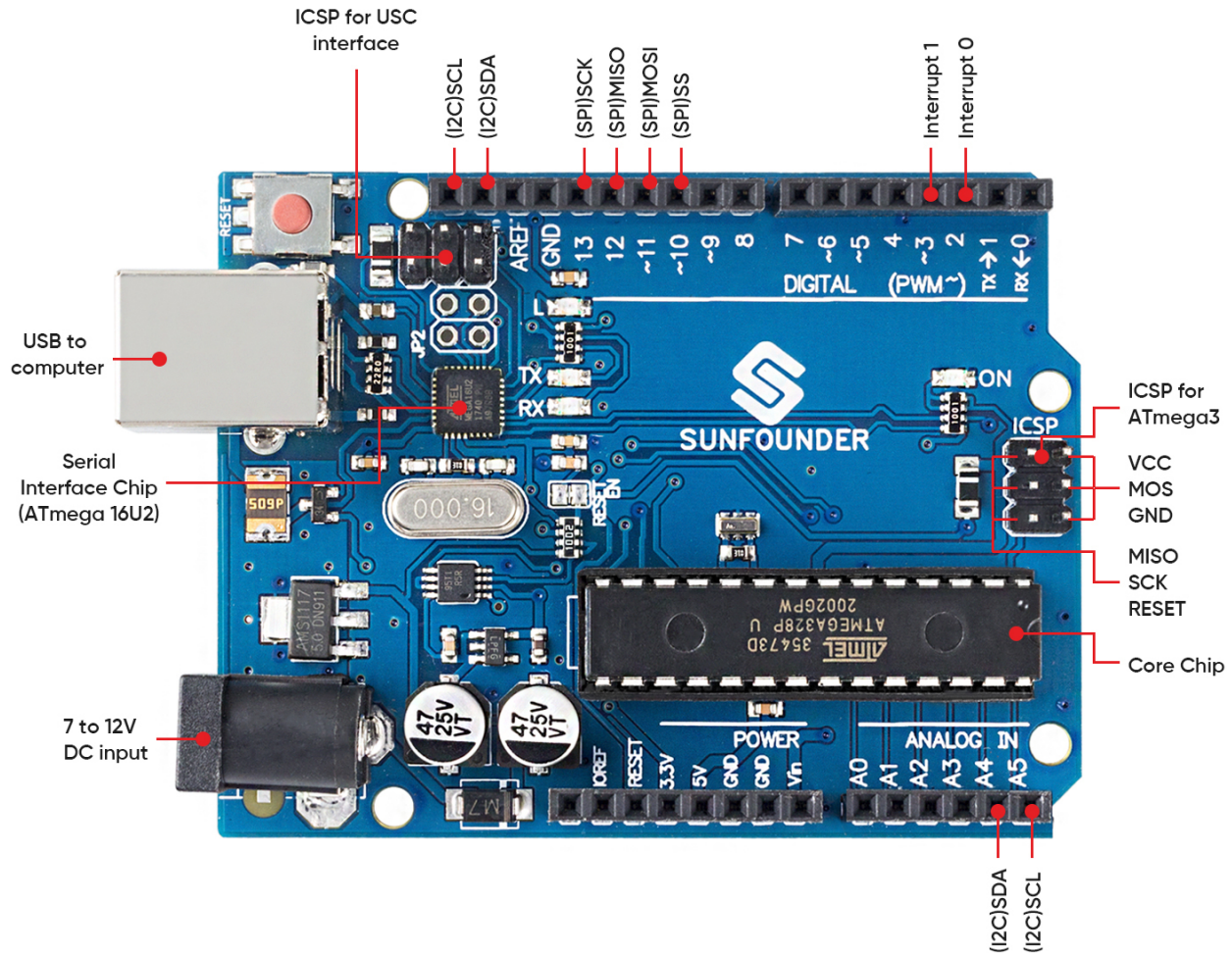
**Bemerkung:** Das SunFounder R3 Board ist ein Hauptboard mit fast denselben Funktionen wie das [Arduino Uno](#), und beide Boards können austauschbar verwendet werden.

---

Das SunFounder R3 Board ist ein Mikrocontroller-Board, das auf dem ATmega328P basiert ([Datenblatt](#)). Es hat 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 benötigt wird, um den Mikrocontroller zu unterstützen; einfach mit einem USB-Kabel an einen Computer anschließen oder mit einem AC-DC-Adapter oder einer Batterie betreiben, um zu starten.

### Technische Parameter





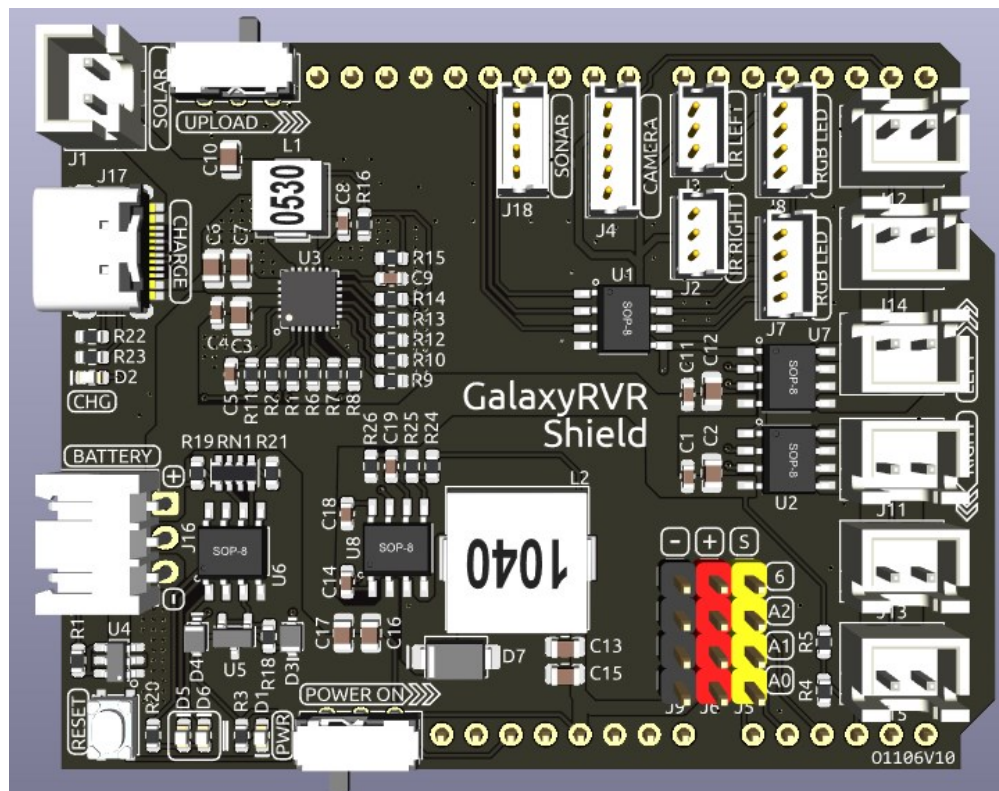
- MIKROCONTROLLER: ATmega328P
- BETRIEBSSPANNUNG: 5V
- EINGANGSSPANNUNG (EMPFOHLEN): 7-12V
- EINGANGSSPANNUNG (LIMIT): 6-20V
- DIGITALE I/O-PINS: 14 (0-13, davon 6 für PWM-Ausgang(3, 5, 6, 9-11))
- PWM DIGITALE I/O-PINS: 6 (3, 5, 6, 9-11)
- ANALOGE EINGANGSPINS: 6 (A0-A5)
- DC STROM PRO I/O-PIN: 20 mA
- DC STROM FÜR 3.3V PIN: 50 mA
- FLASH-SPEICHER: 32 KB (ATmega328P), davon 0,5 KB durch Bootloader verwendet
- 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-Port: A4(SDA), A5(SCL)

### Was gibt's noch

- Arduino IDE
- Arduino Programmiersprachenreferenz
- ATmega328P Datenblatt

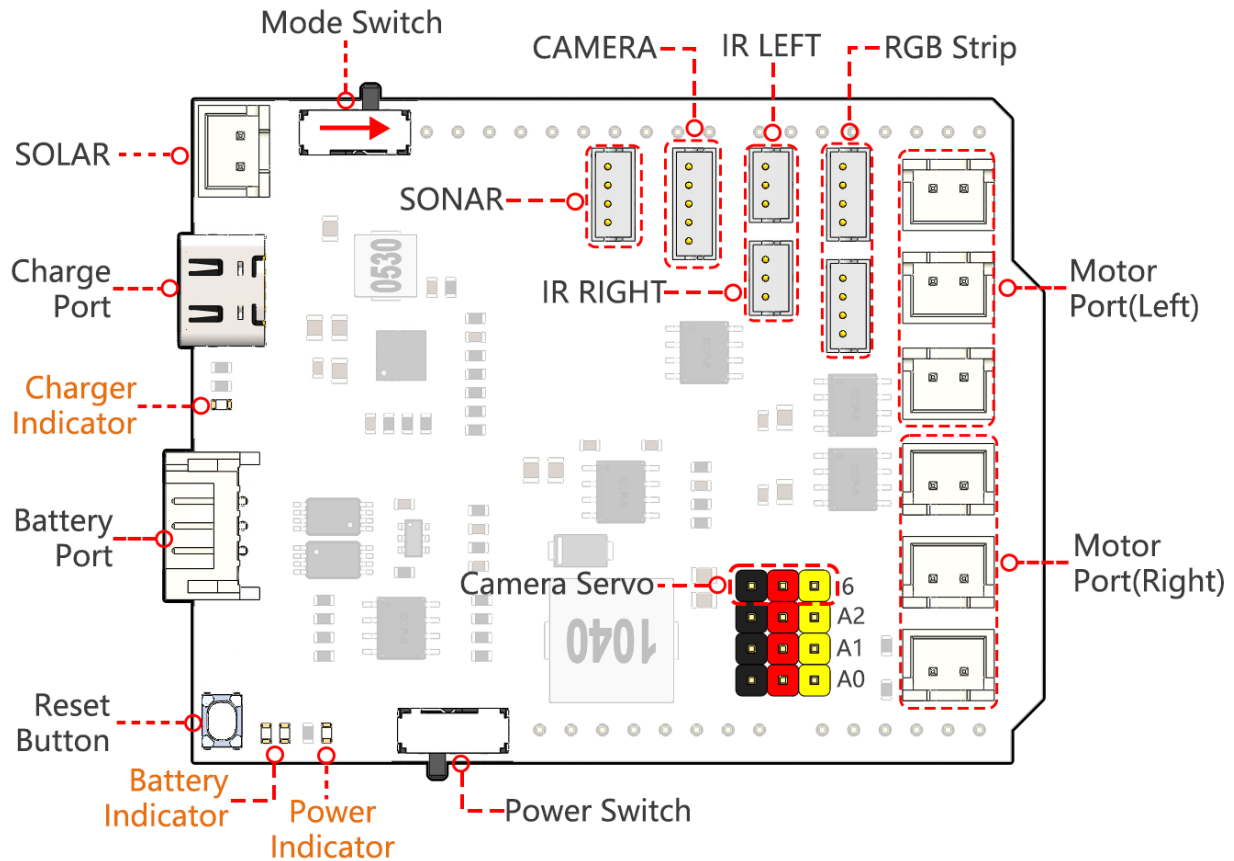
## 4.2 GalaxyRVR Shield



Dies ist ein All-in-One-Erweiterungsboard, das von SunFounder für Arduino entwickelt wurde und verschiedene Modulan­schlüsse enthält, wie Motor, RGB-Streifen, Hindernisvermeidung, Graustufen, ESP32 CAM und Ultraschallmodul.

Dieses Erweiterungsboard verfügt außerdem über einen integrierten Ladekreislauf, der den Akku mit PH2.0-3P Schnittstelle aufladen kann, und die geschätzte Ladezeit beträgt 130 Minuten.

### Pinbelegung



- **Charge Port**

- Nach dem Einstecken in den 5V/2A USB-C-Port kann es zum Aufladen des Akkus für 130 Minuten verwendet werden.

- **Battery Port:**

- 6.6V~8.4V PH2.0-3P-Stromversorgungseingang.
- Versorgt das GalaxyRVR Shield und das Arduino-Board gleichzeitig mit Strom.

- **Reset Button**

- Drücken Sie diese Taste, um das Programm auf dem Arduino-Board zurückzusetzen.

- **Indicators**

- **Charge Indicator:** Leuchtet rot, wenn das Shield über den USB-C-Port aufgeladen wird.
- **Power Indicator:** Leuchtet grün, wenn der Netzschalter in der Position „ON“ ist.
- **Battery Indicator:** Zwei orangefarbene Anzeigen repräsentieren verschiedene Batteriestände. Sie blinken während des Ladens und gehen aus, wenn die Batterie aufgeladen werden muss.

- **Camera Servo**

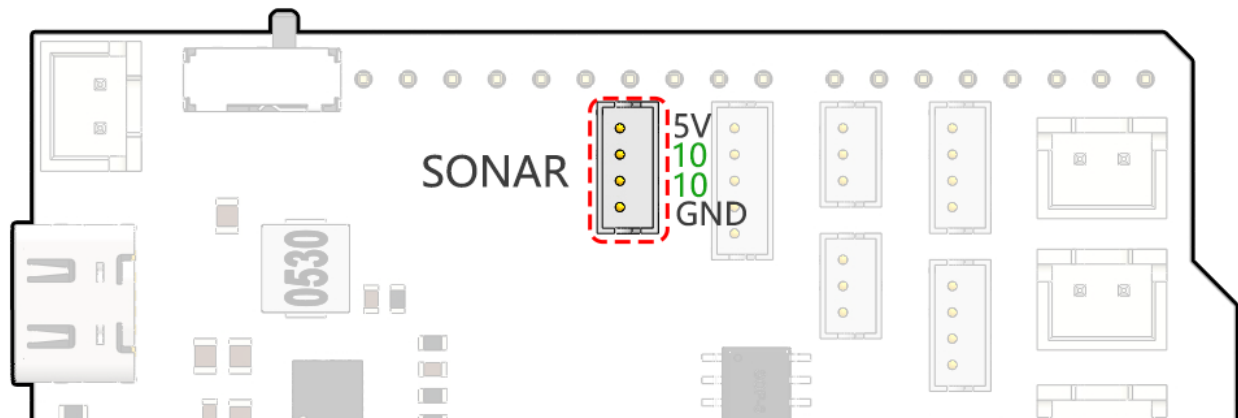
- Das Servo an der Kamera ist hier angeschlossen.
- Das braune Kabel wird mit „-“ verbunden, das rote Kabel wird mit „+“ verbunden und das gelbe Kabel wird mit Pin 6 verbunden.

- **Netzschalter**

- Auf ON schieben, um das GalaxyRVR einzuschalten.
- **Motor Port**
  - **Motor Port(Right)**: 3 Motoren können angeschlossen werden, aber alle 3 Motoren werden von demselben Satz Signalen **Pins 2 und 3** gesteuert.
  - **Motor Port(Left)**: 3 Motoren können angeschlossen werden, aber alle 3 Motoren werden von demselben Satz Signalen **Pins 4 und 5** gesteuert.
  - Anschlusstyp: XH2.54, 2P.
- **RGB Strip**
  - Zum Anschluss von 2 RGB-LED-Streifen, die drei Pins des Streifens sind an **12, 13 und 11** angeschlossen.
  - Anschlusstyp: ZH1.5, 4P.
- **LEFT/RIGHT IR**
  - Zum Anschluss von zwei IR-Hindernisvermeidungsmodulen.
  - Das **left obstacle avoidance module** wird an **Pin 8** angeschlossen, das **right obstacle avoidance module** an **Pin 7**.
  - Anschlusstyp: ZH1.5, 3P.
- **CAMERA**
  - Der Kamera-Adapter-Board-Anschluss.
  - Anschlusstyp: ZH1.5, 5P.
- **SONAR**
  - Zum Anschluss des Ultraschallmoduls, sowohl Trig- als auch Echo-Pins sind an **Pin 10** des Arduino-Boards angeschlossen.
  - Anschlusstyp: ZH1.5, 4P.
- **Mode Switch**
  - Der ESP32-CAM und das Arduino-Board teilen sich dieselben RX (Empfang) und TX (Sendung) Pins.
  - Daher müssen Sie beim Hochladen von Code diesen Schalter auf die **right side** umlegen, um den ESP32-CAM zu trennen und so Konflikte oder potenzielle Probleme zu vermeiden.
  - Wenn Sie die Kamera verwenden möchten, stellen Sie diesen Schalter auf die **left side**, sodass der ESP32-CAM mit dem Arduino-Board kommunizieren kann.
- **SOLAR**
  - Dies ist der Anschluss für das Solarpanel, das den Akku aufladen kann, wenn es an das Solarpanel angeschlossen ist.
  - Anschlusstyp: XH2.54, 2P.

### 4.2.1 SONAR

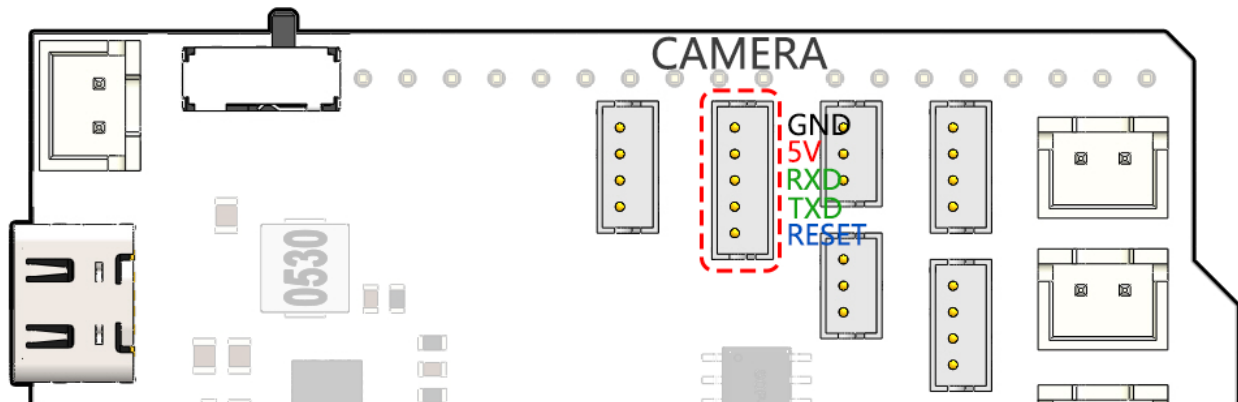
Hier ist die Pinbelegung für den ZH1.5-4P Ultraschallanschluss, wobei die Trig- & Echo-Pins mit Pin 10 des Arduino-Boards verbunden sind.



### 4.2.2 CAMERA

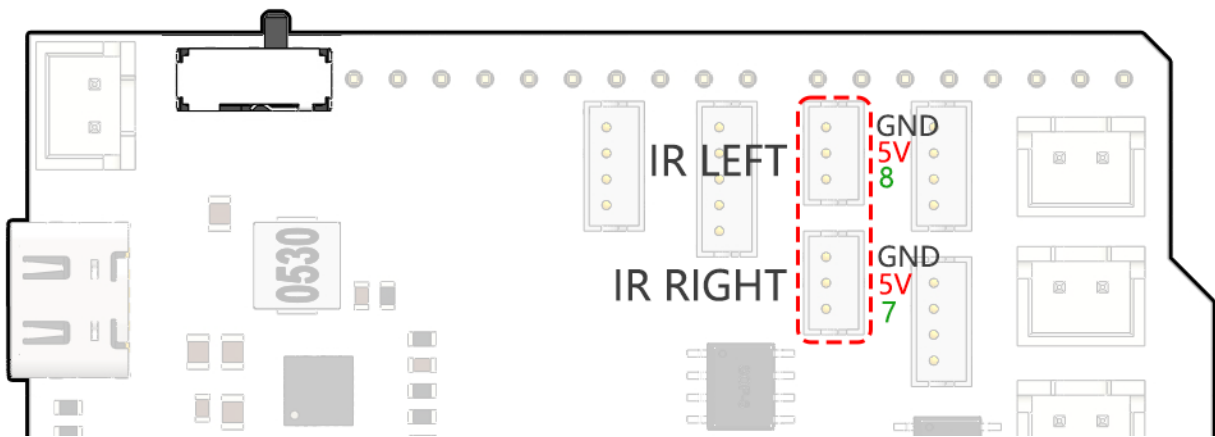
Das Pin-Diagramm des Kameraadapter-Interfaces wird hier gezeigt, der Typ ist ZH1.5-7P.

- TX und RX werden für ESP32 CAM verwendet.



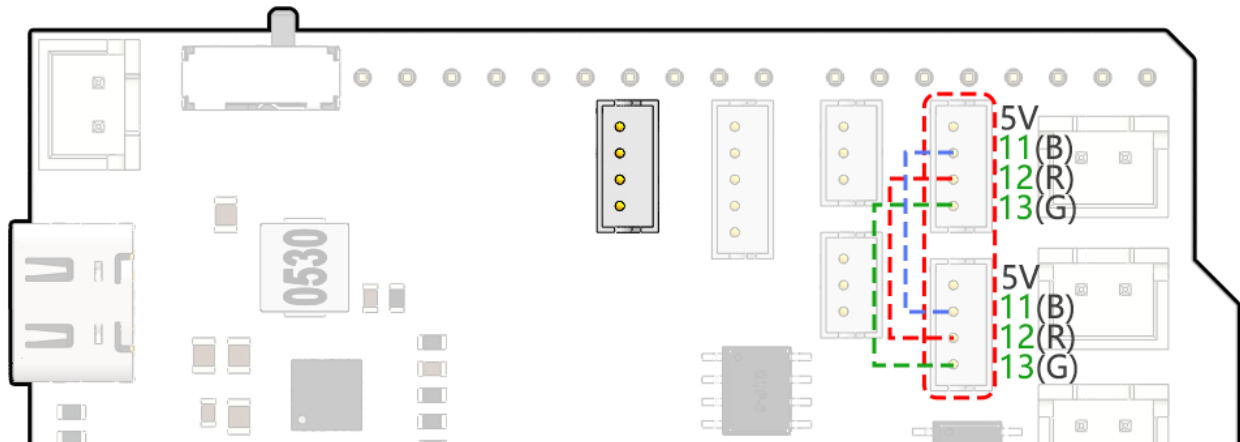
### 4.2.3 LEFT/RIGHT IR

Hier ist die Pinbelegung des Hindernisvermeidungsanschlusses.



### 4.2.4 RGB Strip

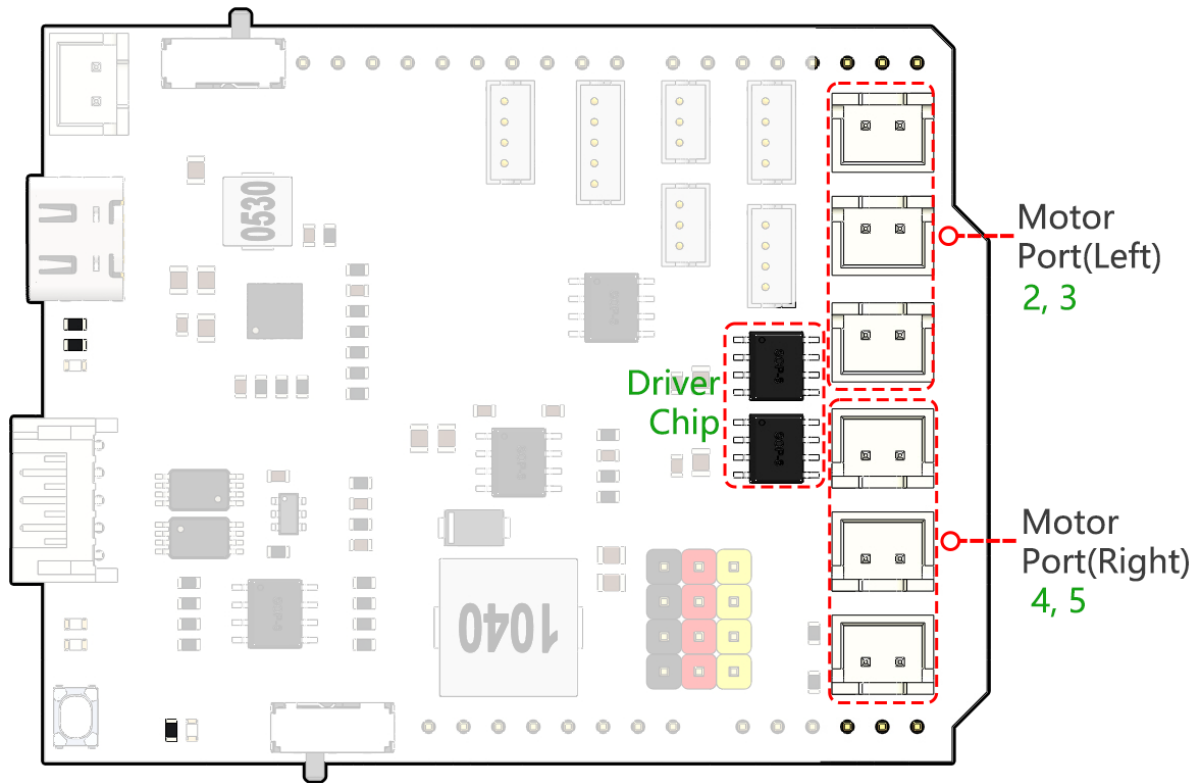
Unten ist das Pin-Diagramm der zwei parallel angeschlossenen RGB LED-Streifen, deren Pinbelegungen gleich sind.



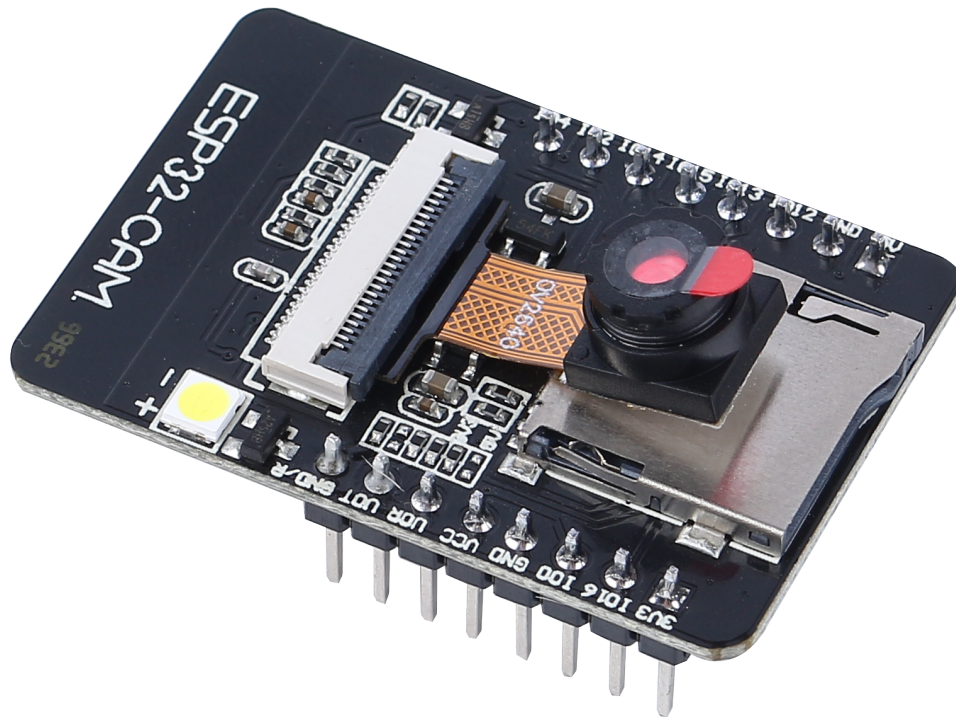


### 4.2.5 Motor Port

Hier ist die Pinbelegung der 2 Sätze von Motoranschlüssen.



### 4.3 ESP32 CAM



Das ESP32-CAM ist ein sehr kleines Kameramodul mit dem ESP32-S Chip, das ungefähr 10 Dollar 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 aufzunehmen oder Dateien für Kunden bereitzustellen.

Das Modul kann als das kleinste System unabhängig arbeiten, mit einer Größe von nur 27\*40.5\*4.5mm und einem Tiefschlafstrom von nur 6mA.

Das ESP32-CAM kann in verschiedenen IoT-Anwendungen weit verbreitet eingesetzt werden, geeignet für Heim-Smart-Geräte, industrielle drahtlose Steuerung, drahtlose Überwachung, QR-drahtlose Identifikation, drahtlose Positionierungssignale und andere IoT-Anwendungen. Es ist eine ideale Lösung für IoT-Anwendungen.

#### Technische Spezifikationen

Modell	ESP32-CAM
Gehäuse	DIP-16
Größe	27*40.5*4.5±0.2mm
SPI Flash	Standard 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	UARTSPII2CPWM
Unterstützt TF-Karte	bis zu 4G
IO Pins	9
Serielle Portgeschwindigkeit	Standard 115200 bps
Bildausgabeformat	JPEG(nur OV2640 unterstützt), BMP, GRAUSTUFE
Spektrumbereich	2400 ~2483.5MHz

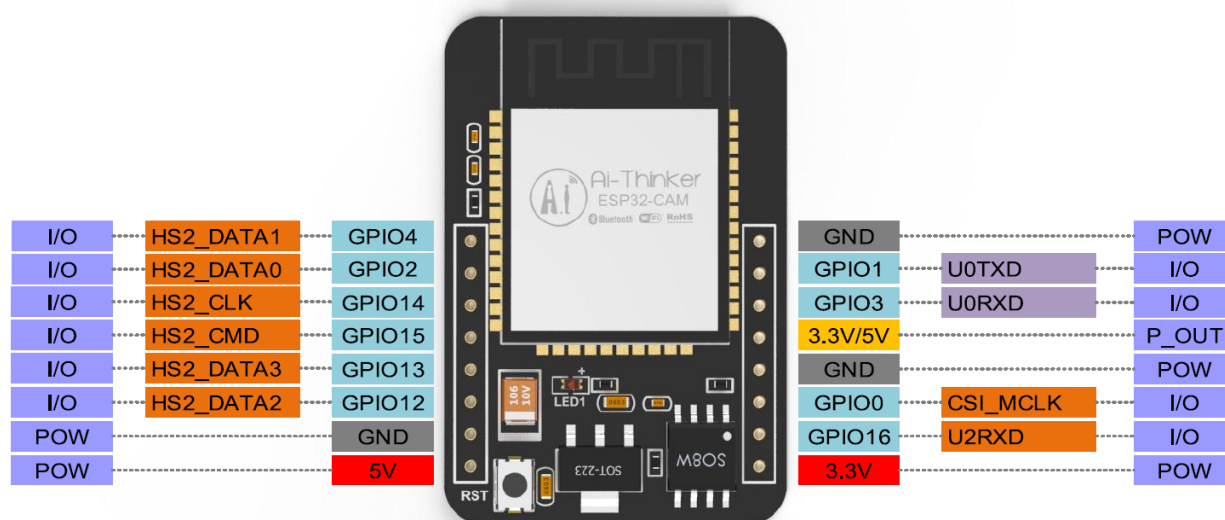
Fortsetzung auf der nächsten Seite

Tab. 1 – Fortsetzung der vorherigen Seite

Antennentyp	On-Board PCB-Antenne, Gewinn 2dBi
Sendeleistung	802.11b: $17\pm 2$ dBm (@11Mbps)
	802.11g: $14\pm 2$ dBm (@54Mbps)
	802.11n: $13\pm 2$ dBm (@MCS7)
Empfindlichkeit	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 maximal: 310mA@5V
	Tiefschlaf: der niedrigste Stromverbrauch kann 6mA@5V erreichen
	Modem-Schlaf: Minimum 20mA@5V
	Licht-Schlaf: Minimum 6.7mA@5V
Sicherheit	WPA/WPA2/WPA2-Enterprise/WPS
Stromversorgungsbereich	4.75-5.25V
Betriebstemperatur	-20 °C ~ 70 °C
Lagerumgebung	-40 °C ~ 125 °C , < 90%RH

### ESP32-CAM Pinbelegung

Die folgende Abbildung 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. Sie benötigen diese Pins, um Code auf Ihr Board hochzuladen.
- Zusätzlich spielt auch **GPIO 0** eine wichtige Rolle, da es bestimmt, ob der ESP32 im Flash-Modus ist oder nicht. Wenn **GPIO 0** mit **GND** verbunden ist, ist der 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 verbunden mit der an Bord befindlichen LED)

- GPIO 12: Daten 2
- GPIO 13: Daten 3

#### Hinweis

- Stellen Sie sicher, dass die Eingangsleistung des Moduls mindestens 5V 2A beträgt, da sonst das Bild Wasserlinien haben kann.
- Der ESP32 GPIO32-Pin steuert die Kameralistung. Wenn die Kamera arbeitet, ziehen Sie bitte GPIO32 nach unten.
- Da GPIO0 mit dem Kamera-XCLK verbunden ist, lassen Sie GPIO0 in der Luft, wenn Sie es verwenden, und verbinden Sie es nicht mit hohem oder niedrigem Pegel.
- Die Standard-Firmware ist bereits in der Fabrik enthalten und es wird kein zusätzlicher Download bereitgestellt. Bitte seien Sie vorsichtig, wenn Sie andere Firmware neu brennen müssen.

#### Dokumentation

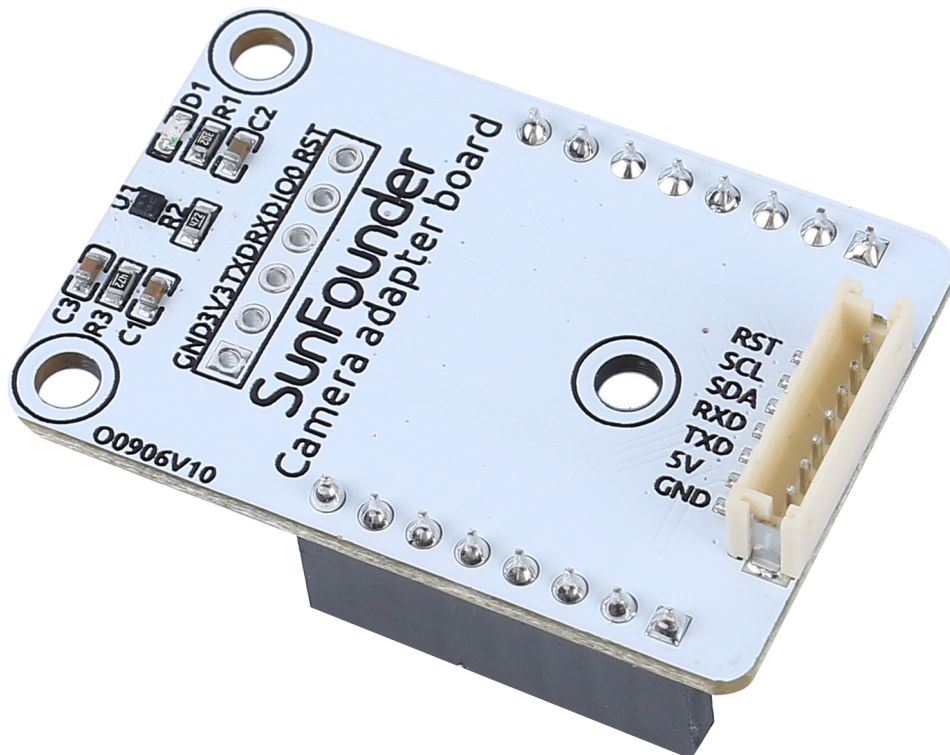
- Schaltplan:
- Kameraspezifikation (englische Version):

---

**Bemerkung:** Alle oben genannten Informationen stammen von

---

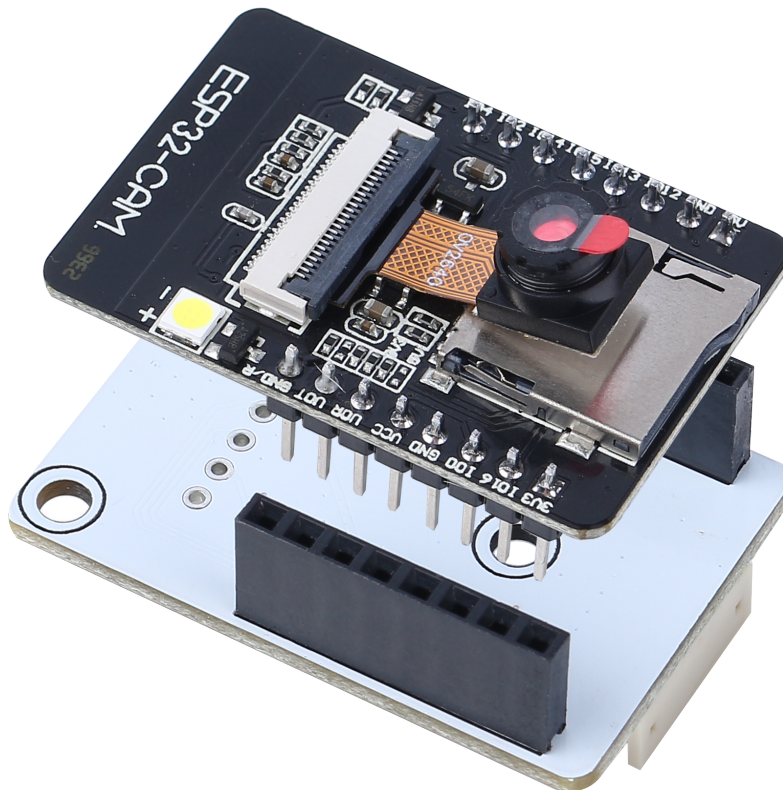
## 4.4 Kamera-Adapterplatine



- **RST:** Wird verwendet, um die ESP32-CAM zurückzusetzen.

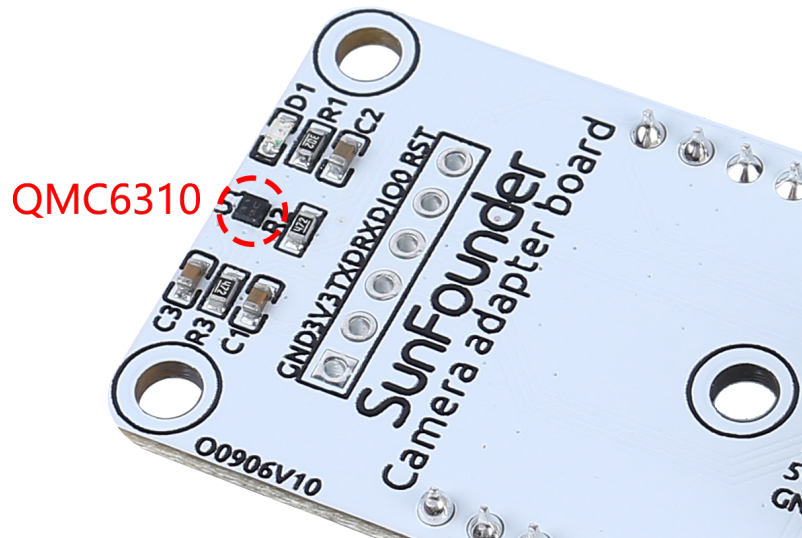
- **SCL:** Serieller Datenpin für QMC6310
- **SDA:** Serieller Uhrenpin des QMC6310
- **RXD:** RXD der ESP32-CAM, Sie benötigen diese beiden seriellen Pins, RXD und TXD, um Code auf die ESP32-CAM hochzuladen.
- **TXD:** TXD der ESP32-CAM
- **5V:** 5V DC-Eingang
- **GND:** Masse-Eingang

Die Kamera-Adapterplatine ist, wie der Name schon sagt, eine Erweiterungsplatine für die ESP-32-CAM, die verwendet wird, um die ESP32-CAM zu erweitern, sodass sie am Roboter befestigt und leicht verkabelt werden kann.



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





#### Merkmale

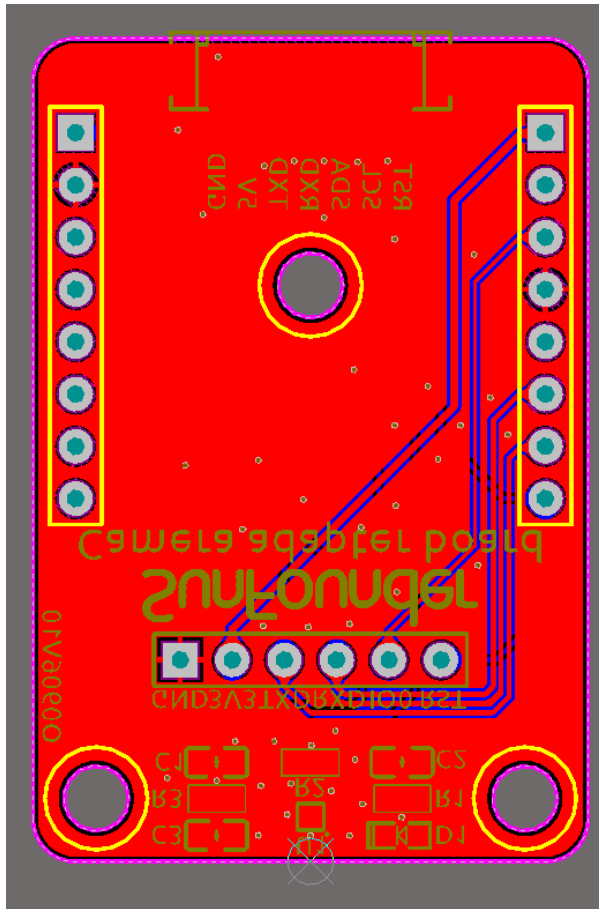
- Betriebsspannung: 5V
- Schnittstellenmodell: ZH1.5, 7P
- Abmessungen: 40mm x 27mm x 15mm
- Kommunikationsprotokoll: UART und I2C

#### Dokumente

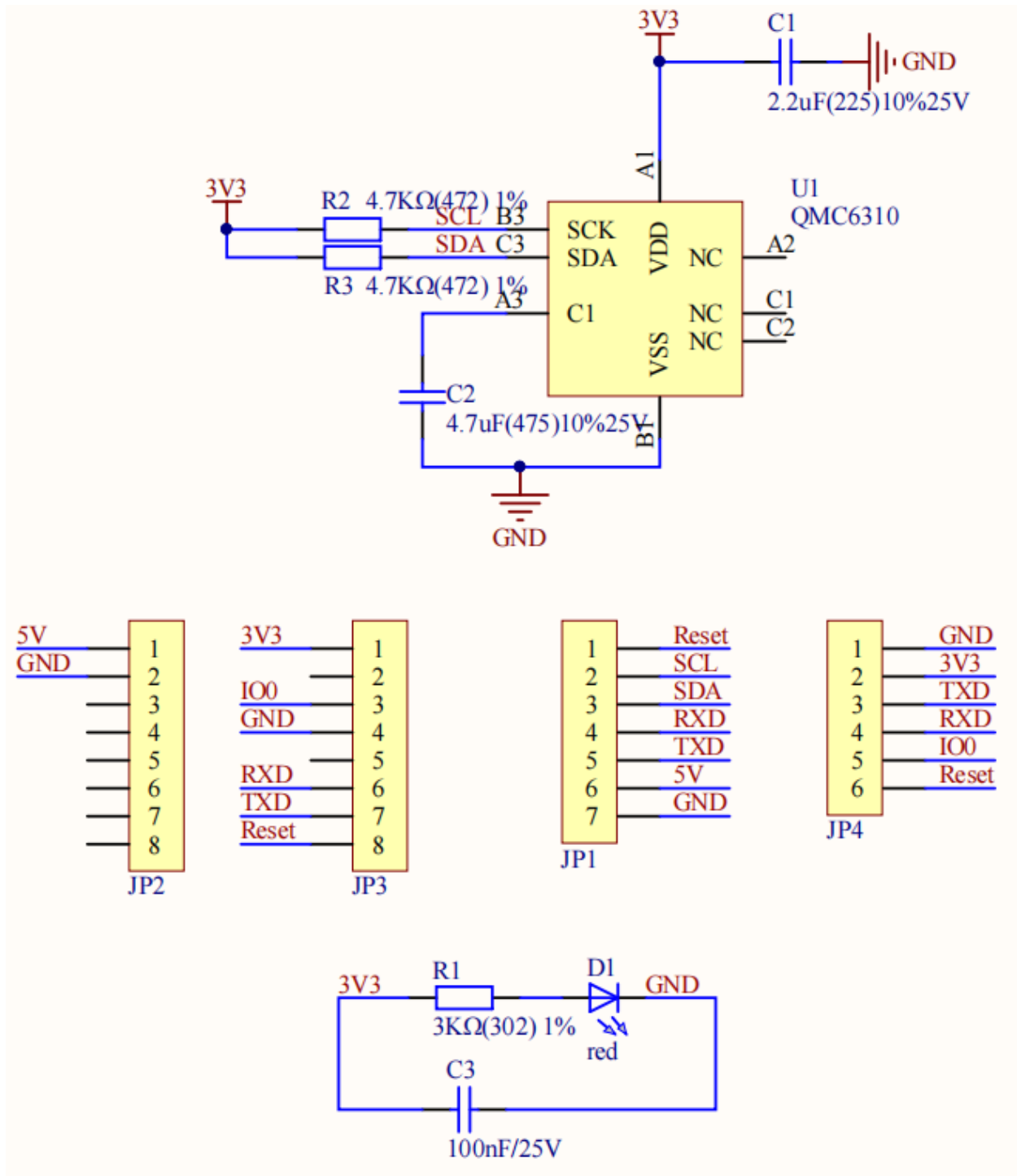
- PCB







- Schaltplan



### Über QMC6310

Der QMC6310 ist ein dreiaxialer Magnetfeldsensor, der magnetische Sensoren und Signalzustands-ASIC in einem Siliziumchip integriert. Dieses Land Grid Array-Paket (LGA) ist für Anwendungen wie E-Kompass, Kartenrotation, Gaming und persönliche Navigation in mobilen und tragbaren Geräten konzipiert.

Der QMC6310 basiert auf modernster, hochauflösender, magnetoresistiver Technologie. Zusammen mit dem eigens entwickelten 16-Bit-ADC-ASIC bietet er die Vorteile von geringem Rauschen, hoher Genauigkeit, geringem Stromverbrauch, Offset-Kompensation und Temperaturkompensationen. QMC6310 ermöglicht eine Kompassgenauigkeit von 1°

bis 2°. Der I<sup>2</sup>C-Serienbus ermöglicht eine einfache Schnittstelle.

Der QMC6310 ist in einem 1,2x1,2x0,53mm<sup>3</sup> großen, oberflächenmontierbaren 8-Pin-LGA-Gehäuse untergebracht.

•

## 4.5 Ultraschallmodul



- **TRIG:** Trigger-Pulseingang
- **ECHO:** Echo-Pulseausgang
- **GND:** Erdung
- **VCC:** 5V Versorgung

Dies ist der HC-SR04 Ultraschall-Abstandssensor, der berührungslose Messungen von 2 cm bis 400 cm mit einer Bereichsgenauigkeit von bis zu 3 mm bietet. Das Modul enthält einen Ultraschall-Sender, einen Empfänger und eine Steuerschaltung.

Sie müssen nur 4 Pins anschließen: VCC (Stromversorgung), Trig (Auslöser), Echo (Empfang) und GND (Erdung), um es leicht für Ihre Messprojekte zu verwenden.

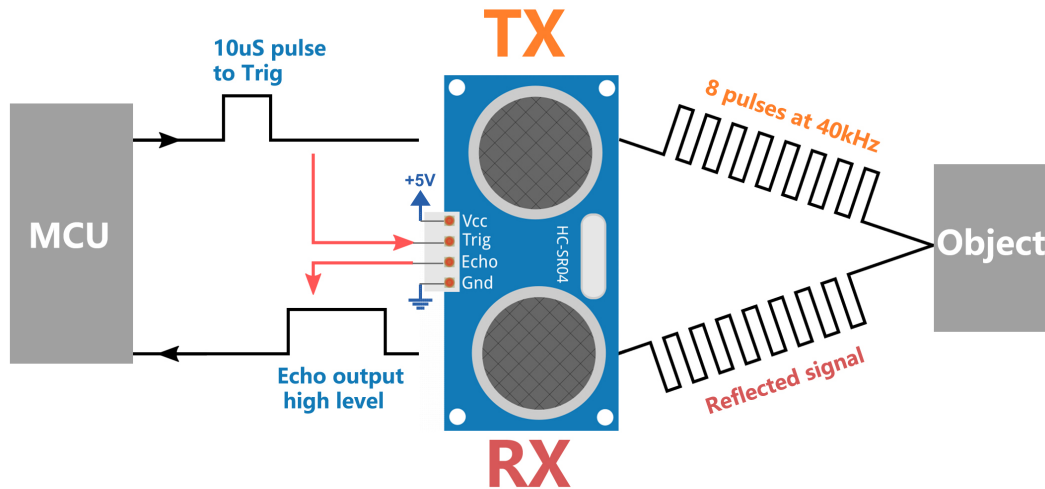
### Merkmale

- Betriebsspannung: DC5V
- Betriebsstrom: 16mA
- Arbeitsfrequenz: 40Hz
- Maximaler Bereich: 500cm
- Minimaler Bereich: 2cm
- Trigger-Eingangssignal: 10uS TTL-Impuls
- Echo-Ausgangssignal: Eingang TTL-Pegelsignal und der Bereich im Verhältnis
- Steckverbinder: XH2.54-4P
- Abmessung: 46x20.5x15 mm

## Prinzip

Die grundlegenden Prinzipien sind wie folgt:

- Verwendung von IO-Trigger für mindestens 10µs hohes Signalniveau.
- Das Modul sendet einen 8-Zyklus-Burst von Ultraschall mit 40 kHz und erkennt, ob ein Pulssignal empfangen wird.
- Echo gibt ein hohes Niveau aus, wenn ein Signal zurückkommt; die Dauer des hohen Niveaus ist die Zeit von der Emission bis zur Rückkehr.
- Entfernung = (Zeit des hohen Niveaus x Schallgeschwindigkeit (340M/S)) / 2

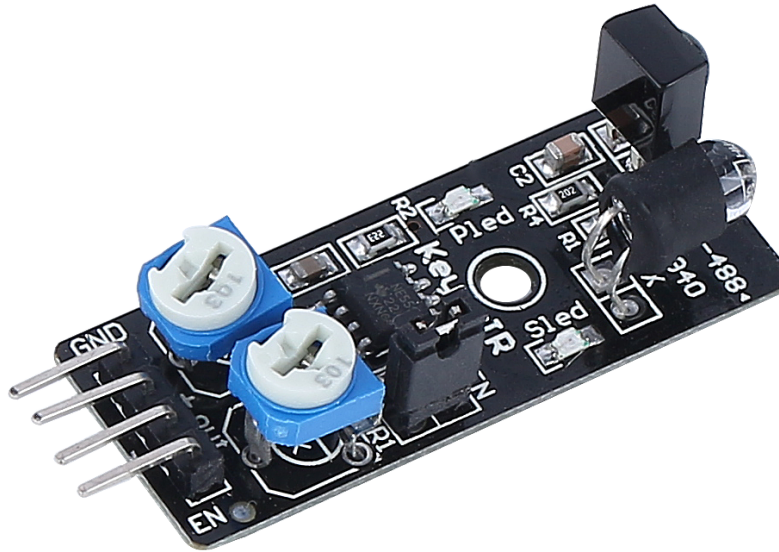


## Anwendungshinweise

- Dieses Modul sollte nicht unter Strom angeschlossen werden, falls nötig, sollte zuerst der GND des Moduls angeschlossen werden. Andernfalls beeinflusst es die Arbeit des Moduls.
- Die Fläche des zu messenden Objekts sollte mindestens 0,5 Quadratmeter betragen und möglichst flach sein. Andernfalls beeinflusst es die Ergebnisse.

## 4.6 IR-Hindernisvermeidungsmodul

Dies ist ein Infrarot-Hindernisvermeidungsmodul, das die Anwesenheit von Objekten vor sich erkennen kann. Es wird häufig in Robotern, Automatisierungssystemen und anderen intelligenten Geräten verwendet. Sein Erfassungsbereich liegt zwischen 2 cm und 40 cm, und Objekte unterschiedlicher Farbe haben unterschiedliche Reflexionsniveaus. Daher ist der Erfassungsbereich umso kürzer, je dunkler das Objekt ist.



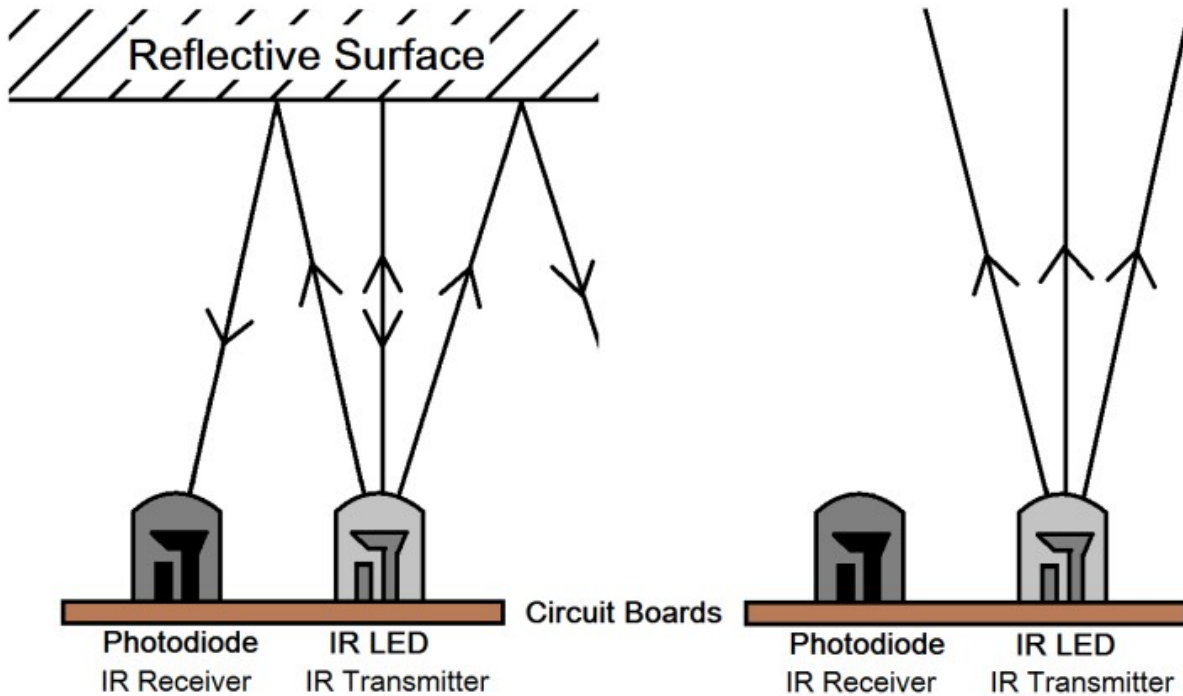
Hier sind die Pin-Definitionen:

- **GND**: Erdung
- **+**: Stromversorgung, 3,3 ~ 5V DC.
- **Out**: Standardmäßig bleibt es hoch und wird nur niedrig, wenn es ein Hindernis erkennt.
- **EN**: Dieser **Enable**-Pin entscheidet, wann das Modul arbeiten soll. Standardmäßig ist er mit GND verbunden, was bedeutet, dass das Modul immer in Betrieb ist.

#### Wie funktioniert das?

Dieses Modul enthält ein Paar IR-Sendungs- und Empfangskomponenten. Der Sender emittiert Infrarotlicht, das, wenn es auf ein Hindernis trifft, zurückreflektiert und vom Empfänger aufgenommen wird. Bei Erkennung leuchtet die Anzeige auf. Nach der Schaltungsverarbeitung gibt es ein niedriges Signal aus.

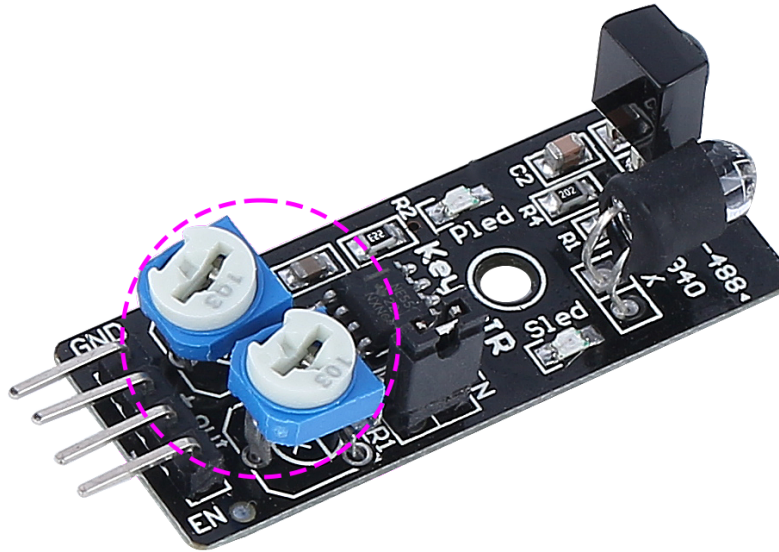




Der niedrige Zustand des **EN**-Pins aktiviert das Modul, wobei die Jumperkappe den **EN**-Pin mit dem GND sichert. Um den EN-Pin programmgesteuert zu steuern, entfernen Sie die Jumperkappe.



Auf dem Modul befinden sich zwei Potentiometer, eines zur Einstellung der Sendeleistung und eines zur Einstellung der Sendefrequenz. Durch die Anpassung dieser beiden Potentiometer können Sie seine effektive Entfernung einstellen.



## Einstellen der Erfassungsdistanz

Der Erfassungsbereich des Hindernisvermeidungsmoduls muss präzise kalibriert werden, um eine optimale Leistung zu erzielen, da die Standardeinstellungen der Fabrik möglicherweise nicht den spezifischen Anforderungen entsprechen.

Die Kalibrierung umfasst die folgenden Schritte:

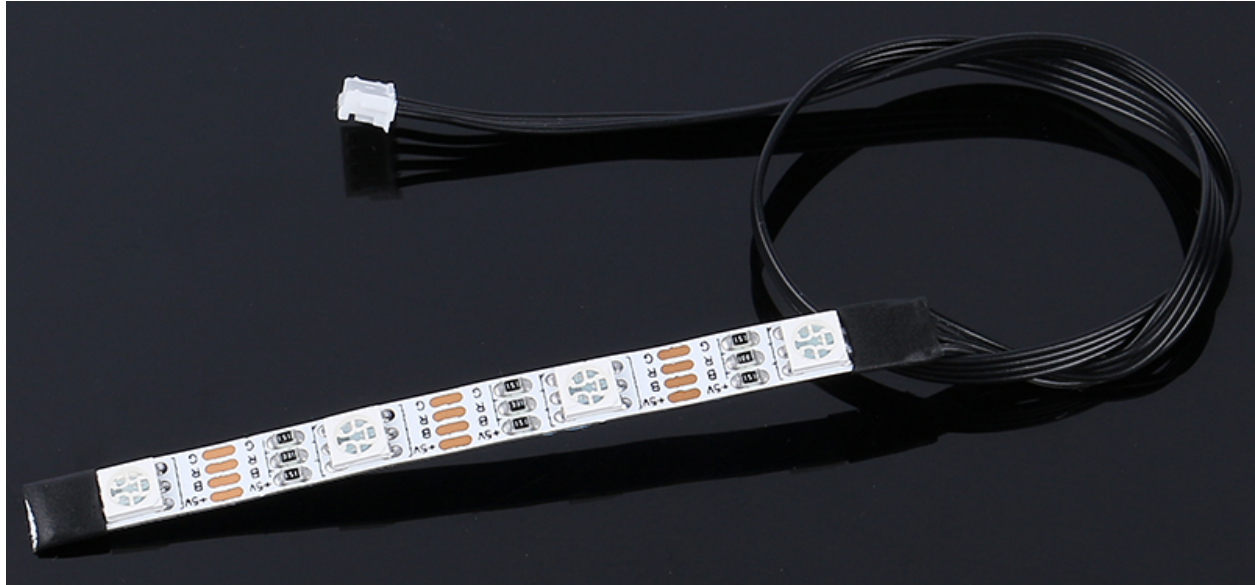
1. Ausrichtung des Moduls: Beginnen Sie mit dem rechten Hindernisvermeidungsmodul. Transport kann gelegentlich die Ausrichtung des Infrarotsenders und -empfängers des Moduls verschieben. Sie sollten manuell neu ausgerichtet werden, um Genauigkeit zu gewährleisten.
2. Platzieren Sie ein Hindernis etwa 20 cm direkt vor dem rechten Modul. Die Box, in der unser Rover-Kit geliefert wurde, ist eine gute Wahl dafür! Drehen Sie nun das Potentiometer am Modul, bis die Anzeige am Modul gerade aufleuchtet. Bewegen Sie dann das Hindernis hin und her, um zu prüfen, ob die Anzeige im gewünschten Abstand aufleuchtet. Wenn das Licht nicht in der richtigen Entfernung aufleuchtet oder ständig eingeschaltet bleibt, müssen Sie das andere Potentiometer einstellen.
3. Wiederholen Sie den gleichen Prozess für ein weiteres Modul.

## Merkmale

- Betriebsspannung: 3,3 V bis 5 V
- Ausgang: digital (ein/aus)
- Einstellungsschwelle: einstellbar durch 2 Potentiometer
- Distanzbereich: 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

## 4.7 4 RGB-LED-Streifen



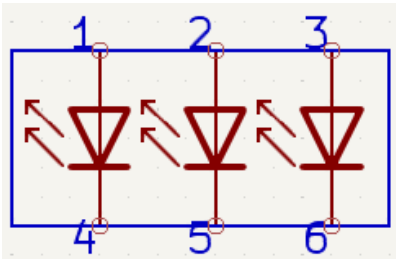
- **+5V**: Gemeinsame Anode der drei LEDs, die an DC 5V angeschlossen werden muss
- **B**: Kathode der blauen LED
- **R**: Kathode der roten LED
- **G**: Kathode der grünen LED

Dieser RGB-LED-Streifen verfügt über vier R5050 RGB-LEDs, die in der Lage sind, jeden Farbton durch Kombination der drei Grundfarben Rot, Blau und Grün zu erzeugen.

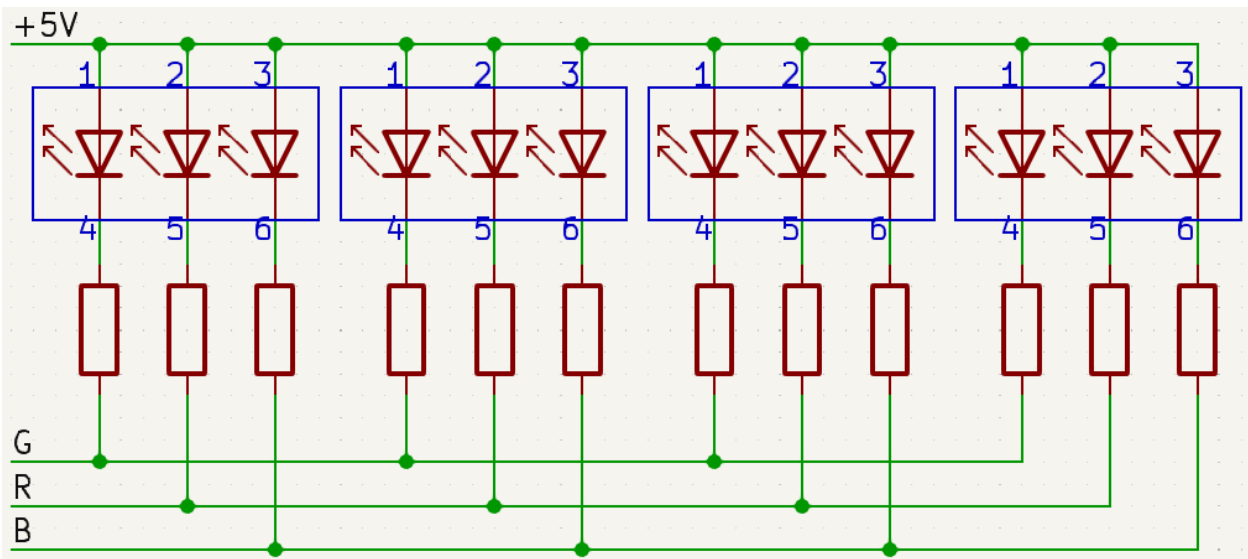
Die R5050 RGB-LEDs sind mit einer gemeinsamen Anodenkonfiguration entworfen. Jede LED auf dem Streifen funktioniert als unabhängiger Schaltkreis, sodass Sie den Streifen an gekennzeichneten Stellen schneiden können, ohne andere Abschnitte zu beeinflussen. Die Flexibilität und Anpassungsfähigkeit des Streifens werden durch seine Konstruktion auf einer FPC-Platine verstärkt, die mit doppelseitigem Klebstoff für eine einfache Installation versehen ist.

### Was ist eine R5050 RGB-LED?

Die R5050 RGB-LED ist eine Art LED, die rote, blaue und grüne lichtemittierende Dioden in einem einzigen Gehäuse kombiniert. Jede LED in diesem Paket hat ihren eigenen Pin, was eine individuelle Steuerung ermöglicht. Diese Konfiguration ermöglicht die Erzeugung einer breiten Palette von Farben, indem die Intensität jeder LED variiert wird.



In einer typischen Anwendung werden mehrere R5050 RGB-LEDs auf einem flexiblen Schaltkreis in einer intelligenten Konfiguration angeordnet. Dies geschieht, indem die „positiven“ Enden (Anoden) aller LEDs miteinander verbunden werden, während die „negativen“ Enden (Kathoden) an ihre jeweiligen Farbleitungen angeschlossen werden. Das bedeutet, dass alle grünen Kathoden miteinander verbunden sind, alle roten Kathoden miteinander verbunden sind und alle blauen Kathoden miteinander verbunden sind. Diese Anordnung ermöglicht eine effiziente Steuerung der Farbmischung und Lichtintensität, was diese LEDs in Anwendungen beliebt macht, in denen eine anpassbare Farbbeleuchtung gewünscht ist, wie z. B. in dekorativer Beleuchtung, Beschilderung und Displaytechnologien.



#### Merkmale

- Arbeitsspannung: DC5V
- Farbe: Vollfarbe RGB
- Arbeitstemperatur: -15-50

- RGB-Typ: 5050RGB
- Strom: 150mA (einzelschaltkreis)
- Leistung: 1.5W
- Dicke des Lichtstreifens: 2mm
- Breite des Lichtstreifens: 5.5mm
- Kabel: ZH1.5-4P, 25cm, 28AWG, Schwarz

## 4.8 Servo

Ein Servo ist ein spezialisierter Motor, der für seine Präzision bei der Steuerung bestimmter Winkelpositionen bekannt ist.



- **Braune Leitung:** GND (Erde)
- **Orangefarbene Leitung:** Signalleitung, an den PWM-Pin der Hauptplatine anschließen.
- **Rotes Kabel:** VCC (Spannungsversorgung)

Im Gegensatz zu herkömmlichen Motoren, die sich kontinuierlich drehen, kann ein Servo zu einer präzisen Position bewegt werden und diese genau halten. Dies wird durch eine Kombination aus Zahnrädern, einem Potentiometer und einer Steuerschaltung erreicht. Servos werden häufig in verschiedenen Anwendungen eingesetzt, bei denen eine präzise Steuerung der Position von Objekten oder Mechanismen erforderlich ist.

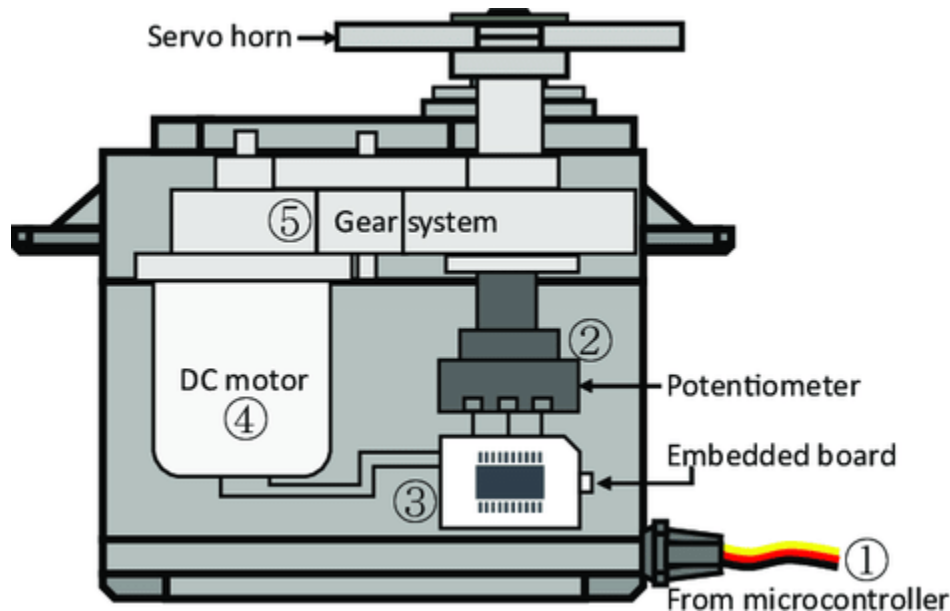
### Merkmale

- **Motortyp:** Kernmotor
- **Betriebsspannung:** 4,8~6V DC
- **Bereitschaftsstrom:** 4 mA
- **Stromverbrauch (bei 4,8V Leerlauf):** 50mA
- **Stromverbrauch (bei 6 V Leerlauf):** 60mA
- **Blockierstrom (bei gesperrtem 4,8V):** 550mA
- **Blockierstrom (bei gesperrtem 6V):** 650mA
- **Nenndrehmoment:** 4,8V, 0,6 kgf·cm; 6V, 0,7 kgf·cm

- **Max. Drehmoment:** 4,8V, 1,4 kgf.cm; 6V, 1,6 kgf.cm
- **Leerlaufdrehzahl:** 4,8V, 0,14s/60°; 6V, 0,12s/60°
- **Hinweis:** Torsionsschutz: 0,9 kgf.cm; Stromausfallabschaltung nach 5 Sekunden Dauerbetrieb
- **Betriebstemperaturbereich:** -10°C~+50°C
- **Lagerungstemperaturbereich:** -20°C~+60°C
- **Betriebsfeuchtigkeitsbereich:** 90%RH
- **Lagerungsfeuchtigkeitsbereich:** 90%RH
- **Gewicht:** 10± 0,5g
- **Material:** ABS
- **Betriebswinkel:** 180°±10°(500~2500us)
- **Mechanischer Begrenzungswinkel:** 360°
- **Linker & Rechter Drehwinkel:** 6°
- **Mittellagenabweichung:** 1°
- **Spiel:** 4 us
- **Verstärkertyp:** Digital
- **Kabelmaterial:** 1.08,19 PVC
- **Kabellänge:** 245±5mm (ohne Stecker)
- **Steckertyp:** JR2.54mm/3Pin

### Funktionsprinzip

Im Inneren eines Servos tragen wesentliche Komponenten zu seiner einzigartigen Funktionalität bei. Im Kern eines Servos befindet sich ein herkömmlicher Motor, der dem Motor ähnelt, der die Räder unseres Marsrovers antreibt. Um den Motor herum befindet sich ein großes Zahnrad, das wiederum mit einem kleineren Zahnrad auf der Motorwelle in Verbindung steht. Diese Getriebeanordnung verwandelt die schnelle kreisförmige Bewegung des Motors in langsamere, aber kraftvollere Bewegungen.





Doch die eigentliche Magie geschieht im Inneren des Servos, dank eines winzigen elektronischen Wunders namens „Potentiometer“ und einer anspruchsvollen „Steuerungsschaltung“. So funktioniert es: Wenn der Servo sich bewegt, dreht sich das Potentiometer und ändert seinen elektrischen Widerstand. Die Steuerungsschaltung erkennt und interpretiert diese Änderung des Widerstands mit bemerkenswerter Präzision und bestimmt so die genaue Position des Servos. Dies ist ein Zeugnis seiner Raffinesse.

In der Welt der Servosteuerung kommt eine einzigartige Signalisierungsmethode namens „Pulsweitenmodulation“ oder PWM zum Einsatz. Durch geschickte Anpassung der Breite dieser Impulse können Bediener den Servo präzise steuern und in einer bestimmten Position halten. Dies ist die Essenz der Servomotortechnologie, ein Bereich, in dem Präzision und Kontrolle zusammenkommen, um eine Vielzahl von Anwendungen zu ermöglichen.

## 4.9 TT-Motor



Dies ist ein TT-DC-Getriebemotor mit einem Übersetzungsverhältnis von 1:120. Er wird mit zwei 250mm Kabeln mit einem XH2.54-2P-Stecker geliefert. Er kann mit 3VDC betrieben werden.

### Wie Motoren funktionieren

Ein Motor fungiert als Herz einer Maschine und wandelt elektrische Energie in mechanische Energie um. Diese Umwandlung bringt verschiedene Geräte zum Leben, von Kinderspielzeug und Haushaltsgeräten bis hin zu großen Fahrzeugen.

Hier ist der Prozess:

Wenn Strom in einen Motor fließt, erzeugt er ein Magnetfeld. Dieses Feld interagiert mit anderen Magneten im Motor und bringt ihn zum Drehen. Diese Drehbewegung, ähnlich wie ein Kreisel, kann dann die Bewegung von Rädern, Propellern oder anderen beweglichen Teilen in einer Maschine antreiben.

Der TT-Getriebemotor ist eine spezialisierte Art von Motor. Er kombiniert einen Standardmotor mit einer Reihe von Zahnrädern, die alle in einem robusten Kunststoffgehäuse untergebracht sind.

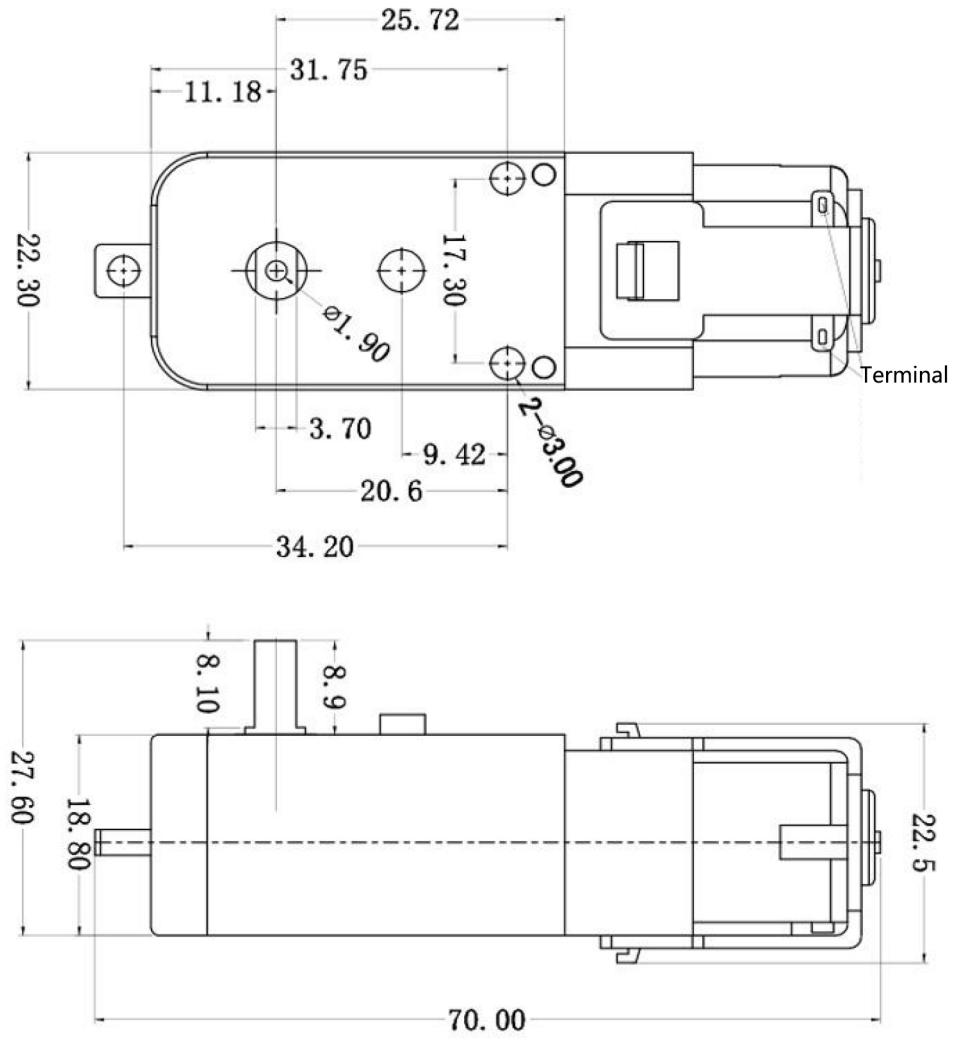
Wenn der Motor sich dreht, übertragen die Zahnräder diese Rotationsbewegung effektiv auf die Räder unseres Rovers. Die Integration von Zahnrädern ist entscheidend, da sie das Drehmoment verstärkt. Diese erhöhte Drehmomentkapazität ermöglicht es dem Motor, größere und schwerere Lasten zu bewegen, eine wesentliche Fähigkeit in verschiedenen Anwendungen.

### Merkmale

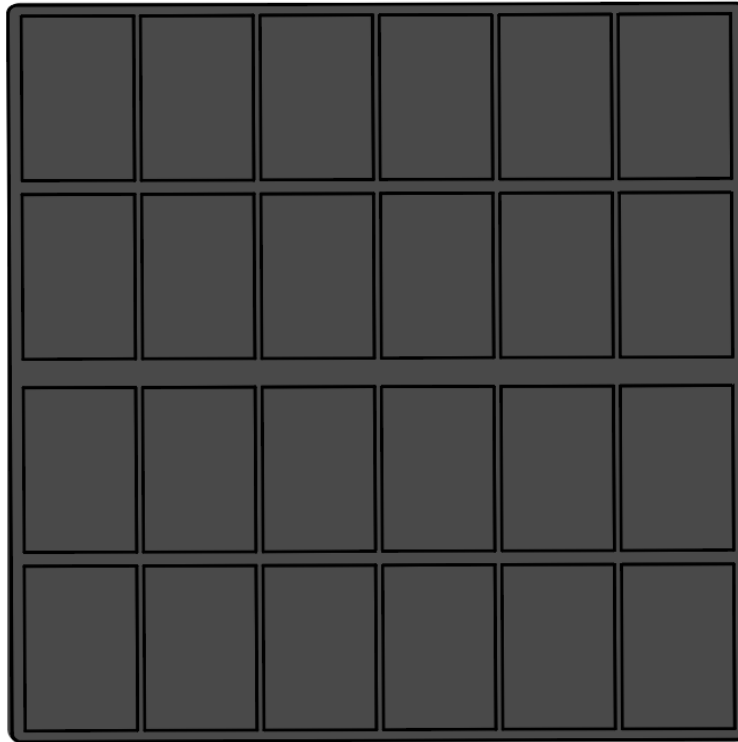
- Vorgeschlagene Spannung 3V~4.5V DC
- Anzahl der Wellen: Einzelwelle
- Übersetzungsverhältnis: 1:120
- Leerlaufstrom: 130mA
- Leerlaufgeschwindigkeit:  $38\text{rpm} \pm 8\%\text{rpm}$
- Startspannung: 2V (max.) im Leerlauf
- Ausgangsdrehmoment: 3V 1.2kgf.cm
- Nützliche Lebensdauer: 70-120H
- Drehrichtung: Beide Richtungen
- Körperabmessungen: 70 x 22,5 x 36,6mm
- Kabel: Grau und Schwarz, 24AWG, 250mm
- Stecker: Weiß, XH2.54-2P
- Gewicht: 28,5g

### Maßzeichnung

Einheit: mm



## 4.10 Solarpanel



Solarpanele sind Geräte, die Sonnenlicht in Elektrizität umwandeln. Sie bestehen aus photovoltaischen (PV) Zellen, die aus Halbleitermaterialien wie Silizium gefertigt sind. Wenn Sonnenlicht auf eine PV-Zelle trifft, werden Elektronen aus ihren Atomen gelöst. Diese Elektronen fließen durch die Zelle und erzeugen einen elektrischen Strom.

Solarpanele können verwendet werden, um Elektrizität für verschiedene Zwecke zu erzeugen, einschließlich der Stromversorgung von Häusern, Unternehmen und sogar ganzen Gemeinschaften. Sie sind eine saubere und erneuerbare Energiequelle, die dazu beitragen kann, unsere Abhängigkeit von fossilen Brennstoffen zu verringern.

### Merkmale

- Ausgangsleistung: 6V/660mA
- Zeit zum vollständigen Aufladen des Akkus: 7,2h (theoretischer Wert, unter der Annahme von starkem Sonnenlicht)
- Größe: 170mm x 170mm
- Kabel: Grau und Schwarz, 24AWG, 200mm
- Stecker: Weiß, XH2.54-2P

## 4.11 18650 Batterie



- **VCC:** Positiver Batterieanschluss, hier gibt es zwei Sätze von VCC und GND, um den Strom zu erhöhen und den Widerstand zu verringern.
- **Middle:** Zur Ausgleicheung der Spannung zwischen den beiden Zellen und somit zum Schutz der Batterie.
- **GND:** Negativer Batterieanschluss.

Dies ist ein von SunFounder hergestelltes spezielles Batteriepaket, bestehend aus zwei 18650 Batterien mit einer Kapazität von 2000mAh. Der Anschluss ist XH2.54 3P, der nach dem Einstecken in das Shield direkt aufgeladen werden kann.

### Merkmale

- Batterieladung: 5V/2A
- Batterieausgang: 5V/5A
- Batteriekapazität: 3.7V 2000mAh x 2
- Batterielebensdauer: 90min
- Ladezeit der Batterie: 130min
- Anschluss: XH2.54 3P





### 5.1 Q1: Kompilierungsfehler: SoftPWM.h: Datei oder Verzeichnis nicht gefunden?

Wenn die Meldung „Compilation error: SoftPWM.h: No such file or directory“ erscheint, bedeutet dies, dass die SoftPWM-Bibliothek nicht installiert ist.

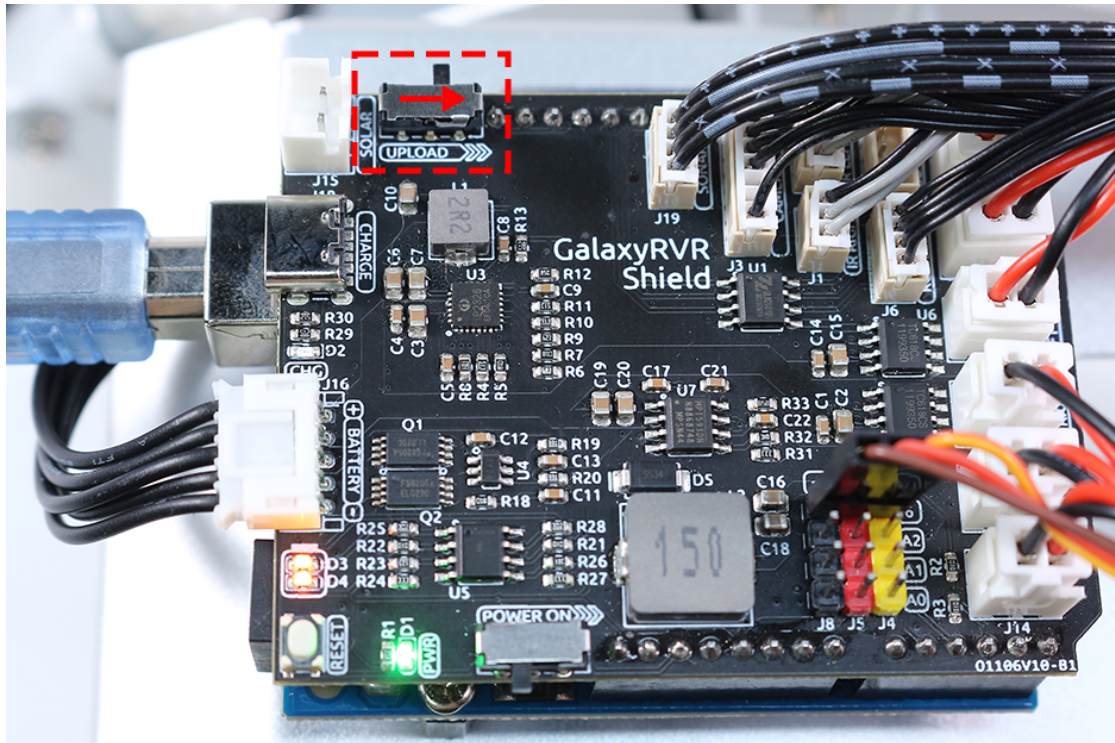
Bitte installieren Sie die beiden erforderlichen Bibliotheken SoftPWM und SunFounder AI Camera, wie gezeigt.

### 5.2 Q2: avrdude: stk500\_getsync() Versuch 10 von 10: nicht synchronisiert: resp=0x6e?

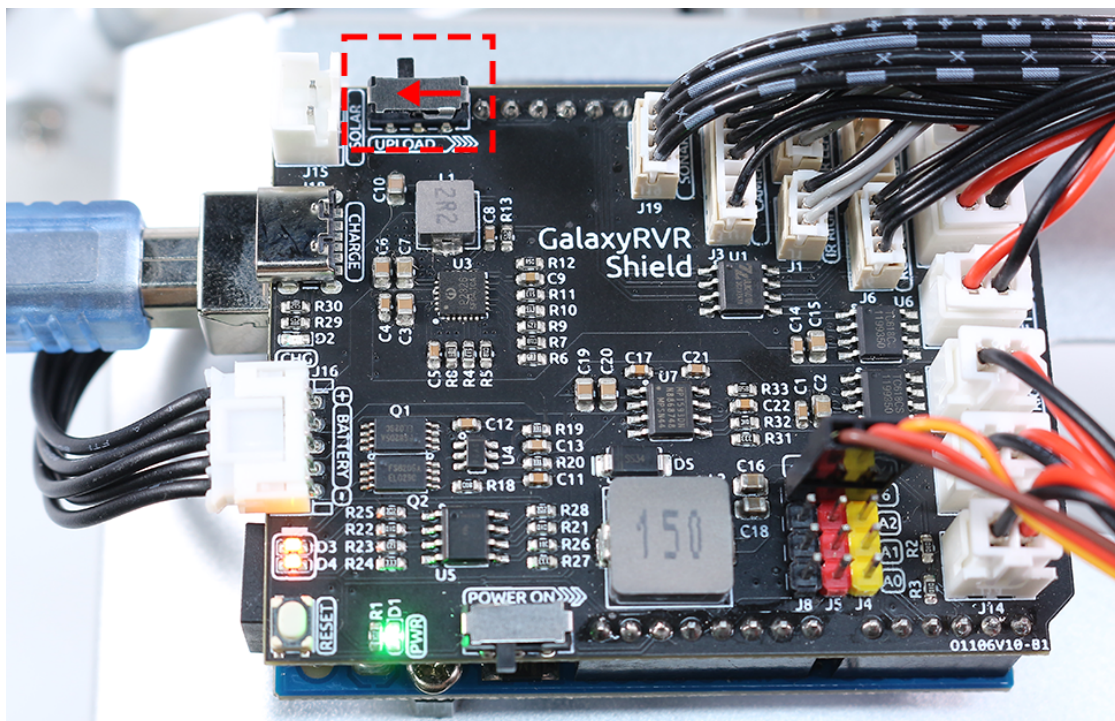
Wenn die folgende Nachricht weiterhin erscheint, nachdem Sie auf die Schaltfläche **Upload** geklickt haben und das Board sowie der Port korrekt ausgewählt wurden.

```
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 1 of 10: not in sync: resp=0x00
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 2 of 10: not in sync: resp=0x00
avrdude: stk500_recv(): programmer is not responding
avrdude: stk500_getsync() attempt 3 of 10: not in sync: resp=0x00
At this point, you need to make sure that the ESP32 CAM is unplugged.
```

Das ESP32-CAM und das Arduino-Board teilen sich die gleichen RX (Empfang) und TX (Sendung) Pins. Daher müssen Sie vor dem Hochladen des Codes zuerst das ESP32-CAM trennen, um Konflikte oder potenzielle Probleme zu vermeiden.



Nachdem der Code erfolgreich hochgeladen wurde, wenn Sie das ESP32 CAM nutzen möchten, müssen Sie den Schalter nach links bewegen, um das ESP32 CAM zu starten.



## 5.3 Q3: Wie kann ich den STT-Modus auf meinem Android-Gerät verwenden?

Der STT-Modus erfordert, dass das Android-Mobilgerät mit dem Internet verbunden ist und die Google-Servicekomponente installiert ist.

Folgen Sie nun den untenstehenden Schritten.

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

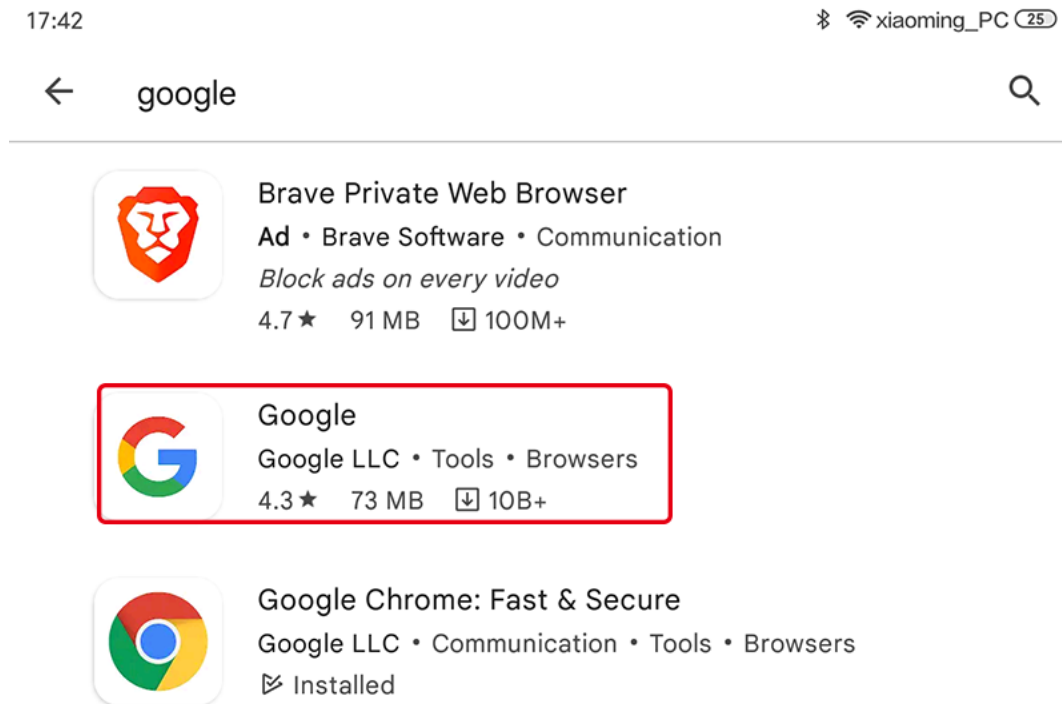
- Öffnen Sie die Datei `galaxy-rvr.ino`, die sich im Verzeichnis `galaxy-rvr-main\galaxy-rvr` befindet.
- Kommentieren Sie dann den AP-Modus bezogenen Code aus. Kommentieren Sie den STA-Modus bezogenen Code ein und füllen Sie das SSID und das PASSWORD Ihres Heim-WLANs ein.

```
/** Configure Wifi mode, SSID, password*/
// #define WIFI_MODE WIFI_MODE_AP
// #define SSID "GalaxyRVR"
// #define PASSWORD "12345678"

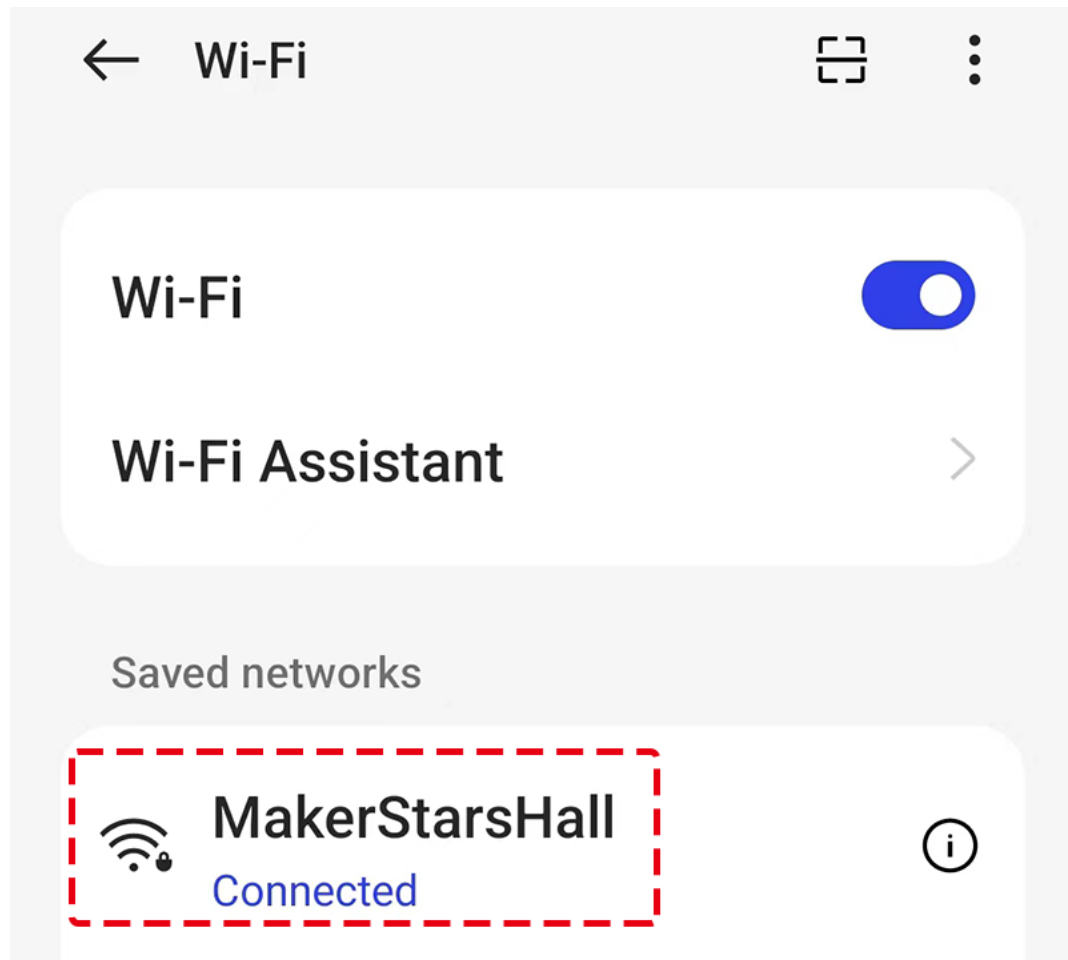
#define WIFI_MODE WIFI_MODE_STA
#define SSID "xxxxxxxxxx"
#define PASSWORD "xxxxxxxxxx"
```

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

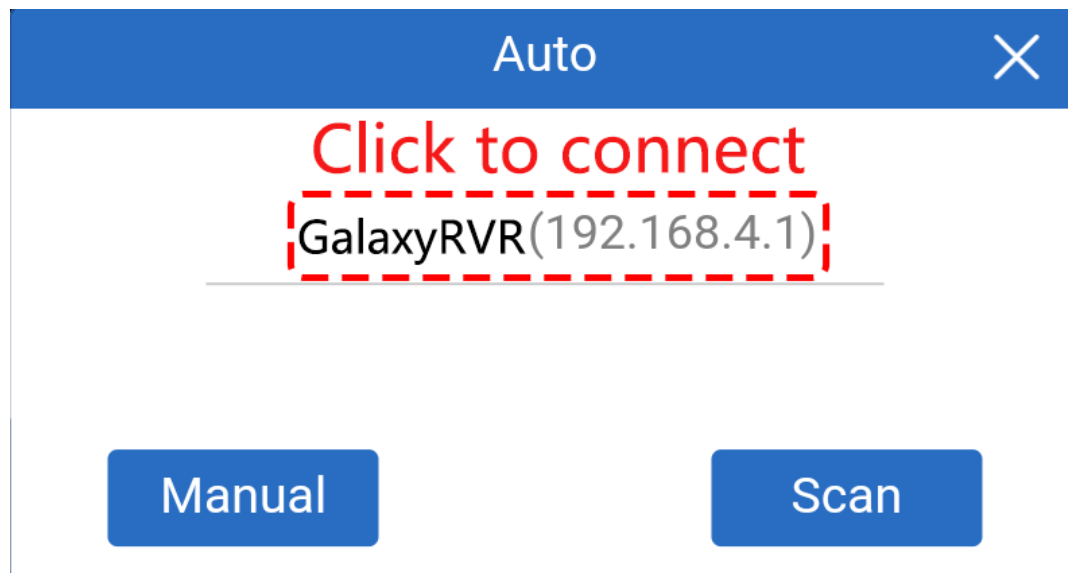
2. Suchen Sie in Google Play nach google, finden Sie die unten gezeigte App und installieren Sie sie.




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

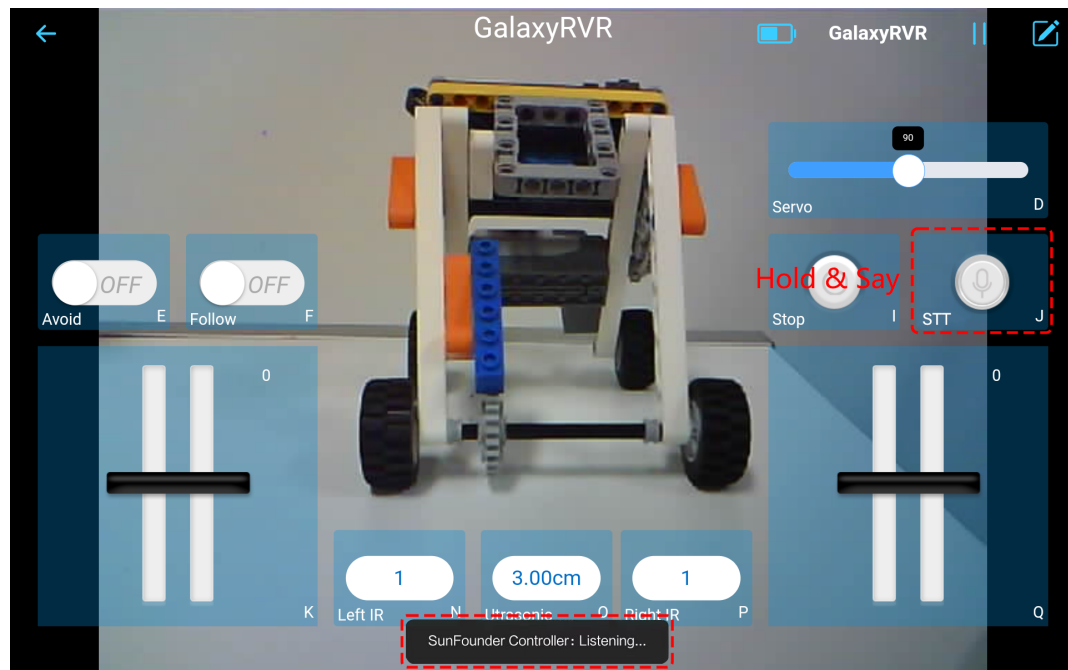


4. Öffnen Sie den zuvor in SunFounder Controller erstellten Controller und verbinden Sie ihn durch den -Button mit GalaxyRVR.



5. Tippen und halten Sie das **STT(J)**-Widget, nachdem Sie den -Button angeklickt haben. Es erscheint ein

Hinweis, dass es zuhört. Sagen Sie den folgenden Befehl, um das Auto zu bewegen.



- stop: Alle Bewegungen des Rovers können gestoppt werden.
- forward: Lassen Sie den Rover vorwärts fahren.
- backward: Lassen Sie den Rover rückwärts fahren.
- left: Lassen Sie den Rover nach links abbiegen.
- right: Lassen Sie den Rover nach rechts abbiegen.

## 5.4 Q4: Über die ESP32 CAM Firmware

Hier ist der Firmware-Link des ESP32 CAM:

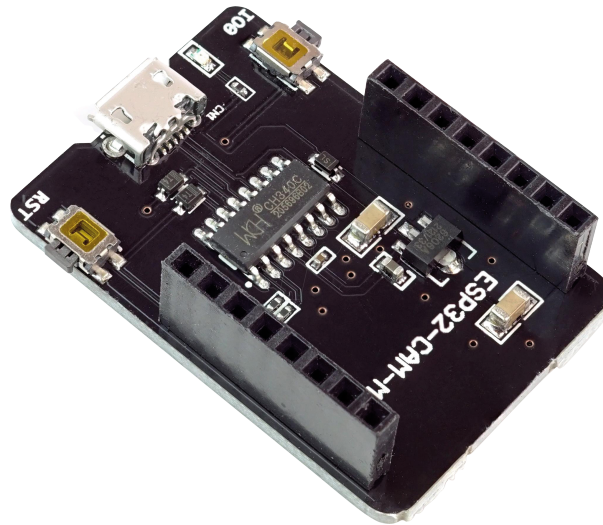
## 5.5 Q5: Wie wird eine neue Firmware auf einen ESP32 CAM geflasht?

Das Kameramodul wird normalerweise ab Werk mit vorinstallierter Firmware geliefert. Wenn Sie jedoch auf ein Problem mit Datenkorruption stoßen oder die Firmware aktualisieren müssen, können Sie dies mit der Arduino IDE tun. Hier ist, wie es gemacht wird:

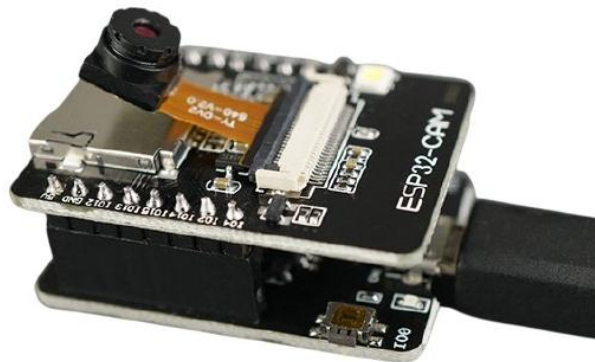
### 1. Vorbereitung des Programmiergeräts

1. Zuerst benötigen Sie ein Programmiergerät.





2. Stecken Sie den ESP32-CAM in das Programmiergerät und schließen Sie dann das Programmiergerät an Ihren Computer an.

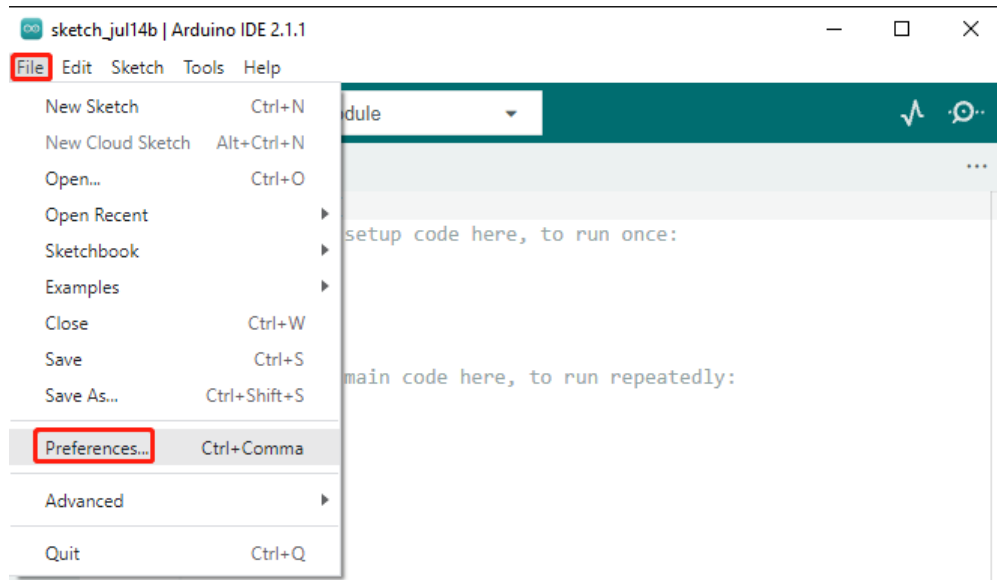


## 2. Installieren Sie das ESP32-Board

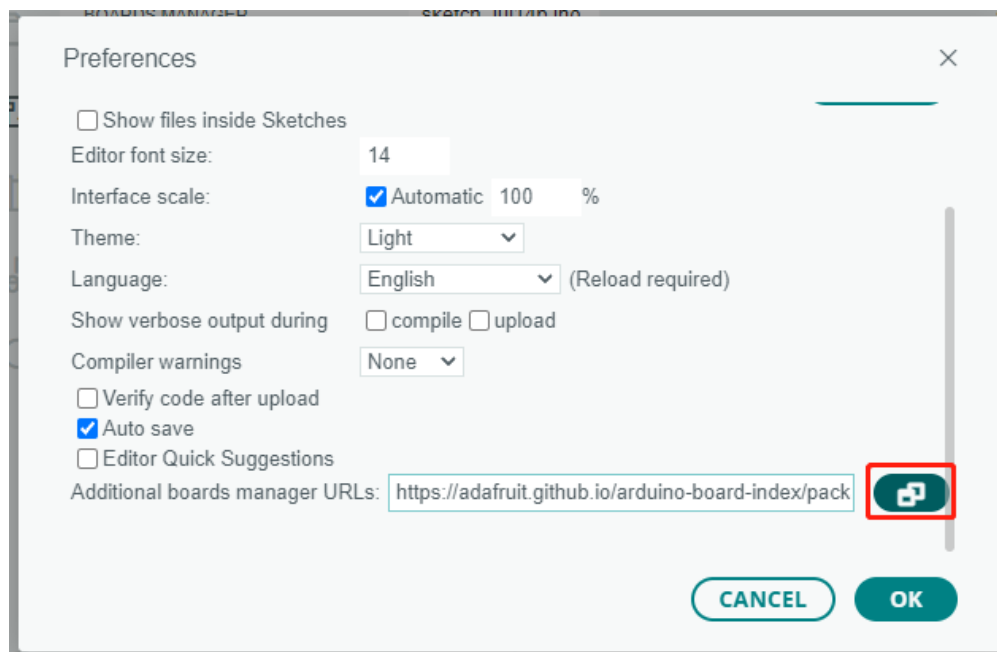
Um den ESP32-Mikrocontroller zu programmieren, müssen Sie das ESP32-Boardpaket in der Arduino IDE installieren. Befolgen Sie diese Schritte:

1. Gehen Sie zu **Datei** und wählen Sie **Einstellungen** im Dropdown-Menü aus.

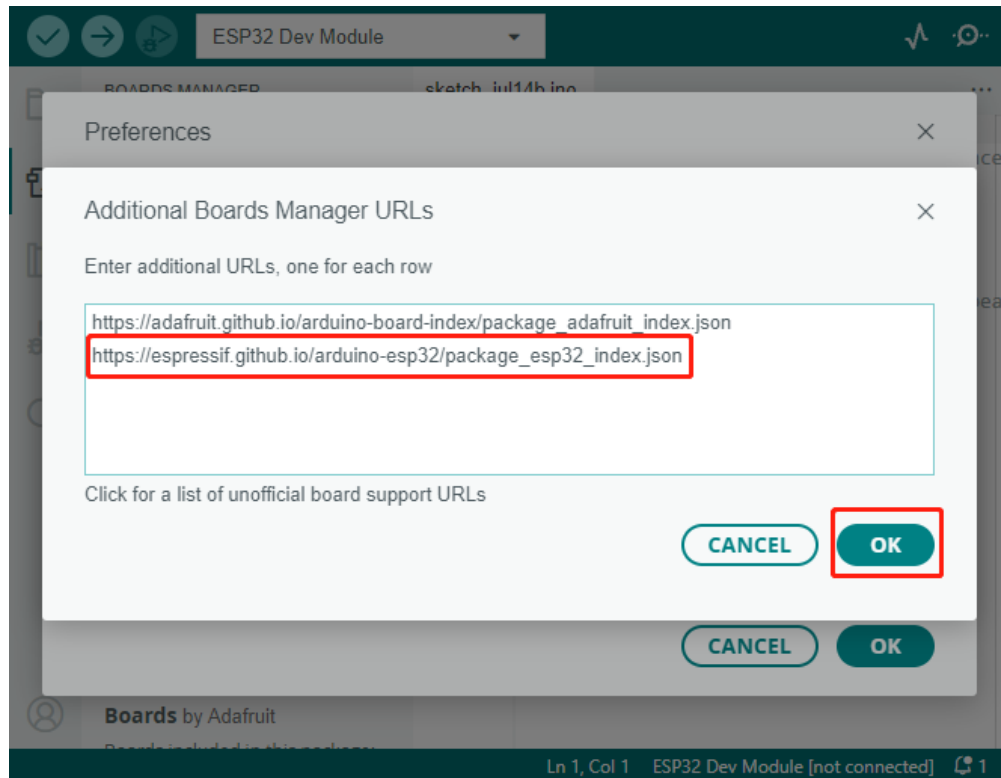




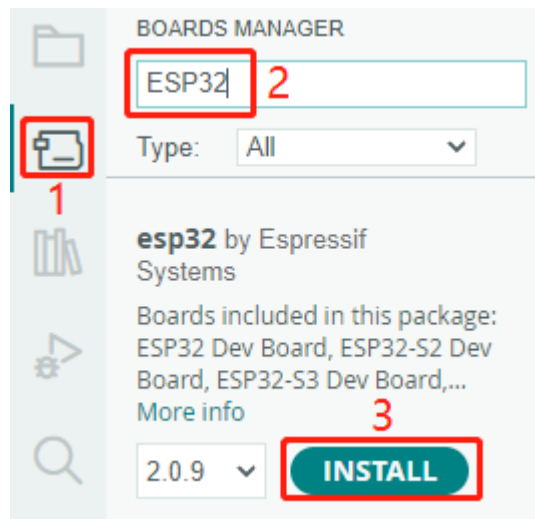
2. In den **Einstellungen** finden Sie das Feld **Zusätzliche Board-Verwalter-URLs**. Klicken Sie darauf, um das Textfeld zu aktivieren.



3. Fügen Sie die folgende URL in das Feld **Zusätzliche Board-Verwalter-URLs** ein: [https://espressif.github.io/arduino-esp32/package\\_esp32\\_index.json](https://espressif.github.io/arduino-esp32/package_esp32_index.json). Diese URL verlinkt auf die Paketindexdatei für ESP32-Boards. Klicken Sie auf **OK**, um die Änderungen zu speichern.

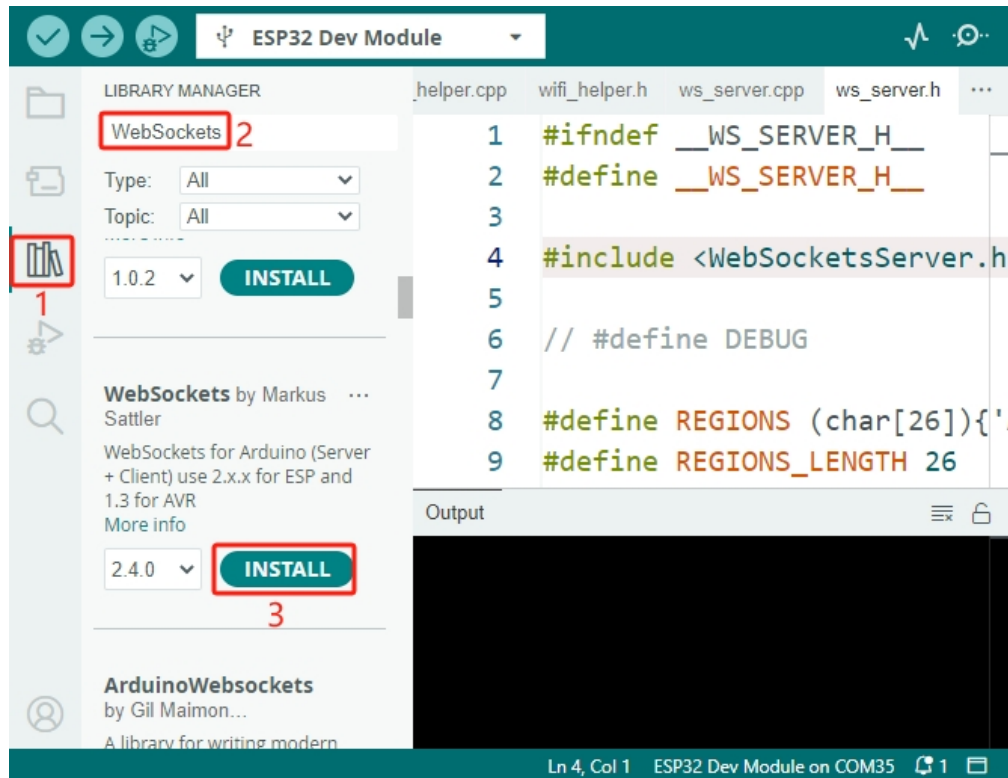


4. Im **Board-Verwalter** suchen Sie nach **ESP32**. Klicken Sie auf die Schaltfläche **Installieren**, um mit der Installation zu beginnen. Dadurch wird das ESP32-Boardpaket heruntergeladen und installiert.

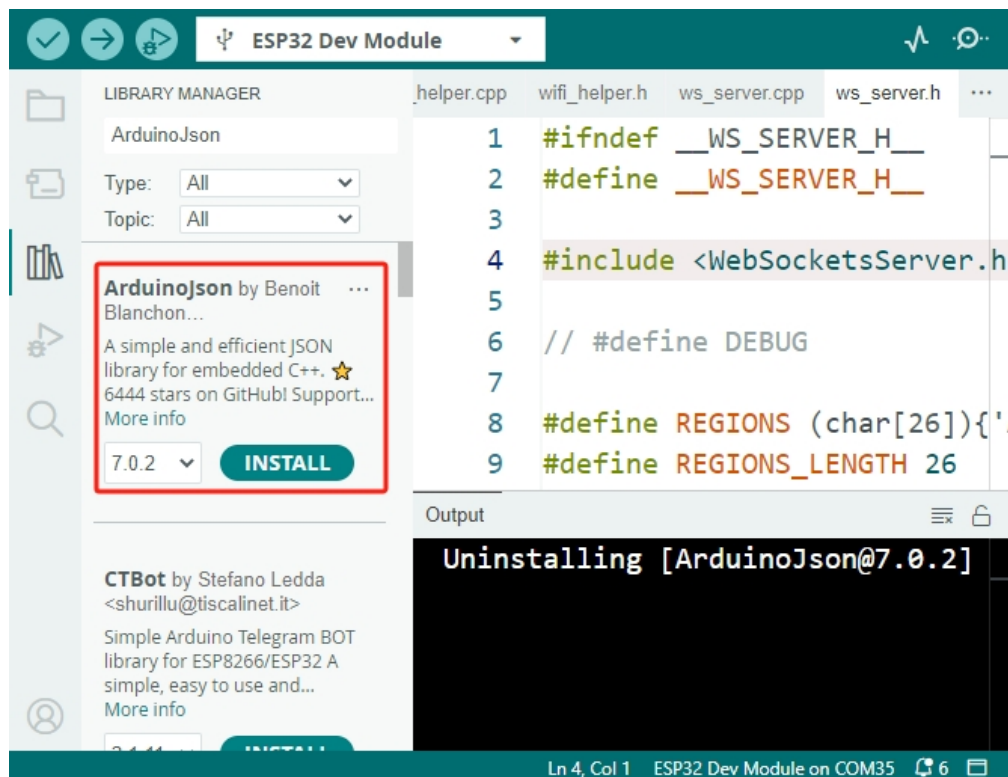


### 3. Installieren Sie die erforderlichen Bibliotheken

1. Installieren Sie die WebSockets-Bibliothek aus dem **BIBLIOTHEKSVERWALTER**.



2. Befolgen Sie die gleichen Schritte, um die ArduinoJson-Bibliothek zu installieren.

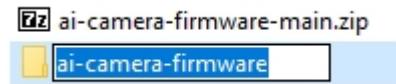


#### 4. Firmware herunterladen und hochladen

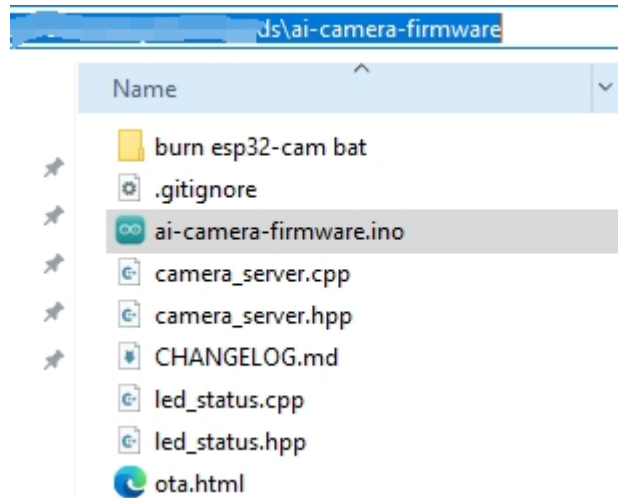
1. Laden Sie die Firmware-Datei hier herunter.

- ai-camera-firmware

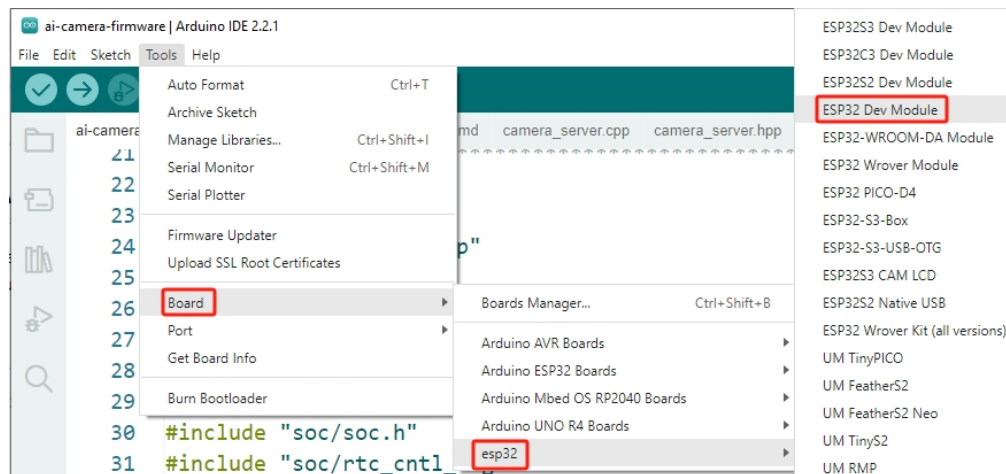
2. Entpacken Sie die heruntergeladene Firmware-Datei und benennen Sie den extrahierten Ordner von ai-camera-firmware-main in ai-camera-firmware um.



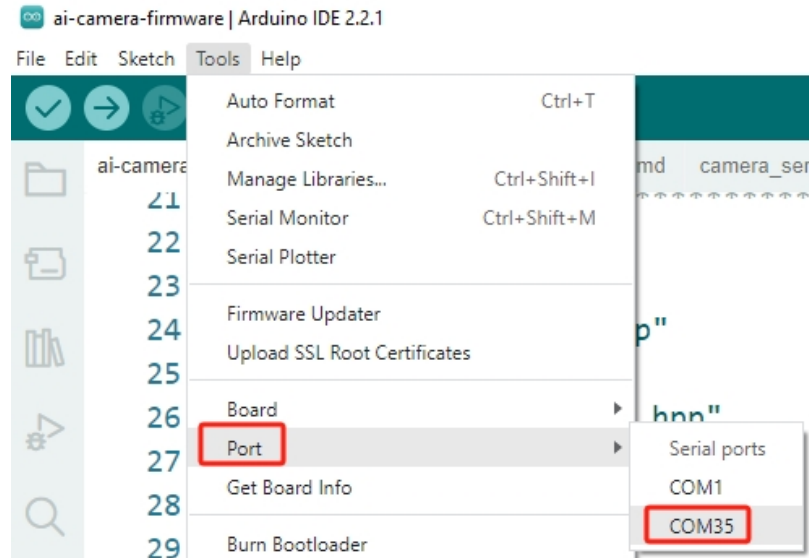
3. Öffnen Sie die Datei ai-camera-firmware.ino mit der Arduino IDE, die auch die zugehörigen Code-Dateien öffnet.



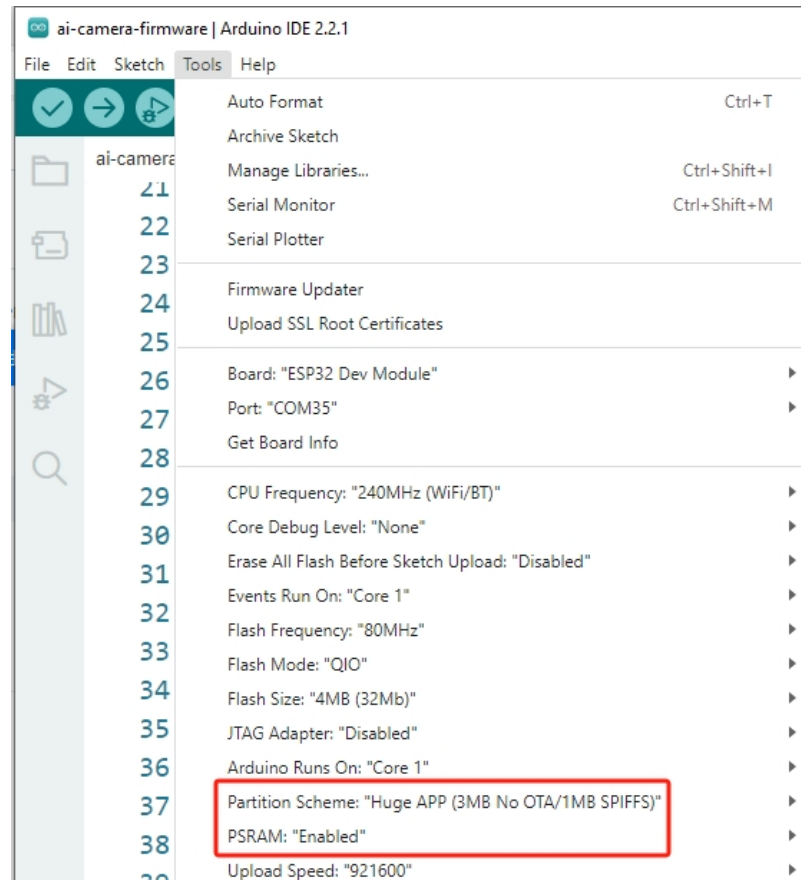
4. Wählen Sie **Board** -> **esp32** -> **ESP32 Dev Module** aus.



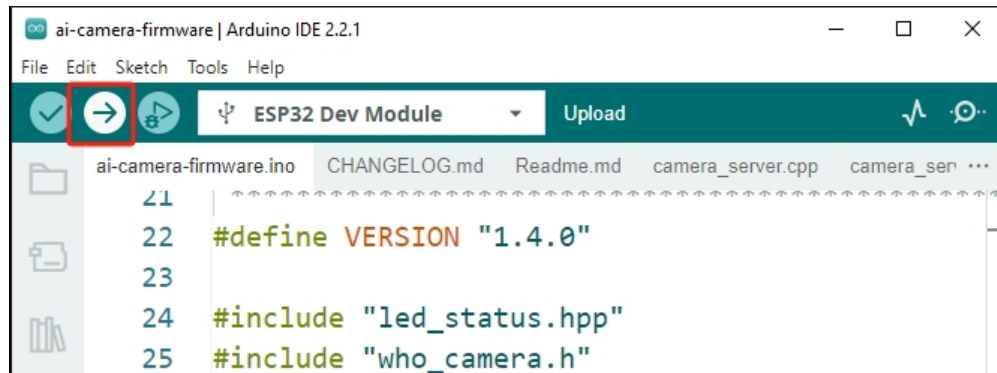
5. Wählen Sie den richtigen Anschluss aus.



6. Stellen Sie sicher, dass Sie **PSRAM** aktivieren und in der **Partitionsschema** die Option **Huge APP** auswählen.



7. Schließlich laden Sie die Firmware auf den ESP32-CAM hoch.



8. Nach erfolgreichem Hochladen der Firmware finden Sie weitere Informationen unter diesem Link: <https://github.com/sunfounder/ai-camera-firmware>.

### Urheberrechtshinweis

Alle Inhalte, einschließlich aber nicht beschränkt auf Texte, Bilder und Code in diesem Handbuch, sind Eigentum der SunFounder Company. Sie sollten es nur für persönliches Studium, Untersuchungen, Vergnügen oder andere nicht-kommerzielle oder gemeinnützige Zwecke verwenden, unter Einhaltung der zugehörigen Vorschriften und Urheberrechtsgesetze, ohne die gesetzlichen Rechte des Autors und der relevanten Rechteinhaber zu verletzen. Für jede Einzelperson oder Organisation, die diese zu kommerziellen Zwecken ohne Erlaubnis nutzt, behält sich das Unternehmen das Recht vor, rechtliche Schritte einzuleiten.