
SunFounder 3in1 Kit

www.sunfounder.com

19.02.2024

Inhaltsverzeichnis

1	Erfahren Sie mehr über die Komponenten in Ihrem Kit	3
1.1	Arduino Uno R4 Minima	3
1.2	ESP8266 Modul	7
1.3	Breadboard	10
1.4	Widerstand	11
1.5	Kondensator	14
1.6	Jumper-Kabel	16
1.7	74HC595	16
1.8	LED	18
1.9	RGB LED	19
1.10	7-Segment-Anzeige	21
1.11	I2C LCD1602	24
1.12	Summer	26
1.13	TT Motor	27
1.14	Servo	28
1.15	Kreiselpumpe	31
1.16	L9110 Motortreibermodul	32
1.17	Knopf	34
1.18	Reedschalter	35
1.19	Potentiometer	36
1.20	Joystick Modul	38
1.21	IR-Empfänger	39
1.22	Fotowiderstand	42
1.23	Thermistor	43
1.24	DHT11 Feuchtigkeits- und Temperatursensor	44
1.25	Linienverfolgungsmodul	46
1.26	Bodenfeuchtigkeitsmodul	47
1.27	Hinderniserkennungsmodul	49
1.28	Ultraschall-Modul	50
2	Mit Arduino loslegen	53
2.1	Was ist Arduino?	53
2.2	Was kann Arduino tun?	54
2.3	Wie erstellt man ein Arduino-Projekt	54
3	Code herunterladen	85

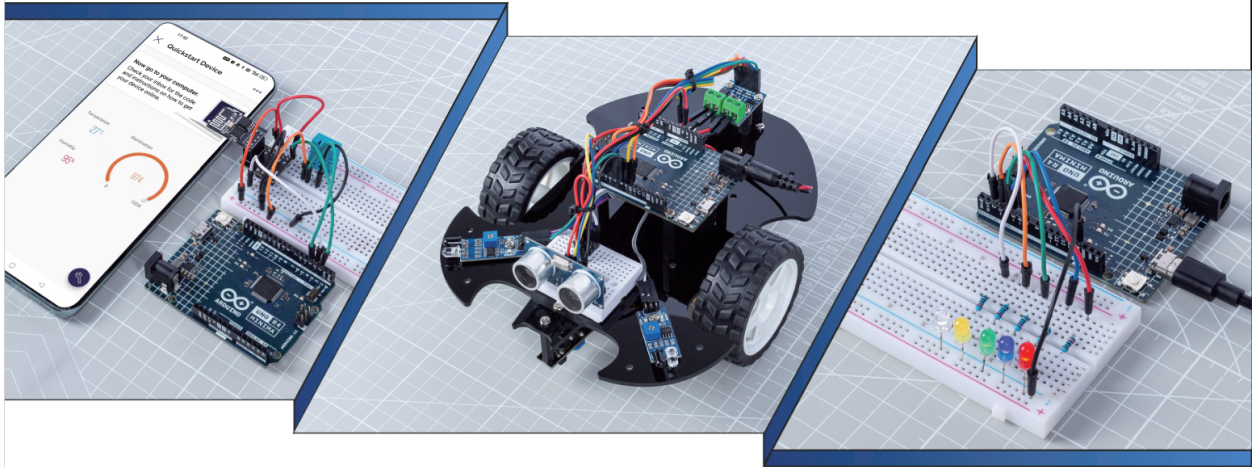
4	Grundprojekte	87
4.1	1. Digital Write	87
4.2	2. Analog schreiben	98
4.3	3. Digital Lesen	106
4.4	4. Analoges Lesen	119
4.5	5. Mehr Syntax	134
4.6	6. Lustiges Projekt	198
5	Auto-Projekte	223
5.1	Auto zusammenbauen	223
5.2	1. Bewegung	235
5.3	2. Steuerung durch Code	240
5.4	3. Beschleunigung	244
5.5	4. Der Linie folgen	245
5.6	5. Spielen mit dem Hindernisvermeidungs-Modul	249
5.7	6. Mit dem Ultraschallmodul spielen	253
5.8	7. Folge deiner Hand	256
5.9	8. Selbstfahrendes Auto	258
5.10	9. Fernbedienung	261
5.11	10. Starten mit einem Tastendruck	266
5.12	11. Geschwindigkeitskalibrierung	268
6	IoT-Projekte	271
6.1	1. Erste Schritte mit Blynk	272
6.2	2. Daten von Blynk abrufen	286
6.3	3. Daten an Blynk senden	297
6.4	4. Cloud-Musikspieler	301
6.5	5. Überwachung des häuslichen Umfelds	308
6.6	6. Pflanzenüberwachung	316
6.7	7. Strombegrenztes Tor	323
6.8	8. IoT-Auto	330
7	FAQ	339
7.1	Wie verwendet man Blynk auf einem Mobilgerät?	339
7.2	Wie lädt man die Firmware für das ESP8266-Modul erneut hoch?	341
8	Danke	353
9	Urheberrechtshinweis	355

Vielen Dank für die Auswahl unseres 3-in-1 Starterkits.

Bemerkung: Dieses Dokument ist in den folgenden Sprachen verfügbar.

-
-
-

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



Haben Sie jemals ein Lernset online gekauft und es kam nur mit einer simplen PDF oder einem kleinen Heftchen, welches lediglich die Schritte zum Aufbau des Projekts anzeigt?

Oder möchten Sie vielleicht Ihr eigenes Smart Car bauen, finden aber, dass die online verfügbaren Angebote teuer und kompliziert sind?

Oder haben Sie nützliche und interessante IoT-Projekte von anderen gesehen, wissen aber nicht, wo Sie anfangen sollen?

All diese Probleme können mit unserem 3-in-1 Starterkit gelöst werden.

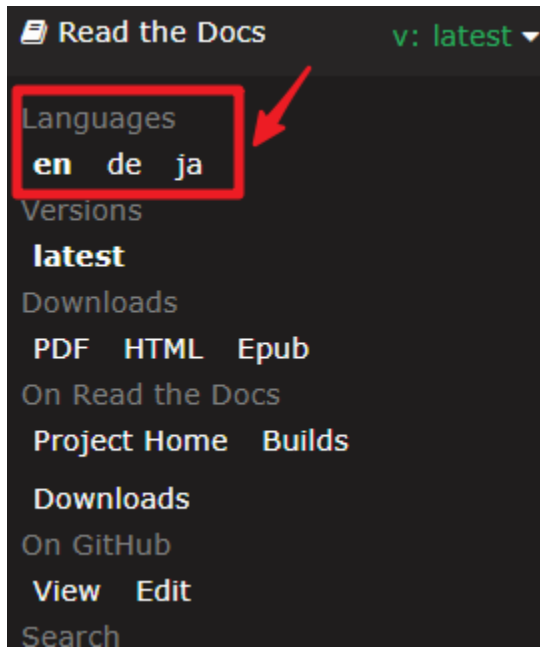
Im 3-in-1 Starterkit finden Sie einen kompletten Arduino-Kurs, der Anfängern hilft, Arduino zu lernen. Zudem gibt es eine Vielzahl von interessanten Projekten, die andere Lernkits nicht bieten, wie z.B. Smart Car-Projekte und IoT-Projekte. Wenn Sie dem Kurs Schritt für Schritt folgen, werden Sie nicht nur Code kopieren und einfügen, sondern Ihren eigenen Code schreiben und Ihr Arduino-Projekt nach Ihren Wünschen umsetzen.

Legen Sie los! Beginnen Sie mit dem Programmieren von Arduino mit Vertrauen, von Null bis zum Profi!

Wenn Sie Fragen haben oder andere interessante Ideen, senden Sie bitte eine E-Mail an service@sunfounder.com.

Über die Anzeigesprache

Dieses Dokument ist auch in anderen Sprachen verfügbar. Um die Anzeigesprache zu wechseln, klicken Sie bitte auf das **Read the Docs** -Symbol in der unteren linken Ecke der Seite.



Erfahren Sie mehr über die Komponenten in Ihrem Kit

Nach dem Öffnen des Pakets überprüfen Sie bitte, ob die Anzahl der Komponenten mit der Produktbeschreibung übereinstimmt und ob alle Komponenten in gutem Zustand sind.

- 3 in 1 Kit Komponentenliste

Im Folgenden wird jede Komponente vorgestellt, einschließlich des Funktionsprinzips der Komponente und der zugehörigen Projekte.

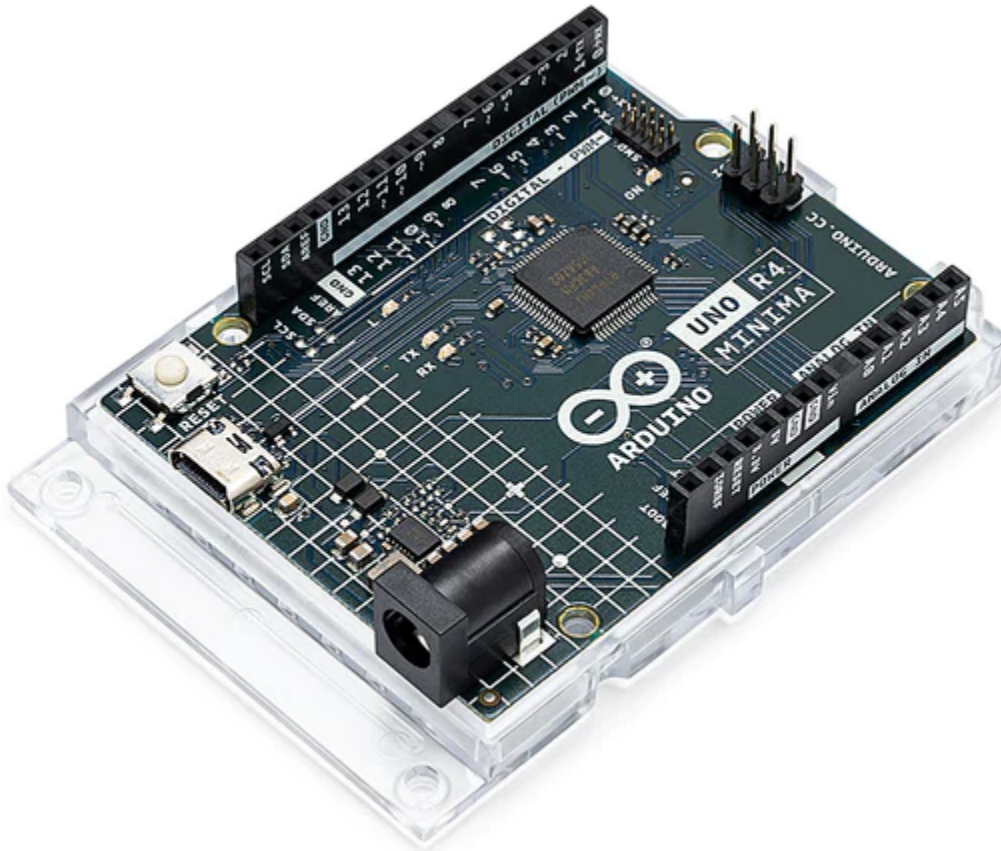
Steuerplatine

1.1 Arduino Uno R4 Minima

Überblick

Erweitert und verbessert, ist das Arduino UNO R4 Minima mit einem leistungsstarken 32-Bit-Mikrocontroller von Renesas ausgestattet. Machen Sie sich bereit für mehr Rechenleistung, erweiterten Speicher und eine ganz neue Ebene von integrierten Peripheriegeräten. Das Beste daran? Die Kompatibilität mit bestehenden Shields und Zubehörteilen bleibt erhalten, und es sind keine Änderungen am Standard-Formfaktor oder der 5-V-Betriebsspannung erforderlich.

Als Teil des Arduino-Ökosystems ist das UNO R4 eine verlässliche Ergänzung sowohl für Anfänger als auch für erfahrene Elektronikbegeisterte. Egal, ob Sie gerade erst anfangen oder die Grenzen Ihrer Projekte erweitern möchten, dieses robuste Board liefert jedes Mal zuverlässige Leistung.



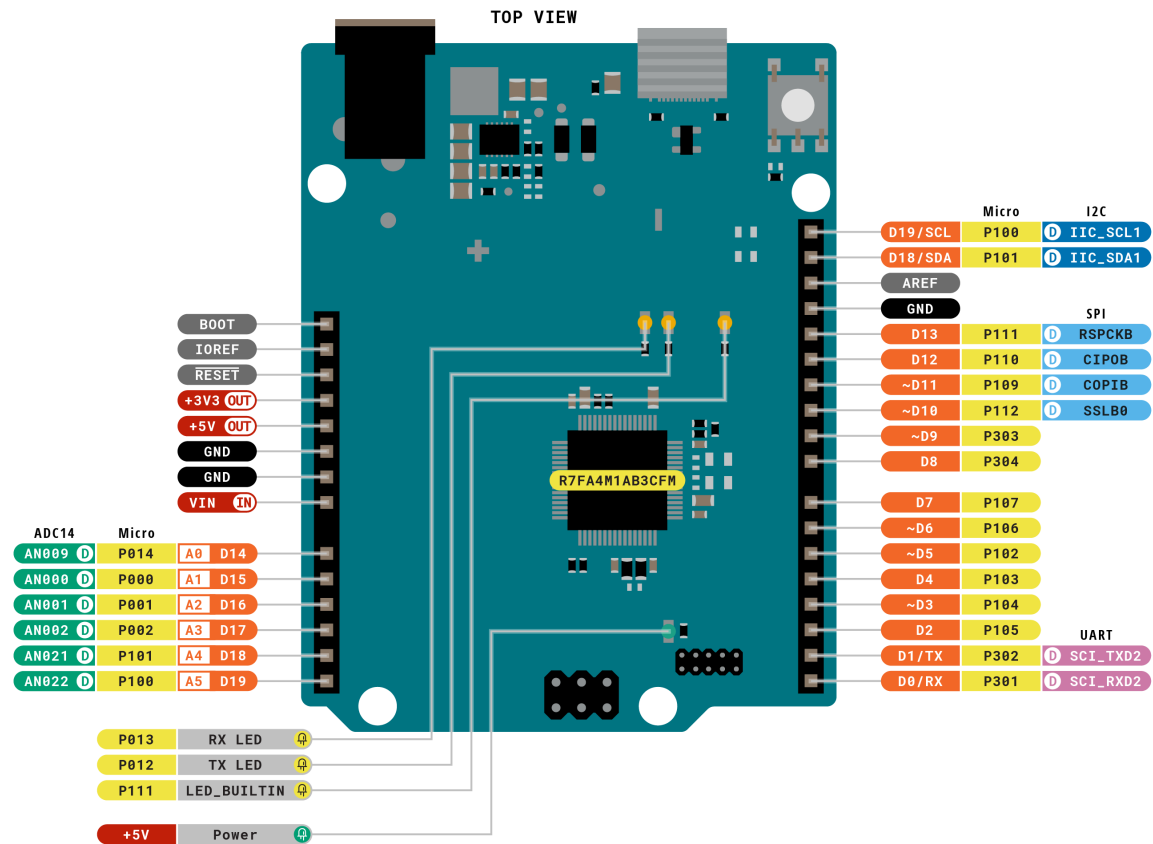
Hier die Vorzüge des UNO R4 Minima:

- **Hardwarekompatibilität mit UNO-Formfaktor:** Das UNO R4 Minima behält den gleichen Formfaktor, Pinbelegung und 5 V Betriebsspannung wie sein Vorgänger, das UNO R3, bei. Dies gewährleistet einen reibungslosen Übergang für bestehende Shields und Projekte und nutzt das umfangreiche und einzigartige bereits etablierte Ökosystem des Arduino UNO.
- **Erweiterter Speicher und schnellerer Takt:** Bereiten Sie sich auf präzisere Berechnungen und die problemlose Handhabung komplexer Projekte vor. Das UNO R4 Minima bietet erweiterten Speicher und eine schnellere Taktung, was Sie in die Lage versetzt, anspruchsvolle Aufgaben mühelos zu bewältigen.
- **Zusätzliche integrierte Peripheriegeräte:** Das UNO R4 Minima bietet eine Reihe von integrierten Peripheriegeräten, darunter ein 12-Bit-DAC, CAN BUS und OP AMP. Diese zusätzlichen Komponenten bieten Ihnen erweiterte Fähigkeiten und Flexibilität in Ihren Designs.
- **Erweiterte 24 V Toleranz:** Das UNO R4 Minima unterstützt jetzt einen breiteren Eingangsspannungsbereich und ermöglicht Stromversorgungen von bis zu 24 V. Dies ermöglicht eine nahtlose Integration mit Motoren, LED-Streifen und anderen Aktuatoren, wodurch Ihre Projekte durch die Nutzung einer einzigen Stromquelle vereinfacht werden.
- **SWD-Anschluss:** Das Debuggen ist ein kritischer Aspekt jedes Projekts. Verbinden Sie einfach einen externen Debugger mit dem UNO R4 Minima und überwachen Sie mühelos das Innenleben Ihres Systems. Behalten Sie die Kontrolle und gewinnen Sie wertvolle Einblicke.
- **HID-Unterstützung:** Das UNO R4 Minima verfügt über integrierte HID (Human Interface Device) Unterstützung und kann eine Maus oder Tastatur simulieren, wenn es über ein USB-Kabel mit einem Computer verbunden ist. Diese praktische Funktion ermöglicht es, Tastenanschläge und Mausbewegungen an einen Computer zu senden, wodurch die Benutzerfreundlichkeit und Funktionalität erhöht wird.

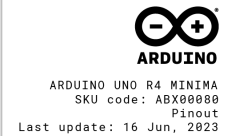
Technische Daten

- **Board:**
 - Name: Arduino® UNO R4 Minima
 - SKU: ABX00080
- Mikrocontroller: Renesas RA4M1 (Arm® Cortex®-M4)
- USB: USB-C® Programmieranschluss
- **Pins:**
 - Digitale I/O-Pins: 14
 - Analogeingangspin: 6
 - DAC: 1
 - PWM-Pins: 6
- **Kommunikation:**
 - UART: Ja, 1x
 - I2C: Ja, 1x
 - SPI: Ja, 1x
 - CAN: Ja, 1 CAN Bus
- **Energie:**
 - Betriebsspannung des Kreises: 5 V
 - Eingangsspannung (VIN): 6-24 V
 - Gleichstrom pro I/O-Pin: 8 mA
- Taktgeschwindigkeit Hauptkern: 48 MHz
- Speicher RA4M1: 256 kB Flash, 32 kB RAM
- **Abmessungen:**
 - Breite: 68,85 mm
 - Länge: 53,34 mm

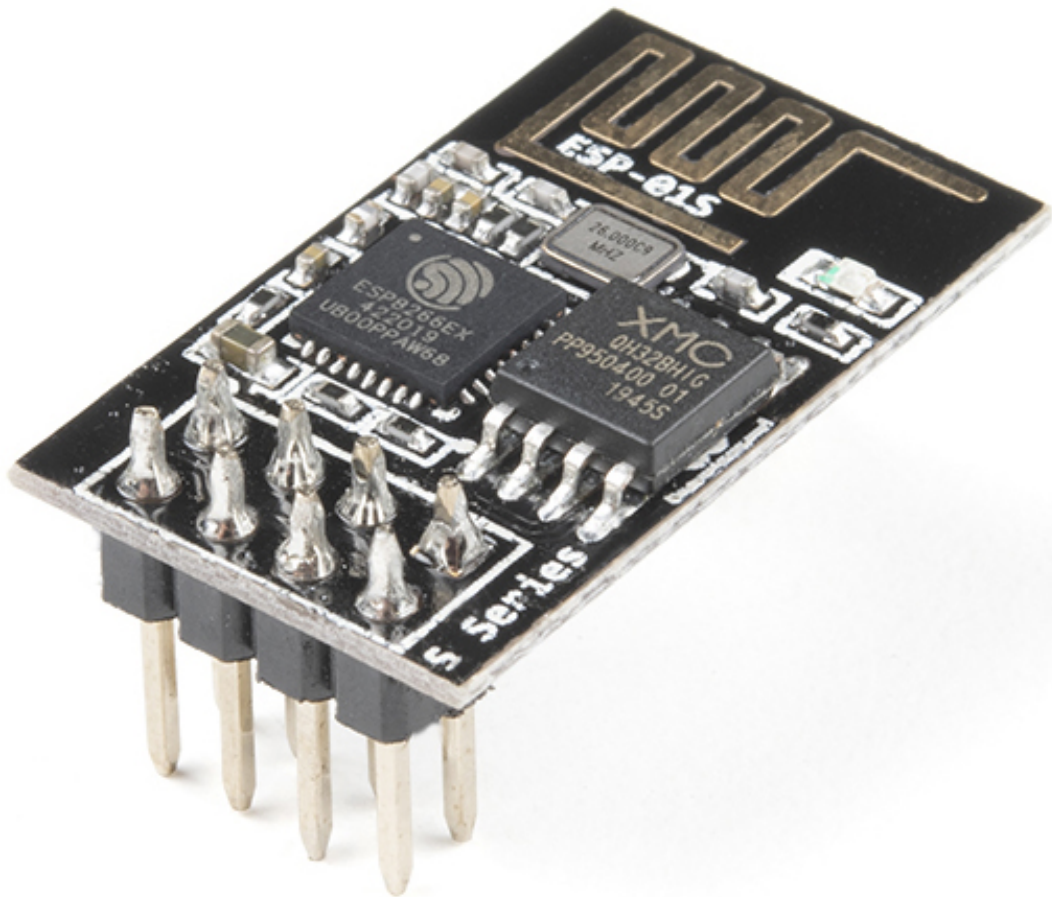
Pinbelegung



Legend:	Digital	I2C	Other SERIAL
Power	Analog	SPI	Analog
Ground	Main Part	UART/USART	PWM/Timer



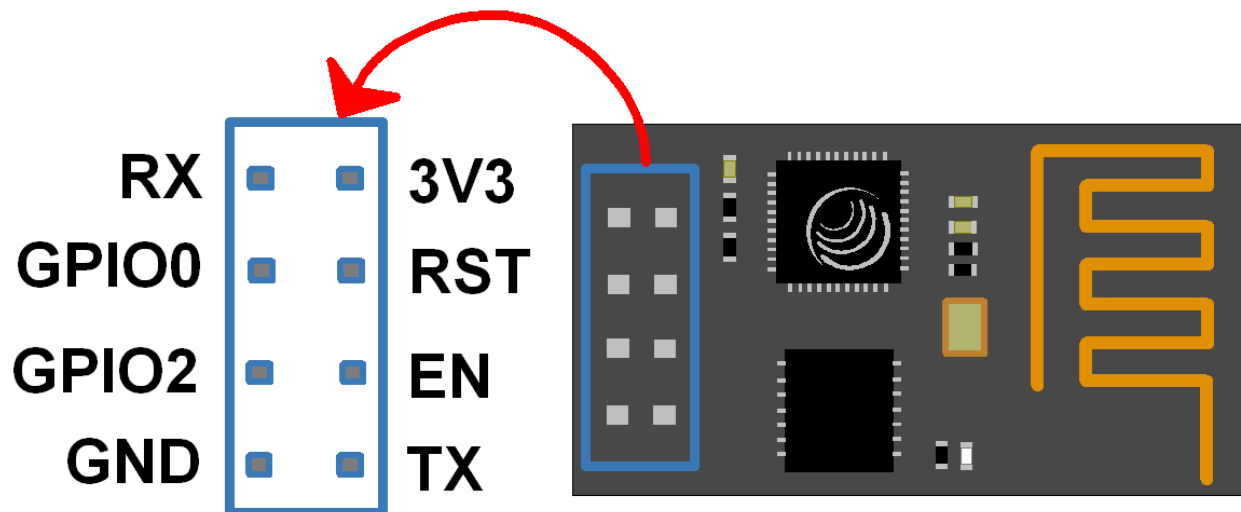
1.2 ESP8266 Modul



Das ESP8266 ist ein kostengünstiger Wi-Fi-Mikrochip mit integrierter TCP/IP-Netzwerksoftware und Mikrocontroller-Fähigkeit, produziert von Espressif Systems in Shanghai, China.

Der Chip erregte erstmals im August 2014 die Aufmerksamkeit westlicher Bastler mit dem ESP-01 Modul, hergestellt von einem Drittanbieter Ai-Thinker. Dieses kleine Modul ermöglicht es Mikrocontrollern, sich mit einem Wi-Fi-Netzwerk zu verbinden und einfache TCP/IP-Verbindungen mit Hayes-Stil-Befehlen herzustellen. Zu Beginn gab es jedoch kaum englischsprachige Dokumentationen über den Chip und die von ihm akzeptierten Befehle. Der sehr niedrige Preis und die Tatsache, dass es auf dem Modul sehr wenige externe Komponenten gab, was darauf hindeutete, dass es in größeren Mengen sehr kostengünstig sein könnte, lockte viele Hacker an, das Modul, den Chip und die darauf befindliche Software zu erkunden, sowie die chinesische Dokumentation zu übersetzen.

Pins des ESP8266 und ihre Funktionen:

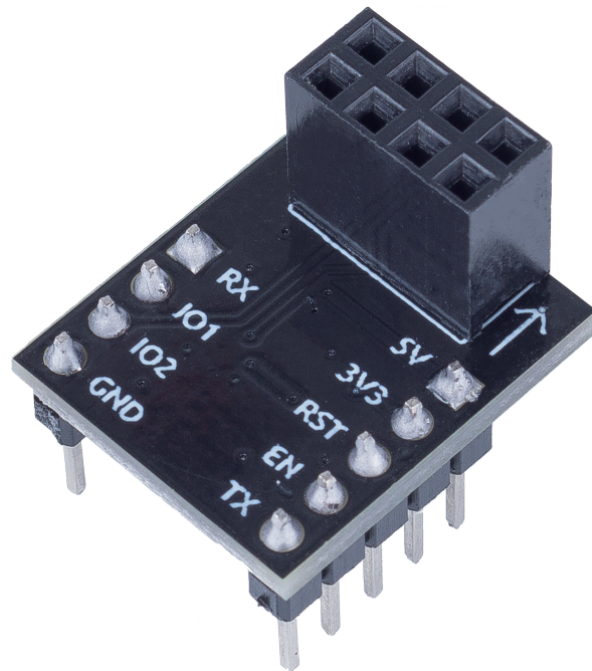


Tab. 1: ESP8266-01 Pins

Pin	Bezeichnung	Beschreibung
1	TXD	UART_TXD, Senden; Allgemeiner Eingabe/Ausgabe-Zweck: GPIO1; Pull-down beim Start nicht erlaubt.
2	GND	GND
3	CU_PD	Funktioniert bei hohem Pegel; Ausschaltung bei niedrigem Pegel.
4	GPIO2	Muss beim Einschalten auf hohem Pegel sein; Pull-down nicht erlaubt; Standardmäßig Pull-up;
5	RST	Externes Reset-Signal, Reset bei niedrigem Pegel; Funktioniert bei hohem Pegel (standardmäßig hoch);
6	GPIO0	WiFi Statusanzeige; Betriebsmodusauswahl: Pull-up: Flash Boot, Betriebsmodus; Pull-down: UART Download, Download-Modus.
7	VCC	Stromversorgung (3,3V)
8	RXD	UART_RXD, Empfangen; Allgemeiner Eingabe/Ausgabe-Zweck: GPIO3;

- ESP8266 - Espressif
- ESP8266 AT Befehlssatz

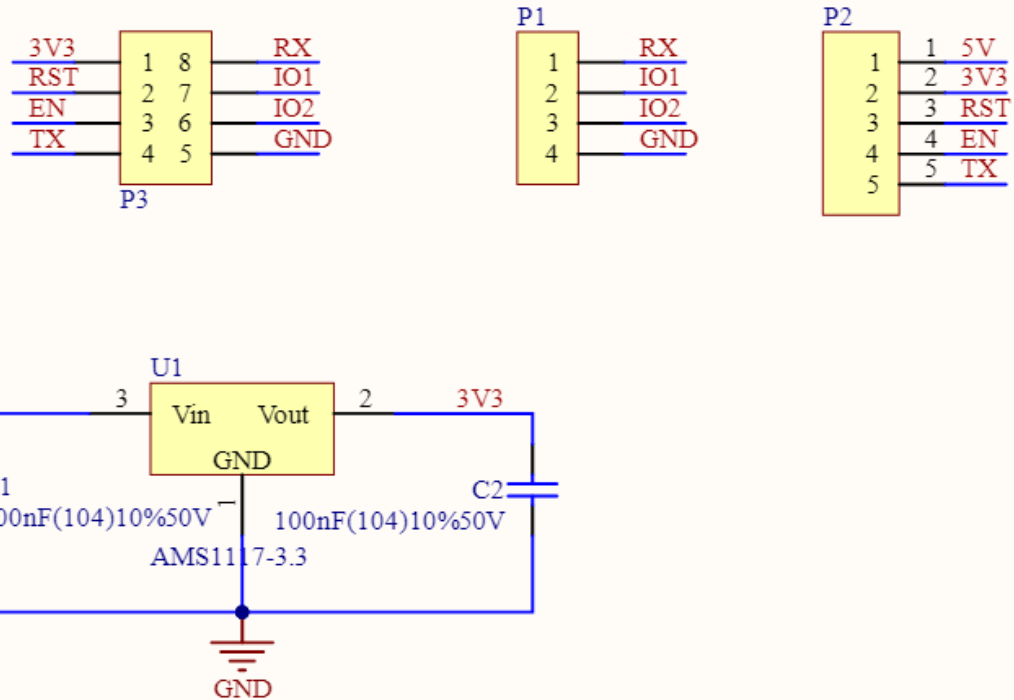
1.2.1 ESP8266 Adapter



Der ESP8266 Adapter ist eine Erweiterungsplatine, die es ermöglicht, das ESP8266 Modul auf einem Steckbrett zu verwenden.

Er passt perfekt zu den Pins des ESP8266 selbst und fügt zudem einen 5V Pin hinzu, um die Spannung von der Arduino-Platine zu erhalten. Der integrierte AMS1117-Chip wird verwendet, um das ESP8266 Modul nach der Spannungsabsenkung auf 3,3V zu betreiben.

Das Schaltbild ist wie folgt:

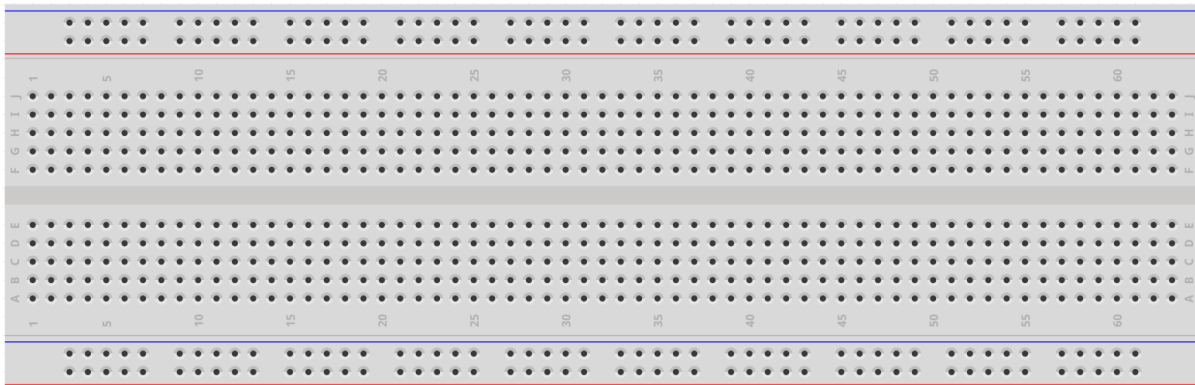


Beispiel

- *IoT-Projekte* (IoT-Projekt)

Grundlagen

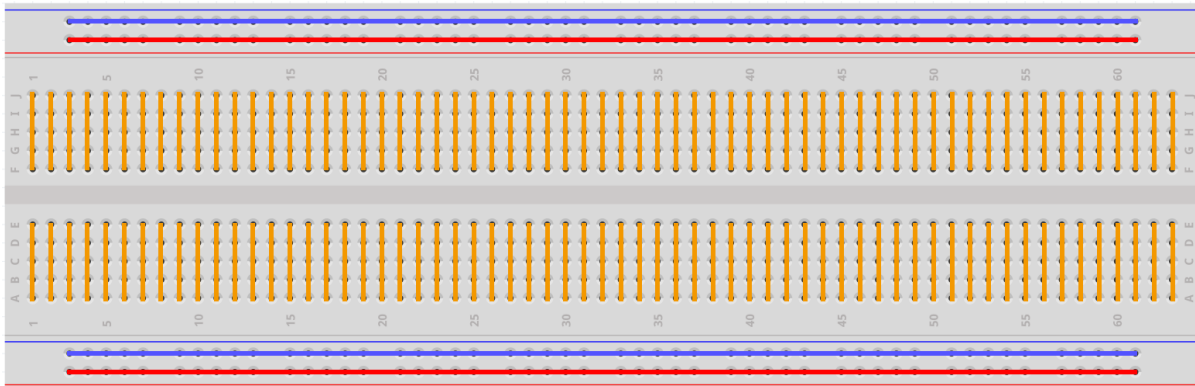
1.3 Breadboard



Ein Breadboard ist eine Konstruktionsbasis für den Prototypenbau von Elektronik. Ursprünglich bezog sich das Wort auf ein buchstäbliches Brotschneidebrett, ein poliertes Stück Holz, das zum Brotschneiden verwendet wurde.[1] In den 1970er Jahren wurde das lötfreie Breadboard (auch als Steckbrett oder Terminalarray-Platine bekannt) eingeführt, und heutzutage bezeichnet der Begriff „Breadboard“ hauptsächlich diese.

Es wird verwendet, um Schaltkreise schnell aufzubauen und zu testen, bevor ein Schaltkreisdesign abgeschlossen wird. Es verfügt über viele Löcher, in die oben erwähnte Komponenten wie ICs und Widerstände sowie Jumperkabel eingesteckt werden können. Das Breadboard ermöglicht es Ihnen, Komponenten leicht ein- und auszustecken.

Das Bild zeigt die interne Struktur eines Breadboards. Obwohl diese Löcher auf dem Breadboard unabhängig voneinander zu sein scheinen, sind sie tatsächlich intern durch Metallstreifen miteinander verbunden.



Wenn Sie mehr über Breadboards erfahren möchten, verweisen wir auf: [Wie man ein Breadboard benutzt - Science Buddies](#)

1.4 Widerstand



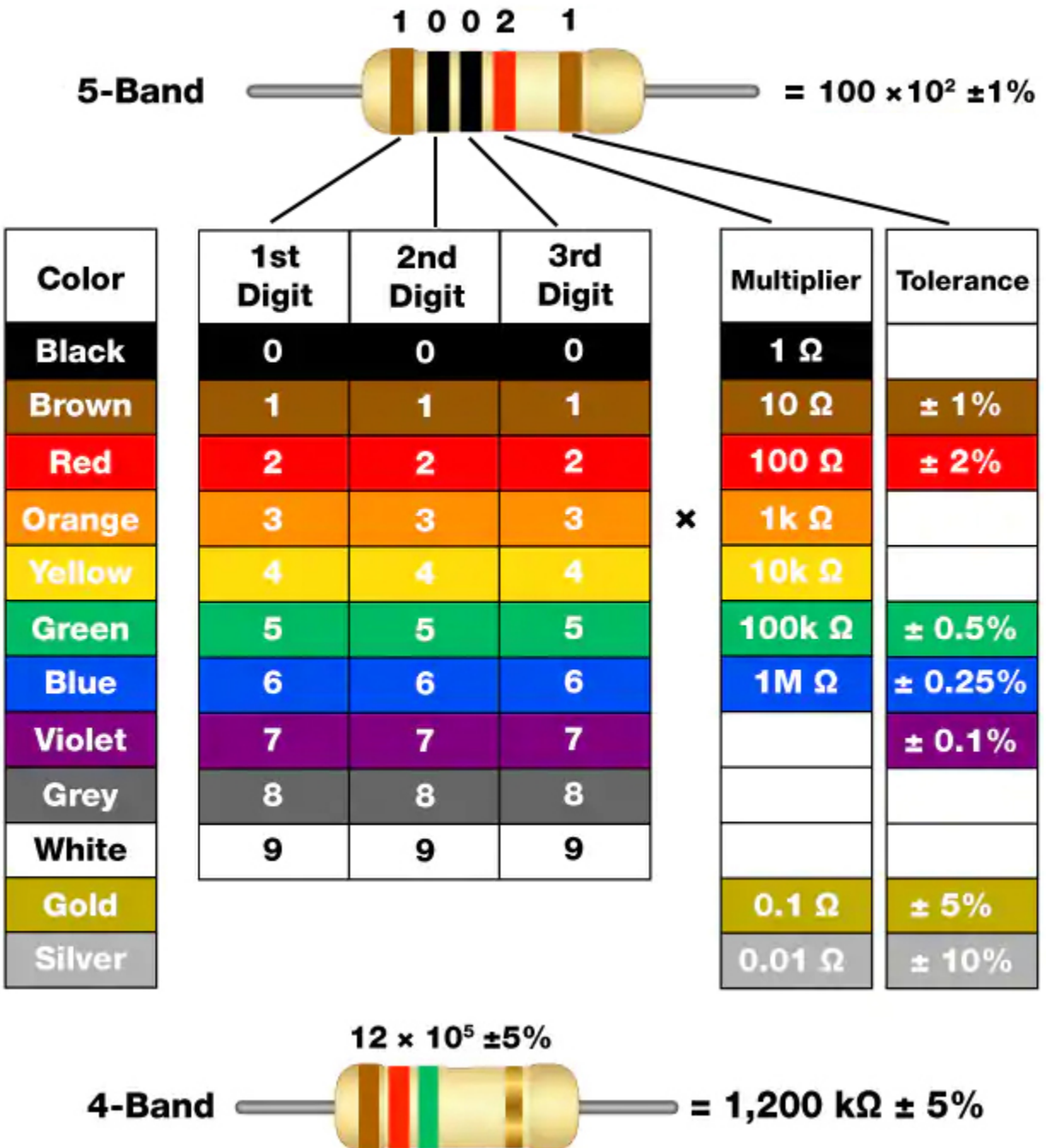
Ein Widerstand ist ein elektronisches Element, das den Strom in einem Zweig begrenzen kann. Ein fester Widerstand ist eine Art von Widerstand, dessen Widerstandswert nicht verändert werden kann, während der eines Potentiometers oder eines variablen Widerstands einstellbar ist.

Es gibt zwei gebräuchliche Schaltsymbole für Widerstände. In der Regel ist der Widerstandswert darauf vermerkt. Wenn Sie diese Symbole in einer Schaltung sehen, repräsentieren sie einen Widerstand.



ist die Einheit des Widerstands und größere Einheiten sind K, M usw. Ihre Beziehung ist wie folgt: 1 M = 1000 K, 1 K = 1000 . Normalerweise ist der Widerstandswert darauf vermerkt.

Bevor man einen Widerstand verwendet, muss man dessen Widerstandswert kennen. Hier sind zwei Methoden: Sie können die Farbbänder auf dem Widerstand betrachten oder ein Multimeter verwenden, um den Widerstand zu messen. Es wird empfohlen, die erste Methode zu verwenden, da sie bequemer und schneller ist.



Wie auf der Karte gezeigt, steht jede Farbe für eine Zahl.

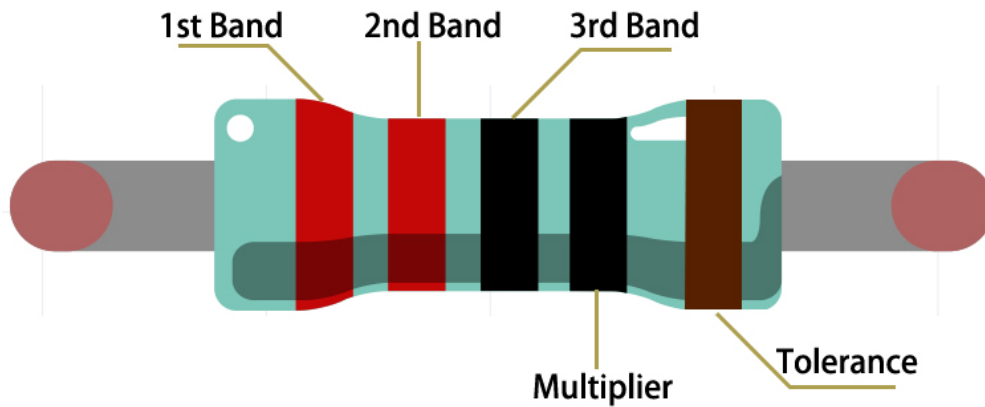
Schwarz	Braun	Rot	Orange	Gelb	Grün	Blau	Violett	Grau	Weiß	Gold	Silber
0	1	2	3	4	5	6	7	8	9	0.1	0.01

Die 4- und 5-Band-Widerstände werden häufig verwendet und haben 4 bzw. 5 Farbbänder.

Normalerweise, wenn Sie einen Widerstand erhalten, kann es schwierig sein zu entscheiden, von welchem Ende aus man die Farben lesen sollte. Ein Tipp ist, dass der Abstand zwischen dem 4. und 5. Band vergleichsweise größer ist.

Daher können Sie die Lücke zwischen den beiden Farbbändern an einem Ende des Widerstands betrachten; wenn sie größer als alle anderen Bandlücken ist, dann sollten Sie von der gegenüberliegenden Seite lesen.

Sehen wir uns an, wie man den Widerstandswert eines 5-Band-Widerstands wie unten gezeigt abliest.



Für diesen Widerstand sollte der Widerstand von links nach rechts gelesen werden. Der Wert sollte in diesem Format vorliegen: 1. Band 2. Band 3. Band $\times 10^{\text{Multiplier}}$ und der zulässige Fehler beträgt $\pm \text{Toleranz}\%$. So ist der Widerstandswert dieses Widerstands 2(rot) 2(rot) 0(schwarz) $\times 10^0(\text{schwarz}) = 220$, und der zulässige Fehler beträgt $\pm 1\%$ (braun).

Mehr über Widerstände können Sie hier nachlesen: [Widerstand - Wikipedia](#).

1.5 Kondensator





Ein Kondensator bezeichnet die Menge an gespeicherter Ladung unter einem gegebenen Potenzialunterschied, gekennzeichnet durch C, wobei die internationale Einheit das Farad (F) ist. Allgemein gesprochen bewegen sich elektrische Ladungen unter einer Kraft in einem elektrischen Feld. Wenn zwischen den Leitern ein Medium vorhanden ist, wird die Bewegung der elektrischen Ladungen behindert und die Ladungen sammeln sich auf den Leitern an, was zu einer Ansammlung von elektrischen Ladungen führt.

Die Menge der gespeicherten elektrischen Ladungen wird Kapazität genannt. Da Kondensatoren zu den am häufigsten verwendeten elektronischen Bauteilen in elektronischen Geräten gehören, finden sie breite Anwendung in Gleichstrom-Isolation, Kopplung, Bypass, Filterung, Schwingkreisen, Energieumwandlung und Steuerungsschaltungen. Kondensatoren werden unterteilt in Elektrolytkondensatoren, Festkörperkondensatoren und so weiter.

Je nach Materialeigenschaften können Kondensatoren unterteilt werden in: Aluminium-Elektrolytkondensatoren, Folienkondensatoren, Tantal-Kondensatoren, Keramik-Kondensatoren, Superkondensatoren und so weiter.

In diesem Set werden Keramikkondensatoren und Elektrolytkondensatoren verwendet.

- [Keramikkondensator - Wikipedia](#)
- [Elektrolytkondensator - Wikipedia](#)

Auf den Keramikkondensatoren sind die Bezeichnungen 103 oder 104 zu finden, welche den Kapazitätswert repräsentieren, $103=10 \times 10^3 \text{pF}$, $104=10 \times 10^4 \text{pF}$.

Einheitenumrechnung

$$1\text{F}=10^3\text{mF}=10^6\text{uF}=10^9\text{nF}=10^{12}\text{pF}$$

1.6 Jumper-Kabel

Kabel, die zwei Anschlüsse verbinden, werden als Jumper-Kabel bezeichnet. Es gibt verschiedene Arten von Jumper-Kabeln. Hier konzentrieren wir uns auf diejenigen, die in Steckbrettern (Breadboards) verwendet werden. Unter anderem werden sie dazu verwendet, elektrische Signale von beliebigen Stellen auf dem Steckbrett zu den Ein-/Ausgangspins eines Mikrocontrollers zu übertragen.

Jumper-Kabel werden durch Einstecken ihrer „Endanschlüsse“ in die vorgesehenen Steckplätze des Steckbretts verbunden. Unter der Oberfläche des Steckbretts befinden sich einige parallele Platten, die die Steckplätze in Gruppen von Reihen oder Spalten verbinden, je nach Bereich. Die „Endanschlüsse“ werden in die Steckbrettschlitze eingeführt, ohne zu löten, in die speziellen Schlitze, die im spezifischen Prototyp verbunden werden müssen.

Es gibt drei Arten von Jumper-Kabeln: Female-to-Female (Weiblich zu Weiblich), Male-to-Male (Männlich zu Männlich) und Male-to-Female (Männlich zu Weiblich). Der Grund, warum wir es Male-to-Female nennen, liegt daran, dass es an einem Ende eine herausragende Spitze und an dem anderen ein versenktes weibliches Ende hat. Male-to-Male bedeutet, dass beide Seiten männlich sind, und Female-to-Female bedeutet, dass beide Enden weiblich sind.



In einem Projekt können mehrere Typen verwendet werden. Die Farbe der Jumper-Kabel kann unterschiedlich sein, dies bedeutet jedoch nicht, dass ihre Funktion entsprechend unterschiedlich ist. Sie sind lediglich so gestaltet, dass man die Verbindung zwischen den einzelnen Schaltungen besser erkennen kann.

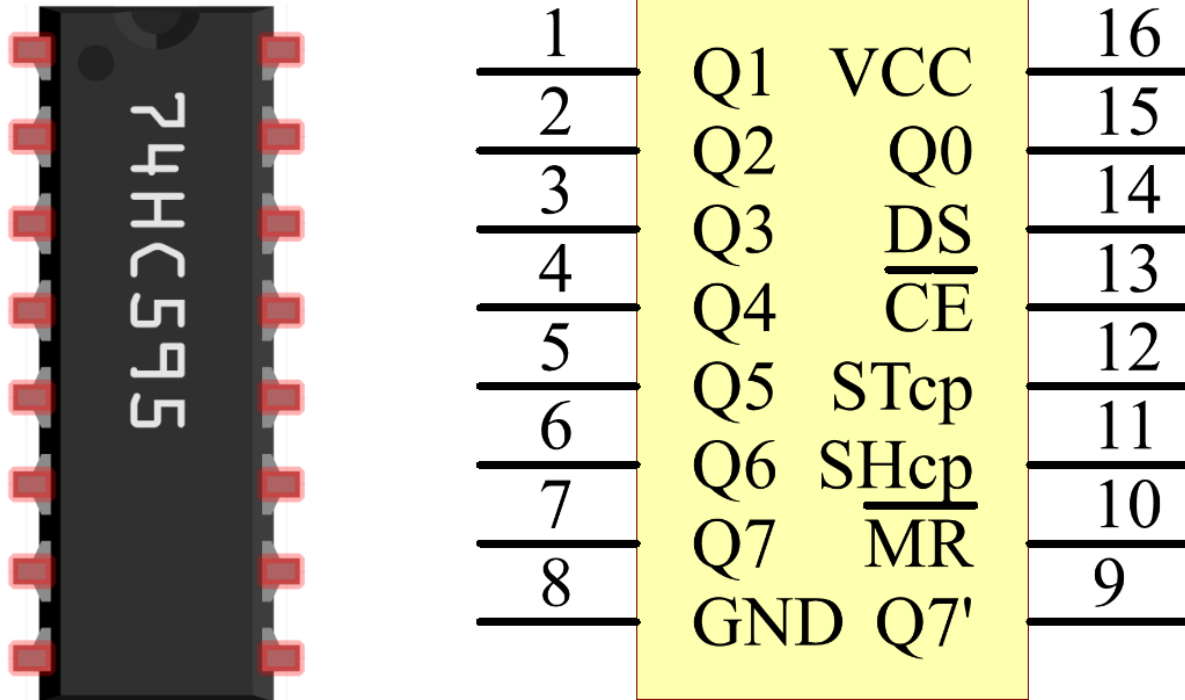
Chip

1.7 74HC595



Der 74HC595 besteht aus einem 8-Bit-Schieberegister und einem Speicherregister mit dreistufigen parallelen Ausgängen. Er wandelt serielle Eingaben in parallele Ausgaben um, sodass Sie IO-Ports eines MCU einsparen können. Wenn MR (Pin10) auf hohem Pegel und OE (Pin13) auf niedrigem Pegel ist, werden Daten am aufsteigenden Flanken von SHcp eingegeben und gelangen über den aufsteigenden Flanken von SHcp ins Speicherregister. Wenn die beiden Taktgeber verbunden sind, liegt das Schieberegister immer einen Puls vor dem Speicherregister. Es gibt einen seriellen Schieberegister-Eingang (Ds), einen seriellen Ausgang (Q) und einen asynchronen Reset-Taster (niedriger Pegel) im Speicherregister. Das Speicherregister gibt einen Bus mit parallelen 8 Bits und in drei Zuständen aus. Wenn OE aktiviert ist (niedriger Pegel), wird die Daten im Speicherregister auf den Bus ausgegeben.

- [74HC595 Datenblatt](#)



Pins des 74HC595 und ihre Funktionen:

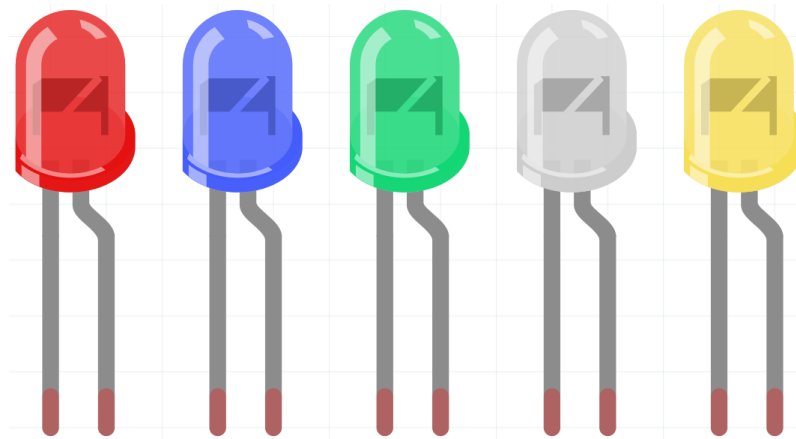
- **Q0-Q7**: 8-Bit-parallele Daten-Ausgangspins, können 8 LEDs oder 8 Pins einer 7-Segment-Anzeige direkt steuern.
- **Q7'**: Serieller Ausgangspin, verbunden mit DS eines anderen 74HC595, um mehrere 74HC595 in Serie zu verbinden.
- **MR**: Reset-Pin, aktiv auf niedrigem Pegel.
- **SHcp**: Zeitsequenzeingabe des Schieberegisters. Am aufsteigenden Rand bewegen sich die Daten im Schieberegister nacheinander um ein Bit, d.h. Daten in Q1 wechseln zu Q2 usw. Am fallenden Rand bleiben die Daten im Schieberegister unverändert.
- **STcp**: Zeitsequenzeingabe des Speicherregisters. Am aufsteigenden Rand wechseln die Daten im Schieberegister ins Speicherregister.
- **CE**: Ausgangsaktivierungspin, aktiv auf niedrigem Pegel.
- **DS**: Serieller Dateneingangspin.
- **VCC**: Positive Versorgungsspannung.
- **GND**: Erdung.

Beispiel

- [5.9 ShiftOut\(LED\)](#) (Grundlegendes Projekt)
- [5.10 ShiftOut\(Segmentanzeige\)](#) (Grundlegendes Projekt)
- [7. Strombegrenztes Tor](#) (IoT-Projekt)

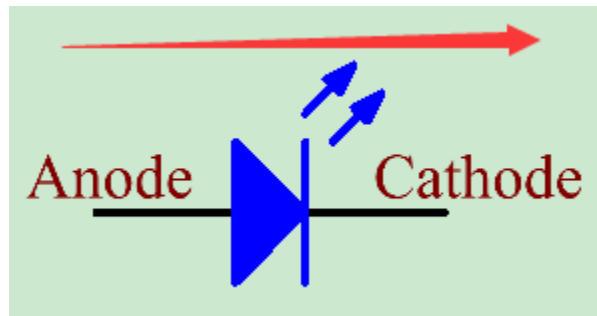
Anzeige

1.8 LED



Halbleiter-Leuchtdioden sind eine Art von Bauteil, das elektrische Energie durch PN-Übergänge in Lichtenergie umwandeln kann. Nach Wellenlänge unterteilt, gibt es Laserdioden, Infrarot-Leuchtdioden und sichtbare Leuchtdioden, die im Allgemeinen als Leuchtdioden (LED) bezeichnet werden.

Dioden haben eine unidirektionale Leitfähigkeit, daher fließt der Strom im Schaltungssymbol so, wie der Pfeil es anzeigt. Man sollte nur der Anode eine positive und der Kathode eine negative Spannung zuführen. Dadurch leuchtet die LED.



Eine LED hat zwei Pins. Der längere ist die Anode und der kürzere die Kathode. Achten Sie darauf, sie nicht verkehrt herum anzuschließen. In der LED gibt es eine feste Vorwärtsspannung, weshalb sie nicht direkt mit der Schaltung verbunden werden kann, da die Versorgungsspannung diesen Abfall übersteigen und die LED dadurch beschädigen kann. Die Vorwärtsspannung der roten, gelben und grünen LED beträgt 1,8 V und die der weißen LED 2,6 V. Die meisten LEDs können einen maximalen Strom von 20 mA verkraften, daher muss ein Strombegrenzungswiderstand in Reihe geschaltet werden.

Die Formel für den Widerstandswert lautet:

$$R = (V_{\text{supply}} - V_D) / I$$

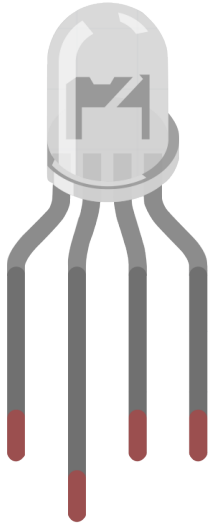
R steht für den Widerstandswert des Strombegrenzungswiderstands, **V_{supply}** für die Versorgungsspannung, **V_D** für den Spannungsabfall und **I** für den Betriebsstrom der LED.

Hier finden Sie eine detaillierte Einführung in die LED: [LED - Wikipedia](#).

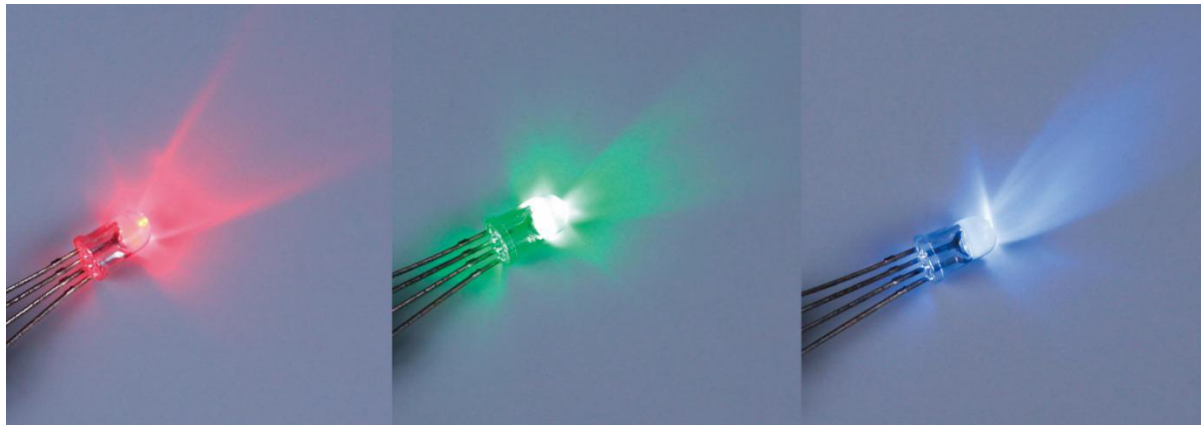
Beispiel

- [1.1 Hallo, LED!](#) (Grundlegendes Projekt)
- [2.1 Verblissen](#) (Grundlegendes Projekt)
- [2. Daten von Blynk abrufen](#) (IoT-Projekt)

1.9 RGB LED

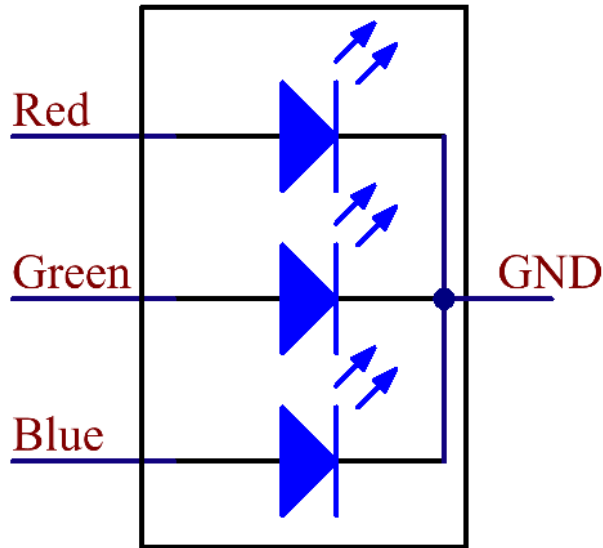


RGB-LEDs emittieren Licht in verschiedenen Farben. Eine RGB-LED kombiniert drei LEDs in den Farben Rot, Grün und Blau in einer transparenten oder halbtransparenten Kunststoffhülle. Durch das Ändern der Eingangsspannung der drei Pins und deren Überlagerung kann sie, statistischen Angaben zufolge, 16.777.216 verschiedene Farben erzeugen.

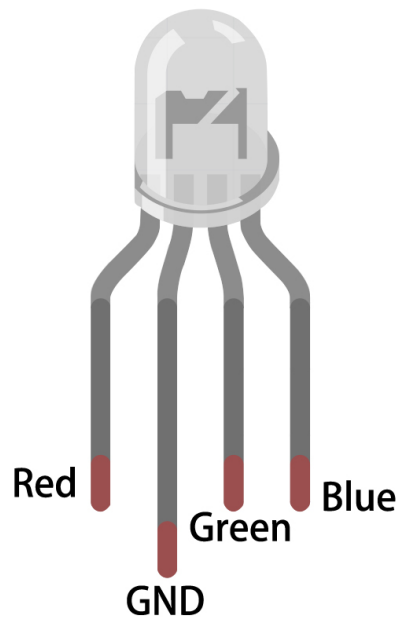


RGB-LEDs können in gemeinsame Anoden und gemeinsame Kathoden unterteilt werden. In diesem Bausatz wird die letztere verwendet. Die **gemeinsame Kathode**, oder GK, bedeutet, dass die Kathoden der drei LEDs verbunden sind. Wenn man sie mit GND verbindet und die drei Pins anschließt, wird die LED die entsprechende Farbe anzeigen.

Ihr Schaltungssymbol ist im folgenden Bild dargestellt.



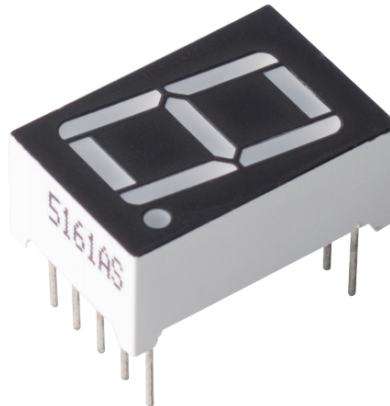
Eine RGB-LED hat 4 Pins: Der längste ist GND; die anderen sind Rot, Grün und Blau. Wenn man die Kunststoffhülle berührt, findet man eine Kerbe. Der dem Einschnitt am nächsten gelegene Pin ist der erste Pin, gekennzeichnet als Rot, dann GND, Grün und anschließend Blau.



Beispiel

- [2.2 Bunttes Licht](#) (Grundlegendes Projekt)
- [5.2 Schwellenwert](#) (Grundlegendes Projekt)

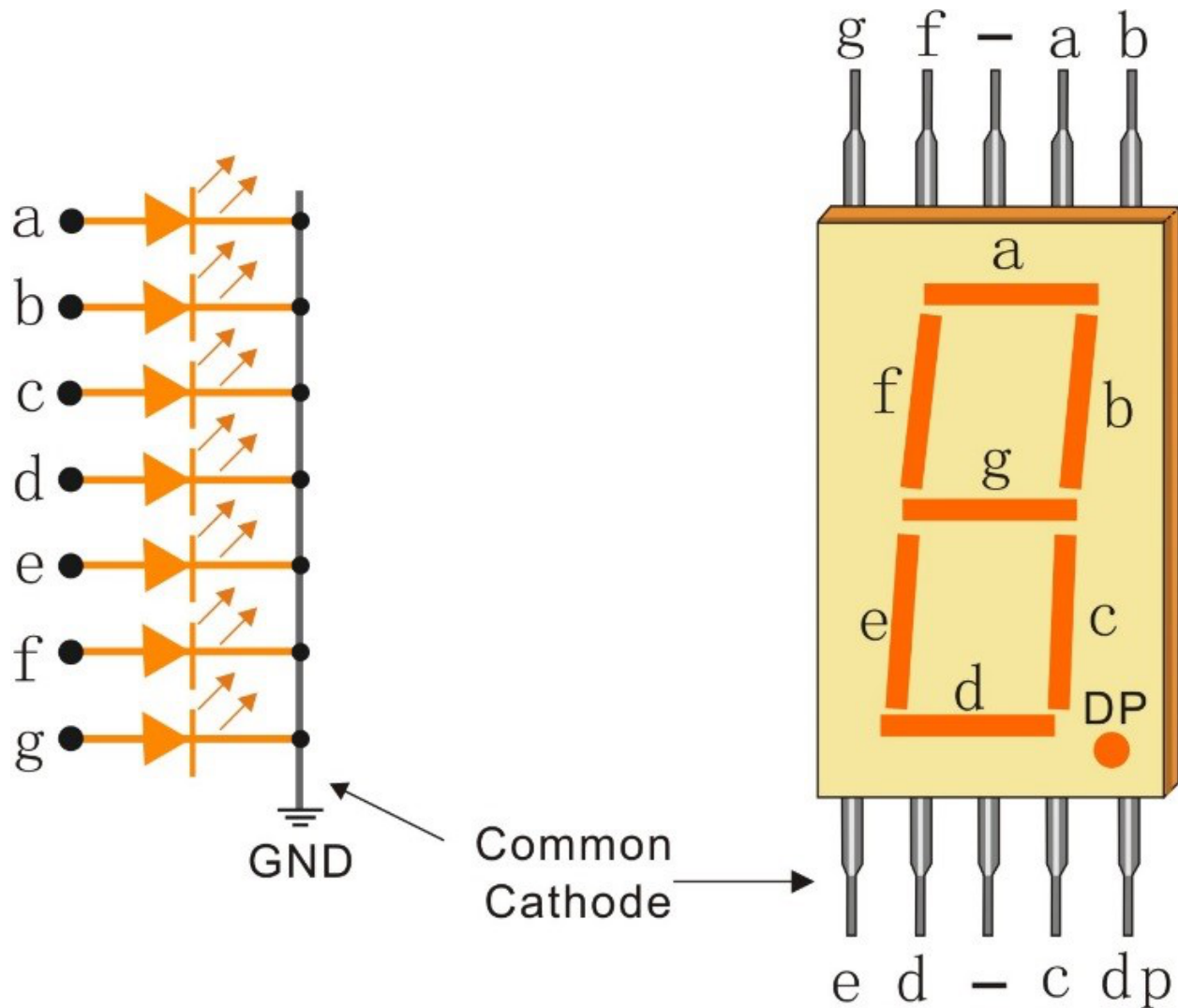
1.10 7-Segment-Anzeige



Eine 7-Segment-Anzeige ist eine Komponente in der Form einer 8, die aus 7 LEDs besteht. Jede LED wird als Segment bezeichnet - wenn sie mit Energie versorgt wird, bildet ein Segment einen Teil einer darzustellenden Zahl.

Es gibt zwei Arten von Pin-Verbindungen: Gemeinsame Kathode (CC) und Gemeinsame Anode (CA). Wie der Name schon sagt, hat eine CC-Anzeige alle Kathoden der 7 LEDs verbunden, während eine CA-Anzeige alle Anoden der 7 Segmente verbunden hat.

In diesem Bausatz verwenden wir die 7-Segment-Anzeige mit gemeinsamer Kathode, hier ist das elektronische Symbol dazu.



Jede LED in der Anzeige erhält ein positionelles Segment, von dem einer ihrer Anschlusspins aus dem rechteckigen Kunststoffgehäuse herausgeführt wird. Diese LED-Pins sind von „a“ bis „g“ beschriftet und repräsentieren jede einzelne LED. Die anderen LED-Pins sind miteinander verbunden und bilden einen gemeinsamen Pin. Durch Vorwärtsbiasierung der entsprechenden Pins der LED-Segmente in einer bestimmten Reihenfolge leuchten einige Segmente auf, während andere dunkel bleiben, wodurch der entsprechende Charakter auf der Anzeige dargestellt wird.

Anzeigecodes

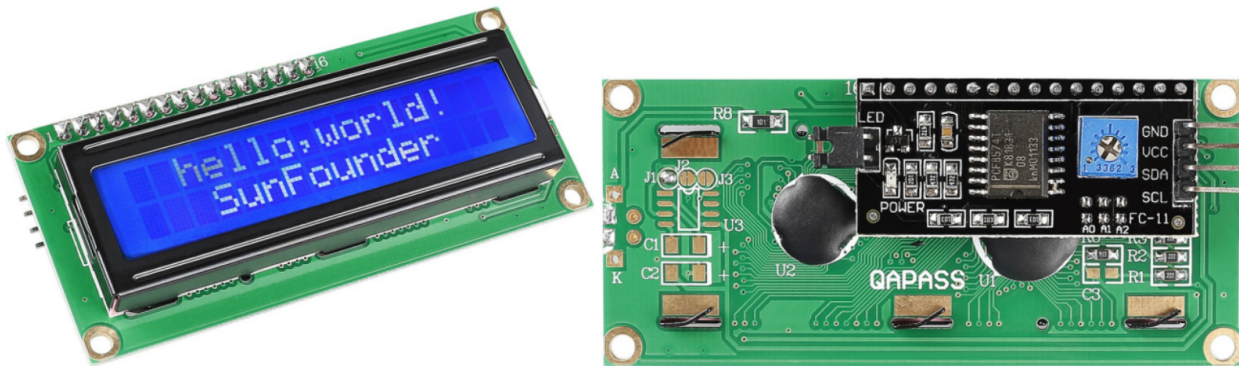
Um Ihnen zu helfen zu verstehen, wie 7-Segment-Anzeigen (Gemeinsame Kathode) Zahlen darstellen, haben wir die folgende Tabelle gezeichnet. Zahlen sind die Zahlen 0-F, die auf der 7-Segment-Anzeige dargestellt werden; (DP) GFEDCBA bezieht sich auf die entsprechende LED, die auf 0 oder 1 eingestellt ist. Zum Beispiel bedeutet 00111111, dass DP und G auf 0 eingestellt sind, während die anderen auf 1 eingestellt sind. Daher wird die Zahl 0 auf der 7-Segment-Anzeige angezeigt, während HEX-Code der entsprechenden Hexadezimalzahl entspricht.

Numbers	Common Cathode		Numbers	Common Cathode	
	(DP)GFEDCBA	Hex Code		(DP)GFEDCBA A	Hex Code
0	00111111	0x3f	A	01110111	0x77
1	00000110	0x06	B	01111100	0x7c
2	01011011	0x5b	C	00111001	0x39
3	01001111	0x4f	D	01011110	0x5e
4	01100110	0x66	E	01111001	0x79
5	01101101	0x6d	F	01110001	0x71
6	01111101	0x7d			
7	00000111	0x07			
8	01111111	0x7f			
9	01101111	0x6f			

Beispiel

- *5.15 EEPROM* (Grundlegendes Projekt)
- *7. Strombegrenztes Tor* (IoT-Projekt)

1.11 I2C LCD1602



- **GND:** Erdung
- **VCC:** Spannungsversorgung, 5V.
- **SDA:** Serielle Datenleitung. Verbinden Sie über einen Pull-up-Widerstand mit VCC.
- **SCL:** Serielle Taktleitung. Verbinden Sie über einen Pull-up-Widerstand mit VCC.

Wie wir alle wissen, bereichern LCDs und einige andere Displays die Mensch-Maschine-Interaktion erheblich. Sie haben jedoch eine gemeinsame Schwäche. Wenn sie an einen Controller angeschlossen sind, werden mehrere IOs des Controllers belegt, der nicht viele externe Ports hat. Dies beschränkt auch andere Funktionen des Controllers.

Deshalb wurde das LCD1602 mit einem I2C-Modul entwickelt, um dieses Problem zu lösen. Das I2C-Modul verfügt über einen eingebauten PCF8574 I2C-Chip, der I2C-Serien-Daten in parallele Daten für die LCD-Anzeige umwandelt.

- [PCF8574 Datenblatt](#)

I2C-Adresse

Die Standardadresse ist in der Regel 0x27, in einigen Fällen kann sie 0x3F sein.

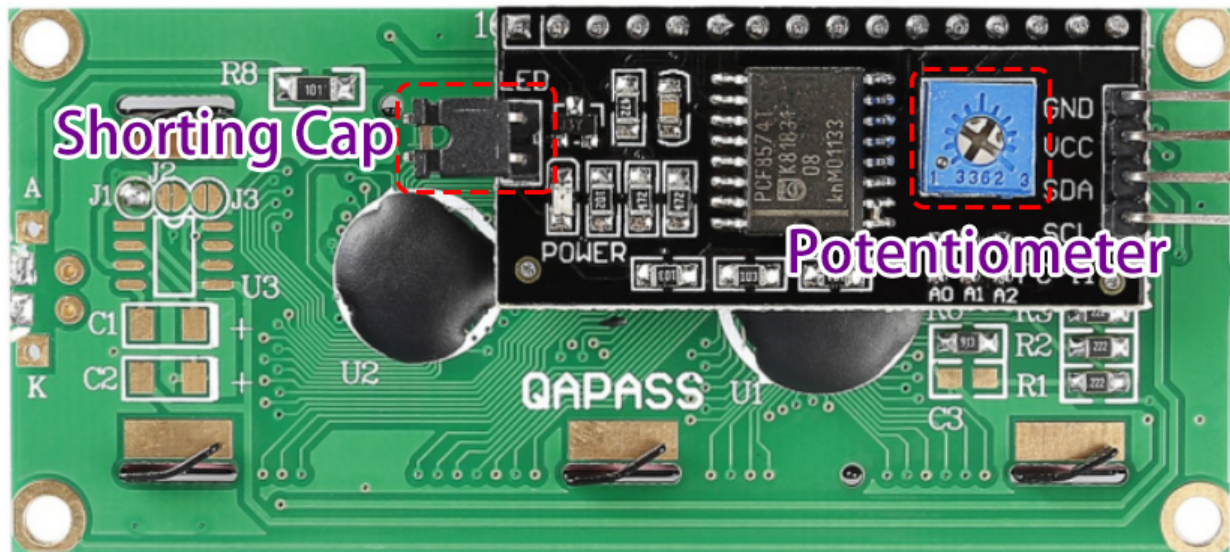
Nimmt man die Standardadresse von 0x27 als Beispiel, kann die Geräteadresse durch Kurzschließen der A0/A1/A2-Pads geändert werden; im Standardzustand sind A0/A1/A2 auf 1, und wenn das Pad kurzgeschlossen ist, sind A0/A1/A2 auf 0.

Slave Address

0	0	1	0	0	A2	A1	A0	
0	0	1	0	0	1	1	1	0x27
0	0	1	0	0	1	1	0	0x26
0	0	1	0	0	1	0	1	0x25
0	0	1	0	0	0	1	1	0x23
.....								
0	0	1	0	0	0	0	0	0x20

Hintergrundbeleuchtung/Kontrast

Die Hintergrundbeleuchtung kann durch das Aufstecken eines Jumper-Caps aktiviert werden, durch das Abziehen wird sie deaktiviert. Das blaue Potentiometer auf der Rückseite dient zur Kontrasteinstellung (Verhältnis der Helligkeit zwischen dem hellsten Weiß und dem dunkelsten Schwarz).



- **Kurzschluss-Cap:** Mit diesem Cap kann die Hintergrundbeleuchtung aktiviert werden, ziehen Sie es ab, um die Beleuchtung zu deaktivieren.

- **Potentiometer:** Es wird verwendet, um den Kontrast (Klarheit des angezeigten Textes) einzustellen, der im Uhrzeigersinn erhöht und gegen den Uhrzeigersinn verringert wird.

Beispiel

- [5.11.1 Flüssigkristallanzeige](#) (Grundlegendes Projekt)
- [5.12 Serielles Lesen](#) (Grundlegendes Projekt)

Ton

1.12 Summer



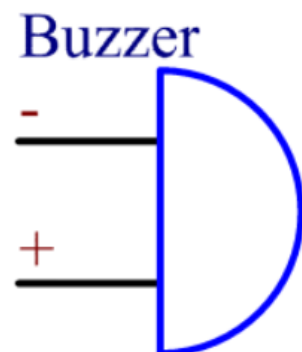
Als eine Art von elektronischem Summer mit integrierter Struktur, die mit Gleichstrom versorgt werden, werden Summer häufig in Computern, Druckern, Fotokopierern, Alarmen, elektronischen Spielzeugen, automobilen Elektronikgeräten, Telefonen, Timern und anderen elektronischen Produkten oder Sprachgeräten verwendet.

Man kann Summer in aktive und passive unterteilen (siehe folgendes Bild). Wenn man den Summer so dreht, dass seine Pins nach oben zeigen, dann ist der Summer mit einer grünen Platine ein passiver Summer, während der mit schwarzem Klebeband umwickelte ein aktiver Summer ist.

Der Unterschied zwischen einem aktiven und einem passiven Summer:

Ein aktiver Summer hat eine eingebaute Oszillationsquelle, daher gibt er Töne von sich, wenn er elektrifiziert wird. Ein passiver Summer hat jedoch keine solche Quelle und gibt daher keinen Ton von sich, wenn Gleichstromsignale verwendet werden. Stattdessen müssen Sie ihn mit Rechteckwellen antreiben, deren Frequenz zwischen 2K und 5K liegt. Der aktive Summer ist oft teurer als der passive aufgrund der eingebauten Oszillationskreise.

Das Folgende ist das elektrische Symbol eines Summers. Er hat zwei Pins, einen positiven und einen negativen Pol. Ein + auf der Oberfläche stellt die Anode dar, und der andere ist die Kathode.



Sie können die Pins des Summers überprüfen: der längere ist die Anode und der kürzere die Kathode. Bitte verwechseln Sie sie nicht beim Anschließen, sonst gibt der Summer keinen Ton von sich.

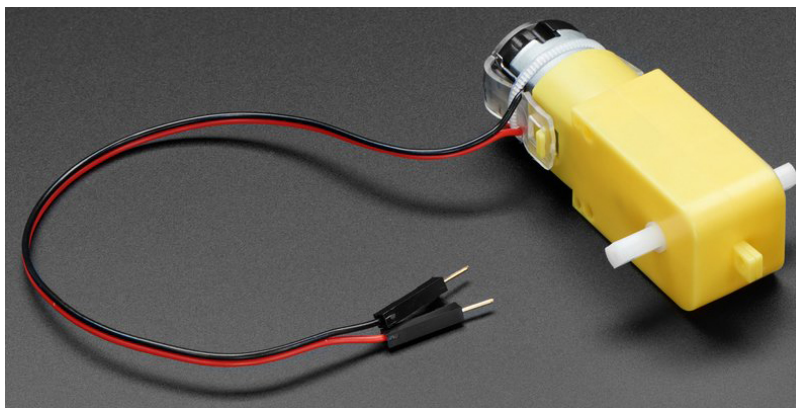
[Summer - Wikipedia](#)

Beispiel

- [1.2 Piepton](#) (Grundlegendes Projekt)
- [5.7 Tone\(\) oder noTone\(\)](#) (Grundlegendes Projekt)
- [4. Cloud-Musikspieler](#) (IoT-Projekt)

Treiber

1.13 TT Motor



Dies ist ein TT Gleichstrom-Getriebemotor mit einem Übersetzungsverhältnis von 1:48. Er wird mit 2 x 200mm Kabeln mit 0,1" Steckern geliefert, die in ein Breadboard passen. Ideal für den Einsatz in einem Breadboard oder Klemmblock.

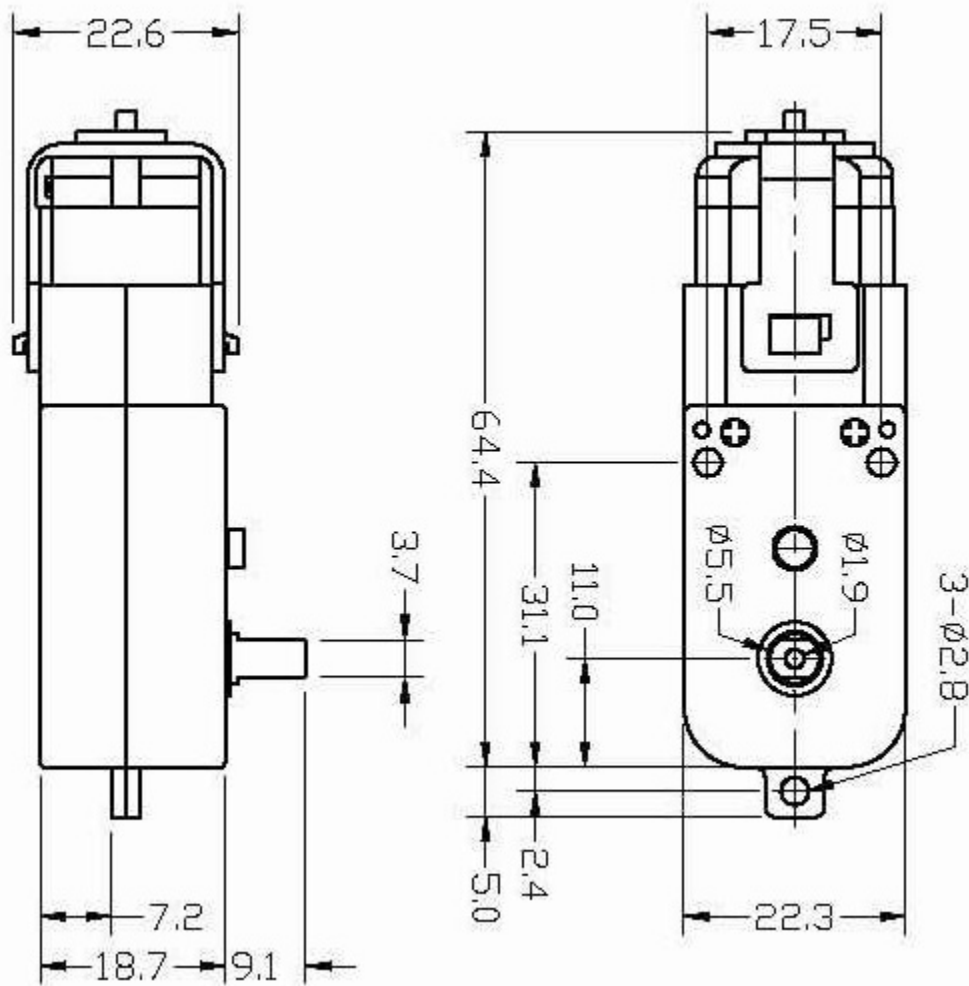
Diese Motoren können mit einer Spannung von 3 ~ 6VDC betrieben werden. Bei höheren Spannungen werden sie natürlich etwas schneller laufen.

Beachten Sie, dass dies sehr einfache Motoren sind, die über keinen eingebauten Encoder, Geschwindigkeitsregelung oder Positionsrückmeldung verfügen. Die Spannung wird zugeführt und die Drehung erfolgt. Es wird Variationen von Motor zu Motor geben. Wenn Sie eine präzise Bewegung benötigen, benötigen Sie ein separates Feedback-System.

Technische Details

- Nennspannung: 3~6V
- Dauerhafter Leerlaufstrom: 150mA +/- 10%
- Min. Betriebsgeschwindigkeit (3V): 90+/- 10% U/min
- Min. Betriebsgeschwindigkeit (6V): 200+/- 10% U/min
- Anhaltemoment (3V): 0,4kg.cm
- Anhaltemoment (6V): 0,8kg.cm
- Übersetzungsverhältnis: 1:48
- Abmessungen des Gehäuses: 70 x 22 x 18mm
- Kabellänge: 200mm & 28 AWG
- Gewicht: 30,6g

Abmessungszeichnung



Beispiel

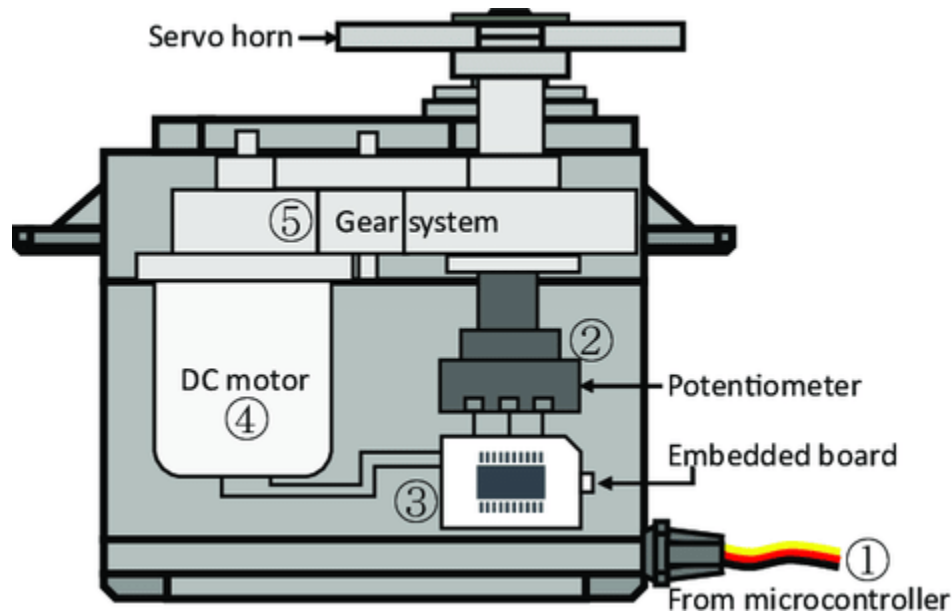
- [1.3 Das Rad drehen](#) (Grundlegendes Projekt)
- [1. Bewegung](#) (Auto-Projekt)
- [3. Beschleunigung](#) (Auto-Projekt)
- [8. IoT-Auto](#) (IoT-Projekt)

1.14 Servo

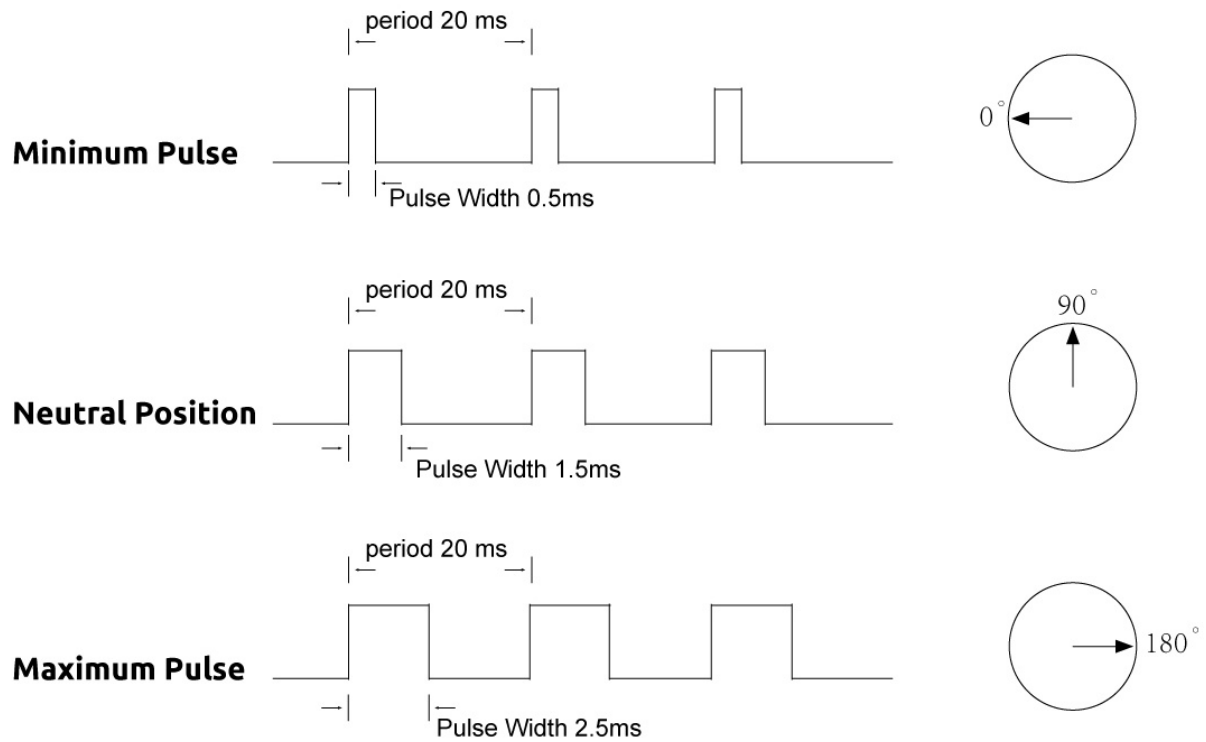


Ein Servo besteht in der Regel aus folgenden Teilen: Gehäuse, Welle, Getriebesystem, Potentiometer, Gleichstrommotor und eingebettetem Board.

So funktioniert es: Der Mikrocontroller sendet PWM-Signale an den Servo. Das eingebettete Board im Servo empfängt diese Signale über den Signalpin und steuert den darin befindlichen Motor an. Dadurch treibt der Motor das Getriebesystem an, welches nach einer Verzögerung die Welle antreibt. Die Welle und das Potentiometer des Servos sind miteinander verbunden. Wenn sich die Welle dreht, treibt sie das Potentiometer an. Das Potentiometer gibt ein Spannungssignal an das eingebettete Board weiter. Das Board bestimmt dann auf Basis der aktuellen Position die Drehrichtung und -geschwindigkeit, sodass es genau an der definierten Position stoppen und dort verharren kann.



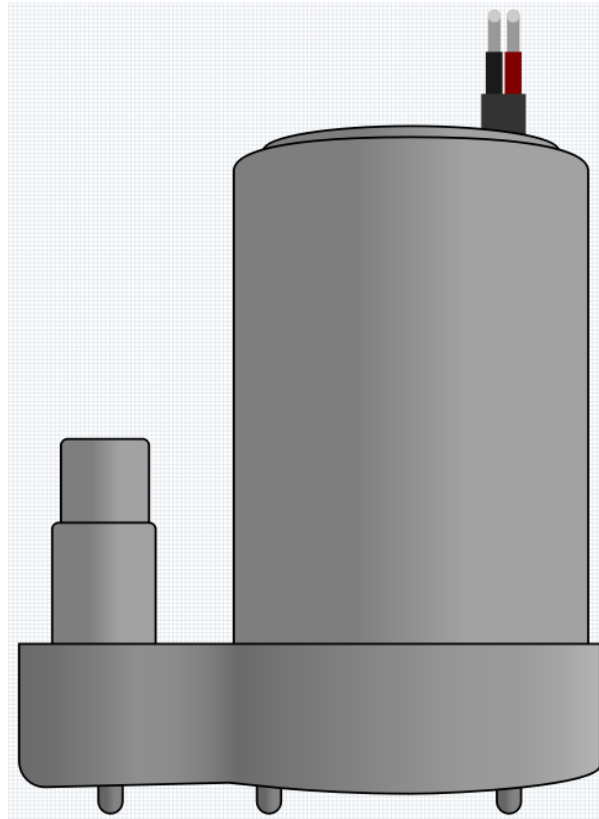
Der Winkel wird durch die Dauer eines Impulses bestimmt, der auf das Steuerkabel gegeben wird. Dies wird als Pulsweitenmodulation bezeichnet. Das Servo erwartet alle 20 ms einen Impuls. Die Länge des Impulses bestimmt, wie weit der Motor dreht. Ein 1,5ms-Impuls lässt den Motor beispielsweise auf die 90-Grad-Position (Neutralstellung) drehen. Wenn ein Impuls, der kürzer als 1,5 ms ist, an einen Servo gesendet wird, dreht sich dieser zu einer Position und hält seine Ausgangswelle einige Grade gegen den Uhrzeigersinn vom Neutralpunkt aus. Bei einem Impuls, der breiter als 1,5 ms ist, tritt das Gegenteil auf. Die minimale Impulsbreite und die maximale Impulsbreite, die den Servo zu einer gültigen Position drehen lassen, sind Funktionen jedes Servos. Üblicherweise liegt die Mindestimpulsbreite bei etwa 0,5 ms und die Maximalimpulsbreite bei 2,5 ms.



Beispiel

- [5.5 Interne Bibliothek nutzen](#) (Grundlegendes Projekt)
- [7. Strombegrenztes Tor](#) (IoT-Projekt)

1.15 Kreiselpumpe



Die Kreiselpumpe wandelt Rotationskinetik in hydrodynamische Energie um, um Flüssigkeiten zu transportieren. Die Rotationsenergie stammt aus dem Elektromotor. Die Flüssigkeit tritt entlang oder in der Nähe der rotierenden Welle in das Pumpenlaufrad ein, wird vom Laufrad beschleunigt, fließt radial nach außen in die Diffusor- oder Spiralgehäusekammer und strömt von dort aus weiter.

Typische Anwendungen von Kreiselpumpen sind das Pumpen von Wasser, Abwasser, in der Landwirtschaft, für Erdöl und Petrochemie.

- [Kreiselpumpe - Wikipedia](#)

Merkmale

- **Spannungsbereich:** DC 3 ~ 4,5V
- **Betriebsstrom:** 120 ~ 180mA
- **Leistung:** 0,36 ~ 0,91W
- **Max. Förderhöhe:** 0,35 ~ 0,55M
- **Max. Durchflussrate:** 80 ~ 100 L/H
- **Dauerbetriebszeit:** 100 Stunden
- **Wasserschutzklasse:** IP68
- **Antriebsart:** DC, magnetischer Antrieb
- **Material:** Technischer Kunststoff
- **Außendurchmesser des Auslasses:** 7,8 mm

- **Innendurchmesser des Auslasses:** 6,5 mm
- Es handelt sich um eine Tauchpumpe, die auch so verwendet werden sollte. Bei oberirdischem Betrieb kann sie sich so stark erhitzen, dass Überhitzungsgefahr besteht.

Beispiel

- *1.4 Pumpen* (Grundlegendes Projekt)
- *6. Pflanzenüberwachung* (IoT-Projekt)

1.16 L9110 Motortreibermodul

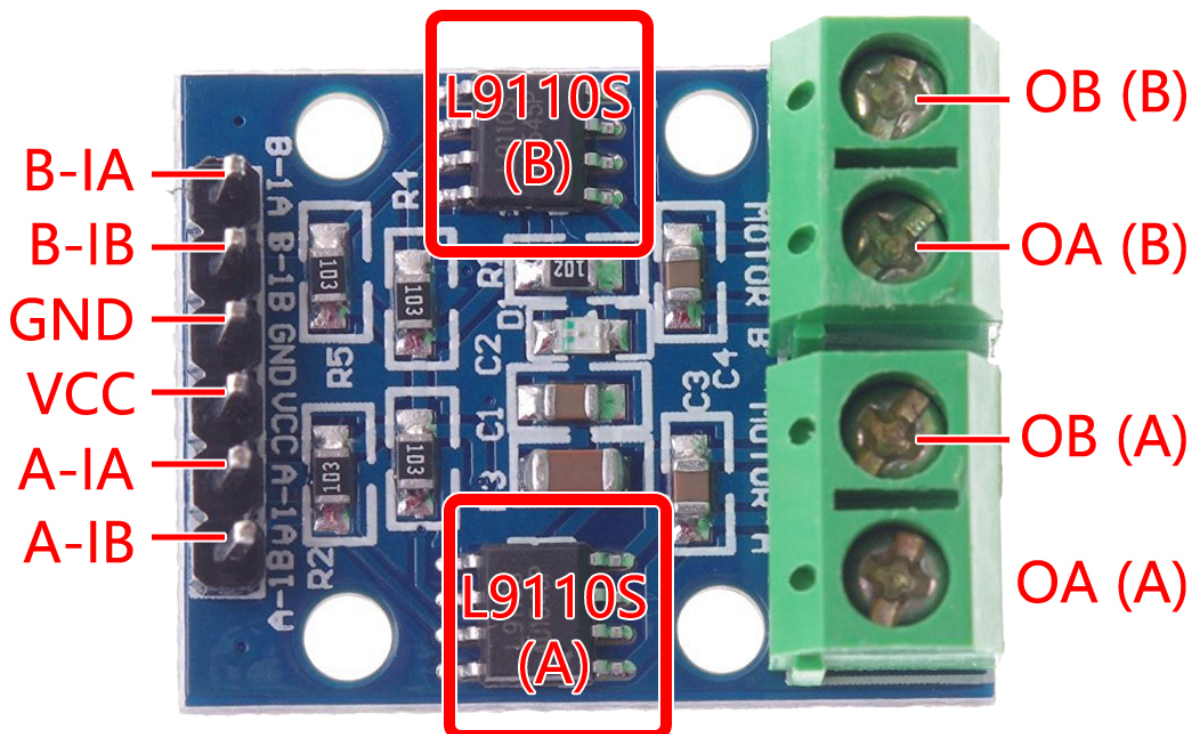
Das L9110 Motortreibermodul eignet sich hervorragend zum Ansteuern von zwei Motoren gleichzeitig. Es beherbergt ein Paar unabhängiger L9110S-Treiberchips, wobei jeder Kanal einen konstanten Stromausgang von bis zu 800mA bietet.

Mit einem Spannungsbereich von 2,5V bis 12V harmonisiert das Modul problemlos sowohl mit 3,3V als auch mit 5V Mikrocontrollern.

Als kompakte Lösung erleichtert das L9110 Motortreibermodul die Motorsteuerung in einer Vielzahl von Anwendungen. Dank seiner Dual-Kanal-Architektur ermöglicht es die unabhängige Steuerung von zwei Motoren – ideal für Projekte, bei denen der Betrieb von zwei Motoren im Vordergrund steht.

Angesichts seiner starken Dauerstromausgabe versorgt dieses Modul sicher Motoren von klein bis mittelgroß und ebnet den Weg für vielfältige robotische, automatisierungsbezogene und motorzentrierte Vorhaben. Sein weitreichender Spannungsbereich erhöht die Anpassungsfähigkeit und passt sich verschiedenen Stromversorgungskonzepten an.

Mit Blick auf Benutzerfreundlichkeit konzipiert, bietet das Modul intuitive Ein- und Ausgabeterminals, die den Anschluss an Mikrocontroller oder ähnliche Steuergeräte vereinfachen. Zudem wurde nicht an Sicherheit gespart – integrierte Überstrom- und Übertemperaturschutzmaßnahmen erhöhen die Zuverlässigkeit und Sicherheit des Motorbetriebs.



- **B-1A & B-1B(B-2A)**: Eingangspins zur Steuerung der Drehrichtung von Motor B.
- **A-1A & A-1B**: Eingangspins zur Steuerung der Drehrichtung von Motor A.
- **0A & 0B(A)**: Ausgangspins von Motor A.
- **0A & 0B(B)**: Ausgangspins von Motor B.
- **VCC**: Stromversorgungspin (2,5V-12V).
- **GND**: Massepin.

Merkmale

- On-Board 2 L9110S Motorsteuerchips
- Dual-Kanal-Motorsteuerung.
- Unabhängige Steuerung der Motordrehrichtung.
- Hoher Stromausgang (800mA pro Kanal).
- Weiter Spannungsbereich (2,5V-12V).
- Kompaktes Design.
- Praktische Ein- und Ausgabeterminals.
- Integrierte Schutzfunktionen.
- Vielseitige Anwendungsmöglichkeiten.
- PCB-Größe: 29,2mm x 23mm
- Betriebstemperatur: -20°C ~ 80°C
- LED-Anzeige bei eingeschaltetem Zustand

Funktionsprinzip

Hier ist die Wahrheitstabelle für Motor B:

Diese Wahrheitstabelle zeigt die verschiedenen Zustände von Motor B basierend auf den Werten der Eingangspins B-1A und B-1B(B-2A). Sie gibt die Drehrichtung (im Uhrzeigersinn oder gegen den Uhrzeigersinn), Bremsen oder Anhalten von Motor B an.

B-1A	B-1B(B-2A)	Zustand von Motor B
1	0	Drehen im Uhrzeigersinn
0	1	Drehen gegen den Uhrzeigersinn
0	0	Bremsen
1	1	Anhalten

Hier ist die Wahrheitstabelle für Motor A:

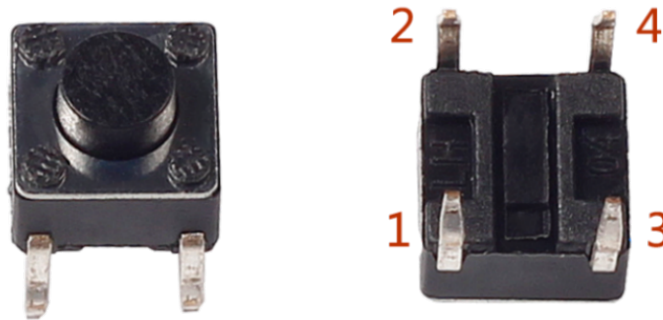
Diese Wahrheitstabelle zeigt die verschiedenen Zustände von Motor A basierend auf den Werten der Eingangspins A-1A und A-1B. Sie gibt die Drehrichtung (im Uhrzeigersinn oder gegen den Uhrzeigersinn), Bremsen oder Anhalten von Motor A an.

A-1A	A-1B	Zustand von Motor A
1	0	Drehen im Uhrzeigersinn
0	1	Drehen gegen den Uhrzeigersinn
0	0	Bremsen
1	1	Anhalten

- *1.3 Das Rad drehen* (Grundprojekt)
- *1. Bewegung* (Auto-Projekt)
- *3. Beschleunigung* (Auto-Projekt)
- *8. IoT-Auto* (IoT-Projekt)

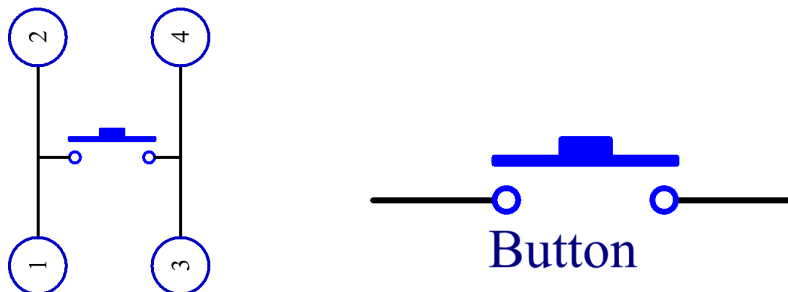
Controller

1.17 Knopf

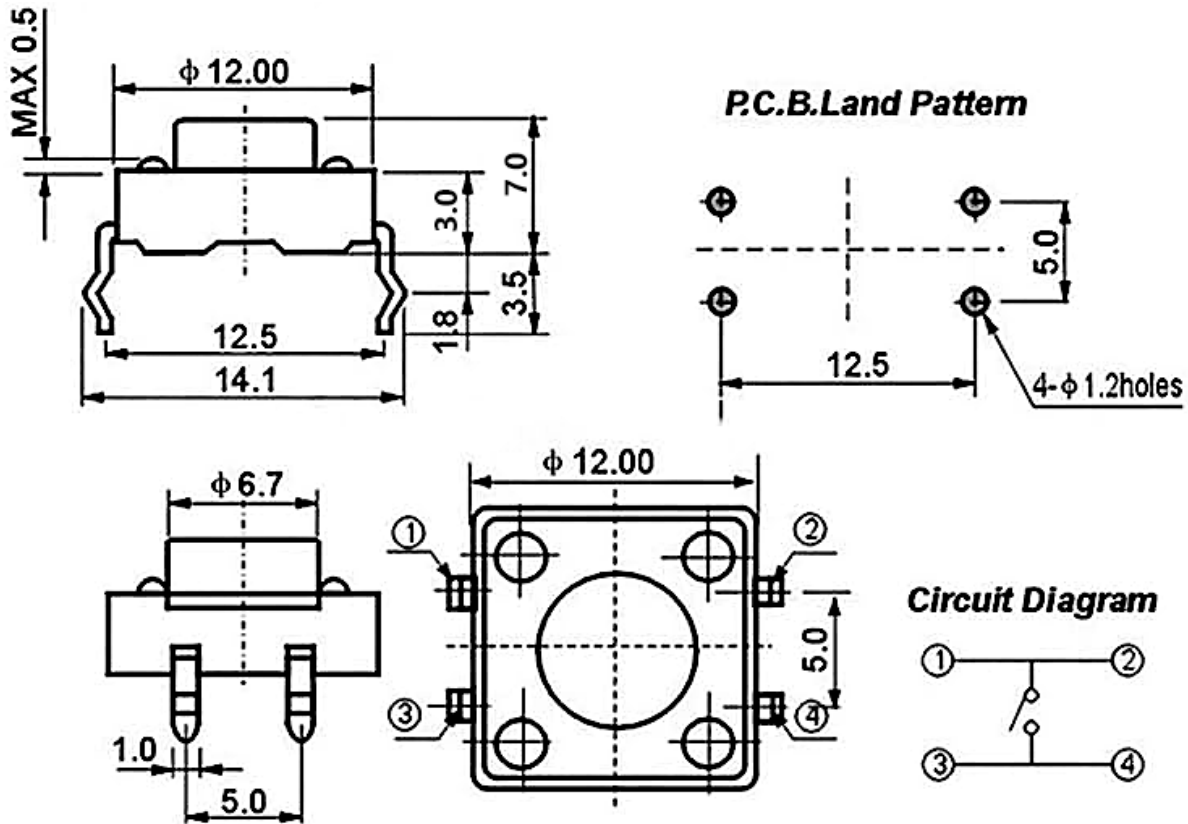


Knöpfe sind eine gängige Komponente zur Steuerung von elektronischen Geräten. Sie werden üblicherweise als Schalter verwendet, um Schaltkreise zu verbinden oder zu unterbrechen. Obwohl es Knöpfe in verschiedenen Größen und Formen gibt, handelt es sich bei dem hier verwendeten um einen 6mm Mini-Knopf, wie auf den folgenden Bildern gezeigt. Pin 1 ist mit Pin 2 und Pin 3 mit Pin 4 verbunden. Sie müssen also nur entweder Pin 1 oder Pin 2 mit Pin 3 oder Pin 4 verbinden.

Das Folgende zeigt den inneren Aufbau eines Knopfes. Das Symbol auf der rechten Seite unten wird normalerweise verwendet, um einen Knopf in Schaltkreisen darzustellen.



Da Pin 1 mit Pin 2 und Pin 3 mit Pin 4 verbunden ist, werden beim Drücken des Knopfes die 4 Pins verbunden, wodurch der Schaltkreis geschlossen wird.



Beispiel

- [3.1 Tastenwert auslesen](#) (Grundlegendes Projekt)

1.18 Reedschalter

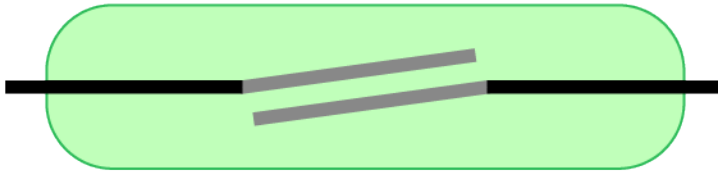


Der Reedschalter ist ein elektrischer Schalter, der mittels eines angelegten Magnetfeldes funktioniert. Er wurde 1936 von Walter B. Ellwood von den Bell Telephone Laboratories erfunden und am 27. Juni 1940 in den USA unter der Patentnummer 2264746 patentiert.

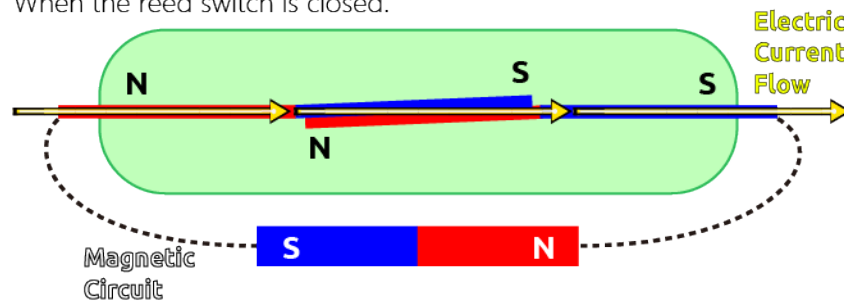
Das Funktionsprinzip eines Reedschalters ist sehr einfach. Zwei Reed-Kontakte (normalerweise aus Eisen und Nickel gefertigt, zwei Metallen), die an den Endpunkten überlappen, sind in einem Glasröhrchen versiegelt. Die beiden Reeds überlappen und sind durch eine kleine Lücke (nur einige Mikrometer) voneinander getrennt. Das Glasröhrchen ist mit einem hochreinen Inertgas (wie Stickstoff) gefüllt, und einige Reedschalter werden so hergestellt, dass sie im Inneren ein Vakuum aufweisen, um ihre Hochspannungsleistung zu verbessern.

Das Reed fungiert als Magnetflussleiter. Die beiden Reeds sind im Ruhezustand nicht in Kontakt; wenn sie durch ein von einem Permanentmagneten oder einer elektromagnetischen Spule erzeugtes Magnetfeld hindurchgehen, führt das angelegte Magnetfeld dazu, dass die beiden Reeds an ihren Endpunkten unterschiedliche Polaritäten aufweisen. Wenn die magnetische Kraft die Federkraft der Reeds selbst übersteigt, werden die beiden Reeds zusammengezogen, um den Schaltkreis zu leiten. Wenn das Magnetfeld schwächer wird oder verschwindet, werden die Reeds aufgrund ihrer eigenen Elastizität freigegeben und die Kontaktflächen trennen sich, um den Schaltkreis zu öffnen.

When the reed switch is open.



When the reed switch is closed.

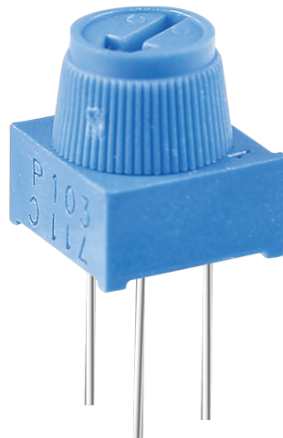


- [Reedschalter – Wikipedia](#)

Beispiel

- *3.2 Magnetismus spüren* (Grundlegendes Projekt)
- *7. Strombegrenztes Tor* (IoT-Projekt)

1.19 Potentiometer



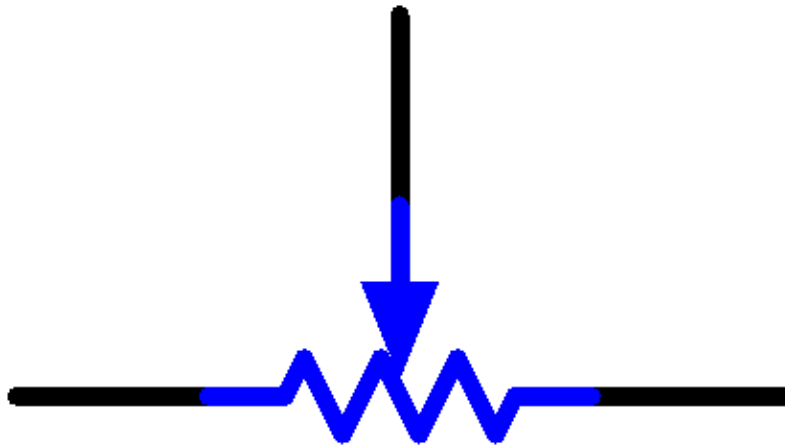
Ein Potentiometer ist ebenfalls ein Widerstandsbauelement mit drei Anschlüssen, dessen Widerstandswert nach einer bestimmten Regelung variiert werden kann.

Potentiometer gibt es in verschiedenen Formen, Größen und Werten, sie haben jedoch alle folgende Merkmale gemeinsam:

- Sie haben drei Anschlüsse (oder Verbindungspunkte).
- Sie verfügen über einen Drehknopf, eine Schraube oder einen Schieber, um den Widerstand zwischen dem mittleren und einem der äußeren Anschlüsse zu variieren.

- Der Widerstand zwischen dem mittleren Anschluss und einem der äußeren Anschlüsse variiert von 0 bis zum maximalen Widerstand des Potentiometers, wenn der Knopf, die Schraube oder der Schieber bewegt wird.

Hier ist das Schaltsymbol des Potentiometers.



Die Funktionen des Potentiometers im Schaltkreis sind:

1. Als Spannungsteiler dienen:

Ein Potentiometer ist ein kontinuierlich einstellbarer Widerstand. Beim Verstellen der Welle oder des Schiebegriffs des Potentiometers gleitet der bewegliche Kontakt über den Widerstand. Zu diesem Zeitpunkt kann, abhängig von der am Potentiometer angelegten Spannung und dem Winkel, um den sich der bewegliche Arm gedreht hat oder der zurückgelegten Strecke, eine Spannung ausgegeben werden.

2. Als Rheostat dienen:

Wenn das Potentiometer als Rheostat verwendet wird, verbinden Sie den mittleren Pin und einen der anderen beiden Pins im Schaltkreis. So erhalten Sie einen sanft und kontinuierlich veränderten Widerstandswert innerhalb des Fahrwegs des beweglichen Kontakts.

3. Als Stromregler dienen:

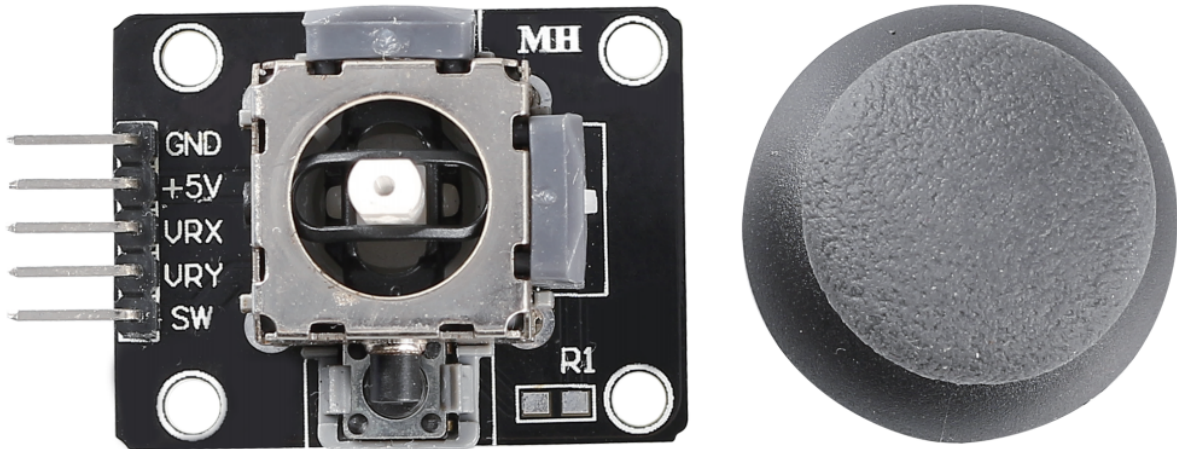
Wenn das Potentiometer als Stromregler fungiert, muss der Schiebekontaktanschluss als einer der Ausgangsanschlüsse verbunden werden.

Wenn Sie mehr über Potentiometer erfahren möchten, verweisen Sie auf: [Potentiometer – Wikipedia](#)

Beispiel

- [4.1 Den Knopf drehen](#) (Grundlegendes Projekt)

1.20 Joystick Modul

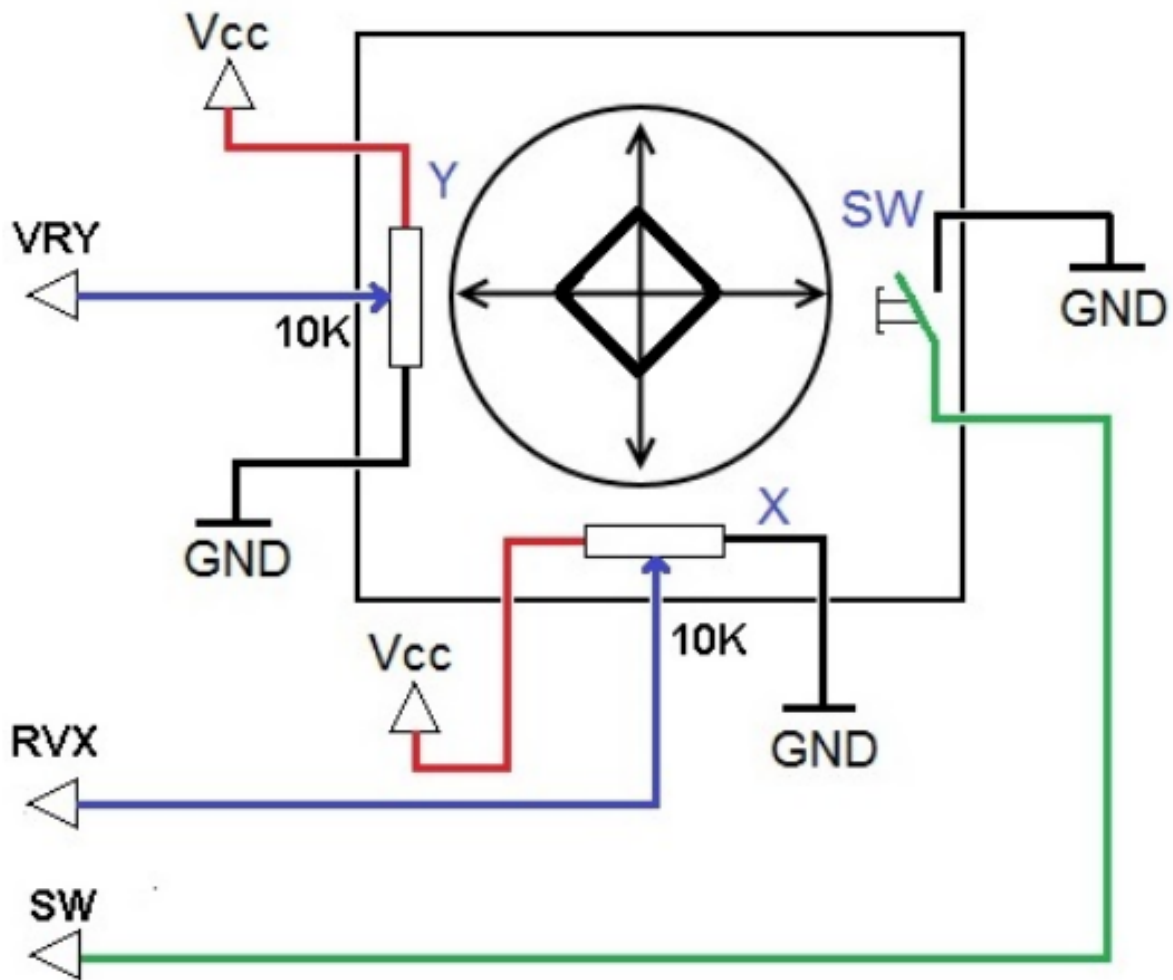


Das Grundkonzept eines Joysticks besteht darin, die Bewegung eines Sticks in elektronische Informationen umzusetzen, die ein Computer verarbeiten kann.

Um dem Computer eine vollständige Bewegungspalette zu übermitteln, muss ein Joystick die Position des Sticks auf zwei Achsen messen - der X-Achse (von links nach rechts) und der Y-Achse (von oben nach unten). Wie in der Grundgeometrie geben die X-Y-Koordinaten die genaue Position des Sticks an.

Um den Standort des Sticks zu bestimmen, überwacht das Joystick-Steuerungssystem einfach die Position jeder Welle. Das herkömmliche analoge Joystick-Design macht dies mit zwei Potentiometern oder variablen Widerständen.

Der Joystick verfügt auch über einen digitalen Eingang, der betätigt wird, wenn der Joystick nach unten gedrückt wird.

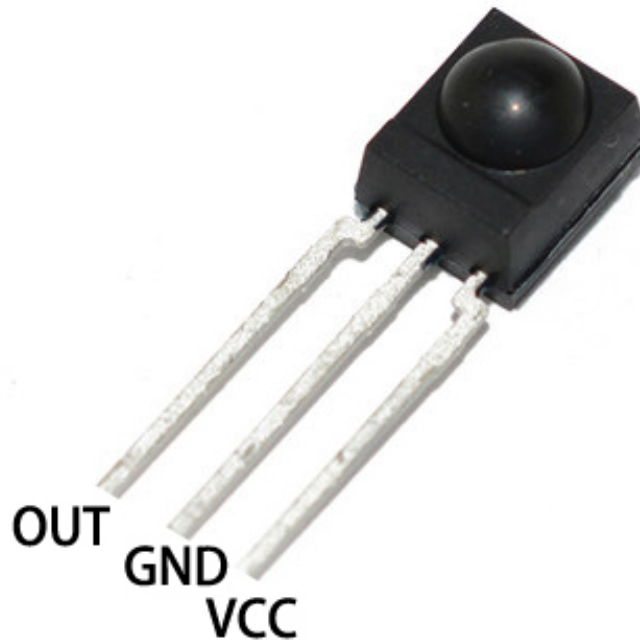


Beispiel

- [4.3 Den Joystick betätigen](#) (Grundlegendes Projekt)

1.21 IR-Empfänger

IR-Empfänger



- OUT: Signal-Ausgang
- GNDGND
- VCC: Stromversorgung, 3.3v~5V

Ein Infrarot-Empfänger ist eine Komponente, die Infrarotsignale empfängt und eigenständig Infrarotstrahlen empfangen sowie Signale ausgeben kann, die mit TTL-Pegel kompatibel sind. Er ist in Größe und Form einem gewöhnlichen Transistor im Plastikgehäuse ähnlich und eignet sich für alle Arten von Infrarot-Fernbedienungen und Infrarot-Übertragungen.

Infrarot, oder kurz IR, ist eine beliebte, kostengünstige und einfach zu verwendende drahtlose Kommunikationstechnologie. Infrarotlicht hat eine etwas längere Wellenlänge als sichtbares Licht und ist daher für das menschliche Auge nicht wahrnehmbar - ideal für die drahtlose Kommunikation. Ein gängiges Modulationsschema für Infrarotkommunikation ist die 38KHz-Modulation.

- Verwendet den [HS0038B](#) IR-Empfänger-Sensor mit hoher Empfindlichkeit
- Kann für Fernbedienungen genutzt werden
- Stromversorgung: 5V
- Schnittstelle: Digital
- Modulationsfrequenz: 38Khz
- Pin-Belegung: (1) Ausgang (2) Vcc (3) GND
- Größe: 23.5mm x 21.5mm

Fernbedienung



Hierbei handelt es sich um eine schlanke Infrarot-Fernbedienung mit 21 Funktionstasten und einer Übertragungsreichweite von bis zu 8 Metern, die sich ideal für den Einsatz in einem Kinderzimmer eignet.

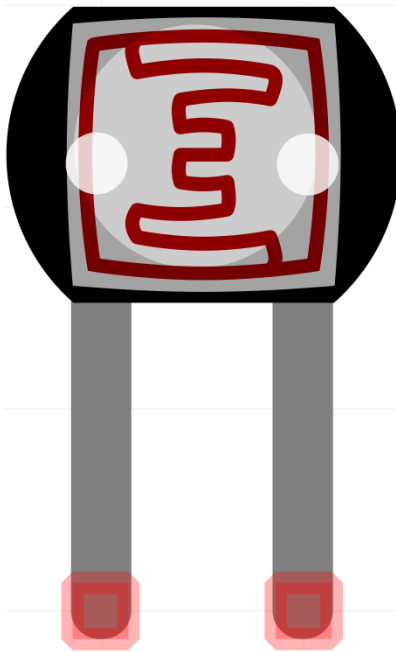
- Größe: 85x39x6mm
- Reichweite der Fernbedienung: 8-10m
- Batterie: 3V Knopfzellen-Lithium-Mangan-Batterie
- Infrarot-Trägerfrequenz: 38KHz
- Oberflächenklebematerial: 0.125mm PET
- Effektive Lebensdauer: mehr als 20.000 Betätigungen

Beispiel

- [5.11.2 IR-Empfänger](#) (Grundlegendes Projekt)
- [9. Fernbedienung](#) (Auto-Projekt)
- [10. Starten mit einem Tastendruck](#) (Auto-Projekt)

Sensor

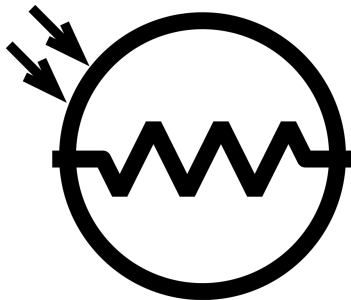
1.22 Fotowiderstand



Ein Fotowiderstand oder eine Fotodiode ist ein lichtgesteuerter variabler Widerstand. Der Widerstand eines Fotowiderstands nimmt mit steigender Lichteinstrahlungsintensität ab; mit anderen Worten, er zeigt Foto-Leitfähigkeit.

Ein Fotowiderstand kann in lichtempfindlichen Detektorschaltungen und licht- sowie dunkelaktivierten Schaltkreisen eingesetzt werden und fungiert dabei als ein Widerstandshalbleiter. Im Dunkeln kann ein Fotowiderstand einen Widerstand von mehreren Megaohm (M) haben, während er im Licht einen Widerstand von nur wenigen hundert Ohm haben kann.

Hier ist das elektronische Symbol des Fotowiderstands.

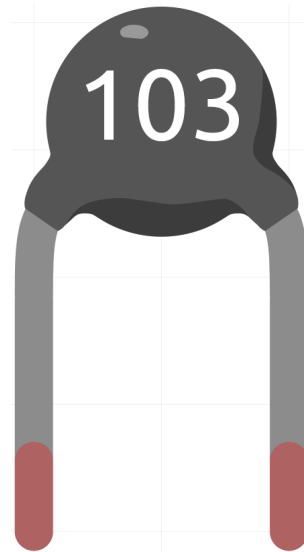


- [Fotowiderstand - Wikipedia](#)

Beispiel

- *4.2 Das Licht fühlen* (Grundlegendes Projekt)
- *5. Überwachung des häuslichen Umfelds* (IoT-Projekt)
- *6. Pflanzenüberwachung* (IoT-Projekt)

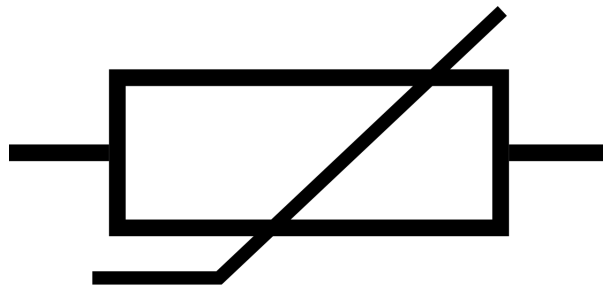
1.23 Thermistor



Ein Thermistor ist eine Art von Widerstand, dessen Widerstand stark temperaturabhängig ist, mehr als bei Standardwiderständen. Das Wort ist eine Kombination aus den Wörtern „thermal“ und „Resistor“. Thermistoren werden häufig als Einschaltstrombegrenzer, Temperatursensoren (typischerweise vom Typ mit negativem Temperaturkoeffizienten oder NTC), selbstrückstellende Überstromschutzelemente und selbstregulierende Heizelemente (typischerweise vom Typ mit positivem Temperaturkoeffizienten oder PTC) eingesetzt.

- [Thermistor - Wikipedia](#)

Hier ist das elektronische Symbol des Thermistors.



Es gibt zwei grundsätzlich entgegengesetzte Typen von Thermistoren:

- Bei NTC-Thermistoren nimmt der Widerstand mit steigender Temperatur ab, meistens aufgrund einer Zunahme von Leitungselektronen, die durch thermische Agitation aus dem Valenzband gehoben werden. Ein NTC wird häufig als Temperatursensor verwendet oder in Reihe zu einer Schaltung als Einschaltstrombegrenzer geschaltet.
- Bei PTC-Thermistoren nimmt der Widerstand mit steigender Temperatur zu, meistens aufgrund erhöhter thermischer Gitteranregungen, insbesondere von Verunreinigungen und Unvollkommenheiten. PTC-Thermistoren werden häufig in Reihe zu einer Schaltung installiert und zum Schutz vor Überstrombedingungen als rückstellbare Sicherungen verwendet.

In diesem Kit verwenden wir einen NTC-Typ. Jeder Thermistor hat einen Normalwiderstand. Hier beträgt er 10k Ohm, gemessen bei 25 Grad Celsius.

Hier ist der Zusammenhang zwischen Widerstand und Temperatur:

$$R_T = R_N * \exp(B(1/T_K - 1/T_N))$$

- **RT** ist der Widerstand des NTC-Thermistors bei der Temperatur TK.
- **RN** ist der Widerstand des NTC-Thermistors bei der Nenntemperatur TN. Hier beträgt der Zahlenwert von RN 10k.
- **TK** ist eine Kelvin-Temperatur und die Einheit ist K. Hier beträgt der Zahlenwert von TK 273,15 + Grad Celsius.
- **TN** ist eine Nenn-Kelvin-Temperatur; die Einheit ist ebenfalls K. Hier beträgt der Zahlenwert von TN 273,15 + 25.
- Und **B(beta)**, die Materialkonstante des NTC-Thermistors, wird auch als Wärmeempfindlichkeitsindex bezeichnet mit einem Zahlenwert von 3950.
- **exp** steht für „exponentiell“, und die Basiszahl e ist eine natürliche Zahl und beträgt ungefähr 2,7.

Wandeln Sie diese Formel $TK = 1 / (\ln(RT/RN) / B + 1 / TN)$ um, um die Kelvin-Temperatur zu erhalten, die minus 273,15 Grad Celsius entspricht.

Dieser Zusammenhang ist eine empirische Formel. Sie ist nur dann genau, wenn die Temperatur und der Widerstand innerhalb des wirksamen Bereichs liegen.

Beispiel

- [6.3 Hohe Temperaturalarm](#) (Grundlegendes Projekt)
- [4.5 Thermometer](#) (Grundlegendes Projekt)

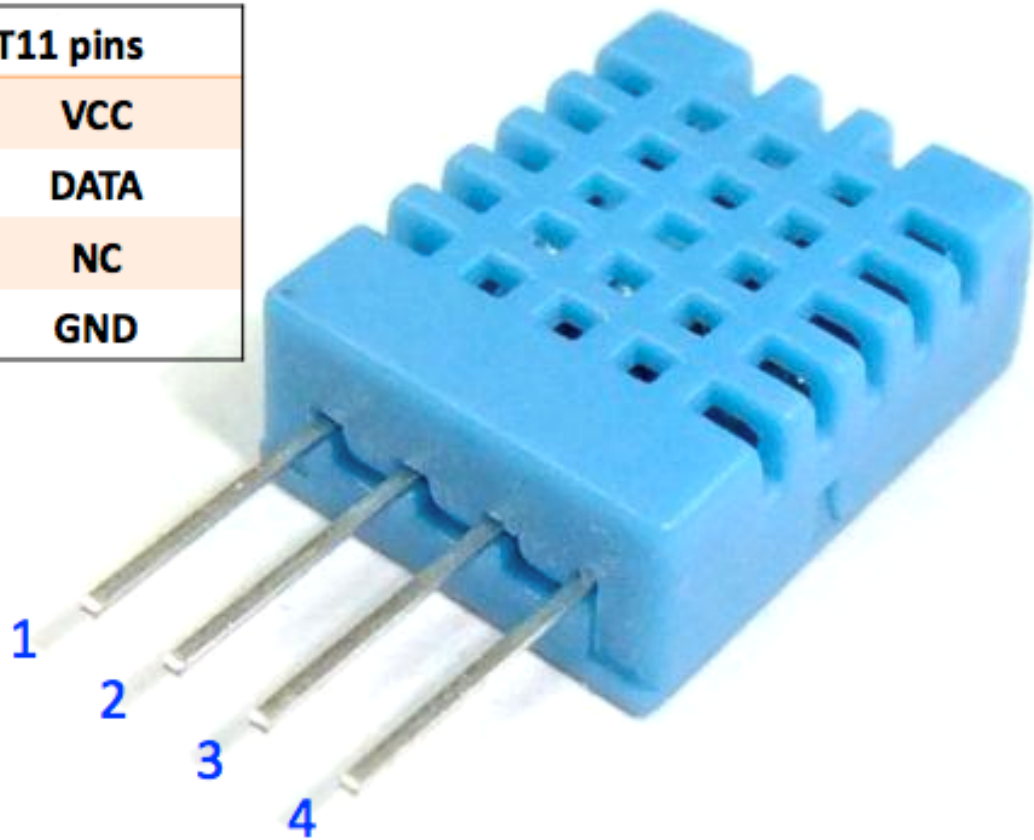
1.24 DHT11 Feuchtigkeits- und Temperatursensor

Der digitale Temperatur- und Feuchtigkeitssensor DHT11 ist ein kombinierter Sensor, der einen kalibrierten digitalen Ausgang für Temperatur und Feuchtigkeit bietet. Die Technologie einer speziellen digitalen Modulsammlung sowie die Technologie zur Temperatur- und Feuchtigkeitserfassung werden verwendet, um sicherzustellen, dass das Produkt eine hohe Zuverlässigkeit und hervorragende Langzeitstabilität aufweist.

Der Sensor beinhaltet einen resistiven Feuchtigkeitssensor und ein NTC-Temperaturmessgerät und ist mit einem leistungsstarken 8-Bit-Mikrocontroller verbunden.

Es stehen nur drei Pins zur Verfügung: VCC, GND und DATA. Der Kommunikationsprozess beginnt damit, dass die DATA-Leitung Startsignale an den DHT11 sendet. Der DHT11 empfängt diese Signale und sendet ein Antwortsignal zurück. Anschließend empfängt der Host das Antwortsignal und beginnt mit dem Empfang von 40-Bit Feuchtigkeitsdaten (8-Bit Feuchtigkeitswert ganzzahlig + 8-Bit Feuchtigkeitswert dezimal + 8-Bit Temperaturwert ganzzahlig + 8-Bit Temperaturwert dezimal + 8-Bit Prüfsumme).

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



Merkmale

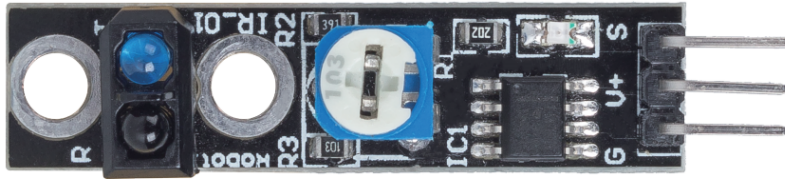
1. Messbereich für Feuchtigkeit: 20 - 90%RH
2. Temperaturmessbereich: 0 - 60°C
3. Ausgabe digitaler Signale für Temperatur und Feuchtigkeit
4. Betriebsspannung: DC 5V; PCB-Größe: 2,0 x 2,0 cm
5. Messgenauigkeit für Feuchtigkeit: $\pm 5\%$ RH
6. Messgenauigkeit für Temperatur: $\pm 2^\circ\text{C}$

- [DHT11 Datenblatt](#)

Beispiel

- *5.11.3 Temperatur - Luftfeuchtigkeit* (Grundlegendes Projekt)
- *5. Überwachung des häuslichen Umfelds* (IoT-Projekt)
- *6. Pflanzenüberwachung* (IoT-Projekt)

1.25 Linienverfolgungsmodul



- S: Normalerweise niedriges Level, hohes Level bei Erkennung einer schwarzen Linie.
- V+Stromversorgung, 3.3v~5V
- G: Masse

Dies ist ein 1-Kanal-Linienverfolgungsmodul, das, wie der Name schon sagt, schwarze Linien auf einem weißen Hintergrund oder weiße Linien auf einem schwarzen Hintergrund verfolgt.



Das Modul verwendet einen TCRT5000 Infrarotsensor, der aus einer Infrarot-LED (blau) und einem lichtempfindlichen Triplet (schwarz) besteht.

- Die blaue Infrarot-LED sendet bei eingeschaltetem Zustand Infrarotlicht aus, das für das menschliche Auge unsichtbar ist.
- Der schwarze Fototransistor, der zum Empfang von Infrarotlicht verwendet wird, hat einen internen Widerstand, dessen Widerstand mit dem empfangenen Infrarotlicht variiert; je mehr Infrarotlicht empfangen wird, desto stärker nimmt sein Widerstand ab und umgekehrt.

Auf dem Modul befindet sich ein LM393-Komparator, der dazu dient, die Spannung des Fototransistors mit der eingestellten Spannung (durch Potentiometer angepasst) zu vergleichen. Ist sie größer als die eingestellte Spannung, beträgt der Ausgang 1; sonst beträgt der Ausgang 0.

Daher, wenn die Infrarot-Senderröhre auf eine schwarze Oberfläche leuchtet, da Schwarz Licht absorbiert, empfängt der lichtempfindliche Transistor weniger Infrarotlicht, sein Widerstand wird erhöhen (Spannungsanstieg). Nach dem LM393-Komparator wird ein hohes Niveau ausgegeben.

Ebenso, wenn er auf eine weiße Oberfläche leuchtet, wird das reflektierte Licht stärker und der Widerstand des lichtempfindlichen Transistors sinkt (Spannungsabfall); daher gibt der Komparator ein niedriges Niveau aus und die Anzeige-LED leuchtet.

- [TCRT5000](#)

Eigenschaften

- Verwendung des Infrarot-Emissionssensors TCRT5000
- Erfassungsentfernung: 1-8mm, Brennweite von 2.5mm

- Komparator-Ausgangssignal ist sauber, mit gutem Wellenform, Antriebskapazität größer als 15mA
- Empfindlichkeitseinstellung durch Potentiometer
- Betriebsspannung: 3.3V-5V
- Digitalausgang: 0 (weiß) und 1 (schwarz)
- Verwendet den weitspannungs LM393 Komparator.
- Größe: 42mmx10mm

Beispiel

- [3.4 Die Linie erkennen](#) (Grundlegendes Projekt)
- [4. Der Linie folgen](#) (Auto-Projekt)

1.26 Bodenfeuchtigkeitsmodul

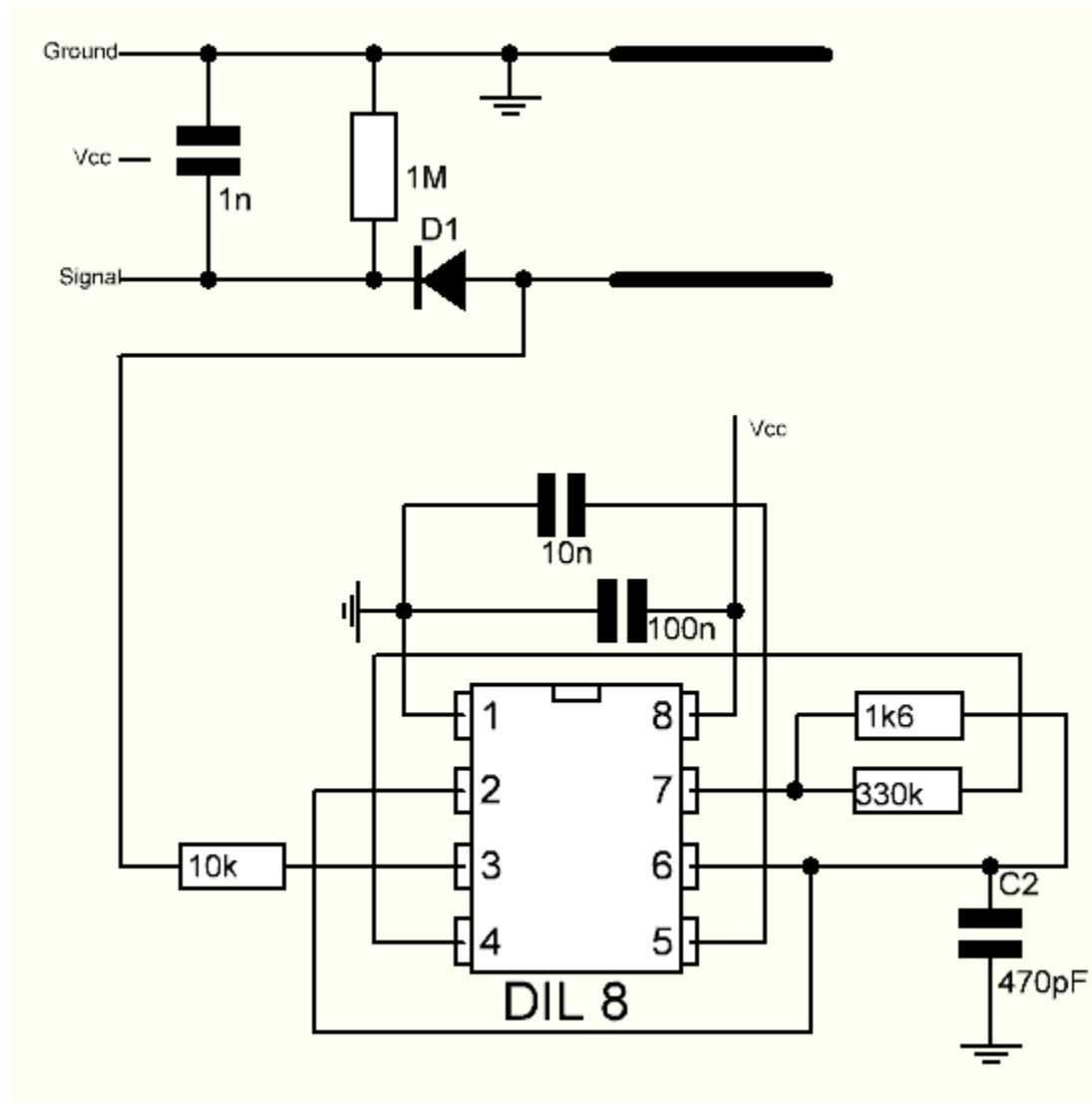


- GND: Masse
- VCC: Stromversorgung, 3.3V~5V
- AUOT: Gibt den Bodenfeuchtigkeitswert aus, je feuchter der Boden, desto geringer ist sein Wert.

Dieser kapazitive Bodenfeuchtigkeitssensor unterscheidet sich von den meisten auf dem Markt erhältlichen Widerstandssensoren, da er das Prinzip der kapazitiven Induktion zur Erfassung der Bodenfeuchtigkeit verwendet. Er umgeht das Problem, dass Widerstandssensoren sehr korrosionsanfällig sind und verlängert so seine Lebensdauer erheblich.

Er besteht aus korrosionsbeständigen Materialien und hat eine hervorragende Lebensdauer. Stecken Sie es in den Boden um Pflanzen herum und überwachen Sie die Bodenfeuchtigkeitsdaten in Echtzeit. Das Modul verfügt über einen integrierten Spannungsregler, der einen Betrieb in einem Spannungsbereich von 3,3 ~ 5,5 V ermöglicht. Es ist ideal für Mikrocontroller mit niedriger Spannung und 3,3 V und 5 V Versorgung geeignet.

Das Hardwareschema des kapazitiven Bodenfeuchtigkeitssensors ist unten dargestellt.



Es gibt einen Festfrequenzoszillator, der mit einem 555-Timer-IC aufgebaut ist. Die erzeugte Rechteckwelle wird dann wie ein Kondensator an den Sensor geleitet. Für das Rechteckwellensignal hat der Kondensator jedoch eine gewisse Reaktanz oder, um der Argumentation willen, einen Widerstand mit einem rein ohmschen Widerstand (10k Widerstand an Pin 3), um einen Spannungsteiler zu bilden.

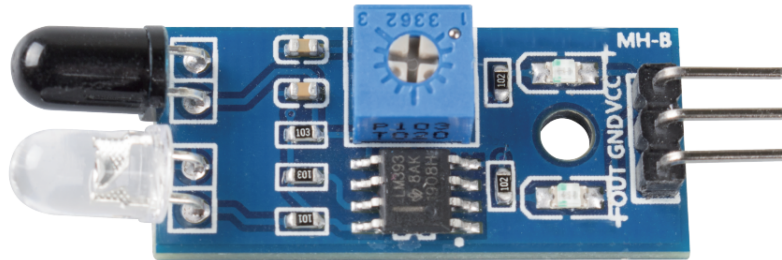
Je höher die Bodenfeuchtigkeit, desto höher die Kapazität des Sensors. Infolgedessen hat die Rechteckwelle eine geringere Reaktanz, was die Spannung an der Signalleitung reduziert, und der Wert des analogen Eingangs durch den Mikrocontroller ist kleiner.

Spezifikation

- Betriebsspannung: 3,3 ~ 5,5 VDC
- Ausgangsspannung: 0 ~ 3,0 VDC
- Betriebsstrom: 5mA
- Schnittstelle: PH2.0-3P
- Abmessungen: 3,86 x 0,905 Zoll (L x B)
- Gewicht: 15g

Beispiel

- 4.4 *Bodenfeuchtigkeit messen* (Grundlegendes Projekt)
- 6. *Pflanzenüberwachung* (IoT-Projekt)

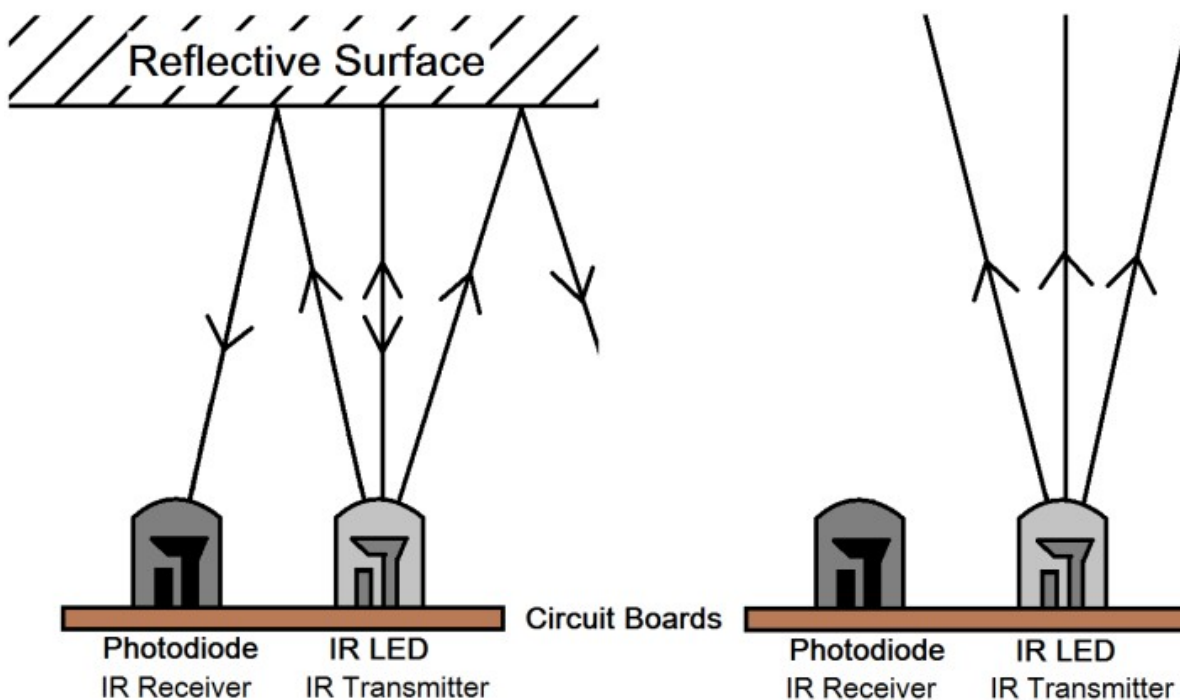
1.27 Hindernisvermeidungsmodul

- **VCC:** Stromversorgung, 3,3 ~ 5V DC.
- **GND:** Masse
- **OUT:** Signalausgang, normalerweise im hohen Zustand, schaltet auf niedrig, wenn ein Hindernis erkannt wird.

Das IR-Hindernisvermeidungsmodul passt sich stark dem Umgebungslicht an und verfügt über ein Paar Infrarot-Sende- und Empfangsröhren.

Die Sendetube gibt eine Infrarotfrequenz ab. Trifft die Erkennungsrichtung auf ein Hindernis, wird die Infrarotstrahlung von der Empfangsröhre aufgenommen. Nach der Verarbeitung im Komparator-Schaltkreis leuchtet die Anzeige auf und gibt ein niedriges Signal aus.

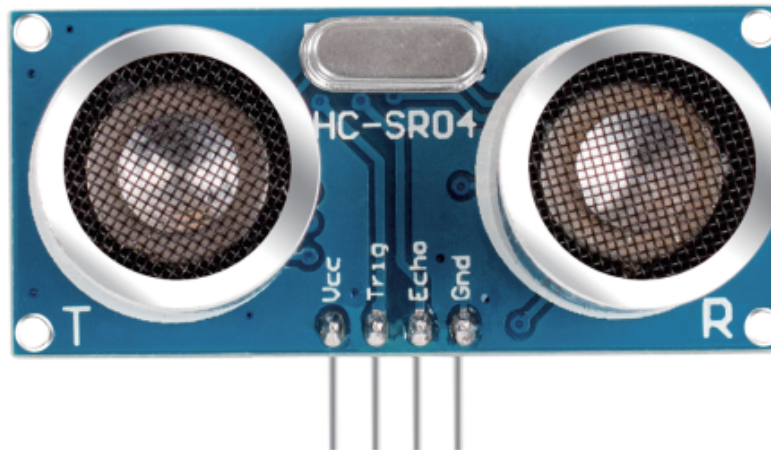
Die Erkennungsentfernung kann mit einem Potentiometer angepasst werden, wobei der effektive Entfernungsbereich zwischen 2-30 cm liegt.



Beispiel

- *3.3 Hindernis erkennen* (Grundlegendes Projekt)
- *5. Spielen mit dem Hindernisvermeidungs-Modul* (Auto-Projekt)
- *8. Selbstfahrendes Auto* (Auto-Projekt)
- *7. Strombegrenztes Tor* (IoT-Projekt)

1.28 Ultraschall-Modul

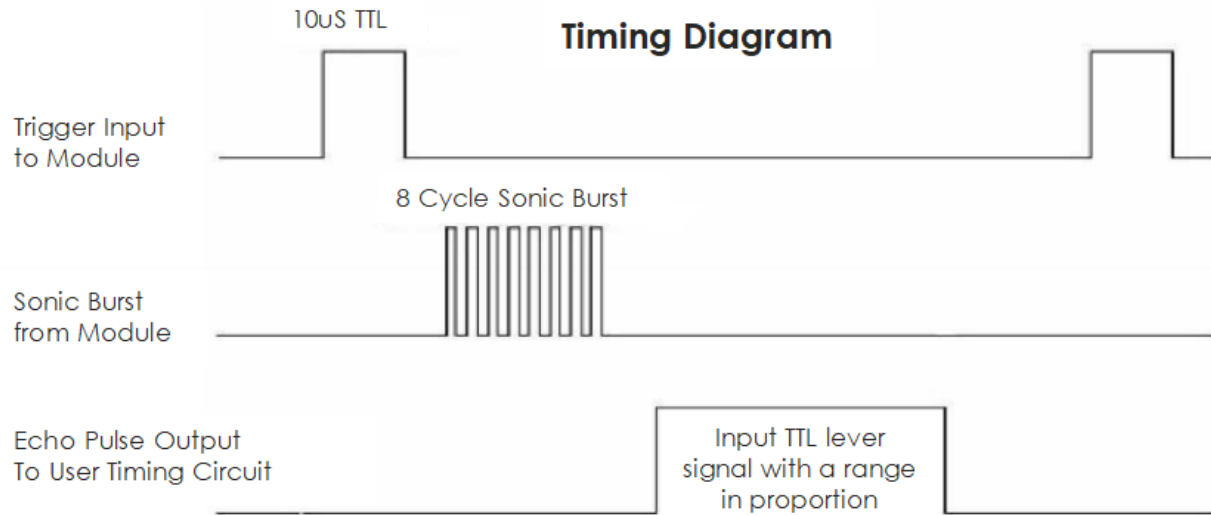


Das Ultraschall-Entfernungsmessmodul bietet eine berührungslose Messfunktion von 2 cm bis 400 cm, wobei die Messgenauigkeit bis zu 3 mm betragen kann. Es gewährleistet, dass das Signal innerhalb von 5 m stabil ist, und das Signal wird nach 5 m allmählich abgeschwächt, bis es an der 7 m Position verschwindet.

Das Modul beinhaltet Ultraschallsender, Empfänger und Steuerschaltung. Die grundlegenden Prinzipien sind wie folgt:

1. Verarbeitung eines High-Level-Signals von mindestens 10us mit einem IO-Flipflop.
2. Das Modul sendet automatisch acht 40kHz aus und überprüft, ob ein Pulssignal zurückkommt.
3. Kehrt das Signal zurück und führt durch das hohe Level, beträgt die Dauer des hohen IO-Ausgangs die Zeit von der Aussendung der Ultraschallwelle bis zu ihrer Rückkehr. Hierbei gilt: Testentfernung = (Hohe Zeit x Schallgeschwindigkeit (340 m/s)) / 2.

Das Zeitdiagramm ist unten dargestellt.



Sie müssen lediglich einen kurzen 10µs-Impuls für den Trigger-Eingang bereitstellen, um die Entfernungsmessung zu starten, und dann sendet das Modul eine 8-Zyklen-Ultraschallburst mit 40 kHz aus und erhebt sein Echo. Sie können die Entfernung über das Zeitintervall zwischen dem Senden des Trigger-Signals und dem Empfangen des Echo-Signals berechnen.

Formel: $\mu s / 58 = \text{Zentimeter}$ oder $\mu s / 148 = \text{Inch}$; oder: Entfernung = hohe Signalzeit * Geschwindigkeit (340M/s) / 2; es wird empfohlen, einen Messzyklus von über 60ms zu verwenden, um Signalüberschneidungen zwischen Trigger-Signal und dem Echo-Signal zu verhindern.

Beispiel

- *5.8 Benutzerdefinierte Funktion* (Grundlegendes Projekt)
- *7. Folge deiner Hand* (Auto-Projekt)
- *6. Mit dem Ultraschallmodul spielen* (Auto-Projekt)

Mit Arduino loslegen

Wenn Sie keine Ahnung von Arduino haben. Es gibt einige Begriffe, die ich Ihnen vorstellen möchte: Elektronik, Design, Programmierung und sogar Maker. Einige von Ihnen denken vielleicht, dass diese Wörter weit von uns entfernt sind, aber tatsächlich sind sie es nicht. Denn Arduino kann uns in die Welt der Programmierung führen und uns helfen, den Traum zu verwirklichen, ein Maker zu sein. In dieser Session lernen wir:

- Was ist Arduino?
- Was kann Arduino tun?
- Wie erstellt man ein Arduino-Projekt?

2.1 Was ist Arduino?

Zunächst möchte ich Ihnen eine kurze Einführung in Arduino geben.

Arduino ist eine praktische, flexible und benutzerfreundliche Open-Source-Plattform für den elektronischen Prototypenbau, die sowohl Hardware-Arduino-Boards verschiedener Modelle als auch die Software Arduino IDE umfasst. Es eignet sich nicht nur für Ingenieure für schnelle Prototypen, sondern auch für Künstler, Designer und Hobbyisten. Es ist fast ein unverzichtbares Werkzeug für moderne Maker.

Arduino ist ein recht umfangreiches System. Es verfügt über Software, Hardware und eine sehr große Online-Community von Menschen, die sich nie getroffen haben, aber dank eines gemeinsamen Hobbys zusammenarbeiten können. Jeder in der Arduino-Familie nutzt seine Weisheit, kreiert mit seinen Händen und teilt eine großartige Erfindung nach der anderen. Und auch Sie können ein Teil davon sein.

2.2 Was kann Arduino tun?

Sie fragen sich vielleicht, was Arduino eigentlich kann. Kurz gesagt, Arduino wird all Ihre Probleme lösen.

Technisch gesehen ist Arduino ein programmierbarer Logikcontroller. Es handelt sich um ein Entwicklungsboard, mit dem viele spannende und kreative elektronische Kreationen erstellt werden können: wie ferngesteuerte Autos, Roboterarme, bionische Roboter, Smart Homes usw.

Arduino-Boards sind unkompliziert, einfach und leistungsstark, geeignet für Schüler, Maker und sogar professionelle Programmierer.

Bis heute entwickeln Elektronikbegeisterte weltweit weiterhin kreative elektronische Kreationen basierend auf Arduino-Entwicklungsboards.

2.3 Wie erstellt man ein Arduino-Projekt

Befolgen Sie diese Schritte, um von Grund auf zu lernen, wie Sie Arduino verwenden!

2.3.1 Download und Installation der Arduino IDE 2.0

Die Arduino IDE, bekannt als Arduino Integrierte Entwicklungsumgebung, bietet alle notwendige Softwareunterstützung, um ein Arduino-Projekt abzuschließen. Es handelt sich um eine speziell für Arduino entwickelte Programmiersoftware, die vom Arduino-Team bereitgestellt wird und es uns ermöglicht, Programme zu schreiben und sie auf das Arduino-Board hochzuladen.

Die Arduino IDE 2.0 ist ein Open-Source-Projekt. Sie ist ein großer Fortschritt gegenüber ihrem robusten Vorgänger, der Arduino IDE 1.x, und kommt mit überarbeitetem UI, verbesserter Board- & Bibliotheksverwaltung, Debugger, Autocomplete-Funktion und vielem mehr.

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

Anforderungen

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

Download der Arduino IDE 2.0

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



Arduino IDE 2.0.0

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

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

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

SOURCE CODE

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

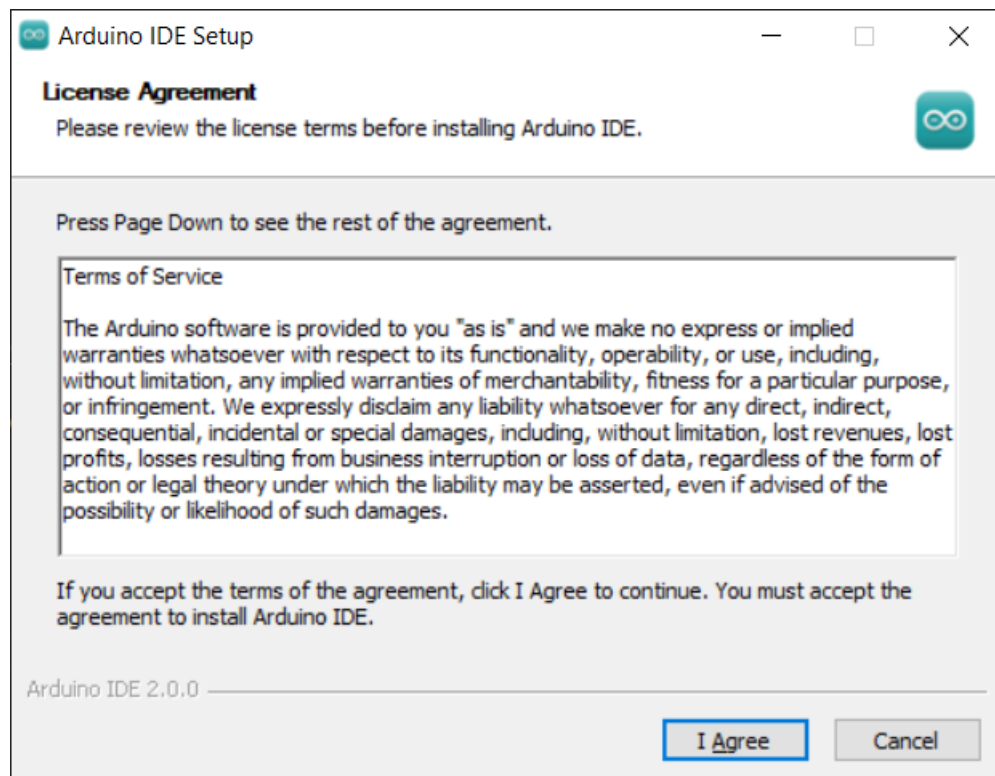
DOWNLOAD OPTIONS

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

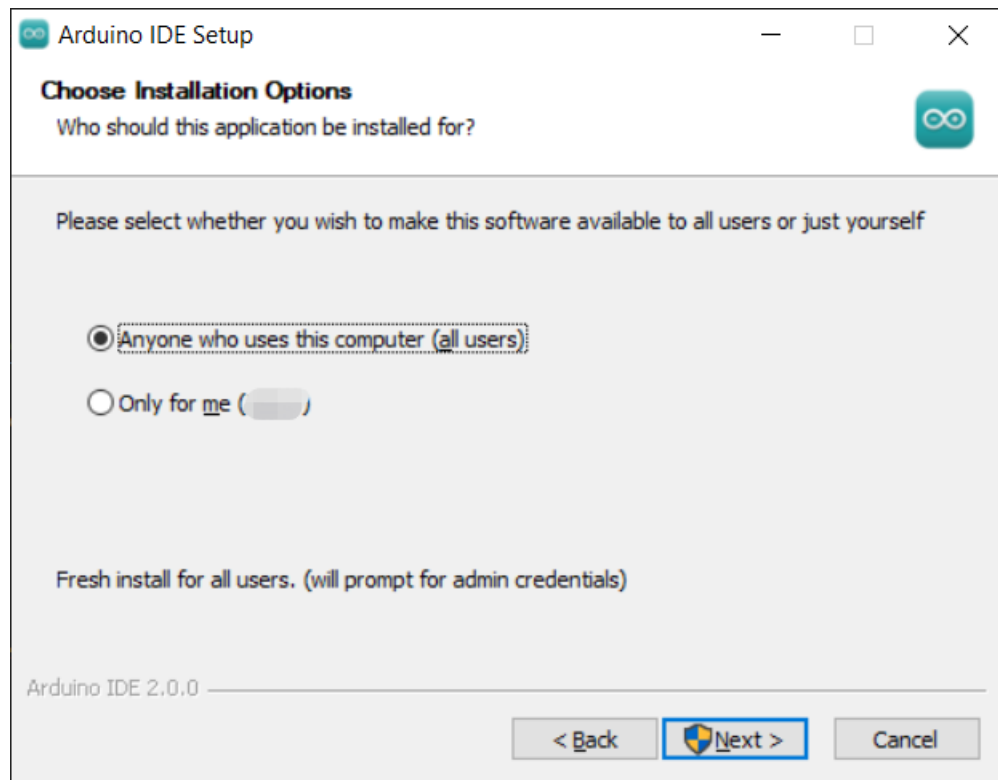
Installation

Windows

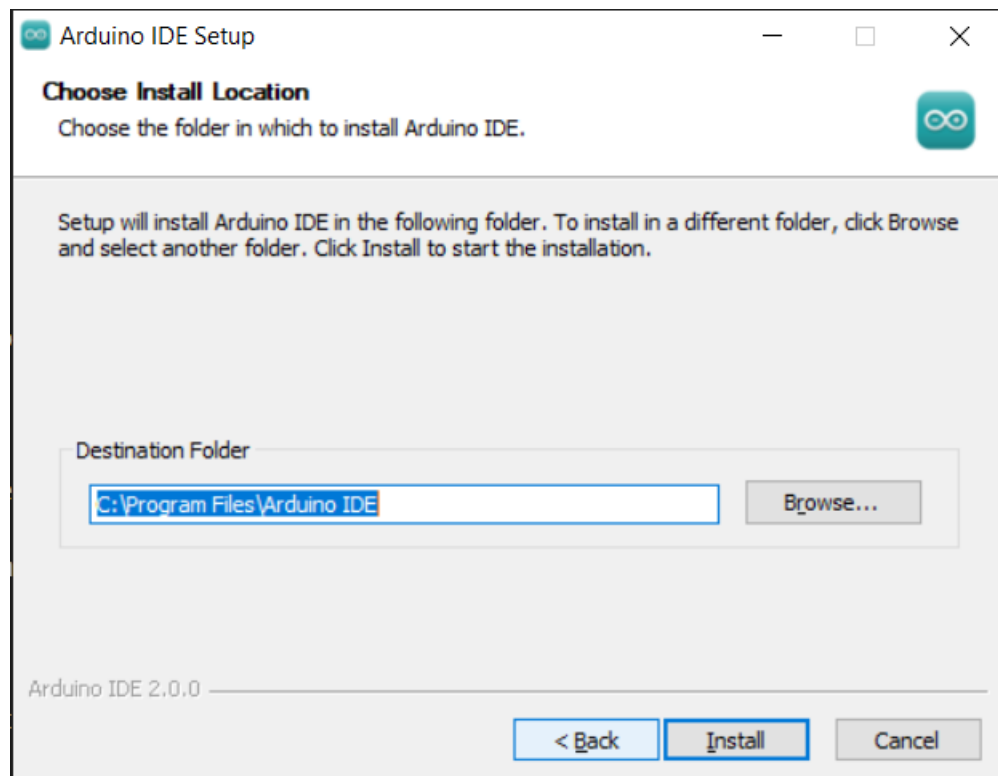
1. Doppelklicken Sie auf die Datei `arduino-ide_xxxx.exe`, um die heruntergeladene Datei auszuführen.
2. Lesen Sie die Lizenzvereinbarung und stimmen Sie dieser zu.



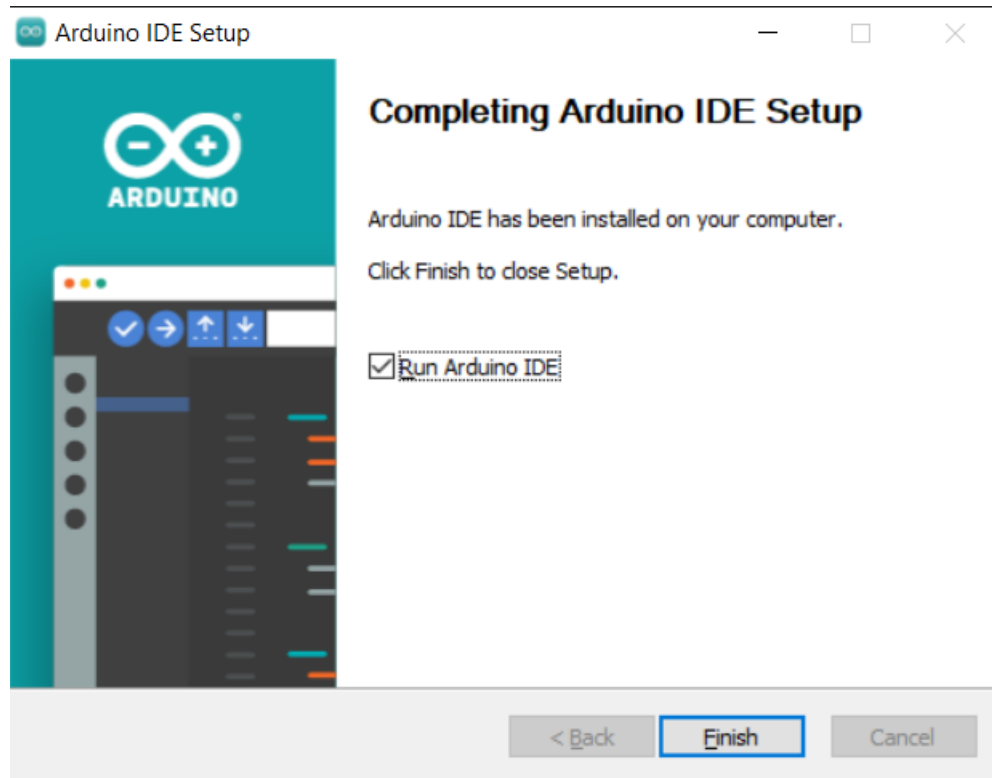
3. Wählen Sie die Installationsoptionen.



4. Wählen Sie den Installationsort. Es wird empfohlen, die Software auf einem Laufwerk zu installieren, das nicht das Systemlaufwerk ist.

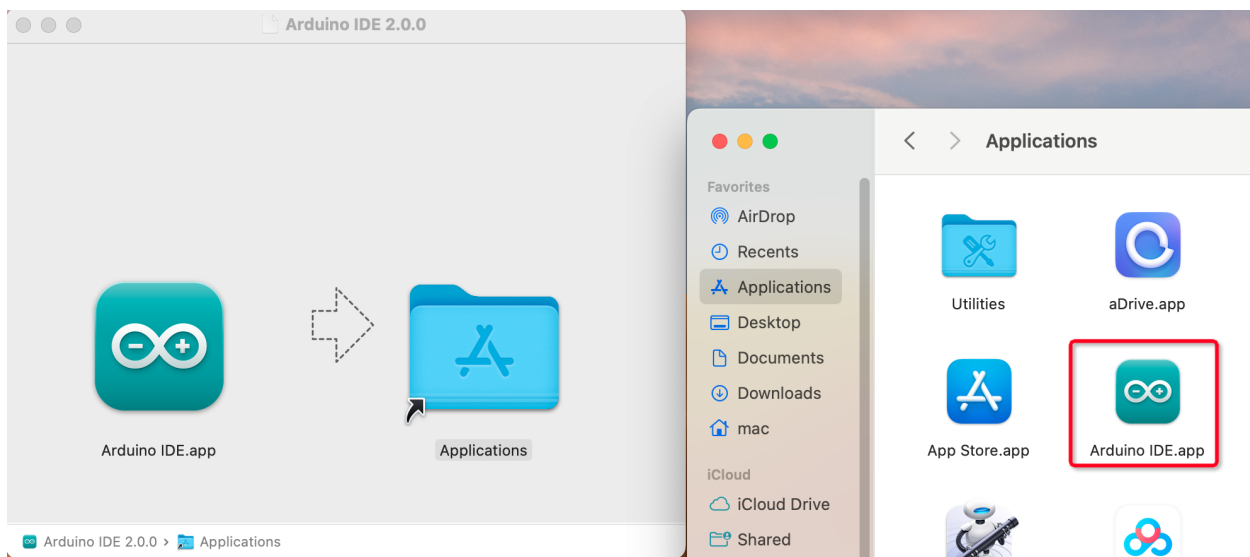


5. Dann beenden Sie die Installation.



macOS

Doppelklicken Sie auf die heruntergeladene Datei `arduino_ide_xxxx.dmg` und befolgen Sie die Anweisungen, um die **Arduino IDE.app** in den **Applications**-Ordner zu kopieren. Nach einigen Sekunden sehen Sie, dass die Arduino IDE erfolgreich installiert wurde.

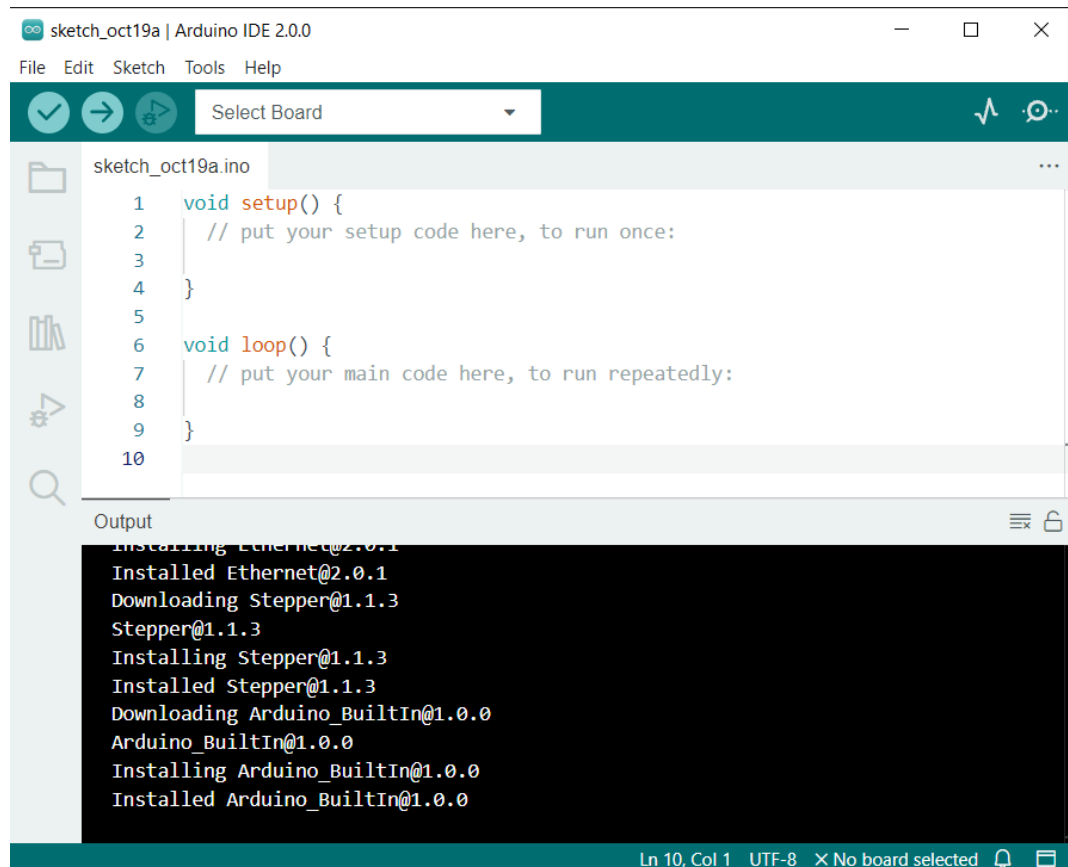


Linux

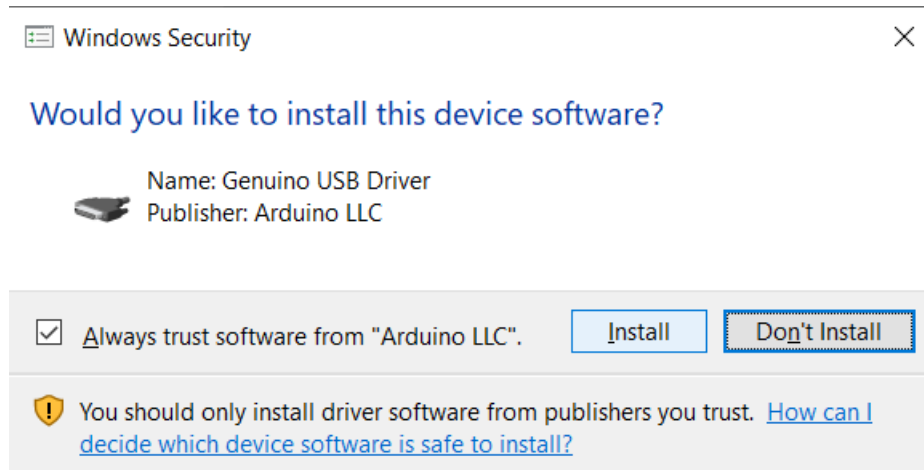
Für das Tutorial zur Installation der Arduino IDE 2.0 auf einem Linux-System, folgen Sie bitte: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing#linux>

Öffnen der IDE

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



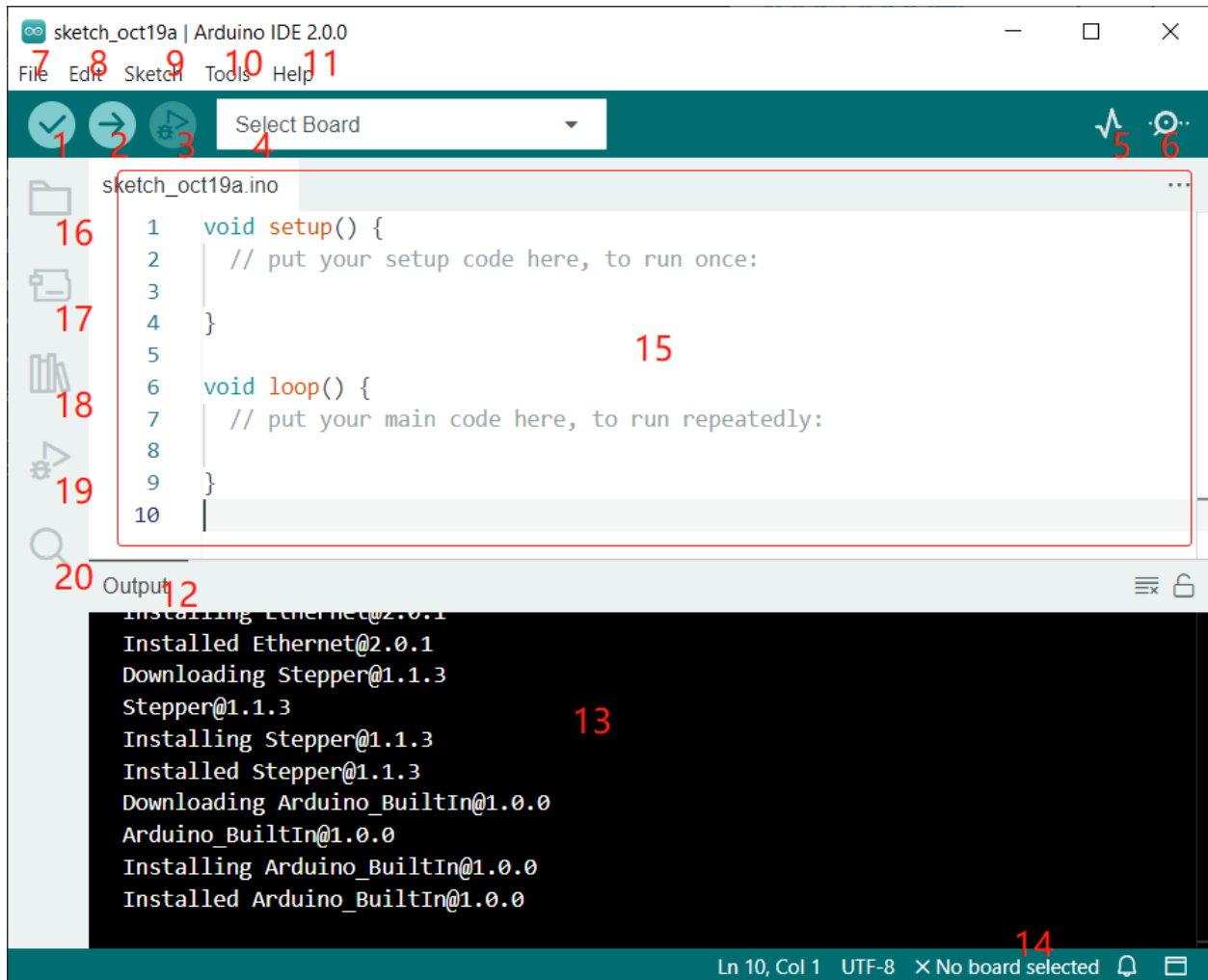
2. Zusätzlich kann Ihr Firewall- oder Sicherheitszentrum einige Male eine Benachrichtigung anzeigen, in der Sie gefragt werden, ob Sie einen bestimmten Gerätetreiber installieren möchten. Bitte installieren Sie alle.



3. Jetzt ist Ihre Arduino IDE einsatzbereit!

Bemerkung: Falls einige Installationen aufgrund von Netzwerkproblemen oder anderen Gründen nicht funktioniert haben, können Sie die Arduino IDE erneut öffnen und sie wird den Rest der Installation abschließen. Das Ausgabefenster öffnet sich nach Abschluss aller Installationen nicht automatisch, es sei denn, Sie klicken auf Überprüfen oder Hochladen.

2.3.2 Vorstellung der Arduino IDE

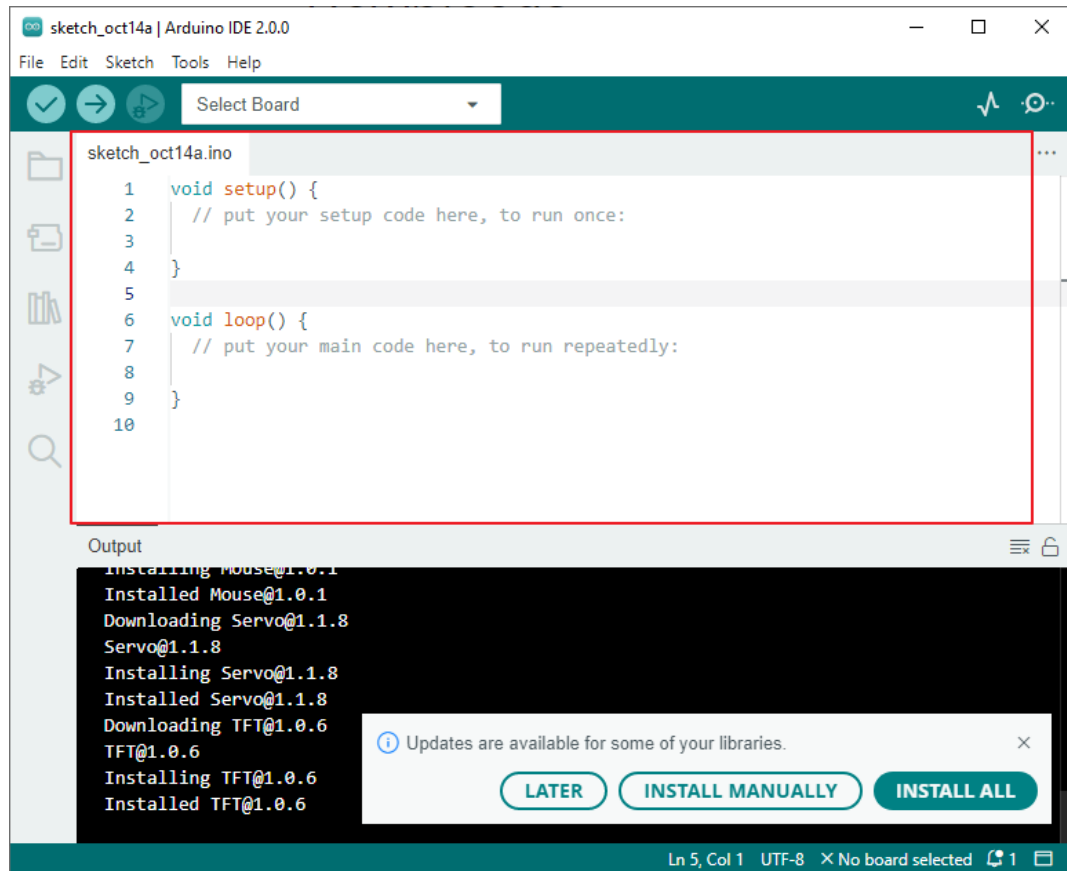


1. **Überprüfen (Verify):** Kompilieren Sie Ihren Code. Bei jeglichen Syntaxproblemen werden Fehlermeldungen angezeigt.
2. **Hochladen (Upload):** Laden Sie den Code auf Ihr Board. Wenn Sie den Button klicken, blinken die RX- und TX-LEDs auf dem Board schnell und hören erst auf, wenn das Hochladen abgeschlossen ist.
3. **Debuggen (Debug):** Zeilenweises Überprüfen auf Fehler.
4. **Board auswählen (Select Board):** Schnelleinstellung von Board und Port.
5. **Serien-Plotter (Serial Plotter):** Überprüfen Sie die Änderung des gelesenen Wertes.
6. **Serien-Monitor (Serial Monitor):** Klicken Sie auf den Button und ein Fenster öffnet sich. Es empfängt die Daten, die von Ihrer Steuerplatine gesendet werden. Sehr nützlich für das Debugging.
7. **Datei (File):** Klicken Sie auf das Menü und eine Dropdown-Liste erscheint. Sie enthält Optionen wie Datei erstellen, öffnen, speichern, schließen, einige Parameter konfigurieren usw.
8. **Bearbeiten (Edit):** Klicken Sie auf das Menü. In der Dropdown-Liste gibt es Bearbeitungsoptionen wie **Ausschneiden**, **Kopieren**, **Einfügen**, **Suchen** usw. mit ihren entsprechenden Tastenkombinationen.
9. **Sketch:** Beinhaltet Aktionen wie **Überprüfen**, **Hochladen**, **Dateien hinzufügen** usw. Eine besonders wichtige Funktion ist **Bibliothek einbinden (Include Library)** - hier können Sie Bibliotheken hinzufügen.

10. **Werkzeug (Tool):** Beinhaltet verschiedene Tools – am häufigsten verwendet sind Board (das verwendete Board) und Port (der Port, an dem sich Ihr Board befindet). Jedes Mal, wenn Sie den Code hochladen möchten, müssen Sie diese auswählen oder überprüfen.
11. **Hilfe (Help):** Wenn Sie Anfänger sind, können Sie die Optionen im Menü überprüfen und die benötigte Hilfe erhalten, einschließlich der Bedienung in der IDE, Einführungsinformationen, Fehlerbehebung, Code-Erklärung usw.
12. **Ausgabeleiste (Output Bar):** Wechseln Sie hier den Ausgabereiter.
13. **Ausgabefenster (Output Window):** Informationen ausgeben.
14. **Board und Port:** Hier können Sie das ausgewählte Board und den Port zur Codeübertragung vorab anzeigen. Falls ein Fehler vorliegt, können Sie diese erneut über **Werkzeuge (Tools)** -> **Board / Port** auswählen.
15. Der Bearbeitungsbereich der IDE. Hier können Sie den Code schreiben.
16. **Skizzenbuch (Sketchbook):** Zur Verwaltung von Skizzen-Dateien.
17. **Board-Verwaltung (Board Manager):** Zur Verwaltung der Board-Treiber.
18. **Bibliotheksverwaltung (Library Manager):** Zur Verwaltung Ihrer Bibliotheksdateien.
19. **Debuggen (Debug):** Unterstützt beim Debugging des Codes.
20. **Suche (Search):** Suchen Sie Codes aus Ihren Skizzen.

2.3.3 Wie erstellt, öffnet oder speichert man den Sketch?

1. Wenn Sie die Arduino IDE zum ersten Mal öffnen oder einen neuen Sketch erstellen, sehen Sie eine Seite wie diese, auf der die Arduino IDE eine neue Datei für Sie erstellt, die als „Sketch“ bezeichnet wird.



Diese Sketch-Dateien haben einen regelmäßigen temporären Namen, aus dem hervorgeht, an welchem Datum die Datei erstellt wurde. `sketch_oct14a.ino` bedeutet den ersten Sketch vom 14. Oktober, `.ino` ist das Dateiformat dieses Sketchs.

2. Versuchen wir nun, einen neuen Sketch zu erstellen. Kopieren Sie den folgenden Code in die Arduino IDE, um den ursprünglichen Code zu ersetzen.



```
void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

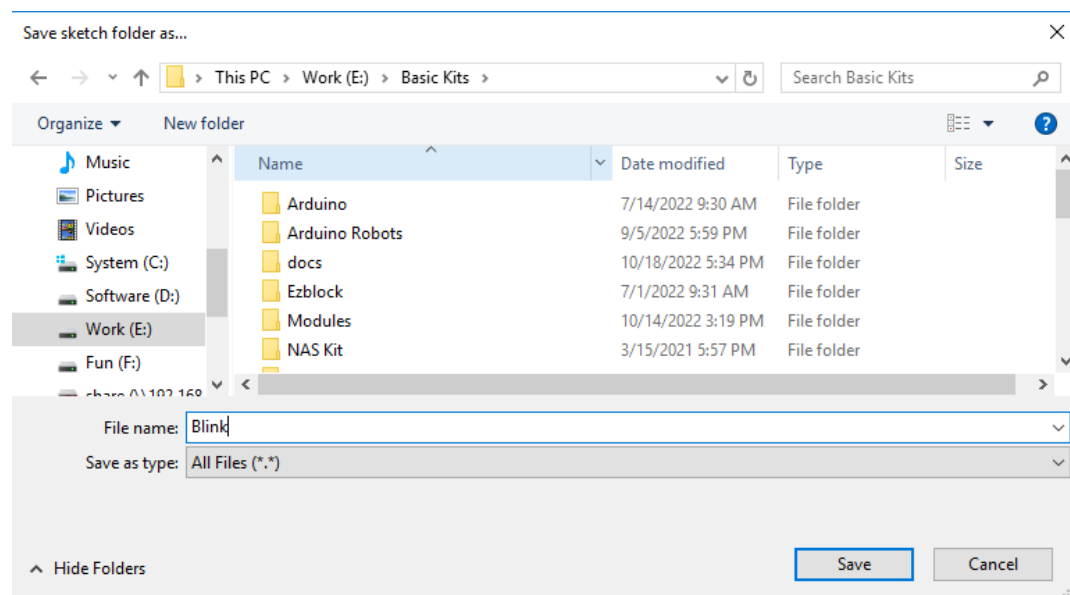
```

}

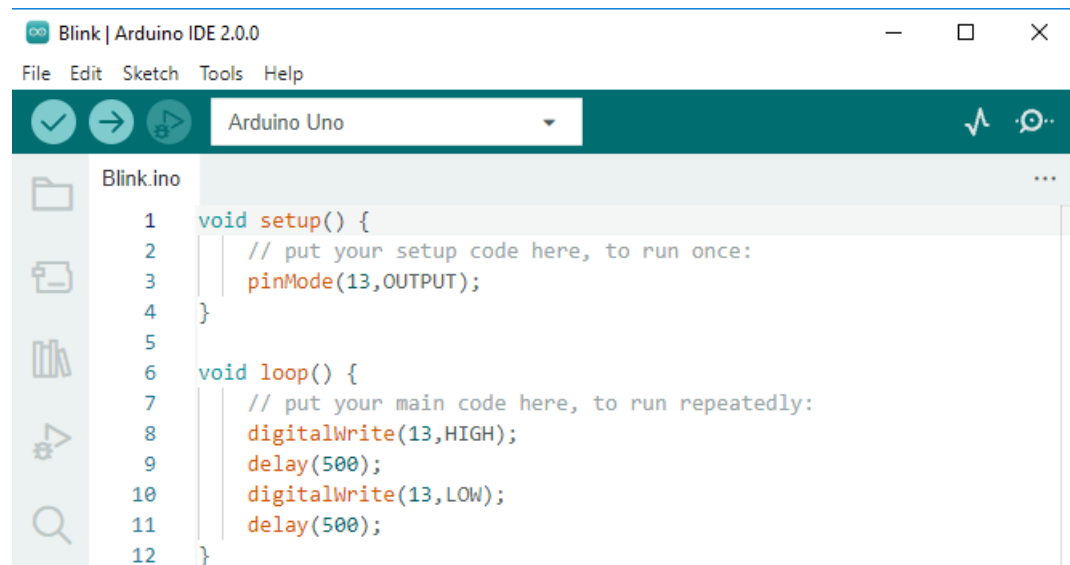
void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13,HIGH);
  delay(500);
  digitalWrite(13,LOW);
  delay(500);
}

```

3. Drücken Sie Ctrl+S oder klicken Sie auf **Datei (File) -> Speichern (Save)**. Der Sketch wird standardmäßig unter C:\Users\{Ihr_Benutzername}\Documents\Arduino gespeichert. Sie können ihn umbenennen oder einen neuen Pfad zum Speichern finden.



4. Nach dem erfolgreichen Speichern sehen Sie, dass der Name in der Arduino IDE aktualisiert wurde.



Bitte fahren Sie mit dem nächsten Abschnitt fort, um zu erfahren, wie Sie diesen erstellten Sketch auf Ihr Arduino-Board hochladen.

2.3.4 Wie lädt man den Sketch auf das Board hoch?

In diesem Abschnitt lernen Sie, wie Sie den zuvor erstellten Sketch auf das Arduino-Board hochladen und welche Aspekte dabei zu beachten sind.

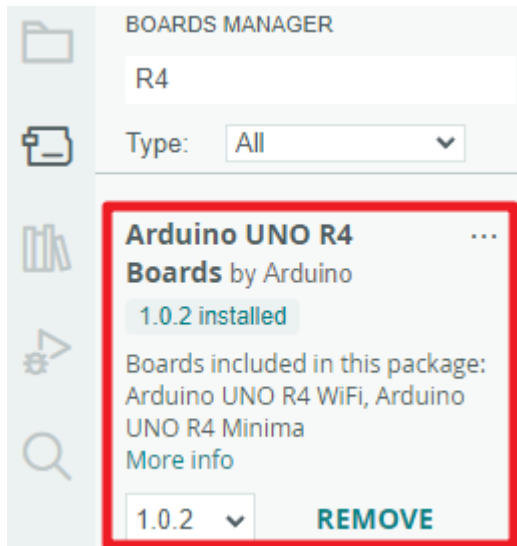
1. Board und Port auswählen

Arduino-Entwicklungsboards werden in der Regel mit einem USB-Kabel geliefert. Sie können es verwenden, um das Board mit Ihrem Computer zu verbinden.

Wählen Sie das richtige **Board** und den richtigen **Port** in der Arduino IDE aus. Normalerweise werden Arduino-Boards vom Computer automatisch erkannt und einem Port zugewiesen, den Sie hier auswählen können.



Wenn Ihr Board bereits angeschlossen ist, aber nicht erkannt wird, überprüfen Sie, ob das **INSTALLED**-Logo im Abschnitt **Arduino UNO R4 Boards** des **Boards Managers** angezeigt wird. Wenn nicht, scrollen Sie bitte ein wenig nach unten und klicken Sie auf **INSTALL**.



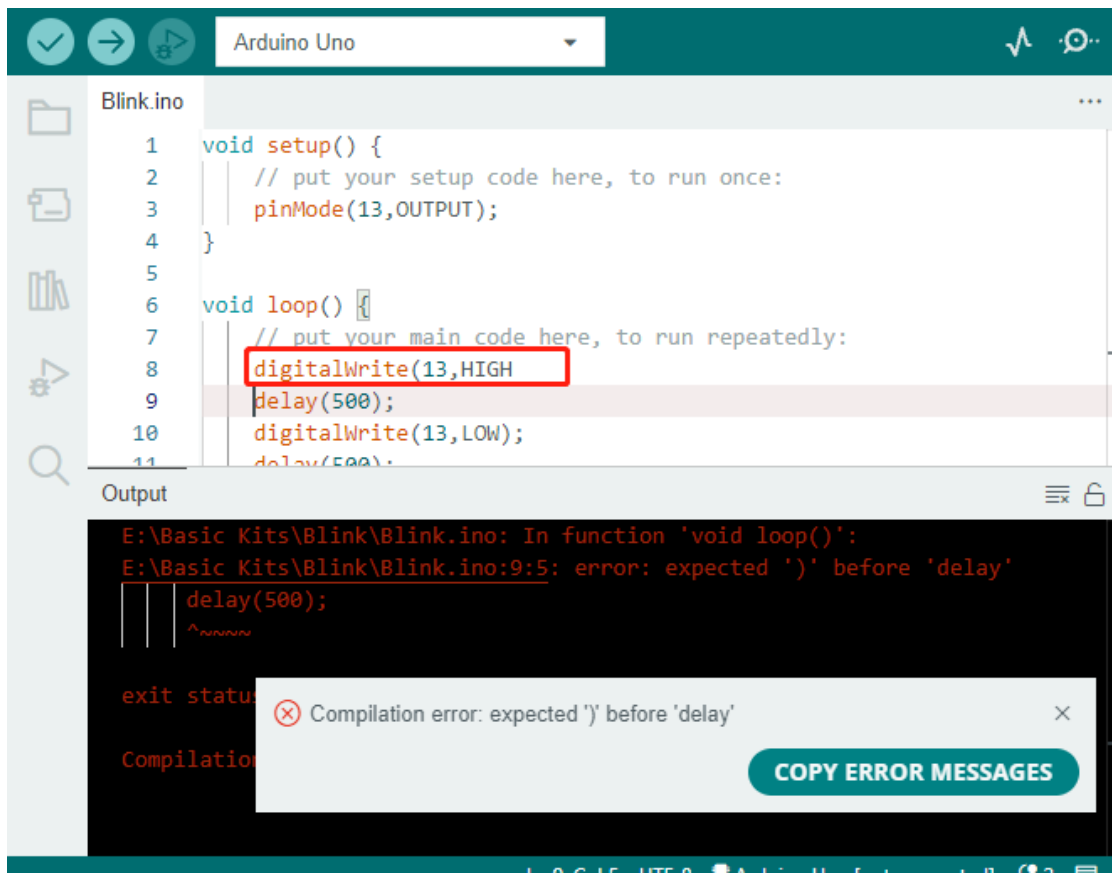
Das erneute Öffnen der Arduino IDE und das erneute Anschließen des Arduino-Boards beheben die meisten Probleme. Sie können auch auf **Werkzeuge (Tools)** -> **Board** oder **Port** klicken, um diese auszuwählen.

2. Den Sketch überprüfen

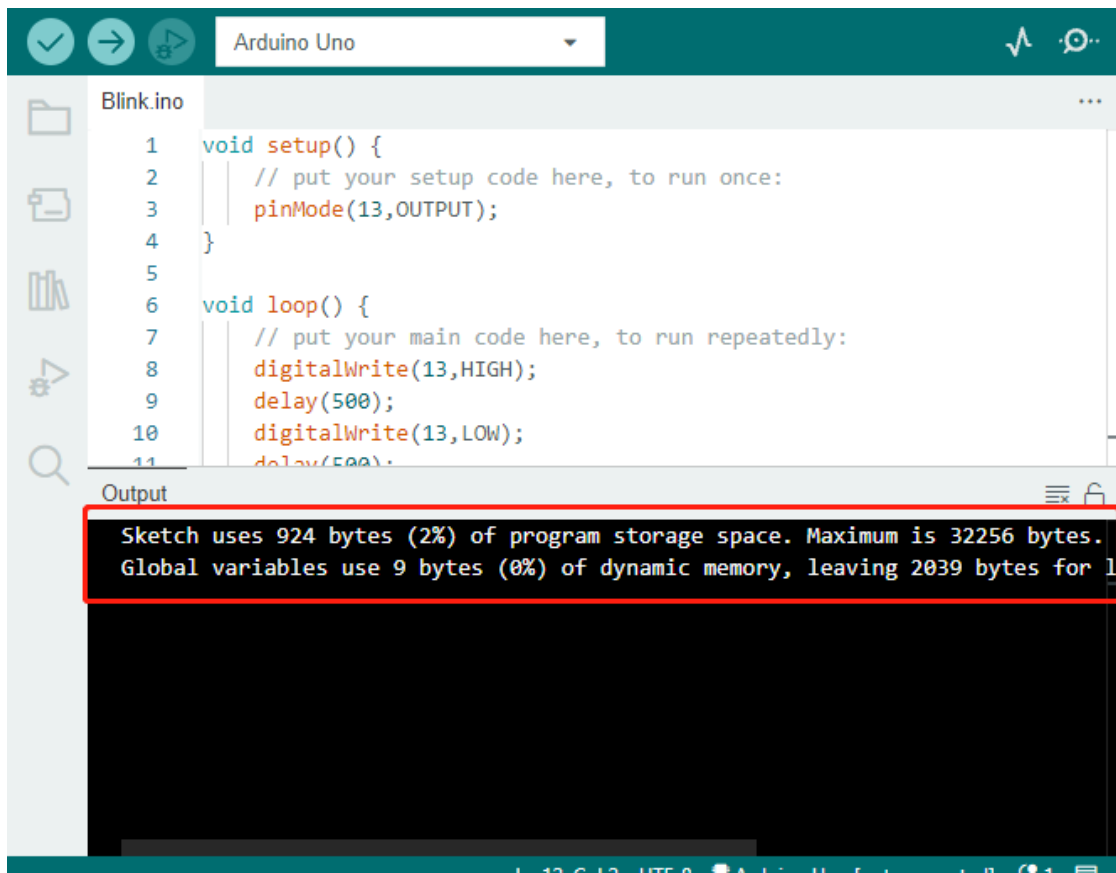
Nachdem Sie auf die Schaltfläche Überprüfen (Verify) geklickt haben, wird der Sketch kompiliert, um zu sehen, ob Fehler vorliegen.



Damit können Sie Fehler finden, falls Sie einige Zeichen gelöscht oder einige Buchstaben versehentlich eingetippt haben. In der Nachrichtenzeile können Sie sehen, wo und welche Art von Fehlern aufgetreten sind.

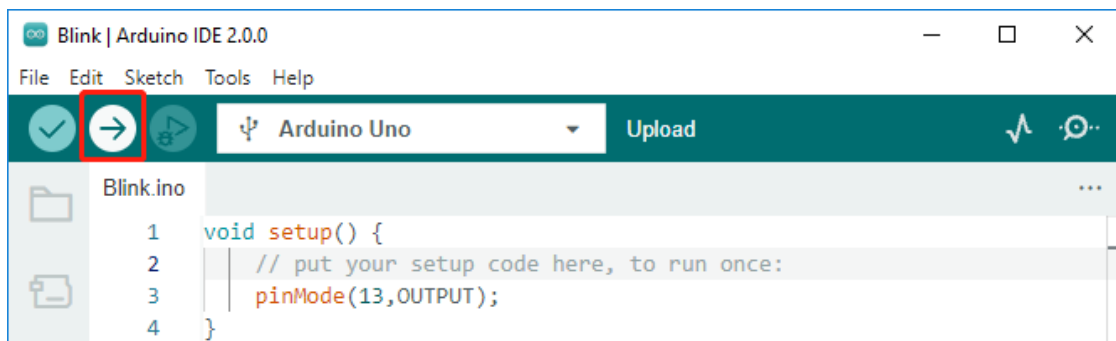


Wenn keine Fehler vorhanden sind, sehen Sie eine Meldung wie die untenstehende.

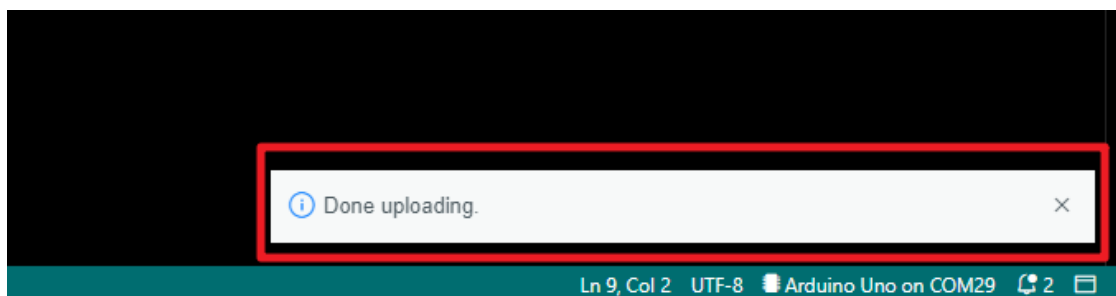


3. Sketch hochladen

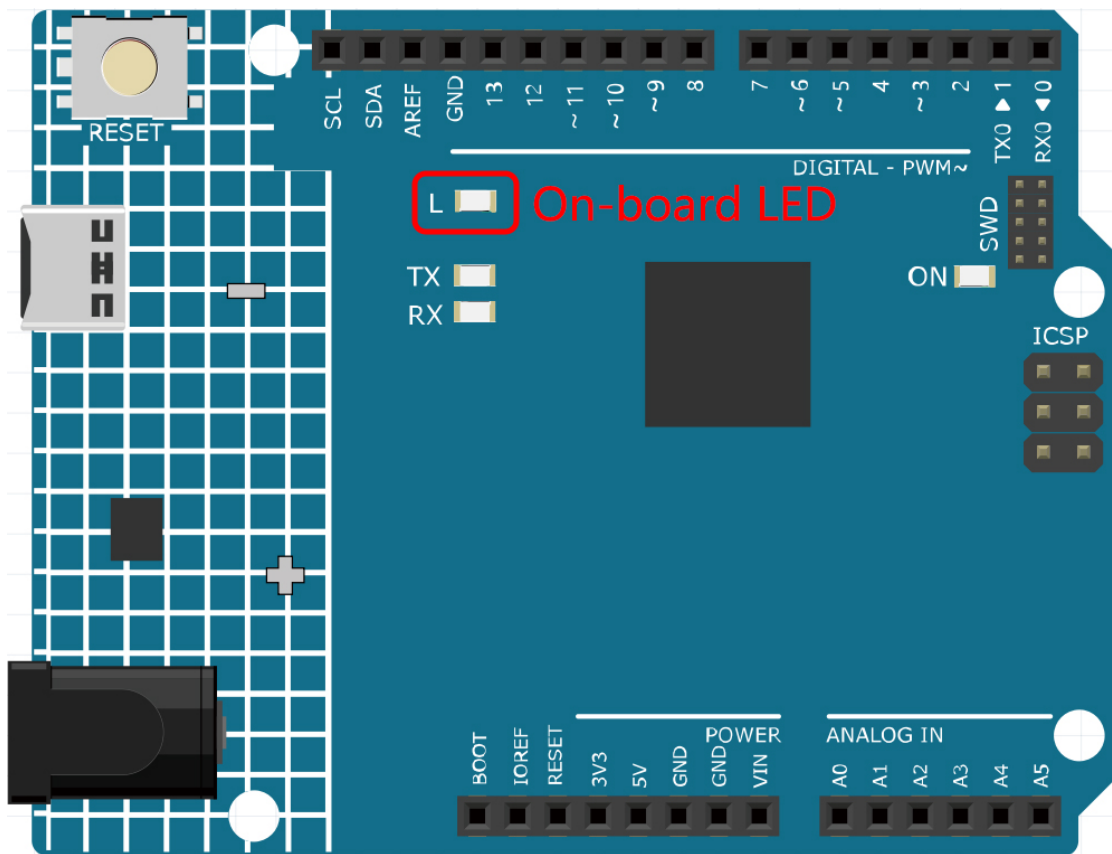
Nach Abschluss der oben genannten Schritte klicken Sie auf die Schaltfläche **Hochladen (Upload)**, um diesen Sketch auf das Board zu laden.



Wenn dies erfolgreich war, sehen Sie die folgende Aufforderung.



Gleichzeitig blinkt die LED auf dem Board.



Das Arduino-Board führt den Sketch automatisch aus, sobald es nach dem Hochladen des Sketchs mit Strom versorgt wird. Das laufende Programm kann durch Hochladen eines neuen Sketches überschrieben werden.

2.3.5 Struktur des Arduino-Programms

Werfen wir einen Blick auf die neue Skizze. Obwohl sie nur wenige Codezeilen enthält, handelt es sich tatsächlich um eine „leere“ Skizze. Wenn Sie diese Skizze auf das Entwicklungsboard hochladen, wird nichts passieren.

```
void setup() {
  // put your setup code here, to run once:
}

void loop() {
  // put your main code here, to run repeatedly:
}
```

Wenn wir `setup()` und `loop()` entfernen und die Skizze zu einer echten blank-Datei machen, werden Sie feststellen, dass sie die Überprüfung nicht besteht. Sie sind das Äquivalent zum menschlichen Skelett und unverzichtbar.

Beim Skizzieren wird `setup()` zuerst ausgeführt, und der darin enthaltene Code (innerhalb von `{}`) wird nach dem Einschalten oder Zurücksetzen des Boards nur einmal ausgeführt. `loop()` wird verwendet, um das Hauptfeature zu schreiben, und der darin enthaltene Code wird in einer Schleife nach der Ausführung von `setup()` ausgeführt.

Um setup() und loop() besser zu verstehen, verwenden wir vier Skizzen. Ihr Ziel ist es, die LED auf dem Arduino-Board blinken zu lassen. Bitte führen Sie jedes Experiment nacheinander durch und notieren Sie die spezifischen Effekte.

- Skizze 1: Lassen Sie die LED auf dem Board kontinuierlich blinken.

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(13,OUTPUT);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    digitalWrite(13,HIGH);  
    delay(500);  
    digitalWrite(13,LOW);  
    delay(500);  
}
```

- Skizze 2: Lassen Sie die LED auf dem Board nur einmal blinken.

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(13,OUTPUT);  
    digitalWrite(13,HIGH);  
    delay(500);  
    digitalWrite(13,LOW);  
    delay(500);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
}
```

- Skizze 3: Lassen Sie die LED auf dem Board einmal langsam blinken und dann schnell blinken.

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(13,OUTPUT);  
    digitalWrite(13,HIGH);  
    delay(1000);  
    digitalWrite(13,LOW);  
    delay(1000);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    digitalWrite(13,HIGH);  
    delay(200);  
    digitalWrite(13,LOW);  
    delay(200);  
}
```

- Skizze 4: Fehlermeldung.

```
void setup() {  
    // put your setup code here, to run once:  
    pinMode(13,OUTPUT);  
}  
  
digitalWrite(13,HIGH);  
delay(1000);  
digitalWrite(13,LOW);  
delay(1000);  
  
void loop() {  
    // put your main code here, to run repeatedly:  
}
```

Mit Hilfe dieser Skizzen können wir mehrere Merkmale von `setup-loop` zusammenfassen.

- `loop()` wird wiederholt ausgeführt, nachdem das Board eingeschaltet wurde.
- `setup()` wird nur einmal ausgeführt, nachdem das Board eingeschaltet wurde.
- Nach dem Einschalten des Boards wird zuerst `setup()` und anschließend `loop()` ausgeführt.
- Der Code muss im `{}`-Bereich von `setup()` oder `loop()` geschrieben werden. Außerhalb des Rahmens wird ein Fehler auftreten.

Bemerkung: Anweisungen wie `digitalWrite(13,HIGH)` dienen zur Steuerung der LED auf dem Board. Ihre genaue Verwendung wird in späteren Kapiteln besprochen.

2.3.6 Skizzieren von Schreibregeln

Wenn Sie einen Freund bitten, das Licht für Sie anzuschalten, können Sie sagen: „Schalte das Licht ein.“ oder „Licht an, Kumpel.“. Sie können jeden beliebigen Tonfall verwenden.

Wenn Sie jedoch möchten, dass das Arduino-Board etwas für Sie erledigt, müssen Sie die Arduino-Programmierschreibregeln befolgen, um die Befehle einzugeben.

Dieses Kapitel enthält die Grundregeln der Arduino-Sprache und wird Ihnen helfen zu verstehen, wie man natürliche Sprache in Code übersetzt.

Natürlich ist dies ein Prozess, mit dem man sich erst mit der Zeit vertraut macht und es ist auch der fehleranfälligste Teil des Prozesses für Anfänger. Also, wenn Sie oft Fehler machen, ist das in Ordnung, versuchen Sie es einfach noch ein paar Mal.

Semikolon ;

Genauso wie beim Schreiben eines Briefes, bei dem Sie am Ende jeden Satzes einen Punkt als Ende setzen, verlangt die Arduino-Sprache von Ihnen, `;` zu verwenden, um dem Board das Ende des Befehls mitzuteilen.

Nehmen wir das bekannte Beispiel „Blinken der Onboard-LED“. Eine fehlerfreie Skizze sollte so aussehen:

Beispiel:


```

void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13,HIGH);
  delay(500);
  digitalWrite(13,LOW);
  delay(500);
}

```

Als Nächstes schauen wir uns die folgenden zwei Skizzen an und raten, ob sie von Arduino korrekt erkannt werden können, bevor wir sie ausführen.

Skizze A:

```

void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13,HIGH)
  delay(500)
  digitalWrite(13,LOW)
  delay(500)
}

```

Skizze B:

```

void setup() {
  // put your setup code here, to run once:
  pinMode(13,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  digitalWrite(13,HIGH);
  delay(500);
  digitalWrite(13,LOW);
  delay(500);
}

```

Das Ergebnis ist, dass **Skizze A** einen Fehler meldet und **Skizze B** läuft.

- Die Fehler in **Skizze A** sind fehlende `;` und obwohl sie normal aussieht, kann Arduino sie nicht lesen.
- **Skizze B** mag für Menschen unübersichtlich wirken, aber in Wirklichkeit gibt es Einrückungen, Zeilenumbrüche und Leerzeichen in Aussagen in Arduino-Programmen nicht. Daher sieht es für den Arduino-Compiler genau so aus wie im Beispiel.

Dennoch, bitte schreiben Sie Ihren Code nicht wie in **Skizze B**, denn in der Regel sind es Menschen, die den Code schreiben und betrachten, also machen Sie sich das Leben nicht unnötig schwer.

Geschweifte Klammern {}

{ } ist eine Hauptkomponente der Arduino-Programmiersprache, und sie müssen immer paarweise auftreten. Eine bessere Programmierkonvention ist, direkt nach dem Tippen der linken geschweiften Klammer die rechte zu tippen und dann den Cursor zwischen die geschweiften Klammern zu setzen, um die Anweisung einzufügen.

Kommentare //

Kommentare sind Teile des Sketches, die vom Compiler ignoriert werden. Sie werden in der Regel verwendet, um anderen zu erklären, wie das Programm funktioniert.

Wenn wir in einer Codezeile zwei aufeinanderfolgende Schrägstriche schreiben, ignoriert der Compiler alles bis zum Ende dieser Zeile.

Wenn wir einen neuen Sketch erstellen, enthält er bereits zwei Kommentare. Werden diese beiden Kommentare entfernt, wird der Sketch in keiner Weise beeinflusst.

```
void setup() {  
    // put your setup code here, to run once:  
  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
  
}
```

Kommentare sind in der Programmierung sehr nützlich, und einige gebräuchliche Verwendungen sind unten aufgeführt.

- Verwendung A: Sich selbst oder anderen mitteilen, was dieser Abschnitt des Codes tut.

```
void setup() {  
    pinMode(13,OUTPUT); //Set pin 13 to output mode, it controls the onboard LED  
}  
  
void loop() {  
    digitalWrite(13,HIGH); // Activate the onboard LED by setting pin 13 high  
    delay(500); // Status quo for 500 ms  
    digitalWrite(13,LOW); // Turn off the onboard LED  
    delay(500); // Status quo for 500 ms  
}
```

- Verwendung B: Temporäres Deaktivieren einiger Anweisungen (ohne sie zu löschen) und ihre Reaktivierung, wenn sie benötigt werden, ohne sie neu schreiben zu müssen. Dies ist beim Debuggen von Code und beim Suchen von Programmfehlern sehr hilfreich.

```
void setup() {  
    pinMode(13,OUTPUT);  
    // digitalWrite(13,HIGH);  
    // delay(1000);  
    // digitalWrite(13,LOW);  
    // delay(1000);  
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
void loop() {
    digitalWrite(13,HIGH);
    delay(200);
    digitalWrite(13,LOW);
    delay(200);
}
```

Bemerkung: Verwenden Sie die Tastenkombination Ctrl+/, um Ihren Code schnell zu kommentieren oder die Kommentierung aufzuheben.

Kommentare /**/

Genauso wie // für Kommentare. Diese Art von Kommentar kann über mehrere Zeilen gehen. Sobald der Compiler /* liest, ignoriert er alles Folgende, bis er auf */ stößt.

Beispiel 1:

```
/* Blink */

void setup() {
    pinMode(13,OUTPUT);
}

void loop() {
    /*
    The following code will blink the onboard LED
    You can modify the number in delay() to change the blinking frequency
    */
    digitalWrite(13,HIGH);
    delay(500);
    digitalWrite(13,LOW);
    delay(500);
}
```

#define

Dies ist ein nützliches Werkzeug in C++.

```
#define Bezeichner Token-String
```

Der Compiler ersetzt automatisch Bezeichner durch Token-String, wenn er diesen liest. Dies wird normalerweise zur Definition von Konstanten verwendet.

Als Beispiel hier ein Sketch, der define verwendet, um die Lesbarkeit des Codes zu verbessern.

```
#define ONBOARD_LED 13
#define DELAY_TIME 500

void setup() {
    pinMode(ONBOARD_LED,OUTPUT);
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
}

void loop() {
    digitalWrite(ONBOARD_LED,HIGH);
    delay(DELAY_TIME);
    digitalWrite(ONBOARD_LED,LOW);
    delay(DELAY_TIME);
}
```

Für den Compiler sieht es tatsächlich so aus:

```
void setup() {
    pinMode(13,OUTPUT);
}

void loop() {
    digitalWrite(13,HIGH);
    delay(500);
    digitalWrite(13,LOW);
    delay(500);
}
```

Man kann erkennen, dass der Bezeichner ersetzt wurde und im Programm nicht vorhanden ist. Daher gibt es einige Vorsichtsmaßnahmen bei der Verwendung.

1. Ein Token-String kann nur manuell geändert werden und kann im Programm nicht durch arithmetische Operationen in andere Werte umgewandelt werden.
2. Vermeiden Sie die Verwendung von Symbolen wie ;. Zum Beispiel:

```
#define ONBOARD_LED 13;

void setup() {
    pinMode(ONBOARD_LED,OUTPUT);
}

void loop() {
    digitalWrite(ONBOARD_LED,HIGH);
}
```

Der Compiler wird es wie folgt erkennen, was als Fehler gemeldet wird:

```
void setup() {
    pinMode(13;,OUTPUT);
}

void loop() {
    digitalWrite(13;,HIGH);
}
```

Bemerkung: Eine Namenskonvention für #define ist, den Bezeichner groß zu schreiben, um Verwechslungen mit Variablen zu vermeiden.

2.3.7 Variable

Die Variable ist eines der mächtigsten und kritischsten Werkzeuge in einem Programm. Sie hilft uns, Daten in unseren Programmen zu speichern und abzurufen.

Die folgende Skizzen-Datei verwendet Variablen. Sie speichert die Pinnummern der Onboard-LED in der Variable `ledPin` und eine Zahl „500“ in der Variable `delayTime`.

```
int ledPin = 13;
int delayTime = 500;

void setup() {
  pinMode(ledPin,OUTPUT);
}

void loop() {
  digitalWrite(ledPin,HIGH);
  delay(delayTime);
  digitalWrite(ledPin,LOW);
  delay(delayTime);
}
```

Wartet, ist das ein Duplikat von dem, was `#define` macht? Die Antwort ist NEIN.

- Die Rolle von `#define` besteht einfach darin, Text direkt zu ersetzen, er wird vom Compiler nicht als Teil des Programms betrachtet.
- Eine Variable hingegen existiert innerhalb des Programms und wird verwendet, um Wert zu speichern und abzurufen. Eine Variable kann ihren Wert auch innerhalb des Programms ändern, was ein `define` nicht tun kann.

Die untenstehende Skizzen-Datei fügt sich selbst der Variable hinzu und lässt die Onboard-LED nach jedem Blinken länger blinken.

```
int ledPin = 13;
int delayTime = 500;

void setup() {
  pinMode(ledPin,OUTPUT);
}

void loop() {
  digitalWrite(ledPin,HIGH);
  delay(delayTime);
  digitalWrite(ledPin,LOW);
  delay(delayTime);
  delayTime = delayTime+200; //Each execution increments the value by 200
}
```

Variable deklarieren

Eine Variable zu deklarieren bedeutet, eine Variable zu erstellen.

Um eine Variable zu deklarieren, benötigt man zwei Dinge: den Datentyp und den Variablennamen. Der Datentyp muss durch ein Leerzeichen von der Variable getrennt werden, und die Variablendeklaration muss mit einem ; beendet werden.

Nehmen wir diese Variable als Beispiel.

```
int delayTime;
```

Datentyp

Hier ist `int` ein Datentyp namens Ganzzahltyp, der verwendet werden kann, um Ganzzahlen von -32768 bis 32766 zu speichern. Er kann auch nicht zum Speichern von Dezimalzahlen verwendet werden.

Variablen können verschiedene Arten von Daten speichern, nicht nur Ganzzahlen. Die Arduino-Sprache (die, erinnern wir uns, C++ ist) hat eingebaute Unterstützung für einige von ihnen (hier sind nur die am häufigsten verwendeten und nützlichsten aufgelistet):

- `float`: Speichert eine Dezimalzahl, zum Beispiel 3,1415926.
- `byte`: Kann Zahlen von 0 bis 255 speichern.
- `boolean`: Speichert nur zwei mögliche Werte, `True` oder `False`, obwohl es einen Byte im Speicher belegt.
- `char`: Speichert eine Zahl von -127 bis 127. Weil es als `char` gekennzeichnet ist, wird der Compiler versuchen, es einem Zeichen aus dem zuzuordnen.
- `string`: Kann eine Zeichenkette speichern, z.B. Halloween.

Variablenname

Sie können die Variable nach Belieben benennen, z.B. `i`, `apple`, `Bruce`, `R2D2`, `Sectumsempra`, aber es gibt einige grundlegende Regeln zu beachten.

1. Beschreiben Sie, wofür sie verwendet wird. Hier habe ich die Variable `delayTime` benannt, so dass man leicht verstehen kann, was sie tut. Es funktioniert gut, wenn ich die Variable `barryAllen` nenne, aber es verwirrt die Person, die den Code ansieht.
2. Verwenden Sie eine gängige Nomenklatur. Sie können CamelCase verwenden, wie ich es getan habe, mit dem Anfangsbuchstaben T in `delayTime`, so dass es leicht zu sehen ist, dass die Variable aus zwei Wörtern besteht. Sie können auch UnderScoreCase verwenden, um die Variable als `delay_time` zu schreiben. Es beeinflusst nicht das Laufen des Programms, aber es würde dem Programmierer helfen, den Code zu lesen, wenn Sie die Nomenklatur verwenden, die Sie bevorzugen.
3. Verwenden Sie keine Schlüsselwörter. Ähnlich wie bei der Eingabe von „`int`“ färbt die Arduino-IDE es, um Sie daran zu erinnern, dass es ein Wort mit einem besonderen Zweck ist und nicht als Variablenname verwendet werden kann. Ändern Sie den Namen der Variable, wenn sie gefärbt ist.
4. Sonderzeichen sind nicht erlaubt. Zum Beispiel: Leerzeichen, #, \$, /, +, % usw. Die Kombination aus englischen Buchstaben (Groß-/Kleinschreibung), Unterstrichen und Zahlen (aber Zahlen können nicht als erstes Zeichen eines Variablennamens verwendet werden) ist reichlich vorhanden.

Einen Wert einer Variable zuweisen

Nachdem wir die Variable deklariert haben, ist es an der Zeit, die Daten zu speichern. Wir verwenden den Zuweisungsoperator (d.h. `=`), um den Wert in die Variable zu setzen.

Wir können der Variable Werte zuweisen, sobald wir sie deklariert haben.

```
int delayTime = 500;
```

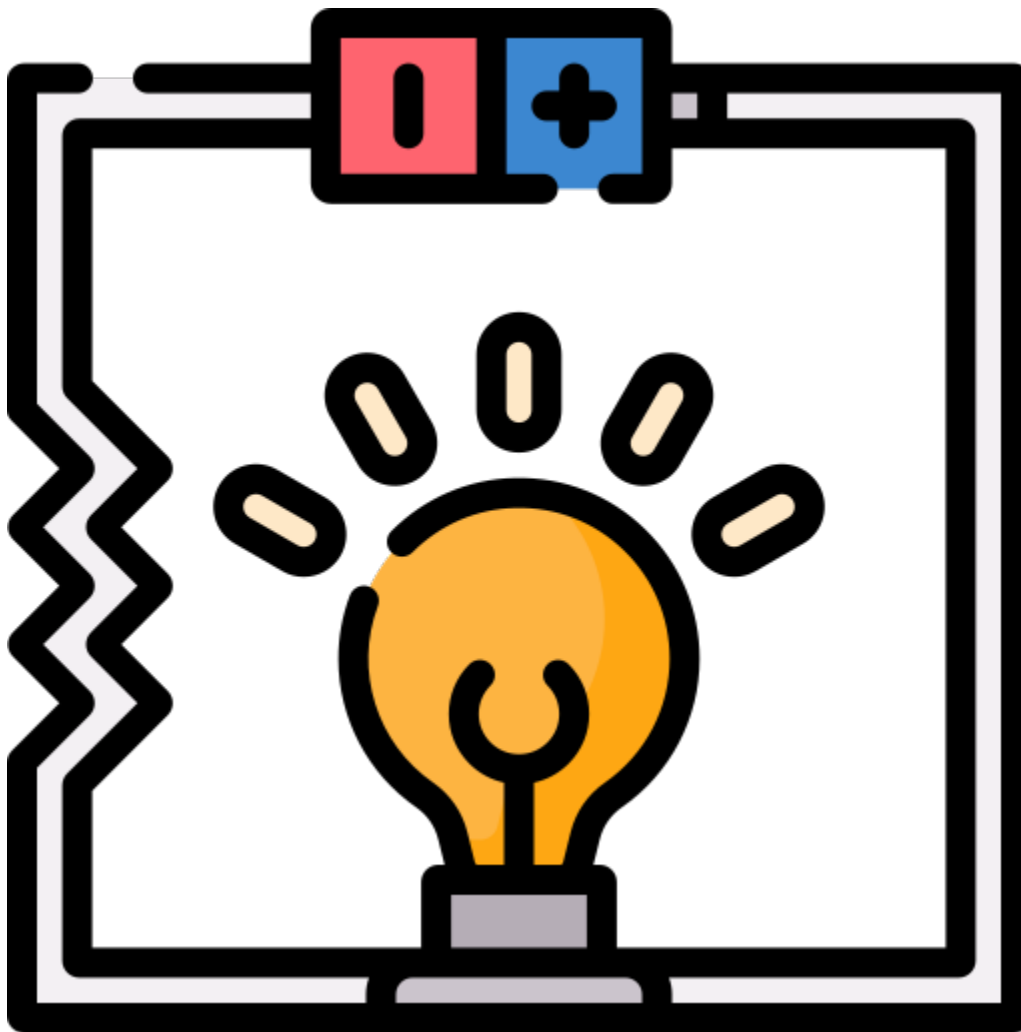
Es ist auch möglich, ihm zu einem späteren Zeitpunkt einen neuen Wert zuzuweisen.

```
int delayTime; // no value  
delayTime = 500; // value is 500  
delayTime = delayTime + 200; // value is 700
```

2.3.8 Wie man den Schaltkreis baut

Viele der Dinge, die Sie täglich verwenden, werden mit Strom betrieben, wie die Lichter in Ihrem Haus und der Computer, auf dem Sie gerade lesen.

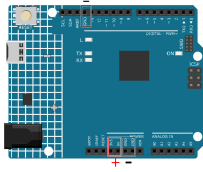
Um Elektrizität zu nutzen, müssen Sie einen elektrischen Schaltkreis bauen. Ein Schaltkreis ist im Grunde ein Pfad, durch den Elektrizität fließt, oder ein elektronischer Schaltkreis, und besteht aus elektrischen Geräten und Komponenten, die in einer bestimmten Weise verbunden sind, wie Widerstände, Kondensatoren, Stromversorgungen und Schalter.



Ein Schaltkreis ist ein geschlossener Pfad, in dem Elektronen fließen, um einen elektrischen Strom zu erzeugen. Um den Strom fließen zu lassen, muss es einen leitenden Pfad zwischen dem positiven und dem negativen Pol der Stromquelle

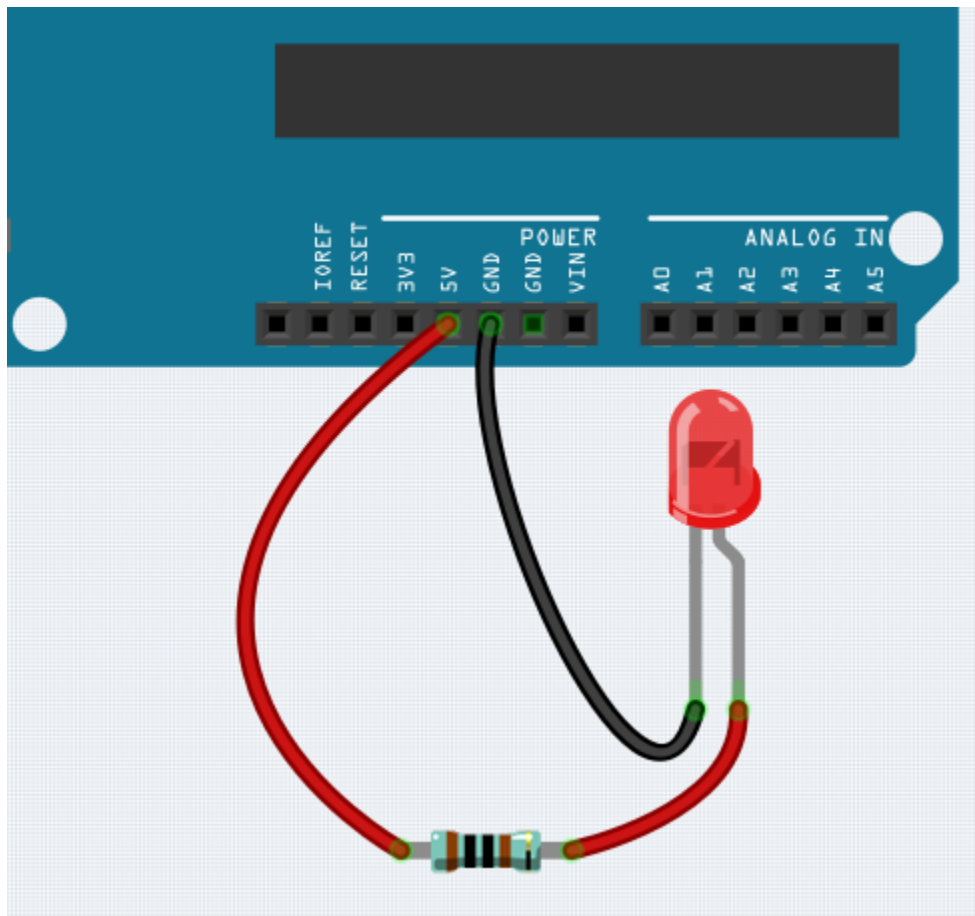
geben, der als geschlossener Schaltkreis bezeichnet wird (wenn er unterbrochen ist, spricht man von einem offenen Schaltkreis).

Das Arduino-Board hat einige Stromausgangspins (positiv) und einige Ground-Pins (negativ). Sie können diese Pins als positive und negative Seiten der Stromquelle verwenden, indem Sie die Stromquelle auf das Board stecken.



Mit Elektrizität können Sie Werke mit Licht, Ton und Bewegung schaffen. Sie können eine LED zum Leuchten bringen, indem Sie den langen Pin an den positiven Pol und den kurzen Pin an den negativen Pol anschließen. Wenn Sie dies tun, wird die LED sehr schnell kaputt gehen, daher müssen Sie einen 1k*-Widerstand in den Schaltkreis einfügen, um ihn zu schützen.

Der von ihnen gebildete Schaltkreis ist unten dargestellt.



Vielleicht fragen Sie sich jetzt: Wie baue ich diesen Schaltkreis? Halte ich die Drähte mit der Hand oder klebe ich die Pins und Drähte fest?

In dieser Situation werden steckbare Steckbretter Ihre stärksten Verbündeten sein.

Hallo, Steckbrett!

Ein Steckbrett ist eine rechteckige Kunststoffplatte mit vielen kleinen Löchern. Diese Löcher ermöglichen es uns, elektronische Komponenten einfach einzufügen und elektronische Schaltkreise aufzubauen. Steckbretter fixieren elektronische Komponenten nicht dauerhaft, sodass wir einen Schaltkreis bei Problemen leicht reparieren und neu starten können.

Bemerkung: Für die Verwendung von Steckbrettern sind keine speziellen Werkzeuge erforderlich. Viele elektronische Komponenten sind jedoch sehr klein, und eine Pinzette kann uns helfen, kleine Teile besser aufzunehmen.

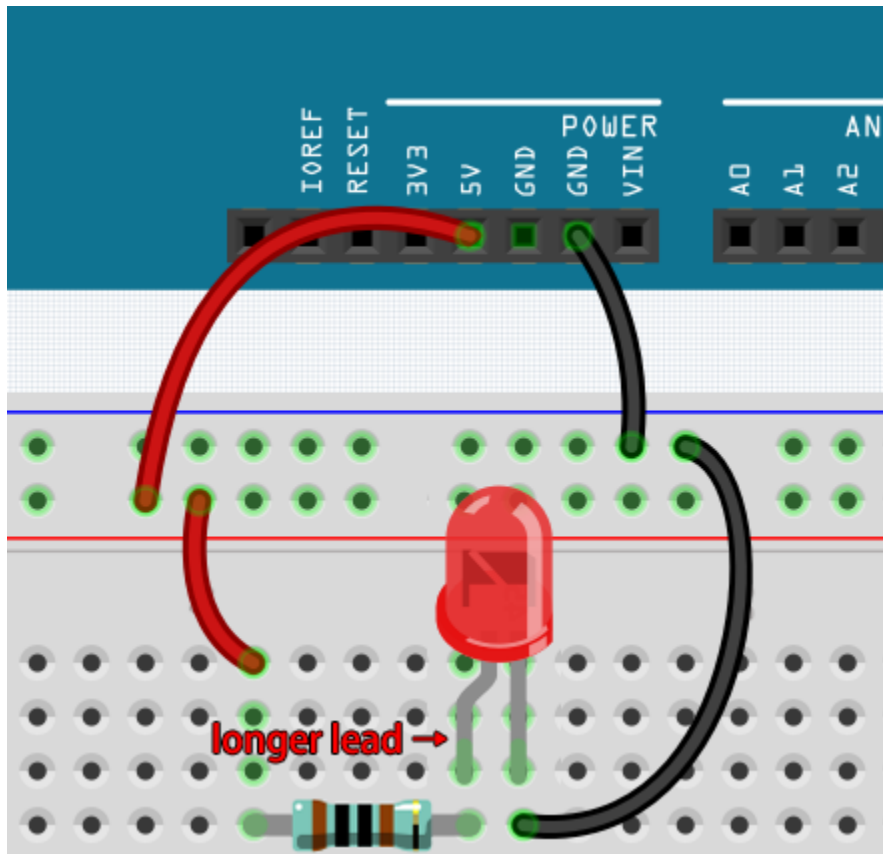
Im Internet finden wir viele Informationen über Steckbretter.

- [Wie man ein Steckbrett benutzt - Science Buddies](#)
- [Was ist ein STECKBRETT? - Makezine](#)

Hier sind einige Dinge, die Sie über Steckbretter wissen sollten.

1. Jede halbe Reihengruppe (wie Spalte A-E in Reihe 1 oder Spalte F-J in Reihe 3) ist verbunden. Daher, wenn ein elektrisches Signal von A1 einfließt, kann es von B1, C1, D1, E1 ausfließen, aber nicht von F1 oder A2.
2. In den meisten Fällen werden beide Seiten des Steckbretts als Stromschienen verwendet, und die Löcher in jeder Spalte (etwa 50 Löcher) sind miteinander verbunden. In der Regel werden positive Stromversorgungen an die Löcher in der Nähe des roten Drahts angeschlossen und negative Stromversorgungen an die Löcher in der Nähe des blauen Drahts.
3. In einem Schaltkreis fließt der Strom vom positiven zum negativen Pol, nachdem er durch die Last geflossen ist. In diesem Fall kann ein Kurzschluss auftreten.

Lasst uns die Richtung des Stroms folgen, um den Schaltkreis aufzubauen!



1. In diesem Schaltkreis verwenden wir den 5V-Pin des Boards, um die LED mit Strom zu versorgen. Verwenden Sie ein Stecker-zu-Stecker (M2M) Jumper-Kabel, um es mit der roten Stromschiene zu verbinden.
2. Um die LED zu schützen, muss der Strom durch einen 1k Ohm-Widerstand fließen. Verbinden Sie ein Ende (beliebiges Ende) des Widerstands mit der roten Stromschiene und das andere Ende mit der freien Reihe des Steckbretts.

Bemerkung: Der Farbring des 1k Ohm-Widerstands ist braun, schwarz, schwarz, braun und braun.

3. Wenn Sie die LED aufnehmen, sehen Sie, dass einer ihrer Anschlüsse länger ist als der andere. Verbinden Sie den längeren Anschluss mit der gleichen Reihe wie der Widerstand und den kürzeren Anschluss mit der anderen Reihe.

Bemerkung: Der längere Anschluss ist die Anode, die die positive Seite des Schaltkreises repräsentiert; der kürzere Anschluss ist die Kathode, die die negative Seite repräsentiert.

Die Anode muss über einen Widerstand mit dem GPIO-Pin verbunden werden; die Kathode muss mit dem GND-Pin verbunden werden.

4. Verwenden Sie ein Stecker-zu-Stecker (M2M) Jumper-Kabel, um den kurzen LED-Pin mit der negativen Stromschiene des Steckbretts zu verbinden.
5. Verbinden Sie den GND-Pin des Boards mit der negativen Stromschiene mit einem Jumper.

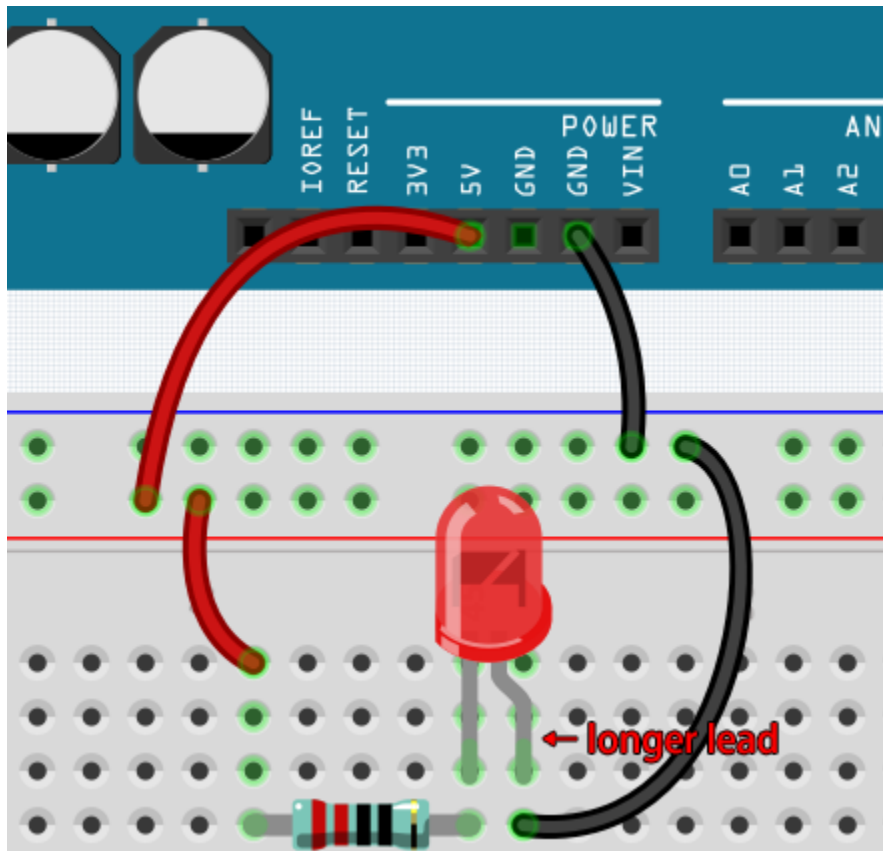
Vorsicht vor Kurzschlüssen

Kurzschlüsse können auftreten, wenn zwei Komponenten, die nicht verbunden sein sollten, „versehentlich“ miteinander verbunden werden. Dieses Kit enthält Widerstände, Transistoren, Kondensatoren, LEDs usw., die lange Metallstifte haben, die aneinander stoßen und einen Kurzschluss verursachen können. Einige Schaltkreise werden einfach daran gehindert, ordnungsgemäß zu funktionieren, wenn ein Kurzschluss auftritt. Gelegentlich kann ein Kurzschluss Komponenten dauerhaft beschädigen, insbesondere zwischen der Stromversorgung und der Masseleiste, wodurch der Schaltkreis sehr heiß wird, das Plastik auf dem Steckbrett schmilzt und sogar die Komponenten verbrennt!

Achten Sie daher immer darauf, dass die Stifte aller Elektronik auf dem Steckbrett sich nicht berühren.

Ausrichtung des Schaltkreises

Schaltkreise haben eine Orientierung, und die Ausrichtung spielt bei bestimmten elektronischen Komponenten eine bedeutende Rolle. Es gibt einige Geräte mit Polarität, was bedeutet, dass sie basierend auf ihren positiven und negativen Polen korrekt verbunden werden müssen. Falsch ausgerichtete Schaltkreise funktionieren nicht ordnungsgemäß.



Wenn Sie die LED in diesem einfachen Schaltkreis, den wir zuvor gebaut haben, umkehren, werden Sie feststellen, dass sie nicht mehr funktioniert.

Im Gegensatz dazu haben einige Geräte keine Richtung, wie die Widerstände in diesem Schaltkreis, so dass Sie versuchen können, sie umzukehren, ohne den normalen Betrieb der LEDs zu beeinflussen.

Die meisten Komponenten und Module mit Bezeichnungen wie „+“, „-“, „GND“, „VCC“ oder mit Stiften unterschiedlicher Länge müssen auf eine bestimmte Weise mit dem Schaltkreis verbunden werden.

Schutz des Schaltkreises

Strom ist die Geschwindigkeit, mit der Elektronen an einem Punkt in einem vollständigen elektrischen Schaltkreis vorbeifließen. Im Grunde genommen entspricht Strom = Fluss. Ein Ampere (AM-pir) oder Amp ist die internationale Einheit zur Messung des Stroms. Es drückt die Menge der Elektronen (manchmal „elektrische Ladung“ genannt) aus, die über einen gegebenen Zeitraum an einem Punkt in einem Schaltkreis vorbeifließt.

Die treibende Kraft (Spannung) hinter dem Stromfluss wird als Spannung bezeichnet und in Volt (V) gemessen.

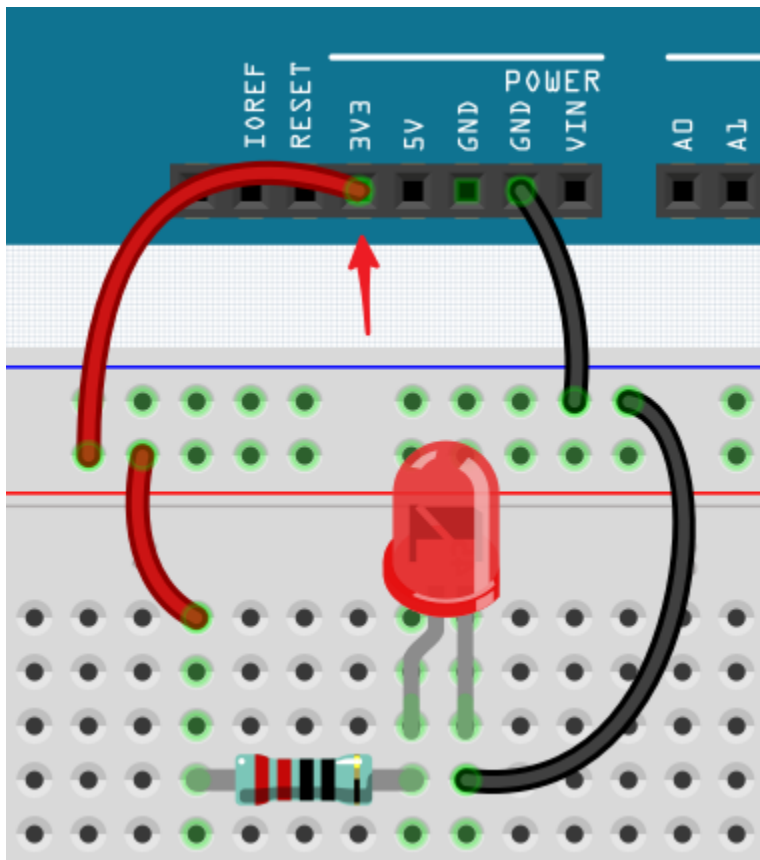
Widerstand (R) ist die Eigenschaft des Materials, die den Stromfluss einschränkt, und er wird in Ohm () gemessen.

Laut Ohmschem Gesetz (solange die Temperatur konstant bleibt) sind Strom, Spannung und Widerstand proportional. Ein Stromkreisstrom ist proportional zu seiner Spannung und umgekehrt proportional zu seinem Widerstand.

Daher Strom (I) = Spannung (V) / Widerstand (R).

- [Ohmsches Gesetz - Wikipedia](#)

Über das Ohmsche Gesetz können wir ein einfaches Experiment durchführen.



Wenn Sie den Draht wechseln, der 5V mit 3,3V verbindet, wird die LED schwächer leuchten. Wenn Sie den Widerstand von 1k Ohm auf 2k Ohm ändern (Farbring: braun, schwarz, schwarz, braun und braun), werden Sie bemerken, dass die LED schwächer leuchtet als zuvor. Je größer der Widerstand, desto schwächer die LED.

Bemerkung: Für eine Einführung in Widerstände und wie man Widerstandswerte berechnet, siehe [Widerstand](#).

Die meisten verpackten Module benötigen nur Zugriff auf die richtige Spannung (normalerweise 3,3V oder 5V), wie das Ultraschallmodul.

In Ihren selbstgebauten Schaltungen müssen Sie jedoch auf die Versorgungsspannung und den Widerstandsgebrauch für elektrische Geräte achten.

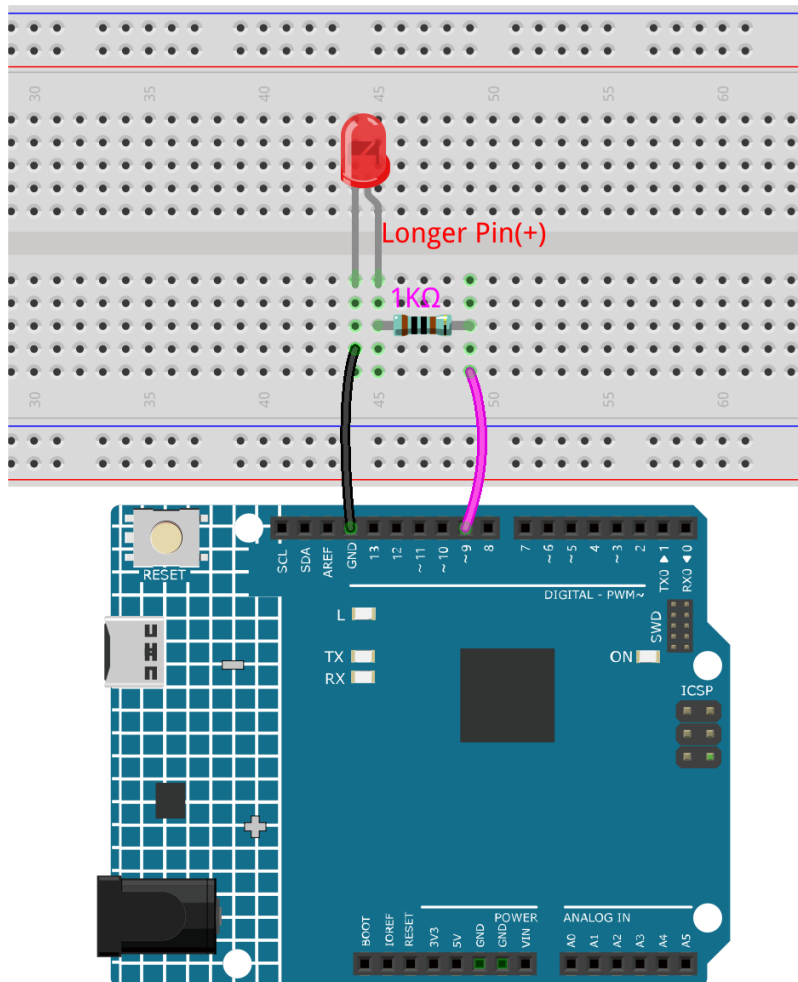
Zum Beispiel verbrauchen LEDs normalerweise 20mA Strom, und ihr Spannungsabfall beträgt etwa 1,8V. Nach dem Ohmschen Gesetz müssen wir, wenn wir eine 5V Stromversorgung verwenden, einen Widerstand von mindestens 160 Ohm $((5-1,8)/20\text{mA})$ in den Schaltkreis einfügen, um die LED nicht zu verbrennen.

Arduino zur Schaltkreissteuerung nutzen

Jetzt, da wir ein grundlegendes Verständnis für Arduino-Programmierung und elektronische Schaltkreise haben, ist es an der Zeit, sich der wichtigsten Frage zu widmen: Wie steuert man Schaltkreise mit Arduino?

Einfach ausgedrückt, steuert Arduino einen Schaltkreis, indem es das Niveau der Pins auf dem Board ändert. Zum Beispiel steuert es eine onboard LED, indem es ein hohes oder niedriges Signal an Pin 13 schreibt.

Versuchen wir nun, das Arduino-Board so zu programmieren, dass die LED auf dem Steckbrett blinkt. Bauen Sie den Schaltkreis so, dass die LED an Pin 9 angeschlossen ist.



Laden Sie anschließend diesen Sketch auf das Arduino-Entwicklungsboard hoch.

```
int ledPin = 9;
int delayTime = 500;
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
void setup() {  
    pinMode(ledPin,OUTPUT);  
}  
  
void loop() {  
    digitalWrite(ledPin,HIGH);  
    delay(delayTime);  
    digitalWrite(ledPin,LOW);  
    delay(delayTime);  
}
```

Dieser Sketch ähnelt sehr dem, den wir verwendet haben, um das Blinken der onboard LED zu steuern. Der Unterschied ist, dass der Wert von `ledPin` nun auf 9 geändert wurde. Das liegt daran, dass wir dieses Mal versuchen, das Niveau von Pin 9 zu steuern.

Jetzt können Sie sehen, dass die LED auf dem Steckbrett blinkt.

KAPITEL 3

Code herunterladen

Laden Sie den entsprechenden Code über den untenstehenden Link herunter.

- SunFounder 3 in 1 Kit für Arduino
- Oder schauen Sie sich den Code auf [SunFounder 3 in 1 Kit für Arduino - GitHub](#) an.

In diesem Kapitel lernen Sie, wie Sie mit Arduino elektronische Schaltungen steuern können.

Je nach den Komponenten können die grundlegenden Steuerungsmethoden von Arduino in vier Typen unterteilt werden:

- *1. Digital Write*: Setzt die Ausgangsspannung des Pins auf hoch oder niedrig, womit z.B. das Licht ein- und ausgeschaltet werden kann.
- *2. Analog schreiben*: Schreibt den analogen Wert (*PWM-Welle*) auf den Pin, um die Helligkeit des Lichts zu regulieren.
- *3. Digital Lesen*: Liest das Pegelsignal des digitalen Pins aus, was zum Ablesen des Arbeitszustands des Schalters verwendet werden kann.
- *4. Analoges Lesen*: Liest die Spannung des analogen Pins aus, wodurch der Arbeitszustand eines Knopfes abgelesen werden kann.

Es gibt auch einige Komponenten, die zusätzliche Bibliotheken für ihre Verwendung benötigen. Diese sind im Abschnitt *5.11 Externe Bibliotheken installieren* zusammengefasst.

Schließlich bietet das Kit auch einige *6. Lustiges Projekt*, die viele einfache und nützliche Manipulationen beinhalten. Probieren Sie diesen Abschnitt des Codes aus, und Sie werden verstehen, wie die meisten einfachen Projekte funktionieren.

4.1 1. Digital Write

Digital Write bedeutet, ein digitales Signal an einen digitalen Pin auszugeben. Das digitale Signal hat nur zwei Zustände, 0 oder 1, 0V oder 5V, so dass es einigen Komponenten, wie z.B. der LED und dem Summer, ermöglicht, ein- oder ausgeschaltet zu werden.

Auf dem Arduino R4-Board gibt es 14 digitale I/O-Pins von 0 bis 13. Jetzt verwenden Sie die Funktionen `pinMode()` und `digitalWrite()` um ein hohes oder niedriges Signal an diese digitalen Pins zu schreiben.

- `pinMode(pin, mode)`: Konfigurieren Sie den bestimmten Pin als INPUT oder OUTPUT. Hier muss er als OUTPUT eingestellt werden.

Syntax

pinMode(pin, mode)

Parameter

- pin: Die Pin-Nummer von Arduino, für die der Modus eingestellt werden soll.
 - mode: INPUT, OUTPUT oder INPUT_PULLUP.
- digitalWrite(pin, value): Schreiben Sie ein hohes Signal (5V) oder ein niedriges Signal (0V) an einen digitalen Pin, um den Betriebszustand der Komponente zu ändern. Wenn der Pin mit pinMode() als OUTPUT konfiguriert wurde, wird seine Spannung auf den entsprechenden Wert eingestellt: 5V (oder 3,3V bei 3,3V Boards) für HIGH, 0V (Masse) für LOW.

Syntax

digitalWrite(pin, value)

Parameter

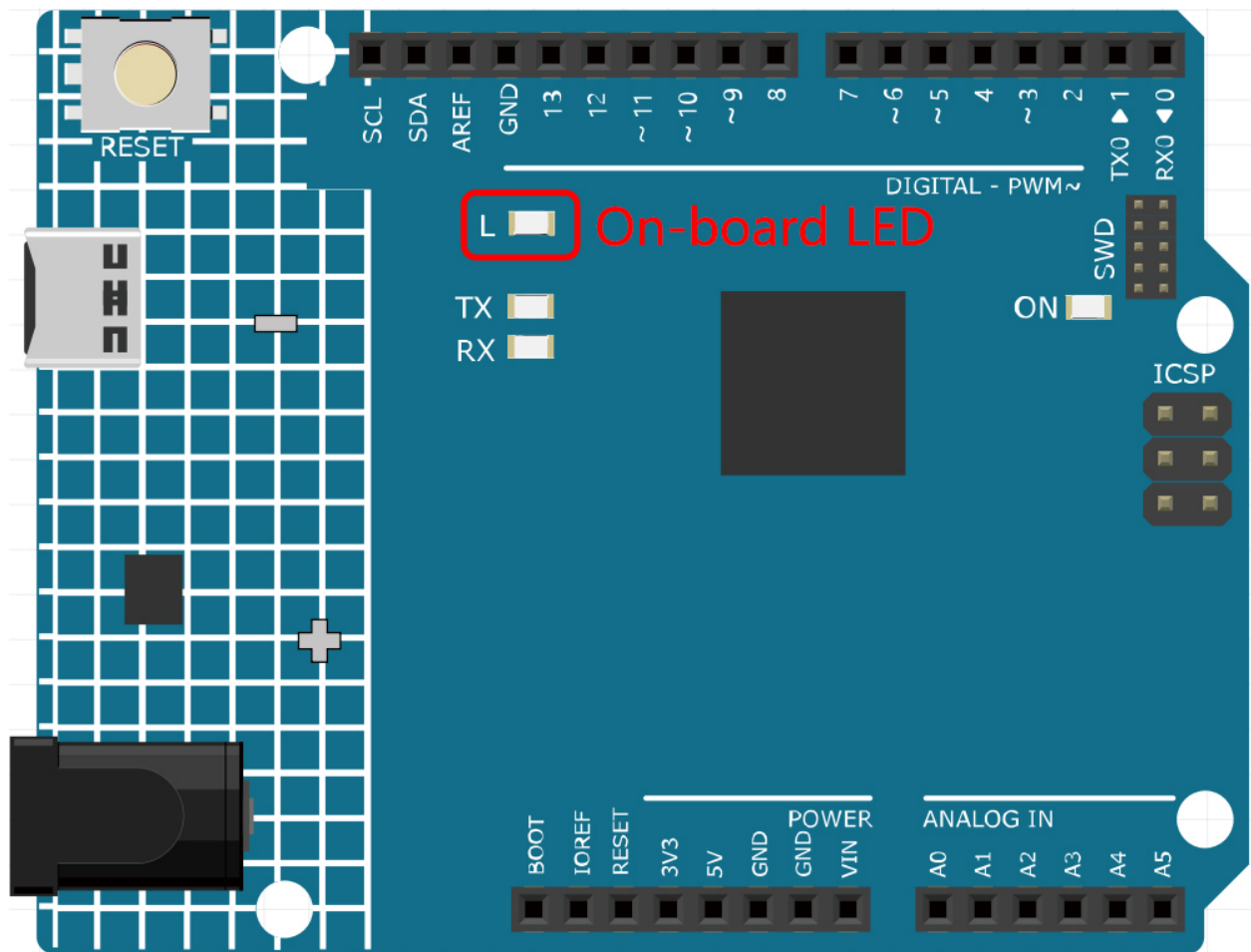
- pin: Die Pin-Nummer von Arduino.
- value: HIGH oder LOW.

Beispiel für Digital Write:

```
const int pin = 13;

void setup() {
  pinMode(pin, OUTPUT);    // sets the digital pin as output
}

void loop() {
  digitalWrite(pin, HIGH); // sets the digital pin on
  delay(1000);             // waits for a second
  digitalWrite(pin, LOW);  // sets the digital pin off
  delay(1000);             // waits for a second
}
```



Hinweise und Warnungen

- Die Pins 0~13 sind alle digitale Pins.
- Verwenden Sie nicht die Pins 0 und 1, da sie zur Kommunikation mit dem Computer verwendet werden. Eine Verbindung zu diesen Pins würde die Kommunikation stören, einschließlich eines fehlgeschlagenen Hochladens auf das Board.
- Wenn die digitalen Pins verbraucht sind, können die analogen Pins (A0-A5) auch als digitale Pins verwendet werden.

Zugehörige Komponenten

Unten finden Sie die zugehörigen Komponenten. Sie können darauf klicken, um zu erfahren, wie man sie verwendet.

4.1.1 1.1 Hallo, LED!

Genauso wie das Ausdrucken von „Hallo, Welt!“ der erste Schritt beim Erlernen des Programmierens ist, so ist das Verwenden eines Programms zum Ansteuern einer LED die traditionelle Einführung in das Erlernen der physischen Programmierung.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

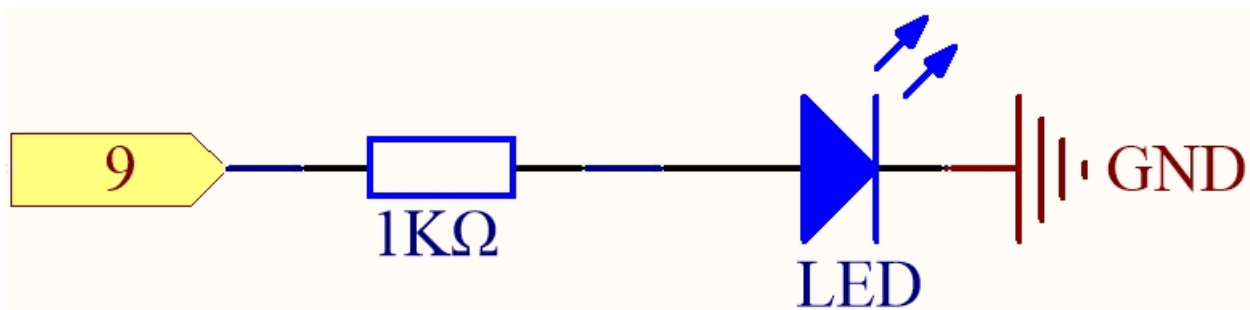
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

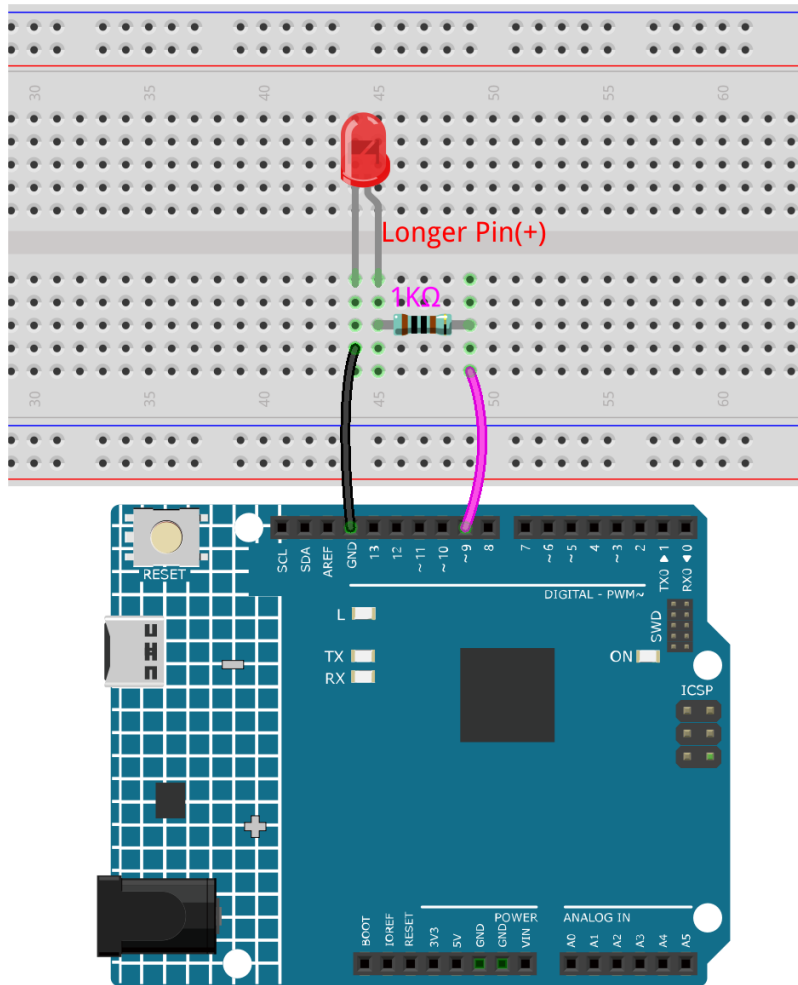
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	

Schaltplan



Das Prinzip dieser Schaltung ist einfach, und die Stromrichtung ist in der Abbildung dargestellt. Wenn Pin 9 ein hohes Signal (5V) ausgibt, leuchtet die LED nach dem 1k Ohm-Strombegrenzungswiderstand auf. Wenn Pin 9 ein niedriges Signal (0V) ausgibt, erlischt die LED.

Verdrahtung



Code

Bemerkung:

- Sie können die Datei 1.1.hello_led.ino unter dem Pfad 3in1-kit\learning_project\1.1.hello_led öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem der Code erfolgreich hochgeladen wurde, sehen Sie die LED blinken.

Wie funktioniert das?

Hier verbinden wir die LED mit dem digitalen Pin 9, daher müssen wir zu Beginn des Programms eine int-Variable namens ledPin deklarieren und ihr den Wert 9 zuweisen.

```
const int ledPin = 9;
```

Jetzt initialisieren Sie den Pin in der setup()-Funktion, wo Sie den Pin auf den Modus OUTPUT setzen müssen.

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

In `loop()` wird `digitalWrite()` verwendet, um ein 5V-Hochpegelsignal für `ledpin` bereitzustellen, was zu einem Spannungsunterschied zwischen den LED-Pins führt und die LED zum Leuchten bringt.

```
digitalWrite(ledPin, HIGH);
```

Wenn das Pegelsignal auf LOW geändert wird, wird das Signal von `ledPin` auf 0 V zurückgesetzt, um die LED auszuschalten.

```
digitalWrite(ledPin, LOW);
```

Es wird ein Intervall zwischen Ein- und Ausschalten benötigt, damit die Menschen die Änderung sehen können. Daher verwenden wir den Code `delay(1000)`, um den Controller 1000 ms lang nichts tun zu lassen.

```
delay(1000);
```

4.1.2 1.2 Piepton

Der aktive Summer ist ein typisches digitales Ausgabegerät, das genauso einfach zu verwenden ist wie das Anzünden einer LED!

In dem Kit sind zwei Arten von Summern enthalten. Wir müssen den aktiven Summer verwenden. Drehen Sie sie um, der versiegelte Rücken (nicht die freiliegende PCB) ist der, den wir wollen.



Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

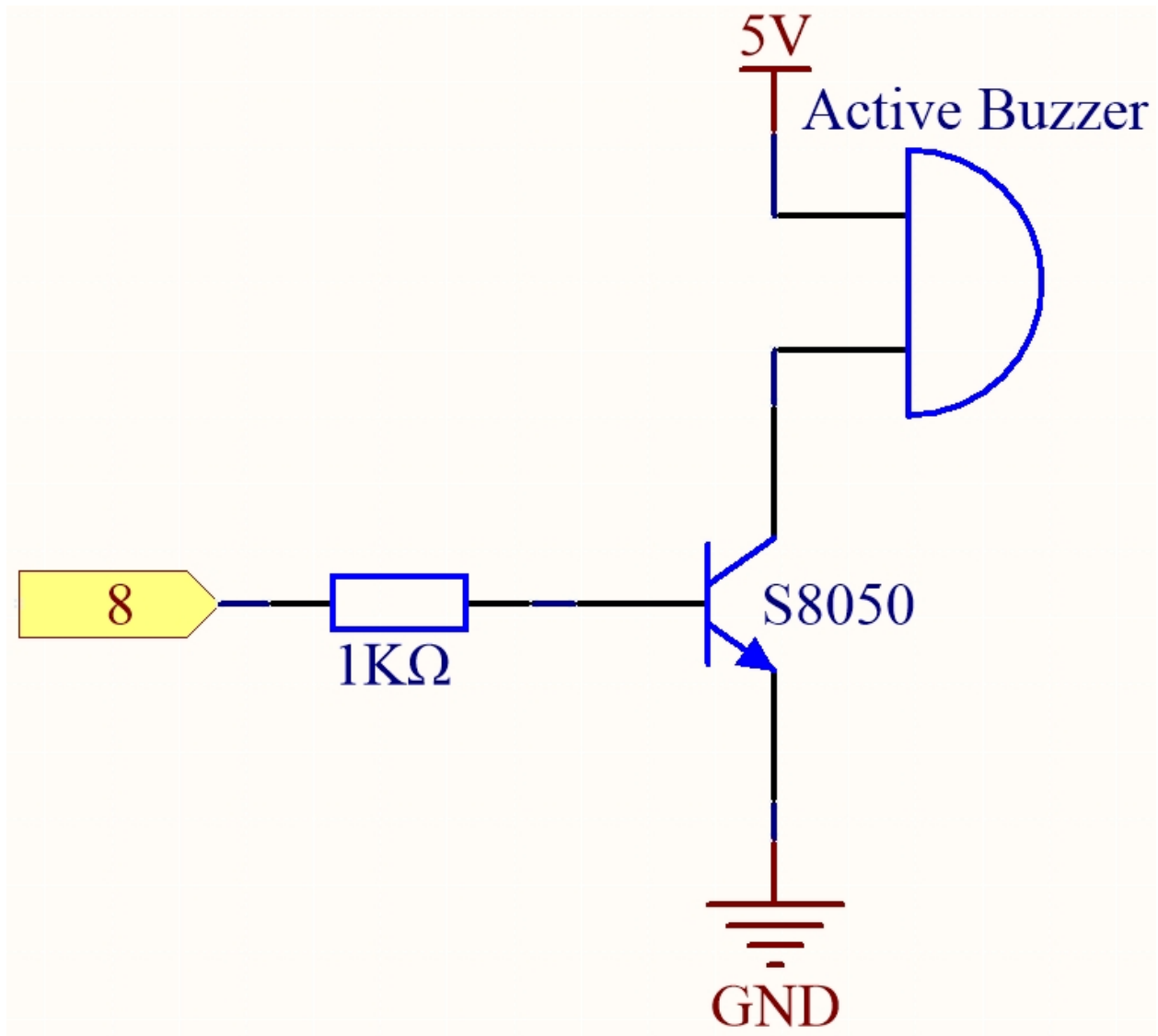
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

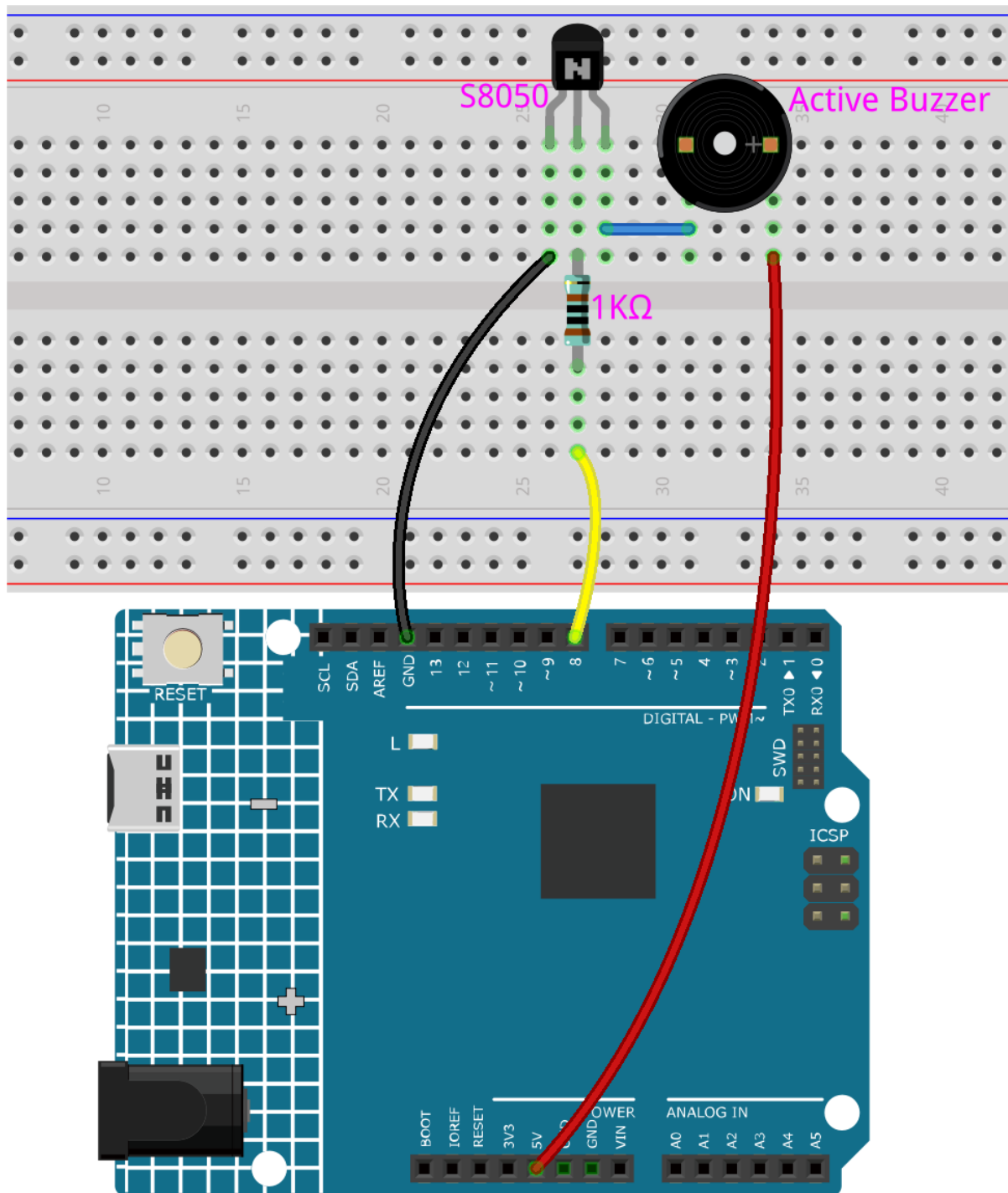
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Summer</i>	-

Schaltplan



Verdrahtung



Code

Bemerkung:

- Sie können die Datei 1.2.beep.ino unter dem Pfad 3in1-kit\learning_project\1.2.beep öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem der Code erfolgreich hochgeladen wurde, hören Sie jede Sekunde einen Piepton.

4.1.3 1.3 Das Rad drehen

Ein Motor ist ein typisches digitales Ausgabegerät und wird genauso verwendet wie eine LED. Der Motor muss jedoch mit einem hohen Strom betrieben werden, und der hohe Strom kann das Hauptsteuerungsboard wie das R4-Board beschädigen. Deshalb wird bei dieser Gelegenheit ein Motor-Treibermodul verwendet, welches eine große Hilfe für das R4-Board ist, um den Motor sicher zu steuern.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

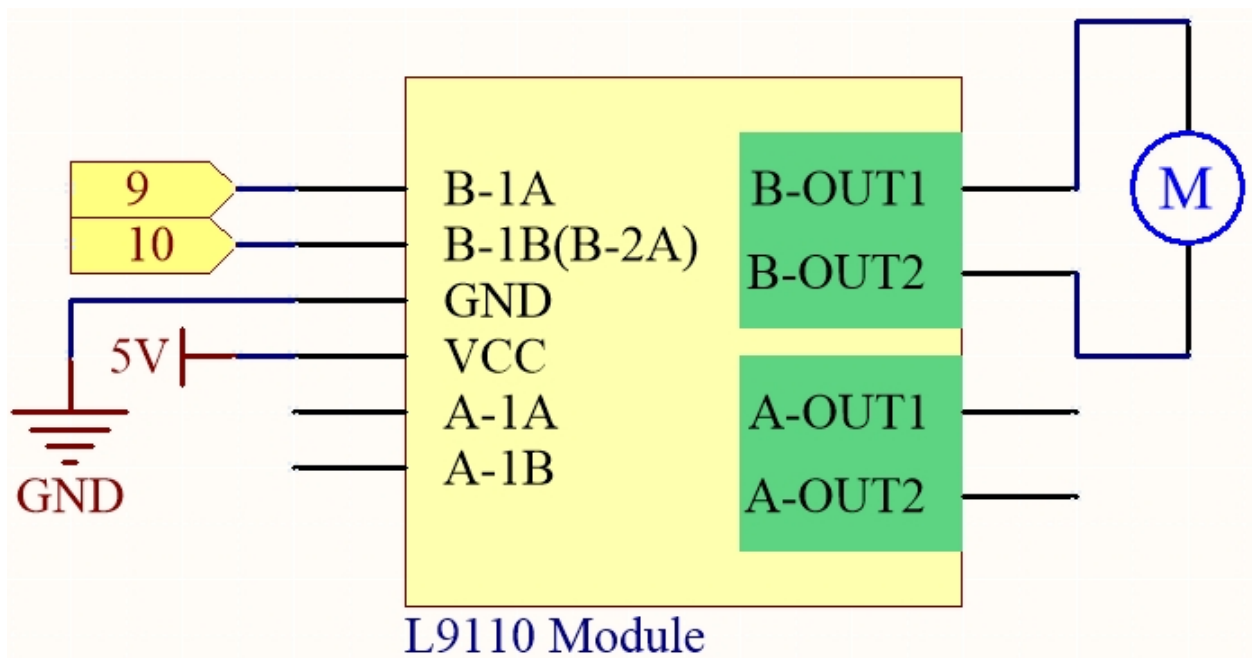
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

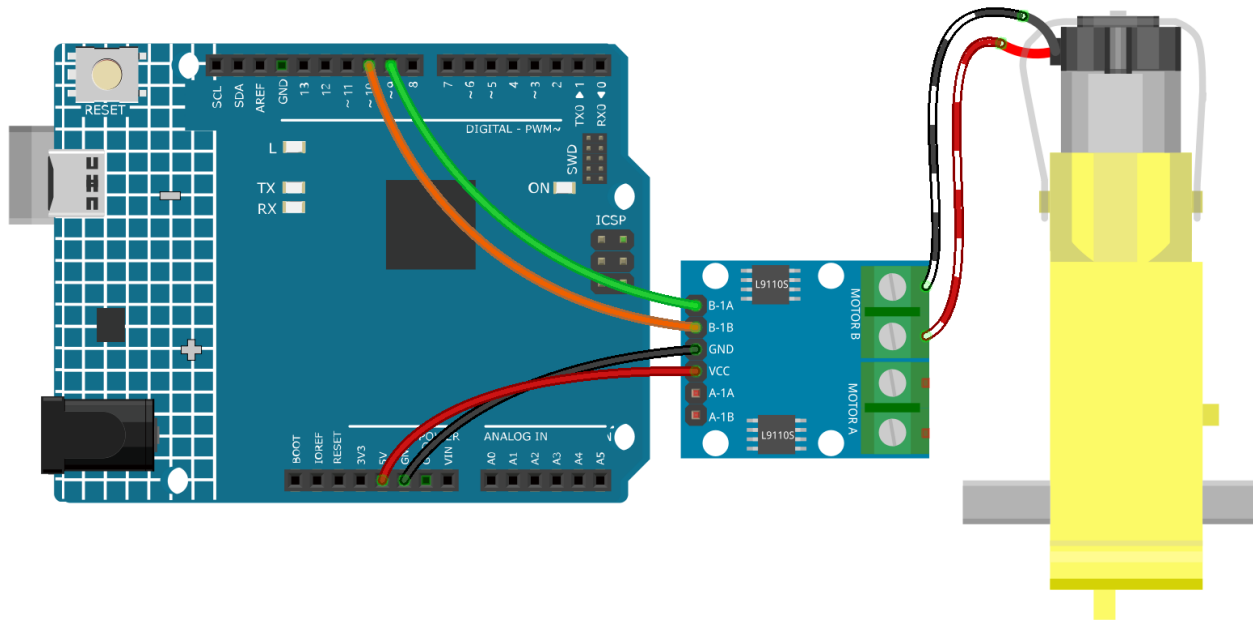
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Jumper-Kabel</i>	
<i>TT Motor</i>	-
<i>L9110 Motortreibermodul</i>	-

Schaltplan



Verdrahtung

L9110S	R4-Board	Motor
VCC	5V	
GND	GND	
B-1B(B-2A)	9	
B-1A	10	
OA		ein Draht des Motors
OB		ein Draht des Motors



Code

Bemerkung:

- Sie können die Datei 1.3.turn_the_wheel.ino unter dem Pfad 3in1-kit\learning_project\1.3.turn_the_wheel öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

4.1.4 1.4 Pumpen

Die Wasserpumpe ist ebenfalls ein Motor, der durch eine spezielle Struktur die mechanische Energie des Motors oder eine andere externe Energie nutzt, um Flüssigkeit zu transportieren.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

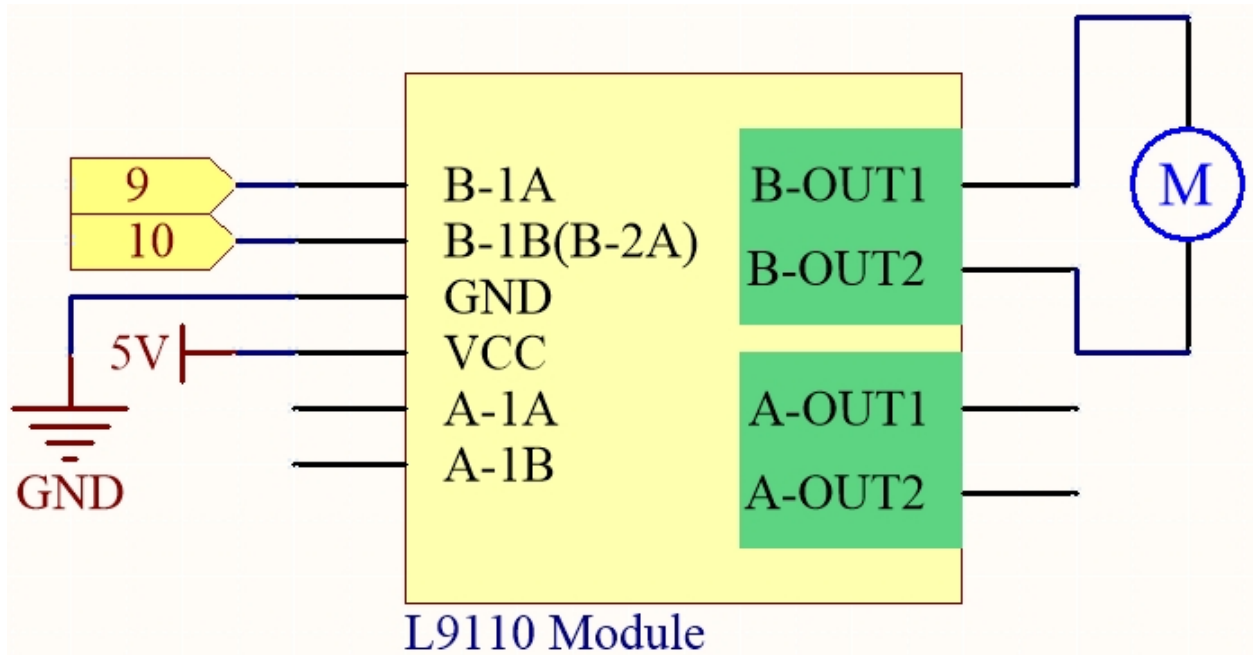
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

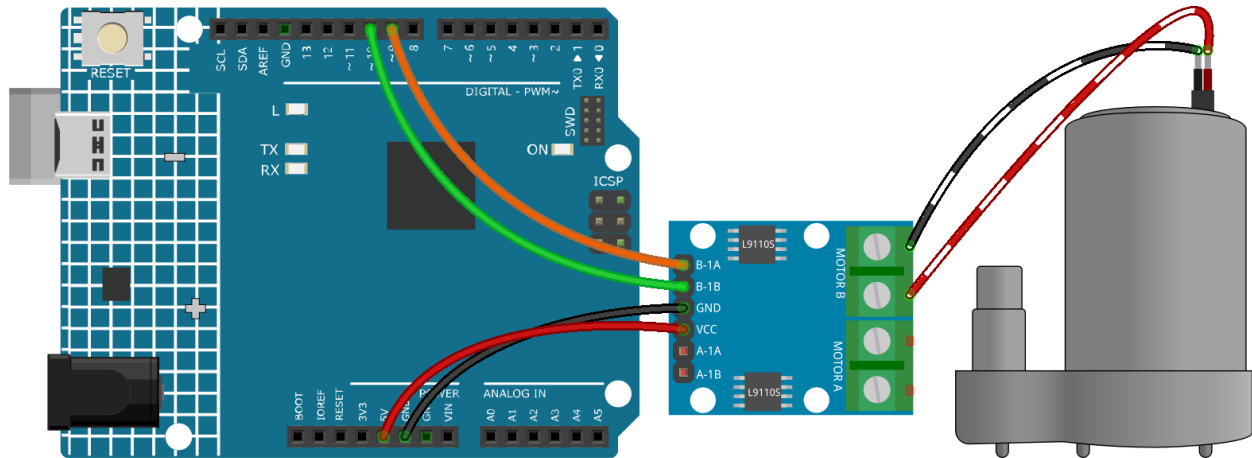
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Jumper-Kabel</i>	
<i>L9110 Motortreibermodul</i>	-
<i>Kreiselpumpe</i>	-

Schaltplan



Verdrahtung

L9110S	R4-Board	Motor
VCC	5V	
GND	GND	
B-1B(B-2A)	9	
B-1A	10	
OA		ein Draht des Motors
OB		ein Draht des Motors



Code

Bemerkung:

- Sie können die Datei `1.4.pumping.ino` unter dem Pfad `3in1-kit\learning_project\1.4.pumping` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Fügen Sie den Schlauch zur Pumpe hinzu und platzieren Sie ihn im Becken. Nachdem der Code erfolgreich hochgeladen wurde, werden Sie sehen, dass das Wasser im Becken nach einer Weile abgelassen wird. Während dieses Experiments halten Sie bitte die Schaltung vom Wasser fern, um einen Kurzschluss zu vermeiden!

4.2 2. Analog schreiben

6 der 14 digitalen Pins des Arduino verfügen auch über eine PWM-Ausgangsfunktion. Daher können Sie diesen 6 Pins neben digitalen Signalen auch analoge Signale (PWM-Wellensignale) schreiben. Auf diese Weise können Sie die LEDs in unterschiedlichen Helligkeiten leuchten lassen oder den Motor mit unterschiedlichen Geschwindigkeiten drehen lassen.

Pulsweitenmodulation oder **PWM** ist eine Technik, um analoge Ergebnisse mit digitalen Mitteln zu erzielen. Da der buchstäbliche Sinn vielleicht schwer zu erfassen ist, folgt ein Beispiel zur Steuerung der Intensität einer LED, um das Konzept besser zu veranschaulichen.

Ein digitales Signal, das aus hohen und niedrigen Pegeln besteht, wird als Puls bezeichnet. Die Pulsbreite dieser Pins kann durch Änderung der EIN/AUS-Geschwindigkeit angepasst werden. Vereinfacht ausgedrückt: Wenn wir die LED kurz ein- und ausschalten (z.B. 20ms, die visuelle Verweilzeit der meisten Menschen), werden wir nicht bemerken, dass sie ausgegangen ist, aber die Helligkeit des Lichts wird etwas schwächer sein. Während dieser Zeit, je länger die LED eingeschaltet ist, desto heller wird sie sein. Das bedeutet, innerhalb einer Periode, je breiter der Puls, desto größer die „elektrische Signalstärke“, die vom Mikrocontroller ausgegeben wird.

Dies ist die Funktion, die benötigt wird, um die PWM-Welle zu schreiben.

- `analogWrite(pin, value)`

Schreibt einen analogen Wert (PWM-Welle) auf einen Pin. Unterschiedliche Ausgangsspannungen (0-5V) können durch Erzeugung eines bestimmten Pulssignals simuliert werden. Der Pin behält dieses Signal bei, bis er durch eine neue Lese- oder Schreibanweisung aufgerufen wird.

Syntax

```
analogWrite(pin, value)
```

Parameters

- pin: Der Arduino-Pin, auf den geschrieben werden soll. Erlaubte Datentypen: int.
- value: Der Tastgrad: zwischen 0 (immer aus) und 255 (immer an). Erlaubte Datentypen: int.

Beispiel für Analog schreiben

```
int pin = 9;           //connect to pwm pin

void setup() {
  pinMode(pin, OUTPUT); // sets the pin as output
}

void loop() {
  for (int i = 0 ;i<255 ; i++){
    analogWrite(pin, i); //analogWrite values from 0 to 255
    delay(30);
  }
}
```

Hinweise und Warnungen

- Wenn Sie genau auf das R4-Board schauen, werden Sie feststellen, dass die Pins mit dem „~“-Symbol über eine analoge Ausgangsfunktion verfügen.
- Die auf den Pins 5 und 6 erzeugten PWM-Ausgänge werden höhere als erwartete Tastverhältnisse aufweisen. Dies liegt an Wechselwirkungen mit den Funktionen millis() und delay(), die denselben internen Timer verwenden, der zur Erzeugung dieser PWM-Ausgänge verwendet wird. Dies wird vor allem bei niedrigen Tastverhältniseinstellungen (z.B. 0 - 10) bemerkt und kann dazu führen, dass ein Wert von 0 den Ausgang auf den Pins 5 und 6 nicht vollständig ausschaltet.

Verwandte Komponenten

Unten sind die verwandten Komponenten aufgeführt. Sie können darauf klicken, um zu erfahren, wie sie verwendet werden.

4.2.1 2.1 Verblassen

Dieses Projekt ähnelt [1.1 Hallo, LED!](#). Der Unterschied liegt in der Signalart. Während zuvor die LED durch Ausgabe eines digitalen Signals (0&1) ein- oder ausgeschaltet wurde, wird in diesem Projekt die Helligkeit der LED durch Ausgabe eines analogen Signals gesteuert.

Benötigte Komponenten

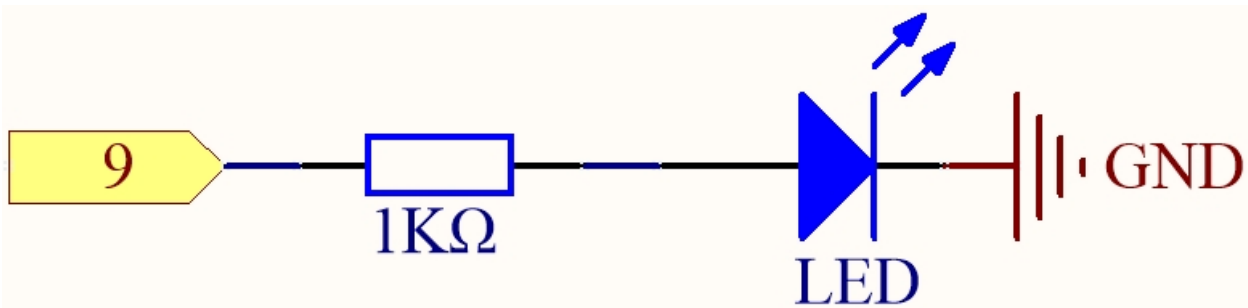
Für dieses Projekt benötigen wir die folgenden Komponenten.

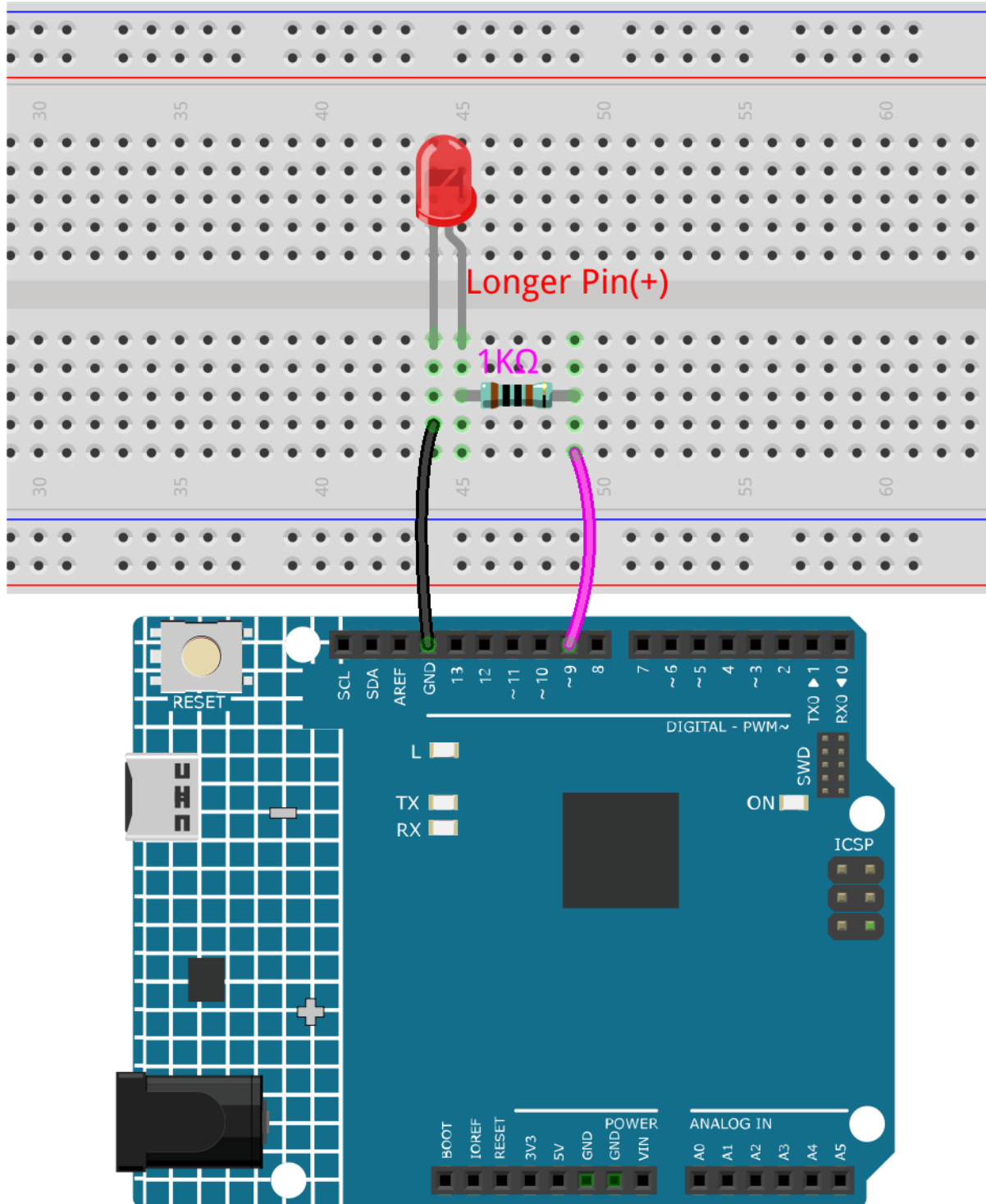
Es ist definitiv praktisch, ein ganzes Kit zu kaufen. Hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die folgenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	

Schaltplan**Verdrahtung**



Code

Bemerkung:

- Sie können die Datei 2.1.fading.ino im Pfad 3in1-kit\learning_project\2.analogWrite\2.1.fading öffnen.

- Oder kopieren Sie diesen Code in die **Arduino IDE**.
-

Nachdem der Code erfolgreich hochgeladen wurde, können Sie die LED atmen sehen.

4.2.2 2.2 Bunttes Licht

Wie wir wissen, kann Licht überlagert werden. Beispielsweise ergibt die Kombination von blauem Licht und grünem Licht cyanfarbendes Licht, rotes Licht und grünes Licht ergeben gelbes Licht. Dies wird als „Das additive Farbmischverfahren“ bezeichnet.

- [Additive Farbmischung - Wikipedia](#)

Basierend auf dieser Methode können wir die drei Grundfarben verwenden, um das sichtbare Licht jeder Farbe gemäß unterschiedlicher spezifischer Gewichtungen zu mischen. Zum Beispiel kann Orange erzeugt werden, indem man mehr Rot und weniger Grün verwendet.

In diesem Kapitel werden wir die RGB-LED verwenden, um das Geheimnis der additiven Farbmischung zu erkunden!

Eine RGB-LED entspricht der Kapselung einer roten LED, einer grünen LED und einer blauen LED unter einem Lampenschirm, wobei die drei LEDs einen gemeinsamen Kathodenpin teilen. Da jedem Anodenpin das elektrische Signal bereitgestellt wird, kann das Licht der entsprechenden Farbe angezeigt werden. Durch Änderung der elektrischen Signalintensität jeder Anode können verschiedene Farben erzeugt werden.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

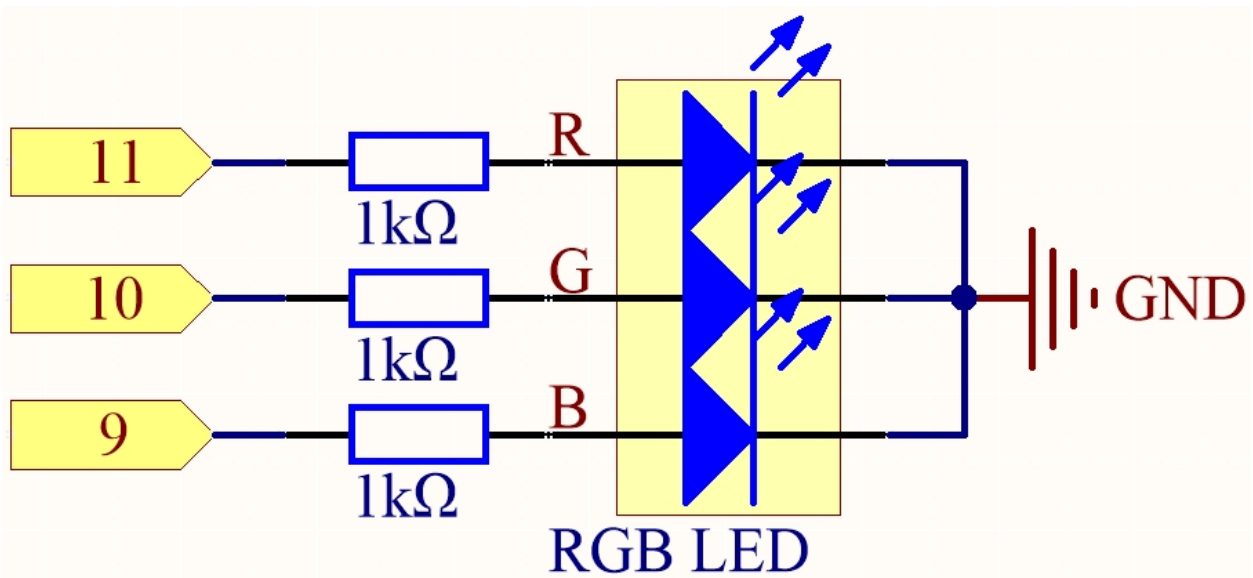
Es ist definitiv praktisch, ein ganzes Kit zu kaufen. Hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die folgenden Links kaufen.

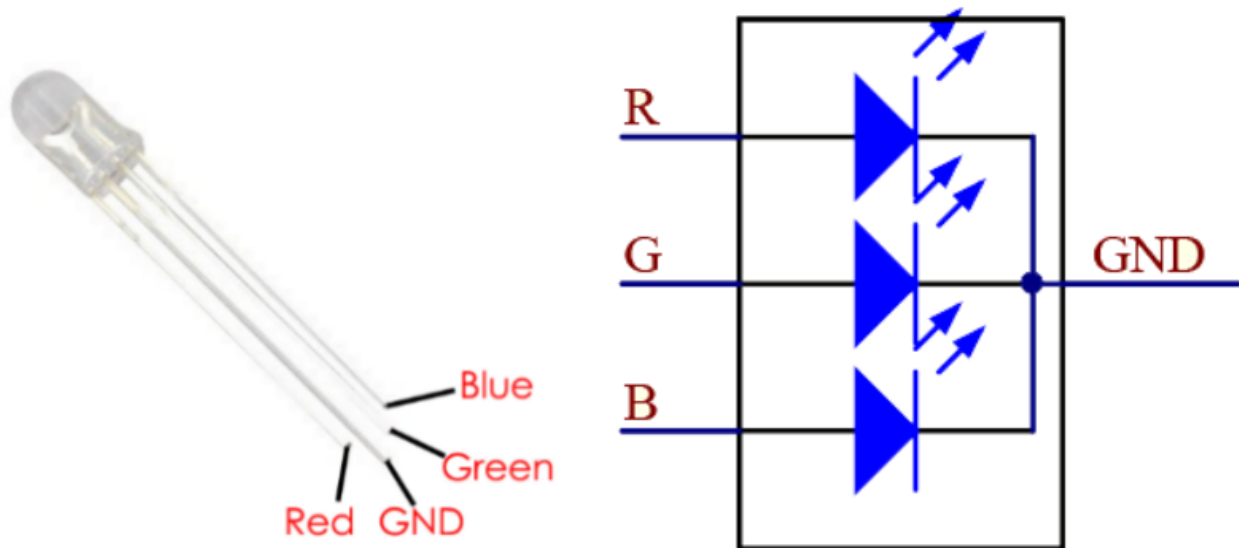
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>RGB LED</i>	

Schaltplan

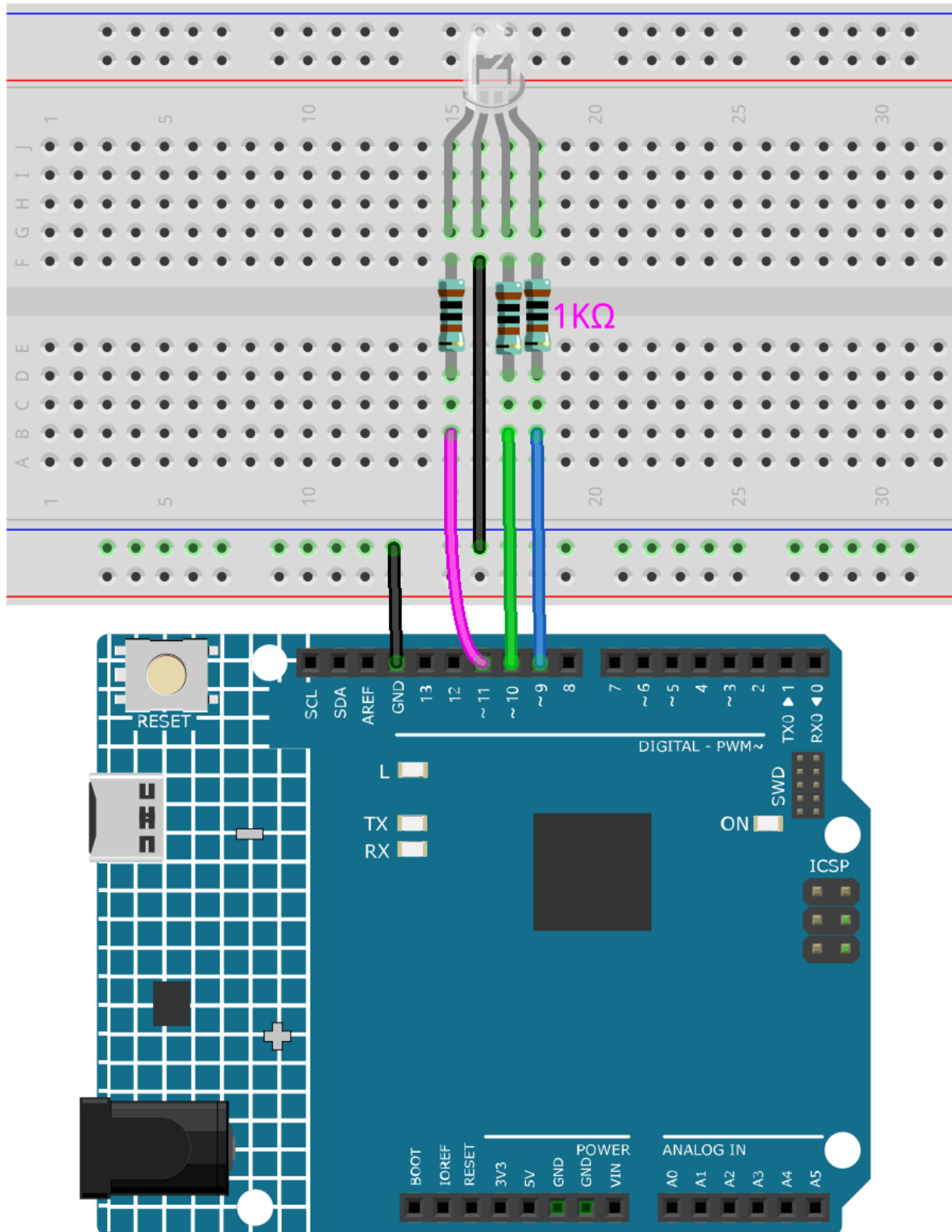


Die PWM-Pins 11, 10 und 9 steuern die roten, grünen und blauen Pins der RGB-LED jeweils an und verbinden den gemeinsamen Kathodenpin mit GND. Dadurch kann die RGB-LED durch Überlagern von Licht auf diesen Pins mit verschiedenen PWM-Werten eine bestimmte Farbe anzeigen.

Verdrahtung



Eine RGB-LED hat 4 Pins: Der längste Pin ist der gemeinsame Kathodenpin, der normalerweise mit GND verbunden ist, der linke Pin neben dem längsten Pin ist Rot und die 2 Pins rechts sind Grün und Blau.

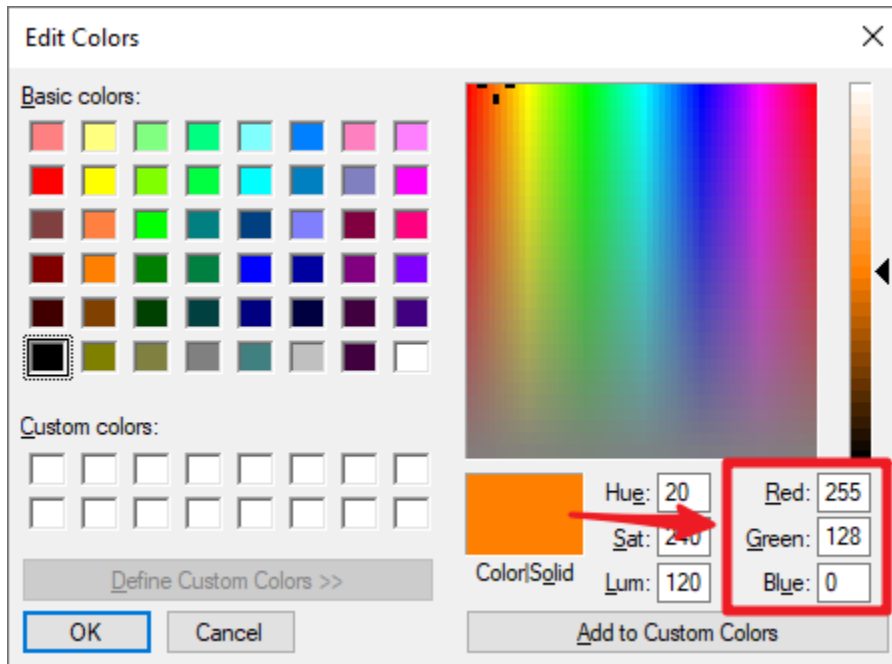


Code

Hier können wir unsere Lieblingsfarbe in einer Zeichensoftware (wie Paint) auswählen und sie mit der RGB-LED anzeigen.

Bemerkung:

- Sie können die Datei `2.2.colorful_light.ino` im Pfad `3in1-kit\learning_project\2.analogWrite\2.2.colorful_light` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.



Geben Sie den RGB-Wert in `color_set()` ein, und Sie werden sehen, dass die RGB-LED die von Ihnen gewünschten Farben anzeigt.

Wie funktioniert das?

In diesem Beispiel wird die Funktion zum Zuweisen von Werten an die drei Pins von RGB in einer unabhängigen Unterfunktion `color()` verpackt.

```
void color (unsigned char red, unsigned char green, unsigned char blue)
{
    analogWrite(redPin, red);
    analogWrite(greenPin, green);
    analogWrite(bluePin, blue);
}
```

In `loop()` arbeitet der RGB-Wert als Eingabeargument, um die Funktion `color()` aufzurufen, sodass die RGB unterschiedliche Farben emittieren kann.

```
void loop()
{
    color(255, 0, 0); // red
    delay(1000);
    color(0, 255, 0); // green
    delay(1000);
    color(0, 0, 255); // blue
}
```

(Fortsetzung auf der nächsten Seite)

```
    delay(1000);  
}
```

4.3 3. Digital Lesen

Sensoren erfassen Informationen aus der realen Welt. Diese Informationen werden dann über Pins (einige digital, andere analog) an das Hauptboard übermittelt, damit der Computer die tatsächliche Situation erkennen kann.

Daher kann das Arduino-Board den Arbeitsstatus von digitalen Sensoren durch Lesen des Wertes von Digitalpins, wie Tasten oder IR-Vermeidungsmodule, erkennen.

Hier sind die benötigten Funktionen.

- `pinMode(pin, mode)`: Konfiguriert den spezifischen Pin als INPUT oder OUTPUT. Hier muss er als INPUT eingestellt werden.

Syntax

```
pinMode(pin, mode)
```

Parameter

- `pin`: Die Arduino-Pinnummer, für die der Modus eingestellt werden soll.
- `mode`: INPUT, OUTPUT oder INPUT_PULLUP.

- `digitalRead(pin)`: Liest den Wert (Pegelzustand) vom angegebenen Digitalpin.

Syntax

```
digitalRead(pin)
```

Parameter

- `pin`: Die Arduino-Pinnummer, die Sie lesen möchten.

Rückgabewert

HIGH oder LOW

Beispiel für Digital Lesen

```
int ledPin = 13; // LED connected to digital pin 13  
int inPin = 7;   // pushbutton connected to digital pin 7  
int val = 0;     // variable to store the read value  
  
void setup() {  
    pinMode(ledPin, OUTPUT); // sets the digital pin 13 as output  
    pinMode(inPin, INPUT);   // sets the digital pin 7 as input  
}  
  
void loop() {  
    val = digitalRead(inPin); // read the input pin  
    digitalWrite(ledPin, val); // sets the LED to the button's value  
}
```

Anmerkungen und Warnungen

1. Pull Up & Pull Down.

`digitalRead()` kann zufällige, unbestimmte Werte erzeugen, wenn der Pin kein Pegelsignal erhält. Das Lenken der Eingangspins in einen bekannten Zustand kann das Projekt zuverlässiger machen. Bei Verwendung einer Eingangskomponente, wie einem Taster, ist es in der Regel notwendig, einen Pull-up- oder Pull-down-Widerstand parallel zum Digital-Eingangspin anzuschließen.

Neben dem Anschließen eines Pull-up-Widerstandes können Sie auch den Pin-Modus im Code auf `INPUT_PULLUP` setzen, zum Beispiel `pinMode(pin, INPUT_PULLUP)`. In diesem Fall greift der Pin über die Software auf den eingebauten Pull-up-Widerstand von Atmega zu und hat den gleichen Effekt wie das Anschließen eines Pull-up-Widerstandes.

2. Über Pin13.

Alle digitalen Pins (1-13) auf dem R4-Board können als `digitalRead()` verwendet werden. Aber der digitale Pin 13 ist schwieriger als andere digitale Pins als digitaler Eingang zu verwenden. Da er eine LED und einen Widerstand verbindet, ist er auf den meisten Boards verlötet. Wenn Sie seinen internen 20k Pull-up-Widerstand aktivieren, bleibt er bei etwa 1,7 V anstelle der erwarteten 5 V, da die onboard LED und der Serienwiderstand das Spannungsniveau herunterziehen, was bedeutet, dass er immer LOW zurückgibt. Wenn Sie Pin 13 unbedingt als digitalen Eingang verwenden müssen, setzen Sie seine `pinMode()` auf `INPUT` und verwenden Sie einen externen Pull-down-Widerstand.

3. Analogpins.

Wenn die digitalen Pins nicht ausreichen, können die Analogpins (A0-A5) auch als digitale Pins verwendet werden. Sie müssen mit `pinMode(pin,mode)` auf `INPUT` eingestellt werden.

Verwandte Komponenten

Im Folgenden finden Sie die zugehörigen Komponenten. Sie können darauf klicken, um zu erfahren, wie sie verwendet werden.

4.3.1 3.0 Serieller Monitor

In der Arduino IDE gibt es einen seriellen Monitor, mit dem Sie Nachrichten von Ihrem Computer an das Arduino-Board (über USB) senden und auch Nachrichten vom Arduino empfangen können.

In diesem Projekt werden wir also lernen, wie man Daten vom Arduino-Board empfängt.

Bemerkung: Bei Uno, Nano, Mini und Mega werden die Pins 0 und 1 für die Kommunikation mit dem Computer verwendet. Wenn Sie etwas an diese Pins anschließen, kann dies die Kommunikation stören, einschließlich fehlgeschlagener Uploads zum Board.

Verwendung des seriellen Monitors

1. Öffnen Sie die Arduino IDE und fügen Sie den folgenden Code ein.

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

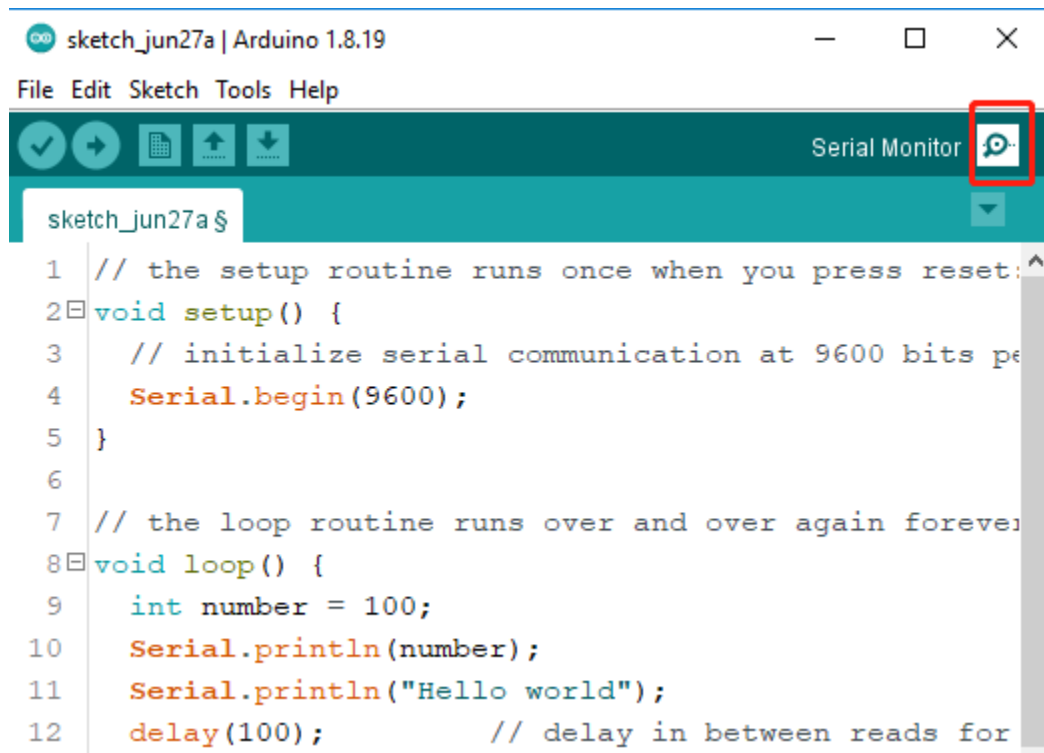
// the loop routine runs over and over again forever:
void loop() {
  int number = 100;
  Serial.println(number);
  Serial.println("Hello world");
}
```

(Fortsetzung auf der nächsten Seite)

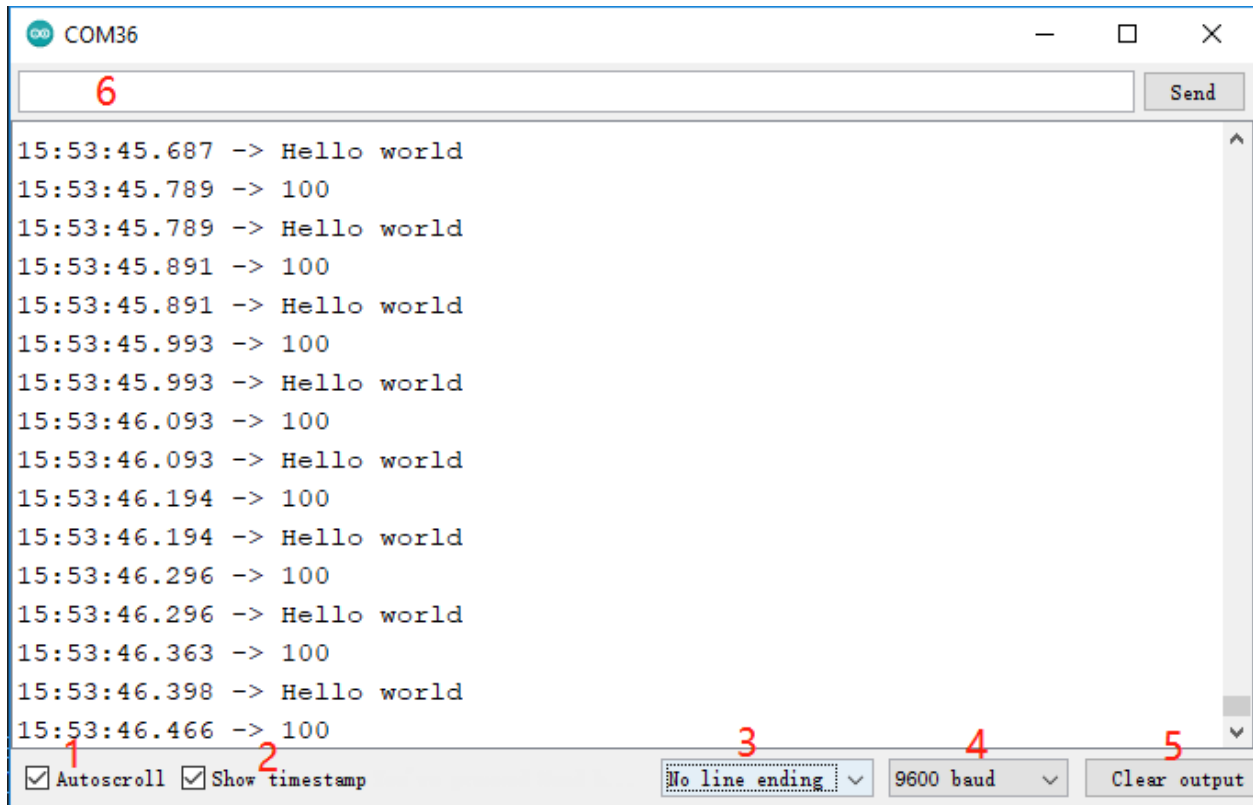
(Fortsetzung der vorherigen Seite)

```
    delay(100);          // delay in between reads for stability
}
```

- `Serial.begin()`: Legt die Datenrate in Bits pro Sekunde (Baud) für die serielle Datenübertragung fest, hier auf 9600 eingestellt.
 - `Serial.println()`: Gibt Daten als menschenlesbaren ASCII-Text an den seriellen Port gefolgt von einem Wagenrücklauf-Zeichen (ASCII 13 oder ,r') und einem Zeilenumbruch-Zeichen (ASCII 10 oder ,n') aus. Dieser Befehl hat dieselben Formen wie `Serial.print()`.
2. Wählen Sie das korrekte Board und den Port, um den Code hochzuladen.
 3. Klicken Sie in der Symbolleiste auf das Vergrößerungsglas-Symbol, um den seriellen Monitor zu starten.



4. Hier sehen Sie den seriellen Monitor.



- 1: Option zum automatischen Scrollen oder Nicht-Scrollen.
- 2: Option zur Anzeige eines Zeitstempels vor den auf dem seriellen Monitor angezeigten Daten.
- 3: Endauswahl: Auswahl der Zeichen, die an die Daten angehängt werden, die an Arduino gesendet werden. Auswahlmöglichkeiten:
 - **Kein Zeilenumbruch**: Sendet einfach, was Sie eingeben;
 - **Neue Zeile** entspricht `\n` und sendet nach Ihrer Eingabe einen ASCII-Zeilenumbruch;
 - **Wagenrücklauf** entspricht `\r`, sendet nach Ihrer Eingabe ein ASCII-Wagenrücklauf-Zeichen;
 - **Beide NL & CR** entsprechen `\r\n` und senden sowohl ein Wagenrücklauf- als auch ein Zeilenumbruch-Zeichen nach Ihrer Eingabe.
- 4: Wählen Sie die Kommunikationsgeschwindigkeit zwischen Arduino-Board und PC. Dieser Wert MUSS dem in `Serial.begin()` eingestellten Wert entsprechen.
- 5: Löscht den gesamten Text auf der Ausgabekonsolle.
- 6: Ein Textfeld zum Senden von Zeichen an das Arduino-Board. Siehe [5.12 Serielles Lesen](#) für ein Tutorial.

4.3.2 3.1 Tastenwert auslesen

In den vorherigen Projekten haben wir die Ausgabefunktion verwendet, in diesem Kapitel werden wir die Eingabefunktion verwenden, um den Tastenwert auszulesen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

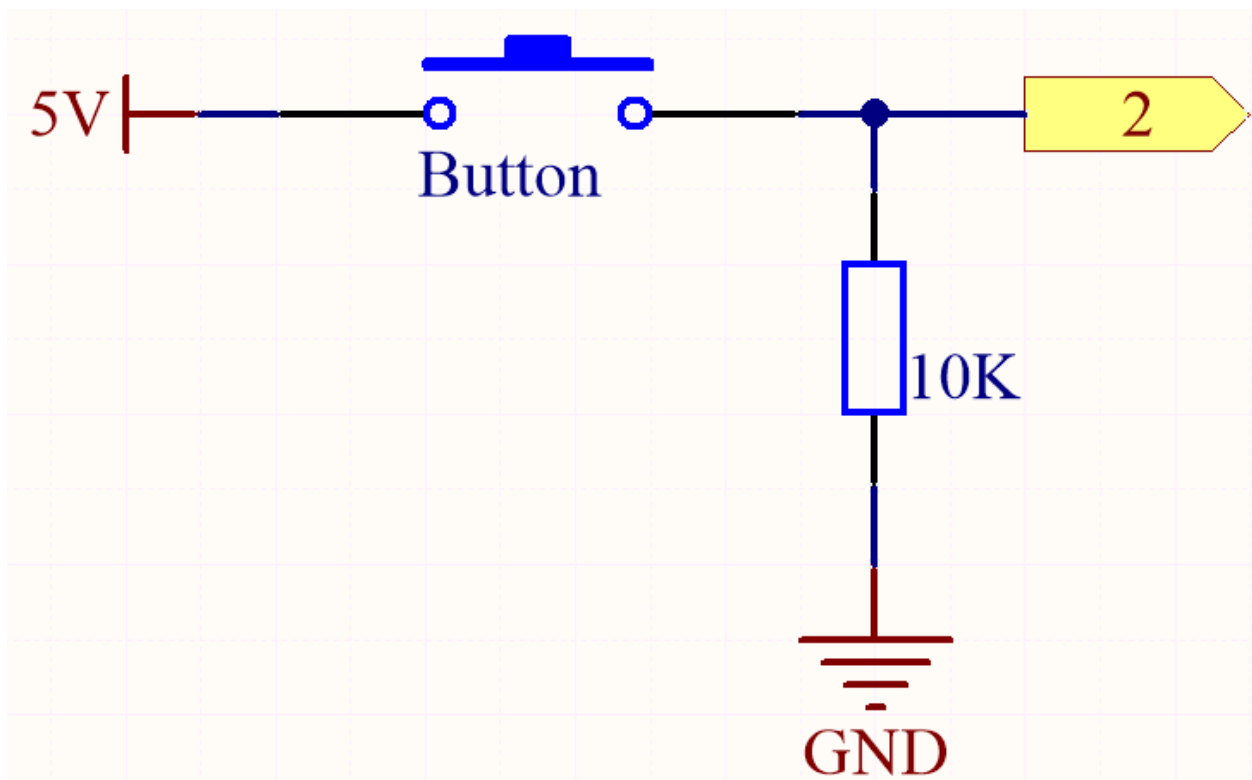
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

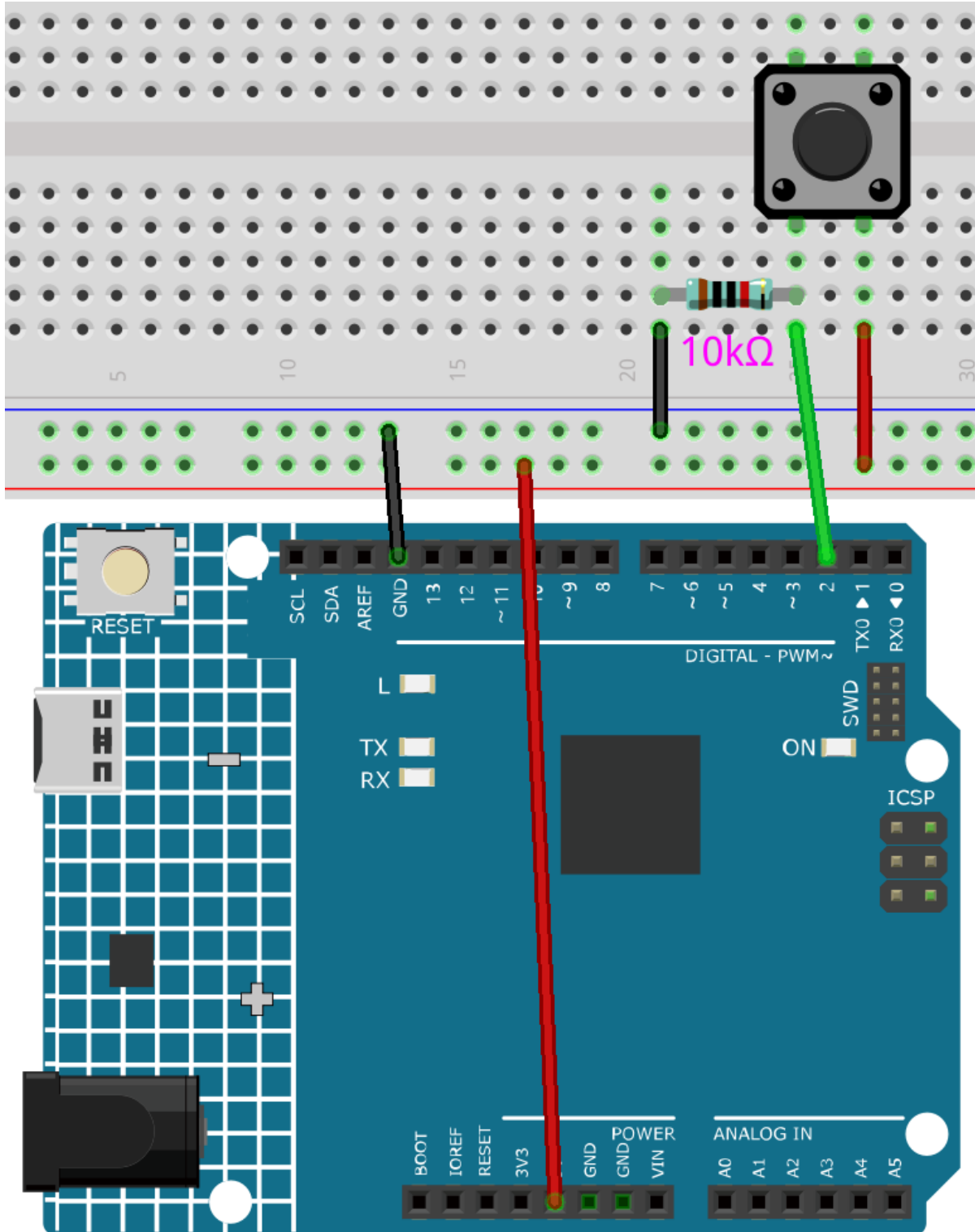
KOMPONENTENEINFÜHRUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Knopf</i>	

Schaltplan



Ein Pin der Taste ist mit 5V verbunden, der andere Pin ist mit Pin 2 verbunden. Wenn die Taste gedrückt wird, wird Pin 2 auf High gesetzt. Wenn die Taste jedoch nicht gedrückt wird, befindet sich Pin 2 in einem schwebenden Zustand und könnte High oder Low sein. Um ein stabiles Low-Level zu erhalten, wenn die Taste nicht gedrückt wird, muss Pin 2 über einen 10K Pull-Down-Widerstand erneut mit GND verbunden werden.

Verdrahtung

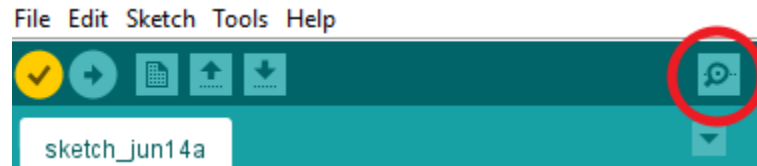


Code

Bemerkung:

- Sie können die Datei `3.1.read_button_value.ino` im Pfad `3in1-kit\learning_project\3.1.read_button_value` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem der Code erfolgreich hochgeladen wurde, klicken Sie auf das Vergrößerungsglas-Symbol in der oberen rechten Ecke der Arduino IDE (Serieller Monitor).



Wenn Sie den Knopf drücken, wird im Seriellen Monitor „1“ ausgedruckt.

4.3.3 3.2 Magnetismus spüren

Die gängigste Art von Reed-Schalter enthält ein Paar magnetisierbare, flexible Metallzungen, deren Endteile durch eine kleine Lücke getrennt sind, wenn der Schalter offen ist.

Ein Magnetfeld von einem Elektromagneten oder einem Permanentmagneten bewirkt, dass sich die Zungen anziehen und so einen elektrischen Stromkreis schließen. Die Federkraft der Zungen bewirkt, dass sie sich trennen und den Kreislauf öffnen, wenn das Magnetfeld nachlässt.

Ein häufiges Anwendungsbeispiel für einen Reed-Schalter ist die Erkennung des Öffnens einer Tür oder eines Fensters für einen Sicherheitsalarm.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

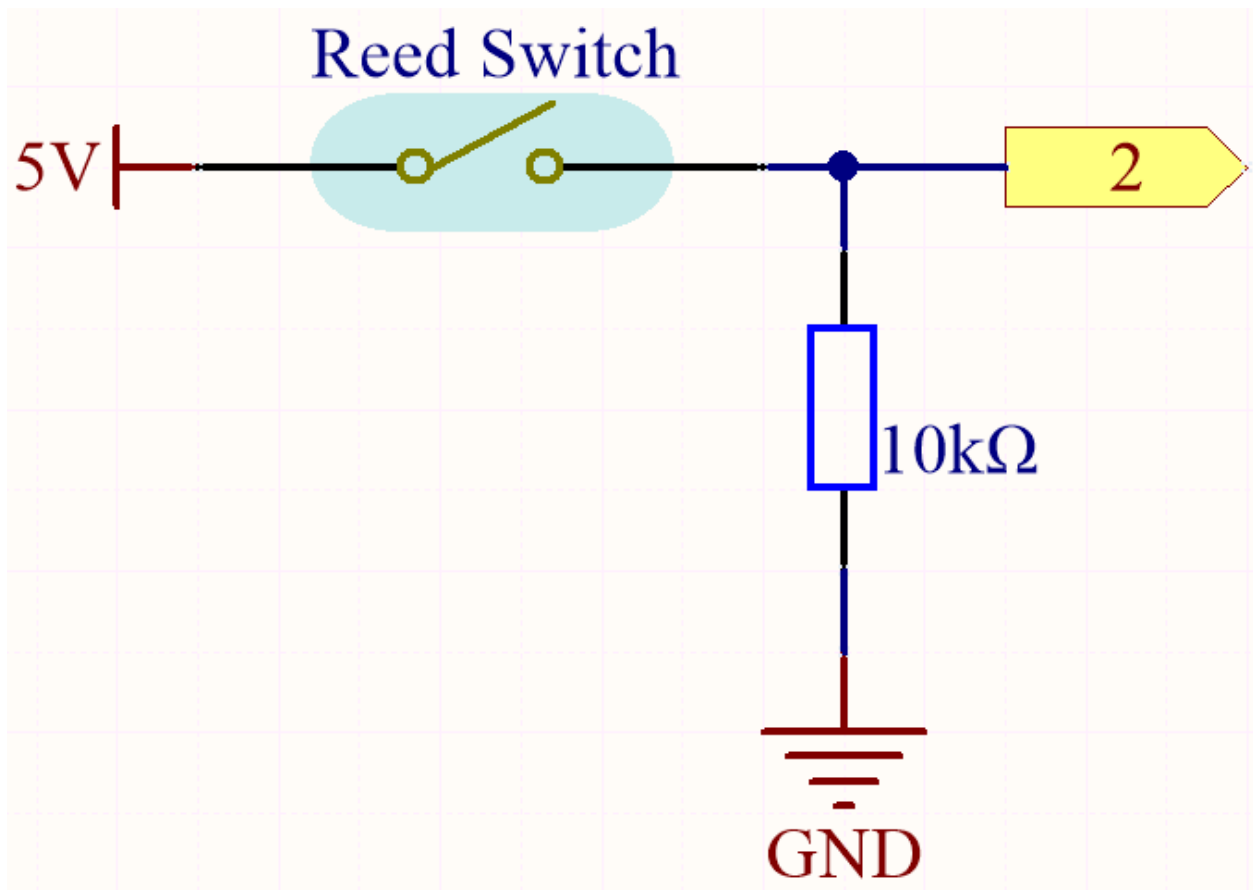
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENEINFÜHRUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Reedschalter</i>	-

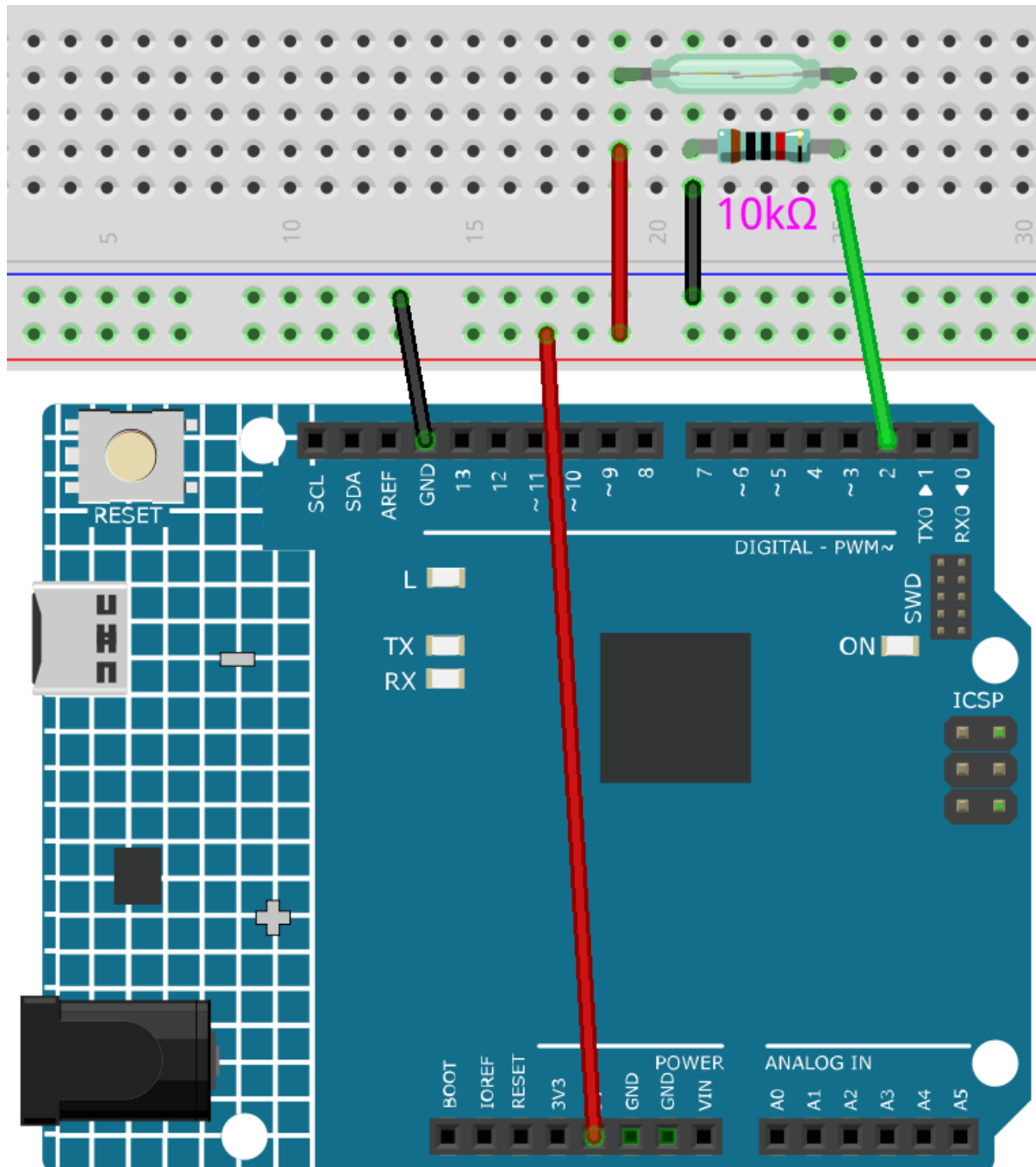
Schaltplan



Standardmäßig ist Pin 2 niedrig; und wird hoch gehen, wenn der Magnet in der Nähe des Reed-Schalters ist.

Der Zweck des 10K-Widerstands ist es, den Pin 2 auf einem konstant niedrigen Niveau zu halten, wenn kein Magnet in der Nähe ist.

Verdrahtung



Code

Bemerkung:

- Sie können die Datei 3.2.feel_the_magnetism.ino im Pfad 3in1-kit\learning_project\3.2.feel_the_magnetism öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem der Code erfolgreich hochgeladen wurde, wird im seriellen Monitor eine 1 ausgedruckt, wenn ein Magnet in

der Nähe des Reed-Schalters ist.

4.3.4 3.3 Hindernis erkennen

Dieses Modul wird häufig in Autos und Robotern installiert, um das Vorhandensein von Hindernissen vor ihnen zu erkennen. Zudem wird es weit verbreitet in Handgeräten, Wasserhähnen und dergleichen verwendet.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

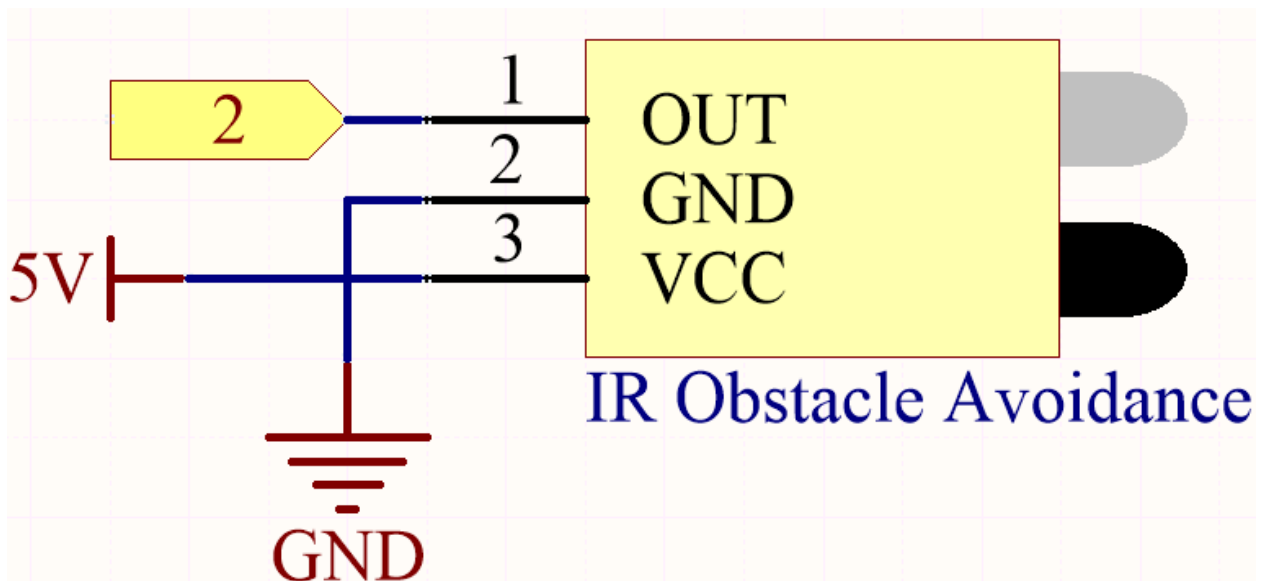
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

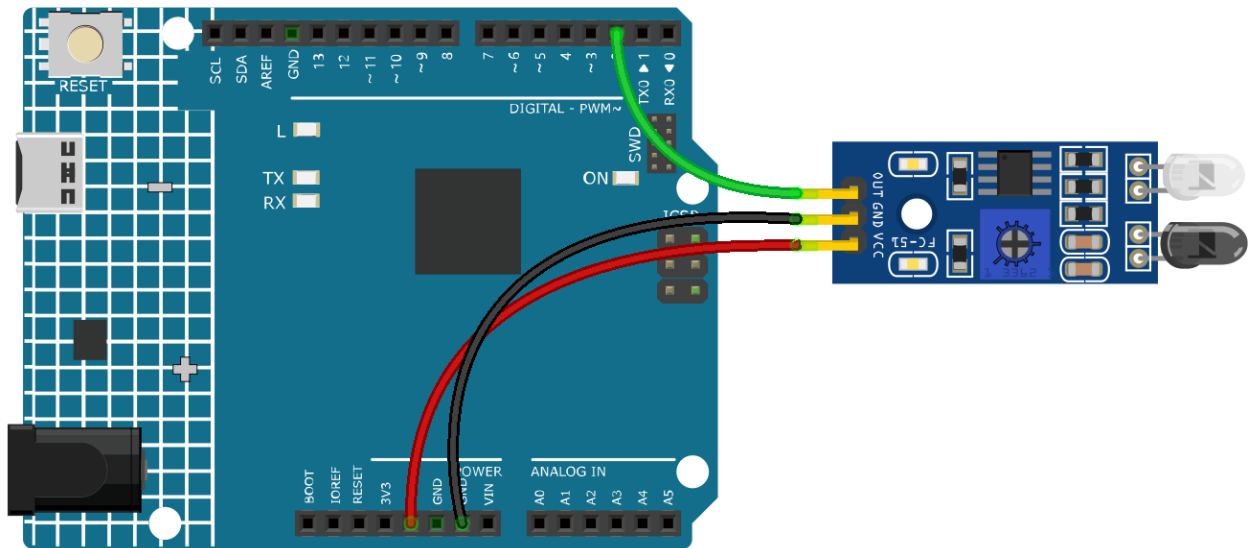
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Jumper-Kabel</i>	
<i>Hindernisvermeidungsmodul</i>	

Schaltplan



Der digitale Pin 2 wird verwendet, um das Signal des IR-Hindernisvermeidungsmoduls zu lesen. Der VCC des IR-Sensor-Moduls wird an 5V angeschlossen, GND an GND und OUT an den digitalen Pin 2.

Verdrahtung



Code

Bemerkung:

- Sie können die Datei `3.3.detect_the_obstacle.ino` im Pfad `3in1-kit\learning_project\3.3.detect_the_obstacle` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Wenn das IR-Hindernisvermeidungsmodul etwas registriert, das sich direkt davor befindet, wird [0] im seriellen Monitor angezeigt, andernfalls wird [1] dargestellt.

4.3.5 3.4 Die Linie erkennen

Das Linienvollzugsmodul wird verwendet, um festzustellen, ob es schwarze Bereiche auf dem Boden gibt, wie beispielsweise mit Isolierband aufgeklebte schwarze Linien.

Eine seiner LEDs sendet geeignetes Infrarotlicht auf den Boden. Die schwarze Oberfläche hat eine relativ starke Fähigkeit, Licht zu absorbieren und eine schwächere Reflexionsfähigkeit. Weiße Oberflächen sind das Gegenteil. Wenn reflektiertes Licht erkannt wird, bedeutet dies, dass der Boden aktuell weiß ist. Wenn nicht erkannt, bedeutet es schwarz.

So funktioniert es.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

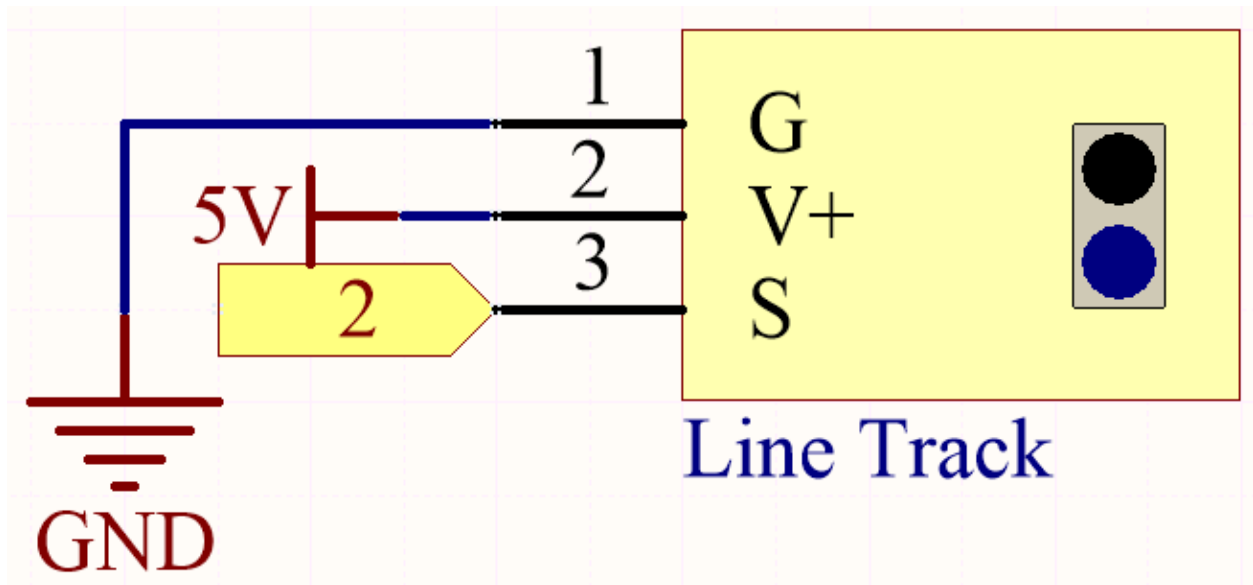
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

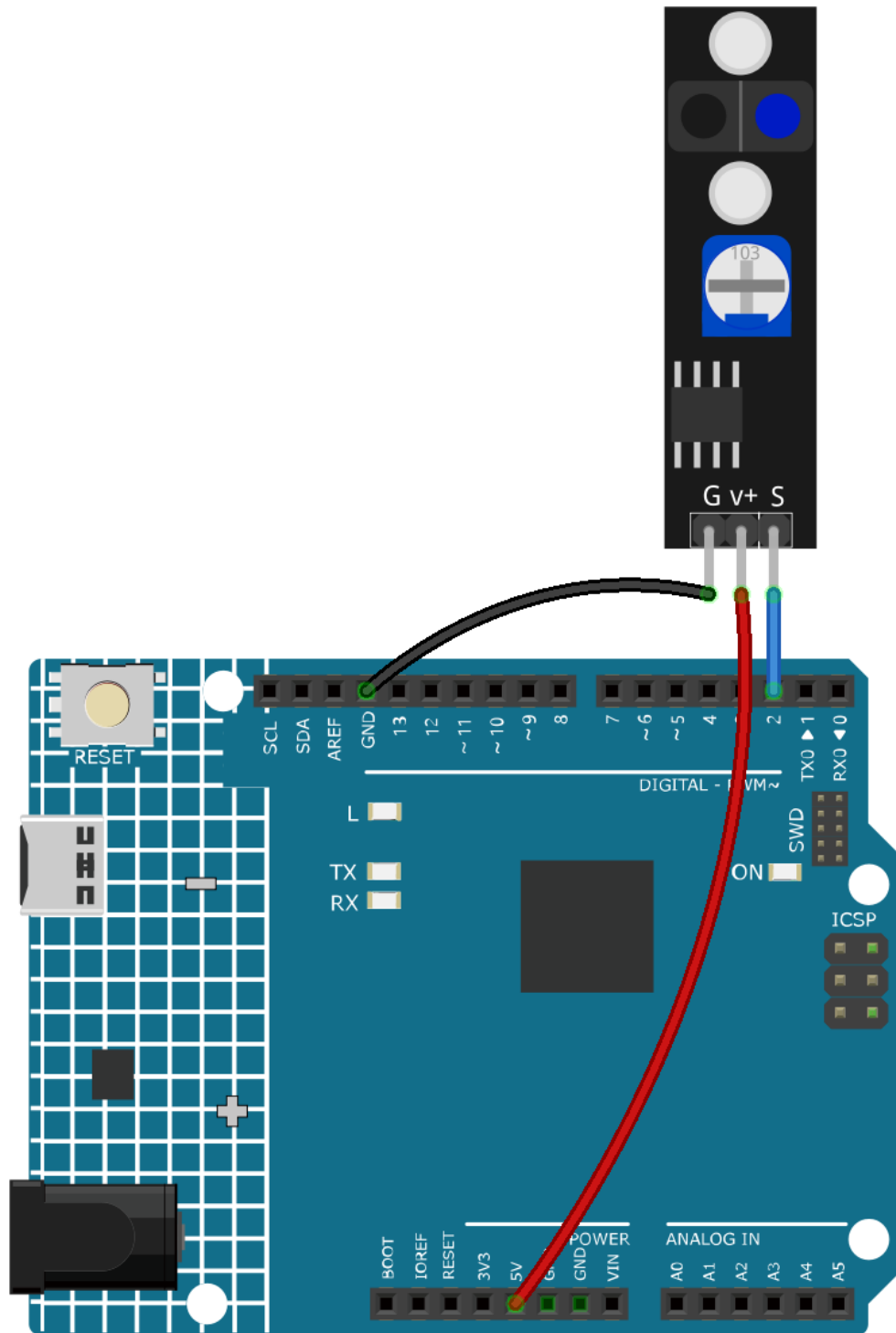
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Jumper-Kabel</i>	
<i>Linienverfolgungsmodul</i>	

Schaltplan



Der digitale Pin 2 wird verwendet, um das Signal des Linienverfolgungsmoduls zu lesen. Der VCC des Moduls wird an 5V angeschlossen, GND an GND und OUT an den digitalen Pin 2.

Verdrahtung



Code

Bemerkung:

- Sie können die Datei `3.4.detect_the_line.ino` im Pfad `3in1-kit\learning_project\3.4.detect_the_line` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Wenn das Linienverfolgungsmodul erkennt, dass eine schwarze Linie vorliegt, wird [1] im Serial Monitor angezeigt; andernfalls wird [0] dargestellt.

4.4 4. Analoges Lesen

Mit dem Arduino können die an den analogen Pins angeschlossenen Sensoren gelesen werden.

Das R4-Board enthält einen Mehrkanal-10-Bit-Analog-Digital-Wandler. Das bedeutet, dass es die Eingangsspannung zwischen 0 und der Betriebsspannung (5V oder 3,3V) einem ganzzahligen Wert zwischen 0 und 1023 zuordnet.

Um den Wert des analogen Pins zu lesen, benötigen Sie die Funktion `analogRead(pin)`.

- `analogRead(pin)`: Liest den Wert vom angegebenen analogen Pin.

Syntax

`analogRead(pin)`

Parameter

- `pin`: Der Name des analogen Eingangspins, von dem gelesen werden soll (A0 bis A5).

Rückgabewert

0-1023. Datentyp: `int`.

Beispiel für analoges Lesen

```
int analogPin = A0; // device connected to analog pin A0
                    // outside leads to ground and +5V
int val = 0; // variable to store the value read

void setup() {
  Serial.begin(9600); // setup serial
}

void loop() {
  val = analogRead(analogPin); // read the input pin
  Serial.println(val); // debug value
}
```

Hinweise und Warnungen

- Die analogen Pins sind A0-A5.
- Es ist nicht notwendig, `pinMode()` vor dem Aufruf des analogen Pins zu verwenden, aber wenn der Pin zuvor auf OUTPUT gesetzt wurde, funktioniert die Funktion `analogRead()` nicht ordnungsgemäß. In diesem Fall müssen Sie `pinMode()` aufrufen, um ihn wieder auf INPUT zu setzen.

Zugehörige Komponenten

Unten sind die zugehörigen Komponenten aufgelistet. Sie können darauf klicken, um zu erfahren, wie sie verwendet werden.

4.4.1 4.1 Den Knopf drehen

Ein Potentiometer ist ein Widerstandskomponente mit 3 Anschlüssen und sein Widerstandswert kann gemäß einer regelmäßigen Variation angepasst werden.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten:

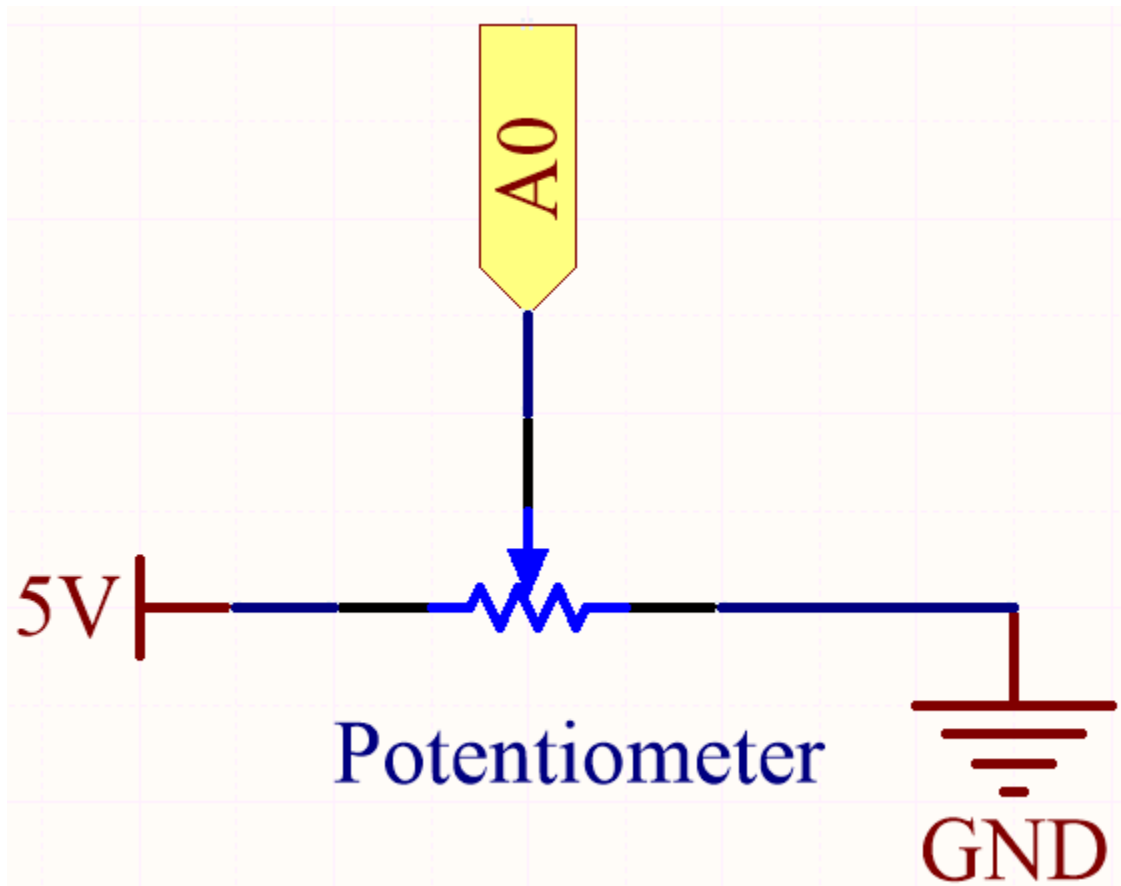
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

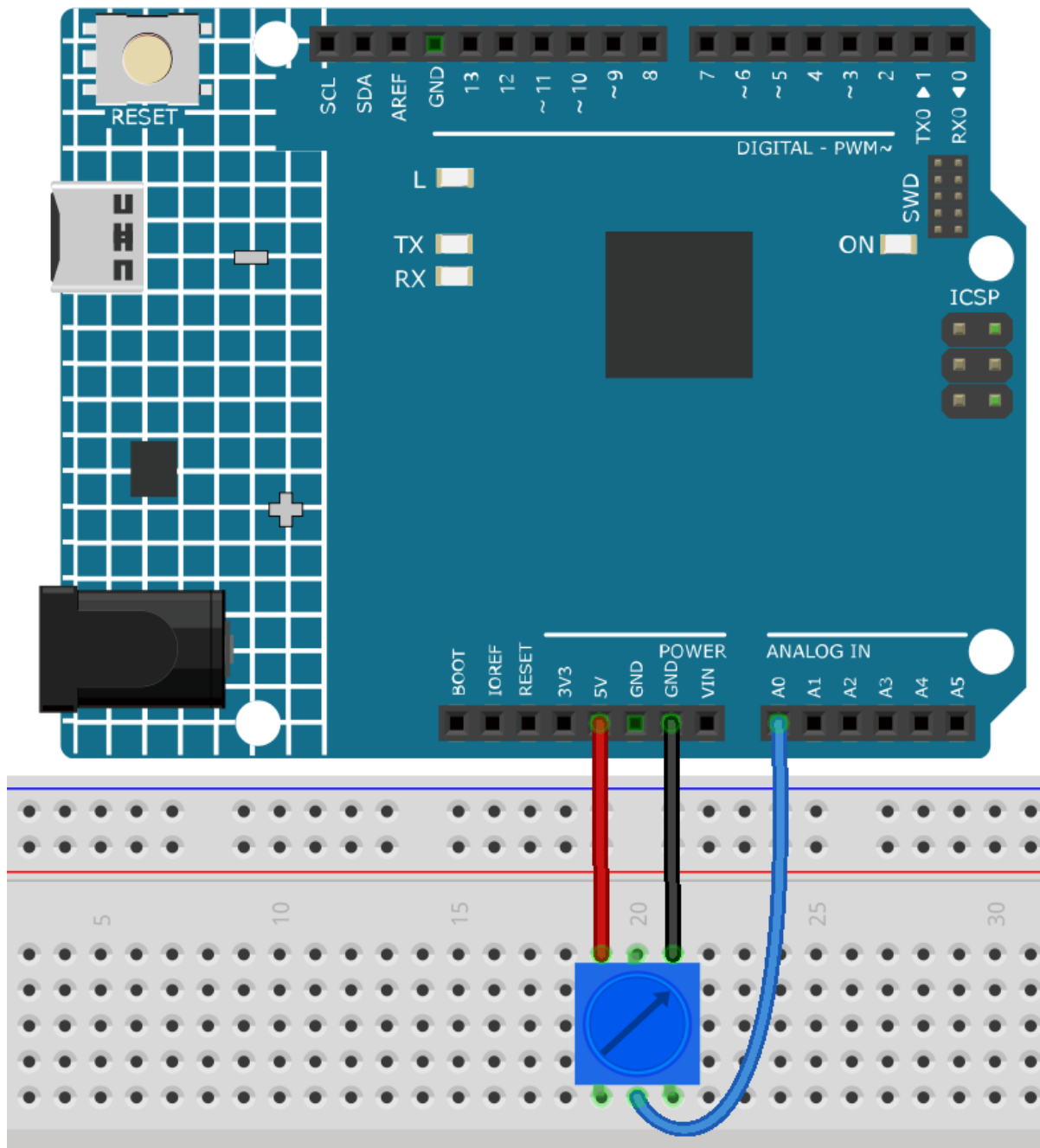
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Potentiometer</i>	

Schaltplan



In diesem Beispiel verwenden wir den analogen Pin (A0), um den Wert des Potentiometers zu lesen. Durch das Drehen der Achse des Potentiometers können Sie die Widerstandsverteilung zwischen diesen drei Pins ändern und so die Spannung am mittleren Pin ändern. Wenn der Widerstand zwischen dem mittleren und einem äußeren, mit 5V verbundenen Pin nahe Null liegt (und der Widerstand zwischen dem mittleren und dem anderen äußeren Pin nahe 10k liegt), liegt die Spannung am mittleren Pin nahe 5V. Die umgekehrte Operation (der Widerstand zwischen dem mittleren und einem äußeren, mit 5V verbundenen Pin liegt nahe 10k) wird die Spannung am mittleren Pin nahe 0V bringen.

Verkabelung



Code

Bemerkung:

- Sie können die Datei `4.1.turn_the_knob.ino` im Pfad `3in1-kit\learning_project\4.1.turn_the_knob` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nach dem Hochladen des Codes auf das Board können Sie den seriellen Monitor öffnen, um den gelesenen Wert des Pins zu sehen. Beim Drehen der Achse des Potentiometers wird der serielle Monitor den Wert 0~1023 ausgeben.

4.4.2 4.2 Das Licht fühlen

Der Fotowiderstand ist ein typisches Gerät für analoge Eingänge und wird sehr ähnlich wie ein Potentiometer verwendet. Sein Widerstandswert hängt von der Lichtintensität ab: Je stärker das eingestrahlte Licht, desto geringer ist sein Widerstandswert und umgekehrt steigt er.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten:

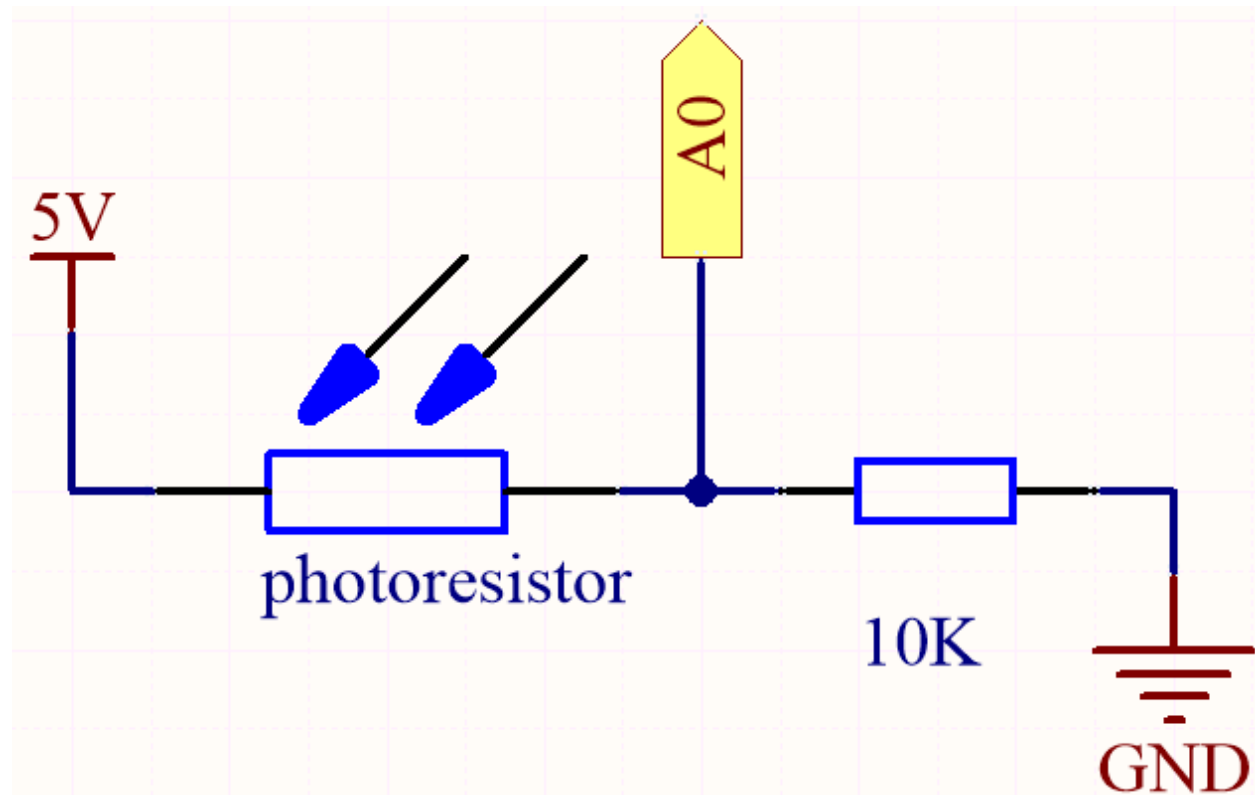
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Fotowiderstand</i>	

Schaltplan

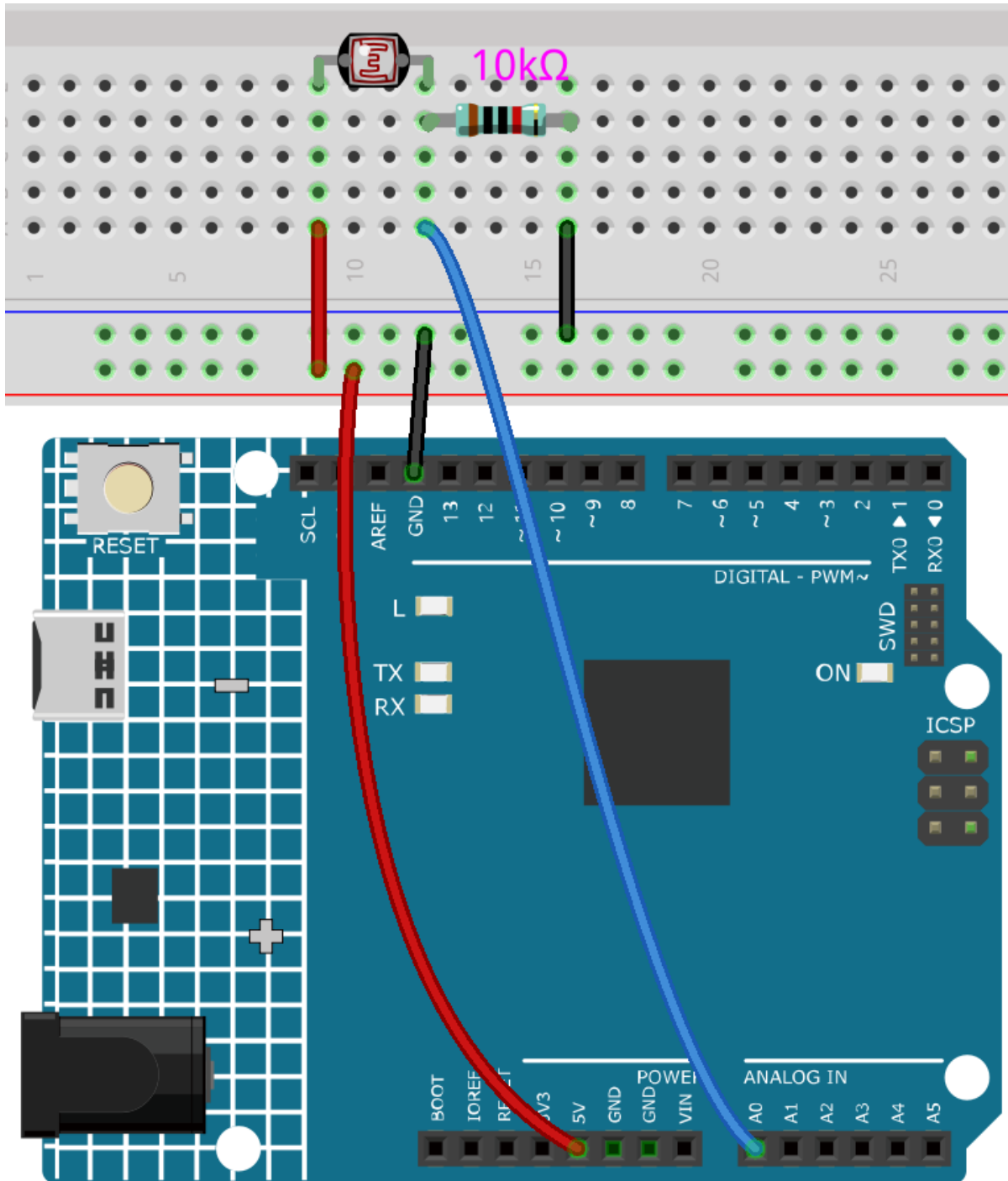


In dieser Schaltung sind der 10K-Widerstand und der Fotowiderstand in Serie geschaltet, und der durch sie fließende Strom ist derselbe. Der 10K-Widerstand dient als Schutz, und der Pin A0 liest den Wert nach der Spannungsumwandlung des Fotowiderstands.

Wenn das Licht verstärkt wird, verringert sich der Widerstand des Fotowiderstands, dann sinkt seine Spannung, sodass der Wert von Pin A0 steigen wird; wenn das Licht stark genug ist, wird der Widerstand des Fotowiderstands nahezu 0 sein, und der Wert von Pin A0 wird nahe 1023 liegen. Zu diesem Zeitpunkt spielt der 10K-Widerstand eine schützende Rolle, damit 5V und GND nicht miteinander verbunden sind, was zu einem Kurzschluss führt.

Wenn Sie den Fotowiderstand in einer dunklen Situation platzieren, wird der Wert von Pin A0 sinken. Ist es dunkel genug, wird der Widerstand des Fotowiderstands unendlich sein, und seine Spannung wird nahe 5V liegen (der 10K-Widerstand ist vernachlässigbar), und der Wert von Pin A0 wird nahe 0 liegen.

Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 4.2.feel_the_light.ino im Pfad 3in1-kit\learning_project\4.2.feel_the_light.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nach erfolgreichem Hochladen des Codes gibt der serielle Monitor die Werte des Fotowiderstands aus. Je stärker die aktuelle Umgebungshelligkeit, desto größer ist der Wert, der auf dem seriellen Monitor angezeigt wird.

4.4.3 4.3 Den Joystick betätigen

Der Joystick sollte jedem, der regelmäßig Videospiele spielt, sehr vertraut sein. Er wird normalerweise verwendet, um Charaktere zu bewegen oder den Bildschirm zu drehen.

Unsere Bewegungen können vom Joystick gelesen werden, der nach einem sehr einfachen Prinzip funktioniert. Er besteht aus zwei Potentiometern, die senkrecht zueinander stehen. Diese beiden Potentiometer messen den analogen Wert des Joysticks in vertikaler und horizontaler Richtung und erzeugen einen Wert (x,y) in einem planaren rechtwinkligen Koordinatensystem.

Dieses Kit enthält auch einen Joystick mit einem digitalen Eingang. Er wird aktiviert, wenn der Joystick gedrückt wird.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten:

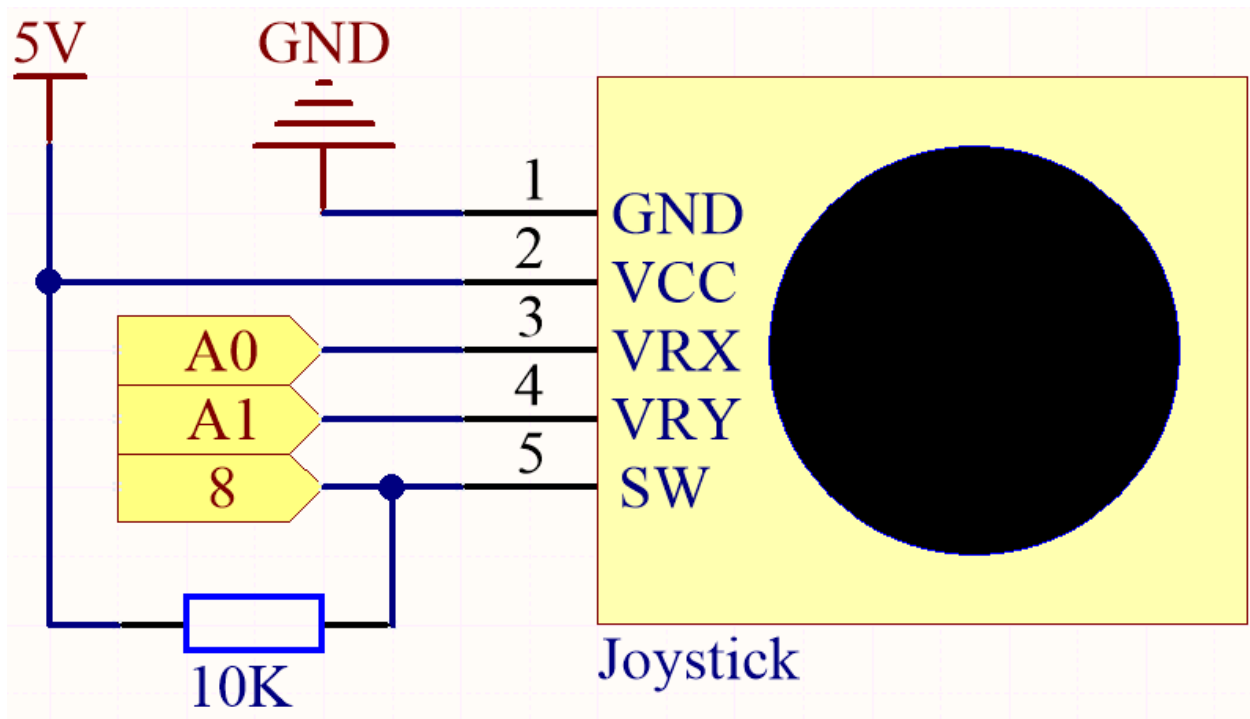
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

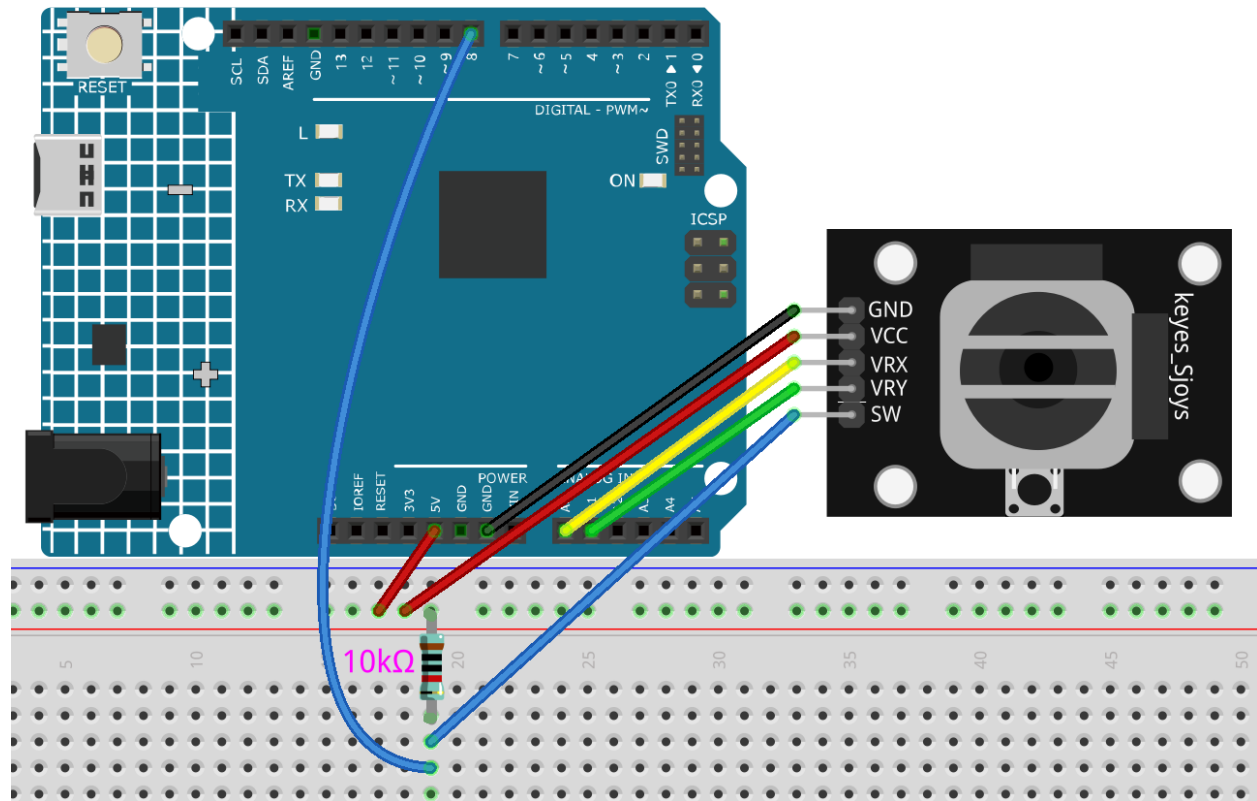
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Joystick Modul</i>	-

Schaltplan



Bemerkung: Der SW-Pin ist mit einem 10K-Pull-up-Widerstand verbunden, der Grund dafür ist, um einen stabilen hohen Pegel am SW-Pin (Z-Achse) zu erhalten, wenn der Joystick nicht gedrückt ist; ansonsten befindet sich der SW in einem Schwebezustand und der Ausgangswert kann zwischen 0/1 variieren.

Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 4.3.toggle_the_joystick.ino im Pfad 3in1-kit\learning_project\4.3.toggle_the_joystick.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Öffnen Sie nach erfolgreichem Hochladen des Codes den seriellen Monitor, um die x,y,z-Werte des Joysticks zu sehen.

- Die Werte der x-Achse und y-Achse sind analoge Werte, die von 0 bis 1023 variieren.
- Die Z-Achse ist ein digitaler Wert mit einem Status von 1 oder 0 (wenn gedrückt, ist es 0).

4.4.4 4.4 Bodenfeuchtigkeit messen

In der Pflanzenbranche können die Kulturen selbst die anorganischen Elemente im Boden nicht direkt aufnehmen. Das Wasser im Boden fungiert als Lösungsmittel, um diese anorganischen Elemente aufzulösen.

Kulturen nehmen Bodenfeuchtigkeit über das Wurzelsystem auf, erhalten Nährstoffe und fördern das Wachstum.

Im Prozess des Pflanzenwachstums und der Entwicklung sind die Anforderungen an die Bodentemperatur ebenfalls unterschiedlich. Daher ist ein Bodenfeuchtigkeitssensor erforderlich.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten:

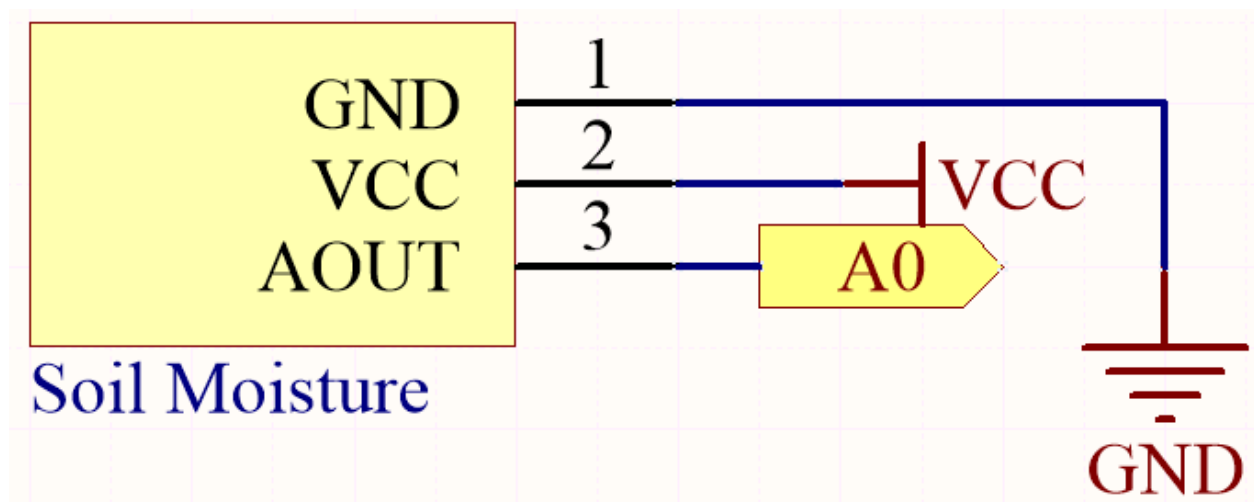
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

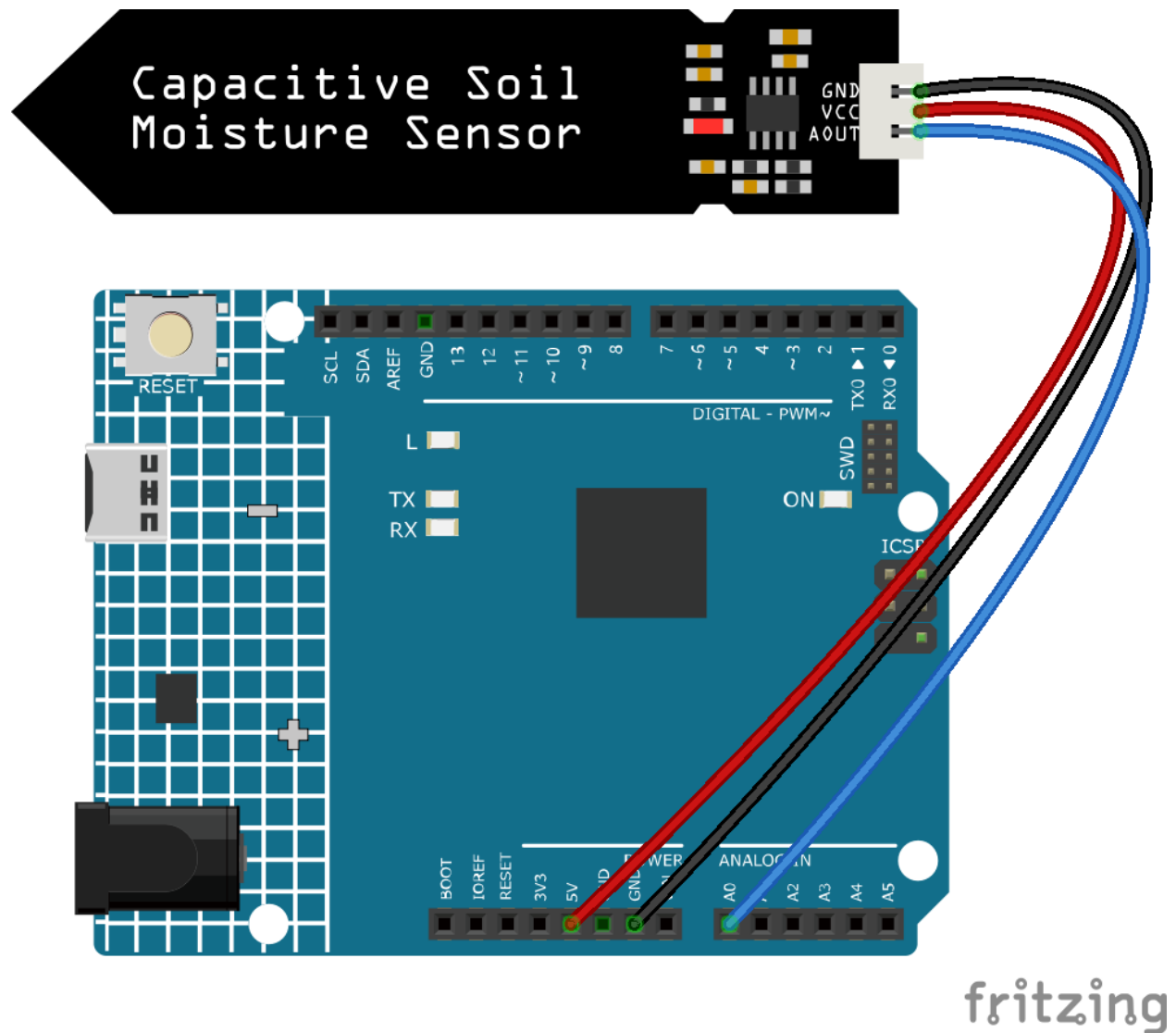
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Jumper-Kabel</i>	
<i>Bodenfeuchtigkeitsmodul</i>	

Schaltplan



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 4.4.measure_soil_moisture.ino im Pfad 3in1-kit\learning_project\4.4.measure_soil_moisture.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem der Code erfolgreich hochgeladen wurde, druckt der serielle Monitor den Bodenfeuchtigkeitswert aus.

Indem man das Modul in den Boden einsetzt und ihn bewässert, wird der Wert des Bodenfeuchtigkeitssensors kleiner werden.

4.4.5 4.5 Thermometer

Ein Thermometer ist ein Gerät, das die Temperatur oder ein Temperaturgefälle misst (den Grad der Wärme oder Kälte eines Objekts). Ein Thermometer verfügt über zwei wichtige Elemente: (1) einen Temperatursensor (z.B. die Birne eines Quecksilber-Thermometers oder den pyrometrischen Sensor eines Infrarot-Thermometers), bei dem sich bei Temperaturänderungen etwas verändert; und (2) eine Methode zur Umwandlung dieser Änderung in einen numerischen Wert (z.B. die sichtbare Skala, die auf einem Quecksilber-Thermometer markiert ist, oder die digitale Anzeige bei einem Infrarot-Modell). Thermometer werden in Technologie und Industrie zur Überwachung von Prozessen, in der Meteorologie, in der Medizin und in der wissenschaftlichen Forschung häufig verwendet.

Ein Thermistor ist eine Art Temperatursensor, dessen Widerstand stark temperaturabhängig ist, und es gibt es in zwei Typen: Negativer Temperaturkoeffizient (NTC) und Positiver Temperaturkoeffizient (PTC), auch bekannt als NTC und PTC. Der Widerstand des PTC-Thermistors steigt mit der Temperatur, während der Zustand des NTC dem des PTC entgegengesetzt ist.

In diesem Experiment verwenden wir einen **NTC thermistor**, um ein Thermometer herzustellen.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten:

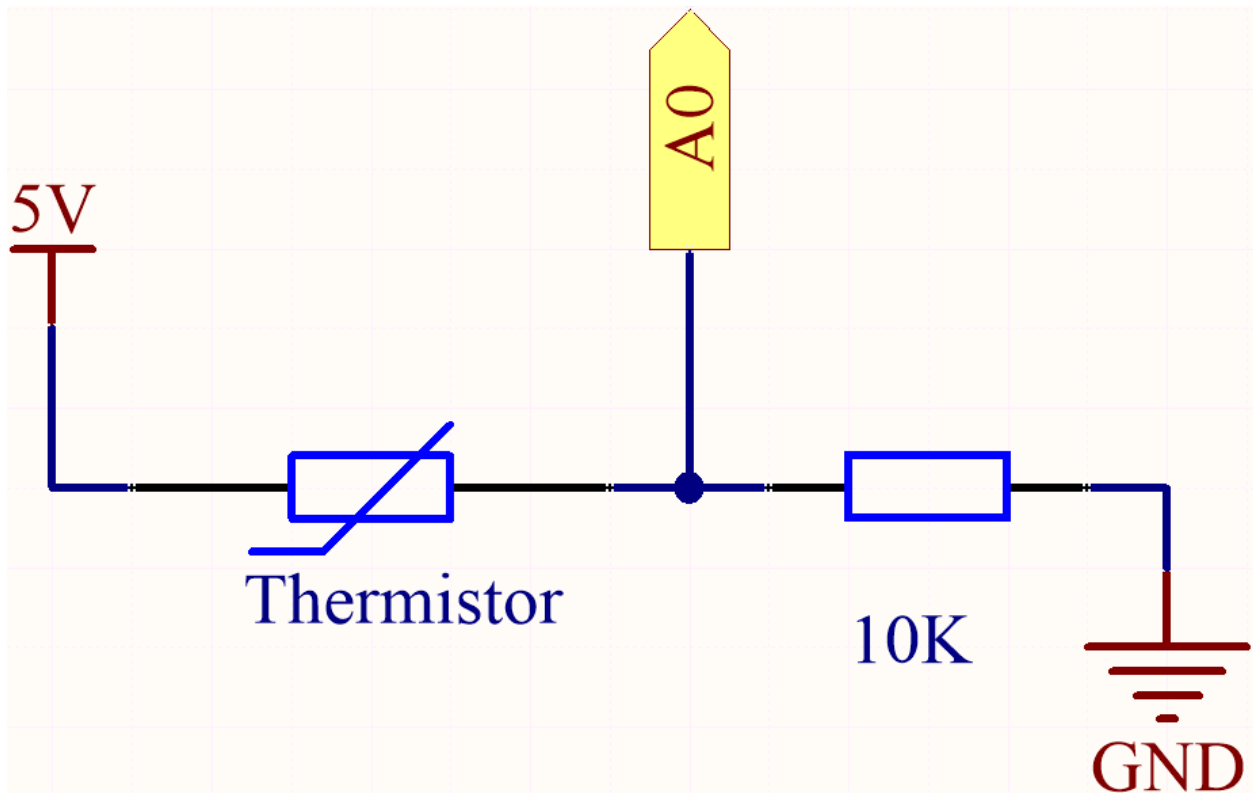
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Thermistor</i>	

Schaltplan



Jeder Thermistor hat einen Normalwiderstand. Hier beträgt er 10k Ohm, gemessen bei 25 Grad Celsius.

Wenn die Temperatur steigt, sinkt der Widerstand des Thermistors. Dann werden die Spannungsdaten vom A/D-Wandler in digitale Werte umgewandelt.

Die Temperatur in Celsius oder Fahrenheit wird über die Programmierung ausgegeben.

Hier ist die Beziehung zwischen Widerstand und Temperatur:

$$R_T = R_N \exp(B(1/T_K - 1/T_N))$$

- **R_T** ist der Widerstand des NTC-Thermistors bei der Temperatur **T_K** .
- **R_N** ist der Widerstand des NTC-Thermistors bei der Nenntemperatur **T_N** . Hier beträgt der Wert von **R_N** 10k.
- **T_K** ist eine Kelvin-Temperatur, und die Einheit ist K. Hier beträgt der Wert von **T_K** 273,15 + Grad Celsius.
- **T_N** ist eine Nenntemperatur in Kelvin; die Einheit ist ebenfalls K. Hier beträgt der Wert von **T_N** 273,15+25.
- Und **$B(\text{beta})$** , die Materialkonstante des NTC-Thermistors, wird auch als Wärmeempfindlichkeitsindex bezeichnet und hat den Wert 3950.
- **\exp** steht für exponentiell, und die Basiszahl **e** ist eine natürliche Zahl und beträgt ungefähr 2,7.

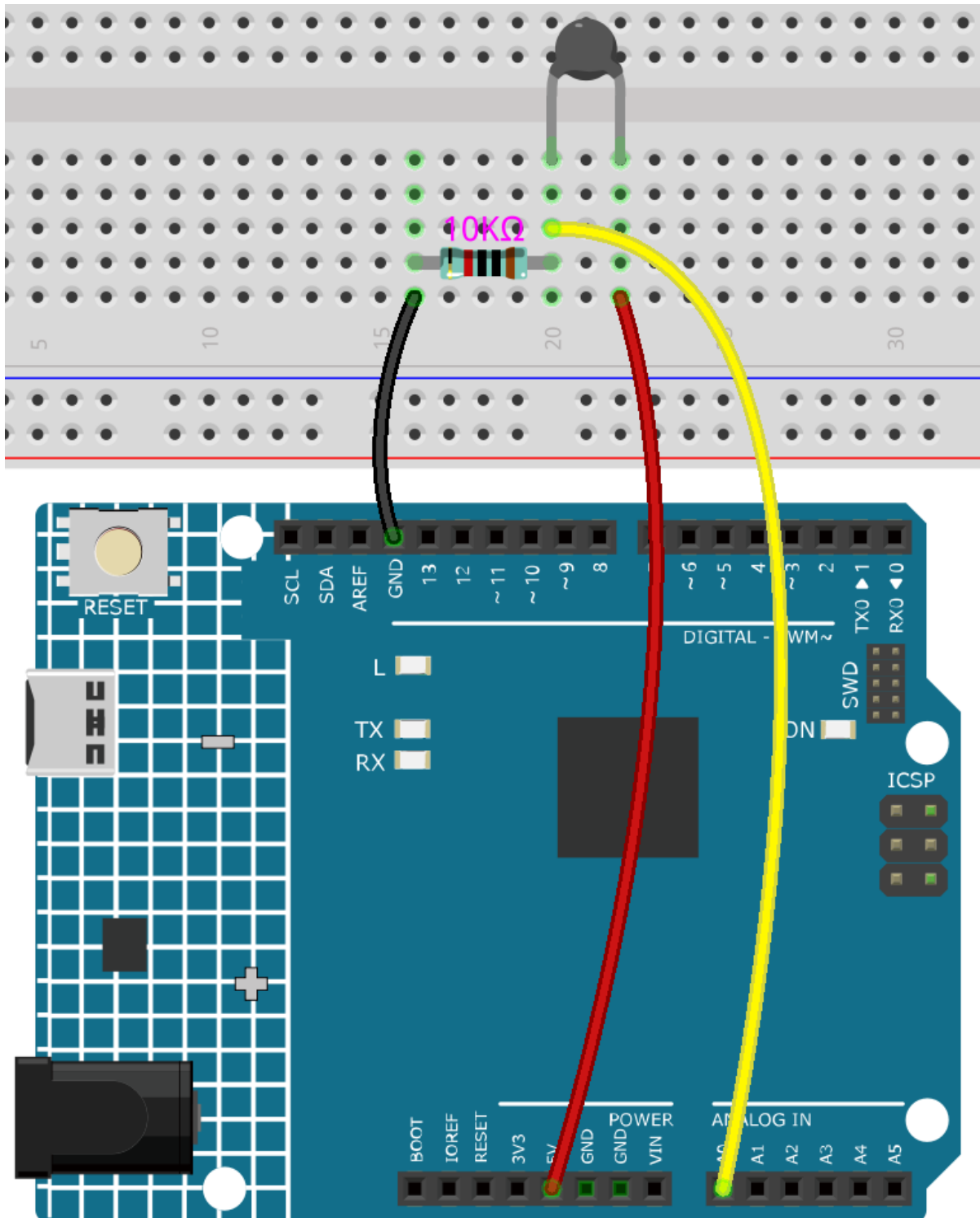
Diese Beziehung wird durch die Formel $T_K = 1 / (1 / (273,15 + 25) + B / R_N - 1 / R_T)$ dargestellt, um die Kelvin-Temperatur zu erhalten, die minus 273,15 Grad Celsius entspricht.

Dieses Verhältnis ist eine empirische Formel. Es ist nur dann genau, wenn Temperatur und Widerstand innerhalb des effektiven Bereichs liegen.

Verdrahtung

Bemerkung:

- Der Thermistor ist schwarz oder grün und mit 103 markiert.



Code

Bemerkung:

- Öffnen Sie die Datei `4.5_thermometer.ino` im Pfad `euler-kit/arduino/4.5_thermometer`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Vergessen Sie nicht, das Raspberry Pi Pico Board und den korrekten Port auszuwählen, bevor Sie auf die Schaltfläche Hochladen klicken.

Nachdem der Code erfolgreich hochgeladen wurde, gibt der serielle Monitor die Temperaturen in Celsius und Fahrenheit aus.

4.5 5. Mehr Syntax

In diesem Kapitel finden Sie einige Beispiele, die die grundlegende Logik veranschaulichen, wie die meisten Programme mit der Realität interagieren. Dies wird Ihnen helfen, sich mit der Arduino-Programmierung vertraut zu machen. Wenn Sie eine kreative Idee im Kopf haben, wird die Programmierung für Sie kein Hindernis mehr sein.

4.5.1 5.1 If else

Normalerweise verwenden wir bedingte Urteile, um die grundlegendsten Realitätsinteraktionsprojekte abzuschließen. Hier bauen wir ein Türerkennungssystem mit einem Reed-Schalter und einer LED, um diese Logik zu zeigen.

Befestigen Sie den Magneten auf einer Seite der Tür und den Reed-Schalter (mit Schaltung) auf der anderen Seite der Tür. Wenn die Tür geschlossen ist, nähert sich der Magnet dem Reed-Schalter, der ihn einschaltet.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

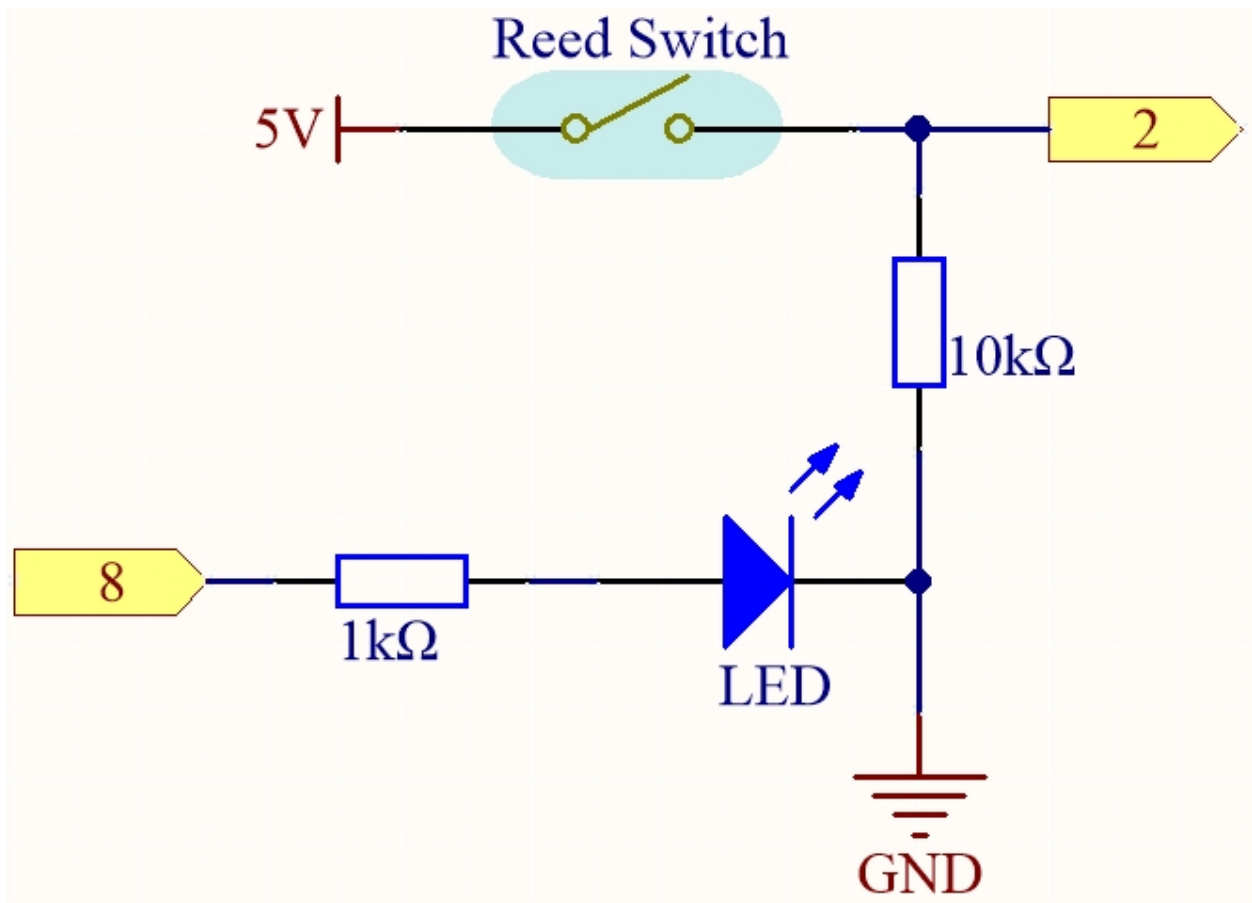
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Reedschalter</i>	-

Schaltplan



Verdrahtung



Bemerkung:

- Öffnen Sie die Datei `5.1.if_else.ino` im Pfad `3in1-kit\learning_project\5.1.if_else`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem der Code erfolgreich hochgeladen wurde, wird die LED leuchten, wenn Sie die Tür nicht schließen, um Sie daran zu erinnern, die Tür zu schließen.

Übrigens, wenn wir den umgekehrten Effekt benötigen (LED leuchtet, wenn die Tür geschlossen ist), müssen wir nur die Bedingung im `if` ändern.

- `if else`

Das `if else` ermöglicht eine größere Kontrolle über den Codefluss als die einfache `if`-Anweisung, indem mehrere Tests gruppiert werden können.

4.5.2 5.2 Schwellenwert

In vielen Projekten werden Sie auf solch ein Bedürfnis stoßen. „Wenn xxx ein bestimmtes Niveau erreicht, dann...“

Zum Beispiel in einem Smart Home: Wenn die Lichtintensität weniger als 50Lux beträgt, schalte das Licht ein; Ein weiteres Beispiel ist auf dem Computer-Motherboard: Wenn die Betriebstemperatur des CPUs höher als 65 Grad Celsius ist, schalte den Lüfter ein, und so weiter.

In diesen Anforderungen spiegelt sich das Schlüsselwort „Schwellenwert“ wider.

Wir können den Wert des Schwellenwerts anpassen, um den Schaltkreis stärker an individuelle Bedürfnisse anzupassen. Zum Beispiel, wenn ich ein helleres Wohnambiente bevorzuge, kann ich den Schwellenwert der automatischen Beleuchtung des Smart Homes auf 80Lux erhöhen. Ein weiteres Beispiel ist, wenn die Belüftungsumgebung meines Studios nicht sehr gut ist und der Bedarf an Wärmeableitung höher ist, kann der Schwellenwert für das automatische Öffnen des Lüfters auf 50 Grad Celsius eingestellt werden.

Hier verwenden wir einen Bodenfeuchtigkeitssensor und 2 LEDs, um einen Topfmonitor zu erstellen. Wenn der Boden zu trocken ist, leuchtet die rote LED; wenn der Boden feucht genug ist, leuchtet die grüne LED. Sie müssen die Schwellenwerte zur Bestimmung der Trockenheit und Feuchtigkeit des Bodens manuell anpassen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

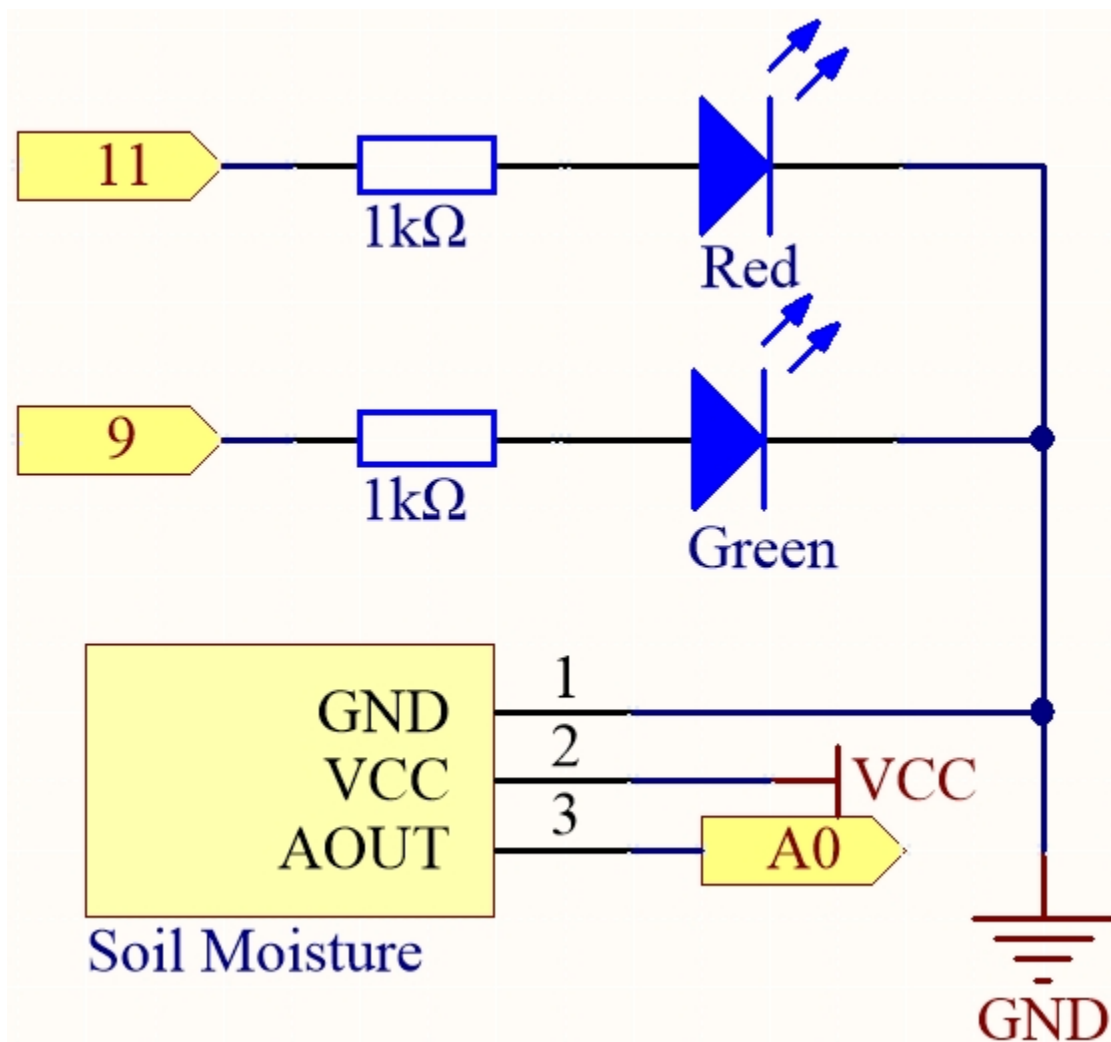
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

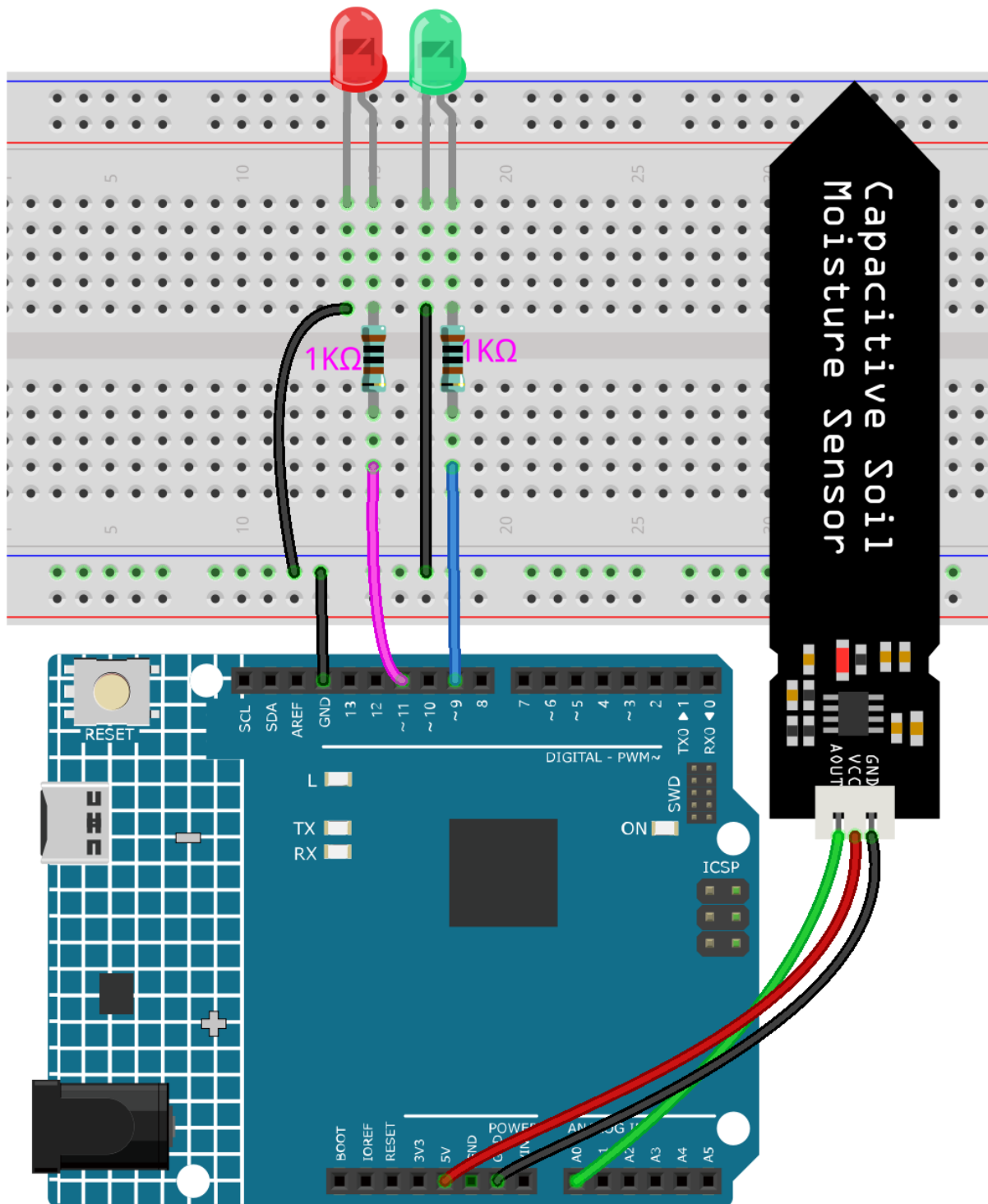
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Bodenfeuchtigkeitsmodul</i>	

Schaltplan



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.2.threshold.ino im Pfad 3in1-kit\learning_project\5.2.threshold.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem der Code erfolgreich hochgeladen wurde, und wenn Ihr Schwellenwert korrekt eingestellt ist, wird die rote LED leuchten, wenn der Boden trocken ist, um Sie daran zu erinnern, dass Sie gießen müssen; nach dem Gießen wird die grüne LED leuchten.

Wie funktioniert das?

```
...  
  
void loop() {  
    int sensorValue = analogRead(soilMoisture);  
    Serial.println(sensorValue);  
    if (sensorValue > threshold) {  
        digitalWrite(redPin, HIGH); // Turn the red LED  
        digitalWrite(greenPin, LOW); // green  
    } else {  
        digitalWrite(greenPin, HIGH); // Turn on the green LED  
        digitalWrite(redPin, LOW); // red  
    }  
}  
...
```

Zunächst setzen Sie einen Schwellenwert und lesen dann den Wert des Bodenfeuchtigkeitsmoduls; sein Wert nimmt ab, wenn der Feuchtigkeitsgrad steigt. Wenn der aktuell gelesene Wert größer als der eingestellte Schwellenwert ist, dann lassen Sie die rote LED leuchten, andernfalls schalten Sie die grüne LED ein.

Dieser Schwellenwert muss je nach tatsächlicher Situation angepasst werden. Sie können den Code zuerst hochladen, dann den seriellen Monitor öffnen, um den Wert zu überprüfen, den Wert in sowohl nassen als auch trockenen Zuständen aufzeichnen und dann einen Mittelwert als Schwellenwert wählen.

4.5.3 5.3 Zustandsänderungserkennung

Wenn der Knopf andere Geräte steuert, kann er nicht nur arbeiten, wenn er gedrückt wird, sondern auch stoppen, wenn er losgelassen wird. Es ist auch möglich, den Arbeitszustand jedes Mal umzuschalten, wenn der Knopf gedrückt wird.

Um diesen Effekt zu erzielen, müssen Sie wissen, wie Sie den Arbeitszustand zwischen Aus und Ein umschalten können, wenn der Knopf gedrückt wird. Das nennt man „Zustandsänderungserkennung“.

In diesem Projekt werden wir den Knopf verwenden, um den Motor zu steuern.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

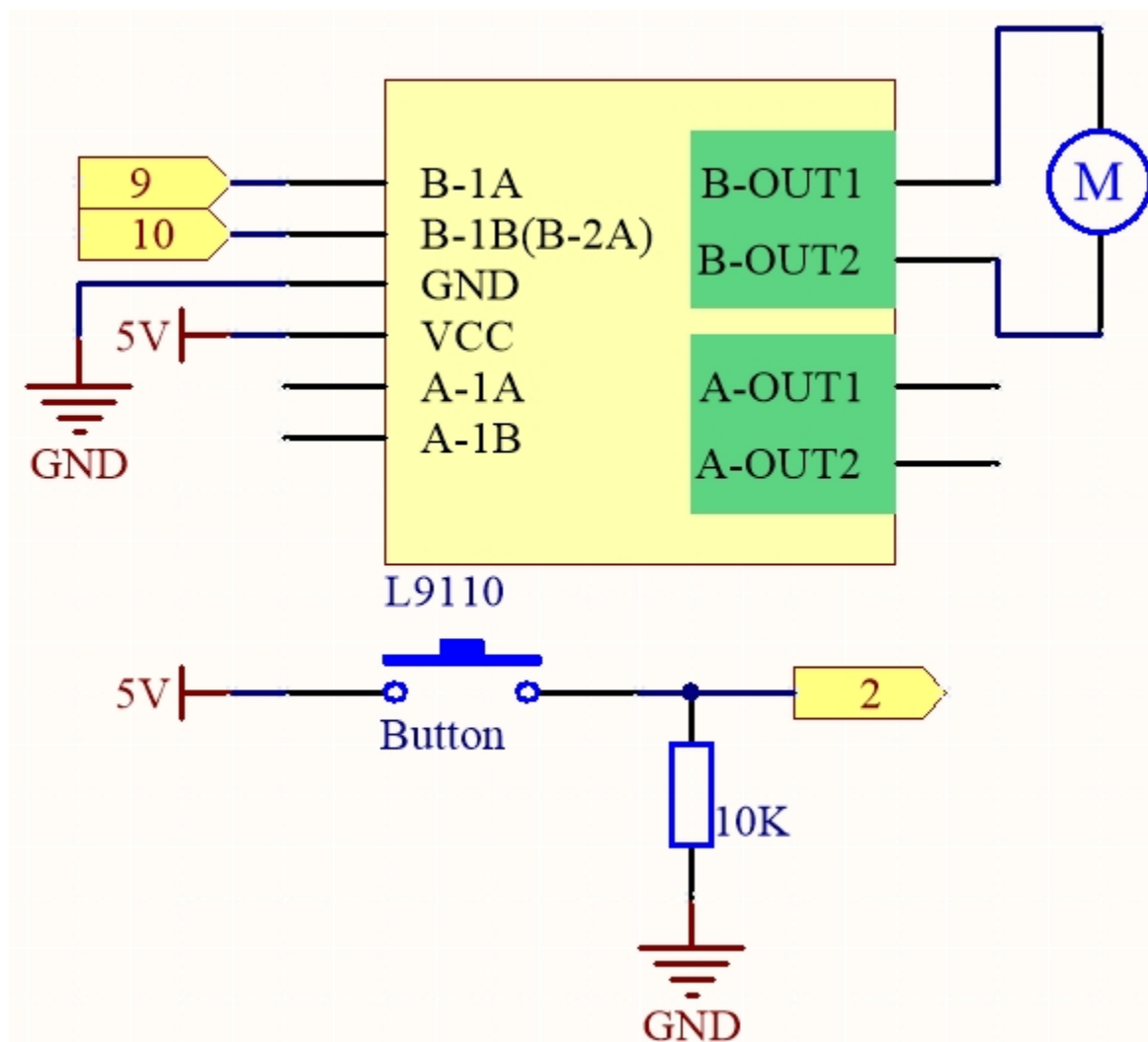
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

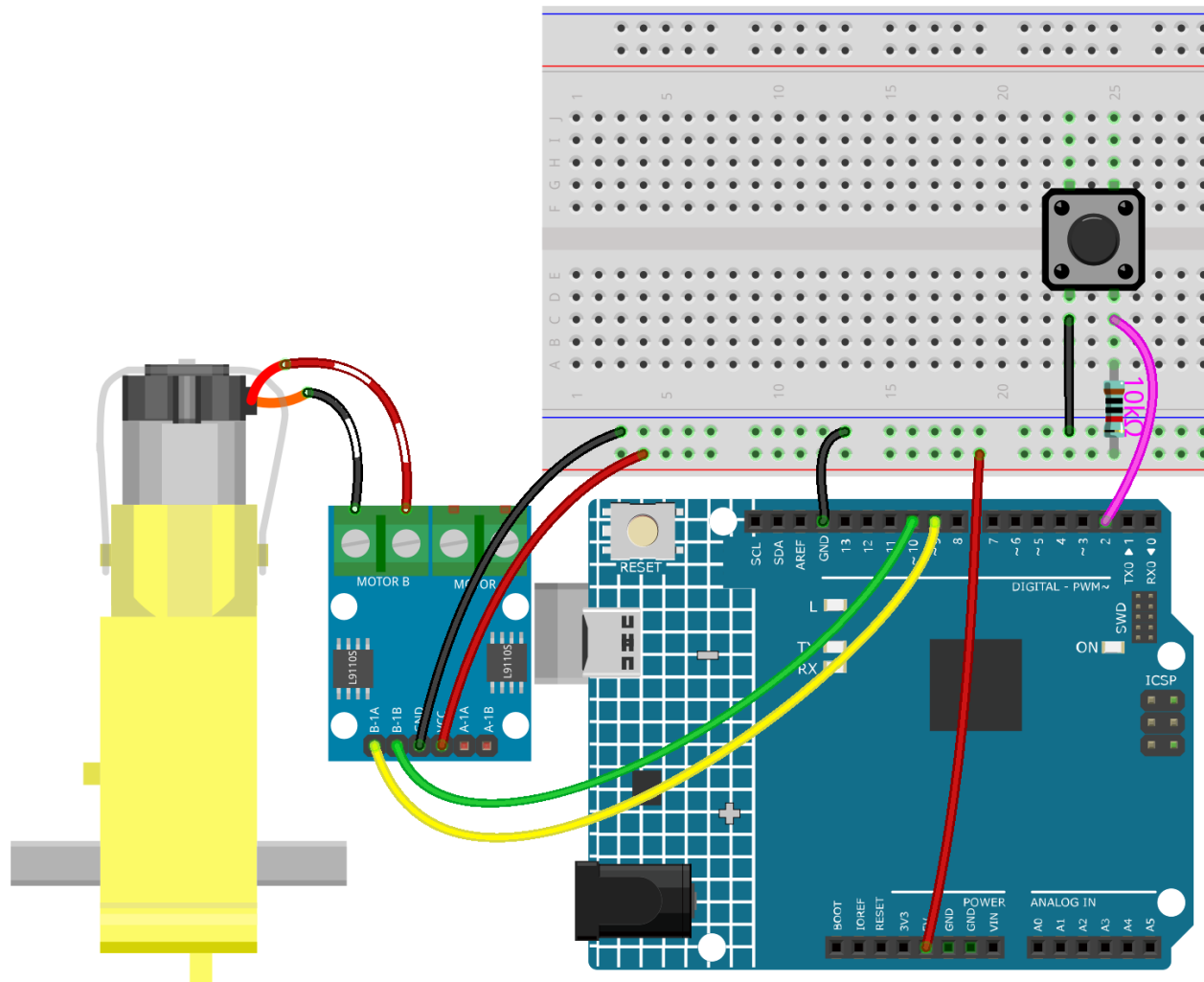
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Knopf</i>	
<i>TT Motor</i>	-
<i>L9110 Motortreibermodul</i>	-

Schaltplan



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.3.state_change_detection.ino im Pfad 3in1-kit\learning_project\5.3.state_change_detection.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem der Code erfolgreich hochgeladen wurde, drücken Sie den Knopf und der Motor wird sich drehen; bis Sie den Knopf erneut drücken, wird der Motor stoppen.

Wie funktioniert das?

1. Erstellen Sie Variablen und definieren Sie Pins für den Motor und den Knopf.

```
...
int detectionState = 0;
int buttonState = 0;
int lastButtonState = 0;
```

- detectionState ist eine Flagge, deren Wert sich jedes Mal ändert, wenn der Knopf gedrückt

wird, z.B. 0 dieses Mal, 1 das nächste Mal und so weiter abwechselnd.

- `buttonState` und `lastButtonState` werden verwendet, um den Zustand des Knopfes dieses Mal und das letzte Mal aufzuzeichnen, um zu vergleichen, ob der Knopf gedrückt oder losgelassen wurde.

2. Initialisieren Sie jeden Pin und setzen Sie die Baudrate des seriellen Monitors.

```
void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
  pinMode(A_1A, OUTPUT);
  pinMode(A_1B, OUTPUT);
}
```

3. Zuerst lesen Sie den Zustand des Knopfes, und wenn der Knopf gedrückt ist, wird die Variable `detectionState` ihren Wert von 0 auf 1 oder von 1 auf 0 umschalten. Wenn `detectionState` 1 ist, wird der Motor eingeschaltet. Es hat die Wirkung, dass dieses Mal, wenn der Knopf gedrückt wird, der Motor sich dreht, das nächste Mal, wenn der Knopf gedrückt wird, der Motor stoppt und so weiter abwechselnd.

```
void loop() {
  // Toggle the detectionState each time the button is pressed
  buttonState = digitalRead(buttonPin);
  if (buttonState != lastButtonState) {
    if (buttonState == HIGH) {
      detectionState=(detectionState+1)%2;
      Serial.print("The detection state is: ");
      Serial.println(detectionState);
    }
    delay(50);
  }
  lastButtonState = buttonState;

  // According to the detectionState, start the motor
  if(detectionState==1){
    digitalWrite(A_1A,HIGH);
    digitalWrite(A_1B,LOW);
  }else{
    digitalWrite(A_1A,LOW);
    digitalWrite(A_1B,LOW);
  }
}
```

Der gesamte Arbeitsablauf ist wie folgt.

- Lesen Sie den Knopfwert.

```
buttonState = digitalRead(buttonPin);
```

- Wenn `buttonState` und `lastButtonState` nicht gleich sind, bedeutet dies, dass sich der Knopfzustand geändert hat. Fahren Sie mit der nächsten Beurteilung fort und speichern Sie den Knopfzustand zu diesem Zeitpunkt in die Variable `lastButtonState`. `delay(50)` wird verwendet, um das Prellen zu eliminieren.

```
if (buttonState != lastButtonState) {
  ...
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

    delay(50);
}
lastButtonState = buttonState;

```

- Wenn der Knopf gedrückt wird, ist sein Wert HIGH. Hier, wenn der Knopf gedrückt wird, wird der Wert der Variable detectionState geändert, z.B. von 0 auf 1 nach einer Operation.

```

if (buttonState == HIGH) {
    detectionState=(detectionState+1)%2;
    Serial.print("The detection state is: ");
    Serial.println(detectionState);
}

```

- Wenn die Variable detectionState 1 ist, lassen Sie den Motor drehen, sonst stoppen Sie ihn.

```

if(detectionState==1){
    digitalWrite(A_1A,HIGH);
    digitalWrite(A_1B,LOW);
}else{
    digitalWrite(A_1A,LOW);
    digitalWrite(A_1B,LOW);
}

```

4.5.4 5.4 Intervall

Manchmal müssen Sie zwei Dinge gleichzeitig tun. Zum Beispiel möchten Sie vielleicht eine LED blinken lassen, während Sie einen Tastendruck lesen. In diesem Fall können Sie nicht `delay()` verwenden, weil Arduino Ihr Programm während des `delay()` pausiert. Wenn der Knopf gedrückt wird, während Arduino pausiert und auf das Ende von `delay()` wartet, wird Ihr Programm den Tastendruck verpassen.

Ein passendes Beispiel wäre, eine Pizza in Ihrer Mikrowelle aufzuwärmen und gleichzeitig auf eine wichtige E-Mail zu warten. Sie legen die Pizza in die Mikrowelle und stellen sie auf 10 Minuten ein. Das Äquivalent zur Verwendung von `delay()` wäre, vor der Mikrowelle zu sitzen und dem Timer zuzusehen, wie er von 10 Minuten auf Null herunterzählt. Wenn die wichtige E-Mail in dieser Zeit ankommt, werden Sie sie verpassen.

Was Sie im wirklichen Leben tun würden, wäre, die Pizza einzuschalten und dann Ihre E-Mails zu überprüfen und vielleicht etwas anderes zu tun (was nicht allzu lange dauert!) und ab und zu würden Sie zur Mikrowelle zurückkehren, um zu sehen, ob der Timer Null erreicht hat und somit Ihre Pizza fertig ist.

Diese Skizze zeigt, wie man einen Buzzer ohne die Verwendung von `delay()` tönen kann. Sie schaltet den Buzzer ein und notiert dann die Zeit. Danach überprüft sie bei jedem Durchlauf von `loop()`, ob das gewünschte Intervall abgelaufen ist. Ist dies der Fall, wird der Buzzer getönt und die neue Zeit notiert. Auf diese Weise tönt der Buzzer kontinuierlich, während die Sketch-Ausführung nie bei einer einzigen Anweisung ins Stocken gerät.

Basierend auf dieser Bedingung können wir den Code des Knopfes hinzufügen, um die LED zu steuern, sie wird nicht durch den Buzzer, der Musik spielt, gestört.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

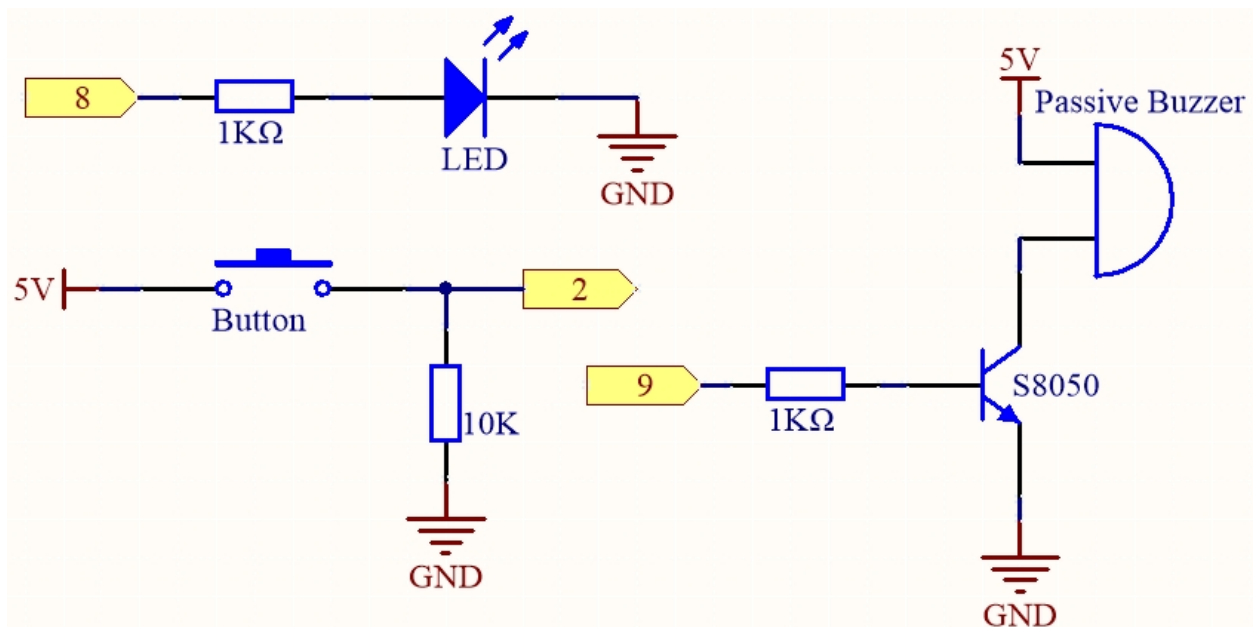
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Knopf</i>	
<i>Summer</i>	

Schaltplan



Verdrahtung



Bemerkung:

- Öffnen Sie die Datei `5.4.interval.ino` im Verzeichnis `3in1-kit\learning_project\5.4.interval`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nach erfolgreichem Hochladen des Codes wird der Summer Musik spielen; immer wenn Sie den Knopf drücken, leuchtet die LED auf. Die Arbeit von LED und Summer beeinflusst sich nicht gegenseitig.

Wie funktioniert das?

Initialisieren Sie eine Variable namens `previousMillis`, um die vorherige Betriebszeit des Mikrocontrollers zu speichern.

```
unsigned long previousMillis = 0;
```

Markieren Sie, welche Note gespielt wird.

```
int thisNote=0;
```

Die Intervallzeit jeder Note.

```
long interval = 1000;
```

In `loop()` deklarieren Sie `currentMillis`, um die aktuelle Zeit zu speichern.

```
unsigned long currentMillis = millis();
```

Wenn das Intervall zwischen der aktuellen Betriebszeit und der letzten Aktualisierungszeit größer als 1000ms ist, werden bestimmte Funktionen ausgelöst. Gleichzeitig wird `previousMillis` auf die aktuelle Zeit aktualisiert, um das nächste Auslösen, das 1 Sekunde später stattfinden soll, vorzubereiten.

```
if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis; // save the last time of the last tone
    //...
}
```

Spielen Sie die Noten in der Melodie nacheinander ab.

```
tone(buzzerPin,melody[thisNote],100);
interval=1000/noteDurations[thisNote]; // interval at which to tone
thisNote=(thisNote+1)%(sizeof(melody)/2); //iterate over the notes of the melody
```

Der Knopf steuert die LED.

```
// play button & led
digitalWrite(ledPin,digitalRead(buttonPin));
```

4.5.5 5.5 Interne Bibliothek nutzen

In der Arduino IDE können Sie viele integrierte Bibliotheken nutzen, indem Sie die entsprechende .h-Datei direkt in Ihren Code einfügen.

In diesem Projekt wird die Bibliothek Servo verwendet, um den Servo anzusteuern, sodass er sich zwischen 0° und 180° drehen kann.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

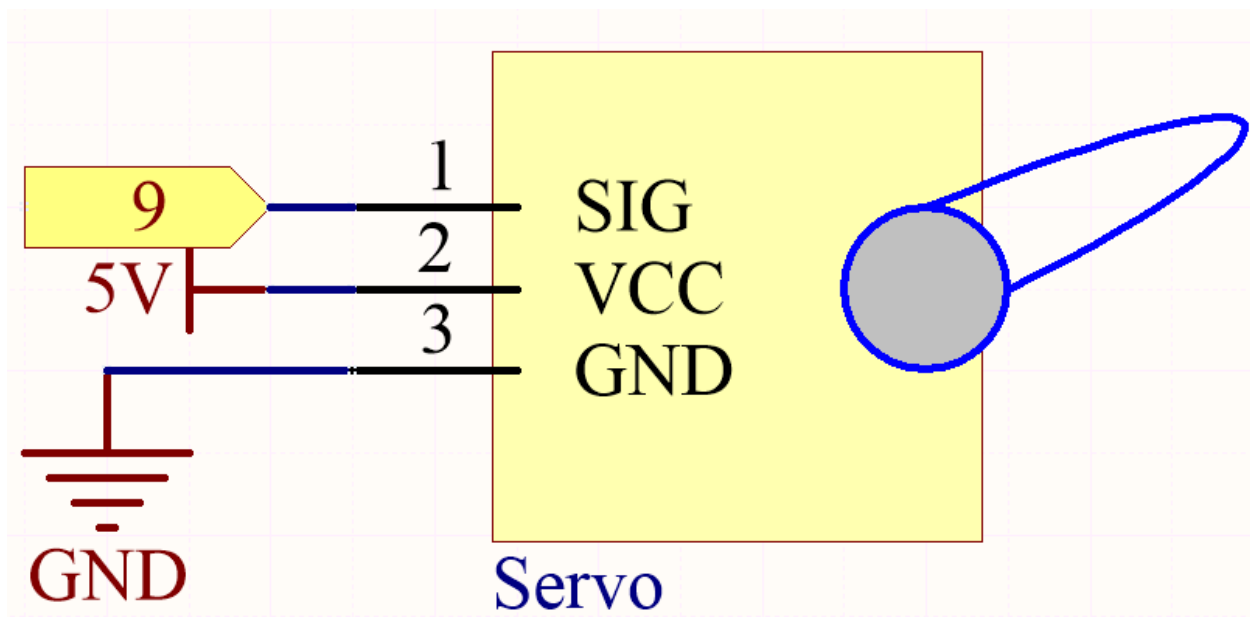
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können diese auch separat über die untenstehenden Links kaufen.

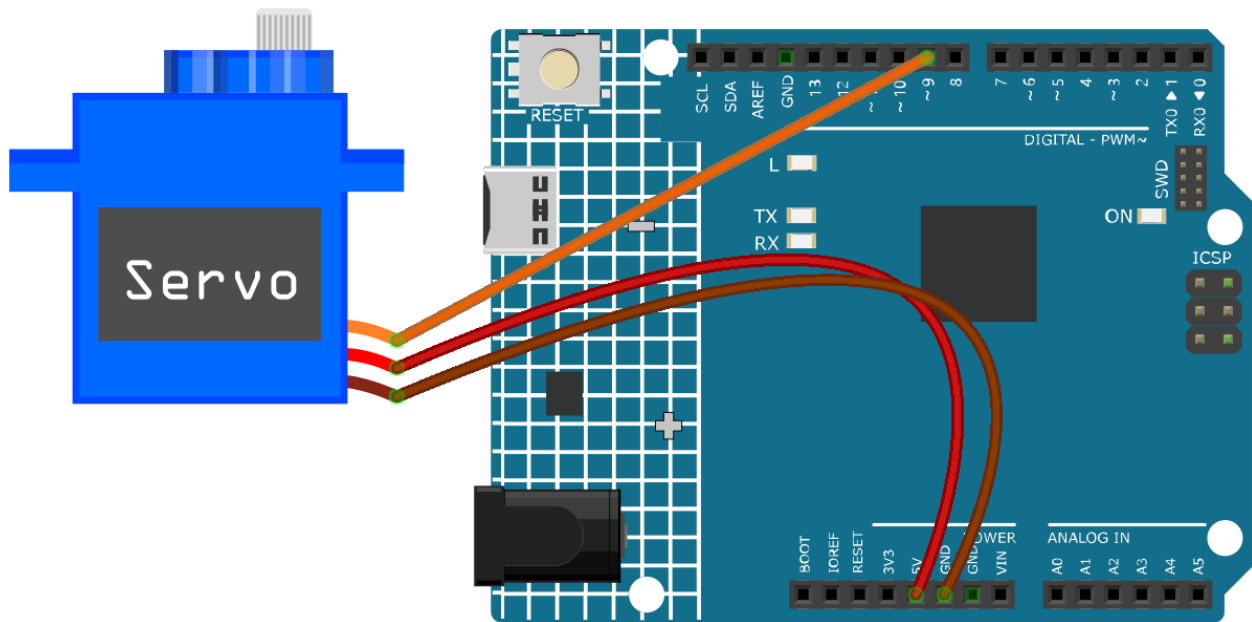
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Jumper-Kabel</i>	
<i>Servo</i>	

Schaltplan



In diesem Projekt verwenden wir den PWM-Pin 9, um den Servo anzusteuern. Das orangefarbene Kabel des Servos wird mit dem PWM-Pin 9, das rote mit 5V und das braune mit GND verbunden.

Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.5.use_internal_library.ino im Pfad 3in1-kit\learning_project\5.5.use_internal_library.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem Sie die Codes erfolgreich auf das R4-Board hochgeladen haben, sehen Sie den Servo-Arm im Bereich von 0° bis 180° rotieren.

Wie funktioniert das?

Mit dem Aufruf der Bibliothek `Servo.h` können Sie den Servo einfach steuern.

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

Bibliotheksfunktionen:

```
Servo
```

Erstellt ein **Servo**-Objekt zur Steuerung eines Servos.

```
uint8_t attach(int pin);
```

Ruft `pinMode()` auf, um einen Pin in einen Servo-Treiber zu verwandeln und gibt bei einem Fehler 0 zurück.

```
void detach();
```

Löst einen Pin vom Servo-Treiben.

```
void write(int value);
```

Stellt den Winkel des Servos in Grad ein, von 0 bis 180.

```
int read();
```

Gibt den Wert zurück, der mit dem letzten `write()` eingestellt wurde.

```
bool attached();
```

Gibt 1 zurück, wenn der Servo derzeit angehängt ist.

4.5.6 5.6 Map

Wenn Sie genau hinschauen, werden Sie feststellen, dass viele Werte in der Programmierung unterschiedliche Bereiche haben. Zum Beispiel liegt der Wertebereich für analoge Eingänge bei (0~1023). Der Wertebereich für den analogen Ausgang liegt bei (0~255). Der Ausgangswinkel des Servos beträgt (0~180).

Das bedeutet, dass wir, wenn wir den Potentiometer nutzen wollen, um die Helligkeit der LED oder den Winkel des Servos zu steuern, eine Zuordnungsoperation durchführen müssen.

Lassen Sie uns nun sehen, wie dies umgesetzt wird.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

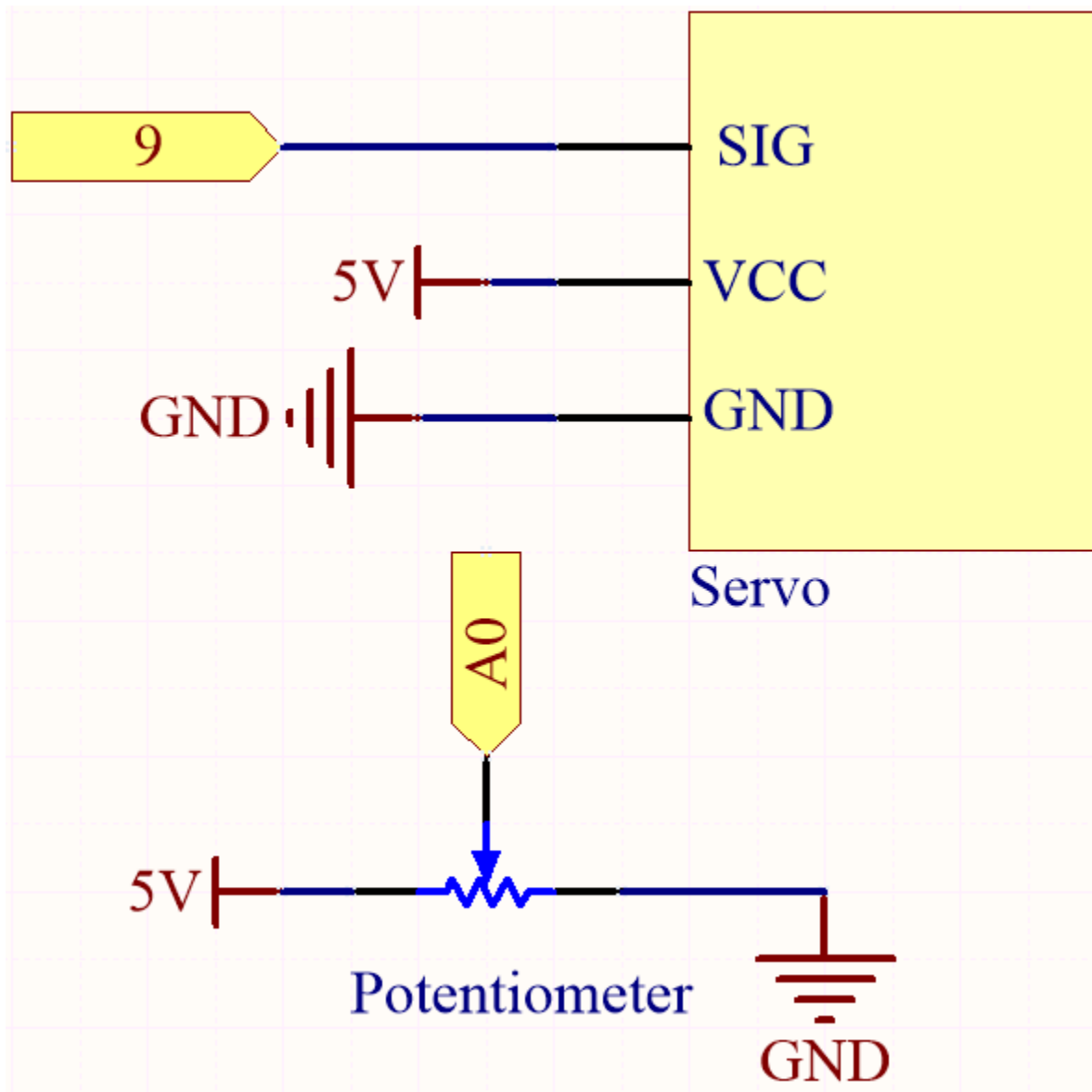
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

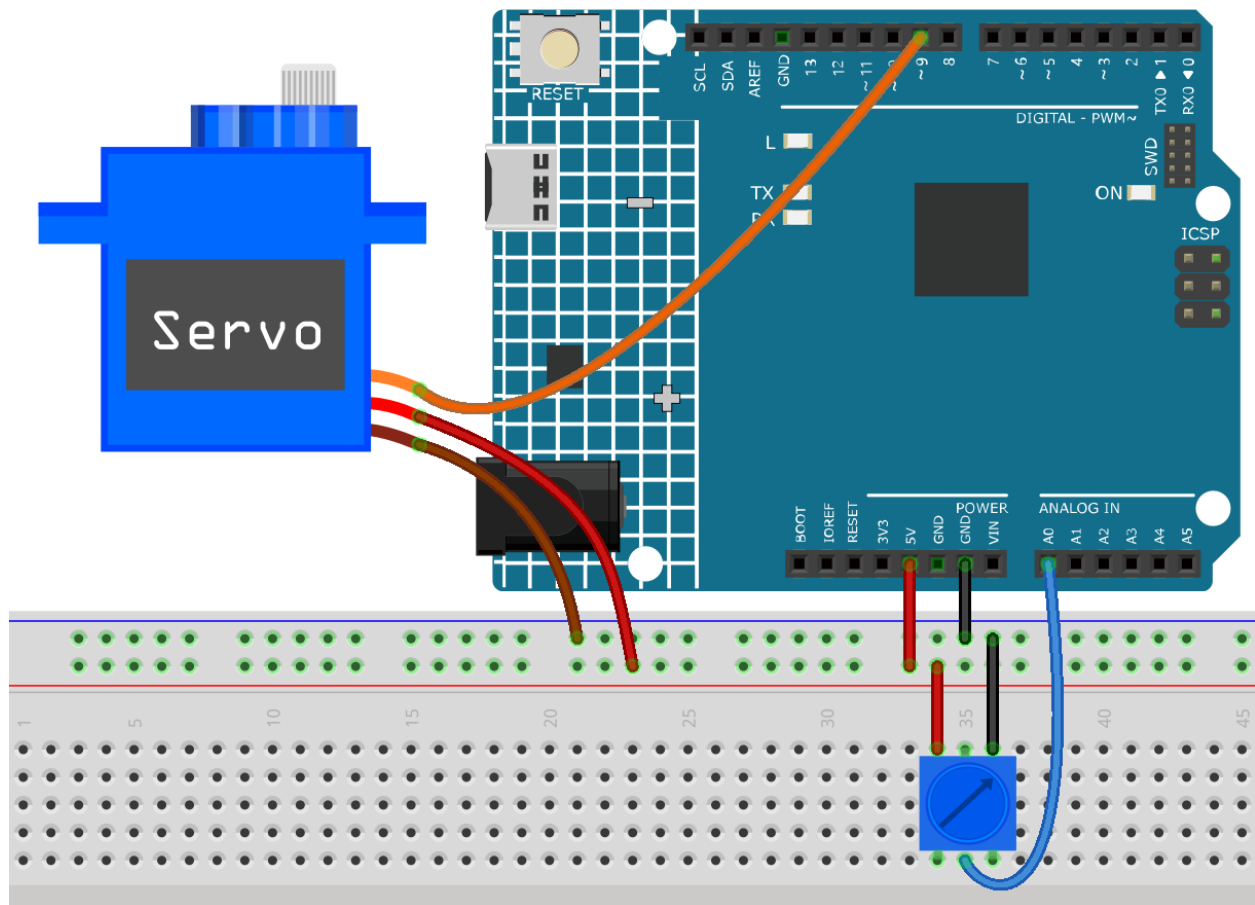
Sie können diese auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Servo</i>	
<i>Potentiometer</i>	

Schaltplan



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.6.map.ino im Pfad 3in1-kit\learning_project\5.6.map.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem Sie den Code erfolgreich hochgeladen haben, können Sie den Potentiometer hin- und herbewegen, und die Ausgangswelle des Servos wird entsprechend rotieren.

Wie funktioniert das?

`map(value, fromLow, fromHigh, toLow, toHigh)`: Ordnet eine Zahl einem anderen Bereich zu. Das bedeutet, dass ein fromLow-Wert zu toLow und ein fromHigh-Wert zu toHigh zugeordnet wird.

Syntax

`map(value, fromLow, fromHigh, toLow, toHigh)`

Parameter

- **value**: die zuordnende Zahl.
- **fromLow**: die untere Grenze des aktuellen Bereichs des Wertes.
- **fromHigh**: die obere Grenze des aktuellen Bereichs des Wertes.
- **toLow**: die untere Grenze des Zielbereichs des Wertes.

- `toHigh`: die obere Grenze des Zielbereichs des Wertes.

Wenn der Potentiometer die LED steuert, können Sie ebenfalls die Funktion `map` verwenden, um die Aufgabe zu erfüllen.

```
int x = analogRead(knob);
int y = map(x,0,1023,0,255);
analogWrite(led,y);
```

Anmerkungen und Warnungen

- Die „untere Grenze“ beider Bereiche kann größer oder kleiner sein als die „obere Grenze“, was bedeutet, dass die Funktion `map()` verwendet werden kann, um einen Bereich von Zahlen umzukehren.

```
y = map(x,0,180,180,0);
```

- Das Mapping funktioniert auch gut für negative Zahlen.

```
y = map(x,0,1023,-90,90);
```

- Das Mapping verwendet Ganzzahlen, und die Dezimalstellen von Fließkommazahlen werden verworfen.

4.5.7 5.7 Tone() oder noTone()

`Tone()` wird verwendet, um eine Rechteckwelle der angegebenen Frequenz (und 50% Tastverhältnis) an einem Pin zu erzeugen. Eine Dauer kann angegeben werden, sonst setzt sich die Welle fort, bis ein Aufruf von `noTone()` erfolgt.

In diesem Projekt nutzen wir diese beiden Funktionen, um den passiven Summer zum Vibrieren und somit zum Tönen zu bringen. Wie der aktive Summer verwendet auch der passive Summer das Phänomen der elektromagnetischen Induktion. Der Unterschied ist, dass ein passiver Summer keine Oszillationsquelle hat, sodass er bei Gleichstromsignalen nicht piept. Dies ermöglicht jedoch dem passiven Summer, seine eigene Oszillationsfrequenz anzupassen und unterschiedliche Töne wie „do, re, mi, fa, sol, la, si“ zu erzeugen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

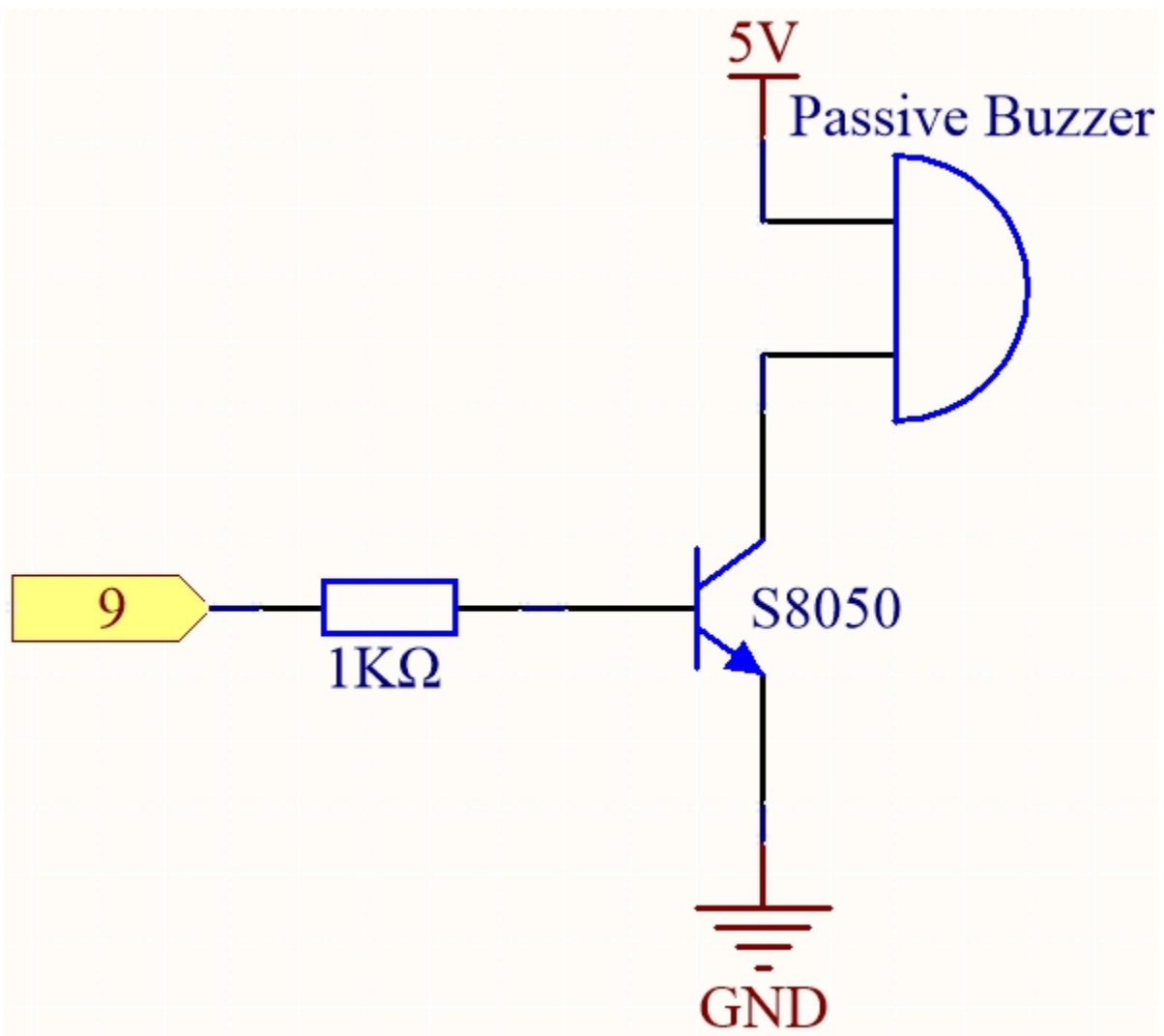
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können diese auch separat über die untenstehenden Links kaufen.

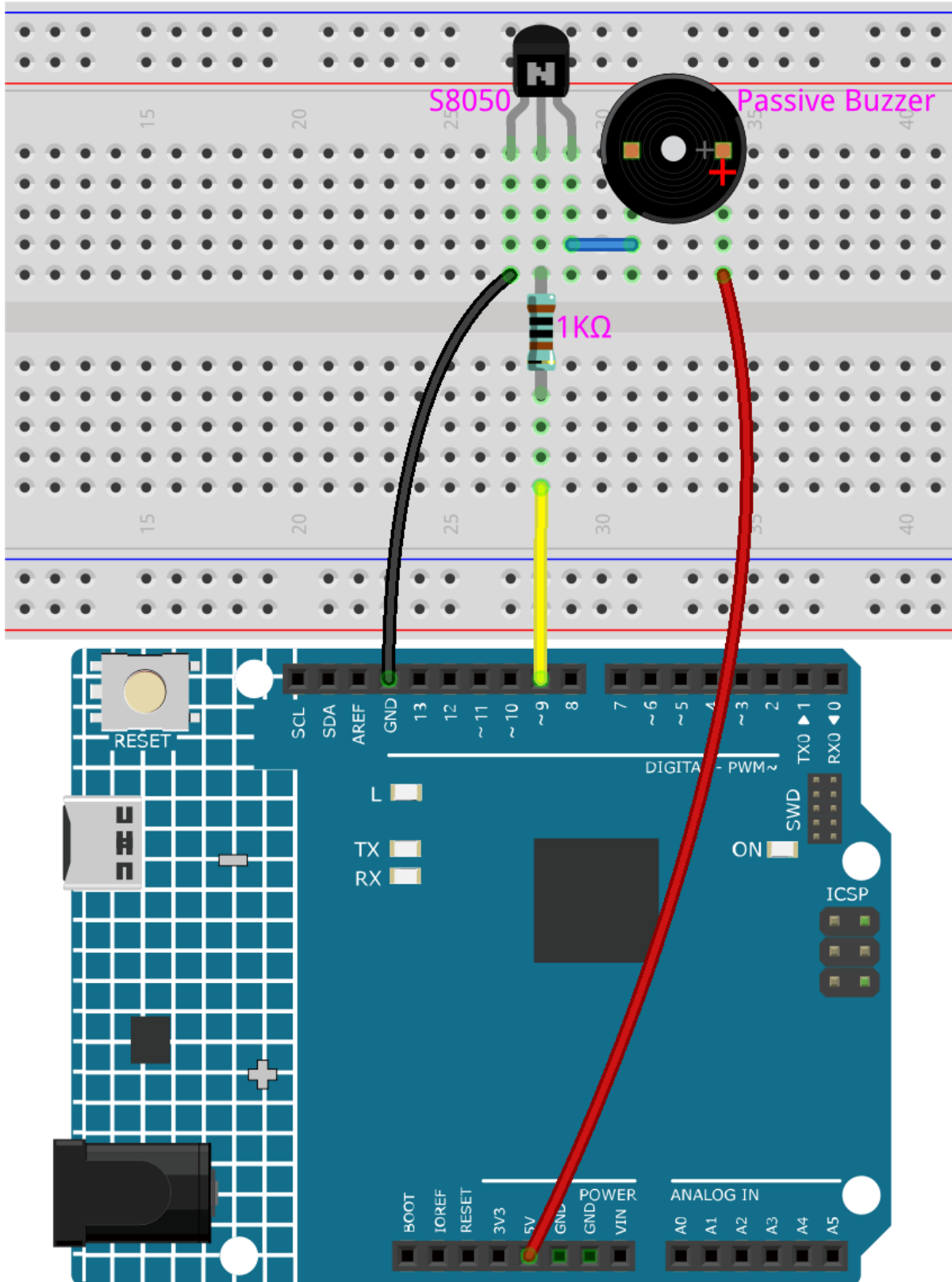
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Summer</i>	

Schaltplan



Verbinden Sie die Kathode des Summers mit GND und die Anode mit dem digitalen Pin 9.

Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei `5.7.tone_notone.ino` im Pfad `3in1-kit\learning_project\5.7.tone_notone`.
 - Oder kopieren Sie diesen Code in die **Arduino IDE**.
-

Wenn Sie den Code erfolgreich auf das R4-Board hochgeladen haben, können Sie eine Melodie mit sieben Tönen hören.

Wie funktioniert das?

Es gibt zwei Punkte, auf die Sie achten sollten:

1. `tone()` & `noTone()`: Mit dieser Funktion können Sie den Klang des passiven Summers direkt steuern, und ihr Prototyp lautet wie folgt:

Syntax

```
void tone(int pin, unsigned int frequency)
```

```
void tone(int pin, unsigned int frequency, unsigned long duration)
```

Parameter

- **pin**: Der Arduino-Pin, auf dem der Ton erzeugt werden soll.
- **frequency**: Die Frequenz des Tons in Hertz.
- **duration**: Die Dauer des Tons in Millisekunden (optional)

Generiert eine Rechteckwelle der angegebenen Frequenz (und 50% Tastverhältnis) an einem Pin (um den passiven Summer zum Tönen zu bringen). Eine Dauer kann angegeben werden, sonst setzt sich die Welle fort, bis ein Aufruf zu `noTone()` erfolgt. Der Pin kann mit einem Piezo-Summer oder einem anderen Lautsprecher verbunden werden, um Töne zu spielen.

Zu einem Zeitpunkt kann nur ein Ton erzeugt werden. Wenn bereits ein Ton auf einem anderen Pin spielt, hat der Aufruf von `tone()` keine Wirkung. Spielt der Ton auf demselben Pin, setzt der Aufruf seine Frequenz.

Die Verwendung der Funktion `tone()` beeinträchtigt die PWM-Ausgabe an den Pins 3 und 11.

Es ist nicht möglich, Töne zu erzeugen, die niedriger sind als 31Hz.

Syntax

```
void noTone(int pin)
```

Parameter

pin: Der Arduino-Pin, auf dem der Ton erzeugt werden soll.

Beendet die Generierung einer durch `tone()` ausgelösten Rechteckwelle. Hat keine Wirkung, wenn kein Ton erzeugt wird.

Nachdem Sie die beiden Funktionen kennengelernt haben, können Sie den Code verstehen – die Erstellung der Arrays `melody[]` und `noteDurations[]` dient als Vorbereitung für die nachfolgenden Aufrufe der Funktion `tone()` und das Ändern von Ton und Dauer in der Schleife, um einen besseren Musikspiel-Effekt zu erzielen.

2. **pitches.h**: Der Code verwendet eine zusätzliche Datei namens `pitches.h`. Diese Datei enthält alle Tonhöhenwerte für typische Noten. Zum Beispiel entspricht `NOTE_C4` dem mittleren C. `NOTE_FS4` ist Fis, und so weiter. Diese Notentabelle wurde ursprünglich von Brett Hagman erstellt, auf dessen Arbeit der Befehl `tone()` basiert. Sie könnte nützlich sein, wenn Sie musikalische Noten erzeugen möchten.

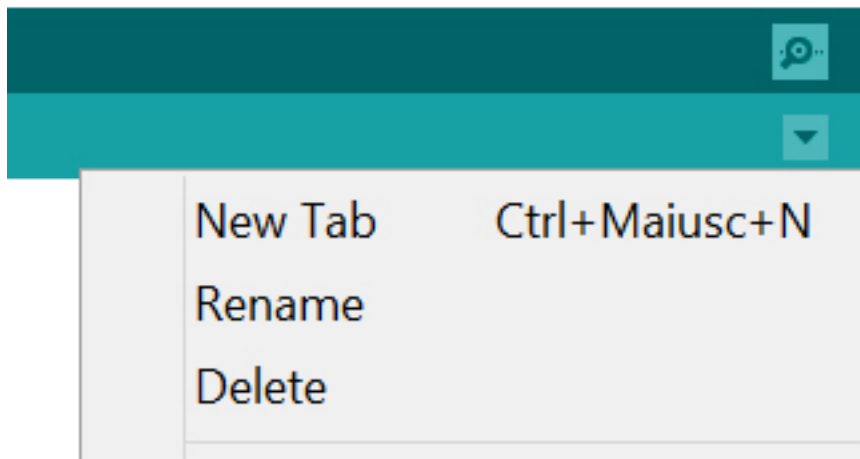
```
#include "pitches.h"
```

Bemerkung: In diesem Beispielprogramm gibt es bereits eine Datei `pitches.h`. Wenn Sie sie zusammen mit dem Hauptcode in einem Ordner ablegen, können die nachfolgenden Schritte zum Installieren von `pitches.h` weggelassen werden.



Wenn Sie die Code-Datei öffnen und die `pitches.h`-Datei nicht öffnen können, können Sie diese einfach manuell erstellen. Die Schritte sind wie folgt:

Um die Datei `pitches.h` zu erstellen, klicken Sie entweder auf die Schaltfläche direkt unter dem Symbol für den seriellen Monitor und wählen Sie **New Tab**, oder verwenden Sie **Ctrl+Shift+N**.



Fügen Sie dann den folgenden Code ein und speichern Sie ihn als `pitches.h`:

```

/*****
Public Constants
*****/
#define NOTE_B0  31
#define NOTE_C1  33
#define NOTE_CS1 35
#define NOTE_D1  37
#define NOTE_DS1 39
#define NOTE_E1  41
#define NOTE_F1  44
#define NOTE_FS1 46
#define NOTE_G1  49
#define NOTE_GS1 52
#define NOTE_A1  55
#define NOTE_AS1 58

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
#define NOTE_B1 62
#define NOTE_C2 65
#define NOTE_CS2 69
#define NOTE_D2 73
#define NOTE_DS2 78
#define NOTE_E2 82
#define NOTE_F2 87
#define NOTE_FS2 93
#define NOTE_G2 98
#define NOTE_GS2 104
#define NOTE_A2 110
#define NOTE_AS2 117
#define NOTE_B2 123
#define NOTE_C3 131
#define NOTE_CS3 139
#define NOTE_D3 147
#define NOTE_DS3 156
#define NOTE_E3 165
#define NOTE_F3 175
#define NOTE_FS3 185
#define NOTE_G3 196
#define NOTE_GS3 208
#define NOTE_A3 220
#define NOTE_AS3 233
#define NOTE_B3 247
#define NOTE_C4 262
#define NOTE_CS4 277
#define NOTE_D4 294
#define NOTE_DS4 311
#define NOTE_E4 330
#define NOTE_F4 349
#define NOTE_FS4 370
#define NOTE_G4 392
#define NOTE_GS4 415
#define NOTE_A4 440
#define NOTE_AS4 466
#define NOTE_B4 494
#define NOTE_C5 523
#define NOTE_CS5 554
#define NOTE_D5 587
#define NOTE_DS5 622
#define NOTE_E5 659
#define NOTE_F5 698
#define NOTE_FS5 740
#define NOTE_G5 784
#define NOTE_GS5 831
#define NOTE_A5 880
#define NOTE_AS5 932
#define NOTE_B5 988
#define NOTE_C6 1047
#define NOTE_CS6 1109
#define NOTE_D6 1175
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
#define NOTE_GS6 1661
#define NOTE_A6 1760
#define NOTE_AS6 1865
#define NOTE_B6 1976
#define NOTE_C7 2093
#define NOTE_CS7 2217
#define NOTE_D7 2349
#define NOTE_DS7 2489
#define NOTE_E7 2637
#define NOTE_F7 2794
#define NOTE_FS7 2960
#define NOTE_G7 3136
#define NOTE_GS7 3322
#define NOTE_A7 3520
#define NOTE_AS7 3729
#define NOTE_B7 3951
#define NOTE_C8 4186
#define NOTE_CS8 4435
#define NOTE_D8 4699
#define NOTE_DS8 49

```

4.5.8 5.8 Benutzerdefinierte Funktion

In C können wir ein umfangreiches Programm in grundlegende Bausteine unterteilen, die als Funktion bekannt sind. Eine Funktion enthält eine Menge von Programmieranweisungen, die von { } eingeschlossen sind. Eine Funktion kann mehrmals aufgerufen werden, um Wiederverwendbarkeit und Modularität im C-Programm zu gewährleisten. Mit anderen Worten, eine Sammlung von Funktionen bildet ein Programm. In anderen Programmiersprachen wird die Funktion auch als Prozedur oder Unterprogramm bezeichnet.

Es gibt folgende Vorteile von Funktionen:

- Mit Funktionen können wir vermeiden, dieselbe Logik/den gleichen Code in einem Programm immer wieder neu zu schreiben.
- Wir können C-Funktionen beliebig oft und von beliebigen Stellen in einem Programm aufrufen.
- Ein umfangreiches C-Programm lässt sich leichter nachverfolgen, wenn es in mehrere Funktionen unterteilt ist.
- Die Hauptleistung von C-Funktionen ist die Wiederverwendbarkeit.
- Ein Funktionsaufruf ist jedoch immer ein Overhead in einem C-Programm.

Es gibt zwei Arten von Funktionen in der C-Programmierung:

- **Library Functions:** die Funktionen, die in den C-Header-Dateien deklariert sind.
- **User-defined functions:** die Funktionen, die vom C-Programmierer erstellt wurden, damit er/sie sie mehrmals verwenden kann. Es reduziert die Komplexität eines großen Programms und optimiert den Code.

In diesem Projekt definieren wir eine Funktion, um den Wert des Ultraschallmoduls zu lesen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

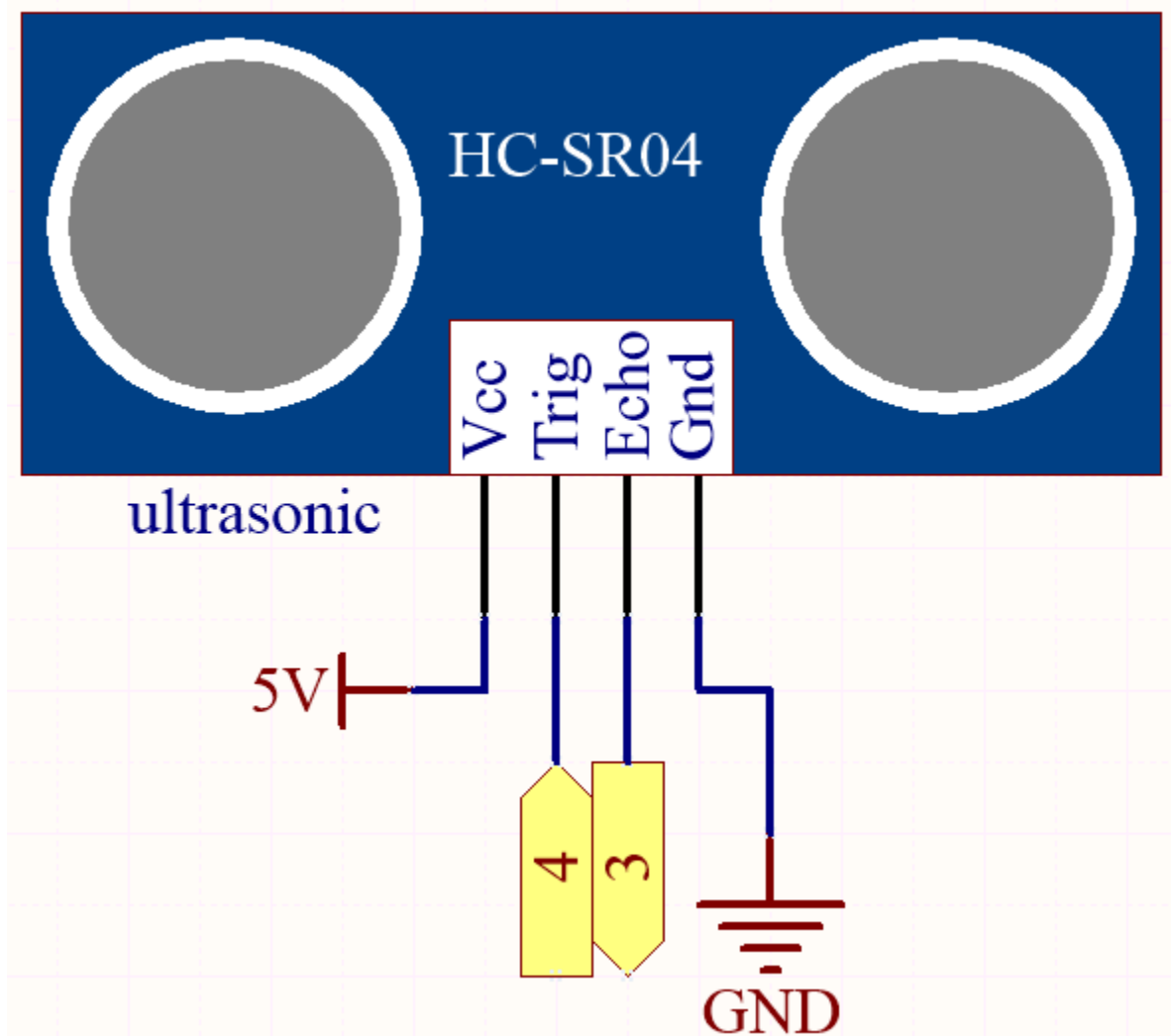
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

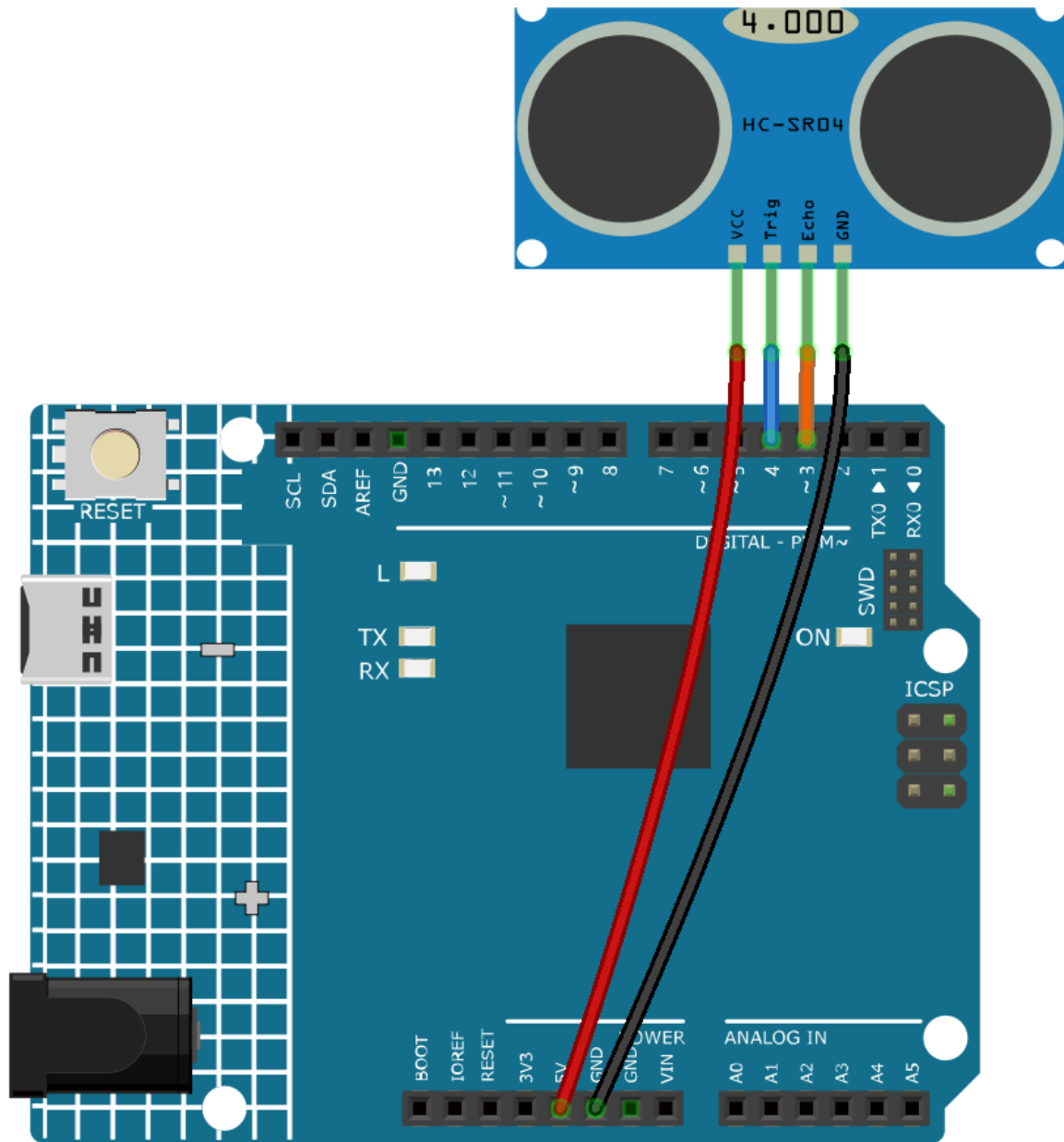
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Jumper-Kabel</i>	
<i>Ultraschall-Modul</i>	

Schaltplan



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei `5.8.user_function.ino` im Pfad `3in1-kit\learning_project\5.8.user_function`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem der Code erfolgreich hochgeladen wurde, wird der serielle Monitor den Abstand zwischen dem Ultraschallsensor und dem Hindernis vor ihm ausgeben.

Wie funktioniert das?

Zur Anwendung des Ultraschallsensors können wir direkt die Unterfunktion überprüfen.

```
float readSensorData(){// ...}
```

Der trigPin des Ultraschallmoduls sendet alle 2us ein 10us Quadratwellensignal.

```
digitalWrite(trigPin, LOW);  
delayMicroseconds(2);  
digitalWrite(trigPin, HIGH);  
delayMicroseconds(10);  
digitalWrite(trigPin, LOW);
```

Der echoPin empfängt ein High-Level-Signal, wenn sich ein Hindernis im Bereich befindet, und verwendet die pulseIn()-Funktion, um die Zeit vom Senden bis zum Empfangen aufzuzeichnen.

```
microsecond=pulseIn(echoPin, HIGH);
```

Die Schallgeschwindigkeit beträgt 340 m/s oder 29 Mikrosekunden pro Zentimeter.

Dies gibt die von der Quadratwelle zurückgelegte Entfernung an, hin und zurück, also teilen wir durch 2, um die Entfernung des Hindernisses zu erhalten.

```
float distance = microsecond / 29.00 / 2;
```

Beachten Sie, dass der Ultraschallsensor das Programm anhält, wenn er arbeitet, was zu Verzögerungen bei der Erstellung komplexer Projekte führen kann.

4.5.9 5.9 ShiftOut(LED)

shiftOut() lässt den 74HC595 8 digitale Signale ausgeben. Dabei wird das letzte Bit der Binärzahl an Q0 und das erste Bit an Q7 ausgegeben. Das bedeutet, die Binärzahl „00000001“ lässt Q0 ein hohes Signal ausgeben und Q1~Q7 ein niedriges Signal.

In diesem Projekt lernen Sie den 74HC595 kennen. Der 74HC595 besteht aus einem 8-Bit-Schieberegister und einem Speicherregister mit dreistufigen parallelen Ausgängen. Er wandelt serielle Eingaben in parallele Ausgaben um, sodass Sie IO-Ports eines MCU sparen können.

Spezifisch kann der 74HC595 8 Pins für den digitalen Signalausgang durch das Schreiben einer 8-Bit-Binärzahl ersetzen.

- [Binärzahl - Wikipedia](#)

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

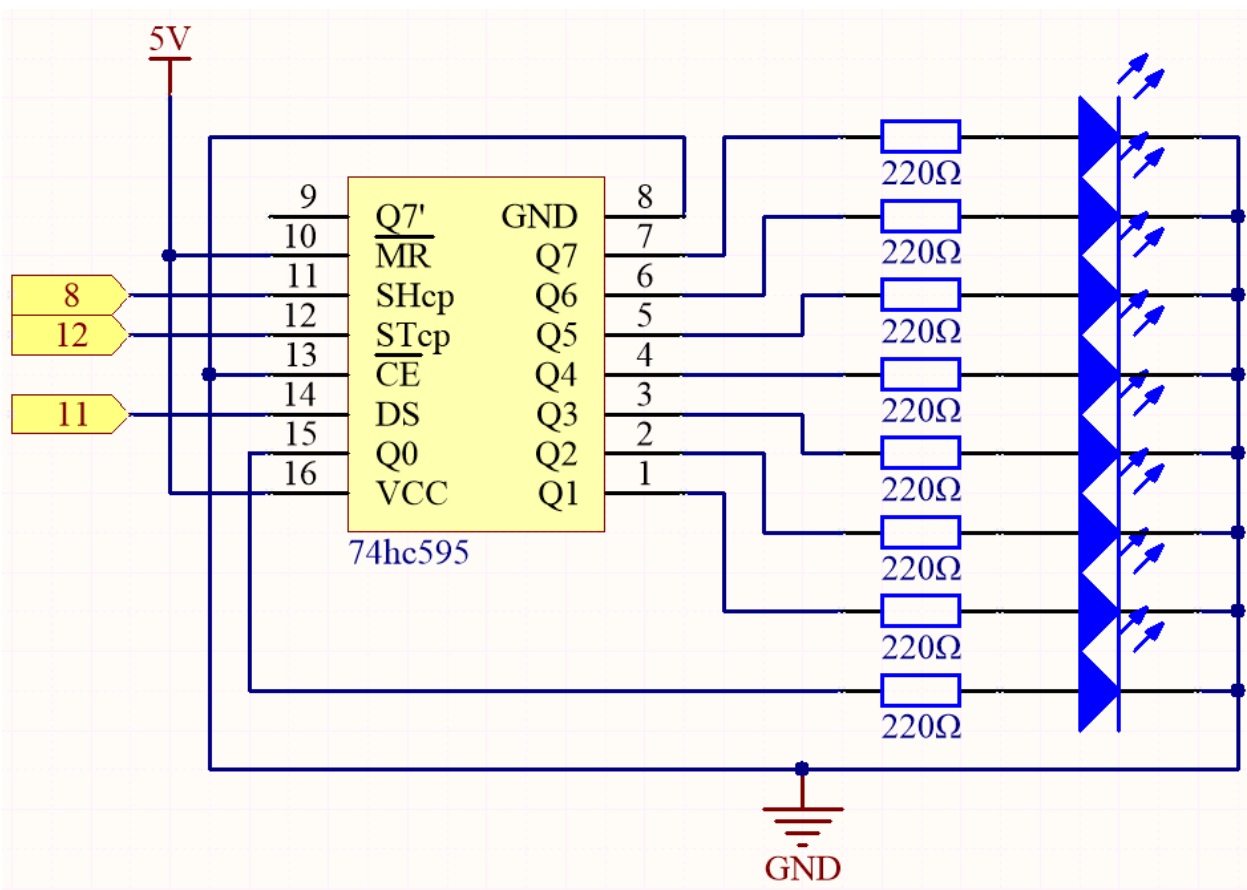
Es ist sicherlich praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>74HC595</i>	

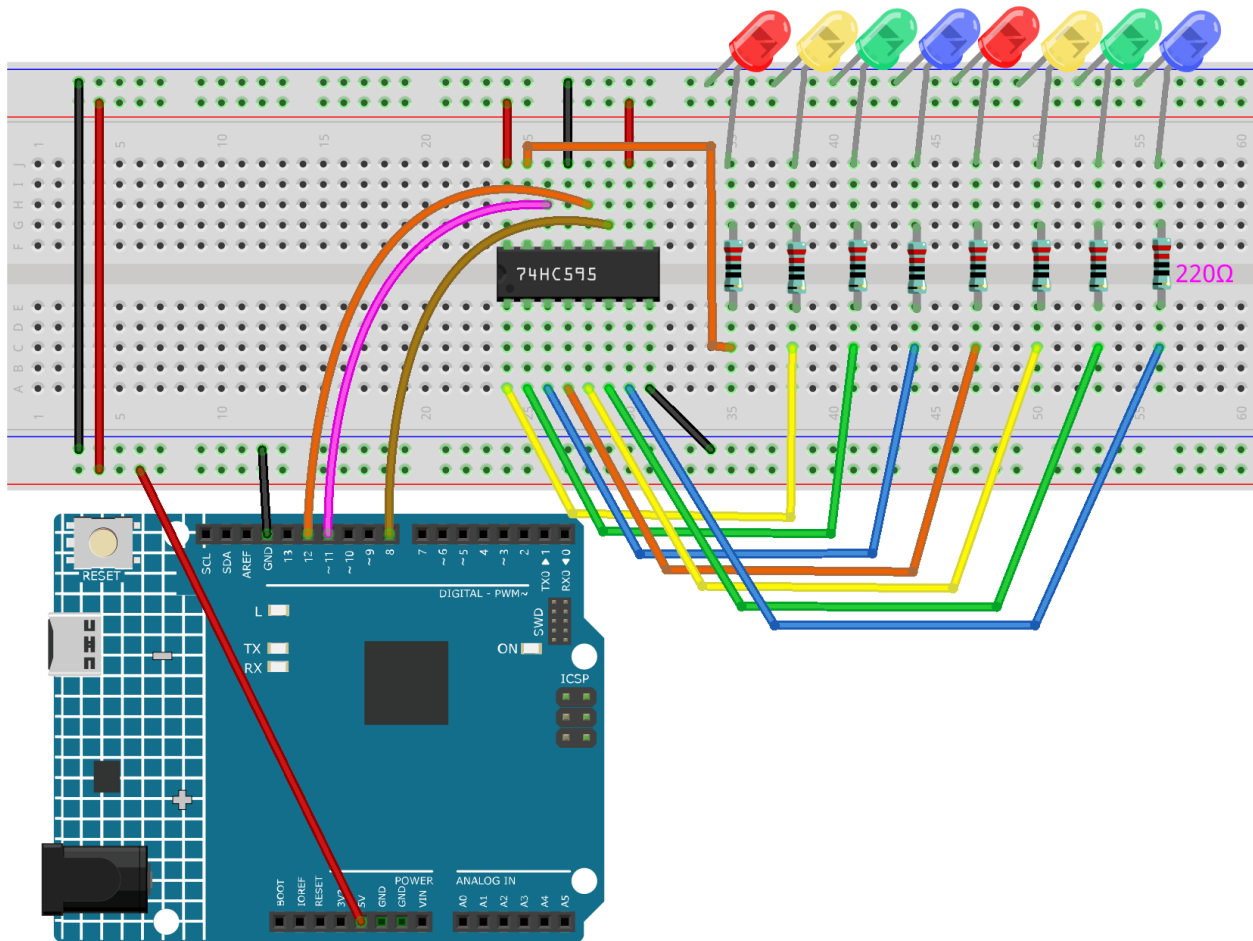
Schaltplan



- Wenn MR (Pin10) auf hohem Niveau und OE (Pin13) auf niedrigem Niveau ist, wird das Daten in der aufsteigenden Flanke von SHcp eingegeben und geht durch die aufsteigende Flanke von SHcp in das Speicherregister.
- Wenn die beiden Uhren miteinander verbunden sind, ist das Schieberegister immer einen Puls vor dem Speicherregister.
- Es gibt einen seriellen Shift-Eingangspin (Ds), einen seriellen Ausgangspin (Q) und einen asynchronen Reset-Button (niedriges Niveau) im Speicherregister.
- Das Speicherregister gibt einen Bus mit einem parallelen 8-Bit und in drei Zuständen aus.

- Wenn OE aktiviert (niedriges Niveau) ist, werden die Daten im Speicherregister an den Bus (Q0 ~ Q7) ausgegeben.

Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.9.shiftout_led.ino im Pfad 3in1-kit\learning_project\5.9.shiftout_led.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem Sie die Codes auf das R4-Board hochgeladen haben, sehen Sie, wie die LEDs nacheinander angehen.

Wie funktioniert das?

Deklarieren Sie ein Array und speichern Sie mehrere 8-Bit-Binärzahlen, die verwendet werden, um den Arbeitszustand der acht von 74HC595 gesteuerten LEDs zu ändern.

```
int dataArray[] = {B00000000, B00000001, B00000011, B00000111, B00001111, B00011111, B00111111, B01111111, B11111111};
```

Zuerst STcp auf niedriges Niveau setzen und dann auf hohes Niveau. Es wird eine aufsteigende Flankenimpuls von STcp erzeugt.

```
digitalWrite(STcp,LOW);
```

shiftOut() wird verwendet, um ein Byte Daten bitweise auszugeben, was bedeutet, ein Byte Daten in dataArray[num] an das Schieberegister mit dem DS-Pin zu senden. **MSBFIRST** bedeutet, von den hohen Bits zu verschieben.

```
shiftOut(DS,SHcp,MSBFIRST,dataArray[num]);
```

Nachdem digitalWrite(STcp,HIGH) ausgeführt wurde, befindet sich STcp an der aufsteigenden Flanke. Zu diesem Zeitpunkt werden die Daten im Schieberegister in das Speicherregister verschoben.

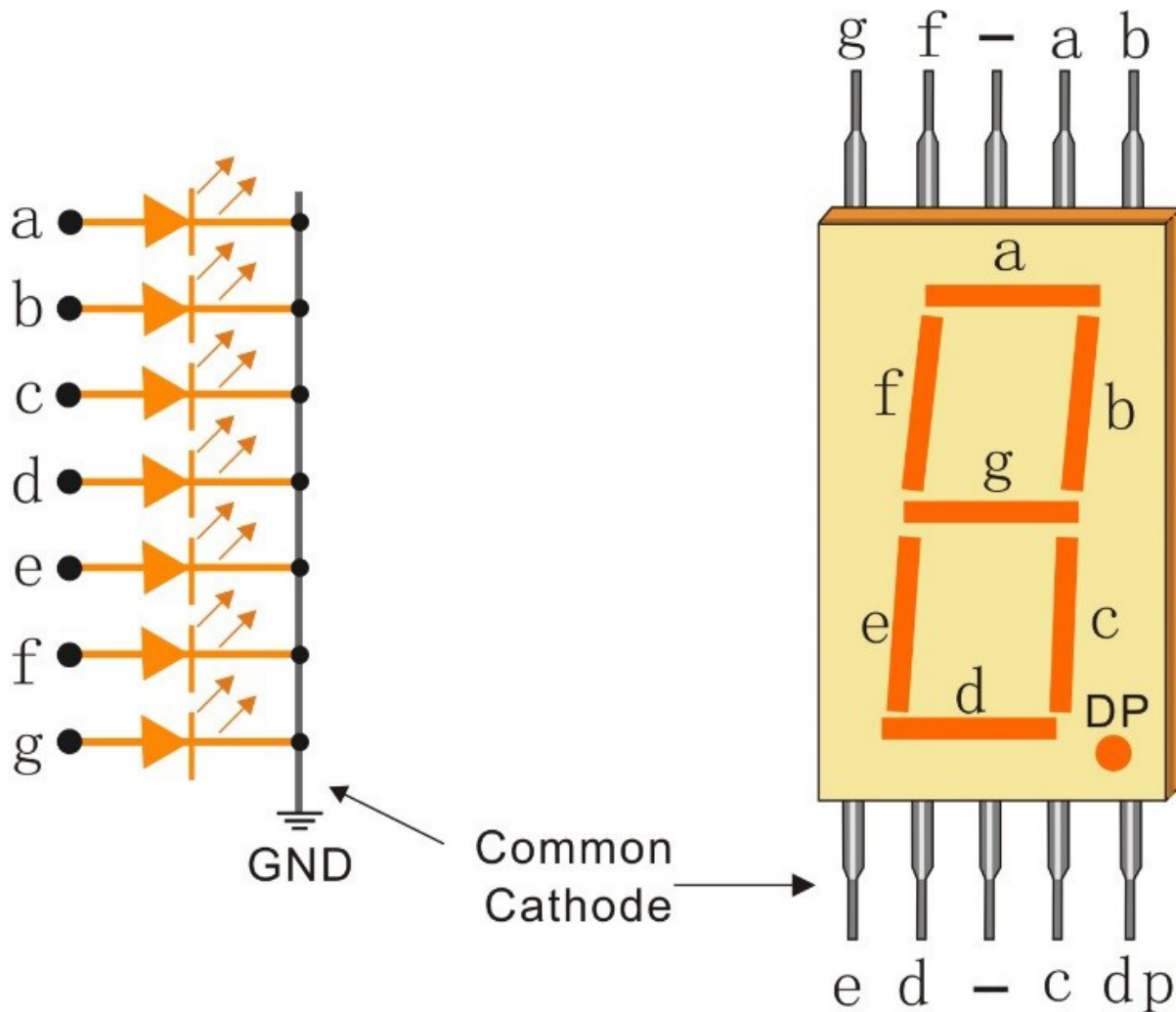
```
digitalWrite(STcp,HIGH);
```

Ein Byte Daten wird nach 8 Mal ins Speicherregister übertragen. Dann werden die Daten des Speicherregisters an den Bus (Q0-Q7) ausgegeben. Zum Beispiel wird shiftout B00000001 die von Q0 gesteuerte LED einschalten und die von Q1~Q7 gesteuerten LEDs ausschalten.

4.5.10 5.10 ShiftOut(Segmentanzeige)

Zuvor haben wir die Funktion shiftout() verwendet, um acht LEDs leuchten zu lassen. Hier nutzen wir sie, um 0-9 auf der 7-Segment-Anzeige darzustellen.

Die 7-Segment-Anzeige ist im Grunde ein Gerät, das aus 8 LEDs zusammengesetzt ist. Davon bilden 7 streifenförmige LEDs die Form einer „8“, und es gibt eine etwas kleinere punktförmige LED als Dezimalpunkt. Diese LEDs sind als a, b, c, d, e, f, g und dp markiert. Sie haben eigene Anodenpins und teilen sich Kathoden. Ihre Pin-Positionen sind in der untenstehenden Abbildung dargestellt.



Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

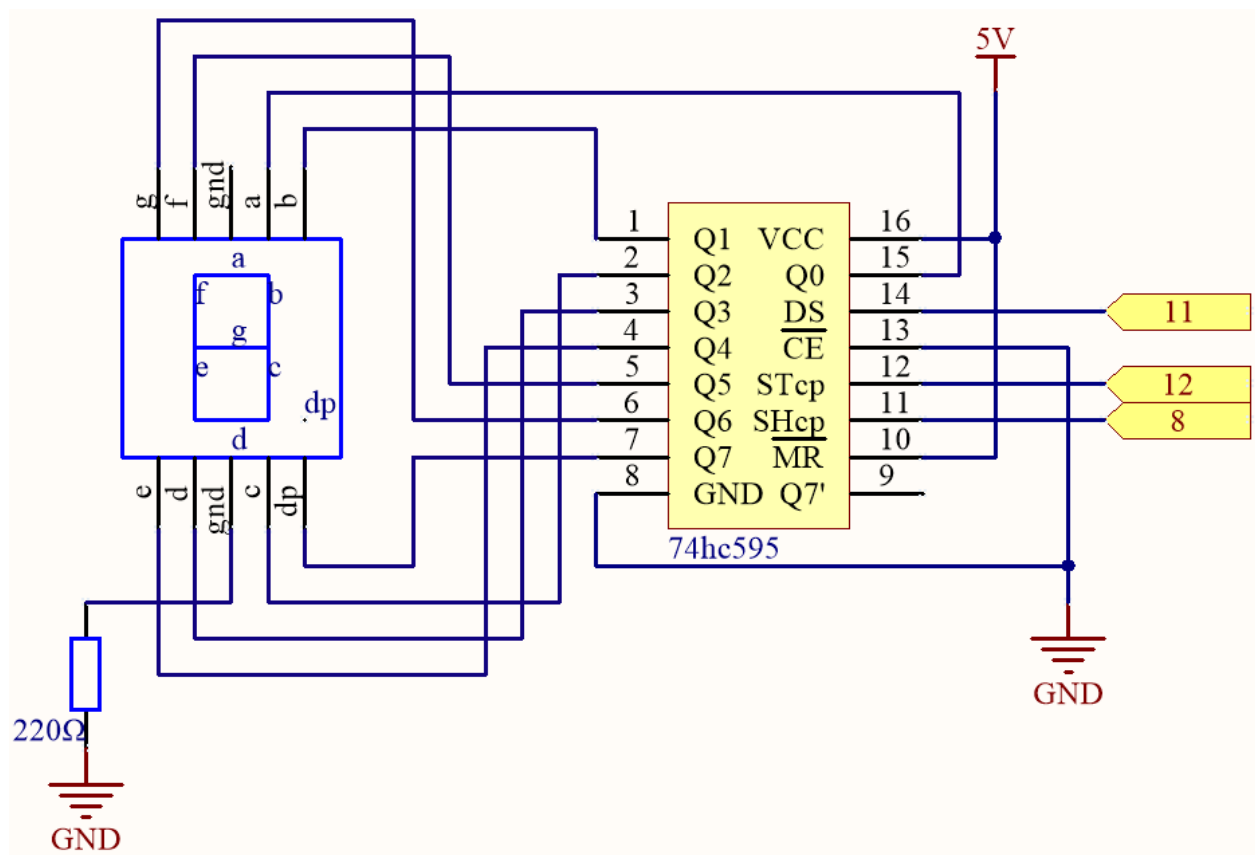
Es ist sicherlich praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 Minima	-
Breadboard	
Jumper-Kabel	
Widerstand	
7-Segment-Anzeige	
74HC595	

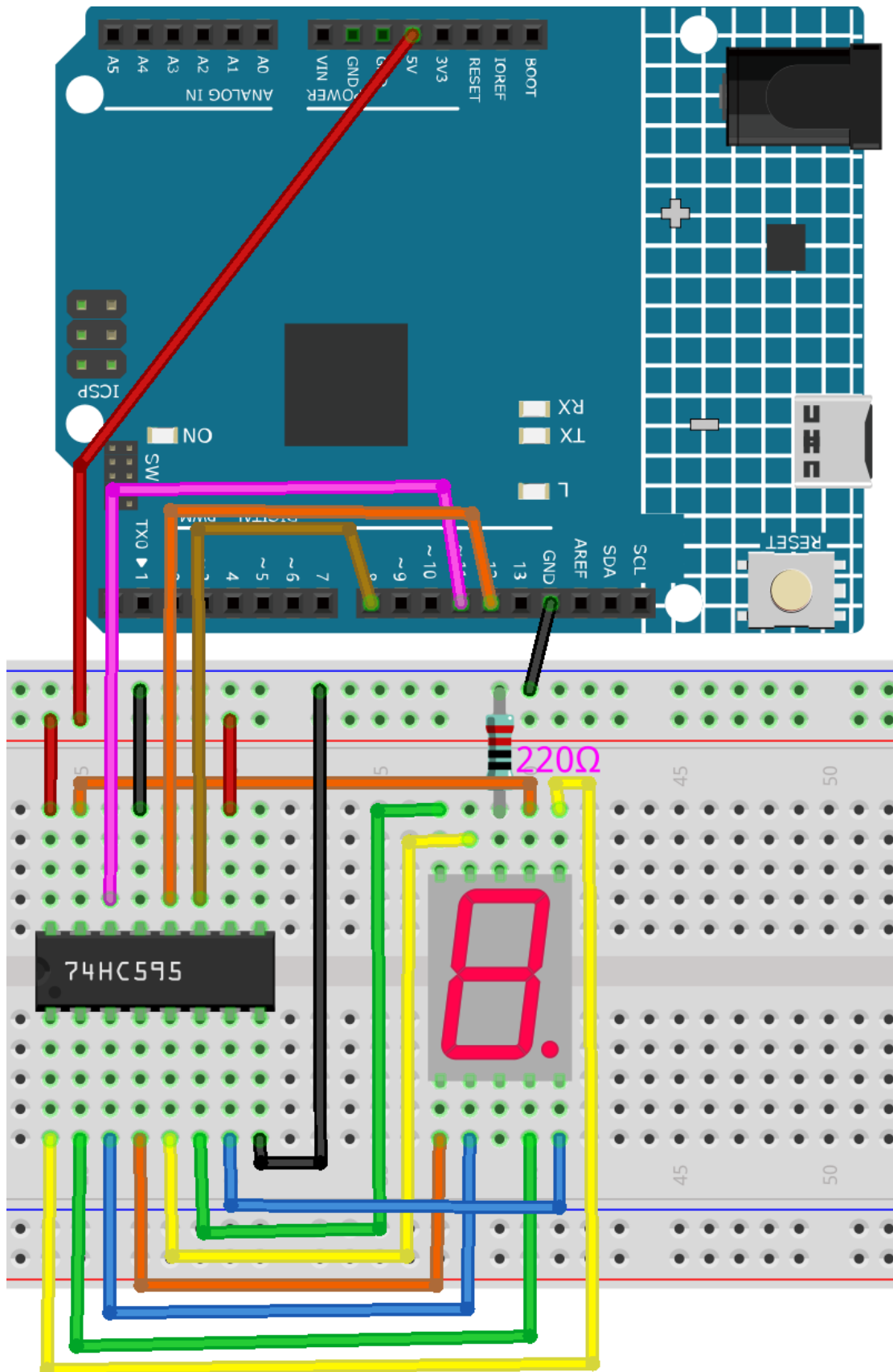
Schaltplan



Verdrahtung

Tab. 1: Verkabelung

74HC595	LED Segmentanzeige
Q0	a
Q1	b
Q2	c
Q3	d
Q4	e
Q5	f
Q6	g
Q7	dp



Code

Bemerkung:

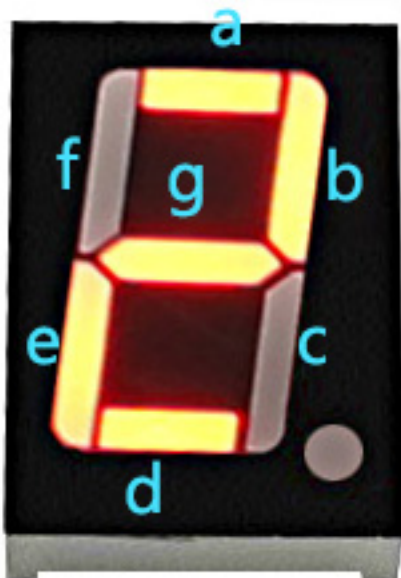
- Öffnen Sie die Datei `5.10.shiftout_segment.ino` im Pfad `3in1-kit\learning_project\5.10.shiftout_segment`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nach erfolgreichem Hochladen des Codes werden Sie sehen, dass die LED Segmentanzeige nacheinander die Zahlen 0~9 anzeigt.

Wie funktioniert das?

`shiftOut()` lässt den 74HC595 8 digitale Signale ausgeben. Er gibt das letzte Bit der Binärzahl an Q0 aus und den Output des ersten Bits an Q7. Das bedeutet, wenn man die Binärzahl „00000001“ schreibt, wird Q0 auf hohem Niveau ausgeben und Q1~Q7 auf niedrigem Niveau.

Angenommen, die 7-Segment-Anzeige zeigt die Zahl „2“, wir müssen ein hohes Niveau für a, b, d, e und g schreiben und ein niedriges Niveau für c, f und dp. Das bedeutet, dass die Binärzahl „01011011“ geschrieben werden muss. Zur besseren Lesbarkeit verwenden wir die hexadezimale Notation „0x5b“.



- [Hexadezimal](#)
- [BinärHex Konverter](#)

Ähnlich können wir die 7-Segment-Anzeige auch auf die gleiche Weise andere Zahlen anzeigen lassen. Die folgende Tabelle zeigt die Codes, die diesen Zahlen entsprechen.

Tab. 2: Zeichencode

Zahlen	Binärer Code	Hex Code
0	00111111	0x3f
1	00000110	0x06
2	01011011	0x5b
3	01001111	0x4f
4	01100110	0x66
5	01101101	0x6d
6	01111101	0x7d
7	00000111	0x07
8	01111111	0x7f
9	01101111	0x6f

Geben Sie diese Codes in `shiftOut()` ein, um die LED Segmentanzeige die entsprechenden Zahlen anzeigen zu lassen.

4.5.11 5.11 Externe Bibliotheken installieren

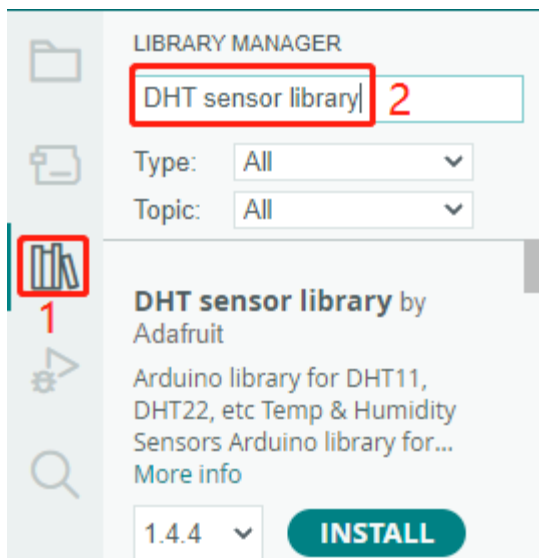
Eine Bibliothek ist eine Sammlung von vorgeschriebenem Code oder Funktionen, die die Fähigkeiten der Arduino IDE erweitern. Bibliotheken bieten gebrauchsfertigen Code für verschiedene Funktionen und ermöglichen es Ihnen, Zeit und Mühe beim Programmieren komplexer Features zu sparen.

Es gibt zwei Hauptwege, um Bibliotheken zu installieren:

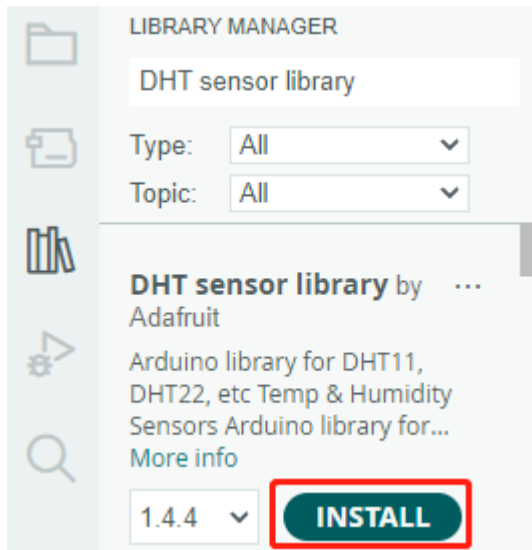
Viele Bibliotheken sind direkt über den Arduino **Library Manager** verfügbar. Sie können den **Library Manager** mit folgenden Schritten aufrufen:

1. Im **Library Manager** können Sie die gewünschte Bibliothek nach Namen suchen oder in verschiedenen Kategorien stöbern.

Bemerkung: Bei Projekten, bei denen eine Bibliotheksinstallation erforderlich ist, gibt es Aufforderungen, welche Bibliotheken zu installieren sind. Befolgen Sie die bereitgestellten Anweisungen, wie „Die DHT sensor library wird hier verwendet. Sie können sie aus dem **Library Manager** installieren.“ Installieren Sie einfach die empfohlenen Bibliotheken wie vorgeschlagen.



2. Sobald Sie die Bibliothek gefunden haben, die Sie installieren möchten, klicken Sie darauf und dann auf die Schaltfläche **Install**.



3. Die Arduino IDE wird die Bibliothek automatisch für Sie herunterladen und installieren.

Benötigte Komponenten

Unten sind die verwandten Komponenten aufgeführt, auf die Sie klicken können, um zu erfahren, wie sie verwendet werden.

5.11.1 Flüssigkristallanzeige

Ein I2C LCD1602 besteht aus einem LCD1602 und einem I2C-Modul. Mit dem LCD1602 können Zeichen, Zahlen usw. angezeigt werden, was jedoch viele Pins des Hauptsteuergeräts in Anspruch nimmt. Nachdem ein I2C-Modul konfiguriert wurde, sind nur 2 I/O-Pins erforderlich, um dieses LCD1602 zu steuern.

Sehen wir uns nun an, wie dieses I2C LCD1602 funktioniert.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

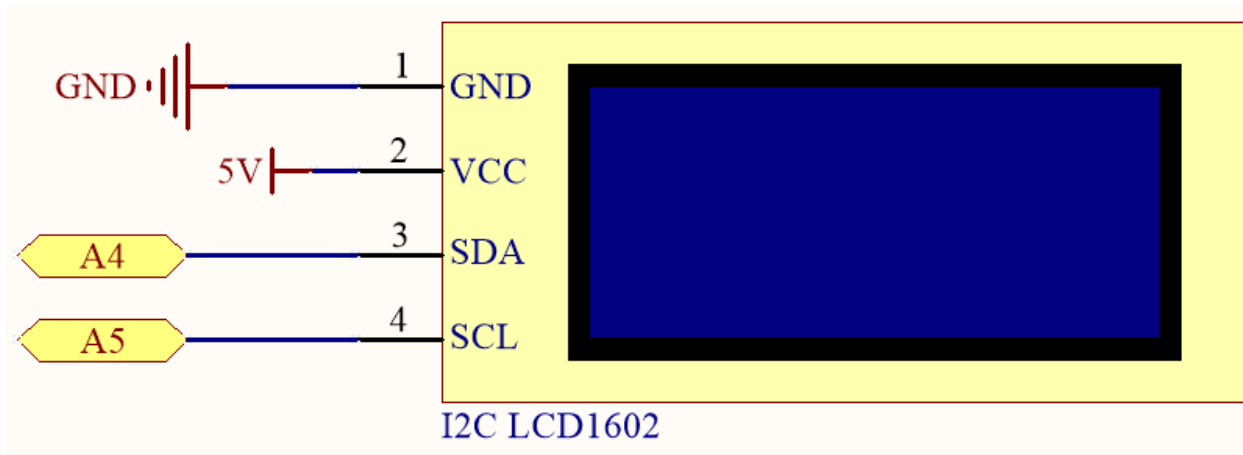
Es ist sicherlich praktisch, ein gesamtes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

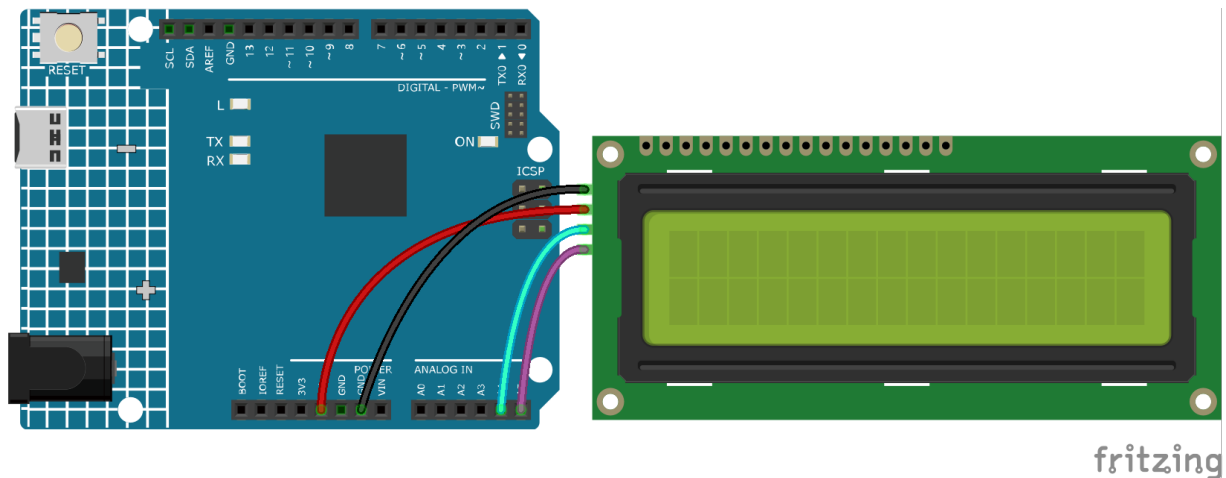
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 Minima	-
Jumper-Kabel	
I2C LCD1602	

Schaltplan



Verdrahtung

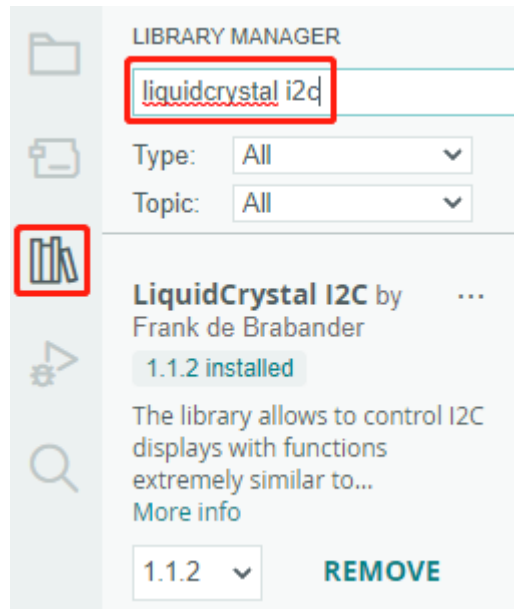


Bemerkung: Die SDA- und SCL-Pins des R4-Boards sind die Pins A4 und A5.

Code

Bemerkung:

- Öffnen Sie die Datei `5.11.liquid_crystal_display.ino` im Pfad `3in1-kit\learning_project\5.11.liquid_crystal_display`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Hier wird die `LiquidCrystal I2C`-Bibliothek verwendet. Sie können sie über den **Library Manager** installieren.



Nachdem der Code erfolgreich hochgeladen wurde, sehen Sie „SunFounder“ und „Hello World“ auf dem I2C LCD1602.

Bemerkung: Wenn der Code und die Verdrahtung korrekt sind, das LCD aber trotzdem keinen Inhalt anzeigt, können Sie das Potentiometer auf der Rückseite drehen.

Wie funktioniert das?

Mit dem Aufruf der Bibliothek `LiquidCrystal_I2C.h` können Sie das LCD einfach steuern.

```
#include "LiquidCrystal_I2C.h"
```

Bibliotheksfunktionen:

```
LiquidCrystal_I2C(uint8_t lcd_Addr, uint8_t lcd_cols, uint8_t lcd_rows)
```

Erstellt eine neue Instanz der Klasse `LiquidCrystal_I2C`, die ein bestimmtes LCD repräsentiert, das an Ihr Arduino-Board angeschlossen ist.

- `lcd_Addr`: Die Adresse des LCDs ist standardmäßig 0x27.
- `lcd_cols`: Das LCD1602 verfügt über 16 Spalten.
- `lcd_rows`: Das LCD1602 verfügt über 2 Zeilen.

```
void init()
```

Initialisiert das LCD.

```
void backlight()
```

Schaltet die (optionale) Hintergrundbeleuchtung ein.

```
void nobacklight()
```

Schaltet die (optionale) Hintergrundbeleuchtung aus.


```
void display()
```

Schaltet das LCD-Display ein.

```
void nodisplay()
```

Schaltet das LCD-Display schnell aus.

```
void clear()
```

Löscht die Anzeige und setzt die Cursor-Position auf Null.

```
void setCursor(uint8_t col, uint8_t row)
```

Setzt die Cursor-Position auf col,row.

```
void print(data, BASE)
```

Gibt Text auf dem LCD aus.

- **data:** Die auszudruckenden Daten (char, byte, int, long oder string).
- **BASE (optional):** Die Basis, in der Zahlen gedruckt werden sollen: BIN für Binär (Basis 2), DEC für Dezimal (Basis 10), OCT für Oktal (Basis 8), HEX für Hexadezimal (Basis 16).

5.11.2 IR-Empfänger

In diesem Projekt lernen Sie die Verwendung eines IR-Empfängers kennen.

Ein Infrarot-Empfänger ist eine Komponente, die Infrarotsignale empfängt und in der Lage ist, Infrarotstrahlen unabhängig zu empfangen und Signale auszugeben, die mit dem TTL-Niveau kompatibel sind. Er hat ungefähr die Größe eines gewöhnlichen Transistors in Plastikverpackung und eignet sich für alle Arten von Infrarot-Fernbedienungen und Infrarotübertragungen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

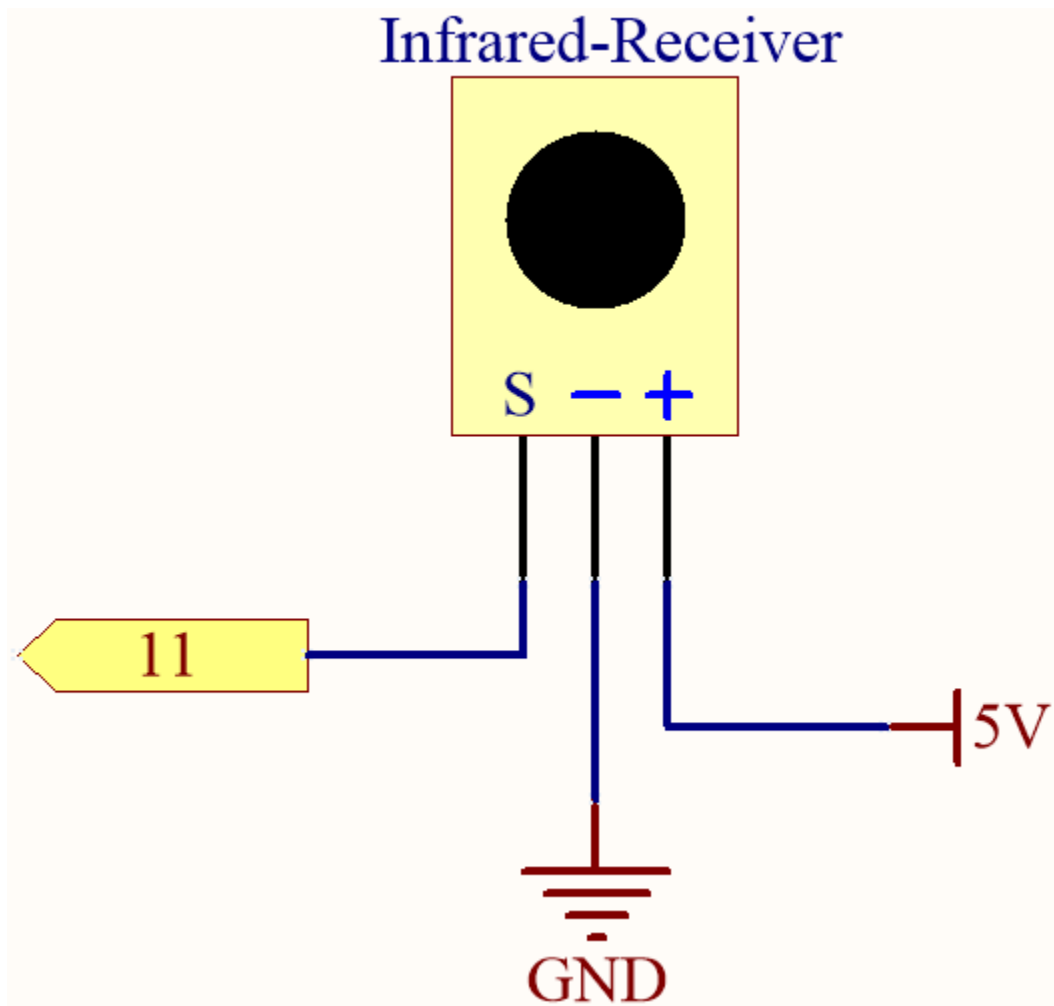
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können diese auch einzeln über die untenstehenden Links kaufen.

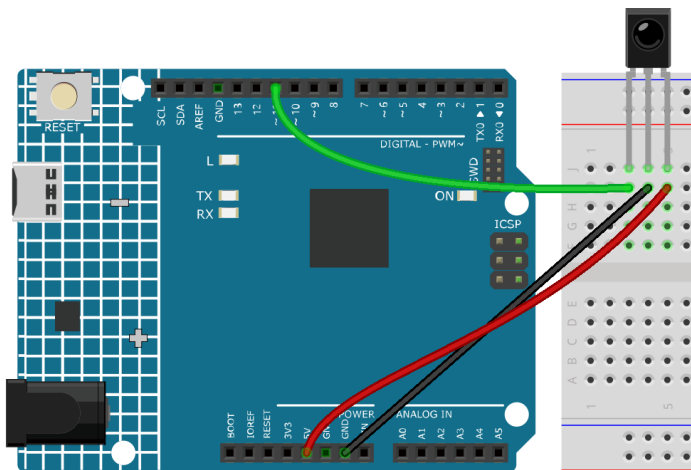
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>IR-Empfänger</i>	-

Schaltplan



Verdrahtung

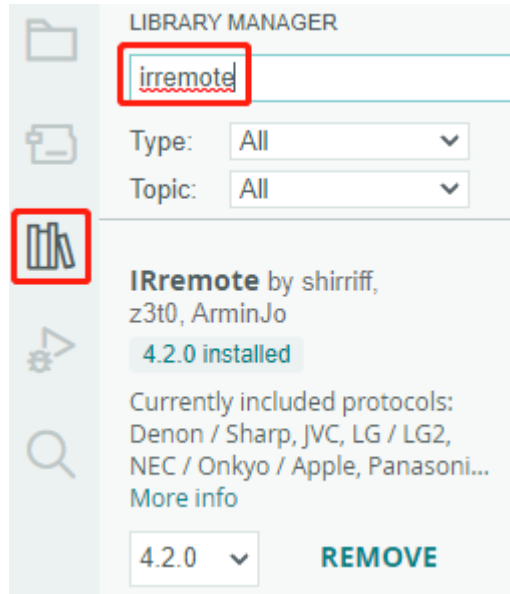
In diesem Beispiel verbinden wir den linken Pin des IR-Empfängers mit Pin 11, den mittleren Pin mit GND und den rechten Pin mit 5V.



Code

Bemerkung:

- Öffnen Sie die Datei 5.11.ir_receiver.ino im Pfad 3in1-kit\learning_project\5.11.ir_receiver.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Hier wird die Bibliothek IRremote verwendet. Sie können sie über den **Library Manager** installieren.



Nachdem Sie die Codes auf das R4-Board hochgeladen haben, können Sie sehen, dass der aktuelle Wert der gedrückten Taste der IR-Fernbedienung im Serienmonitor angezeigt wird.

Wie funktioniert das?

Dieser Code ist so konzipiert, dass er mit einer Infrarot-(IR-)Fernbedienung zusammen mit der IRremote-Bibliothek funktioniert. Hier die Erklärung:

1. Bibliotheken einbinden: Hier wird die IRremote-Bibliothek eingebunden, die Funktionen zur Arbeit mit IR-Fernbedienungen bietet.

```
#include <IRremote.h>
```

2. Definiert den Arduino-Pin, an den der Signalpin des IR-Sensors angeschlossen ist, und deklariert eine Variable, um den zuletzt dekodierten IR-Wert zu speichern.

```
const int IR_RECEIVE_PIN = 11; // Define the pin number for the IR Sensor
```

3. Initialisiert die serielle Kommunikation mit einer Baudrate von 9600. Startet den IR-Empfänger am angegebenen Pin (IR_RECEIVE_PIN) und aktiviert die LED-Rückmeldung (falls zutreffend).

```
void setup() {
  Serial.begin(9600); // Start serial
  // communication at 9600 baud rate
  IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK); // Start the IR
  // receiver
}
```

4. Die Schleife läuft kontinuierlich ab, um eingehende IR-Fernbedienungssignale zu verarbeiten.

```
void loop() {  
    if (IrReceiver.decode()) {  
        String decodedValue = decodeKeyValue(IrReceiver.decodedIRData.  
↪command);  
        if (decodedValue != "ERROR") {  
            Serial.println(decodedValue);  
            delay(100);  
        }  
        IrReceiver.resume(); // Enable receiving of the next value  
    }  
}
```

- Überprüft, ob ein IR-Signal empfangen und erfolgreich dekodiert wurde.
- Dekodiert den IR-Befehl und speichert ihn in decodedValue mit Hilfe einer benutzerdefinierten Funktion decodeKeyValue().
- Überprüft, ob der dekodierte Wert kein Fehler ist.
- Gibt den dekodierten IR-Wert auf dem seriellen Monitor aus.
- Setzt den IR-Signalempfang für das nächste Signal fort.

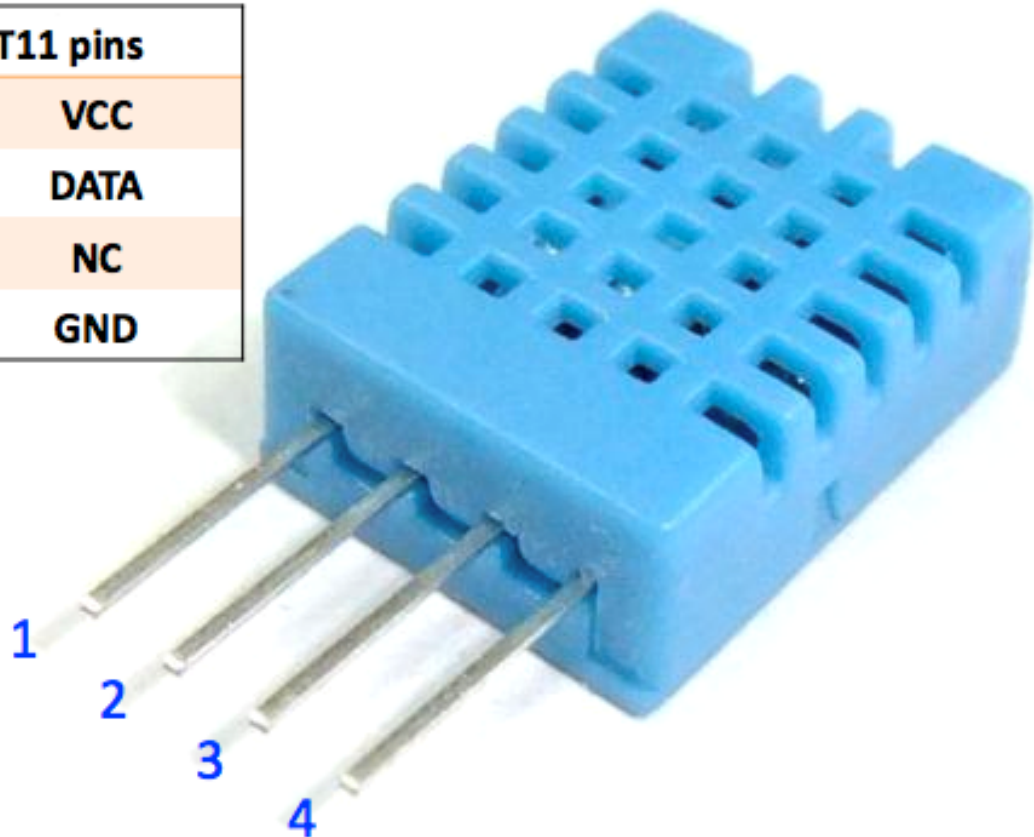
5.11.3 Temperatur - Luftfeuchtigkeit

Luftfeuchtigkeit und Temperatur sind sowohl in Bezug auf die physikalische Größe selbst als auch im tatsächlichen Leben der Menschen eng miteinander verbunden. Die Temperatur und Luftfeuchtigkeit der menschlichen Umgebung beeinflussen direkt die thermoregulatorische Funktion und den Wärmeübertragungseffekt des menschlichen Körpers. Dies hat wiederum Auswirkungen auf die Denkaktivität und den geistigen Zustand, wodurch die Effizienz unseres Lernens und Arbeitens beeinflusst wird.

Die Temperatur ist eine der sieben Grundgrößen im Internationalen Einheitensystem und dient zur Messung des Wärmezustands eines Objekts. Celsius ist eine der weltweit am häufigsten verwendeten Temperaturskalen und wird durch das Symbol „°C“ dargestellt.

Luftfeuchtigkeit bezeichnet die Konzentration von Wasserdampf in der Luft. Die relative Luftfeuchtigkeit, die im täglichen Leben häufig verwendet wird, wird in %RH angegeben. Sie hängt eng mit der Temperatur zusammen. Für ein bestimmtes Volumen abgeschlossenen Gases gilt: Je höher die Temperatur, desto niedriger die relative Luftfeuchtigkeit und umgekehrt.

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



Der dht11, ein digitaler Temperatur- und Luftfeuchtigkeitssensor, ist in diesem Kit enthalten. Er verwendet einen kapazitiven Luftfeuchtigkeitssensor und einen Thermistor, um die umgebende Luft zu messen und gibt ein digitales Signal am Datenpin aus.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

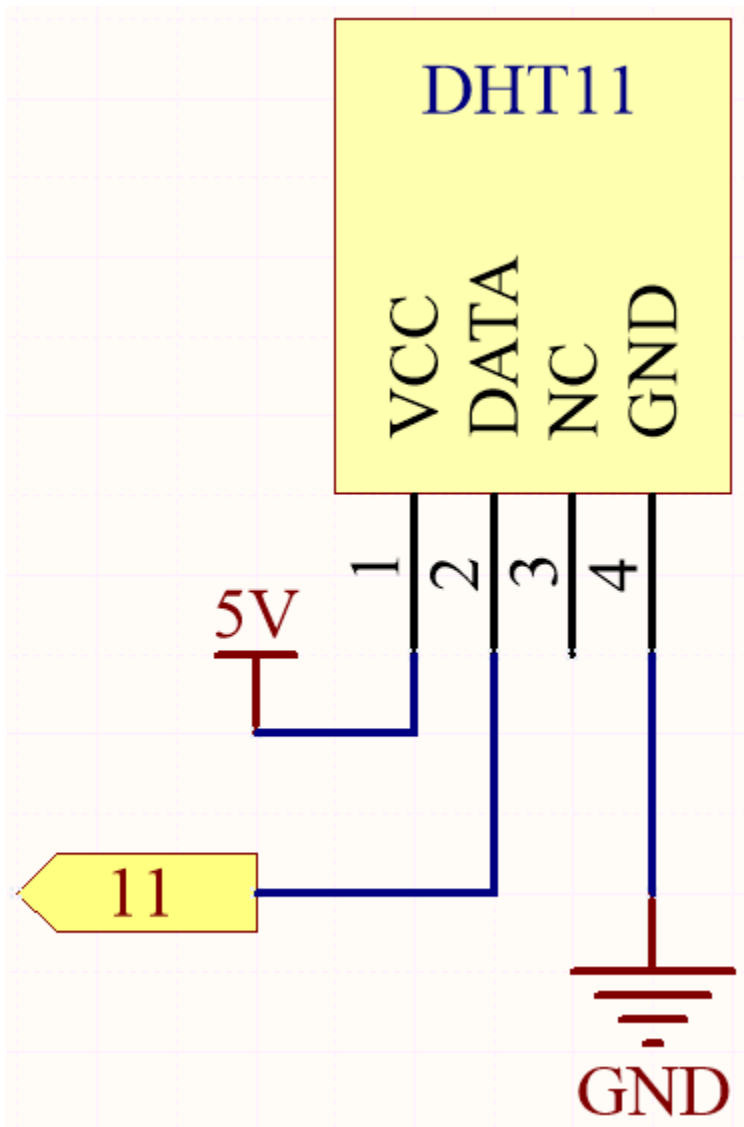
Es ist definitiv praktisch, ein komplettes Kit zu kaufen. Hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>DHT11 Feuchtigkeits- und Temperatursensor</i>	-

Schaltplan



Verdrahtung



Nach erfolgreichem Hochladen des Codes werden Sie sehen, dass der serielle Monitor kontinuierlich Temperatur und Luftfeuchtigkeit ausgibt. Mit der stabilen Ausführung des Programms werden diese beiden Werte immer genauer.

Wie funktioniert das?

1. Bindet die Bibliothek `DHT.h` ein, die Funktionen zur Interaktion mit den DHT-Sensoren bereitstellt. Danach wird der Pin und der Typ für den DHT-Sensor festgelegt.

```
#include "DHT.h"

#define DHTPIN 11 // Set the pin connected to the DHT11 data pin
#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);
```

2. Initialisiert die serielle Kommunikation mit einer Baudrate von 115200 und startet den DHT-Sensor.

```
void setup() {
  Serial.begin(115200);
  Serial.println("DHT11 test!");
  dht.begin();
}
```

3. In der Funktion `loop()` werden Temperatur- und Luftfeuchtigkeitswerte vom DHT11-Sensor gelesen und im seriellen Monitor angezeigt.

```
void loop() {
  // Wait a few seconds between measurements.
  delay(2000);

  // Reading temperature or humidity takes about 250 milliseconds!
  // Sensor readings may also be up to 2 seconds 'old' (it's a very slow
  // sensor)
  float humidity = dht.readHumidity();
  // Read temperature as Celsius (the default)
  float temperture = dht.readTemperature();

  // Check if any reads failed and exit early (to try again).
  if (isnan(humidity) || isnan(temperture)) {
    Serial.println("Failed to read from DHT sensor!");
    return;
  }
  // Print the humidity and temperature
  Serial.print("Humidity: ");
  Serial.print(humidity);
  Serial.print(" %\t");
  Serial.print("Temperature: ");
  Serial.print(temperture);
  Serial.println(" *C");
}
```

- Mit der Funktion `dht.readHumidity()` wird der Luftfeuchtigkeitswert vom DHT-Sensor gelesen.

- Mit der Funktion `dht.readTemperature()` wird der Temperaturwert vom DHT-Sensor gelesen.
- Die Funktion `isnan()` wird verwendet, um zu überprüfen, ob die Messungen gültig sind. Wenn entweder der Luftfeuchtigkeits- oder der Temperaturwert NaN (keine Zahl) ist, zeigt dies ein fehlerhaftes Lesen vom Sensor an, und eine Fehlermeldung wird ausgegeben.

4.5.12 5.12 Serielles Lesen

Vielleicht haben Sie dies bemerkt, wenn Sie die Funktion `Serial.print()` verwendet haben. Wo es Drucken gibt, gibt es auch Lesen? Wofür wird das Textfeld im Serial Monitor verwendet? Richtig geraten, es gibt Möglichkeiten, Programme und Schaltungen durch Eingabe von Informationen über das Textfeld im Serial Monitor zu steuern.

In diesem Projekt werden wir das I2C LCD1602 verwenden, um den im Serial Monitor eingegebenen Text anzuzeigen und die Verwendung von `Serial.read()` zu erleben.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

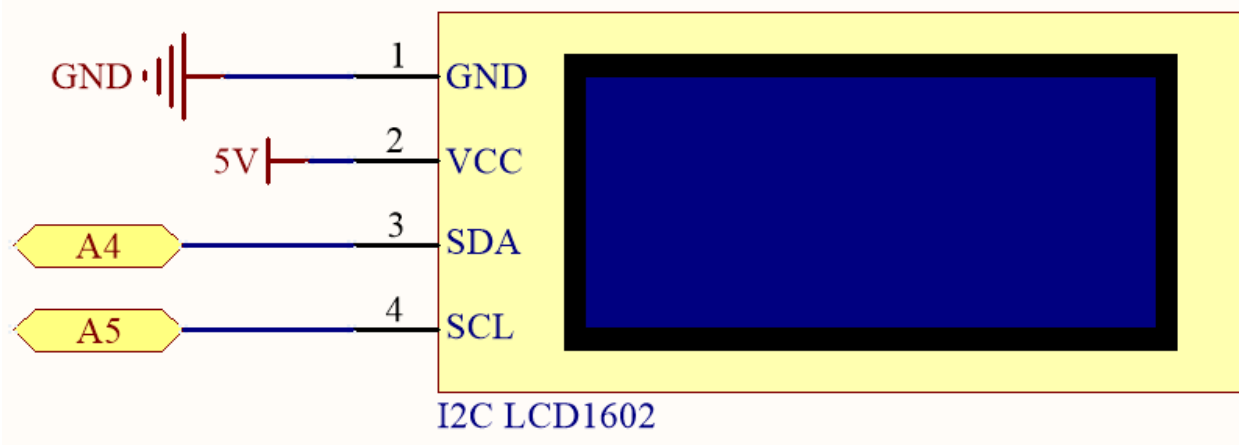
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

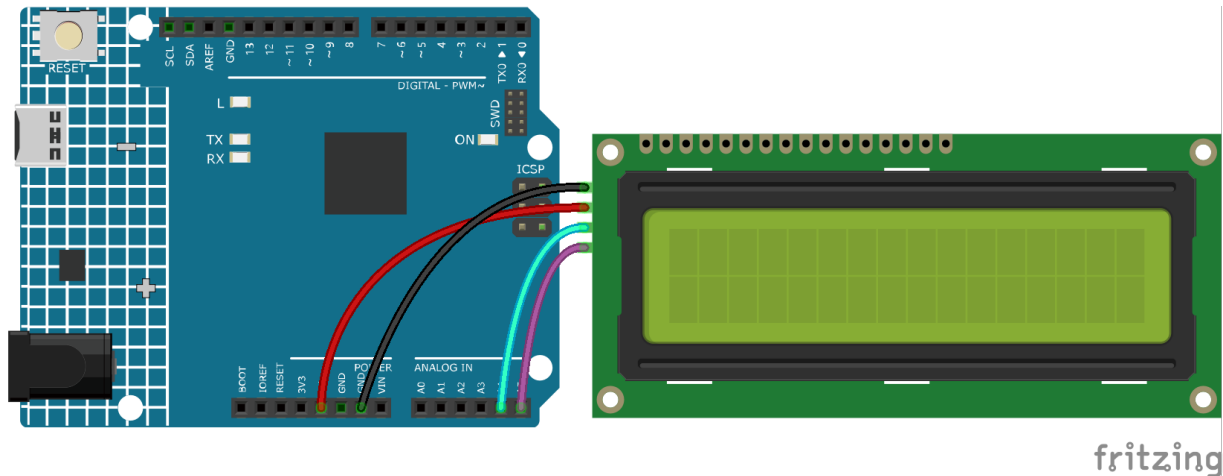
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Jumper-Kabel</i>	
<i>I2C LCD1602</i>	

Schaltplan



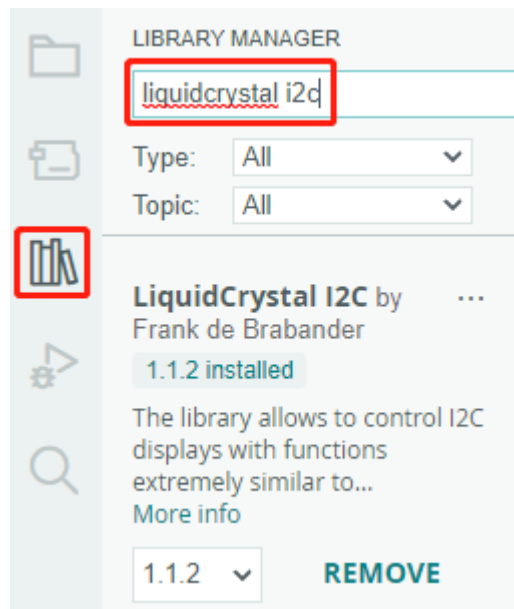
Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.12.serial_read.ino im Pfad 3in1-kit\learning_project\5.12.serial_read.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Die Bibliothek LiquidCrystal I2C wird hier verwendet. Sie können sie aus dem **Library Manager** installieren.



Nachdem der Code erfolgreich hochgeladen wurde, können Sie Text in das Textfeld des Serial Monitors eingeben, und das LCD wird die Informationen anzeigen.

Wie funktioniert das?

```
void loop()
{
  // when characters arrive over the serial port...
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

if (Serial.available()) {
    // wait a bit for the entire message to arrive
    delay(100);
    // clear the screen
    lcd.clear();
    // read all the available characters
    while (Serial.available() > 0) {
        // display each character to the LCD
        lcd.write(Serial.read());
    }
}
}

```

- Mit `Serial.available()` können Sie die Anzahl der verfügbaren Zeichen im eingehenden Stream abrufen, wenn Sie etwas aus dem Textfeld eingeben. Da es zwei Terminator in der Eingabe gibt, haben Sie tatsächlich 3 Zeichen, wenn Sie A eingeben, und 4 Zeichen, wenn Sie AB eingeben.
- `Serial.read()` nimmt das erste Zeichen aus dem eingehenden Stream. Wenn Sie beispielsweise AB eingeben haben und `Serial.read()` nur einmal aufrufen, erhalten Sie das Zeichen A; beim zweiten Aufruf erhalten Sie B; beim dritten und vierten Aufruf erhalten Sie zwei Endsymbole. Wenn dieser Funktion aufgerufen wird und im Eingabestream keine Zeichen verfügbar sind, führt dies zu einem Fehler.

Zusammenfassend wird häufig kombiniert, die beiden oben genannten, indem eine `while` Schleife verwendet wird, um alle jedes Mal eingegebenen Zeichen zu lesen.

```

while (Serial.available() > 0) {
    Serial.print(Serial.read());
}

```

Übrigens, wenn Sie `Serial.read()` nicht verwenden, um Zeichen aus dem eingehenden Stream zu holen, werden die Zeichen des eingehenden Streams zusammengefügt. Zum Beispiel, wenn Sie A gefolgt von AB eingeben, wird der eingehende Stream 7 Zeichen ansammeln.

4.5.13 5.13 Unterbrechung

Wenn Sie in einem Projekt, das Sensoren verwendet, einige `delay()`-Befehle verwenden, werden Sie vielleicht feststellen, dass das Programm nicht reagiert, wenn diese Sensoren ausgelöst werden. Dies liegt daran, dass die Delay-Anweisung das Programm anhält und es nicht in der Lage ist, das Signal zu empfangen, das der Sensor an die Hauptsteuerplatine sendet.

In diesem Fall kann eine Unterbrechung (Interrupt) verwendet werden. Ein Interrupt stellt sicher, dass das Programm keinen Impuls verpasst.

In diesem Kapitel verwenden wir den aktiven Summer und Tasten, um den Vorgang der Verwendung von Interrupts zu erleben.

In der `loop()`-Funktion wird `delay(1000)` verwendet, um Sekunden zu zählen. Legen Sie die Taste, die den Summer steuert, in die ISR, sodass sie nicht durch die Verzögerung gestört wird und die Aufgabe reibungslos abgeschlossen werden kann.

Bemerkung: ISRs sind spezielle Funktionen, die einige einzigartige Einschränkungen haben, die die meisten anderen Funktionen nicht haben. Eine ISR kann keine Parameter haben und sollte nichts zurückgeben. Grundsätzlich sollte eine ISR so kurz und schnell wie möglich sein. Wenn Ihr Sketch mehrere ISRs verwendet, kann immer nur eine gleichzeitig

ausgeführt werden. Andere Unterbrechungen werden nach Abschluss der aktuellen in einer Reihenfolge ausgeführt, die von ihrer Priorität abhängt.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

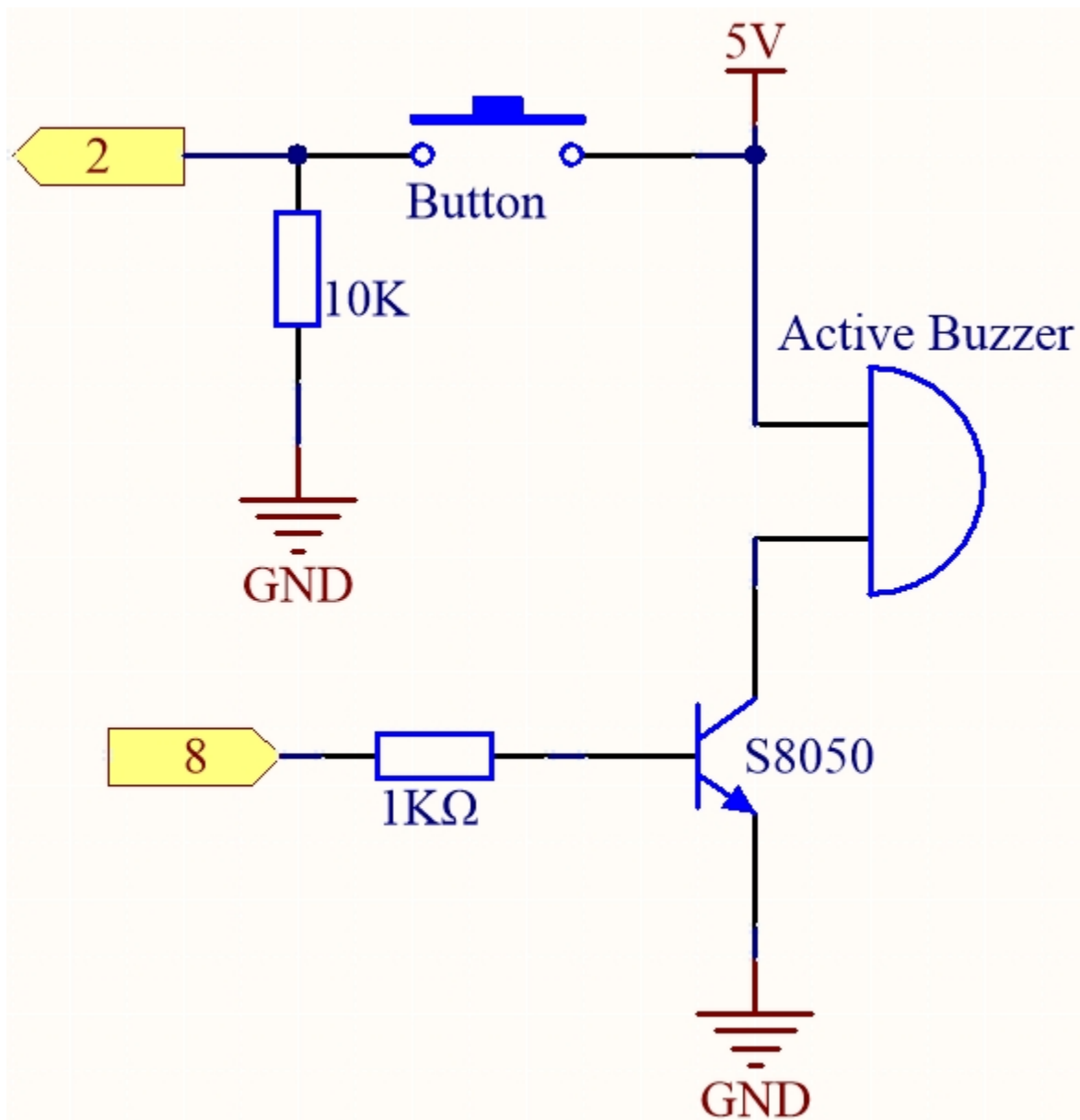
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

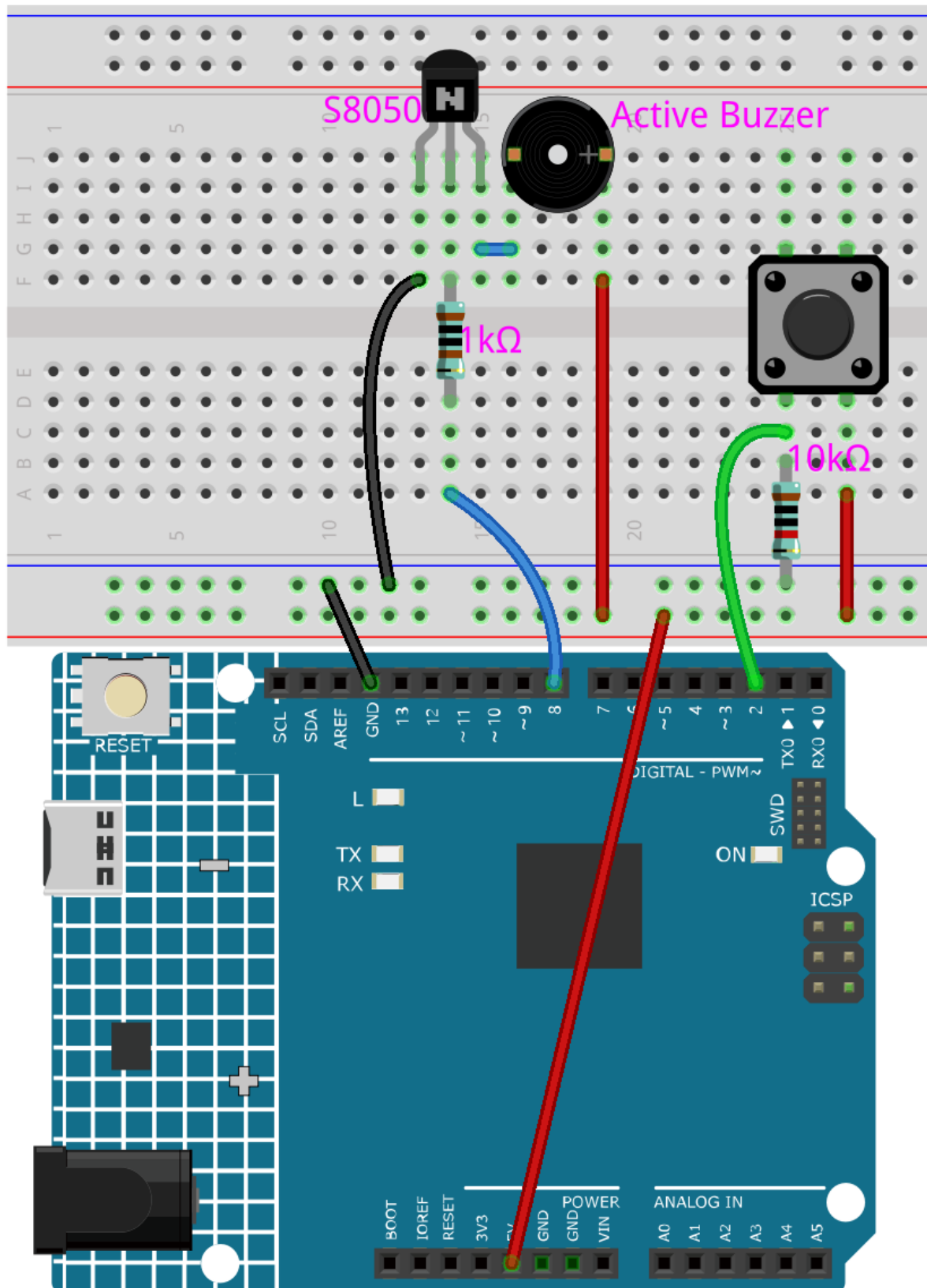
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Knopf</i>	
<i>Summer</i>	-

Schaltplan



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei `5.13.interrupt.ino` unter dem Pfad `3in1-kit\learning_project\5.13.interrupt`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nachdem der Code erfolgreich hochgeladen wurde, öffnen Sie den Serial Monitor und Sie werden sehen, dass jede Sekunde eine automatisch erhöhende Nummer ausgegeben wird. Wenn Sie den Knopf drücken, wird der Summer ertönen. Die von der Taste gesteuerte Summerfunktion und die Zeitfunktion stehen nicht in Konflikt miteinander.

Wie funktioniert das?

- `attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)`: Fügt einen Unterbrecher hinzu.

Syntax

```
attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)
```

Parameter

- `pin`: Die Arduino-Pin-Nummer. Sie sollten `digitalPinToInterrupt(pin)` verwenden, um den tatsächlichen Digital-Pin in eine spezifische Unterbrecher-Nummer zu konvertieren. Zum Beispiel, wenn Sie an Pin 3 anschließen, verwenden Sie sein `digitalPinToInterrupt(3)` als ersten Parameter.
- `ISR`: Die ISR, die aufgerufen wird, wenn der Unterbrecher auftritt; diese Funktion darf keine Parameter annehmen und nichts zurückgeben. Diese Funktion wird manchmal als Interrupt-Service-Routine bezeichnet.
- `mode`: definiert, wann der Unterbrecher ausgelöst werden sollte. Vier Konstanten sind als gültige Werte vordefiniert:
 - * `LOW`, um den Unterbrecher auszulösen, wenn der Pin niedrig ist,
 - * `CHANGE`, um den Unterbrecher auszulösen, wenn sich der Pin-Wert ändert.
 - * `RISING`, um auszulösen, wenn der Pin von niedrig auf hoch wechselt.
 - * `FALLING`, um auszulösen, wenn der Pin von hoch auf niedrig wechselt.

Bemerkung: Verschiedene Hauptsteuerplatinen können Unterbrecherpins unterschiedlich verwenden. Auf Ihrer Platine können nur Pin 2 und Pin 3 Unterbrechungen nutzen.

4.5.14 5.14 Kalibrierung

Wenn Sie analoge Eingangskomponenten verwenden, wie Fotowiderstände, Bodenfeuchtigkeitssensoren usw., stellen Sie möglicherweise fest, dass deren Messbereich nicht 0 bis 1023 beträgt, sondern eher ein Bereich wie 0 bis 800 oder 600 bis 1000. Denn unter normalen Bedingungen ist es unmöglich, die Grenzen dieser Geräte zu erreichen.

In diesem Fall kann eine Technik zur Kalibrierung der Sensoreingänge verwendet werden. Beim Start sollte das Steuerboard den Sensor fünf Sekunden lang messen und die höchsten und niedrigsten Werte aufzeichnen. Diese fünfsekündige Messung definiert die minimalen und maximalen erwarteten Werte der Messungen im Zyklus.

In diesem Projekt verwenden wir einen Fotowiderstand und einen passiven Summer, um ein Spiel zu implementieren, das einem [theremin](#) ähnelt, unter Verwendung der oben beschriebenen Kalibrierungstechnik.

Bemerkung: Das *theremin* ist ein elektronisches Musikinstrument, das keinen physischen Kontakt erfordert. Es erzeugt verschiedene Töne, indem es die Position der Hände des Spielers erkennt.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

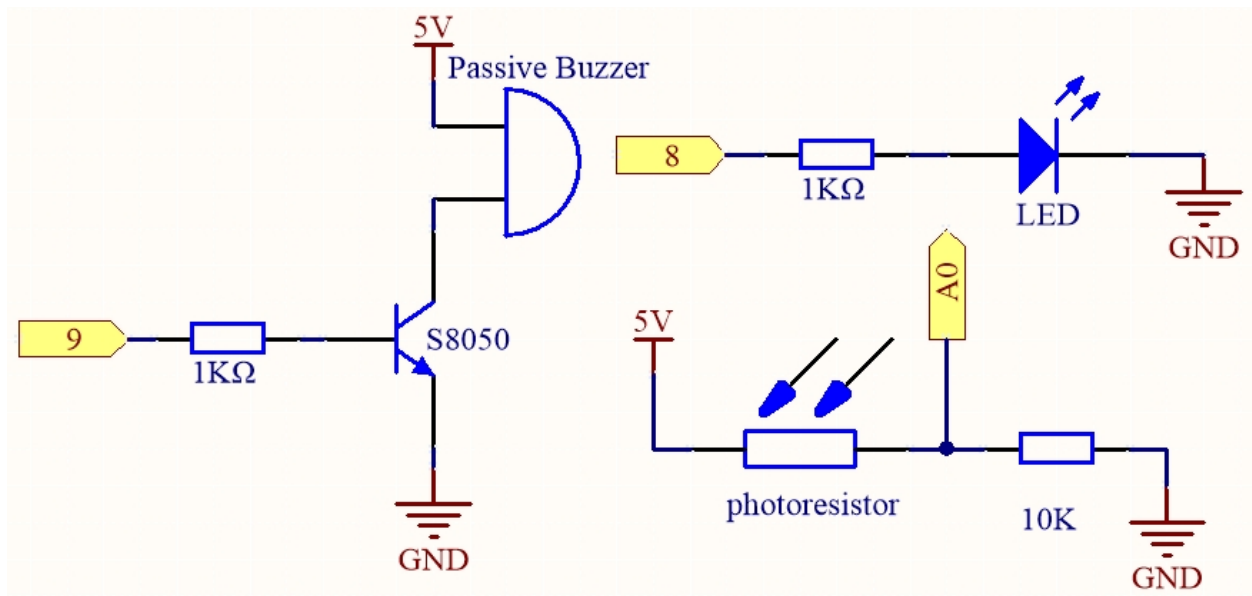
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

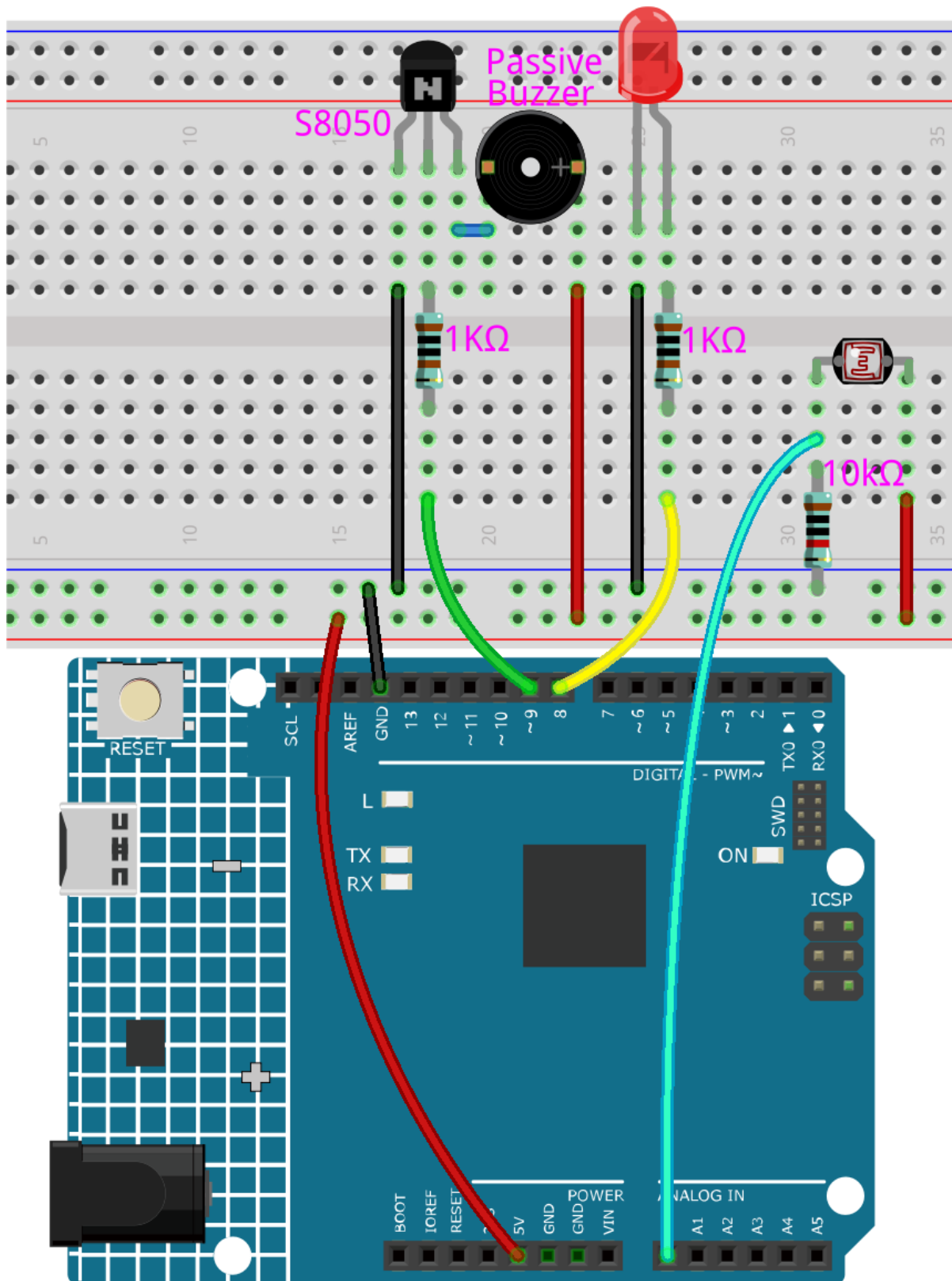
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Summer</i>	
<i>LED</i>	
<i>Fotowiderstand</i>	

Schaltplan



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.14.calibration.ino unter dem Pfad 3in1-kit\learning_project\5.14.calibration.
 - Oder kopieren Sie diesen Code in die **Arduino IDE**.
-

Nachdem der Code erfolgreich hochgeladen wurde, leuchtet die LED auf und wir haben 5 Sekunden Zeit, um den Erfassungsbereich des Fotowiderstands zu kalibrieren. Dies liegt daran, dass wir uns jedes Mal in einer anderen Lichtumgebung befinden könnten, wenn wir ihn verwenden (z.B. unterscheidet sich die Lichtintensität zwischen Mittag und Dämmerung).

Zu diesem Zeitpunkt müssen wir unsere Hände über dem Fotowiderstand auf und ab schwingen, und der Bewegungsbereich der Hand wird auf den Spielbereich dieses Instruments kalibriert.

Nach 5 Sekunden erlischt die LED und wir können unsere Hände über den Fotowiderstand schwingen, um zu spielen.

Wie funktioniert das?

1. Initialwerte und Pins aller Komponenten festlegen.

```
const int buzzerPin = 9;
const int ledPin = 8;
const int photocellPin = A0; //photoresistor attach to A2

int lightLow = 1023;
int lightHigh = 0;

int sensorValue = 0; // value read from the sensor
int pitch = 0; // sensor value converted into LED 'bars'

unsigned long previousMillis = 0;
const long interval = 5000;
```

2. Ein Kalibrierungsprozess in setup() einrichten.

```
void setup()
{
  pinMode(buzzerPin, OUTPUT); // make buzzer output
  pinMode(ledPin, OUTPUT); // make the LED pin output

  /* calibrate the photoresistor max & min values */
  previousMillis = millis();
  digitalWrite(ledPin, HIGH);
  while (millis() - previousMillis <= interval) {
    sensorValue = analogRead(photocellPin);
    if (sensorValue > lightHigh) {
      lightHigh = sensorValue;
    }
    if (sensorValue < lightLow) {
      lightLow = sensorValue;
    }
  }
  digitalWrite(ledPin, LOW);
}
```

Der Ablauf ist wie folgt:

- Verwendung von `millis()` zur Zeitmessung mit einem Intervall von 5000ms.

```
previousMillis = millis();
...
while (millis() - previousMillis <= interval) {
...
}
```

- Während dieser fünf Sekunden wird mit der Hand um den Fotowiderstand gewedelt, die maximalen und minimalen Werte des erkannten Lichts werden erfasst und den Variablen `lightHigh` und `lightLow` zugewiesen.

```
sensorValue = analogRead(photocellPin);
if (sensorValue > lightHigh) {
    lightHigh = sensorValue;
}
if (sensorValue < lightLow) {
    lightLow = sensorValue;
}
```

3. Nun können Sie dieses Theremin spielen. Lesen Sie den Wert des Fotowiderstands in `sensorValue` und übertragen Sie ihn aus dem kleinen Bereich in den großen Bereich, um ihn als Frequenz des Summers zu verwenden.

```
void loop()
{
    /* play*/
    sensorValue = analogRead(photocellPin); //read the value of A0
    pitch = map(sensorValue, lightLow, lightHigh, 50, 6000); // map to the
    ↪ buzzer frequency
    if (pitch > 50) {
        tone(buzzerPin, pitch, 20);
    }
    delay(10);
}
```

4.5.15 5.15 EEPROM

EEPROM ist ein Speicher, sodass die darin gespeicherten Daten nicht gelöscht werden, wenn das Hauptsteuerbrett ausgeschaltet wird. Sie können es verwenden, um einige Daten zu speichern und diese beim nächsten Einschalten wieder auszulesen.

Als Beispiel könnten Sie einen Sportzähler erstellen, der aufzeichnet, wie viele Seilsprünge Sie täglich machen.

Sie können auch Daten in einem Programm schreiben und sie in einem anderen auslesen. Wenn Sie zum Beispiel an einem Auto-Projekt arbeiten und die Geschwindigkeiten der beiden Motoren nicht übereinstimmen, können Sie ein Kalibrierungsprogramm schreiben, um den Ausgleichswert für die Motorgeschwindigkeit zu speichern.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

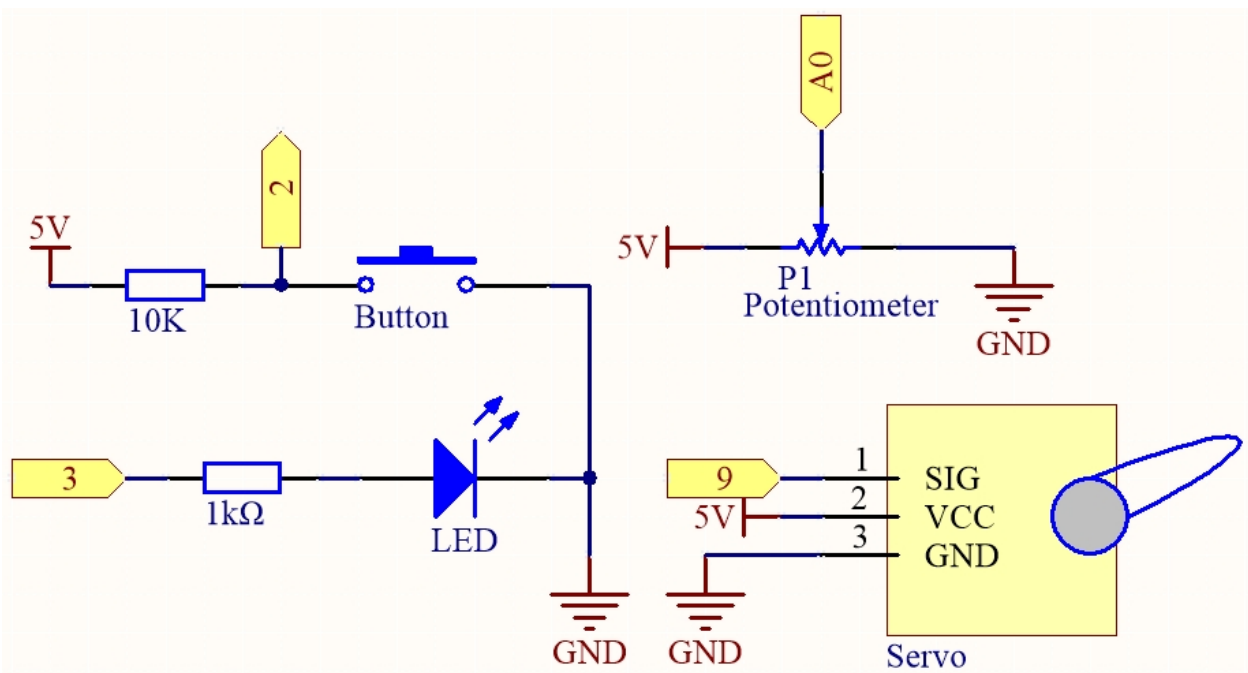
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

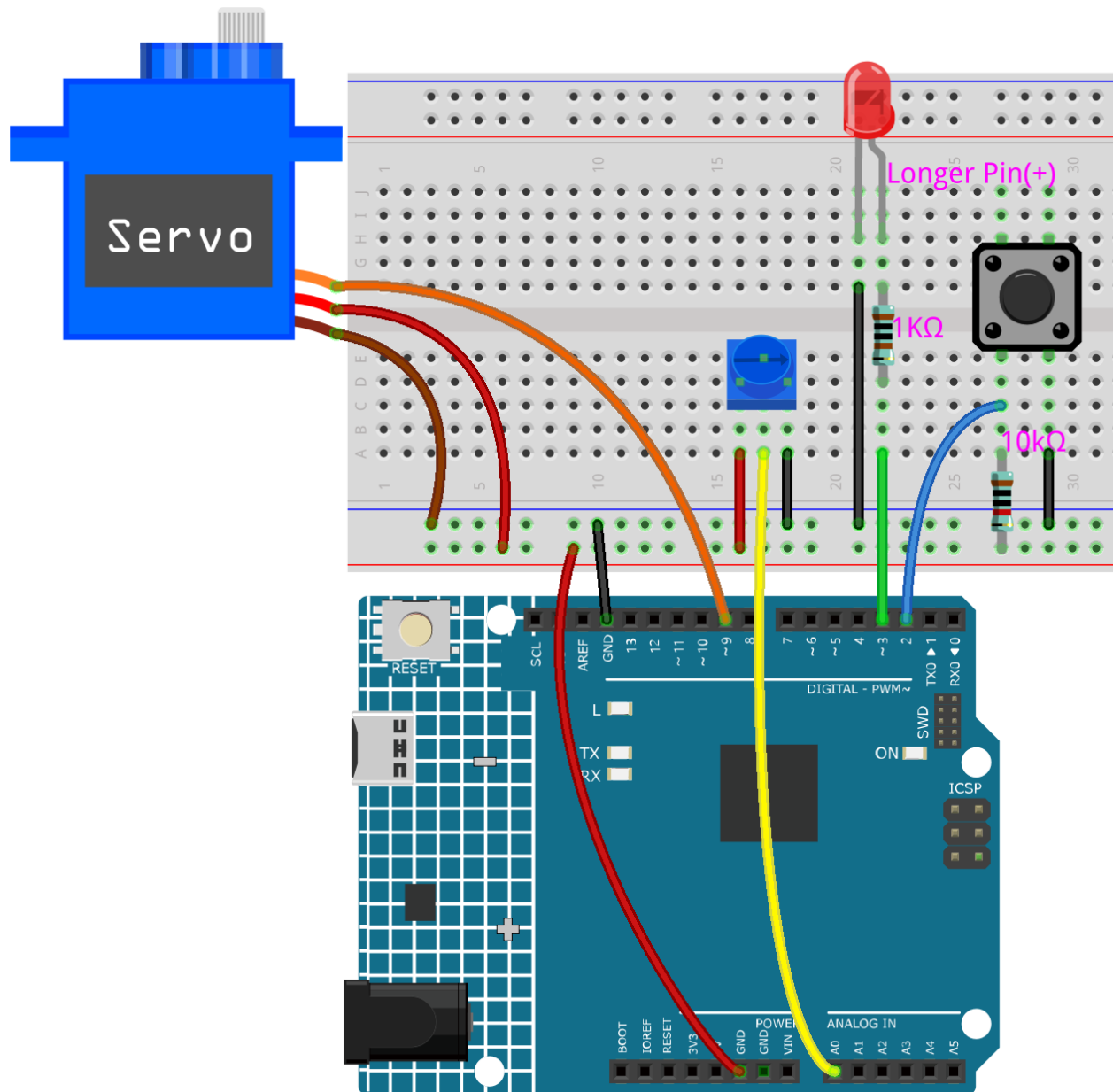
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Servo</i>	
<i>Knopf</i>	
<i>Potentiometer</i>	

Schaltplan



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.15.eeprom.ino im Pfad 3in1-kit\learning_project\5.15.eeprom.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Um diesen Schaltkreis zu verwenden, drücken Sie einfach den Knopf, um die Aufnahme zu starten, und geben die gewünschten Informationen über ein Potentiometer ein. Das Board wird nun Ihre Aktionen endlos wiederholen (und bei jeder Wiederholung blinkt eine LED), bis Sie den Knopf erneut drücken, um neue Aktionen aufzunehmen. Sie können auch die Aufnahmezeit ändern, indem Sie die Werte von resolution und recordTime ändern.

Wie funktioniert das?

1. Importieren Sie die EEPROM.h Bibliothek und initialisieren Sie den EEPROM-Speicher.

```

...
#include <EEPROM.h> //used to store recorded values

...
float resolution = 1000; //MUST be less than EEPROM.length()
float recordTime = 5; //delay time
bool recording = false;
...

```

Bitte beachten Sie, dass `/MUST be less than EEPROM.length()`, in `setup()` wird der Speicher Ihres Board-EEPROMs ausgedruckt, welcher für Ihr Board 1024 sein sollte. Wenn Sie ein anderes Board verwenden, können Sie den Wert der Variable `resolution` ändern.

2. Das EEPROM-Speicher Ihres Boards ausdrucken.

```

void setup() {
    ...
    Serial.begin(9600);
    //Serial.println(EEPROM.length());
}

```

Um die Größe des EEPROM-Speichers Ihres Boards herauszufinden, kommentieren Sie die Zeile `Serial.println(EEPROM.read(i))` aus. Dies wird die Größe des EEPROM im seriellen Monitor ausgeben, und Sie können den Wert der Variable `resolution` entsprechend ändern.

3. Sobald ein Tastendruck erkannt wird, beginnt die Aufzeichnung und die benötigten Informationen werden über ein Potentiometer eingegeben. Nun wiederholt das Board Ihre Aktion endlos (und blinkt bei jeder Wiederholung eine LED) bis Sie die Taste erneut drücken, um eine neue Aktion aufzuzeichnen.

```

void loop() {
    if (recording == true) { //record
        for (int i = 1; i <= resolution; i++) {
            digitalWrite(ledPin, HIGH); //light status led
            int val = map(analogRead(A0), 0, 1023, 0, 180);
            EEPROM.write(i, val);
            //Serial.println(EEPROM.read(i));
            myServo.write(val);
            delay(recordTime);
        }
        digitalWrite(ledPin, LOW); //turn off status led
        delay(1000); //give time for person
        recording = false;
    }
    else {
        for (int i = 1; i <= resolution; i++) { //playback
            if (digitalRead(buttonPin) == 0) { //stop playback and record
                ↪ new values
                recording = true;
                break;
            }
            int readval = EEPROM.read(i);
            myServo.write(readval);
            //Serial.println(readval);
            delay(recordTime);
        }
    }
}

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

    }
    digitalWrite(ledPin, HIGH); //show a new repeat
    delay(100);
    digitalWrite(ledPin, LOW);
  }
}

```

- Die Variable `recording` auf `true` setzen, wenn die Taste gedrückt wird.
- Wenn die Variable `recording` `true` ist, beginnt die Aufzeichnung im Speicherbereich.
- Den Wert des Potentiometers auslesen und auf 0-180 mappen, um ihn im EEPROM zu speichern und die Drehung des Servos zu steuern.
- Die LED leuchtet zu Beginn der Aufzeichnung und erlischt am Ende.
- Die aufgezeichnete Aktion mit einem kurzen Blinken der LED wiederholen, um Sie an eine neue Wiederholung zu erinnern.

4. Über die EEPROM-Bibliothek.

Hier sind einige ihrer Funktionen.

- `write(address, value)`: Schreibt ein Byte ins EEPROM.
 - `address`: der Ort, an den geschrieben werden soll, beginnend bei 0 (int)
 - `value`: der zu schreibende Wert, von 0 bis 255 (Byte)
 - Ein EEPROM-Schreibvorgang dauert 3,3 ms. Das EEPROM hat eine angegebene Lebensdauer von 100.000 Schreib-/Löschzyklen, daher sollten Sie vorsichtig sein, wie oft Sie darauf schreiben.
- `Read(address)`: Liest ein Byte aus dem EEPROM. Orte, die noch nie beschrieben wurden, haben den Wert 255.
- `update(address, value)`: Schreibt ein Byte ins EEPROM. Der Wert wird nur geschrieben, wenn er sich von dem bereits unter der gleichen Adresse gespeicherten unterscheidet.
 - Ein EEPROM-Schreibvorgang dauert 3,3 ms. Da das EEPROM eine Lebensdauer von 100.000 Schreib-/Löschzyklen hat, kann die Verwendung dieser Funktion anstelle von `write()` Zyklen sparen, wenn die geschriebenen Daten sich nicht oft ändern.
- `EEPROM.put(address, data)`: Schreibt jeden Datentyp oder jedes Objekt ins EEPROM.
 - `address`: der Ort, von dem gelesen werden soll, beginnend bei 0 (int)
 - `data`: die zu lesenden Daten, kann ein einfacher Typ (z. B. `float`) oder eine benutzerdefinierte Struktur sein.
 - Diese Funktion verwendet `EEPROM.update()`, um den Schreibvorgang durchzuführen, schreibt also den Wert nicht erneut, wenn er sich nicht geändert hat.
- `EEPROM.get(address, data)`: Liest jeden Datentyp oder jedes Objekt aus dem EEPROM.
 - `address`: der Ort, von dem gelesen werden soll, beginnend bei 0 (int)
 - `data`: die zu lesenden Daten, kann ein einfacher Typ (z. B. `float`) oder eine benutzerdefinierte Struktur sein.

4.6 6. Lustiges Projekt

In diesem Kapitel finden Sie einige Beispiele, die die grundlegende Logik veranschaulichen, wie die meisten Programme mit der Realität interagieren. Dies wird Ihnen helfen, sich mit der Arduino-Programmierung vertraut zu machen. Wenn Sie eine kreative Idee im Kopf haben, wird die Programmierung für Sie kein Hindernis mehr sein.

4.6.1 6.1 Lichtempfindliches Array

Ein Fotowiderstand oder Fotodiode ist ein lichtgesteuerter variabler Widerstand. Der Widerstand eines Fotowiderstands verringert sich mit zunehmender Lichteinstrahlung; anders ausgedrückt, zeigt er eine Photoleitfähigkeit. Ein Fotowiderstand kann in lichtempfindlichen Detektorschaltungen sowie in licht- und dunkelaktivierten Schaltkreisen eingesetzt werden.

Der Widerstand eines Fotowiderstands ändert sich mit der Lichteinstrahlung. Wenn die Lichtintensität steigt, verringert sich der Widerstand; wenn sie abnimmt, erhöht sich der Widerstand. In diesem Experiment werden wir acht LEDs verwenden, um die Lichtintensität anzuzeigen. Je höher die Lichtintensität ist, desto mehr LEDs leuchten auf. Wenn die Lichtintensität hoch genug ist, leuchten alle LEDs. Wenn kein Licht vorhanden ist, erlöschen alle LEDs.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

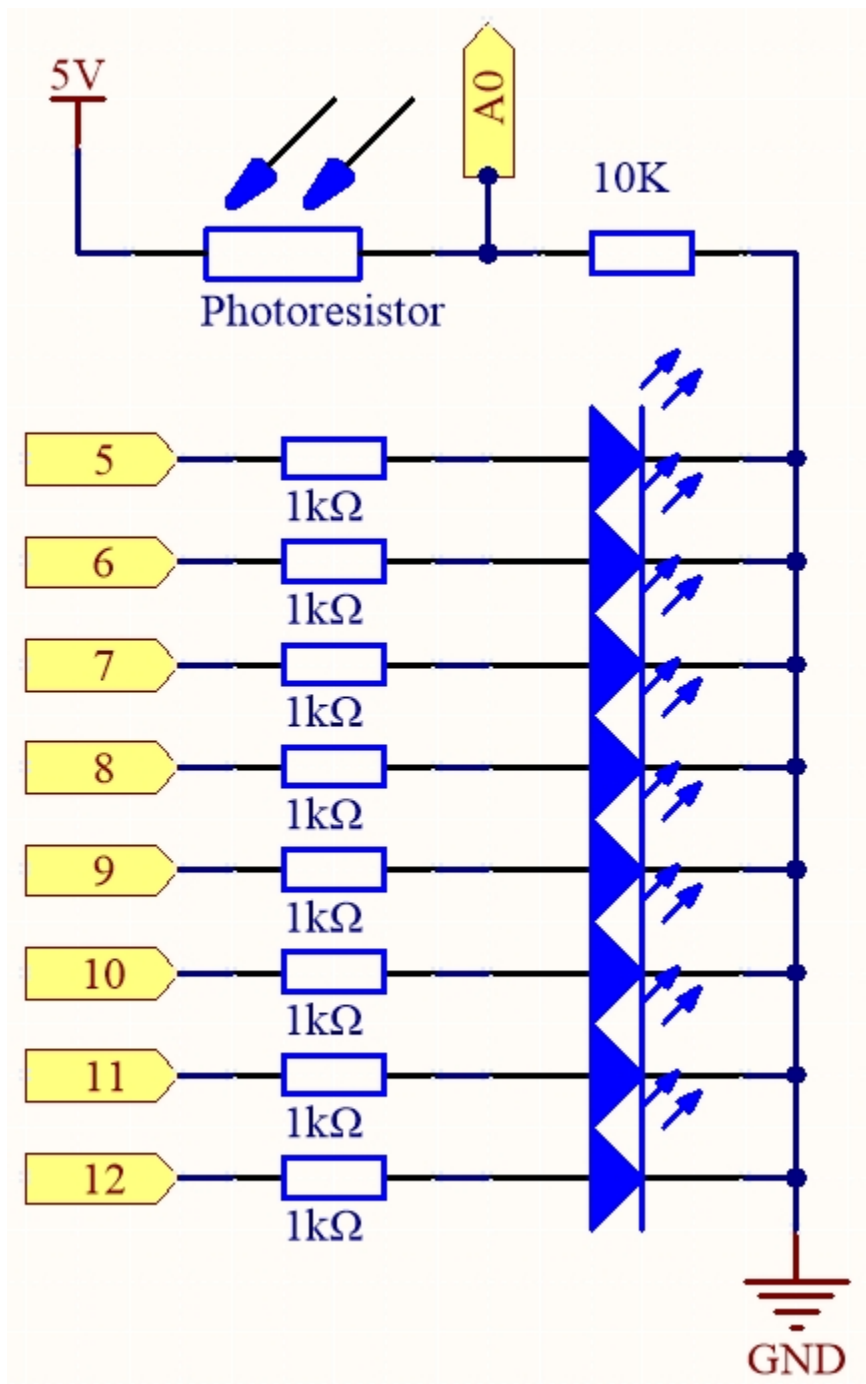
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

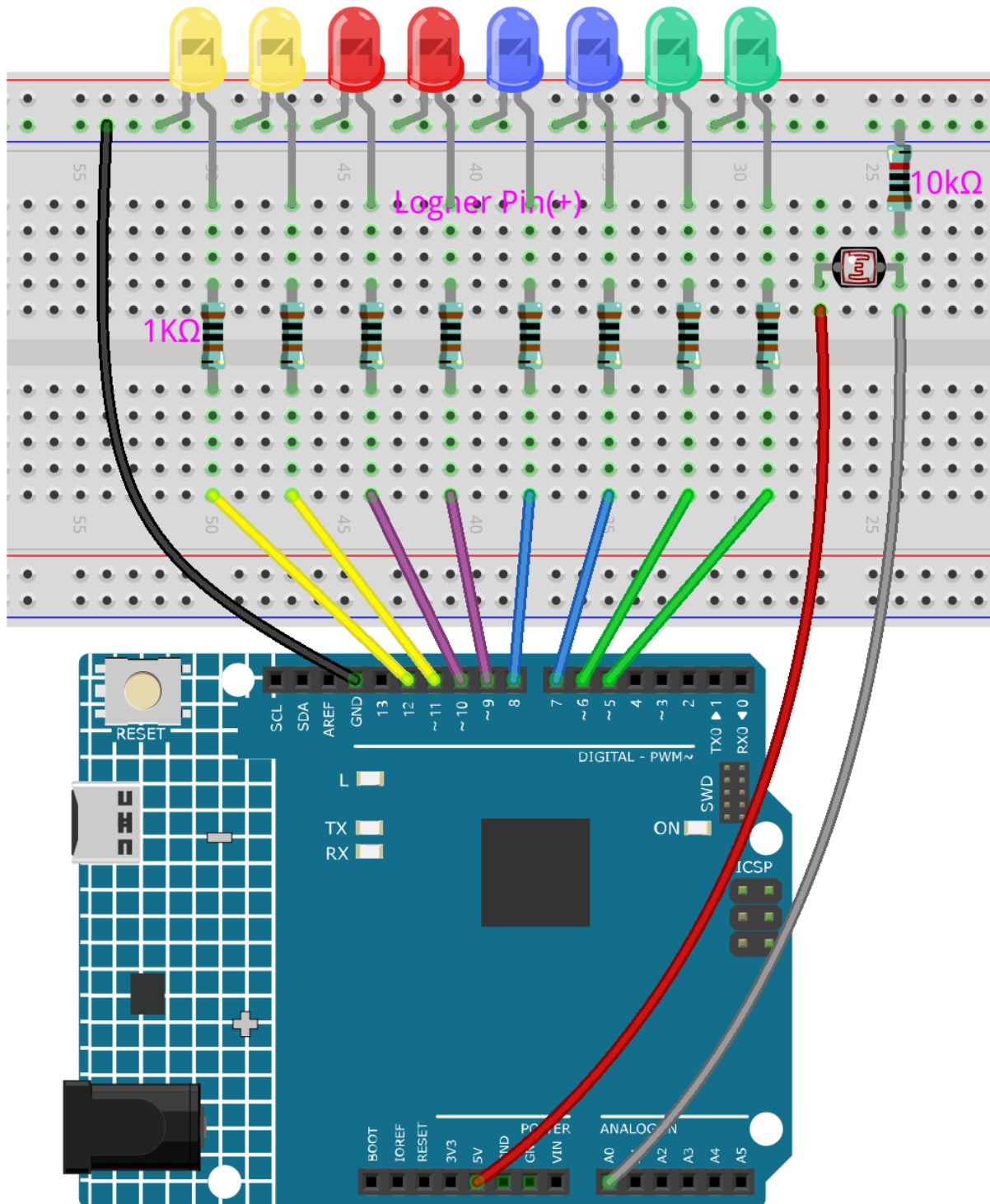
Sie können diese auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Fotowiderstand</i>	

Schaltplan



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 6.1.light_control_led.ino im Pfad 3in1-kit\learning_project\6.1.light_control_led.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Jetzt leuchten Sie mit einer Lichtquelle auf den Fotowiderstand, und Sie werden sehen, dass mehrere LEDs aufleuchten. Strahlen Sie mehr Licht darauf, leuchten mehr LEDs auf. Wenn Sie ihn in eine dunkle Umgebung bringen, erlöschen alle LEDs.

Wie funktioniert das?

```
void loop()
{
    sensorValue = analogRead(photocellPin); //read the value of A0
    ledLevel = map(sensorValue, 300, 1023, 0, NbrLEDs); // map to the number of LEDs
    for (int led = 0; led < NbrLEDs; led++)//
    {
        if (led < ledLevel ) //When led is smaller than ledLevel, run the following code.
        {
            digitalWrite(ledPins[led], HIGH); // turn on pins less than the level
        }
        else
        {
            digitalWrite(ledPins[led],LOW); // turn off pins higher than
        }
    }
}
```

Mit der Funktion map() können Sie den Wert des Fotowiderstands den 8 LEDs zuordnen, zum Beispiel, wenn sensor-Value 560 ist, dann ist ledLevel 4, also sollten zu diesem Zeitpunkt ledPins[0] bis ledPins[4] leuchten und ledPins[5] bis ledPins[7] sollten ausgeschaltet sein.

4.6.2 6.2 Digitaler Würfel

Hier verwenden wir einen Taster, ein 7-Segment-Display und den 74hc595, um einen elektronischen Würfel zu erstellen. Jedes Mal, wenn der Taster gedrückt wird, wird eine zufällige Zahl zwischen 1 und 6 erzeugt und auf dem 7-Segment-Display angezeigt.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

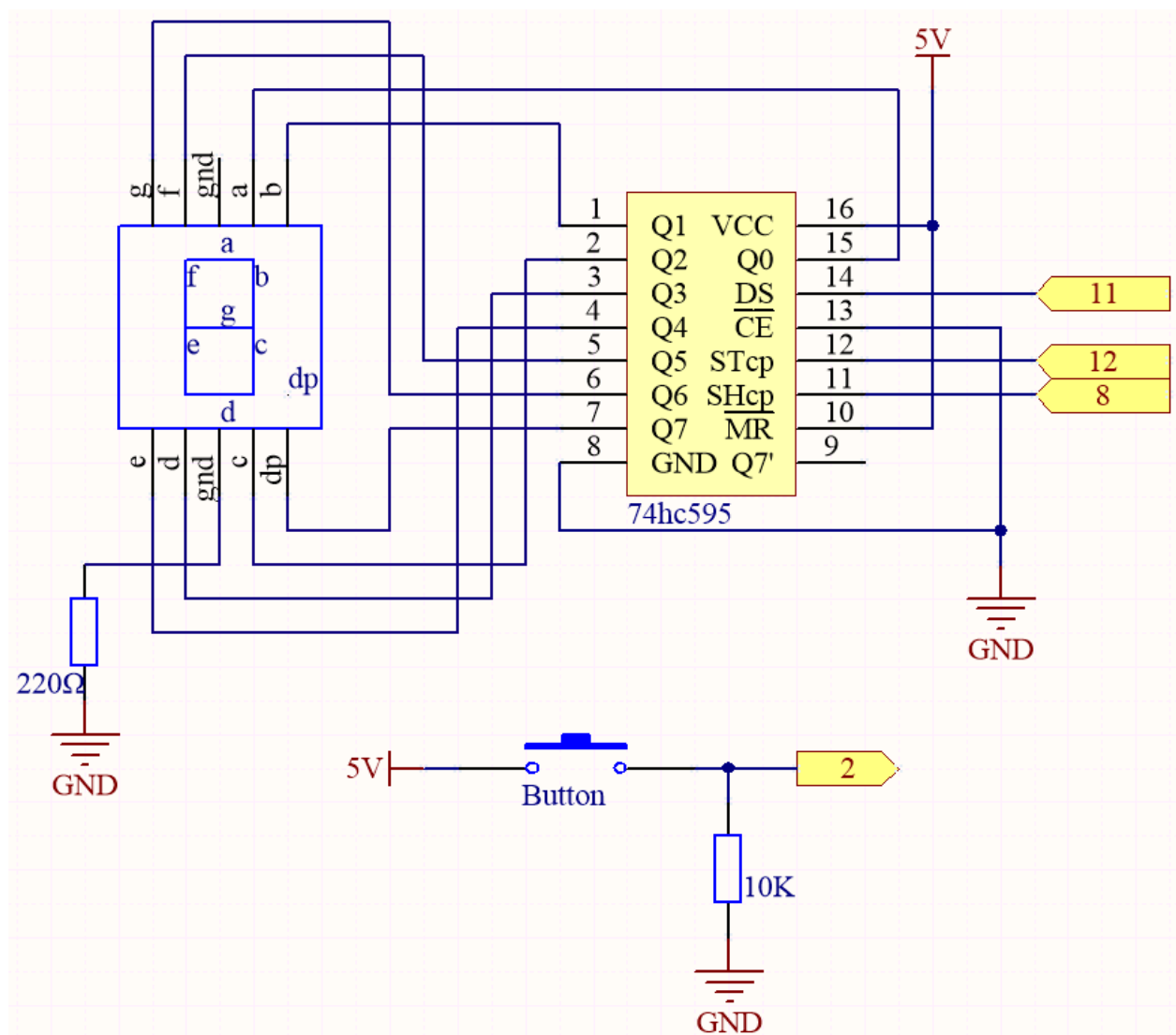
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 Minima	-
Breadboard	
Jumper-Kabel	
Widerstand	
Knopf	
74HC595	
7-Segment-Anzeige	

Schaltplan



Verdrahtung

Nachdem der Code erfolgreich hochgeladen wurde, zeigt das 7-Segment-Display schnell die Zahlen 0-7 an. Wenn Sie den Taster drücken, wird eine zufällige Zahl angezeigt und das schnelle Scrollen stoppt. Das schnelle Scrollen startet wieder, wenn Sie den Taster erneut drücken.

Wie funktioniert das?

Dieses Projekt basiert auf [5.10 ShiftOut\(Segmentanzeige\)](#) mit einem Taster, um das Scroll-Display auf dem 7-Segment-Display zu starten/pausieren.

1. Initialisieren Sie jeden Pin und lesen Sie den Wert des Tasters.

```
void setup ()
{
    ...
    attachInterrupt(digitalPinToInterrupt(buttonPin), rollDice, FALLING);
}
```

- Das Interrupt wird hier verwendet, um den Zustand des Tasters zu lesen. Der Standardwert von `buttonPin` ist niedrig, der sich ändert, wenn der Taster gedrückt wird.
- `rollDice` stellt die Funktion dar, die aufgerufen wird, wenn das Interrupt ausgelöst wird. Sie dient dazu, den Wert der Variablen `state` umzuschalten.
- `FALLING` bedeutet, dass das Interrupt ausgelöst wird, wenn der `buttonPin` von niedrig auf hoch wechselt.

2. Wenn die Variable `state` 0 ist, wird die Funktion `showNumber()` aufgerufen, um das 7-Segment-Display zufällig eine Zahl zwischen 1 und 7 anzeigen zu lassen.

```
void loop()
{
    if (state == 0) {
        showNumber((int)random(1, 7));
        delay(50);
    }
}
```

3. Über die Funktion `rollDice()`.

```
void rollDice() {
    state = !state;
}
```

Wenn diese Funktion aufgerufen wird, wechselt sie den Wert von `state`, zum Beispiel 1 beim letzten Mal und 0 diesmal.

4. Über die Funktion `showNumber()`.

```
void showNumber(int num) {
    digitalWrite(STcp, LOW); //ground ST_CP and hold low for as long as you
    are transmitting
    shiftOut(DS, SHcp, MSBFIRST, dataArray[num]);
    //return the latch pin high to signal chip that it
    //no longer needs to listen for information
    digitalWrite(STcp, HIGH); //pull the ST_CPST_CP to save the data
}
```

Dies ist der Code innerhalb der `loop()` im Projekt *5.10 ShiftOut(Segmentanzeige)* in der Funktion `showNumber()`.

4.6.3 6.3 Hohe Temperaturalarm

Im nächsten Schritt werden wir mit einem Thermistor, einem Druckknopf, einem Potentiometer und einem LCD ein Alarmgerät für hohe Temperaturen herstellen. Das LCD1602 zeigt die vom Thermistor erfasste Temperatur und den Schwellenwert für hohe Temperaturen an, der mit einem Potentiometer eingestellt werden kann. Der Schwellenwert wird gleichzeitig im EEPROM gespeichert. Wenn die aktuelle Temperatur den Schwellenwert überschreitet, wird der Summer ertönen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

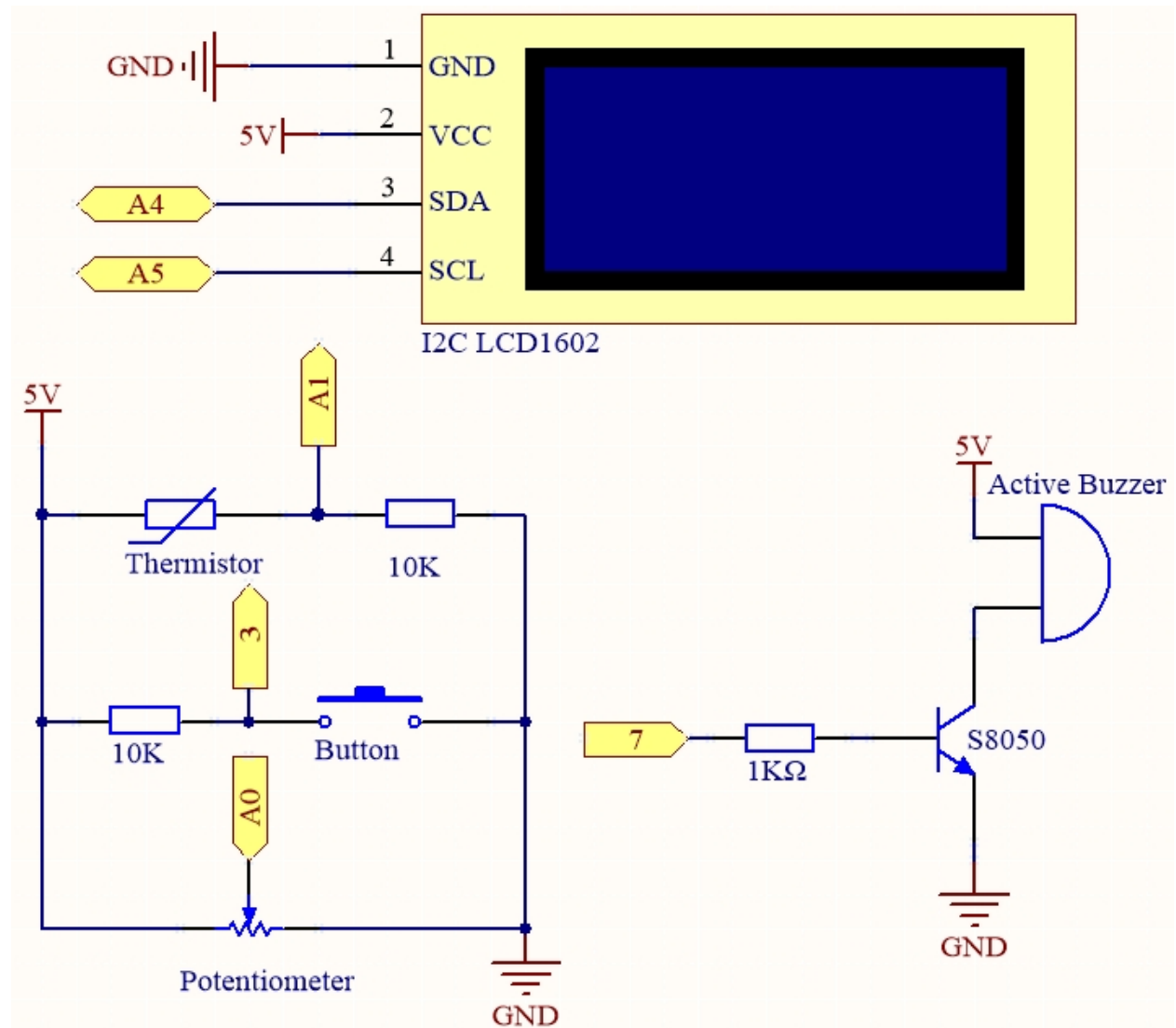
Es ist definitiv praktisch, ein ganzes Kit zu kaufen. Hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

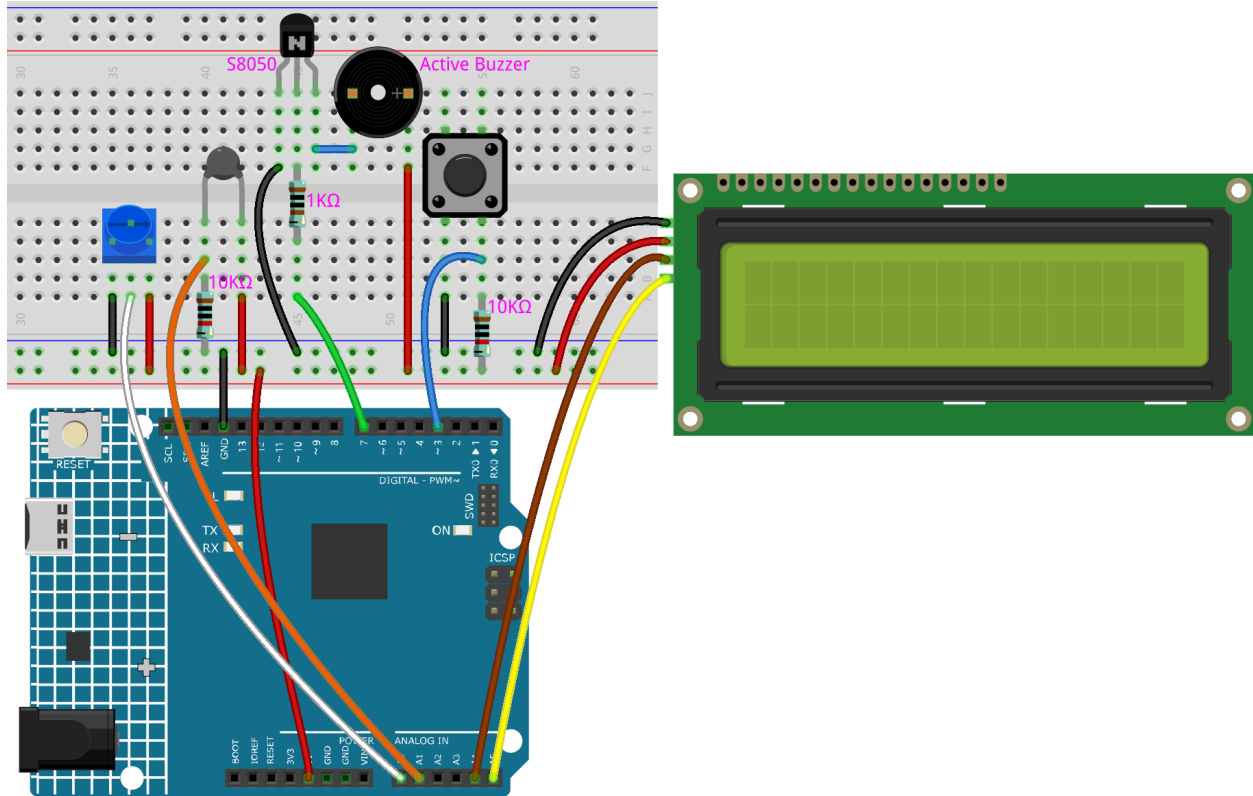
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Summer</i>	-
<i>Knopf</i>	
<i>I2C LCD1602</i>	
<i>Thermistor</i>	
<i>Potentiometer</i>	

Schaltplan



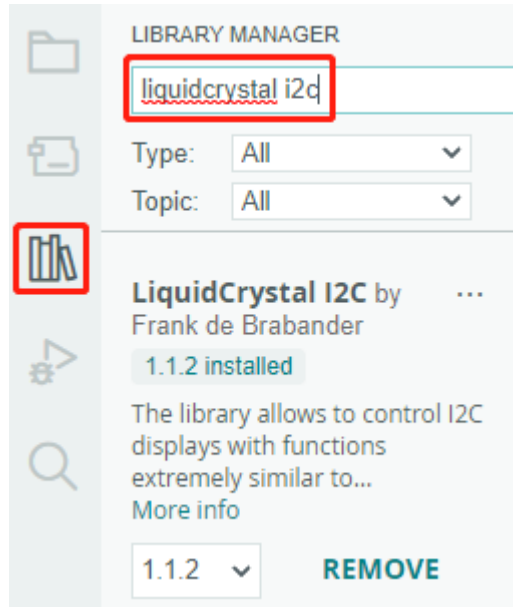
Verdrahtung



Code

Bemerkung:

- Sie können die Datei 6.3.high_tem_alarm.ino direkt im Pfad 3in1-kit\learning_project\6.3.high_tem_alarm öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.
- Hier wird die Bibliothek LiquidCrystal I2C verwendet. Sie können sie über den **Library Manager** installieren.



Nachdem der Code erfolgreich hochgeladen wurde, zeigt das LCD1602 die vom Thermistor erfasste Temperatur und den Schwellenwert für hohe Temperaturen an, der mit einem Potentiometer eingestellt werden kann. Der Schwellenwert wird gleichzeitig im EEPROM gespeichert. Wenn die aktuelle Temperatur den Schwellenwert überschreitet, wird der Summer ertönen.

Bemerkung: Wenn der Code und die Verkabelung in Ordnung sind, das LCD jedoch immer noch keinen Inhalt anzeigt, können Sie das Potentiometer auf der Rückseite drehen.

Wie funktioniert das?

1. Initialisieren Sie den Button, den Summer und das I2C LCD1602 und lesen Sie die Werte aus dem EEPROM aus. Ein Interrupt wird ebenfalls verwendet, um den Status des Buttons zu lesen.

```
void setup()
{
  pinMode(buzzerPin, OUTPUT);
  pinMode(buttonPin, INPUT);
  lcd.init();
  lcd.backlight();
  upperTem = EEPROM.read(0);
  delay(1000);
  attachInterrupt(digitalPinToInterrupt(buttonPin), buttonState, FALLING);
}
```

- Der Interrupt wird hier verwendet, um den Status des Buttons zu lesen. Wenn der Button gedrückt wird, wechselt `buttonPin` von niedrig zu hoch.
 - Die Funktion `buttonState` wird aufgerufen, wenn der Interrupt ausgelöst wird, und sie schaltet den Wert der Variable `state` um.
 - `FALLING` bedeutet, dass der Interrupt auftritt, wenn `buttonPin` von niedrig auf hoch wechselt.
2. Um den Schwellenwert für hohe Temperaturen festzulegen, wird die Funktion `upperTemSetting()` aufgerufen, wenn `state` gleich 1 ist (`state` wechselt zwischen 0 und 1 mit Button-Druck) im Hauptprogramm. Andernfalls wird

monitoringTemp() aufgerufen, um die aktuelle Temperatur und den eingestellten Schwellenwert anzuzeigen.

```
void loop()
{
    if (state == 1)
    {
        upperTemSetting();
    }
    else {
        monitoringTemp();
    }
}
```

3. Über die Funktion upperTemSetting().

```
void upperTemSetting()
{
    int setTem = 0;

    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Adjusting...");
    lcd.setCursor(0, 1);
    lcd.print("Upper Tem: ");

    while (1) {
        lcd.setCursor(11, 1);
        setTem = map(analogRead(potPin), 0, 1023, 0, 100);
        lcd.print(setTem);
        if (state == 0)
        {
            EEPROM.write(0, setTem);
            upperTem = setTem;
            lcd.clear();
            return;
        }
    }
}
```

- Mit dieser Funktion kann ein Schwellenwert festgelegt werden. Wenn Sie diese Funktion betreten, zeigt das LCD1602 den aktuellen Schwellenwert an, der mit dem Potentiometer geändert werden kann. Dieser Schwellenwert wird im EEPROM gespeichert und verlassen, wenn der Button erneut gedrückt wird.

4. Über die Funktion monitoringTemp().

```
void monitoringTemp()
{
    long a = analogRead(temPin);
    float tempC = beta / (log((1025.0 * 10 / a - 10) / 10) + beta / 298.0) -
    ↪ 273.0;
    float tempF = 1.8 * tempC + 32.0;
    lcd.setCursor(0, 0);
    lcd.print("Temp: ");
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
lcd.print(tempC);  
lcd.print(char(223));  
lcd.print("C  ");  
lcd.setCursor(0, 1);  
lcd.print("Upper: ");  
lcd.print(upperTem);  
lcd.print(char(223));  
lcd.print("C  ");  
delay(300);  
if (tempC >= upperTem)  
{  
    digitalWrite(buzzerPin, HIGH);  
    delay(50);  
    digitalWrite(buzzerPin, LOW);  
    delay(10);  
}  
else  
{  
    digitalWrite(buzzerPin, LOW);  
}  
}
```

- Mit dieser Funktion können Sie die Temperatur anzeigen und einen Alarm setzen.
- Der Wert des Thermistors wird gelesen und dann mit der Formel in Celsius-Temperatur umgewandelt und auf dem LCD1602 angezeigt.
- Der eingestellte Schwellenwert wird ebenfalls auf dem LCD angezeigt.
- Wenn die aktuelle Temperatur höher als der Schwellenwert ist, wird der Summer einen Alarm ausgeben.

4.6.4 6.4 Rückwärts-Hilfssystem

Mit dem Fortschritt von Wissenschaft und Technologie wurden viele Hightech-Produkte in Autos eingebaut. Eines davon ist das Rückwärts-Hilfssystem. In diesem Projekt verwenden wir ein Ultraschallmodul, LCD, LED und Buzzer, um ein einfaches ultraschallbasiertes Rückwärts-Hilfssystem zu erstellen.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

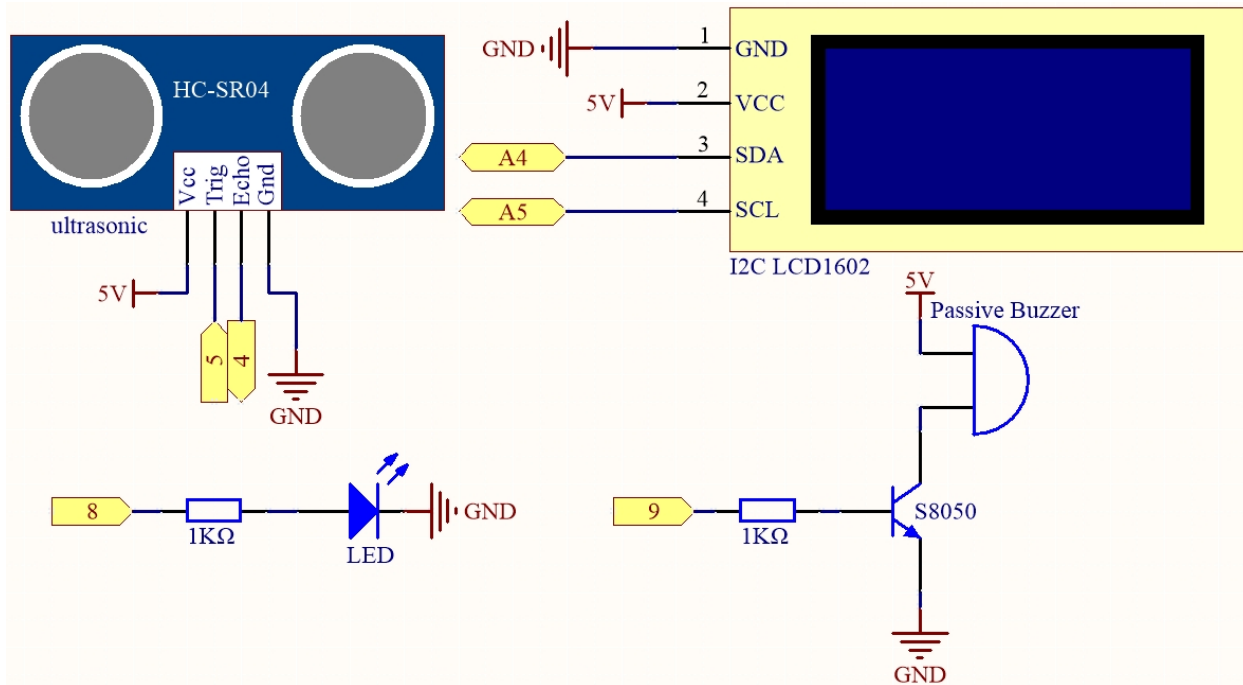
Es ist sicherlich praktisch, ein ganzes Set zu kaufen, hier der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

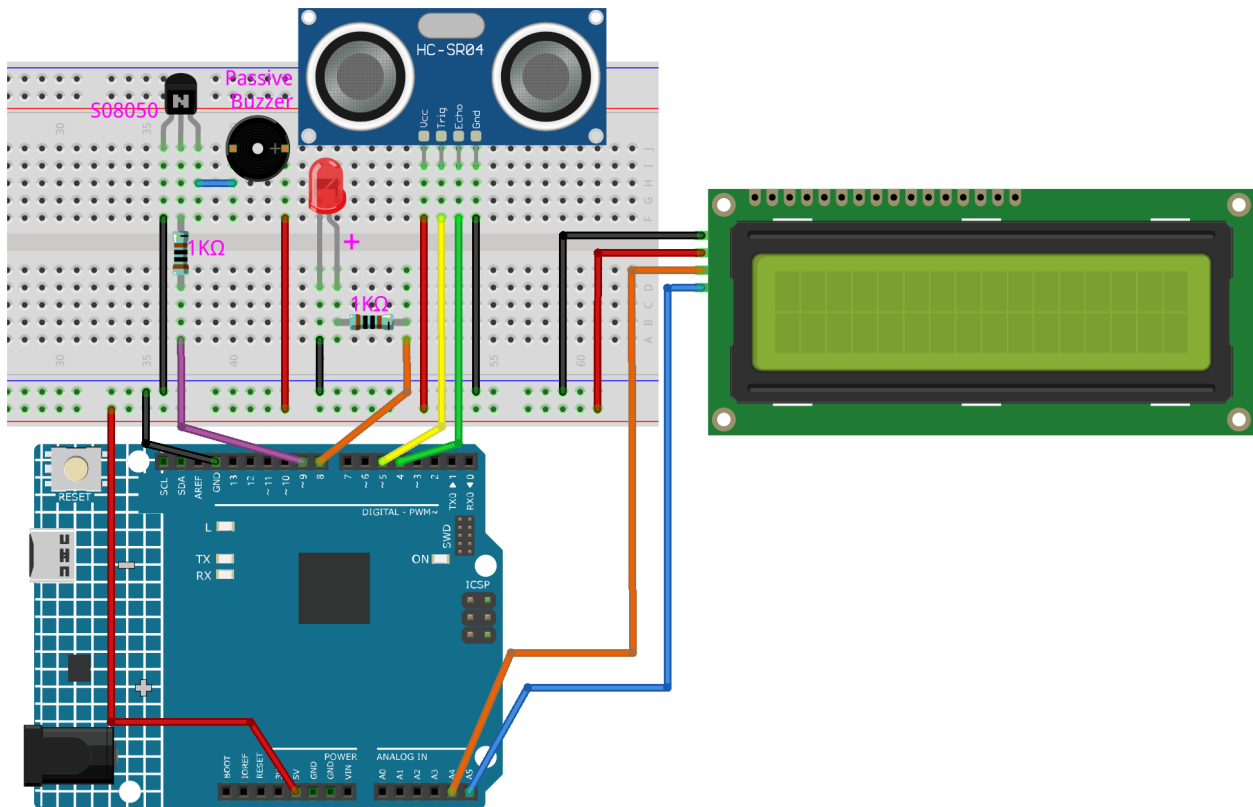
Sie können diese auch separat über die folgenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 Minima	-
Breadboard	
Jumper-Kabel	
Widerstand	
LED	
Summer	
I2C LCD1602	
Ultraschall-Modul	

Schaltplan



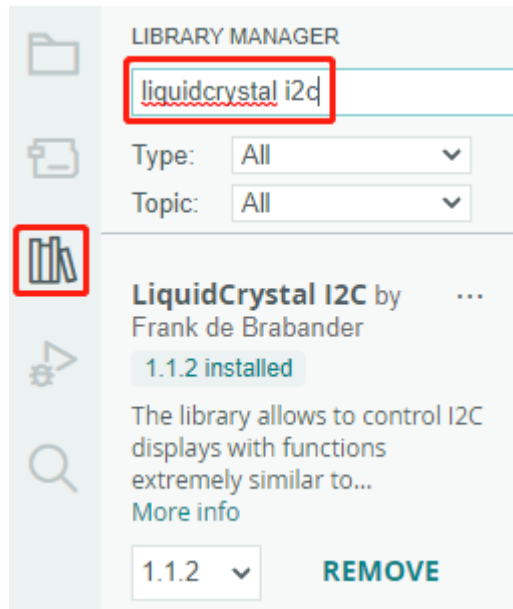
Verdrahtung



Code

Bemerkung:

- Sie können die Datei 6.4_reversingAid.ino direkt im Pfad 3in1-kit\learning_project\6.4_reversingAid öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE.
- Die Bibliothek LiquidCrystal I2C wird hier verwendet. Sie können diese aus dem **Library Manager** installieren.



Nachdem der Code erfolgreich hochgeladen wurde, wird die aktuell erkannte Entfernung auf dem LCD angezeigt. Der Buzzer ändert dann die Tonfrequenz je nach Entfernung.

Bemerkung: Wenn Code und Verkabelung in Ordnung sind, das LCD aber dennoch keinen Inhalt anzeigt, können Sie das Potentiometer auf der Rückseite drehen.

Wie funktioniert das?

Dieser Code hilft uns, ein einfaches Entfernungsmessgerät zu erstellen, das die Entfernung zwischen Objekten messen und Feedback über ein LCD-Display und einen Buzzer geben kann.

Die `loop()`-Funktion enthält die Hauptlogik des Programms und läuft kontinuierlich. Werfen wir einen genaueren Blick auf die `loop()`-Funktion.

1. Schleife zum Lesen der Entfernung und Aktualisieren von Parametern

In der `loop` liest der Code zuerst die vom Ultraschallmodul gemessene Entfernung und aktualisiert den Intervall-Parameter basierend auf der Entfernung.

```
// Update the distance
distance = readDistance();

// Update intervals based on distance
if (distance <= 10) {
  intervals = 300;
} else if (distance <= 20) {
  intervals = 500;
} else if (distance <= 50) {
  intervals = 1000;
} else {
  intervals = 2000;
}
```

2. Überprüfen, ob es Zeit zum Piepen ist

Der Code berechnet den Unterschied zwischen der aktuellen Zeit und der vorherigen Piep-Zeit. Wenn der Unterschied größer oder gleich der Intervall-Zeit ist, löst er den Buzzer aus und aktualisiert die vorherige Piep-Zeit.

```
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= intervals) {
    Serial.println("Beeping!");
    beep();
    previousMillis = currentMillis;
}
```

3. LCD-Display aktualisieren

Der Code löscht das LCD-Display und zeigt dann „Dis:“ und die aktuelle Entfernung in Zentimetern auf der ersten Zeile an.

```
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Dis: ");
lcd.print(distance);
lcd.print(" cm");

delay(100);
```

4.6.5 6.5 Reaktionsspiel

Unser Körper hat viele Reaktionszeiten, wie z.B. akustische RT, visuelle RT, taktile RT usw.

Reaktionszeiten haben viele Auswirkungen auf unser tägliches Leben. Zum Beispiel kann eine langsamer als normale Reaktionszeit beim Autofahren zu schwerwiegenden Folgen führen.

In diesem Projekt verwenden wir 3 Tasten und 2 LEDs, um unsere visuelle Reaktionszeit zu messen.

Das serielle Monitor des Arduino zeigt die Nachricht „Warte...“ an. Nachdem die „Bereit“-Taste gedrückt wurde, muss eine der beiden LEDs nach einer zufälligen Zeit zufällig aufleuchten. Es ist wichtig, dass der Proband die entsprechende Taste so schnell wie möglich drückt. Der Arduino zeichnet den Zeitunterschied zwischen dem Aufleuchten der LED und dem Drücken der entsprechenden Taste durch die Person auf und gibt die gemessene Reaktionszeit auf dem Arduino seriellen Monitor aus.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

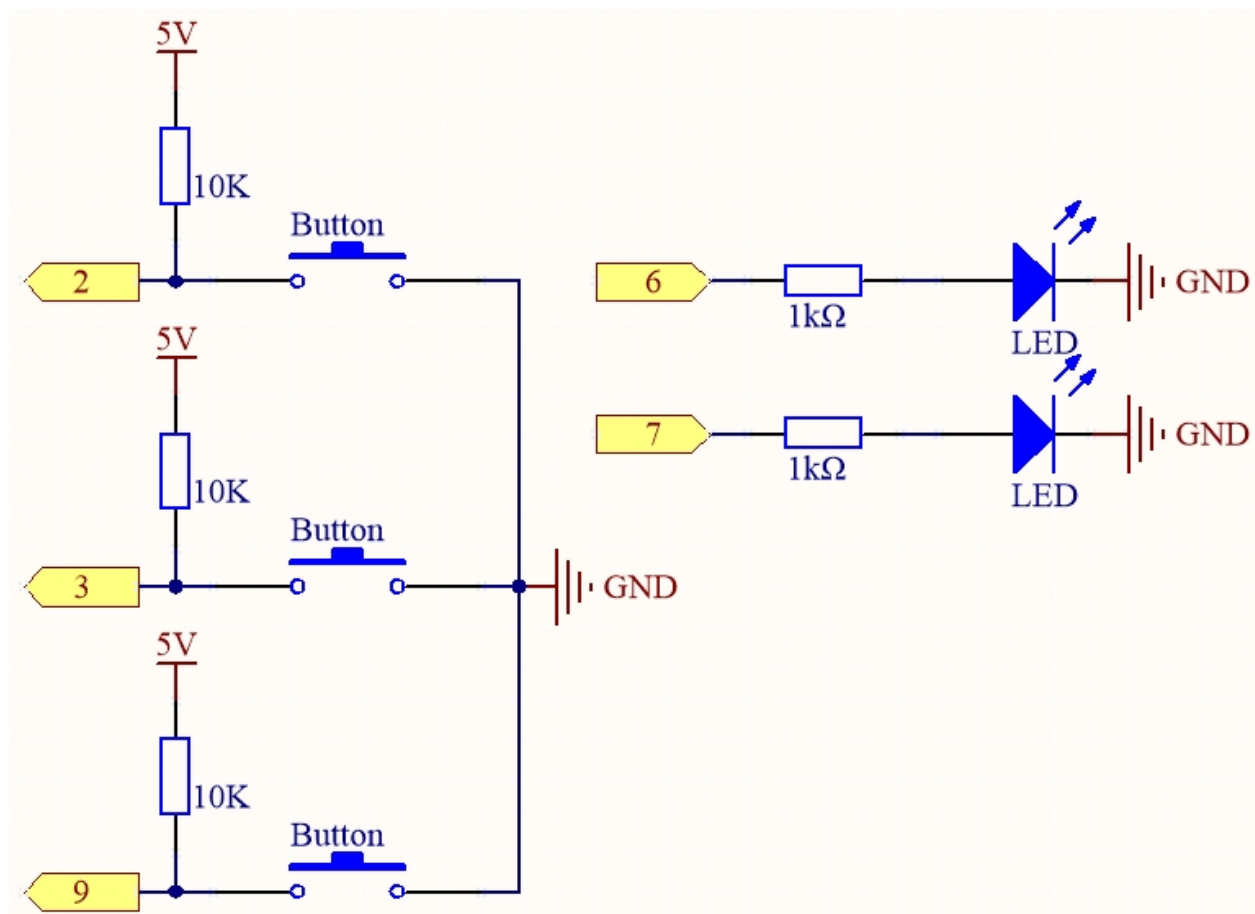
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Knopf</i>	

Schaltplan



Verdrahtung



- Oder kopieren Sie diesen Code in die Arduino IDE.
- Stellen Sie sicher, dass Sie die LiquidCrystal_I2C-Bibliothek hinzugefügt haben. Detaillierte Anleitungen finden Sie unter [5.11 Externe Bibliotheken installieren](#).

Wie funktioniert das?

1. Initialisieren Sie die Tasten und LEDs, hier werden 2 Interrupts verwendet, um den Status der Tasten auszulesen.

```
void setup()
{
    ...

    attachInterrupt(digitalPinToInterrupt(buttonPin1), pressed1, FALLING);
    attachInterrupt(digitalPinToInterrupt(buttonPin2), pressed2, FALLING);
    ...
}
```

2. Wenn die rstBtn-Taste gedrückt wird, beginnt das Spiel von Neuem. Nach einer zufälligen Zeit zwischen 2 und 5ms wird eine der LEDs eingeschaltet.

```
void loop()
{
    if (flag == -1 && digitalRead(rstBtn) == LOW) {
        digitalWrite(ledPin1, LOW);
        digitalWrite(ledPin2, LOW);
        Serial.println("Waiting...");
        int randomTime = random(2000, 5000);
        delay(randomTime);

        timer = millis();
        flag = randomTime % 2;
        Serial.println("Light!");

        if (flag == 0) {
            digitalWrite(ledPin1, HIGH);
        } else if (flag == 1) {
            digitalWrite(ledPin2, HIGH);
        }
    }
    delay(200);
}
```

- Wenn flag -1 ist und die rstBtn-Taste gedrückt wird, wird die random()-Funktion verwendet, um eine zufällige Zeit von 2-5s zu generieren.
 - Diese Zeit wird dann verwendet, um das Licht der LEDs zu steuern.
 - Das Einschalten der 2 LEDs wird zufällig mit randomTime % 2 mit 0 und 1 generiert. Wenn flag 0 ist, dann leuchtet LED1; wenn 1, dann leuchtet LED2.
3. Über die Funktion pressed1()

```
void pressed1() {
    if (flag == -1) {
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

        return;
    }
    if (flag == 0) {
        int currentTime = millis();
        Serial.print("Correct! You reaction time is : ");
        Serial.print(currentTime - timer);
        Serial.println(" ms");
    } else if (flag == 1) {
        Serial.println("Wrong Click!");
    }
    flag = -1;
}

```

Dies ist die Funktion, die ausgelöst wird, wenn die Taste 1 gedrückt wird. Wird die Taste 1 gedrückt und ist flag zu diesem Zeitpunkt 0, wird die Reaktionszeit ausgegeben, ansonsten wird ein Druckfehler angezeigt.

4. Über die Funktion pressed2()

```

void pressed2() {
    if (flag == -1) {
        return;
    }
    if (flag == 1) {
        int currentTime = millis();
        Serial.print("Correct! You reaction time is : ");
        Serial.print(currentTime - timer);
        Serial.println(" ms");
    } else if (flag == 0) {
        Serial.println("Wrong Click!");
    }
    flag = -1;
}

```

Dies ist die Funktion, die ausgelöst wird, wenn die Taste 2 gedrückt wird. Wird die Taste 2 gedrückt und ist flag zu diesem Zeitpunkt 1, wird die Reaktionszeit ausgegeben, ansonsten wird ein Druckfehler angezeigt.

4.6.6 6.6 Nummer raten

„Nummer raten“ ist ein unterhaltsames Partyspiel, bei dem Sie und Ihre Freunde abwechselnd eine Zahl (0~99) eingeben. Der Bereich wird mit der Eingabe der Zahl immer kleiner, bis ein Spieler das Rätsel richtig löst. Dann wird der Spieler besiegt und bestraft. Zum Beispiel, wenn die Glückszahl 51 ist, die die Spieler nicht sehen können, und Spieler 1 gibt 50 ein, ändert sich der Zahlenbereich auf 50~99; gibt Spieler 2 die Zahl 70 ein, kann der Bereich der Zahl 50~70 sein; gibt Spieler 3 die Zahl 51 ein, ist er oder sie der Unglückliche. Hier verwenden wir eine IR-Fernbedienung zur Eingabe von Zahlen und ein LCD zur Ausgabe der Ergebnisse.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

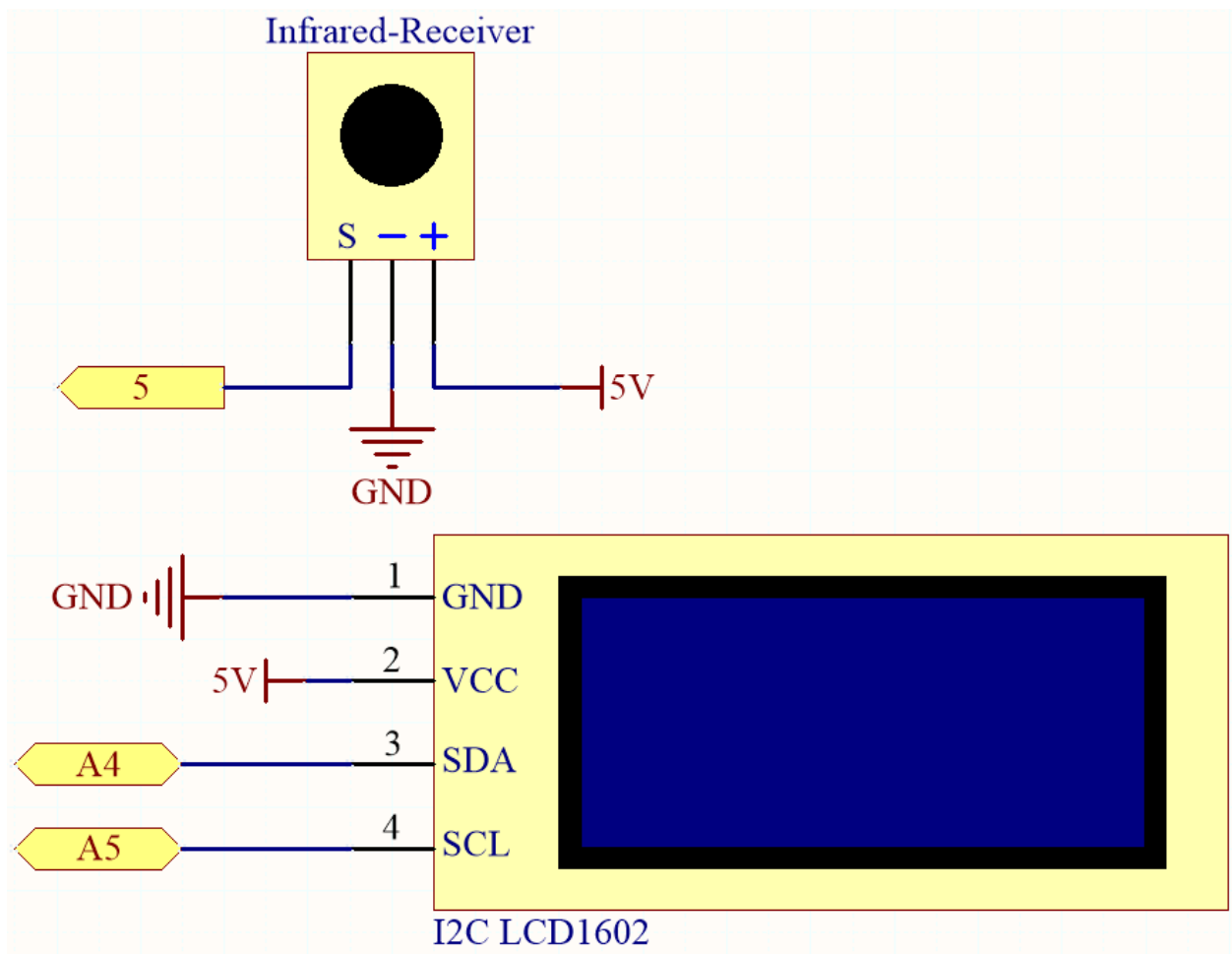
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

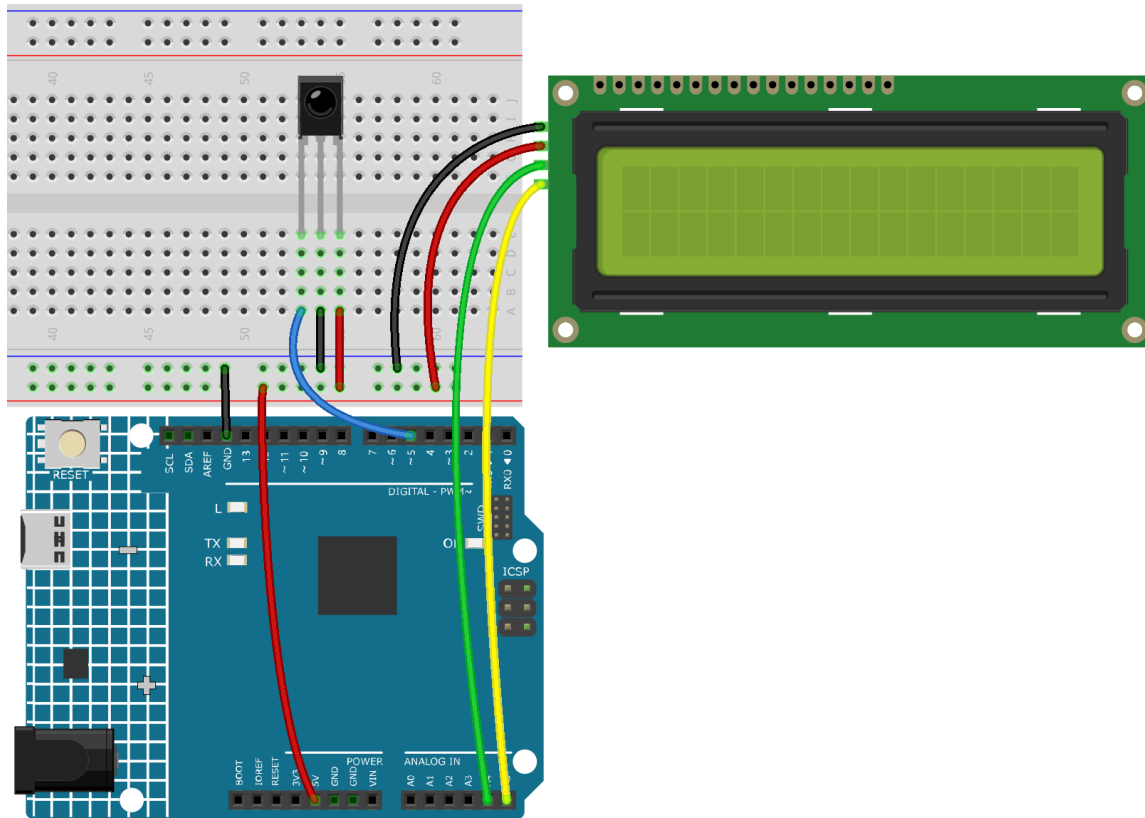
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>I2C LCD1602</i>	
<i>IR-Empfänger</i>	-

Schaltplan



Verdrahtung

In diesem Beispiel ist die Verkabelung des LCD1602 und des Infrarot-Empfangsmoduls wie folgt:



Code

Bemerkung:

- Sie können die Datei `6.6.guess_number.ino` direkt unter dem Pfad `3in1-kit\learning_project\6.6.guess_number` öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE.
 - Hier werden die Bibliotheken `LiquidCrystal`, `I2C` und `IRremote` libraries verwendet, die Sie über den **Library Manager** installieren können.
-

Nachdem der Code erfolgreich hochgeladen wurde, erscheinen die Begrüßungszeichen auf dem LCD1602. Drücken Sie nun die Zahl entsprechend dem Bereichshinweis auf dem Bildschirm, die Anzeige wird immer kleiner, es sei denn, Sie erraten diese Glückszahl.

Bemerkung: Wenn Code und Verkabelung in Ordnung sind, das LCD aber trotzdem keinen Inhalt anzeigt, können Sie das Potentiometer auf der Rückseite drehen, um den Kontrast zu erhöhen.

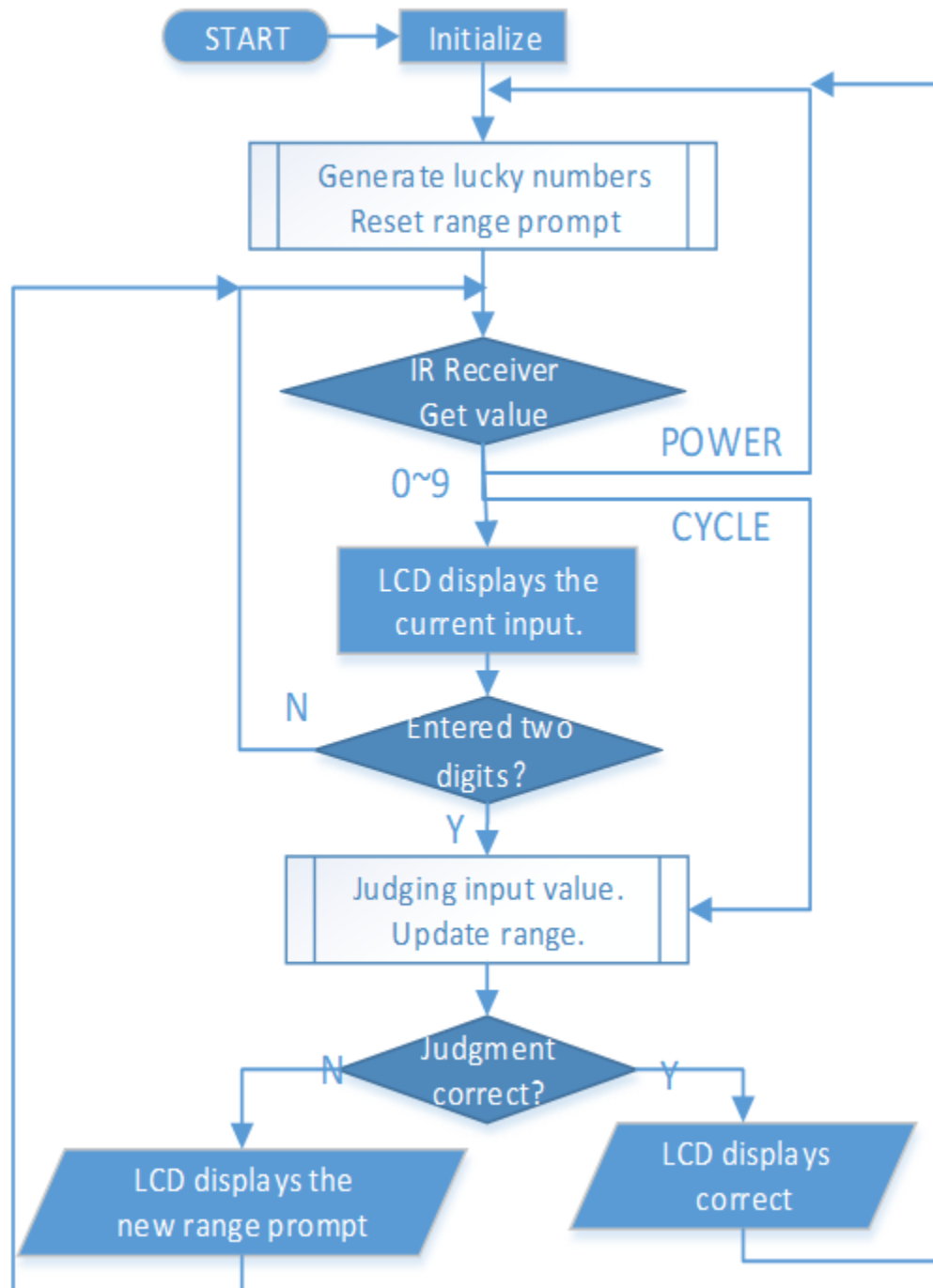
Wie funktioniert das?

Um das Zahlenratenspiel lebendig und lustig zu gestalten, müssen wir die folgenden Funktionen umsetzen:

1. Die Glückszahl wird angezeigt, wenn wir das Spiel starten und zurücksetzen, und der Zahlenbereichshinweis wird auf 0 ~ 99 zurückgesetzt.
2. Das LCD zeigt die eingegebene Zahl und den Bereichshinweis an.
3. Nach Eingabe von zwei Ziffern erscheint automatisch eine Ergebnisbewertung.

4. Wenn Sie eine einzelne Ziffer eingeben, können Sie die CYCLE-Taste (die Taste in der Mitte des Controllers) drücken, um die Ergebnisbewertung zu starten.
5. Wenn die Antwort nicht erraten wird, wird der neue Bereichshinweis angezeigt (wenn die Glückszahl 51 ist und Sie 50 eingeben, ändert sich der Bereichshinweis auf 50~99).
6. Das Spiel wird automatisch zurückgesetzt, nachdem die Glückszahl erraten wurde, sodass der Spieler eine neue Runde spielen kann.
7. Das Spiel kann durch direktes Drücken der POWER-Taste (die Taste in der oberen linken Ecke) zurückgesetzt werden.

Abschließend wird der Arbeitsablauf des Projekts im Flussdiagramm dargestellt.



Ich bin sicher, Sie haben bereits viele verschiedene intelligente Roboter-Autos gesehen. Ihre Grundfunktionen sind ähnlich: grundlegende Bewegungen, Hindernisvermeidung, Linienverfolgung, Nachfahren und Steuerung per Fernbedienung usw.

Hier nutzen wir die einfachste Struktur, um ein intelligentes Roboter-Auto zu bauen, welches ebenso all diese Funktionen erfüllen kann. Zudem können Sie es mit Ihrem Handy steuern. Bitte entnehmen Sie die Anleitung dazu unter [8. IoT-Auto](#).

Montageanleitung

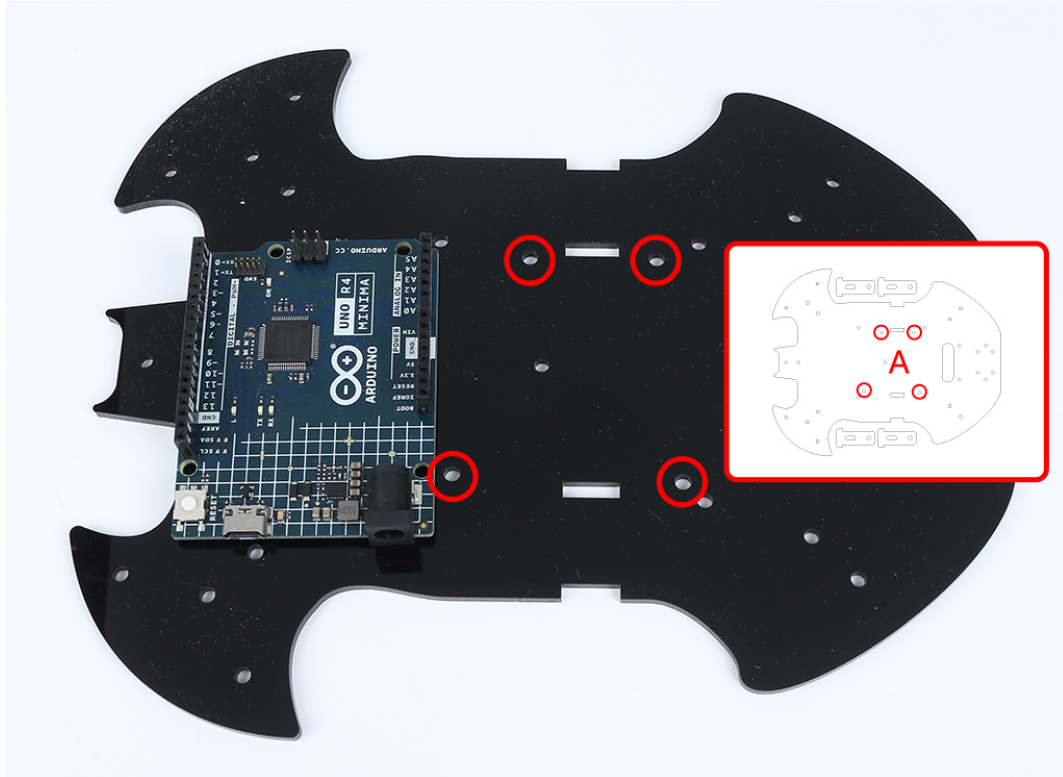
5.1 Auto zusammenbauen

Bitte befolgen Sie die folgenden Schritte, um die Montage des Autos abzuschließen.

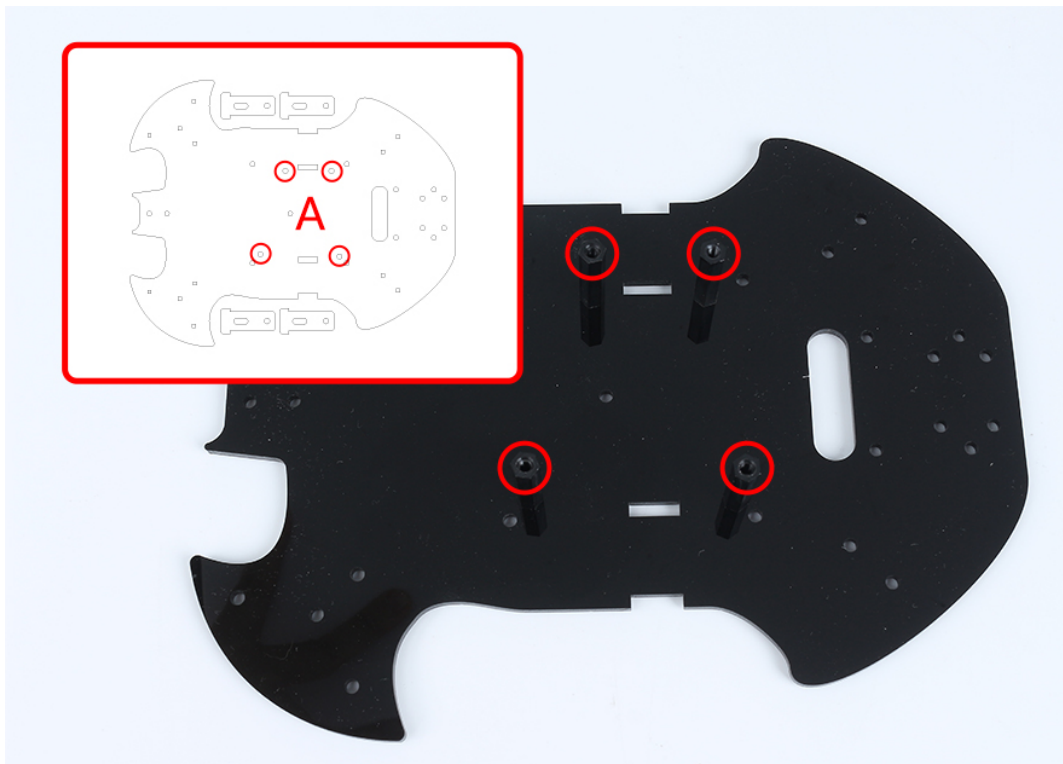
1. Entfernen Sie die Schutzfolie vom Acryl.



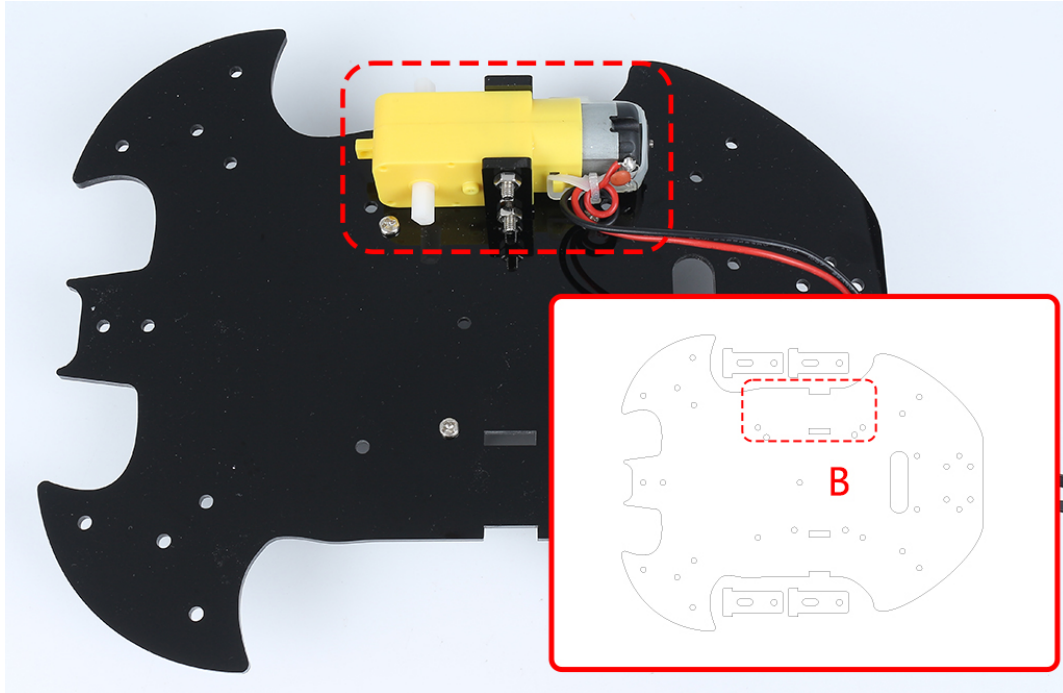
2. Legen Sie die Platte wie auf dem Bild gezeigt auf den Tisch, die Seite mit dem gleichen Loch wie die R4-Platte nennen wir A; die Rückseite ist B. Dies wird Ihnen helfen, Fehler während der Montage zu vermeiden.



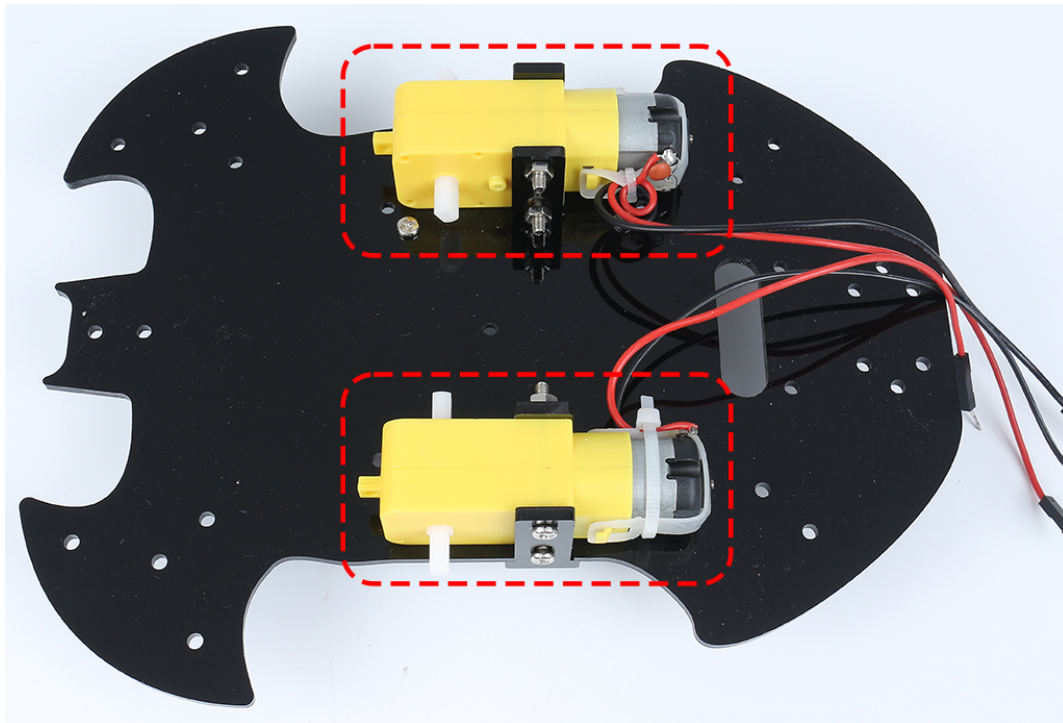
3. Montieren Sie den **M3x24mm Abstandshalter** mit **M3x6mm Schrauben** an der unten gezeigten Position.



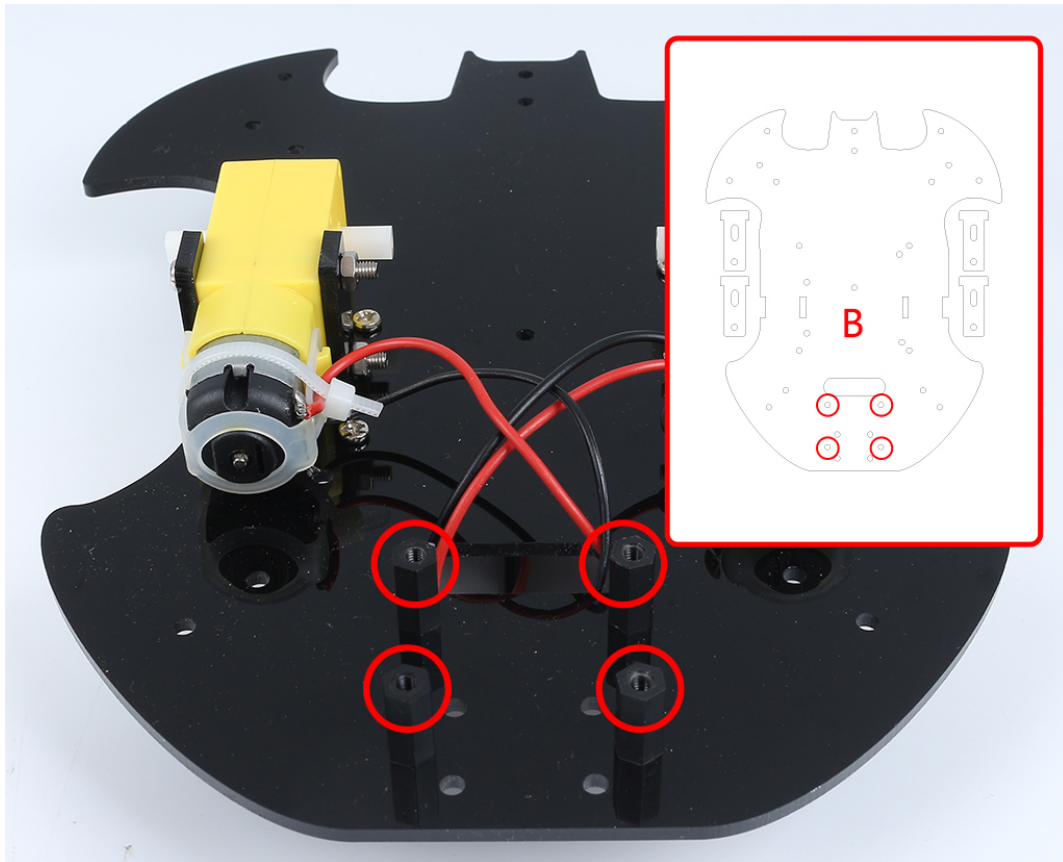
4. Drehen Sie zur B-Seite und verwenden Sie **M3x30mm Schrauben** und **M3 Muttern**, um den TT-Motor anzubringen. Beachten Sie 2 Details: 1 - Die Ausgangswelle zeigt zur fledermausförmigen Seite; 2 - Das Motorkabel zeigt nach innen.



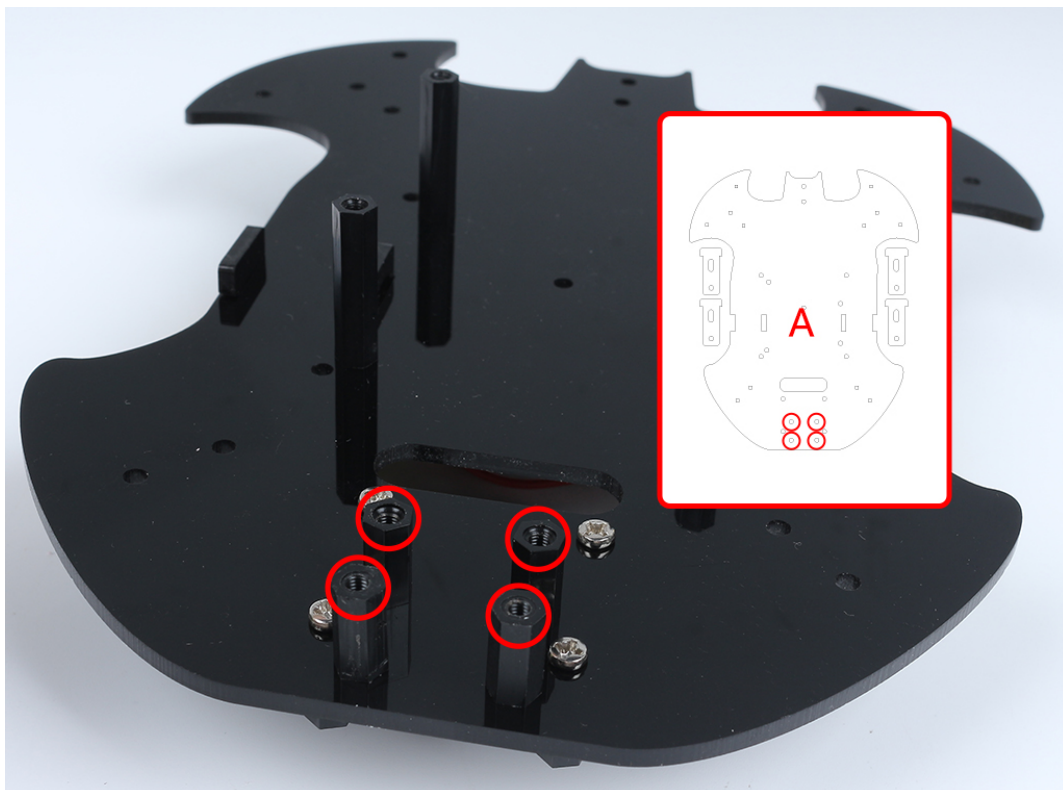
5. Montieren Sie einen weiteren TT-Motor. Beachten Sie dabei die Ausrichtung der Ausgangswelle und die Ausrichtung des Kabels.



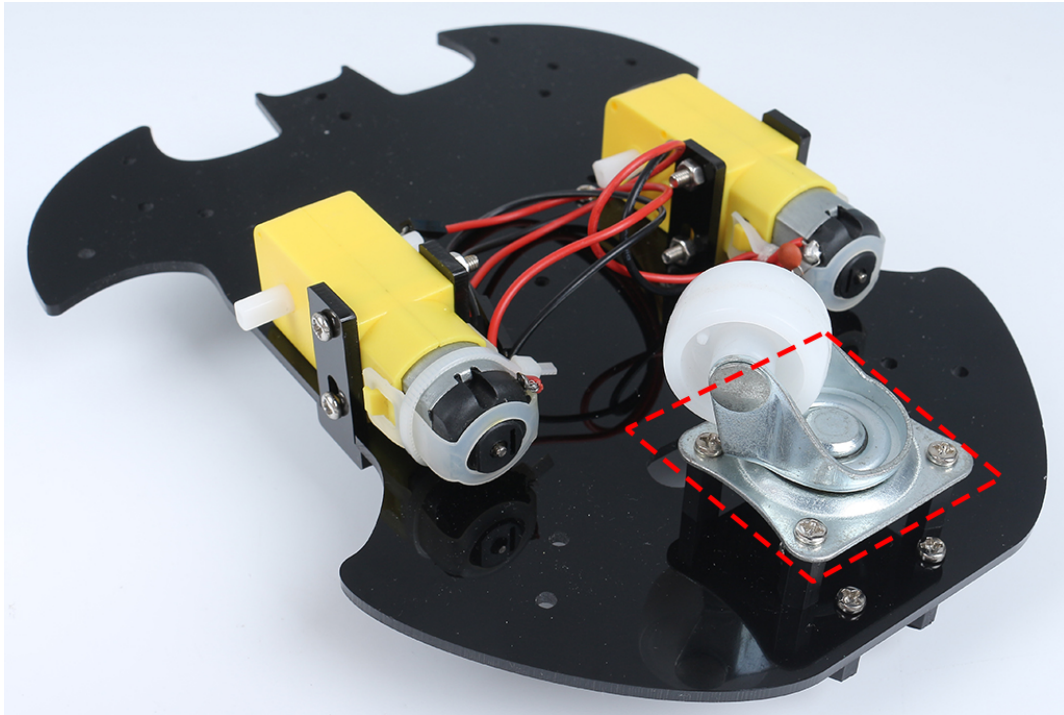
6. Verwenden Sie **M3x6mm Schrauben**, um den **M3x10mm Abstandshalter** an der unten gezeigten Position zu montieren.



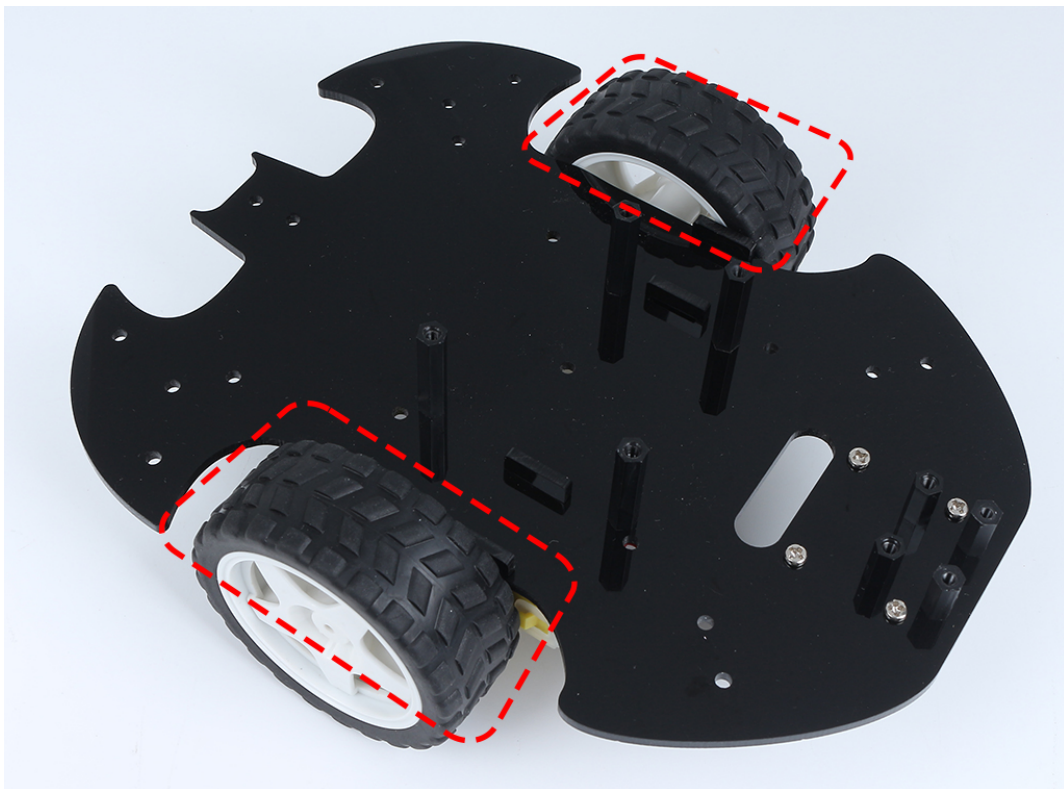
7. Befestigen Sie den **M2.5x11mm Abstandshalter** mit **M2.5x6mm Schrauben** an der Rückseite des Autos.



8. Verwenden Sie **M3x6mm Schrauben**, um das Universalrad zu montieren.



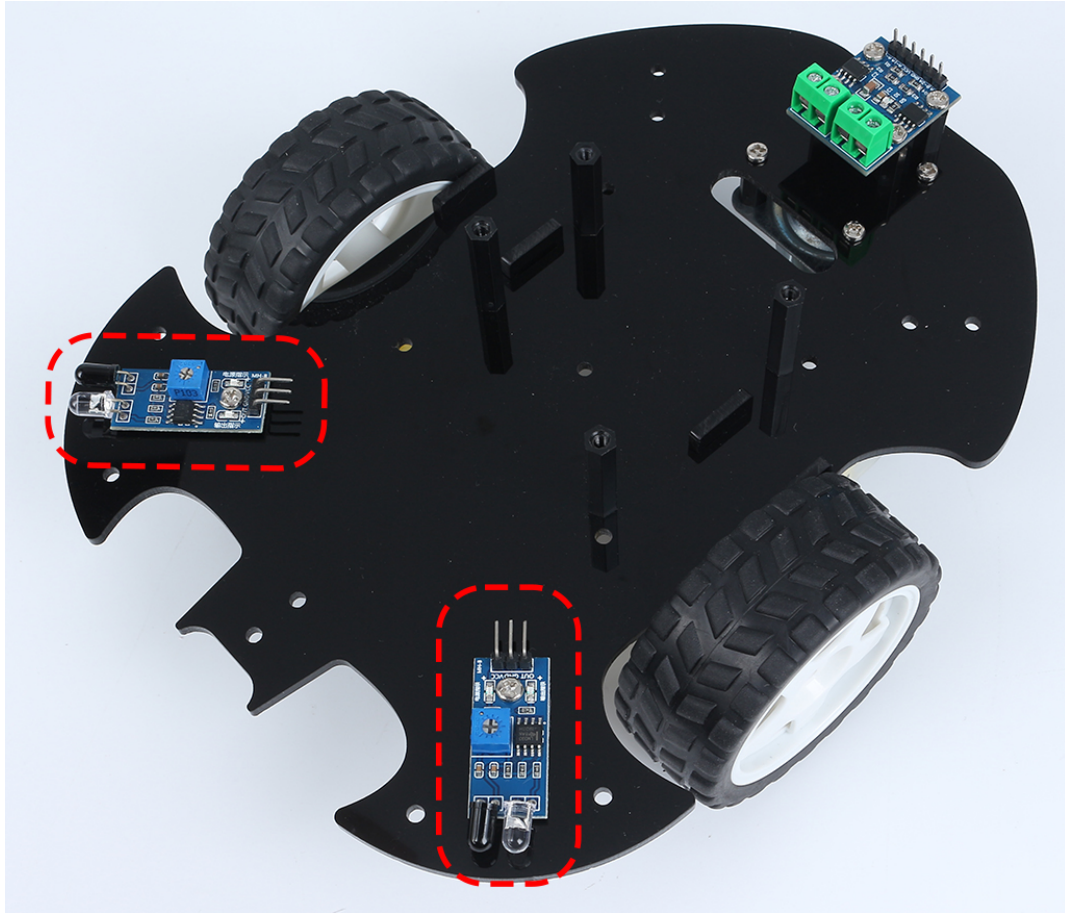
9. Montieren Sie die 2 Räder und die Grundstruktur des Autos ist fertiggestellt.



10. Befestigen Sie das L9110-Modul mit **M2.5x6mm Schrauben**.



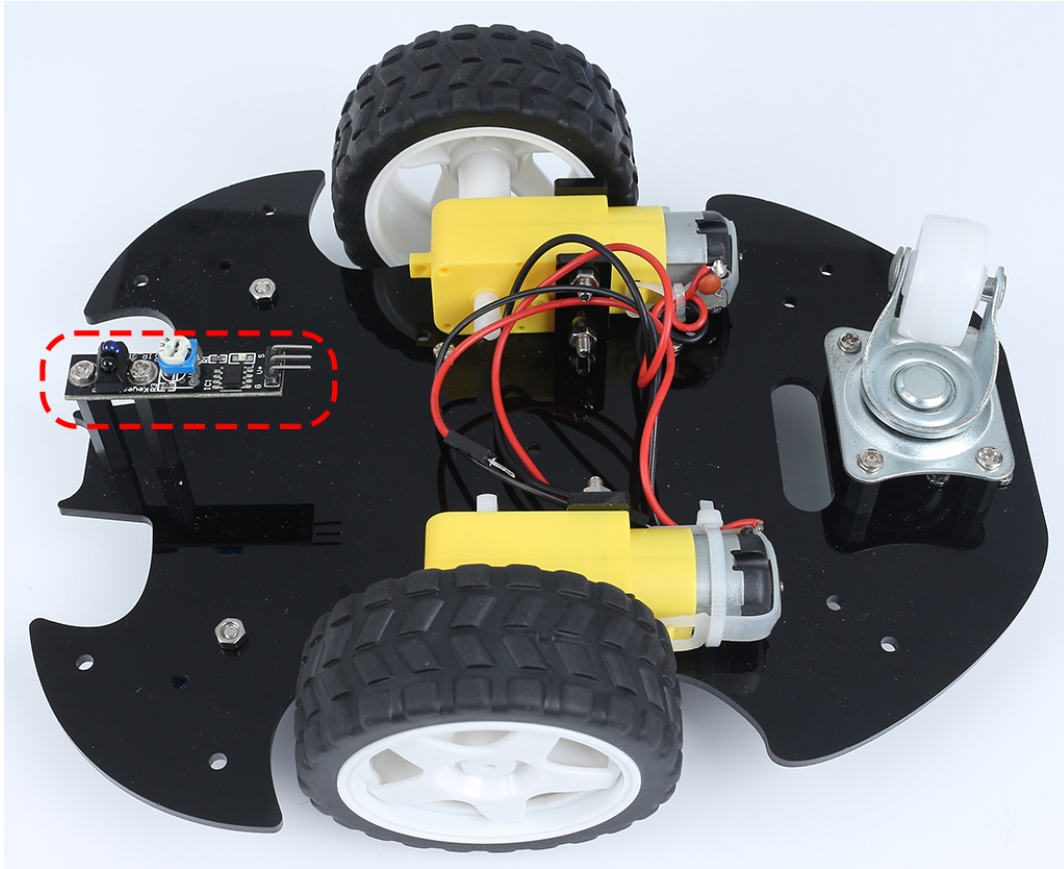
11. Montieren Sie die zwei IR-Hindernismodule mit **M3x10mm Schrauben** und **M3 Muttern**.



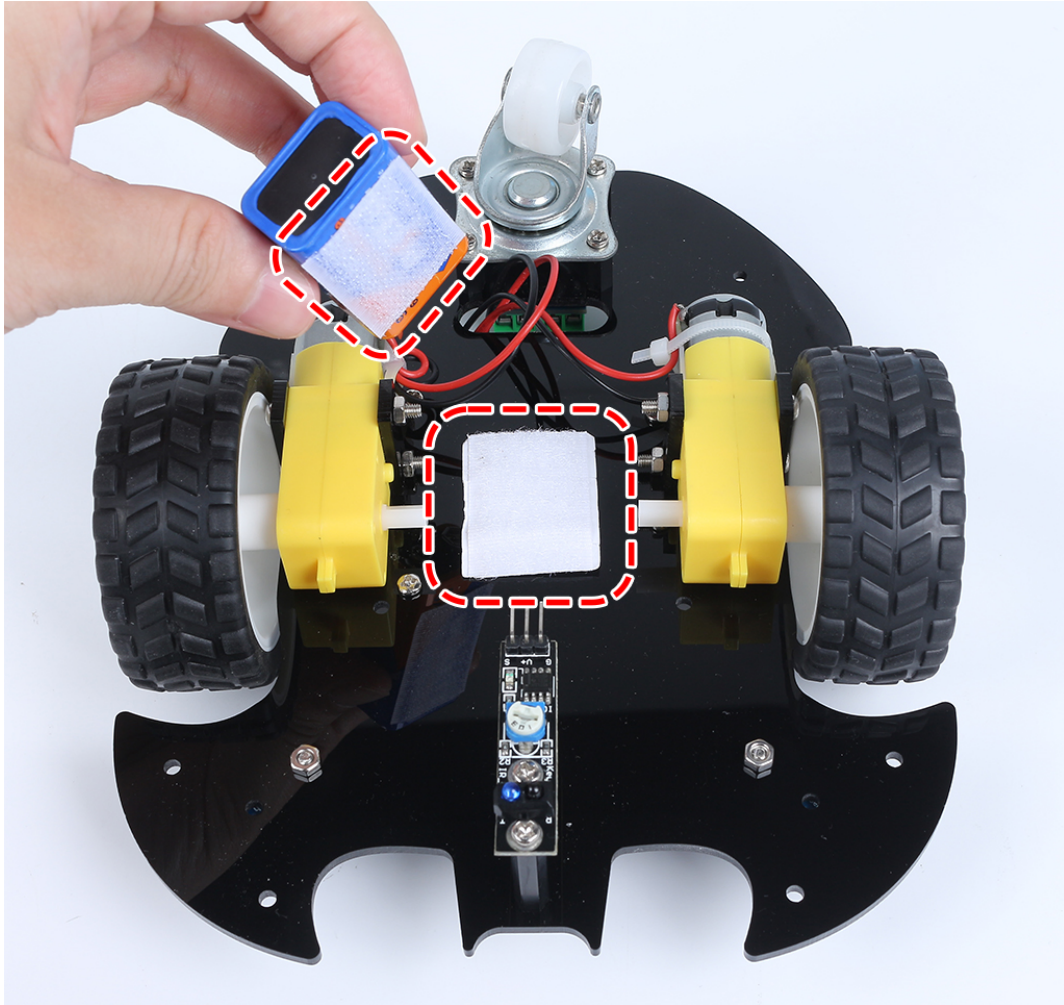
12. Drehen Sie zu Seite B und befestigen Sie das Linienverfolgungsmodul mit vier **M3x6mm Schrauben** und zwei **M3x24mm Abstandshaltern**.

Bemerkung: Es ist ratsam, zuerst die **M3x24mm Abstandshalter** am Linienverfolgungsmodul zu befestigen.

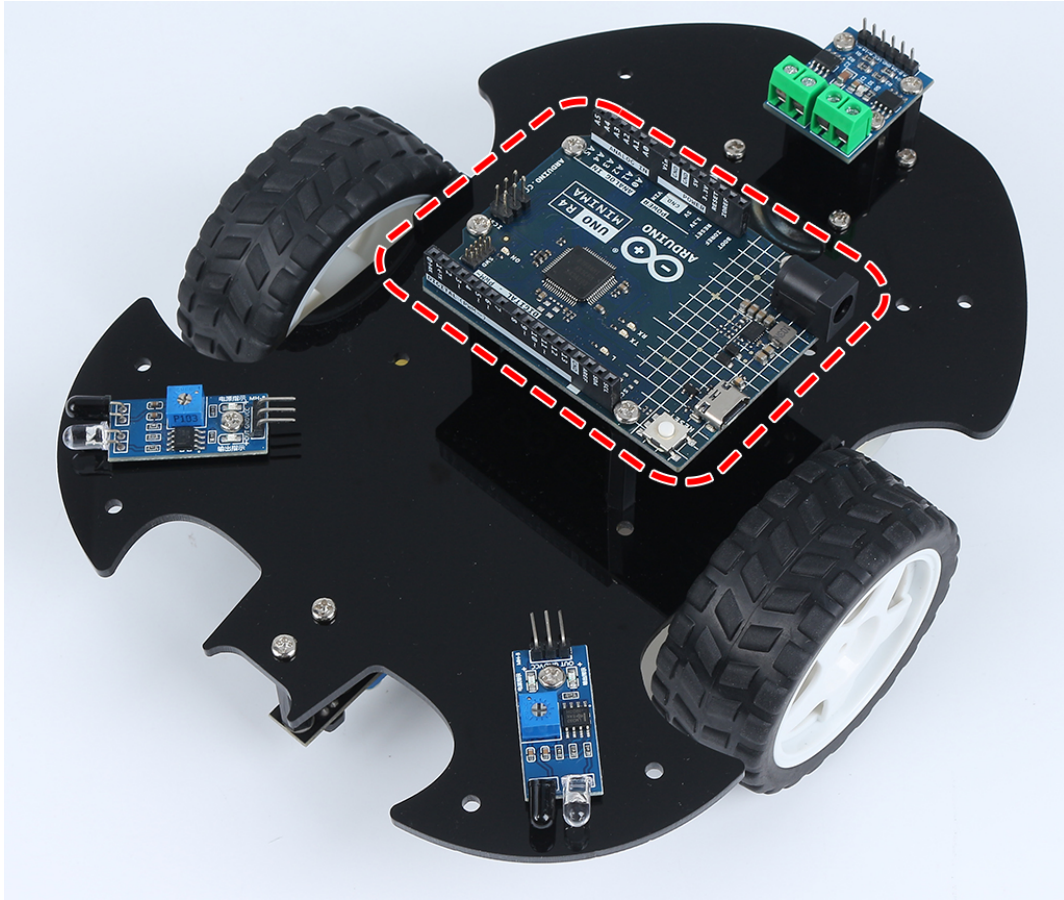
Ein wichtiger Hinweis: Die Pins des Liniensensors sind etwas weich und stehen etwas über die Löcher hinaus. Wenn Sie die **M3x24mm Abstandshalter** einschrauben, drücken Sie die Sensorkontakte vorsichtig beiseite.



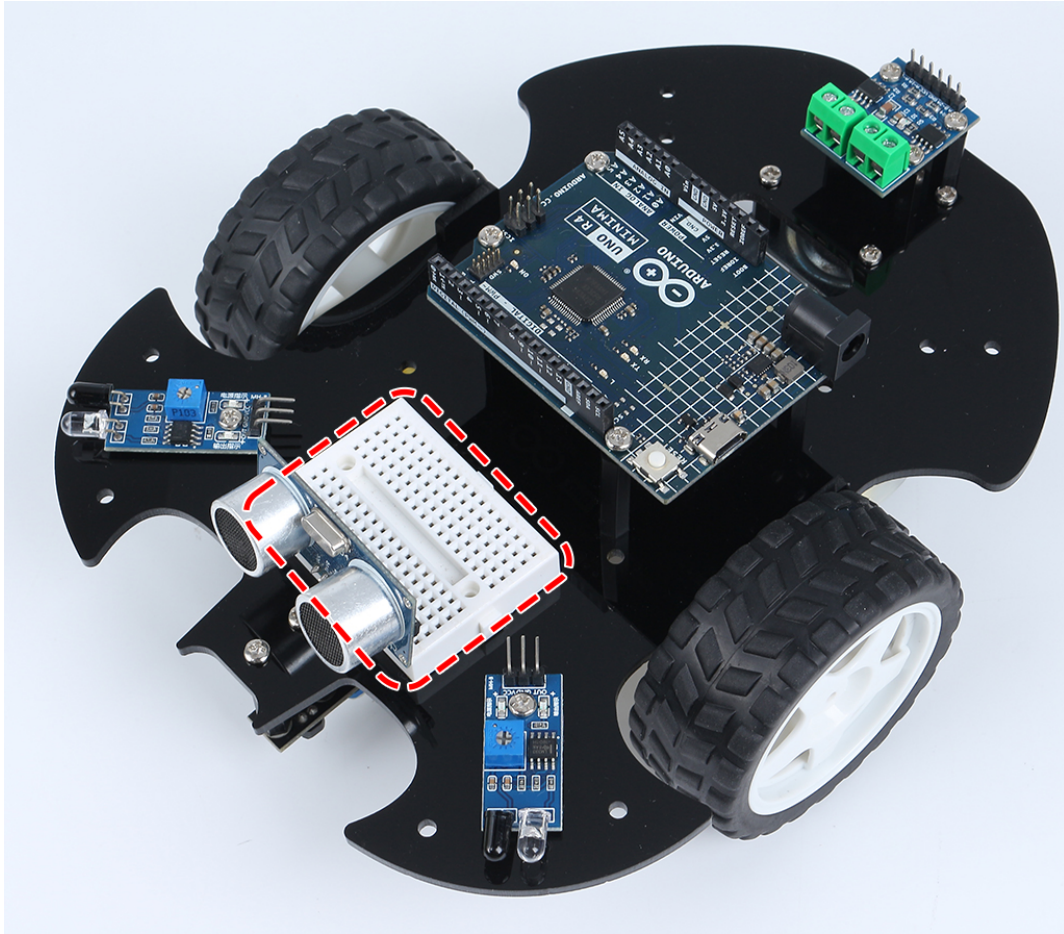
13. Befestigen Sie den Klettverschluss am 9V-Akku und setzen Sie den Batterieclip auf. Kleben Sie den anderen Teil des Klettverschlusses auf das Auto, um die Batterie zu sichern.



14. Drehen Sie zu Seite A und montieren Sie die R4-Platte mit **M3x6mm Schrauben**.



15. Befestigen Sie das Steckbrett vorne am Auto. Danach können Sie je nach Projektanforderung verschiedene Komponenten (z. B. Ultraschallmodul) auf das Steckbrett setzen.



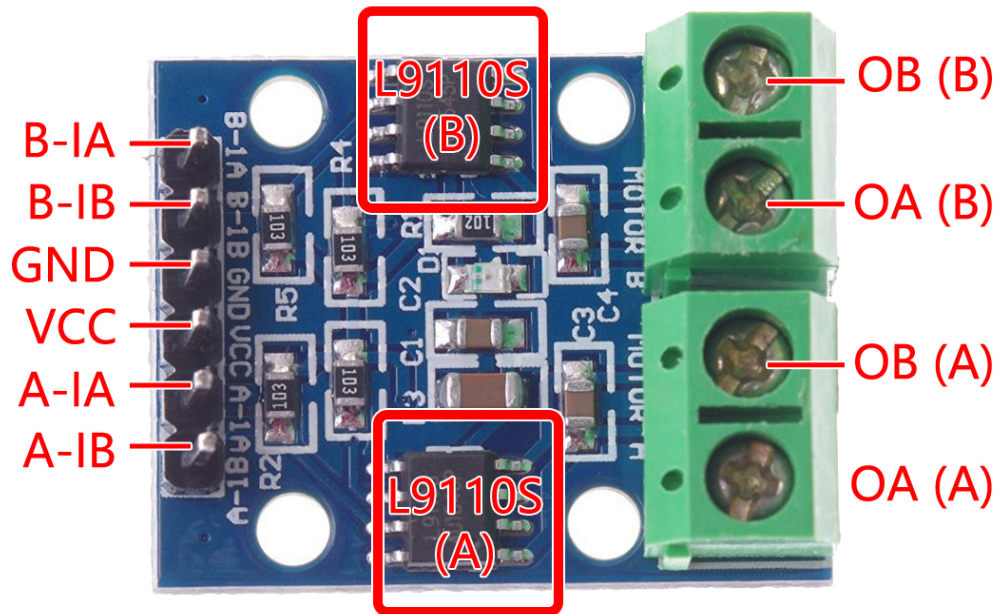
16. Um das Auto zum Laufen zu bringen, müssen Sie es auch verkabeln und einen Code schreiben, der in den folgenden Abschnitten verfasst wird.

Projekte

Hier sind einige Projekte für das Auto, programmiert in C mit der Arduino IDE. Wenn Sie noch nicht so geübt im Umgang mit Arduino sind, können Sie sich an [Mit Arduino loslegen](#) wenden.

Die folgenden Projekte sind nach dem Grad der Programmierschwierigkeit geordnet. Es wird empfohlen, diese in der angegebenen Reihenfolge zu lesen.

5.2 1. Bewegung



Bevor wir mit der Programmierung beginnen, betrachten wir das Funktionsprinzip des L9110-Moduls.

Hier ist die Wahrheitstabelle für Motor B:

B-1A	B-1B(B-2A)	Zustand des Motors B
1	0	Drehung im Uhrzeigersinn
0	1	Drehung gegen den Uhrzeigersinn
0	0	Bremse
1	1	Stopp

Hier ist die Wahrheitstabelle für Motor A:

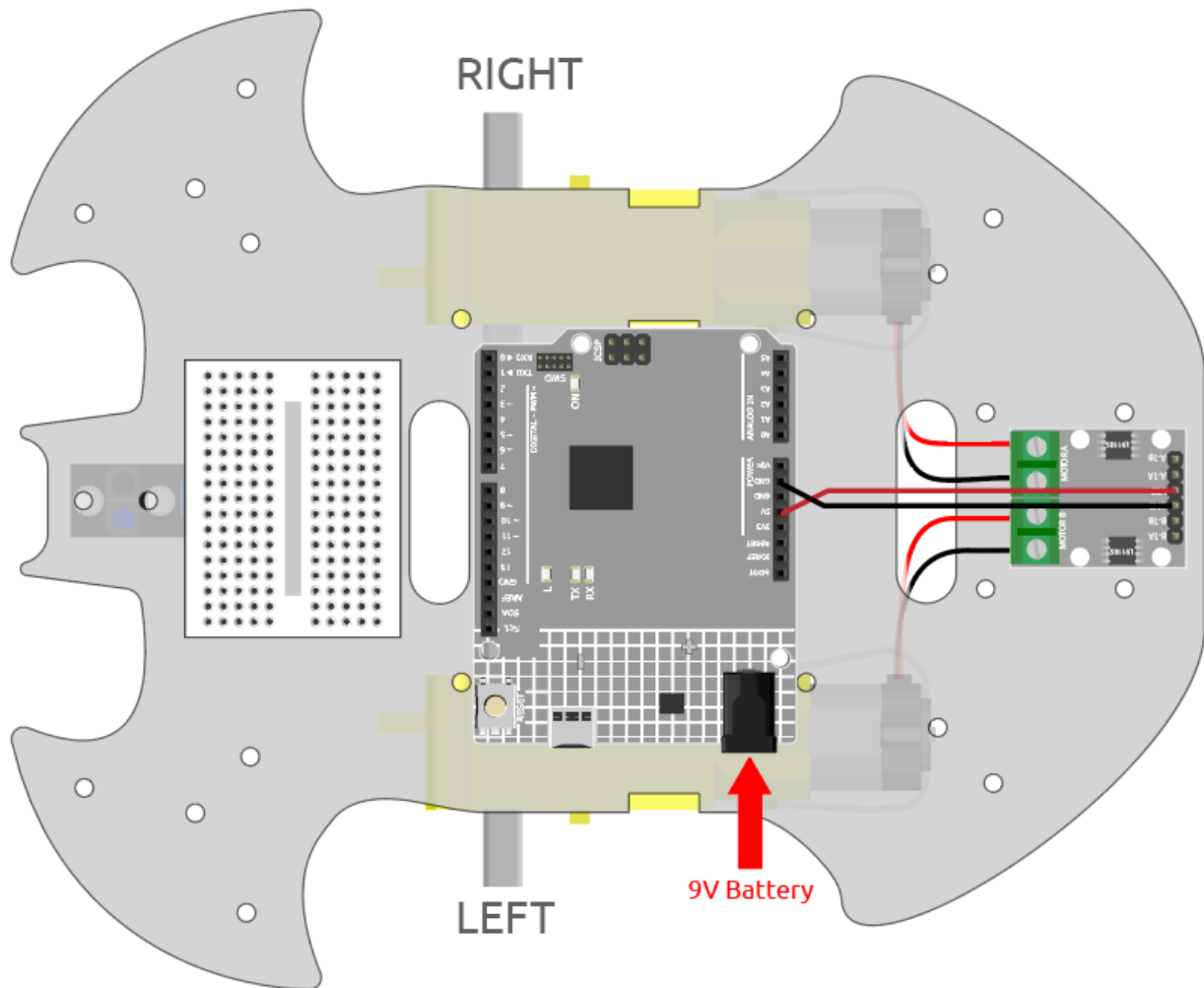
A-1A	A-1B	Zustand des Motors A
1	0	Drehung im Uhrzeigersinn
0	1	Drehung gegen den Uhrzeigersinn
0	0	Bremse
1	1	Stopp

- *L9110 Motortreibermodul*

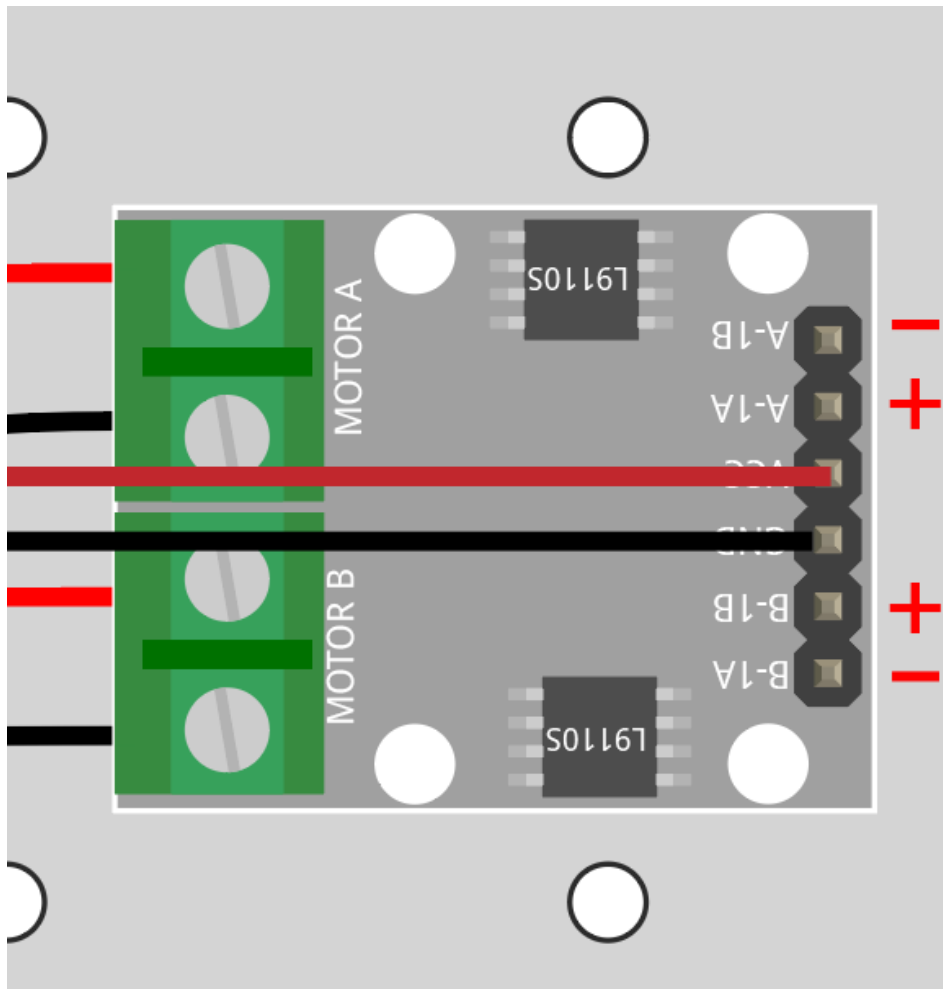
Vorwärts

Verbinden Sie nun den Eingang des L9110-Moduls direkt mit 12V und GND, um das Auto zum Bewegen zu bringen.

1. Verbinden Sie das R4-Board, das L9110-Modul und die beiden Motoren.



2. Verbinden Sie B-1B(B-2A) und A-1A mit VCC und B-1A und A-1B mit GND, dann sehen Sie, wie sich das Auto vorwärts bewegt.

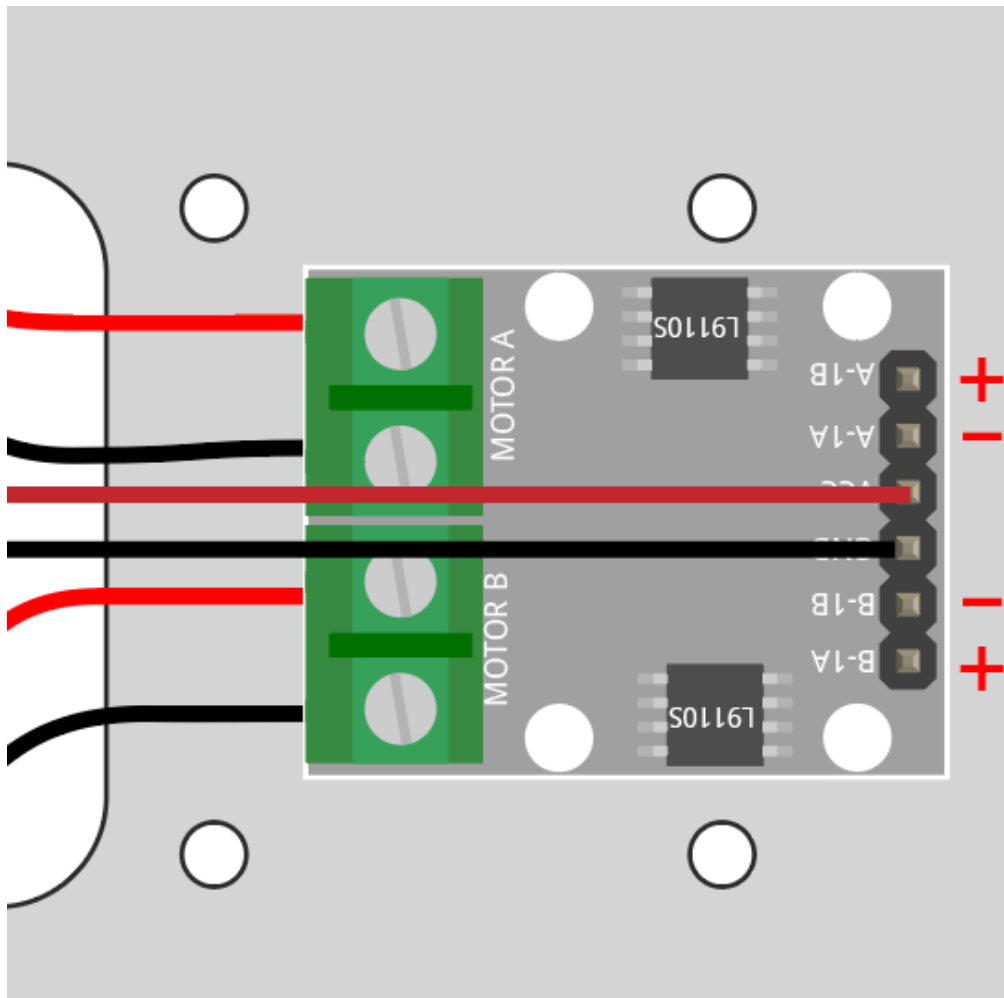


Wenn nicht beide nach vorne drehen, aber die folgenden Situationen auftreten, müssen Sie die Verkabelung der beiden Motoren neu justieren.

- Wenn beide Motoren gleichzeitig rückwärts drehen (linker Motor im Uhrzeigersinn, rechter Motor gegen den Uhrzeigersinn), tauschen Sie die Verkabelung der linken und rechten Motoren gleichzeitig. OA(A) und OB(A) tauschen, OA(B) und OB(B) tauschen.
- Wenn der linke Motor rückwärts dreht (im Uhrzeigersinn), tauschen Sie die Verkabelung von OA(B) und OB(B) des linken Motors.
- Wenn der rechte Motor rückwärts dreht (gegen den Uhrzeigersinn), tauschen Sie die Verkabelung von OA(A) und OB(A) des rechten Motors.

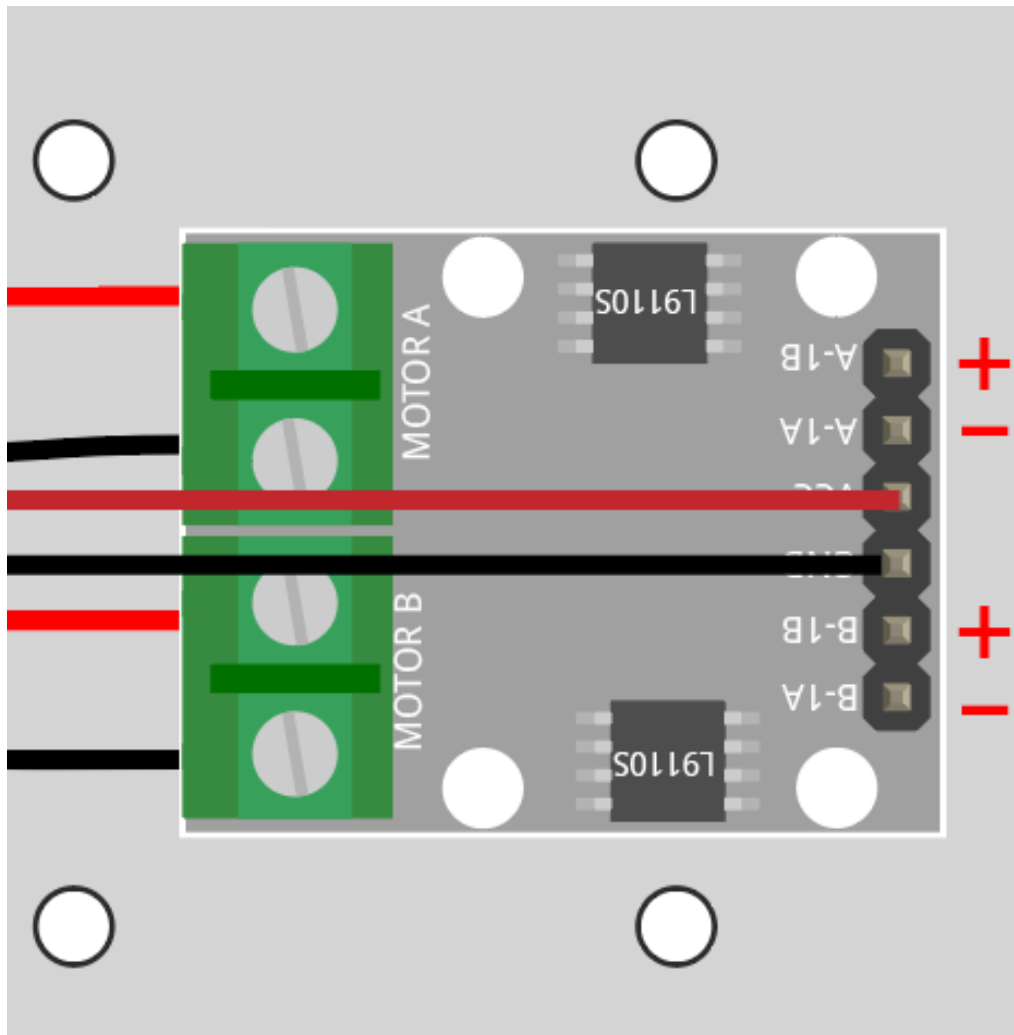
Rückwärts

Verbinden Sie B-1B(B-2A) und A-1A mit GND und B-1A und A-1B mit VCC, dann sehen Sie, wie sich das Auto rückwärts bewegt.



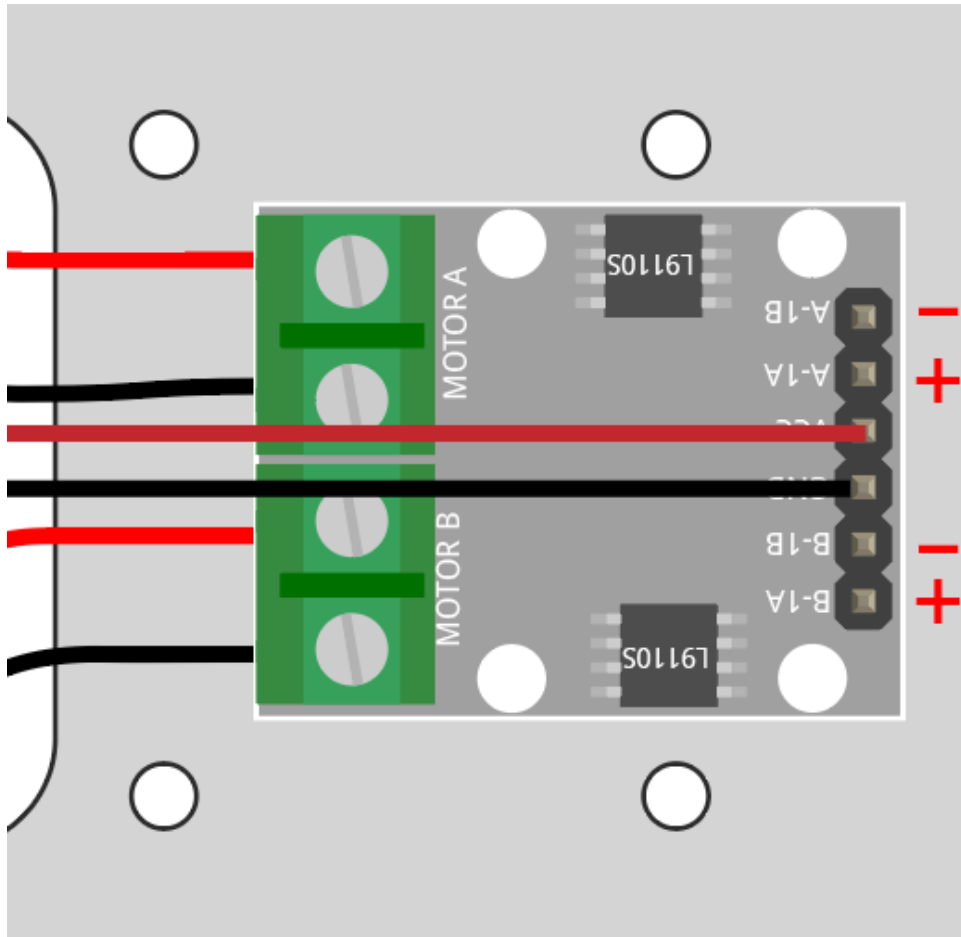
Nach Links abbiegen

Wenn Sie das Auto nach links abbiegen lassen möchten, also beide Motoren im Uhrzeigersinn drehen lassen möchten. Sie müssen B-1A und A-1A mit GND und B-1B(B-2A) und A-1B mit VCC verbinden.



Nach Rechts abbiegen

Umgekehrt, wenn Sie das Auto nach rechts abbiegen lassen möchten, also beide Motoren gegen den Uhrzeigersinn drehen lassen möchten. Sie müssen B-1A und A-1A mit VCC und B-1B(B-2A) und A-1B mit GND verbinden.



Stopp

Um den Motor anzuhalten, verbinden Sie die Eingänge auf der gleichen Seite gleichzeitig mit 12V oder GND, z.B. B-1A und B-1B(B-2A) gleichzeitig mit 12V oder 5V verbinden und dasselbe für A-1A und A-1B.

Dies ist natürlich theoretisch und wird später bei der Steuerung mit Code benötigt. Hier kann das Entfernen der Stromversorgung vom Auto es anhalten.

5.3 2. Steuerung durch Code

Im vorherigen Projekt haben wir versucht, den Betrieb des Motors zu steuern, indem wir für den Eingang des L9110-Moduls verschiedene Pegelsignale verwendet haben.

Wenn wir die Pegelsignale durch das Programm ändern, können wir die Bewegung des Autos flexibel steuern.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

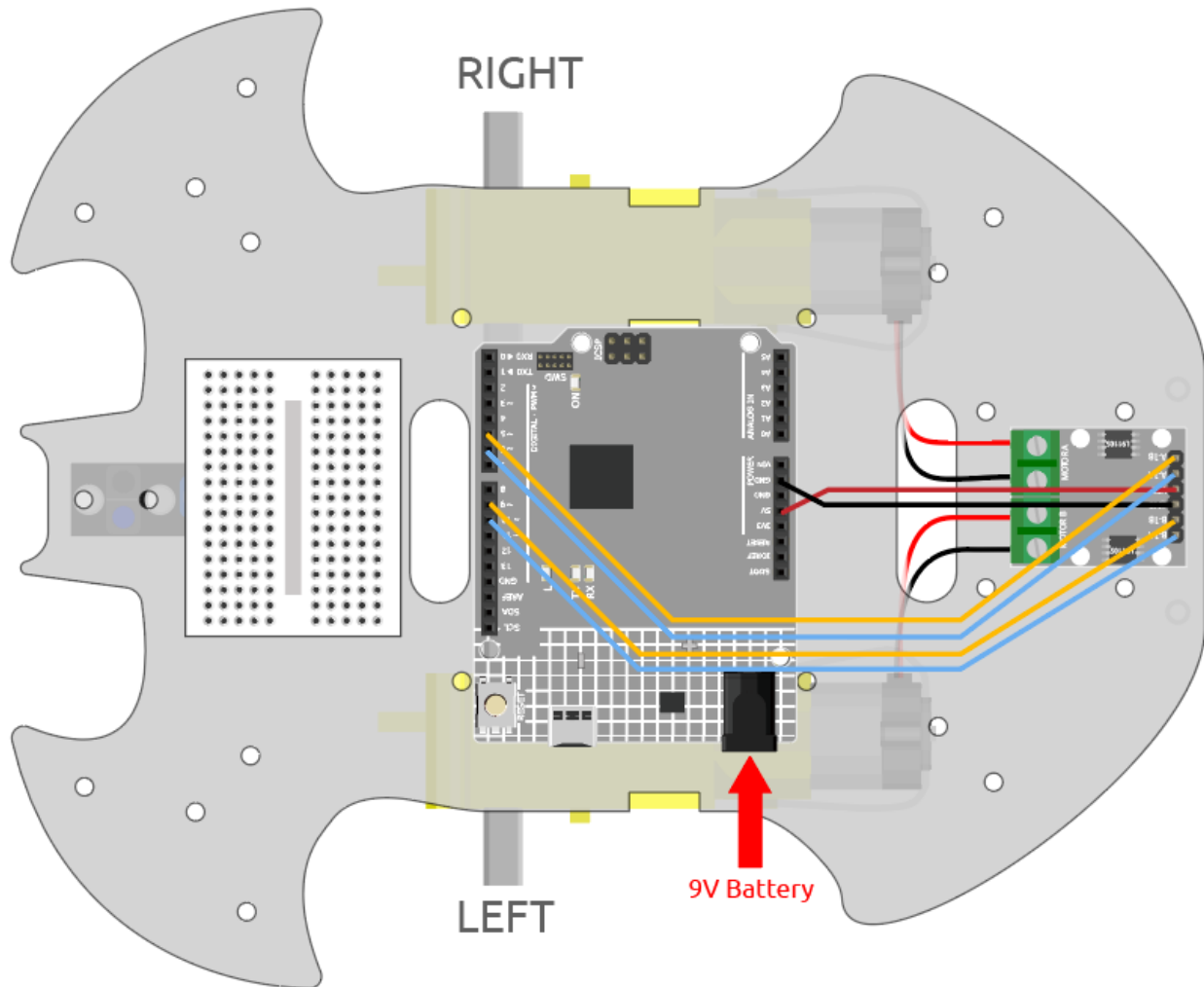
Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-

Verdrahtung

Verbinden Sie die Kabel zwischen dem L9110-Modul und dem R4-Board gemäß dem untenstehenden Diagramm.

L9110 Modul	R4 Board	Motor
A-1B	5	
A-1A	6	
B-1B(B-2A)	9	
B-1A	10	
OB(B)		Schwarzes Kabel des rechten Motors
OA(B)		Rotes Kabel des rechten Motors
OB(A)		Schwarzes Kabel des linken Motors
OA(A)		Rotes Kabel des linken Motors



Code

Bemerkung:

- Öffnen Sie die Datei 2.move.ino im Pfad 3in1-kit\car_project\2.move.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

Nach dem Hochladen des Codes wird das Auto jeweils für zwei Sekunden vorwärts, rückwärts, links und rechts fahren.

Wie funktioniert das?

Dieses Projekt entspricht im Wesentlichen dem vorherigen. Es beinhaltet das Vorwärts-, Rückwärts-, Links- und Rechtsfahren des Autos sowie das Anhalten, indem verschiedene Signalpegel an die Eingangspins des L9110-Moduls gesendet werden.

1. Initialisieren Sie die Pins des L9110-Moduls.

```
const int A_1B = 5;
const int A_1A = 6;
const int B_1B = 9;
const int B_1A = 10;
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

void setup() {
    pinMode(A_1B, OUTPUT);
    pinMode(A_1A, OUTPUT);
    pinMode(B_1B, OUTPUT);
    pinMode(B_1A, OUTPUT);
}

```

2. Legen Sie die Eingangspins auf unterschiedlich hohe oder niedrige Pegel, um die Drehung der linken und rechten Motoren zu steuern, und kapseln Sie sie in einzelne Funktionen.

```

void moveForward() {
    digitalWrite(A_1B, LOW);
    digitalWrite(A_1A, HIGH);
    digitalWrite(B_1B, HIGH);
    digitalWrite(B_1A, LOW);
}

void moveBackward() {
    digitalWrite(A_1B, HIGH);
    digitalWrite(A_1A, LOW);
    digitalWrite(B_1B, LOW);
    digitalWrite(B_1A, HIGH);
}

...

```

3. Rufen Sie diese Funktionen in loop() auf.

```

void loop() {
    moveForward();
    delay(2000);
    stopMove();
    delay(500);

    moveBackward();
    delay(2000);
    stopMove();
    delay(500);

    ...
}

```

- `digitalWrite(pin, value)`

- pin: die Arduino-Pin-Nummer.
- value: HIGH oder LOW.

Schreibt einen HIGH- oder LOW-Wert an einen digitalen Pin. Wenn der Pin mit `pinMode()` als OUTPUT konfiguriert wurde, wird seine Spannung auf den entsprechenden Wert gesetzt: 5V (oder 3,3V bei 3,3V-Boards) für HIGH, 0V (Masse) für LOW.

- `pinMode(pin, mode)`

- pin: die Arduino-Pin-Nummer, für die der Modus festgelegt werden soll.
- mode: INPUT, OUTPUT oder INPUT_PULLUP.

Konfiguriert den angegebenen Pin entweder als Eingang oder Ausgang.

- `delay(ms)`
 - ms: die Anzahl der Millisekunden zum Pausieren. Erlaubte Datentypen: unsigned long.
- Pausiert das Programm für die angegebene Zeit (in Millisekunden). (In einer Sekunde sind 1000 Millisekunden.)

5.4 3. Beschleunigung

Zusätzlich zum digitalen Signal (HIGH/LOW) kann der Eingang des L9110-Moduls auch ein PWM-Signal empfangen, um die Geschwindigkeit des Ausgangs zu steuern.

Das bedeutet, dass wir `AnalogWrite()` verwenden können, um die Fahrgeschwindigkeit des Autos zu steuern.

In diesem Projekt lassen wir das Auto seine Vorwärtsgeschwindigkeit schrittweise ändern, zuerst beschleunigend und dann verlangsamend.

Verdrahtung

Die Verkabelung für dieses Projekt ist identisch mit *2. Steuerung durch Code*.

Code

Bemerkung:

- Öffnen Sie die Datei `3.speed_up.ino` im Pfad `3in1-kit\car_project\3.speed_up`.
 - Oder kopieren Sie diesen Code in die **Arduino IDE**.
 - Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.
-

Nachdem das Programm läuft, wird das Auto schrittweise beschleunigen und dann allmählich abbremsen.

Wie funktioniert das?

Ziel dieses Projekts ist es, unterschiedliche PWM-Werte an die Eingangspins des L9110-Moduls zu schreiben, um die Vorwärtsgeschwindigkeit des Autos zu steuern.

1. Verwenden Sie die `for()`-Schleife, um `speed` in Schritten von 5 zu setzen, schreibend Werte von 0 bis 255, damit Sie die Veränderung der Vorwärtsgeschwindigkeit des Autos sehen können.

```
void loop() {  
    for(int i=0;i<=255;i+=5){  
        moveForward(i);  
        delay(500);  
    }  
    for(int i=255;i>=0;i-=5){  
        moveForward(i);  
        delay(500);  
    }  
}
```

2. Über die `moveForward()` Funktion.

Im Gegensatz zu *2. Steuerung durch Code*, das direkt hohe/niedrige Pegel an die Eingangspins des L9110-Moduls gibt, übergeben wir hier einen Parameter `speed`, dort, wo wir hohe Pegel geben müssen.

```
void moveForward(int speed) {
    analogWrite(A_1B, 0);
    analogWrite(A_1A, speed);
    analogWrite(B_1B, speed);
    analogWrite(B_1A, 0);
}
```

- for

Die for-Anweisung wird verwendet, um einen Anweisungsblock in geschweiften Klammern zu wiederholen. Ein Inkrementzähler wird in der Regel verwendet, um die Schleife zu inkrementieren und zu beenden.

```
for (initialization; condition; increment) {
    // statement(s);
}
```

- **initialization:** findet zuerst und genau einmal statt.
- **condition:** bei jedem Durchlauf der Schleife wird condition getestet; wenn es wahr ist, wird der Anweisungsblock und das Inkrement ausgeführt, dann wird die condition erneut getestet. Wenn die condition falsch wird, endet die Schleife.
- **increment:** wird bei jedem Durchlauf der Schleife ausgeführt, wenn condition wahr ist.

5.5 4. Der Linie folgen

Das Auto ist mit einem Linienverfolgungsmodul ausgestattet, mit dem es der schwarzen Linie folgen kann.

Wenn das Linienverfolgungsmodul die schwarze Linie erkennt, dreht sich der rechte Motor, während der linke Motor stillsteht, sodass das Auto einen Schritt nach links vorne macht. Während das Auto sich bewegt und das Modul von der Linie wegrückt, dreht sich der linke Motor und der rechte Motor bleibt stehen. Das Auto wird einen Schritt nach rechts machen, um zur Linie zurückzukehren. Wiederholen Sie die oben genannten Schritte, kann das Auto entlang der schwarzen Linie fahren.

Bevor Sie mit dem Projekt beginnen, müssen Sie eine Kurvenkarte mit schwarzem Klebeband erstellen. Die empfohlene Linienbreite liegt zwischen 0,8-1,5 cm und der Winkel der Kurve sollte nicht weniger als 90 Grad betragen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Kit zu kaufen. Hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

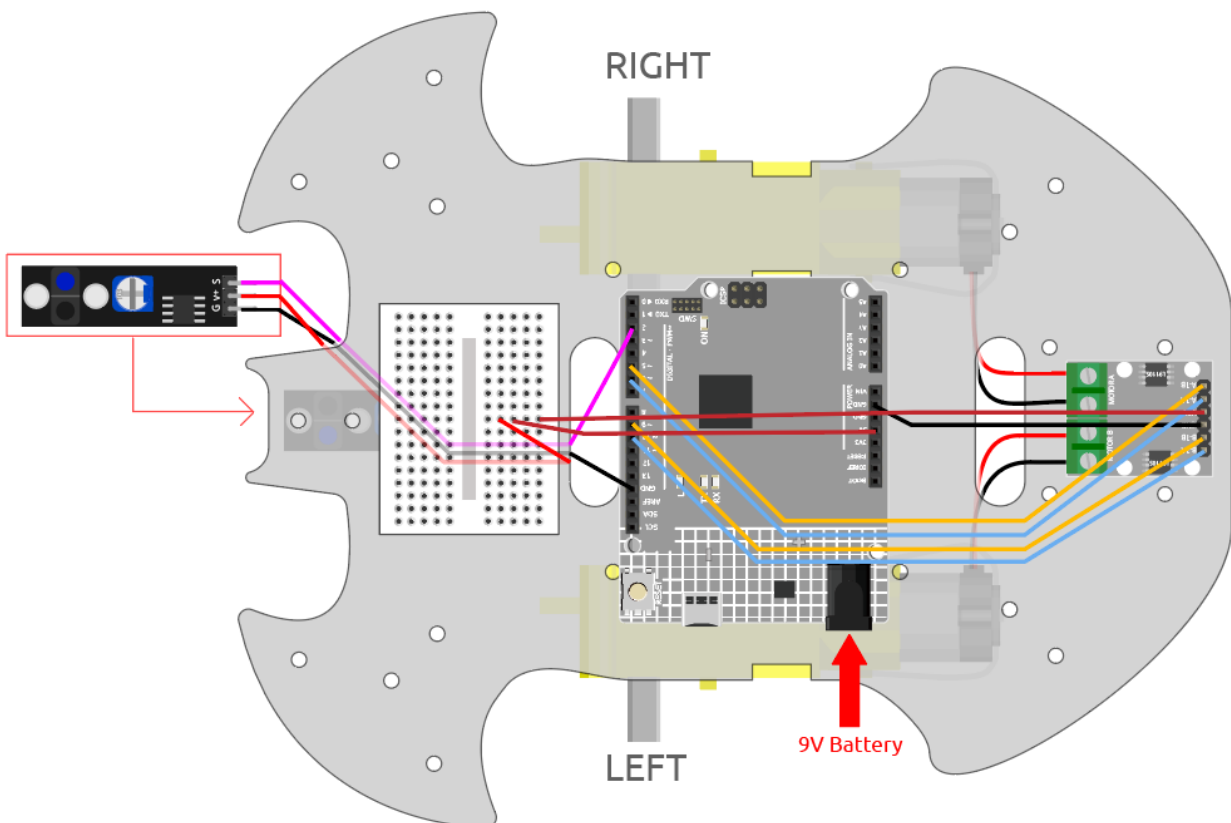
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Linienverfolgungsmodul</i>	

Verdrahtung

Dies ist ein digitales Linienverfolgungsmodul. Wenn eine schwarze Linie erkannt wird, gibt es 1 aus; bei Erkennung einer weißen Linie gibt es den Wert 0 aus. Darüber hinaus können Sie den Erfassungsabstand über das Potentiometer am Modul einstellen.

Bauen Sie den Schaltkreis gemäß dem folgenden Diagramm.

Linienverfolgungsmodul	R4-Platine
S	2
V+	5V
G	GND



Das Modul einstellen

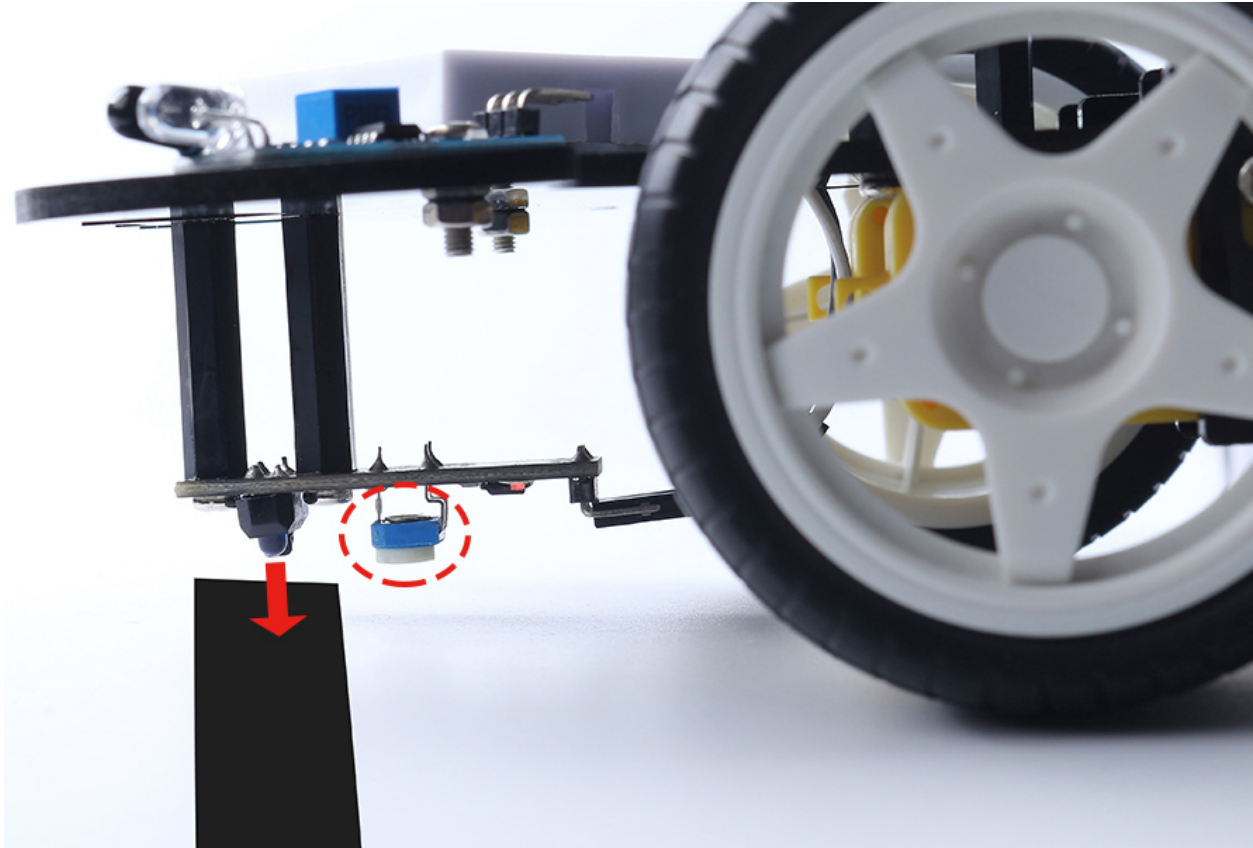
Bevor Sie mit dem Projekt beginnen, müssen Sie die Empfindlichkeit des Moduls einstellen.

Verkabeln Sie gemäß dem obigen Diagramm und versorgen Sie dann die R4-Platine mit Strom (entweder direkt über das USB-Kabel oder das 9V Batteriekabel), ohne den Code hochzuladen.

Kleben Sie ein schwarzes Isolierband auf den Tisch und stellen Sie den Wagen darauf.

Beobachten Sie die Signal-LED am Modul, um sicherzustellen, dass sie auf dem weißen Tisch leuchtet und auf dem schwarzen Band erlischt.

Wenn nicht, müssen Sie das Potentiometer am Modul so einstellen, dass es den oben beschriebenen Effekt erzielen kann.



Code

Bemerkung:

- Öffnen Sie die Datei `4.follow_the_line.ino` im Pfad `3in1-kit\car_project\4.follow_the_line`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nachdem Sie den Code auf das R4-Board geladen haben, richten Sie das Linienverfolgungsmodul unter dem Auto an der schwarzen Linie aus, und Sie werden sehen, dass das Auto der Linie folgt.

Wie funktioniert das?

In diesem Code werden die beiden Motoren entsprechend dem Wert des Linienverfolgungsmoduls mikrorotiert, sodass Sie sehen können, wie das Auto der schwarzen Linie folgt.

1. Fügen Sie die Pin-Definition für das Linienverfolgungsmodul hinzu. Hier ist es auf `INPUT` eingestellt. Initialisieren Sie hier auch den Seriellen Monitor und setzen Sie die Baudrate auf 9600bps.

```

...
const int lineTrack = 2;
Serial.begin(9600);
void setup() {
    ...
    pinMode(lineTrack, INPUT);
}

```

2. Lesen Sie den Wert des Linienverfolgungsmoduls aus. Wenn dieser 1 ist, lassen Sie das Auto nach links vorwärts fahren; ansonsten fahren Sie nach rechts vorwärts. Sie können auch den Seriellen Monitor öffnen, indem Sie auf das Lupensymbol in der oberen rechten Ecke klicken, um die Änderung des Linienverfolgungsmodulwertes auf der schwarz-weißen Linie zu sehen, bevor Sie das USB-Kabel ausstecken.

```

void loop() {

    int speed = 150;

    int lineColor = digitalRead(lineTrack); // 0:white    1:black
    Serial.println(lineColor);
    if (lineColor) {
        moveLeft(speed);
    } else {
        moveRight(speed);
    }
}

```

3. Über die Funktionen `moveLeft()` und `moveRight()`.

Anders als die Links-Rechts-Drehfunktion im Projekt 2. *Steuerung durch Code* sind hier nur kleine Links-Rechts-Drehungen erforderlich. Daher müssen Sie nur den Wert von A_1A oder B_1B jedes Mal anpassen. Zum Beispiel, wenn Sie sich nach links vorne bewegen (`moveLeft()`), müssen Sie nur die Geschwindigkeit auf A_1A setzen und alle anderen auf 0. Dies bewirkt, dass der rechte Motor im Uhrzeigersinn dreht und der linke Motor sich nicht bewegt.

```

void moveLeft(int speed) {
    analogWrite(A_1B, 0);
    analogWrite(A_1A, speed);
    analogWrite(B_1B, 0);
    analogWrite(B_1A, 0);
}

void moveRight(int speed) {
    analogWrite(A_1B, 0);
    analogWrite(A_1A, 0);
    analogWrite(B_1B, speed);
    analogWrite(B_1A, 0);
}

```

- **Serial**

Wird zur Kommunikation zwischen dem Arduino-Board und einem Computer oder anderen Geräten verwendet.

- `Serial.begin()`: Legt die Datenrate in Bits pro Sekunde (Baud) für die serielle Datenübertragung fest.

- `Serial.println()`: Gibt Daten als menschenlesbaren ASCII-Text an den seriellen Port weiter, gefolgt von einem Car-Return-Zeichen (ASCII 13 oder ,r') und einem Newline-Zeichen (ASCII 10 oder ,n').

- `if else`

Das `if else` ermöglicht im Vergleich zur einfachen `if`-Anweisung eine größere Kontrolle über den Codefluss, da mehrere Tests gruppiert werden können.

5.6 5. Spielen mit dem Hindernisvermeidungs-Modul

Zwei Infrarot-Hindernisvermeidungsmodule sind an der Vorderseite des Autos angebracht, mit denen nahegelegene Hindernisse erkannt werden können.

In diesem Projekt kann das Auto frei vorwärts fahren. Wenn es auf ein Hindernis trifft, kann es dieses vermeiden und in andere Richtungen weiterfahren.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist sicherlich praktisch, ein komplettes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch über die untenstehenden Links einzeln kaufen.

KOMPONENTENEINFÜHRUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Hindernisvermeidungsmodul</i>	

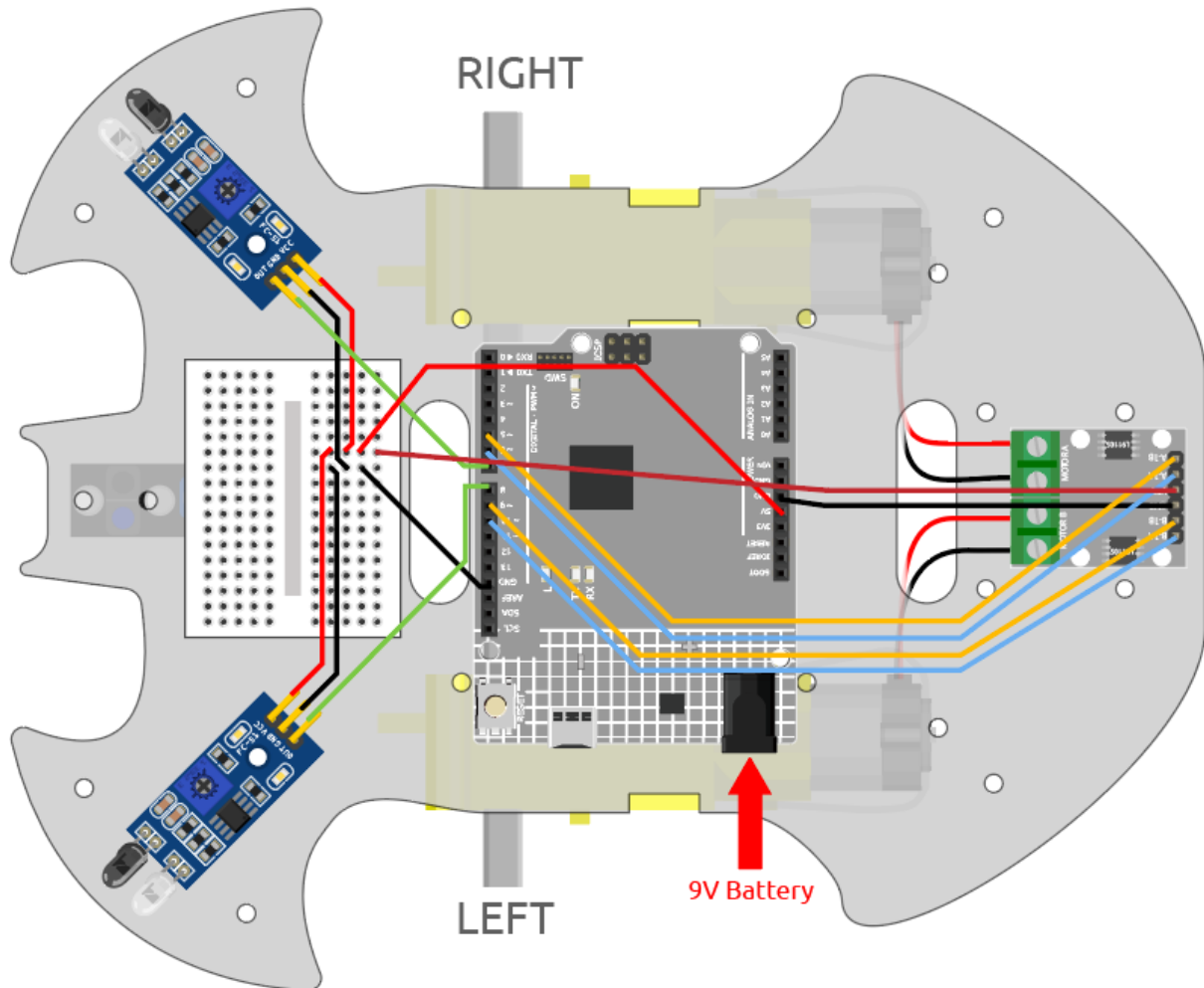
Verdrahtung

Das Hindernisvermeidungsmodul ist ein infraroter Näherungssensor mit einstellbarer Entfernung. Sein Ausgang ist normalerweise hoch und wird niedrig, wenn ein Hindernis erkannt wird.

Bauen Sie den Schaltkreis entsprechend dem untenstehenden Diagramm auf.

Linkes IR-Modul	R4 Board
AUSGANG	8
ERDE	GND
VCC	5V

Rechtes IR-Modul	R4 Board
AUSGANG	7
ERDE	GND
VCC	5V



Modul einstellen

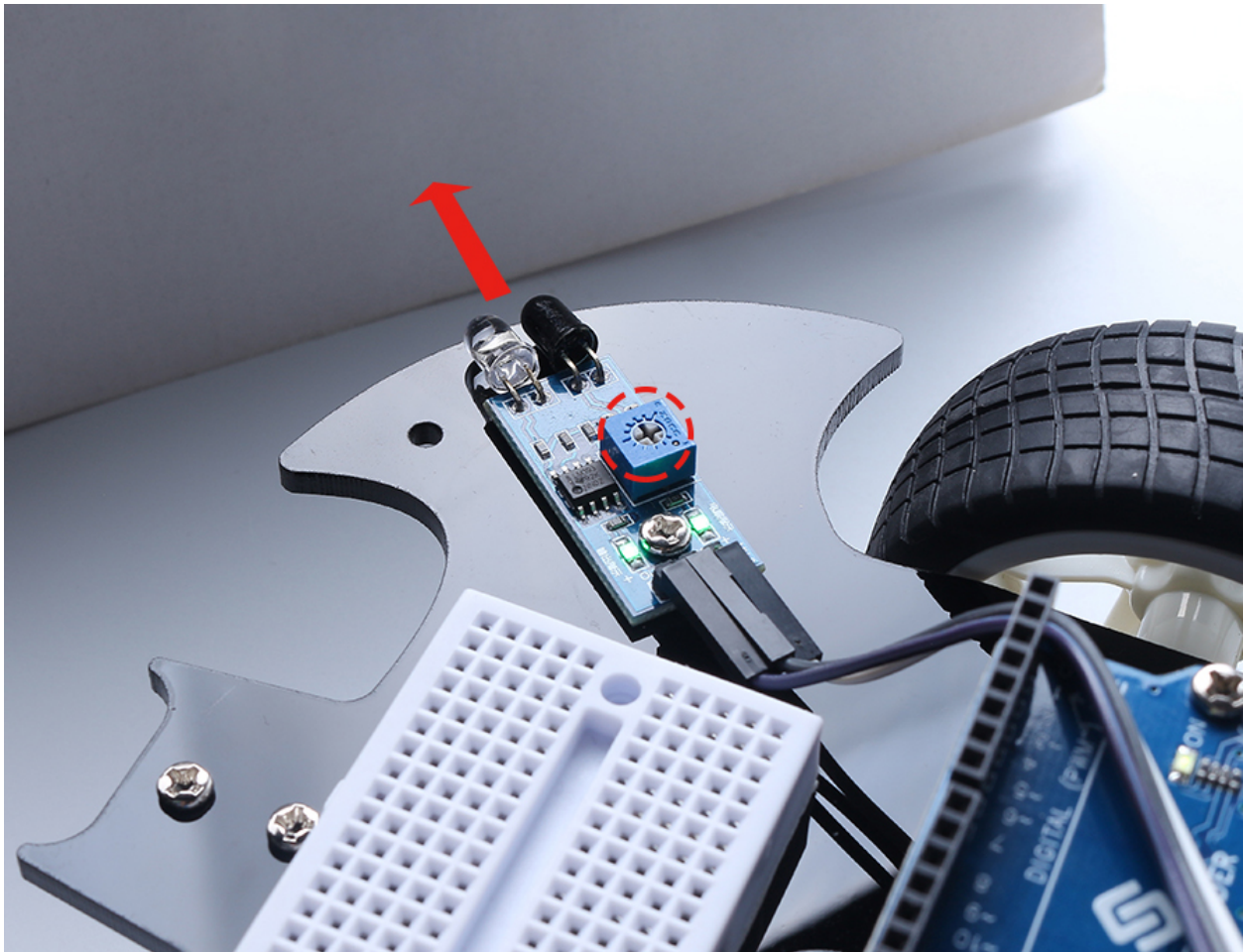
Bevor Sie mit dem Projekt beginnen, müssen Sie den Erfassungsabstand des Moduls anpassen.

Verkabeln Sie gemäß dem obigen Schema, schalten Sie das R4-Board ein (entweder indem Sie das USB-Kabel direkt einstecken oder das 9V-Batteriekabel anschließen), ohne den Code hochzuladen.

Stellen Sie ein Notizbuch oder ein anderes flaches Objekt etwa 5 cm vor das IR-Hindernisvermeidungsmodul.

Verwenden Sie dann einen Schraubenzieher, um das Potentiometer auf dem Modul so zu drehen, dass die Signalanzeige auf dem Modul gerade aufleuchtet, um seinen maximalen Erfassungsabstand von 5 cm einzustellen.

Verwenden Sie die gleiche Methode, um ein weiteres Infrarotmodul anzupassen.



Code

Bemerkung:

- Öffnen Sie die Datei 5.obstacle_avoidance_module.ino im Pfad 3in1-kit\car_project\5.obstacle_avoidance_module.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Das Auto wird vorwärts fahren, sobald der Code erfolgreich hochgeladen wurde. Wenn das linke Infrarot-Modul ein Hindernis erkennt, fährt es rückwärts nach links; erkennt das rechte Infrarot-Modul ein Hindernis, fährt es rückwärts nach rechts; erkennen beide Seiten ein Hindernis, fährt es geradlinig rückwärts.

Wie funktioniert das?

Dieses Projekt basiert auf den Werten der linken und rechten Infrarot-Hindernisvermeidungsmodule, um das Auto die entsprechende Aktion ausführen zu lassen.

1. Fügen Sie die Pin-Definition für die 2 Hindernisvermeidungsmodule hinzu, hier sind sie auf INPUT eingestellt.

```
...
const int rightIR = 7;
const int leftIR = 8;
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
void setup() {  
  ...  
  
  //IR obstacle  
  pinMode(leftIR, INPUT);  
  pinMode(rightIR, INPUT);  
}
```

2. Lesen Sie die Werte der linken und rechten Infrarotmodule und lassen Sie das Auto die entsprechende Aktion ausführen.

```
void loop() {  
  
  int left = digitalRead(leftIR);  // 0: Obstructed  1: Empty  
  int right = digitalRead(rightIR);  
  int speed = 150;  
  
  if (!left && right) {  
    backLeft(speed);  
  } else if (left && !right) {  
    backRight(speed);  
  } else if (!left && !right) {  
    moveBackward(speed);  
  } else {  
    moveForward(speed);  
  }  
}
```

- Wenn das linke IR-Modul 0 ist (Hindernis erkannt) und das rechte IR-Modul 1 ist, fährt das Auto rückwärts nach links.
- Wenn das rechte IR-Modul 0 ist (Hindernis erkannt), fährt das Auto rückwärts nach rechts.
- Wenn beide IR-Module gleichzeitig ein Hindernis erkennen, fährt das Auto rückwärts.
- Ansonsten wird das Auto weiter vorwärts fahren.

3. Über die Funktion `backLeft()`.

Wenn der rechte Motor gegen den Uhrzeigersinn dreht und der linke Motor nicht dreht, fährt das Auto rückwärts nach links.

```
void backLeft(int speed) {  
  analogWrite(A_1B, speed);  
  analogWrite(A_1A, 0);  
  analogWrite(B_1B, 0);  
  analogWrite(B_1A, 0);  
}
```

4. Über die Funktion `backRight()`.

Wenn der linke Motor im Uhrzeigersinn dreht und der rechte Motor nicht dreht, fährt das Auto rückwärts nach rechts.


```
void backRight(int speed) {
  analogWrite(A_1B, 0);
  analogWrite(A_1A, 0);
  analogWrite(B_1B, 0);
  analogWrite(B_1A, speed);
}
```

- &&: Logisches UND ergibt nur dann wahr, wenn beide Operanden wahr sind.
- !: Logisches NICHT ergibt wahr, wenn der Operand falsch ist und umgekehrt.

5.7 6. Mit dem Ultraschallmodul spielen

Im Projekt 5. *Spiele mit dem Hindernisvermeidungs-Modul* werden die 2 Infrarot-Hindernisvermeidungsmodule zur Hindernisvermeidung eingesetzt. Jedoch ist die Erkennungsreichweite des IR-Hindernisvermeidungsmoduls kurz, wodurch das Auto möglicherweise zu spät ausweicht.

In diesem Projekt verwenden wir ein Ultraschallmodul für Fernerkennungen, damit das Auto Hindernisse aus größerer Entfernung wahrnehmen und eine Entscheidung treffen kann.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

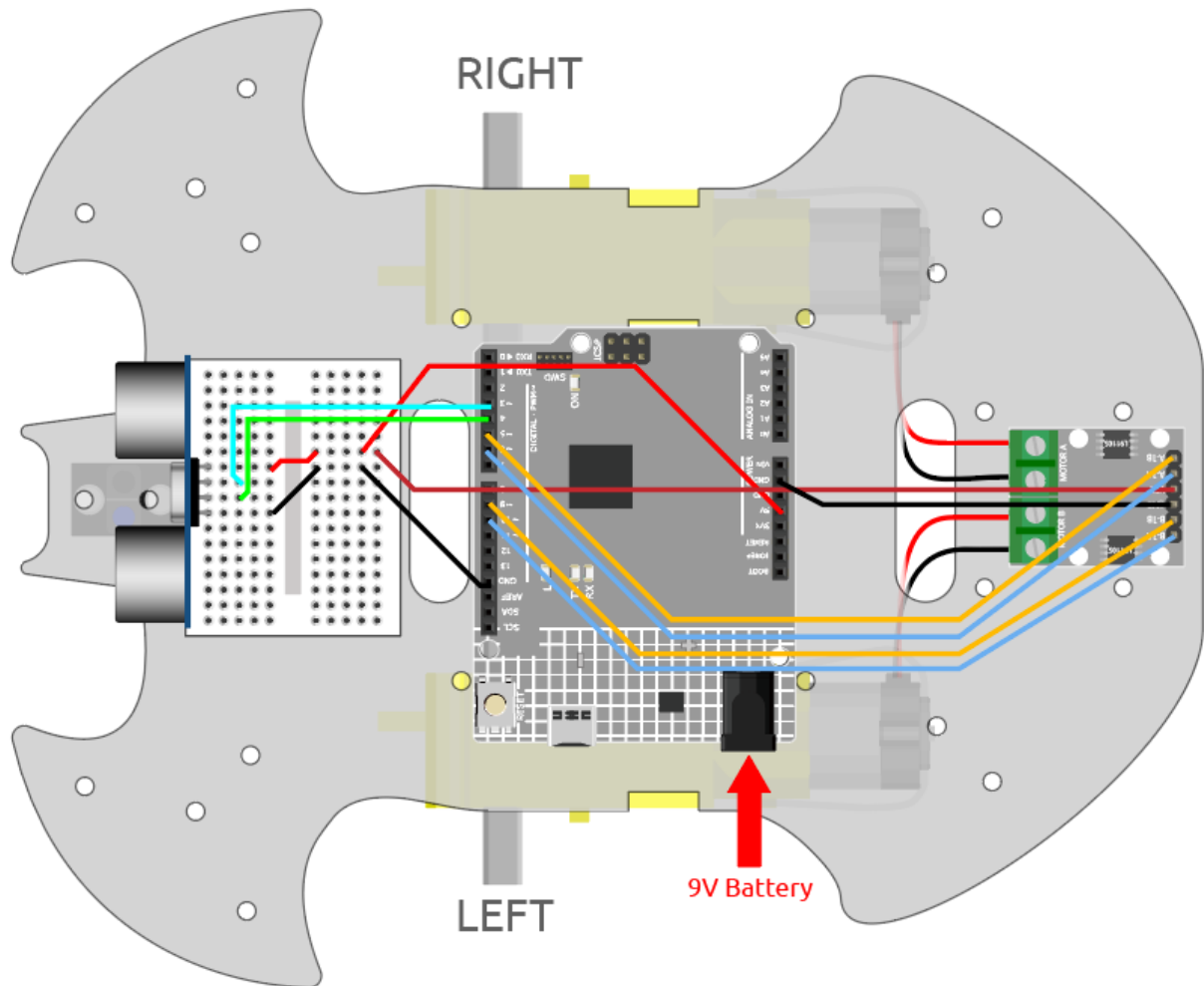
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Ultraschall-Modul</i>	

Verdrahtung

Ein Ultraschallsensormodul ist ein Instrument, das mit Ultraschallwellen den Abstand zu einem Objekt misst. Es hat zwei Sonden. Eine sendet Ultraschallwellen, die andere empfängt die Wellen und wandelt die Sende- und Empfangszeit in eine Entfernung um, wodurch die Entfernung zwischen dem Gerät und einem Hindernis erkannt wird.

Bauen Sie die Schaltung entsprechend dem folgenden Diagramm auf.

Ultraschallmodul	R4 Board
Vcc	5V
Trig	3
Echo	4
Gnd	GND



Code

Bemerkung:

- Öffnen Sie die Datei `6.ultrasonic_module.ino` unter dem Pfad `3in1-kit\car_project\6.ultrasonic_module`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nachdem der Code erfolgreich hochgeladen wurde, richten Sie das Auto auf eine Wand. Wenn der Abstand zu groß ist, wird es vorwärts fahren; wenn es zu nah ist, wird es rückwärts fahren; wenn es in sicherem Abstand ist, wird es stoppen.

Wie funktioniert das?

Dieses Projekt basiert auf der vom Ultraschallmodul gelesenen Entfernung, um das Auto entsprechend zu steuern.

1. Fügen Sie die Pin-Definition für das Ultraschallmodul hinzu. `trigPin` wird verwendet, um Ultraschallwellen zu senden, setzen Sie diesen also auf `OUTPUT`; `echoPin` wird auf `INPUT` gesetzt, um Ultraschallwellen zu empfangen.


```

...
const int trigPin = 3;
const int echoPin = 4;

void setup() {
  ...

  //ultrasonic
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
}

```

2. Lesen Sie zuerst den Entfernungswert, der vom Ultraschallmodul abgerufen wird. Wenn die Entfernung größer als 25 ist, lassen Sie das Auto vorwärts fahren; wenn die Entfernung zwischen 2-10cm liegt, lassen Sie das Auto rückwärts fahren, ansonsten (zwischen 10~25) stoppen.

```

void loop() {
  float distance = readSensorData();
  if (distance > 25) {
    moveForward(200);
  }
  else if (distance < 10 && distance > 2) {
    moveBackward(200);
  } else {
    stopMove();
  }
}

```

3. Über die Funktion readSensorData().

Der Sender des Ultraschallmoduls sendet alle 2us ein 10us Quadratsignal, und der Empfänger empfängt ein High-Level-Signal, wenn sich ein Hindernis innerhalb des Bereichs befindet. Verwenden Sie die Funktion pulseIn(), um die Zeit von der Sendung bis zum Empfang zu protokollieren, teilen Sie durch die Schallgeschwindigkeit von 340m/s und dann durch 2. Das Ergebnis ist die Entfernung zwischen diesem Modul und dem Hindernis in Einheiten: cm.

```

float readSensorData() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  float distance = pulseIn(echoPin, HIGH) / 58.00; //Equivalent to (340m/
  ↪ s*1us)/2
  return distance;
}

```

- pulseIn(pin, value)
 - pin: die Nummer des Arduino-Pins, auf dem Sie den Impuls lesen möchten. Zulässige Datentypen: int.
 - value: Art des zu lesenden Pulses: entweder HIGH oder LOW. Zulässige Datentypen: int.

Liest einen Impuls (entweder HIGH oder LOW) an einem Pin. Wenn beispielsweise der Wert HIGH

ist, wartet `pulseIn()` darauf, dass der Pin von LOW auf HIGH wechselt, beginnt mit der Zeitmessung, dann wartet darauf, dass der Pin auf LOW geht und stoppt die Zeitmessung.

5.8 7. Folge deiner Hand

Betrachte dieses Auto hier als dein Haustier. Wenn du ihm zuwinkst, kommt es sofort zu dir gelaufen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Set zu kaufen. Hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Du kannst sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Ultraschall-Modul</i>	
<i>Hindernisvermeidungsmodul</i>	

Verdrahtung

Schließe das Ultraschallmodul und die beiden IR-Hindernisvermeidungsmodule gleichzeitig an.

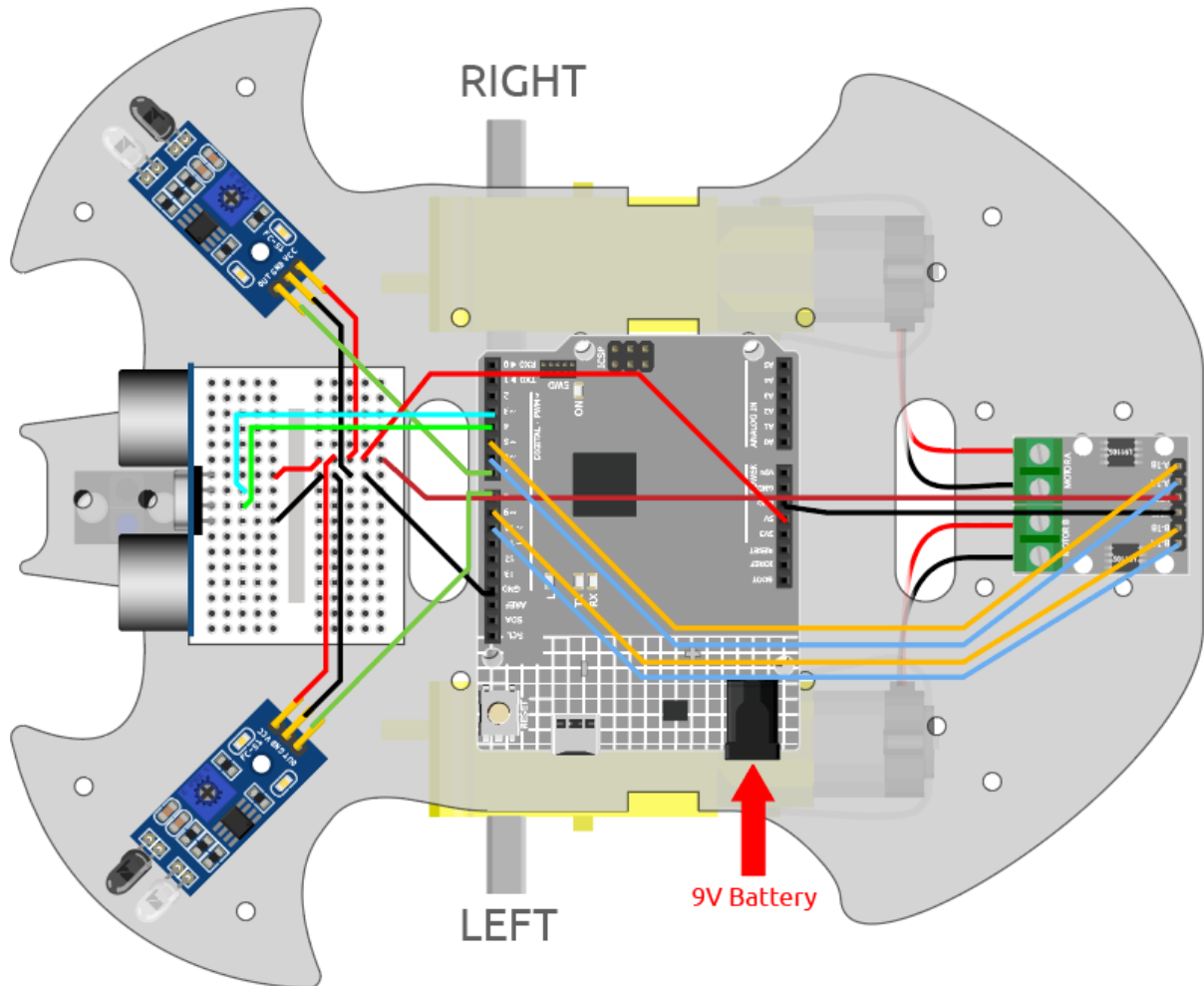
Die Verbindung des Ultraschallmoduls mit dem R4-Board erfolgt wie folgt.

Ultraschallmodul	R4 Board
Vcc	5V
Trig	3
Echo	4
Gnd	GND

Die Verkabelung der beiden IR-Hindernisvermeidungsmodule zum R4-Board ist wie folgt.

Linkes IR-Modul	R4 Board
OUT	8
GND	GND
VCC	5V

Rechtes IR-Modul	R4 Board
OUT	7
GND	GND
VCC	5V



Code

Bemerkung:

- Öffnen Sie die Datei `7. follow_your_hand.ino` im Pfad `3in1-kit\car_project\7. follow_your_hand`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Stellen Sie das Auto auf den Boden, nachdem der Code erfolgreich hochgeladen wurde. Halten Sie Ihre Hand in einem Abstand von 5*10cm vor das Auto, und es wird Ihrer Hand folgen. Wenn Sie Ihre Hand nahe am IR-Hindernismodul auf beiden Seiten positionieren, wird es auch in die entsprechende Richtung abbiegen.

Wie funktioniert das?

Dieses Projekt kombiniert die beiden vorherigen Projekte *6. Mit dem Ultraschallmodul spielen* und *5. Spielen mit dem Hindernisvermeidungs-Modul*, aber die umgesetzte Wirkung ist anders. In den vorherigen 2 Projekten wurde ein Hindernis rückwärts erkannt, hier jedoch wird erkannt, dass Ihre Hand der Vorwärts- oder Abbiege-Richtung folgt. Der Arbeitsablauf dieses Projekts ist wie folgt:

- Lese den vom Ultraschallmodul erkannten Abstand und den Wert beider Infrarotmodule.
- Wenn der Abstand 5~10cm beträgt, lasse das Auto Ihrer Hand folgen.

- Wenn das linke IR-Modul Ihre Hand erkennt, biege links ab.
- Wenn das rechte IR-Modul Ihre Hand erkennt, biege rechts ab.
- Wenn weder das Infrarotmodul noch das Ultraschallmodul Ihre Hand erkennen, lasse das Auto anhalten.

```
void loop() {

    float distance = readSensorData();

    int left = digitalRead(leftIR);    // 0: Obstructed  1: Empty
    int right = digitalRead(rightIR);
    int speed = 150;

    if (distance>5 && distance<10){
        moveForward(speed);
    }
    if(!left&&right){
        turnLeft(speed);
    }else if(left&&!right){
        turnRight(speed);
    }else{
        stopMove();
    }
}
```

5.9 8. Selbstfahrendes Auto

Dieses Projekt kombiniert die beiden Projekte *6. Mit dem Ultraschallmodul spielen* und *5. Spielen mit dem Hindernisvermeidungs-Modul*. 2 Infrarot-Hindernisvermeidungsmodule führen Kurzstrecken- oder Kantenerkennung durch, und Ultraschallmodule sorgen für Langstreckenerkennung, um sicherzustellen, dass das Auto während des freien Fahrens kein Hindernis trifft.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Ultraschall-Modul</i>	
<i>Hindernisvermeidungsmodul</i>	

Verdrahtung

Schließen Sie das Ultraschallmodul und die 2 IR-Hindernisvermeidungsmodule gleichzeitig an.

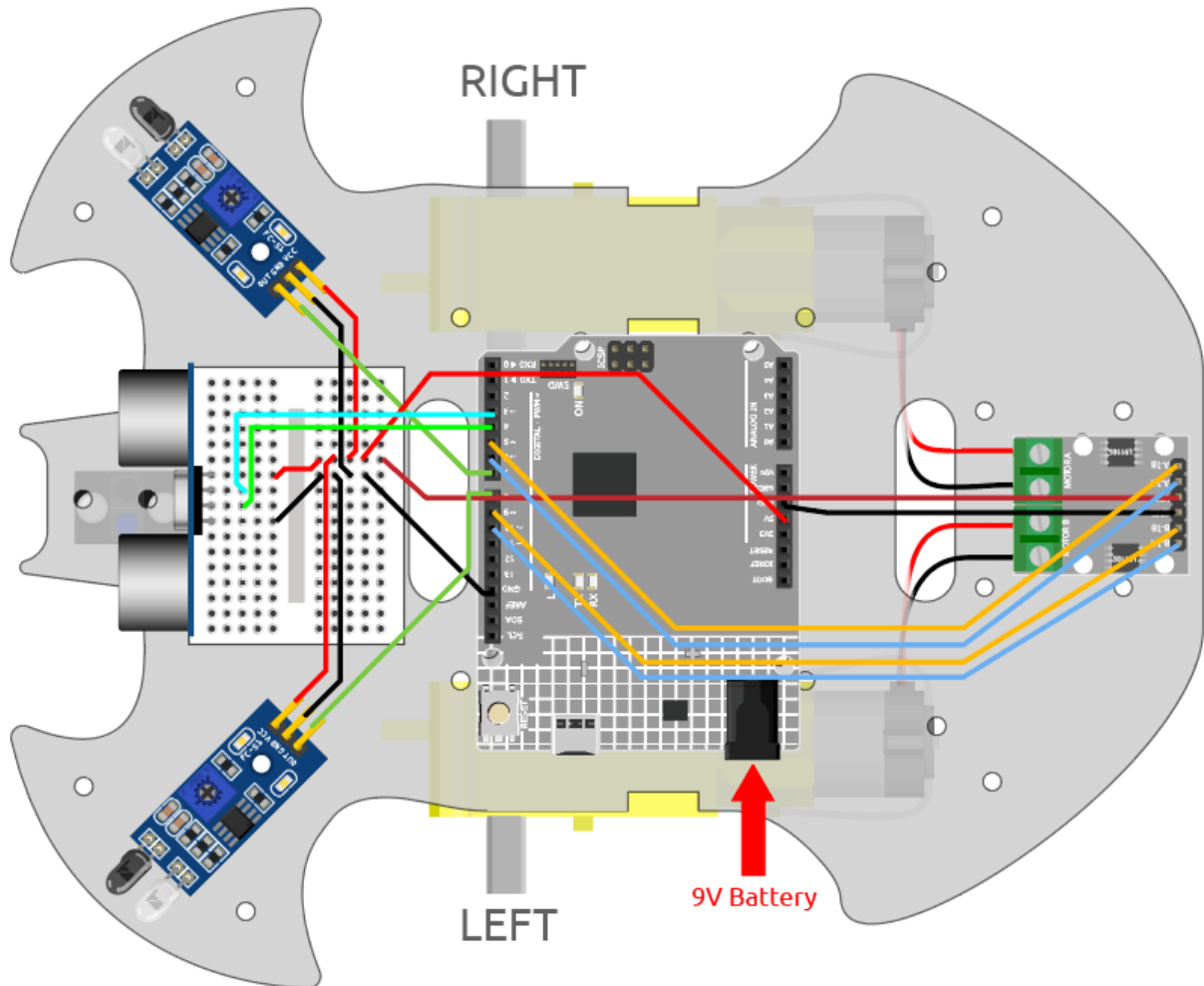
Verdrahten Sie das Ultraschallmodul wie folgt mit dem R4-Board.

Ultraschallmodul	R4 Board
Vcc	5V
Trig	3
Echo	4
Gnd	GND

Die Verdrahtung der 2 IR-Hindernisvermeidungsmodule zum R4-Board ist wie folgt:

Linkes IR-Modul	R4 Board
OUT	8
GND	GND
VCC	5V

Rechtes IR-Modul	R4 Board
OUT	7
GND	GND
VCC	5V



Code

Bemerkung:

- Öffnen Sie die Datei `8.self_driving_car.ino` unter dem Pfad `3in1-kit\car_project\8.self_driving_car`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Sobald der Code erfolgreich hochgeladen wurde, wird das Auto frei fahren. Wenn das IR-Hindernismodul auf beiden Seiten ein Hindernis erkennt, wird es sich zur Notausweichung in die entgegengesetzte Richtung bewegen; befindet sich ein Hindernis direkt vor dem Auto im Abstand von 2~10cm, wird es nach links zurücksetzen, seine Richtung anpassen und dann vorwärts fahren.

Wie funktioniert das?

Der Ablauf dieses Projekts ist wie folgt.

- Priorität beim Auslesen der Werte des linken und rechten IR-Hindernisvermeidungsmoduls.
- Wenn das linke IR-Modul 0 ist (Hindernis erkannt) und das rechte IR-Modul 1 ist, soll das Auto nach links zurücksetzen.

- Wenn das rechte IR-Modul 0 ist (Hindernis erkannt), soll das Auto nach rechts zurücksetzen.
- Wenn beide IR-Module gleichzeitig ein Hindernis erkennen, wird das Auto zurücksetzen.
- Ansonsten wird der Abstand, der vom Ultraschallmodul erkannt wurde, ausgelesen.
- Wenn der Abstand größer als 50cm ist, soll das Auto vorwärts fahren.
- Wenn der Abstand zwischen 2-10cm liegt, soll das Auto zurücksetzen, bevor es abbiegt.
- Wenn der Abstand zwischen 10-50cm liegt, soll das Auto mit geringer Geschwindigkeit vorwärts fahren.

```
void loop() {

    int left = digitalRead(leftIR);    // 0: Obstructed  1: Empty
    int right = digitalRead(rightIR);

    if (!left && right) {
        backLeft(150);
    } else if (left && !right) {
        backRight(150);
    } else if (!left && !right) {
        moveBackward(150);
    } else {
        float distance = readSensorData();
        Serial.println(distance);
        if (distance > 50) { // Safe
            moveForward(200);
        } else if (distance < 10 && distance > 2) { // Attention
            moveBackward(200);
            delay(1000);
            backLeft(150);
            delay(500);
        } else {
            moveForward(150);
        }
    }
}
```

5.10 9. Fernbedienung

Dieses Kit enthält einen IR-Empfänger, der es ermöglicht, das Auto mit einer IR-Fernbedienung zu steuern.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

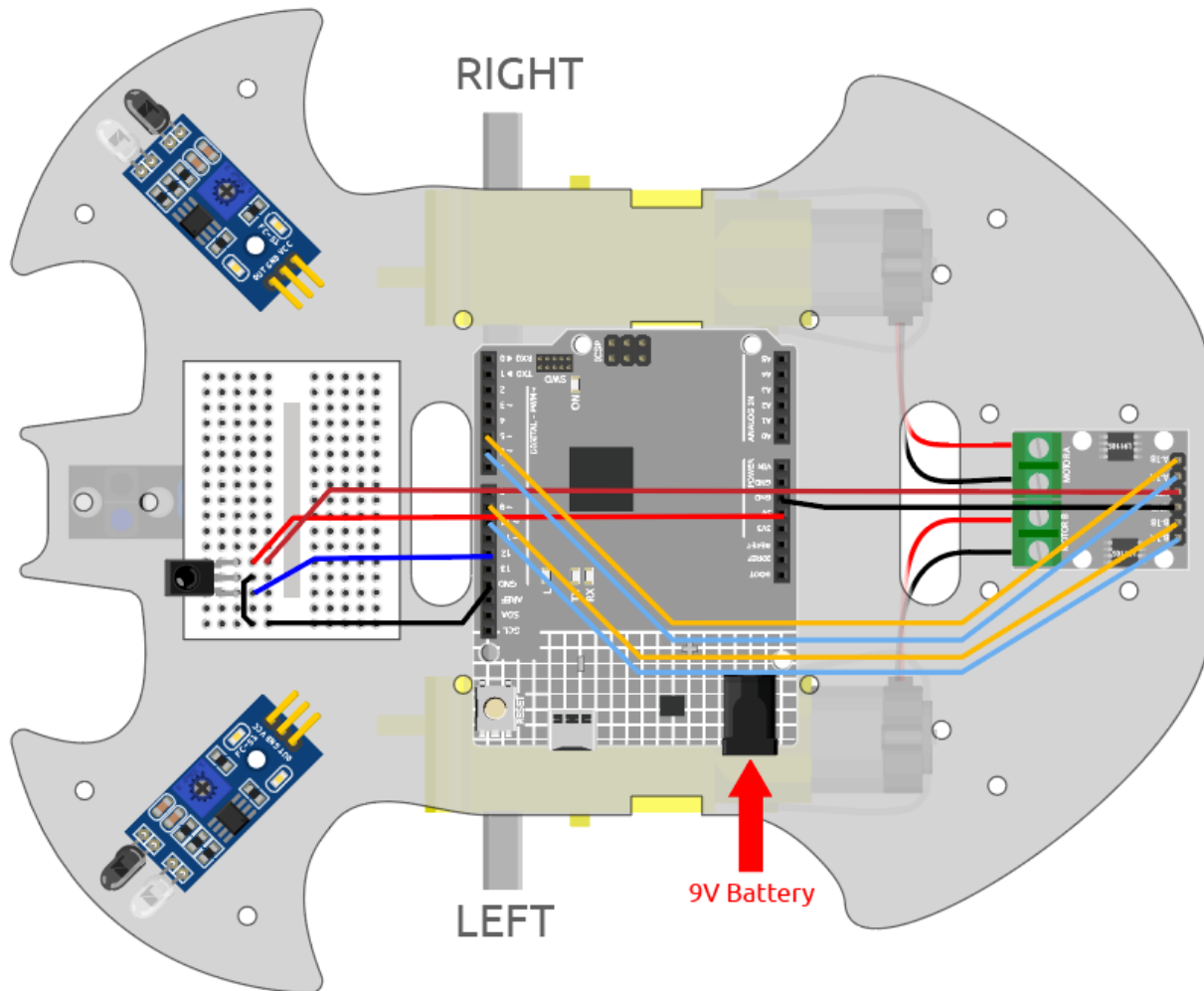
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>LED</i>	
<i>IR-Empfänger</i>	-

Verdrahtung

Bauen Sie nun den Schaltkreis entsprechend dem untenstehenden Diagramm.

IR-Empfänger	R4-Platine
OUT	12
GND	GND
VCC	5V

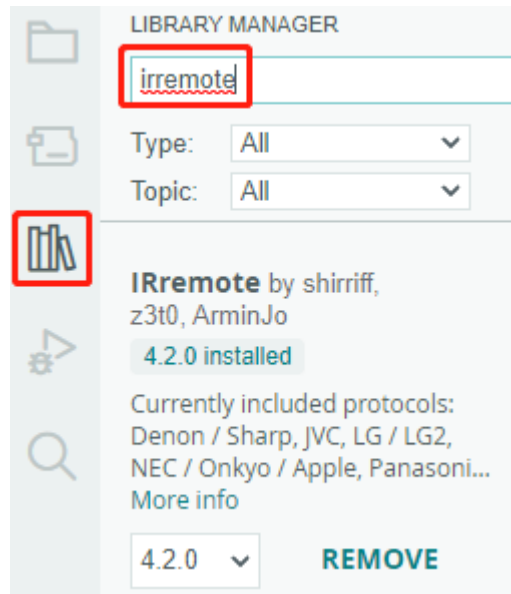
LED	R4-Platine
Anode(Der längere Pin)	13
Kathode	GND



Code

Bemerkung:

- Öffnen Sie die Datei 9.remote_control.ino im Pfad 3in1-kit\car_project\9.remote_control.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Hier wird die Bibliothek IRremote verwendet, die Sie über den **Library Manager** installieren können.



Nachdem der Code erfolgreich hochgeladen wurde, drücken Sie die Taste auf der Fernbedienung. Die LED wird einmal blinken, um anzuzeigen, dass das Signal empfangen wurde, und das Auto wird sich entsprechend der gedrückten Taste bewegen. Sie können die folgenden Tasten drücken, um das Auto zu steuern.

- +: Beschleunigen
- -: Verlangsamen
- 1: Vorwärts nach links
- 2: Vorwärts
- 3: Vorwärts nach rechts
- 4: Nach links drehen
- 6: Nach rechts drehen
- 7: Rückwärts nach links
- 8: Rückwärts
- 9: Rückwärts nach rechts

Wie funktioniert das?

Das Ziel dieses Projekts ist es, das Auto durch das Lesen des Tastenwertes der IR-Fernbedienung zu bewegen. Zusätzlich wird eine LED hinzugefügt, um anzuzeigen, dass das IR-Signal erfolgreich empfangen wurde.

1. Importieren Sie die IRremote-Bibliothek. Sie können sie über den **Library Manager** installieren.

```
#include <IRremote.h>

const int IR_RECEIVE_PIN = 12;  // Define the pin number for the IR Sensor
```

2. Initialisiert die serielle Kommunikation mit einer Baudrate von 9600. Startet den IR-Empfänger am angegebenen Pin (IR_RECEIVE_PIN) und aktiviert die LED-Rückmeldung (falls zutreffend).

...

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

void setup() {
    ...
    //IR remote
    IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK); // Start the
↪receiver
    Serial.println("REMOTE CONTROL START");
}

```

3. Wenn Sie die Tasten auf der Fernbedienung drücken, der Infrarotempfänger erkennt, welche Taste gedrückt wurde. Anschließend bewegt sich das Auto entsprechend dem zugehörigen Tastenwert.

```

void loop() {
    if (IrReceiver.decode()) {
        // Serial.println(results.value, HEX);
        String key = decodeKeyValue(IrReceiver.decodedIRData.command);
        if (key != "ERROR") {
            Serial.println(key);

            if (key == "+") {
                speed += 50;
            } else if (key == "-") {
                speed -= 50;
            } else if (key == "2") {
                moveForward(speed);
                delay(1000);
            }
            ...
        }
        IrReceiver.resume(); // Enable receiving of the next value
    }
}

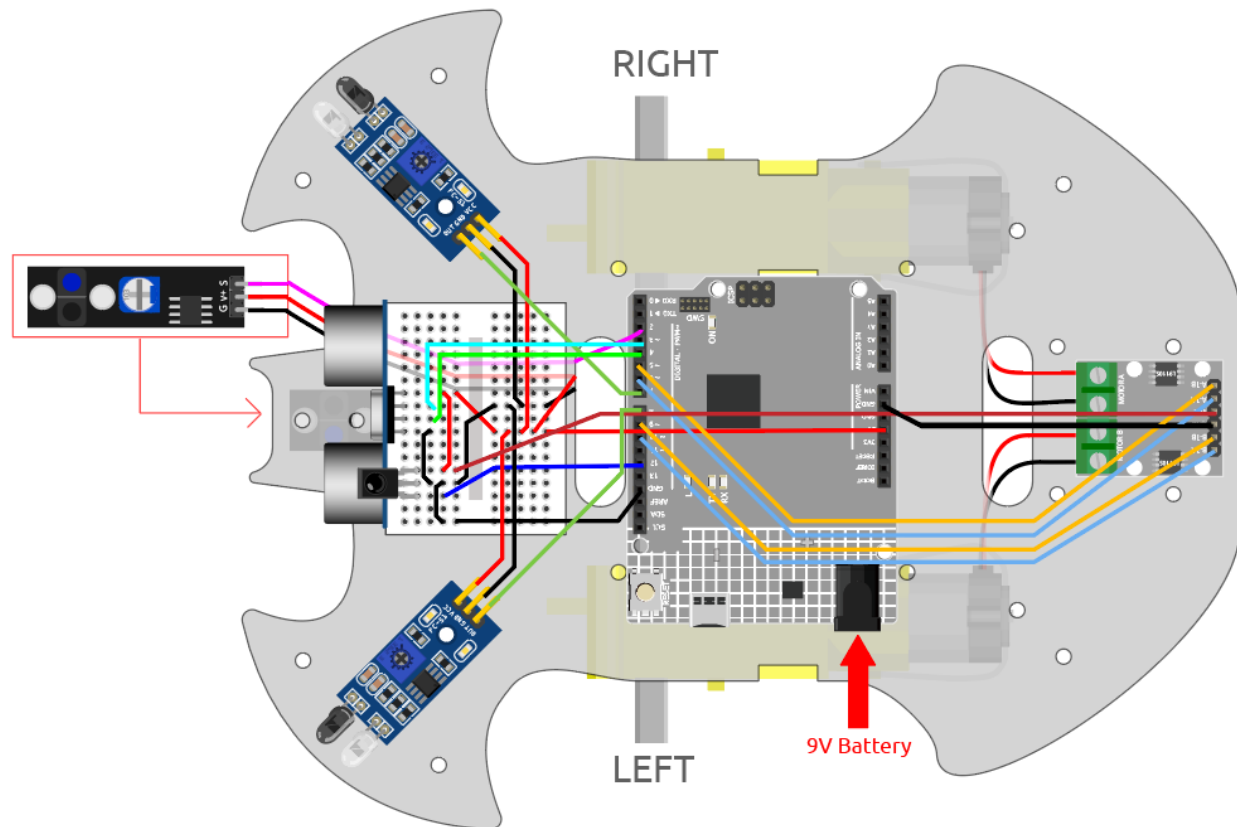
```

- Überprüft, ob ein IR-Signal empfangen und erfolgreich entschlüsselt wurde.
- Entschlüsselt den IR-Befehl und speichert ihn in key mit Hilfe einer benutzerdefinierten decodeKeyValue()-Funktion.
- Überprüft, ob der entschlüsselte Wert kein Fehler ist.
- Gibt den entschlüsselten IR-Wert auf dem seriellen Monitor aus.
- Setzt den IR-Signalempfang für das nächste Signal fort.

5.11 10. Starten mit einem Tastendruck

In diesem Projekt haben wir die vorherigen Projekte - Linienverfolgung, Verfolgung, Hindernisvermeidung, autonomes Fahren usw. zusammengeführt. Sie können durch Tasten auf der Fernbedienung umgeschaltet werden, sodass Sie das Auto starten und alle Funktionen gleichzeitig erleben können.

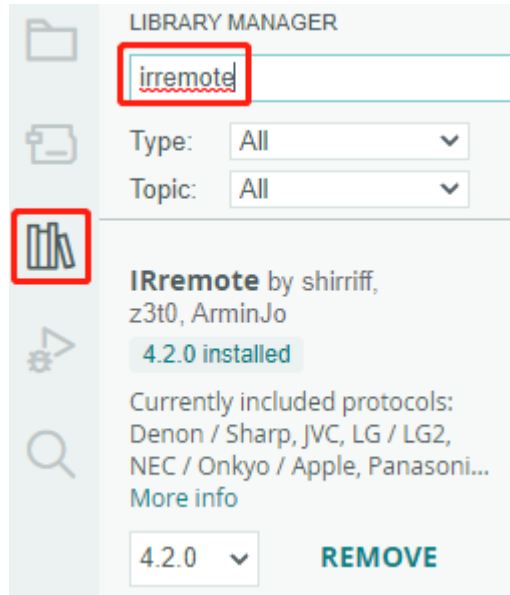
Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei `10.one_touch_start.ino` unter dem Pfad `3in1-kit\car_project\10.one_touch_start`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Hier wird die `IRremote` Bibliothek verwendet. Sie können sie über den **Library Manager** installieren.



Nach erfolgreichem Hochladen des Codes wird die LED für jedes vom IR-Empfänger empfangene Signal von der Fernbedienung 3 Mal schnell blinken. Sie können die folgenden Tasten drücken, um den Wagen zu steuern.

- **+**: Beschleunigen
- **-**: Verlangsamen
- **1**: Bewegen Sie sich zur linken Vorderseite
- **2**: Vorwärts
- **3**: Bewegen Sie sich nach rechts.
- **4**: Nach links drehen
- **6**: Nach rechts drehen
- **7**: Rückwärts nach links
- **8**: Rückwärts
- **9**: Rückwärts nach rechts
- **CYCLE**: Der Linie folgen
- **U/SD**: Autonomes Fahren
- **|**: Hindernisvermeidung mit Ultraschallmodul
- **|**: Hindernisvermeidung mit IR-Hindernismodul
- **EQ**: Deiner Hand folgen
- **0**: Anhalten

5.12 11. Geschwindigkeitskalibrierung

Wenn Sie das Auto vorwärts bewegen, stellen Sie möglicherweise fest, dass es nicht geradeaus fährt. Dies liegt daran, dass die beiden Motoren ab Werk möglicherweise nicht die gleiche Geschwindigkeit haben. Wir können jedoch einen Offset zu den beiden Motoren hinzufügen, um ihre Drehgeschwindigkeiten anzugleichen.

In diesem Projekt werden wir lernen, den Offset im **EEPROM** zu speichern. Der Vorteil davon ist, dass nach jeder Kalibrierung alle Projekte den Offset-Wert direkt aus dem EEPROM abrufen können, damit das Auto reibungslos geradeaus fährt.

Verdrahtung

Dieses Projekt hat die gleiche Verdrahtung wie *2. Steuerung durch Code*.

Wie spielt man?

1. Öffnen Sie die Datei `11.speed_calibration.ino` im Pfad `3in1-kit\car_project\11.speed_calibration`. Oder kopieren Sie diesen Code in die **Arduino IDE**.
2. Nachdem der Code erfolgreich hochgeladen wurde, verbinden Sie das Auto mit einer 9V-Batterie, stellen Sie es auf den Boden und beobachten Sie, zu welcher Seite es versetzt ist.
 - Wenn das Auto nach links vorne fährt, bedeutet dies, dass die Geschwindigkeit des rechten Motors zu hoch ist und reduziert werden muss.

```
EEPROM.write(1, 100) // 1 means the right motor, 100 means 100% speed, can
↳ be set to 90, 95, etc., depending on the actual situation.
```

- Wenn das Auto nach rechts fährt, bedeutet dies, dass die Geschwindigkeit des linken Motors zu hoch ist und reduziert werden muss.

```
EEPROM.write(0, 100) // 0 means the right motor, 100 means the speed is 100
↳ %, can be set to 90, 95, etc., depending on the actual situation. 3.
```

3. Nachdem Sie den Code geändert haben, laden Sie den Code auf das R4-Board, um den Effekt zu sehen. Wiederholen Sie die oben genannten Schritte, bis das Auto fast geradeaus fährt.
4. Dieser Offset wird im **EEPROM** gespeichert. Sie müssen diesen Offset nur lesen, wenn Sie ihn in anderen Projekten verwenden, nehmen Sie zum Beispiel *5. Spielen mit dem Hindernisvermeidungs-Modul*.

```
#include <EEPROM.h>

float leftOffset = 1.0;
float rightOffset = 1.0;

const int A_1B = 5;
const int A_1A = 6;
const int B_1B = 9;
const int B_1A = 10;

const int rightIR = 7;
const int leftIR = 8;

void setup() {
  Serial.begin(9600);

  //motor
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

pinMode(A_1B, OUTPUT);
pinMode(A_1A, OUTPUT);
pinMode(B_1B, OUTPUT);
pinMode(B_1A, OUTPUT);

//IR obstacle
pinMode(leftIR, INPUT);
pinMode(rightIR, INPUT);

leftOffset = EEPROM.read(0) * 0.01; //read the offset of the left motor
rightOffset = EEPROM.read(1) * 0.01; //read the offset of the right motor
}

void loop() {

    int left = digitalRead(leftIR); // 0: Obstructed 1: Empty
    int right = digitalRead(rightIR);
    int speed = 150;

    if (!left && right) {
        backLeft(speed);
    } else if (left && !right) {
        backRight(speed);
    } else if (!left && !right) {
        moveBackward(speed);
    } else {
        moveForward(speed);
    }
}

void moveForward(int speed) {
    analogWrite(A_1B, 0);
    analogWrite(A_1A, int(speed * leftOffset));
    analogWrite(B_1B, int(speed * rightOffset));
    analogWrite(B_1A, 0);
}

void moveBackward(int speed) {
    analogWrite(A_1B, speed);
    analogWrite(A_1A, 0);
    analogWrite(B_1B, 0);
    analogWrite(B_1A, speed);
}

void backLeft(int speed) {
    analogWrite(A_1B, speed);
    analogWrite(A_1A, 0);
    analogWrite(B_1B, 0);
    analogWrite(B_1A, 0);
}

void backRight(int speed) {

```

(Fortsetzung auf der nächsten Seite)

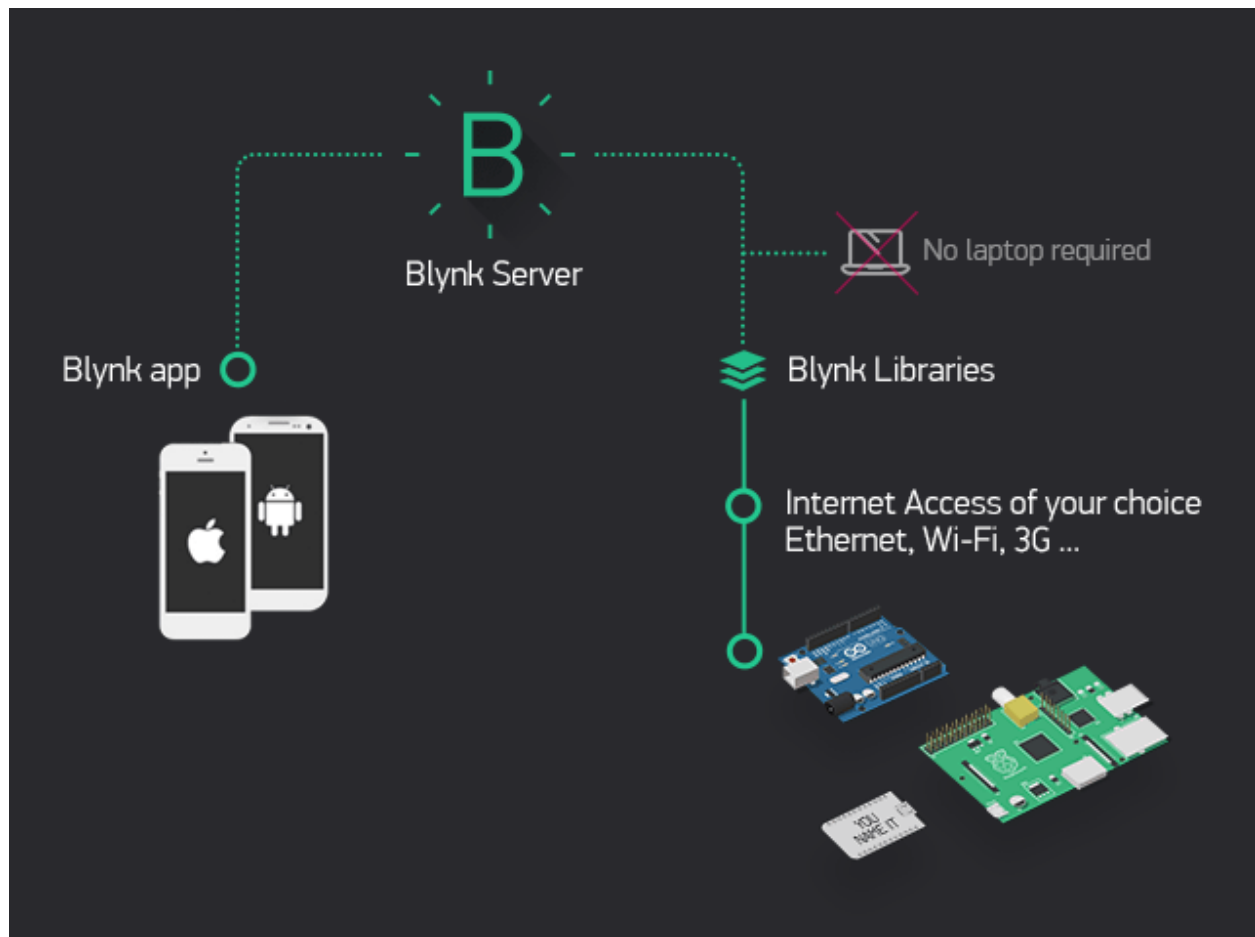
(Fortsetzung der vorherigen Seite)

```
    analogWrite(A_1B, 0);  
    analogWrite(A_1A, 0);  
    analogWrite(B_1B, 0);  
    analogWrite(B_1A, speed);  
}
```


Dieses Kit beinhaltet das ESP8266 Wifi-Modul, welches es dem Arduino ermöglicht, sich für IoT-Experimente mit dem Internet zu verbinden.

In diesem Leitfaden zeigen wir Ihnen, wie Sie den Arduino mithilfe des ESP8266 Wifi-Moduls mit der **BLYNK** Plattform verbinden können, um spannende IoT-Projekte umzusetzen. Zudem können Sie die Blynk-App auf Ihrem Smartphone verwenden, um das Smart-Car zu steuern.

Blynk bietet eine umfassende Softwarelösung, die für das Prototyping, die Implementierung und die Fernverwaltung von vernetzten elektronischen Geräten in jeder Größenordnung erforderlich ist: von persönlichen IoT-Projekten bis hin zu Millionen kommerziell vernetzter Produkte. Mit Blynk kann jeder seine Hardware mit der Cloud verbinden und ohne Programmierkenntnisse iOS-, Android- und Webanwendungen erstellen, um Echtzeit- und historische Daten von Geräten zu analysieren, sie aus der Ferne aus aller Welt zu steuern, wichtige Benachrichtigungen zu erhalten und vieles mehr...



6.1 1. Erste Schritte mit Blynk

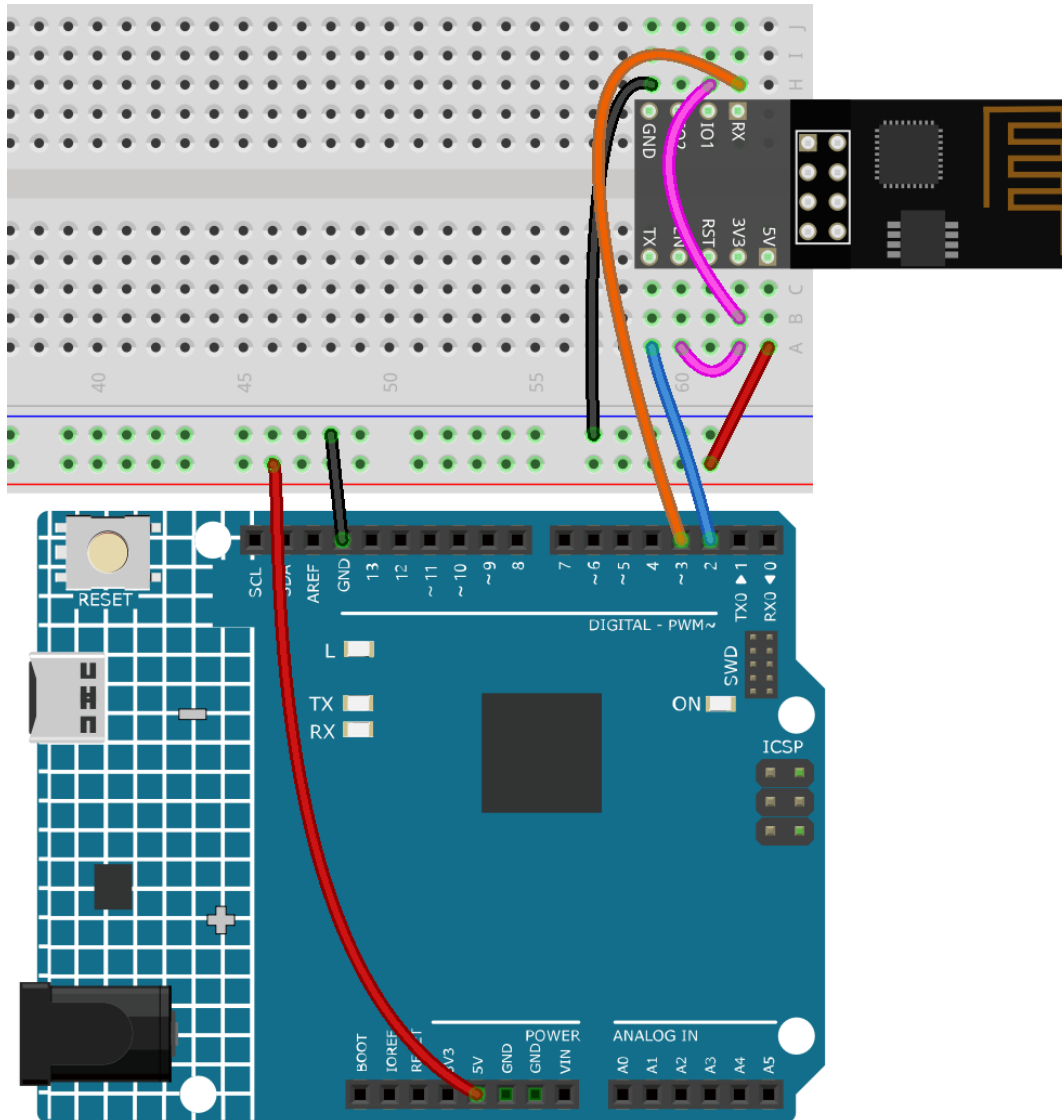
Um die Kommunikation des R4-Boards mit Blynk zu ermöglichen, ist bei der ersten Verwendung von Blynk eine gewisse Konfiguration erforderlich.

Befolgen Sie die unten aufgeführten Schritte und beachten Sie, dass Sie sie in der angegebenen Reihenfolge durchführen und kein Kapitel überspringen sollten.

6.1.1 1.1 Konfiguration des ESP8266

Das ESP8266-Modul, das mit dem Kit geliefert wird, ist bereits mit der AT-Firmware vorgebrannt. Dennoch müssen Sie dessen Konfiguration ändern, indem Sie die untenstehenden Schritte befolgen.

1. Bauen Sie den Schaltkreis.



2. Öffnen Sie die Datei `1.set_software_serial.ino` unter dem Pfad `3in1-kit\iot_project\1.set_software_serial`. Oder kopieren Sie diesen Code in die **Arduino IDE**.

```
#include <SoftwareSerial.h>
SoftwareSerial espSerial(2, 3); //Rx,Tx

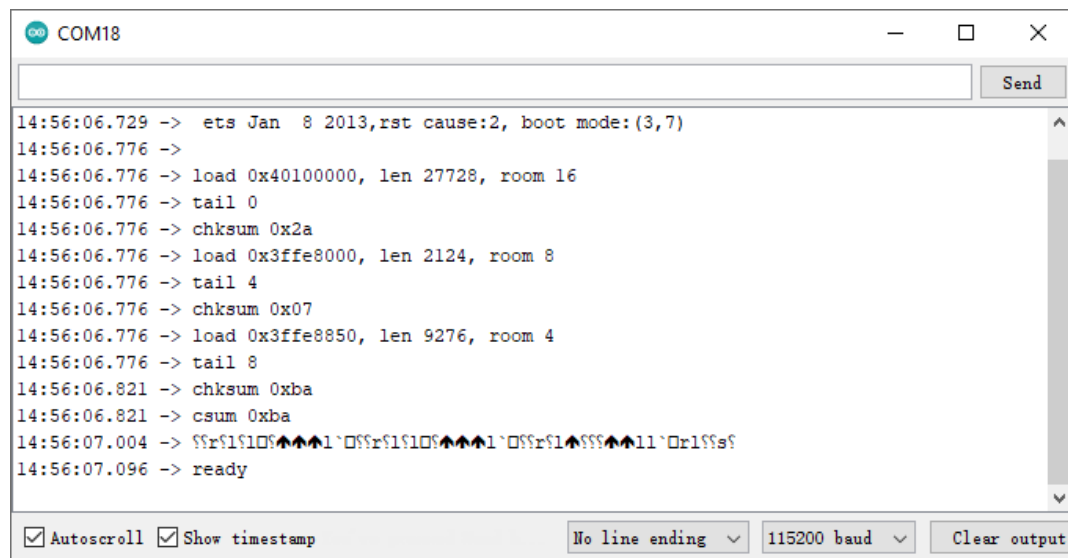
void setup() {
  // put your setup code here, to run once:
  Serial.begin(115200);
  espSerial.begin(115200);
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
void loop() {
    if (espSerial.available()) {
        Serial.write(espSerial.read());
    }
    if (Serial.available()) {
        espSerial.write(Serial.read());
    }
}
```

3. Klicken Sie auf das Lupensymbol (Serial Monitor) in der oberen rechten Ecke und setzen Sie die Baudrate auf **115200**. (Es kann sein, dass bei Ihnen Informationen angezeigt werden, wie bei mir, oder auch nicht. Das spielt keine Rolle, gehen Sie einfach zum nächsten Schritt weiter.)



Warnung:

- Wenn ready nicht erscheint, können Sie versuchen, das ESP8266-Modul zurückzusetzen (RST mit GND verbinden) und den Serial Monitor erneut zu öffnen.
- Zudem, wenn das Ergebnis OK ist, müssen Sie möglicherweise die Firmware erneut brennen. Bitte beziehen Sie sich für Details auf *[Wie lädt man die Firmware für das ESP8266-Modul erneut hoch?](#)*. Wenn Sie das Problem immer noch nicht lösen können, nehmen Sie bitte einen Screenshot vom Serial Monitor und senden Sie diesen an sevice@sunfounder.com. Wir werden Ihnen so schnell wie möglich helfen.

4. Klicken Sie auf das **NEWLINE DROPDOWN BOX** und wählen Sie im Dropdown-Menü both NL & CR aus. Geben Sie AT ein. Wenn OK zurückgegeben wird, bedeutet das, dass ESP8266 erfolgreich eine Verbindung mit Ihrem Board hergestellt hat.

```

COM18
AT2
14:56:06.776 -> load 0x40100000, len 27728, room 16
14:56:06.776 -> tail 0
14:56:06.776 -> chksum 0x2a
14:56:06.776 -> load 0x3ffe8000, len 2124, room 8
14:56:06.776 -> tail 4
14:56:06.776 -> chksum 0x07
14:56:06.776 -> load 0x3ffe8850, len 9276, room 4
14:56:06.776 -> tail 8
14:56:06.821 -> chksum 0xba
14:56:06.821 -> csum 0xba
14:56:07.004 -> $$$r$1$10$####1`0$$r$1$10$####1`0$$r$1$10$####1`0r1$$s$
14:56:07.096 -> ready
15:00:05.119 -> AT
15:00:05.119 ->
15:00:05.119 -> OK

```

☒ Autoscroll ☒ Show timestamp Both NL & CR 115200 baud Clear output

5. Geben Sie `AT+CWMODE=3` ein und der verwaltete Modus wird zu **Station and AP-Koexistenz** geändert.

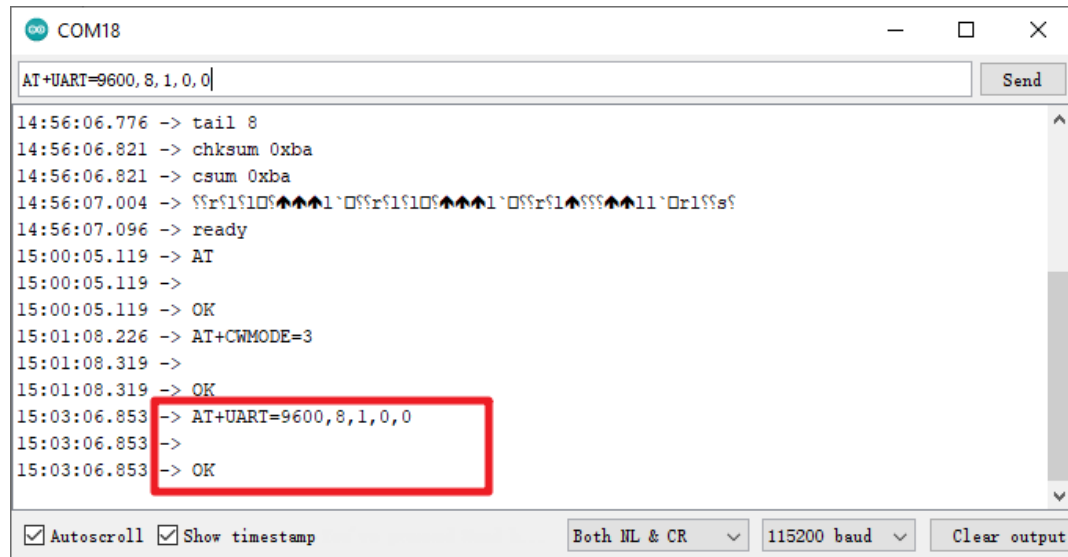
```

COM18
AT+CWMODE=3
14:56:06.776 -> tail 4
14:56:06.776 -> chksum 0x07
14:56:06.776 -> load 0x3ffe8850, len 9276, room 4
14:56:06.776 -> tail 8
14:56:06.821 -> chksum 0xba
14:56:06.821 -> csum 0xba
14:56:07.004 -> $$$r$1$10$####1`0$$r$1$10$####1`0$$r$1$10$####1`0r1$$s$
14:56:07.096 -> ready
15:00:05.119 -> AT
15:00:05.119 ->
15:00:05.119 -> OK
15:01:08.226 -> AT+CWMODE=3
15:01:08.319 ->
15:01:08.319 -> OK

```

☒ Autoscroll ☒ Show timestamp Both NL & CR 115200 baud Clear output

6. Um später die Software-Serial verwenden zu können, müssen Sie `AT+UART=9600,8,1,0,0` eingeben, um die Baudrate des ESP8266 auf 9600 zu ändern.

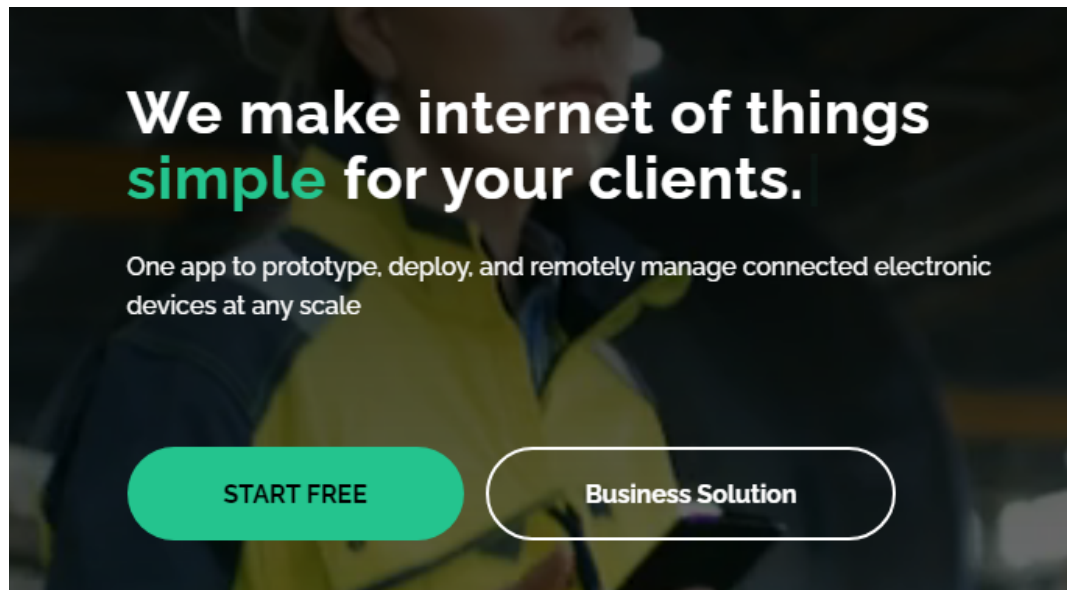


```
COM18
AT+UART=9600,8,1,0,0
14:56:06.776 -> tail 8
14:56:06.821 -> chksum 0xba
14:56:06.821 -> csum 0xba
14:56:07.004 -> $$$1$10$1111`0$$$1$10$1111`0$$$1$1111`0r1$$$
14:56:07.096 -> ready
15:00:05.119 -> AT
15:00:05.119 ->
15:00:05.119 -> OK
15:01:08.226 -> AT+CWMODE=3
15:01:08.319 ->
15:01:08.319 -> OK
15:03:06.853 -> AT+UART=9600,8,1,0,0
15:03:06.853 ->
15:03:06.853 -> OK


☒ Autoscroll ☒ Show timestamp Both NL & CR 115200 baud Clear output
```

6.1.2 1.2 Konfiguration von Blynk

1. Gehen Sie zu [BLYNK](#) und klicken Sie auf **START FREE**.



2. Geben Sie Ihre E-Mail-Adresse ein, um ein Konto zu registrieren.



Sign Up

Welcome! Fill in your email address and we will send an account activation link.

EMAIL

☐ I agree to [Terms and Conditions](#) and accept [Privacy Policy](#)

[Sign Up](#)

[Back to Login](#)

3. Überprüfen Sie Ihre E-Mail-Adresse, um die Kontoregistrierung abzuschließen.

Welcome!

We're excited to see you on board.

To get started, you'll need to create a password for your account.

Create Password

The link will expire in 30 days.

4. Anschließend wird die **Blynk Tour** angezeigt. Sie können sie lesen, um grundlegende Informationen über Blynk zu erhalten.

Blynk Tour

1 Welcome 2 Platform 3 Modes 4 Devices 5 Template 6 Template components 7 Features 8 Business

Hi Blynker!

You've just joined the community of more than 500,000+ developers building amazing IoT products and projects.

With Blynk you can connect your devices to the Internet and create mobile and web dashboards to control your devices from anywhere in the world.

Let's save your learning time with a few quick steps.



Skip

Let's go!

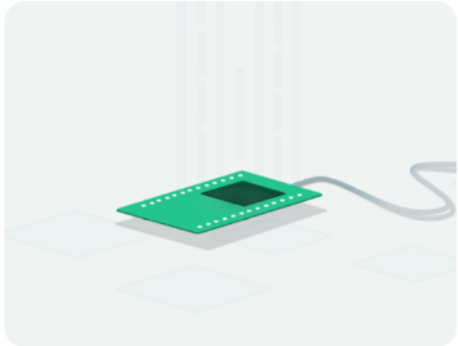
5. Als Nächstes müssen wir mit diesem **Quick Start** eine Vorlage und ein Gerät erstellen. Klicken Sie auf **Let's go**.

Quickstart

This is a step by step guide to get your first device online and start controlling it from anywhere in the world in **less than 5 minutes**

What you will need:

- Supported hardware. Check the full list of supported hardware [here](#).
- IDE. You can use Arduino IDE or PlatformIO or any other editor.
- Blynk Library
- It will be beneficial if you already know how to upload code to your hardware.



CancelLet's go!

6. Wählen Sie die Hardware und den Verbindungstyp.

Quickstart

1

 Hardware —

2

 IDE —

3

 Blynk Library —

4

 Code —

5

 Device activation

Which hardware are you using?

We will help you prepare the code for you board

ESP8266

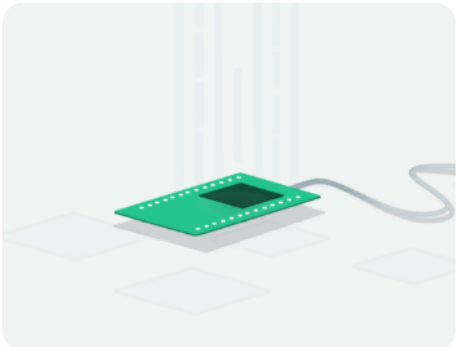
▼

What is your device connectivity type

Blynk supports various connection types (BLE is not supported yet).

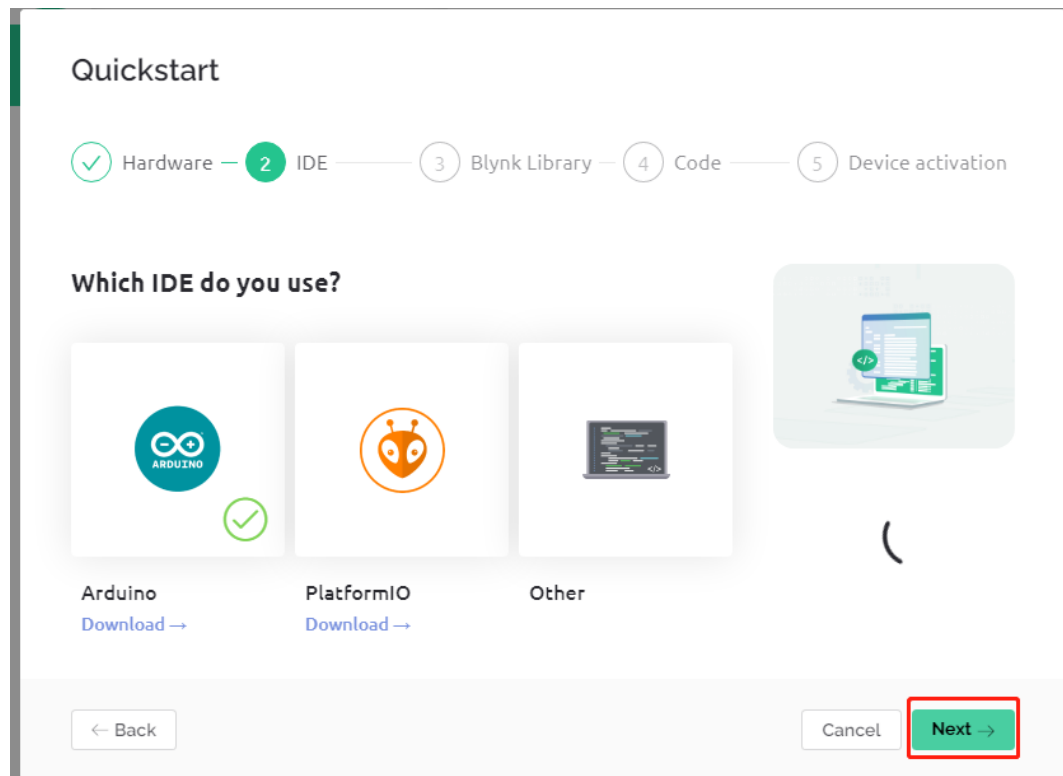
WiFi

▼

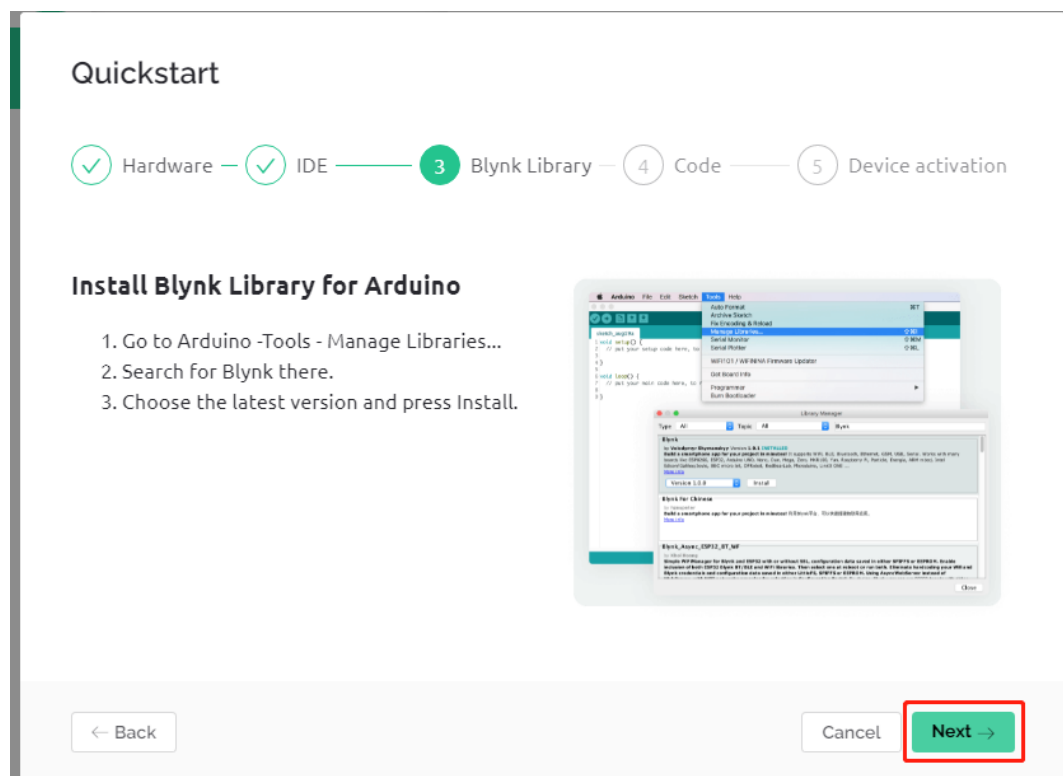


CancelNext →

7. Hier wird Ihnen mitgeteilt, welche IDE Sie vorbereiten müssen. Wir empfehlen die **Arduino IDE**.

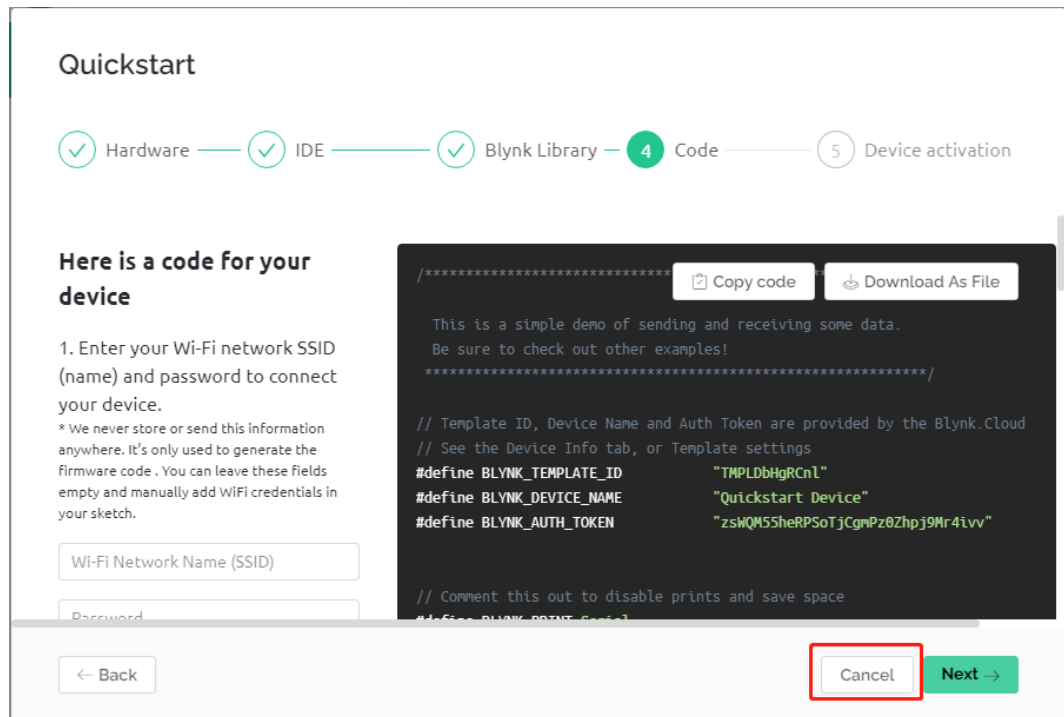


8. Hier ist die Bibliothek, die Sie hinzufügen müssen. Aber die hier empfohlene Bibliothek ist ein wenig problematisch. Wir müssen andere Bibliotheken manuell hinzufügen (wir werden das später erwähnen). Klicken Sie hier auf **Next** und eine neue Vorlage und ein Gerät werden erstellt.

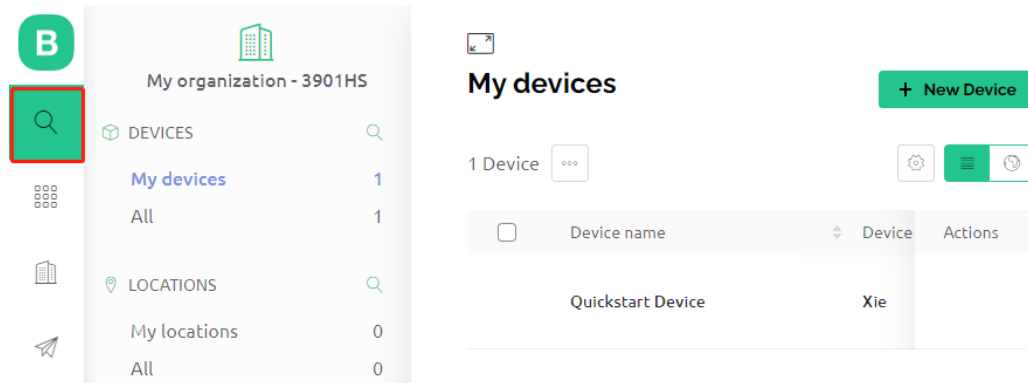


9. Die nächsten Schritte bestehen darin, den relevanten Code hochzuladen und Ihr Board mit Blynk zu verbinden.

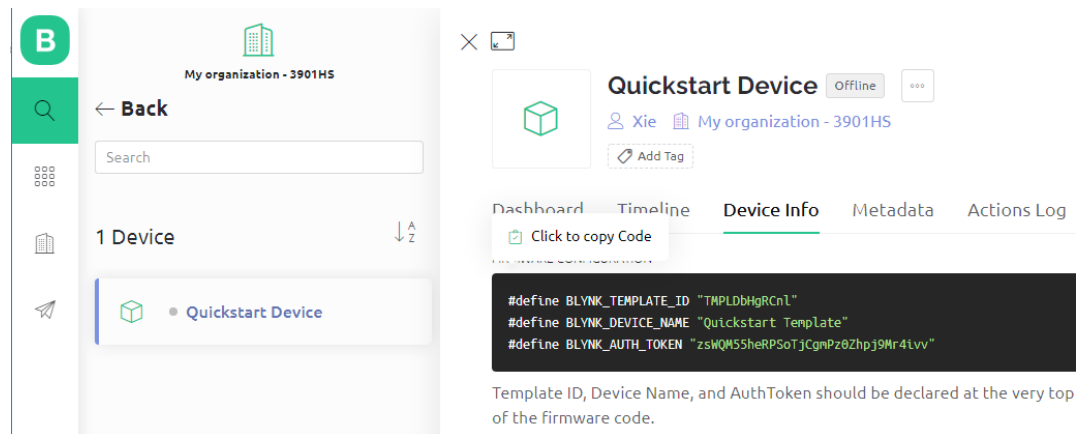
Da jedoch ein Problem mit der zuvor bereitgestellten Bibliothek besteht, müssen Sie andere Bibliotheken erneut hinzufügen. Klicken Sie daher hier auf **Cancel**, um **Quick Start** zu beenden.



10. Klicken Sie auf den **Suchen**-Button und Sie werden das neue Gerät sehen, das Sie gerade erstellt haben.



11. Gehen Sie zu diesem **Quickstart Device** und auf der **Device info**-Seite sehen Sie **TEMPLATE_ID**, **DEVICE_NAME** und **AUTH_TOKEN**. Sie werden diese später kopieren müssen.



6.1.3 1.3 Hinzufügen der benötigten Bibliotheken

Um Blynk mit der Arduino IDE zu nutzen, müssen Sie die richtigen Bibliotheken hinzufügen.

1. Klicken Sie [HIER](#), scrollen Sie bis zum Ende der Seite und laden Sie die erste .zip Datei herunter.

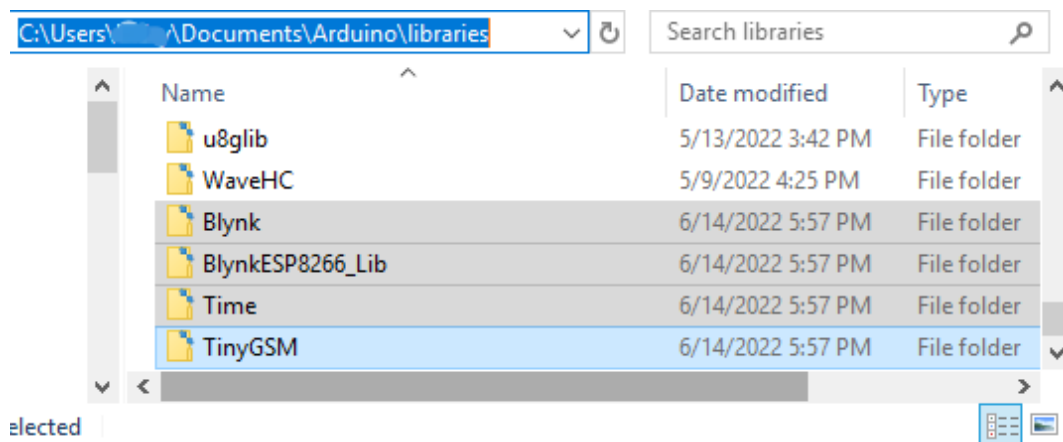
▼ Assets 3

Blynk_Release_v1.1.0.zip	779 KB	22 days ago
Source code (zip)		22 days ago
Source code (tar.gz)		22 days ago

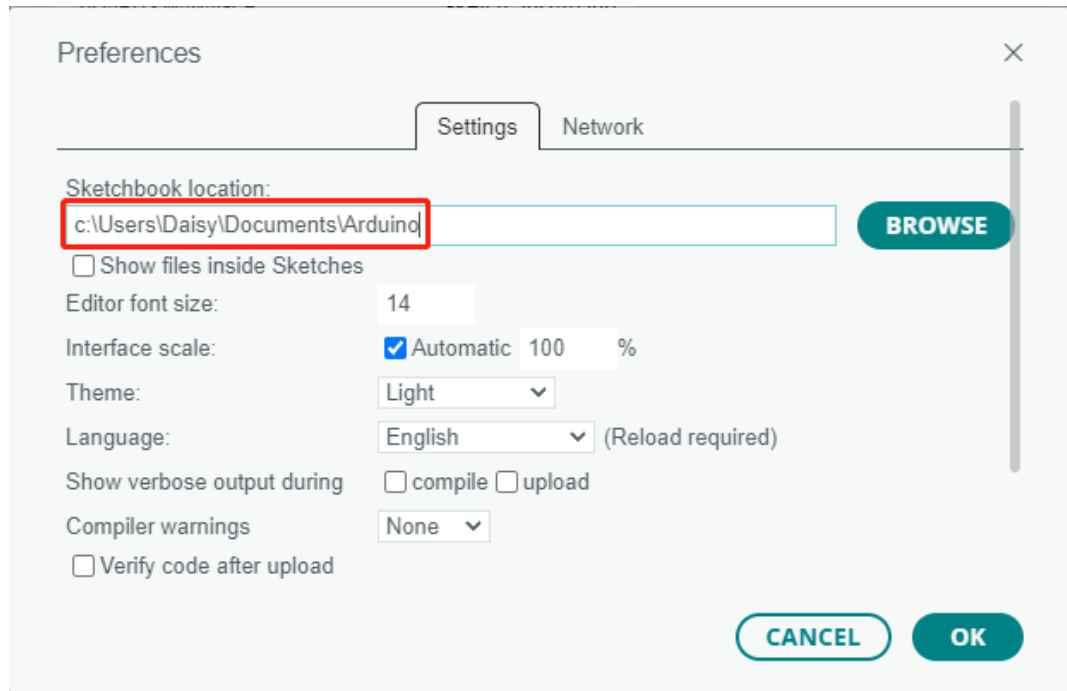
2. Entpacken Sie diese Datei und Sie werden die folgenden Ordner sehen.

..			File folder	
Blynk	1,060,482	459,412	File folder	7/16/2021 1:13 ...
BlynkESP8266_Lib	57,090	11,208	File folder	5/25/2021 5:12 ...
Time	63,774	25,512	File folder	5/25/2021 5:12 ...
TinyGSM	602,241	172,967	File folder	5/25/2021 5:12 ...

3. Kopieren Sie alle und fügen Sie sie in das Standard-Bibliothekenverzeichnis der Arduino IDE ein, welches üblicherweise unter C:\Users\xxx\Documents\Arduino\libraries zu finden ist.



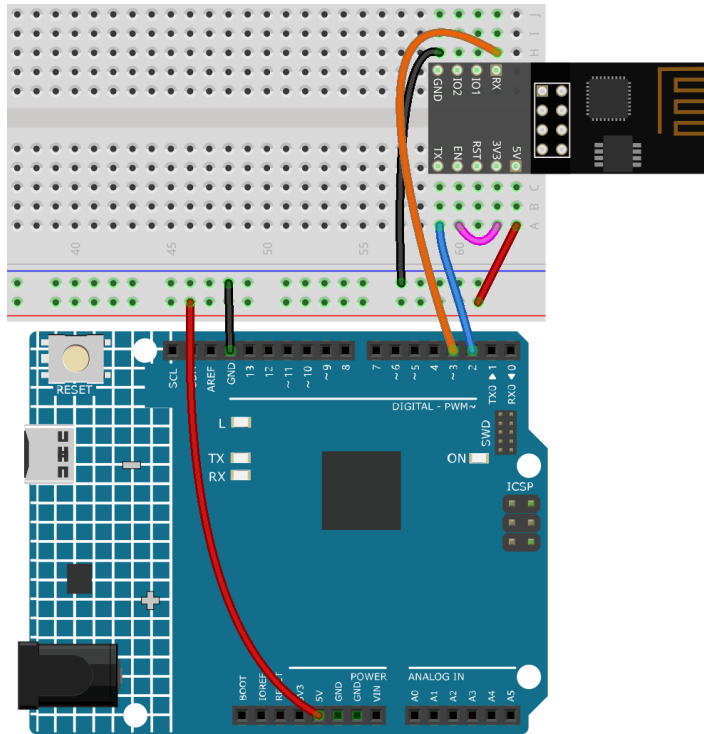
4. Wenn Ihr Bibliothekenverzeichnis anders ist, können Sie es überprüfen, indem Sie auf **File -> Preferences** gehen.



6.1.4 1.4 Verbindung des R4-Boards mit Blynk herstellen

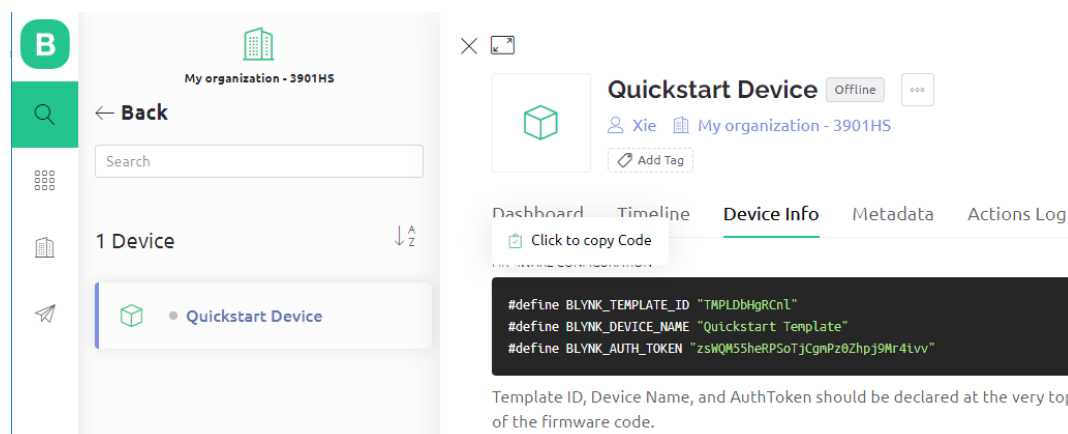
1. Schließen Sie das ESP8266-Modul und das R4-Board erneut an. Hier wird die Software-Serielle verwendet, daher sind TX und RX jeweils an die Pins 2 und 3 des R4-Boards angeschlossen.

Bemerkung: Das ESP8266-Modul benötigt einen hohen Strom, um eine stabile Betriebsumgebung zu gewährleisten. Stellen Sie daher sicher, dass die 9V-Batterie angeschlossen ist.



- Öffnen Sie die Datei 1.connect.ino unter dem Pfad 3in1-kit\iot_project\1.connect. Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Ersetzen Sie die folgenden drei Codezeilen, die Sie von der **Device info**-Seite Ihres Kontos kopieren können. Diese drei Codezeilen ermöglichen es Ihrem R4-Board, Ihr Blynk-Konto zu finden.

```
#define BLYNK_TEMPLATE_ID "TMPLxxxxxx"
#define BLYNK_DEVICE_NAME "Device"
#define BLYNK_AUTH_TOKEN "YourAuthToken"
```

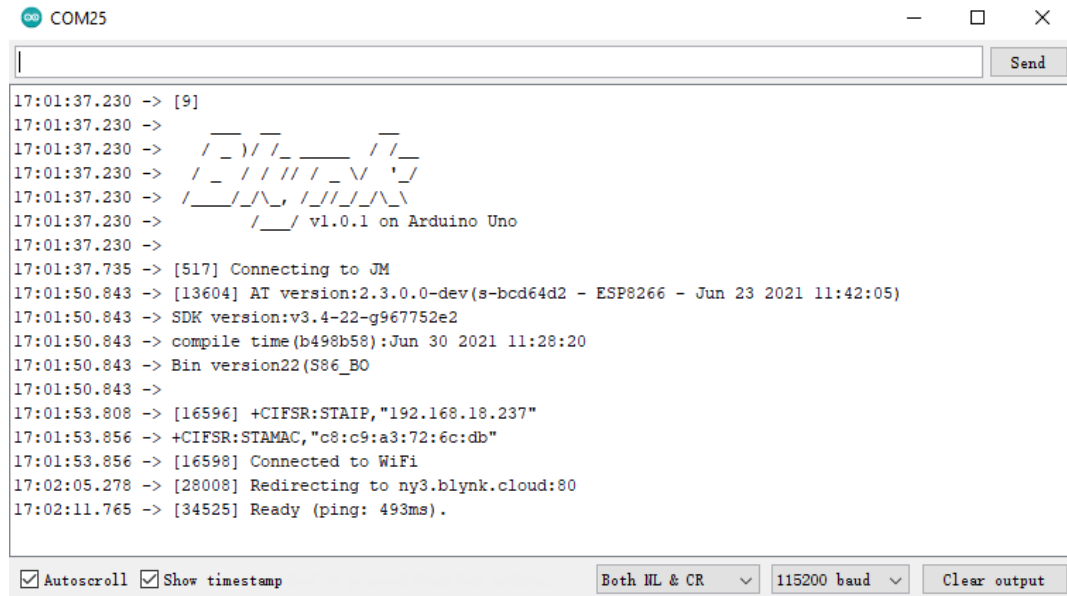


- Tragen Sie die ssid und das password des von Ihnen verwendeten WLANs ein.

```
char ssid[] = "ssid";
char pass[] = "password";
```

- Laden Sie den Code auf das R4-Board, öffnen Sie dann den seriellen Monitor und stellen Sie die Baudrate auf

115200 ein. Wenn das R4-Board erfolgreich mit Blynk kommuniziert, zeigt der serielle Monitor den Schriftzug ready an.



```

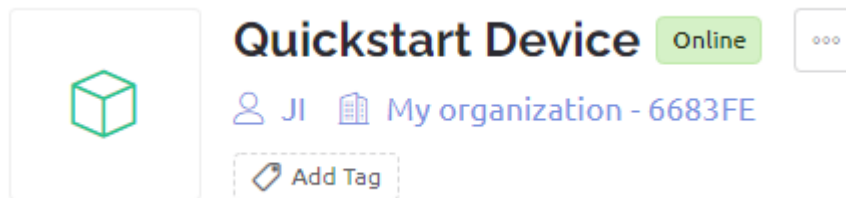
17:01:37.230 -> [9]
17:01:37.230 ->
17:01:37.230 ->  / _ ) / / _ _ _ _ / / _
17:01:37.230 ->  / _ / / / / / _ _ \ ' /
17:01:37.230 ->  / _ / / \ _ / / / / \ _ \
17:01:37.230 ->      / _ / v1.0.1 on Arduino Uno
17:01:37.230 ->
17:01:37.735 -> [517] Connecting to JM
17:01:50.843 -> [13604] AT version:2.3.0.0-dev(s-bcd64d2 - ESP8266 - Jun 23 2021 11:42:05)
17:01:50.843 -> SDK version:v3.4-22-g967752e2
17:01:50.843 -> compile time(b498b58):Jun 30 2021 11:28:20
17:01:50.843 -> Bin version22(S86_BO
17:01:50.843 ->
17:01:53.808 -> [16596] +CIFSR:STAIP,"192.168.18.237"
17:01:53.856 -> +CIFSR:STAMAC,"c8:c9:a3:72:6c:db"
17:01:53.856 -> [16598] Connected to WiFi
17:02:05.278 -> [28008] Redirecting to ny3.blynk.cloud:80
17:02:11.765 -> [34525] Ready (ping: 493ms).
  
```

Bemerkung: Wenn beim Verbinden die Meldung ESP reagiert nicht erscheint, befolgen Sie bitte diese Schritte.

- Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.
- Setzen Sie das ESP8266-Modul zurück, indem Sie den RST-Pin für 1 Sekunde mit GND verbinden und dann trennen.
- Drücken Sie den Reset-Knopf auf dem R4-Board.

Manchmal müssen Sie den obigen Vorgang 3-5 Mal wiederholen. Bitte haben Sie Geduld.

6. Der Status von Blynk wird von **offline** auf **online** wechseln.



6.2 2. Daten von Blynk abrufen

In diesem Kapitel lernen Sie, wie Sie den Schaltkreis mit Blynk steuern können. Lassen Sie uns die LEDs über das Internet einschalten!

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können diese auch einzeln über die untenstehenden Links kaufen.

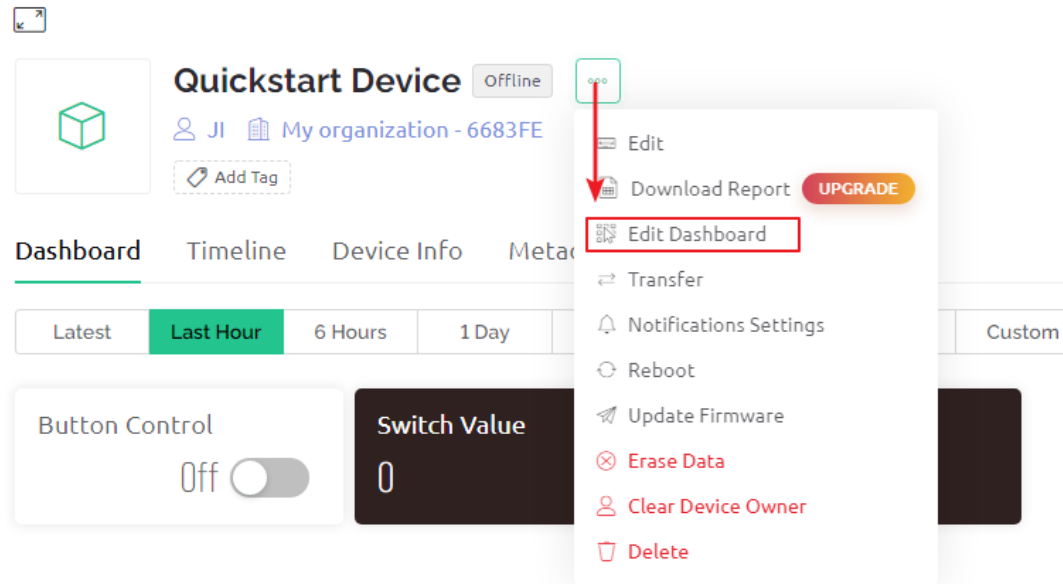
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>ESP8266 Modul</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	

1. Schaltkreis aufbauen

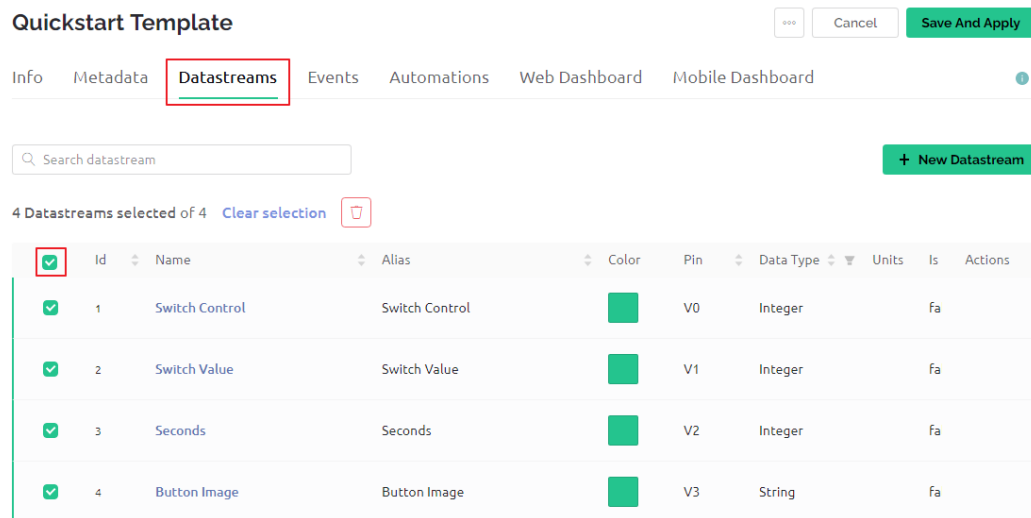
Bemerkung: Das ESP8266-Modul benötigt einen hohen Strom, um eine stabile Betriebsumgebung zu gewährleisten. Stellen Sie daher sicher, dass die 9V-Batterie angeschlossen ist.



1. Gehen Sie zum **Quickstart Device**, das Sie zuvor erstellt haben, klicken Sie auf das Menüsymbol in der oberen rechten Ecke und wählen Sie **edit dashboard**.




2. Datastreams ermöglichen es den Widgets auf Blynk und dem Code auf Ihrem Board, einander zu erkennen. Um den vollständigen Konfigurationsprozess zu erleben, entfernen Sie alle Datastreams von der Datastreams-Seite.



3. Bitte lesen Sie die Warnung sorgfältig und bestätigen Sie, dass sie korrekt ist, bevor Sie die Datastreams löschen.

DANGER ZONE

 Deleting Datastream(s) can lead to breaking changes.

- All widgets using this Datastream(s) will stop working
- All Automation scenarios using this Datastream(s) will stop working
- All active rules in Rules Engine using this Datastream(s) will stop working

This can not be undone.

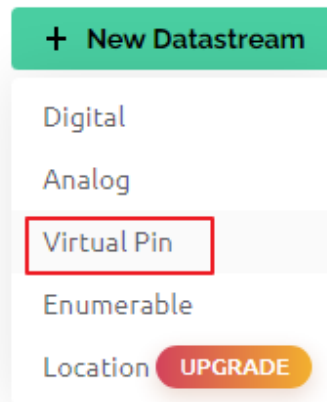
Type DELETE in the field below to proceed with the deletion.

☒ I fully understand that this action is critical and will lead to data loss

4. Erstellen Sie einen Datastream vom Typ **Virtual Pin**, der zum Steuern der LED mit Blynks Schalter verwendet wird.

Datastreams

Datastreams is a way to structure data that regularly flows in and out from device. Use it for sensor data, any telemetry, or actuators.



5. Konfigurieren Sie den **Virtual Pin**. Da der Button und die LED nur EIN oder AUS benötigen, stellen Sie DATENTYP auf Integer und MIN und MAX auf 0 und 1.

Virtual Pin Datastream

NAME: Button Control ALIAS: Button Control

PIN: V0 DATA TYPE: Integer

UNITS: None

MIN: 0 MAX: 1 DEFAULT VALUE: 0

[+ ADVANCED SETTINGS](#) Cancel Save

Region: ny3 [Privacy Policy](#)

6. Gehen Sie zur **Web Dashboard**-Seite und löschen Sie die vorhandenen Widgets.

Quickstart Template Cancel Save And Apply

Info Metadata Datastreams Events Automations Web Dashboard Mobile Dashboard

Widget Box
3 of 30 widgets

CONTROL

Switch ☒

Slider

Number Input

Device name Online Show map UPGRADE

Device Owner Company Name

Tag

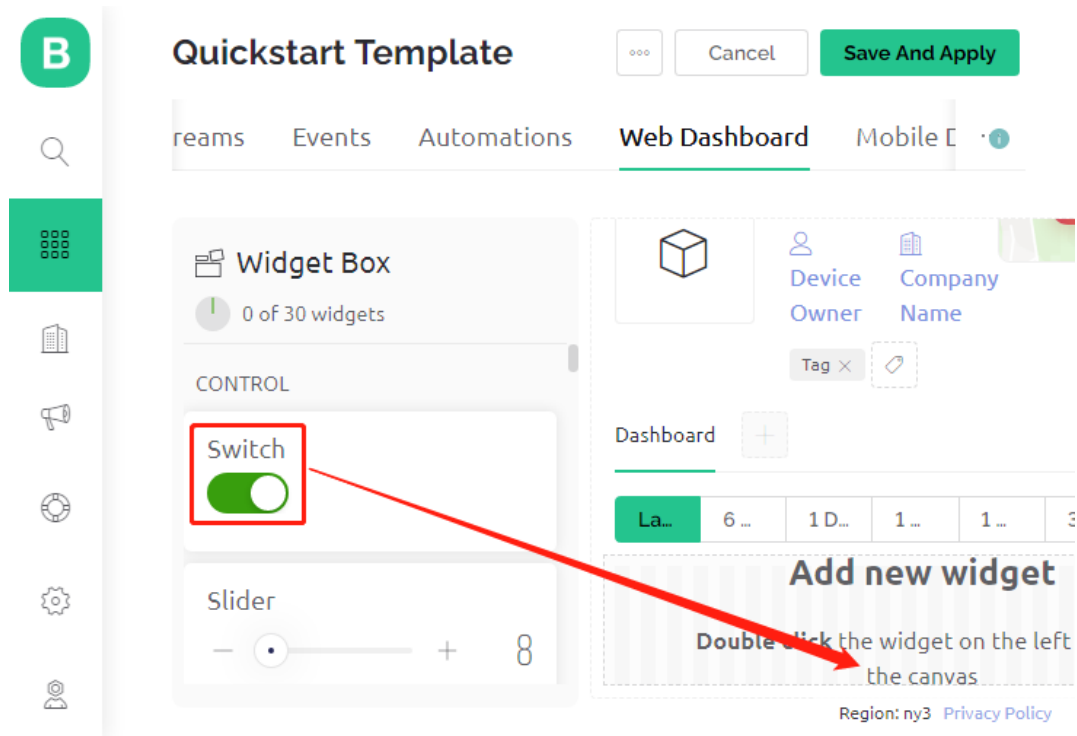
Dashboard

Last H... 6 Hours 1 Day 1 Week 1 Month 3 Mont... Custom

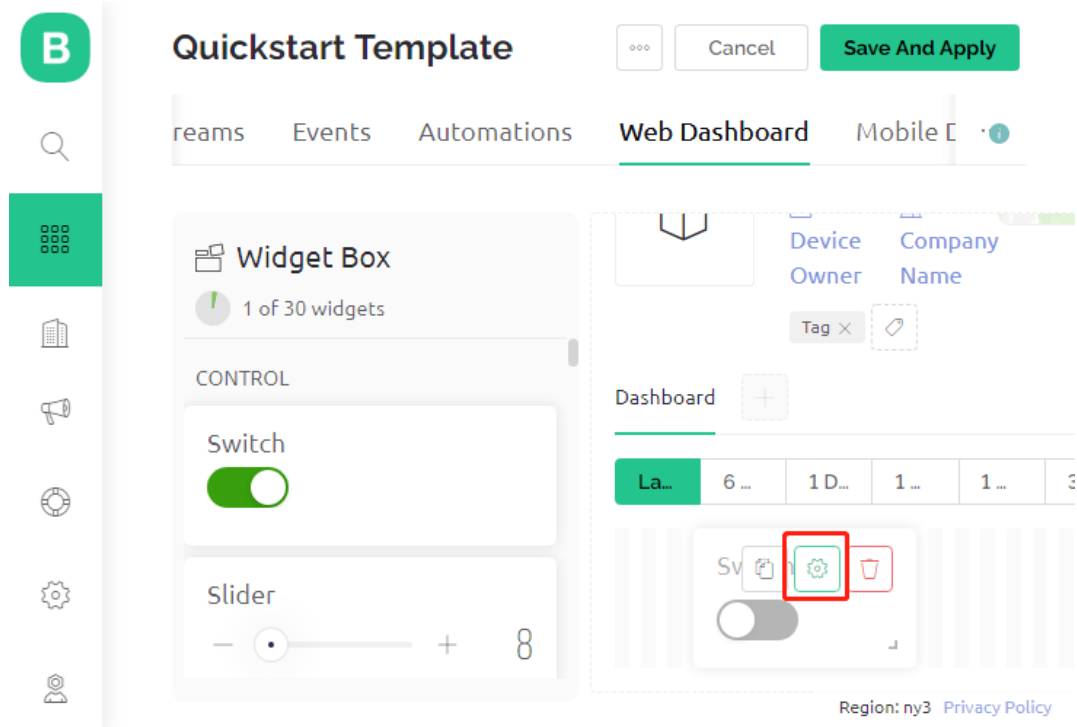
Switch... Uptime

🗑️

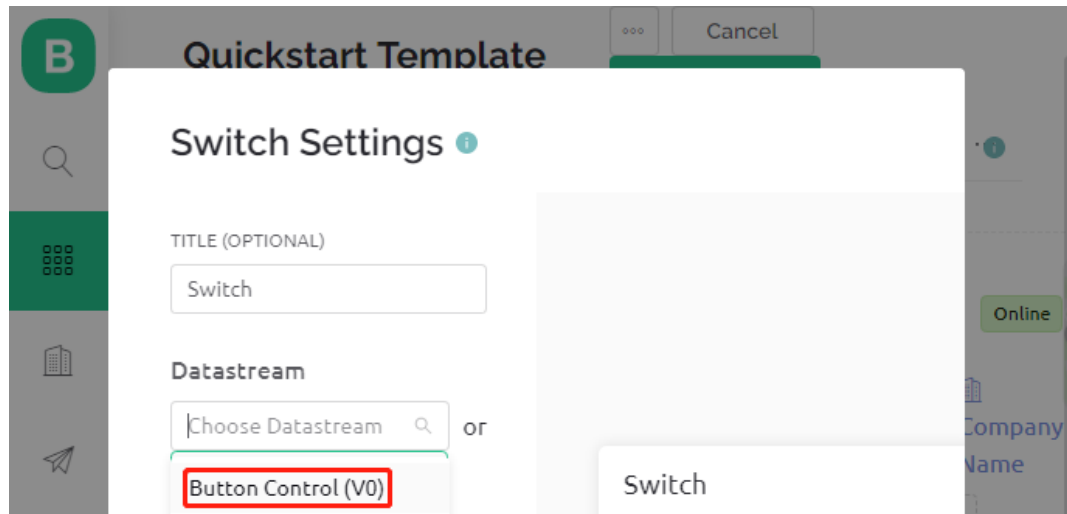
7. Ziehen Sie ein **switch**-Widget aus der **Widget-Box** auf der linken Seite und legen Sie es ab.



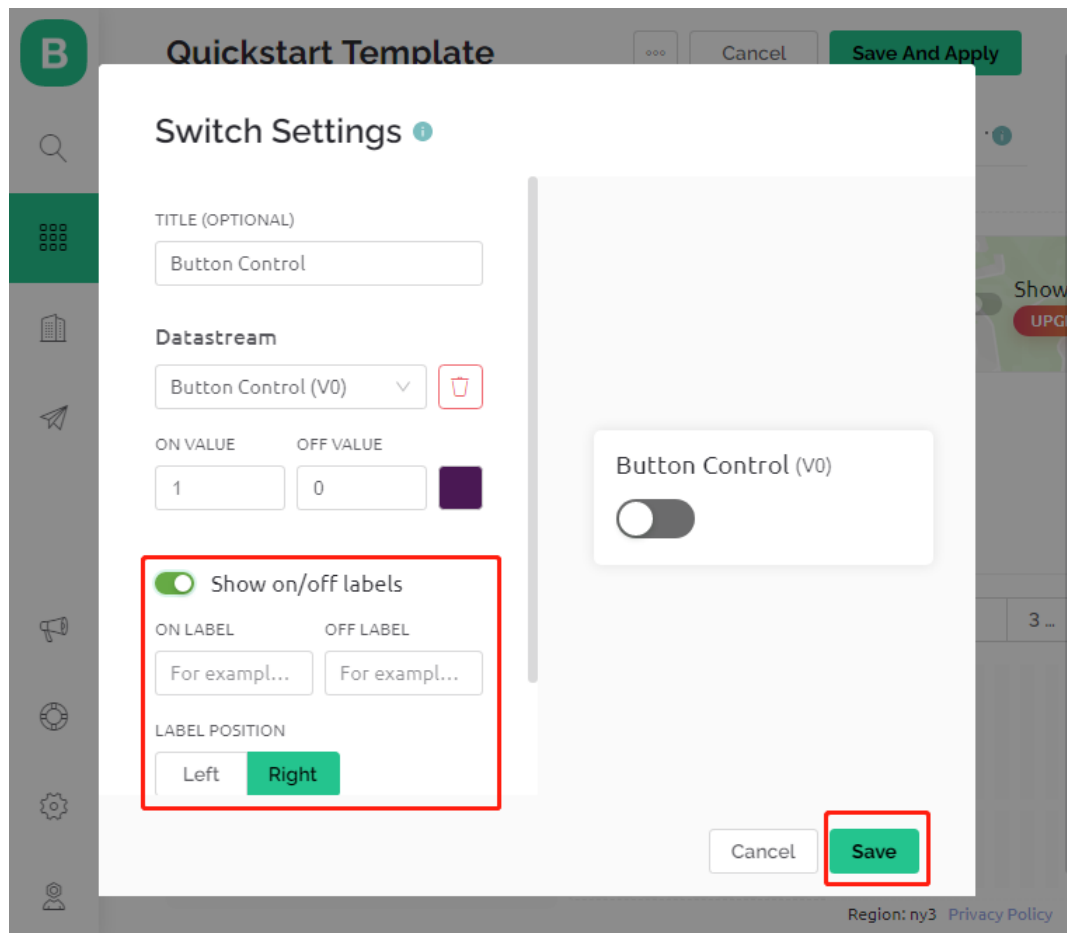
8. Nun zur Einrichtung.



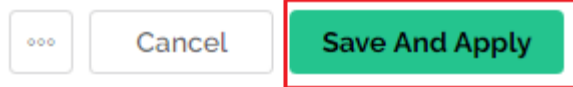
9. Wählen Sie **Datastream** als den zuvor eingerichteten aus.



10. Nachdem Sie Datastream ausgewählt haben, sehen Sie einige benutzerdefinierte Einstellungen. Drücken Sie dann auf Speichern.

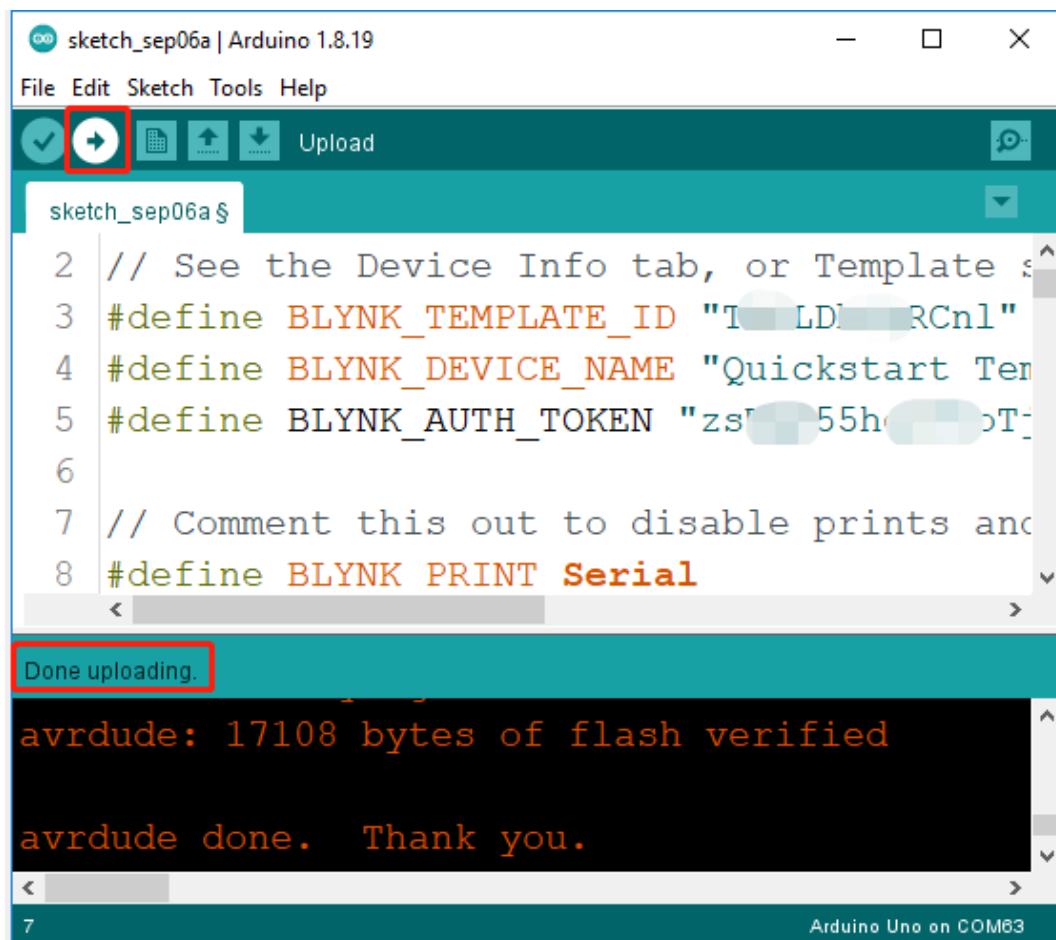


11. Klicken Sie abschließend auf **Save And Apply**.



3. Den Code ausführen

1. Öffnen Sie die Datei `2.get_data_from_blynk.ino` im Pfad `3in1-kit\iot_project\2.get_data_from_blynk` oder kopieren Sie diesen Code in die **Arduino IDE**.
2. Ersetzen Sie die Template ID, den Device Name und den Auth Token durch Ihre eigenen. Sie müssen auch die ssid und das password Ihres WLANs eingeben. Für detaillierte Anleitungen verweisen Sie bitte auf [1.4 Verbindung des R4-Boards mit Blynk herstellen](#).
3. Wählen Sie das richtige Board und den richtigen Port aus und klicken Sie auf den **Upload**-Button.



4. Öffnen Sie den seriellen Monitor (Baudrate auf 115200 einstellen) und warten Sie auf eine Meldung wie eine erfolgreiche Verbindung.


```

COM63

#StandWithUkraine    https://bit.ly/swua

[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"1...1...75"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).

☒ Autoscroll ☐ Show timestamp
Both NL & CR 115200 baud Clear output

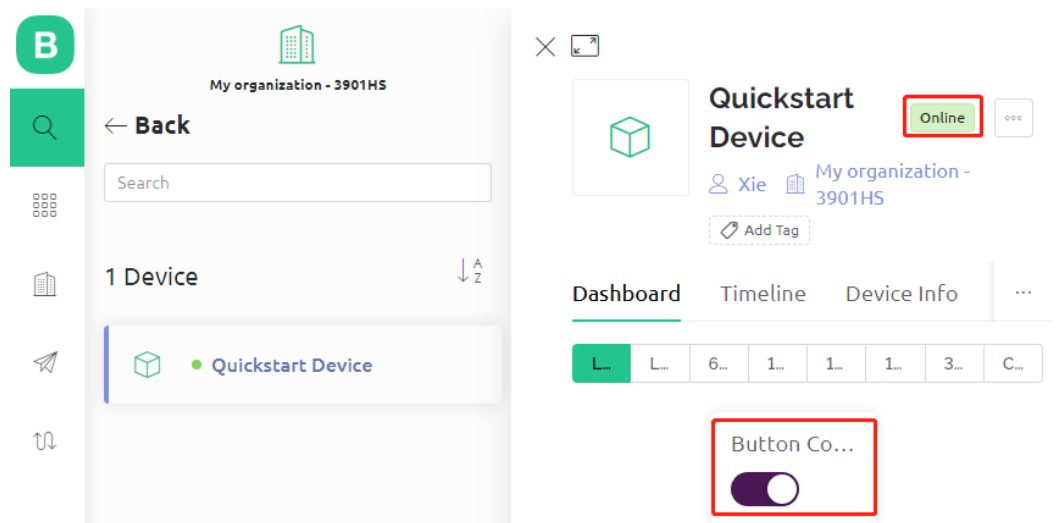
```

Bemerkung: Wenn die Meldung ESP is not responding erscheint, befolgen Sie bitte diese Schritte.

- Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.
- Setzen Sie das ESP8266-Modul zurück, indem Sie den Pin RST 1 Sekunde lang auf GND verbinden und dann trennen.
- Drücken Sie die Reset-Taste auf Ihrem Board.

Manchmal müssen Sie den oben genannten Vorgang 3-5 Mal wiederholen. Bitte haben Sie Geduld.

5. Zurück bei Blynk können Sie sehen, dass der Status auf online geändert wurde und Sie können nun das Schalter-Widget auf Blynk verwenden, um die LED auf dem R4-Board zu steuern.



6. Wenn Sie Blynk auf mobilen Geräten verwenden möchten, verweisen Sie bitte auf *Wie verwendet man Blynk auf einem Mobilgerät?*.

Wie funktioniert das?

Der Unterschied zwischen dem Code in diesem Projekt und dem Code im vorherigen Kapitel *1.4 Verbindung des R4-Boards mit Blynk herstellen* sind die folgenden Zeilen.

```
const int ledPin=6;

BLYNK_WRITE(V0)
{
    int pinValue = param.asInt(); // assigning incoming value from pin V0 to a variable
    // You can also use:
    // String i = param.asStr();
    // double d = param.asDouble();
    digitalWrite(ledPin,pinValue);
}

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

Bezüglich des `pinMode` und des `digitalWrite` des `ledPin` bin ich sicher, dass Sie bereits vertraut sind, also werde ich sie nicht noch einmal durchgehen. Worauf Sie sich konzentrieren sollten, ist die Funktion `BLYNK_WRITE(V0)`.

Was es tun wird: Wenn sich der Wert von Blynks `V0` ändert, wird Blynk.Cloud Ihrem Gerät mitteilen: „Ich schreibe an **Virtual Pin** `V0`“, und Ihr Gerät wird in der Lage sein, etwas auszuführen, sobald es diese Information erhält.

Wir haben im vorherigen Schritt den `V0` Datastream erstellt und ihn dem Schalter-Widget zugewiesen. Das bedeutet, dass jedes Mal, wenn wir das Schalter-Widget betätigen, `BLYNK_WRITE(V0)` ausgelöst wird.

Wir schreiben zwei Anweisungen in diese Funktion.

```
int pinValue = param.asInt();
```

Holen Sie sich den Wert von `V0` und weisen Sie ihn der Variable `pinValue` zu.

```
digitalWrite(ledPin,pinValue);
```

Schreiben Sie den erlangten Wert von V0 auf den ledPin, damit das Schalter-Widget auf Blynk die LED steuern kann.

6.3 3. Daten an Blynk senden

In diesem Kapitel zeigen wir Ihnen, wie Sie Daten an Blynk senden können.

Wir erstellen hier ein Tür- und Fenstererkennungsgesät. Die Schaltung mit dem Reed-Kontakt wird neben der Tür und dem Fenster angebracht, und der Magnet wird am Rand der Tür bzw. des Fensters montiert. Wenn die Tür oder das Fenster geschlossen ist, wird der Reed-Kontakt durch die Magnetkraft eingeschaltet und der entsprechende Pin-Wert auf dem R4-Board ändert sich. Blynk.cloud wird diesen Wert empfangen, sodass Sie auch von unterwegs aus sehen können, ob die Türen und Fenster Ihres Hauses geschlossen sind.

Nun werden wir ein LED-Widget in Blynk verwenden, um anzuzeigen, ob Ihre Fenster und Türen geschlossen sind (d.h. ob der Reed-Kontakt ein- oder ausgeschaltet ist).

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist sicherlich praktisch, ein komplettes Set zu kaufen, hier ist der Link:

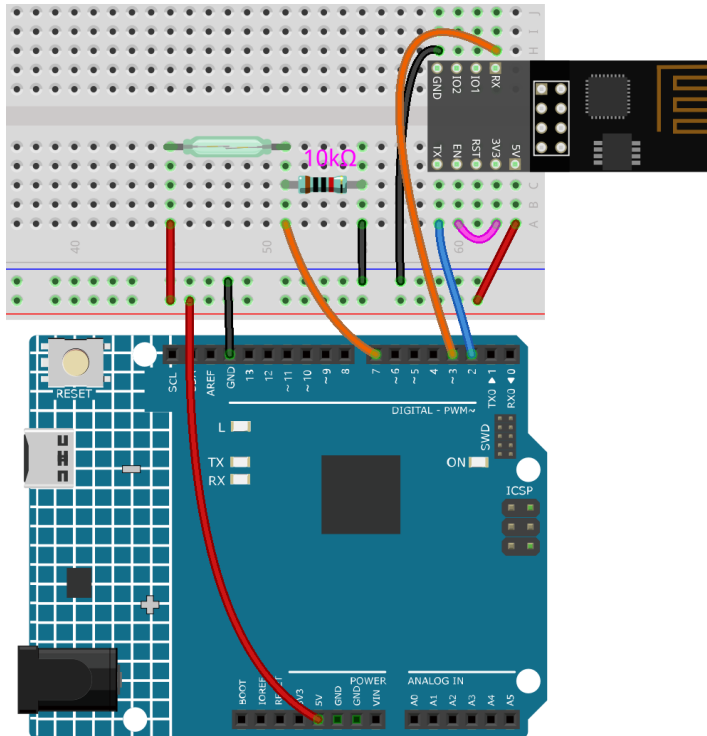
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>ESP8266 Modul</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Reedschalter</i>	-

1. Schaltung aufbauen

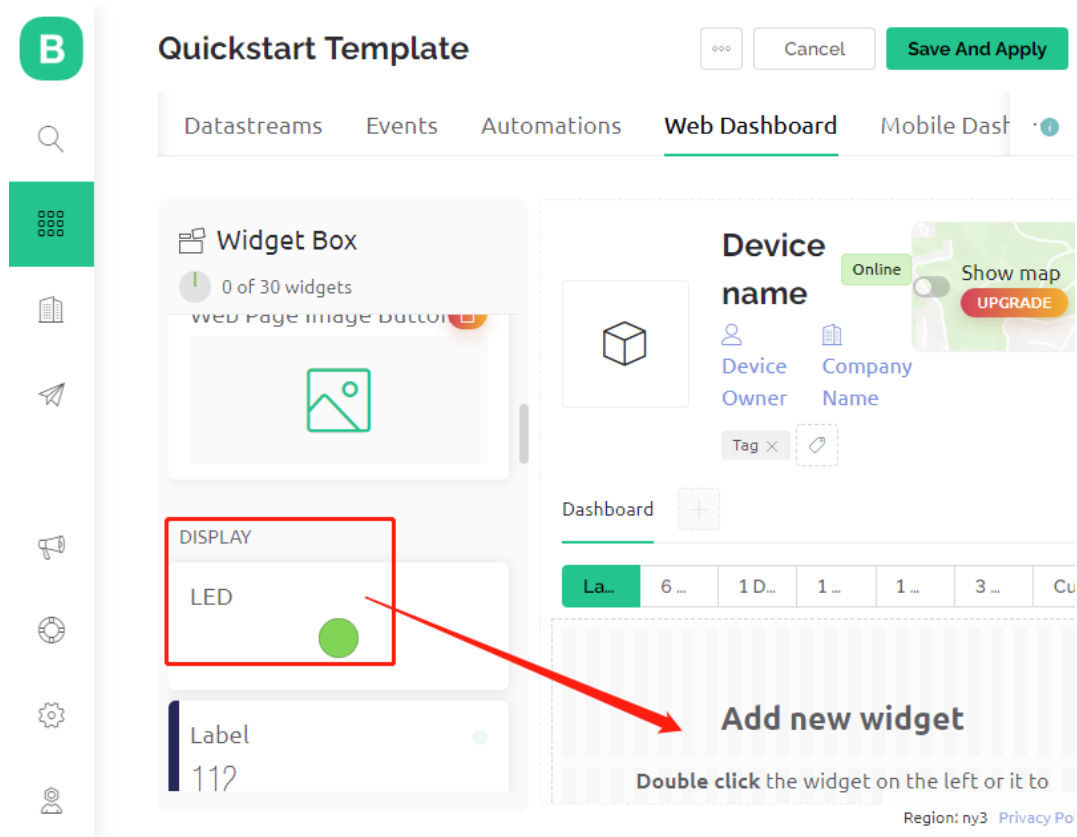
Bemerkung: Das ESP8266-Modul benötigt einen hohen Strom, um einen stabilen Betrieb zu gewährleisten. Stellen Sie daher sicher, dass die 9V-Batterie angeschlossen ist.



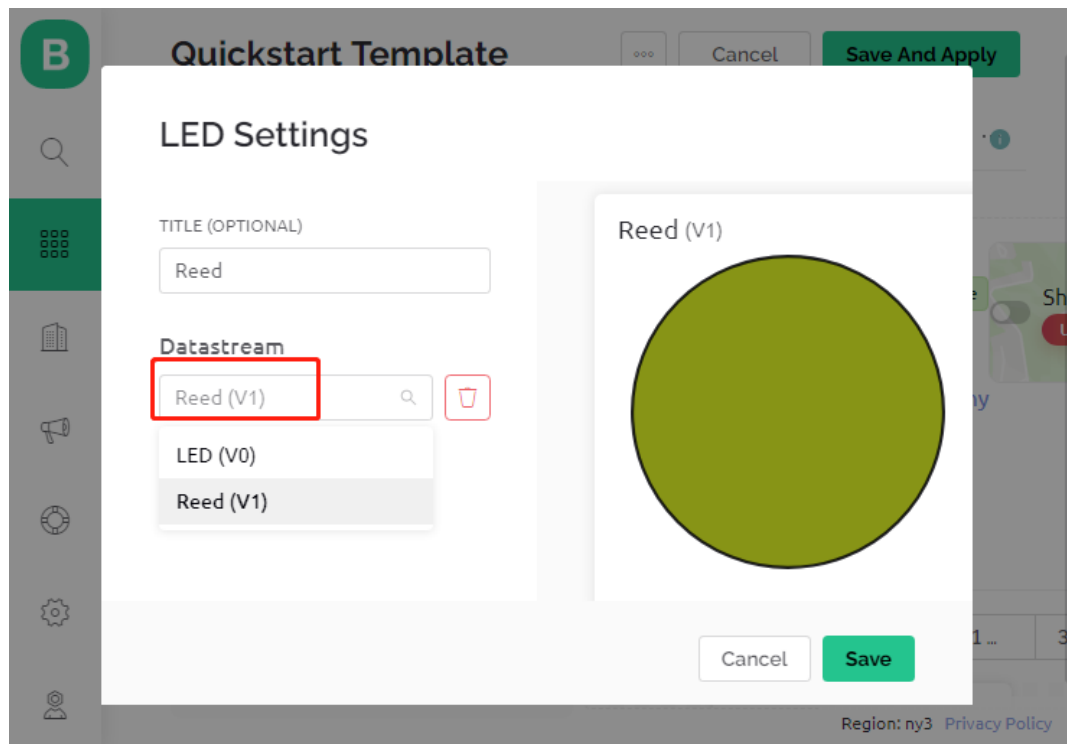
2. Dashboard bearbeiten

1. Erstellen Sie einen **Datastream** vom Typ **Virtual Pin** auf der **Datastream**-Seite, um den Wert des Reed-Kontakts zu erhalten. Stellen Sie den DATENTYP auf **Integer** und MIN und MAX auf **0** und **1**.

2. Ziehen Sie ein **LED widget** auf die **Wed Dashboard**-Seite. Bei einem Wert von 1 leuchtet es (farbig), sonst ist es weiß.

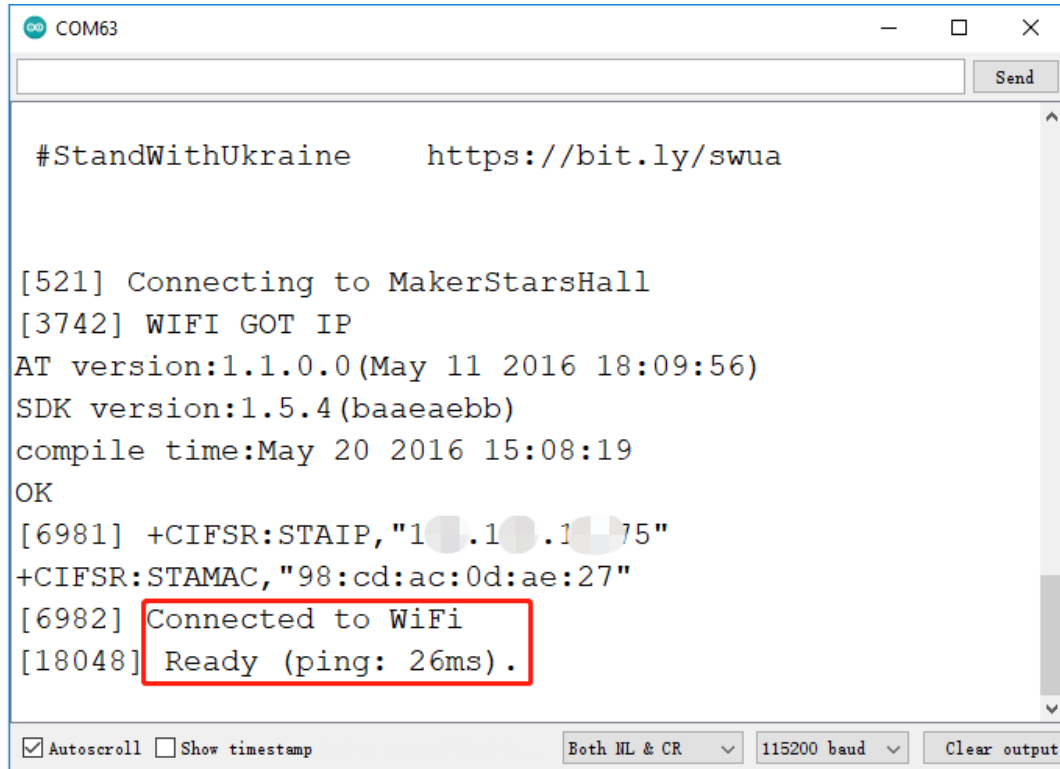


3. Wählen Sie auf der Einstellungsseite des **LED widget** als **DataStream Reed(V1)** aus und speichern Sie es.



3. Den Code ausführen

1. Öffnen Sie die Datei `3.push_data_to_blynk.ino` unter dem Pfad `3in1-kit\iot_project\3.push_data_to_blynk` oder kopieren Sie diesen Code in die **Arduino IDE**.
2. Ersetzen Sie die Template ID, Device Name und Auth Token durch Ihre eigenen Angaben. Sie müssen auch die ssid und das password Ihres verwendeten WLANs eingeben. Für detaillierte Anleitungen beachten Sie bitte [1.4 Verbindung des R4-Boards mit Blynk herstellen](#).
3. Wählen Sie das korrekte Board und den Port aus und klicken Sie auf den **Upload**-Button.
4. Öffnen Sie den Seriellen Monitor (Baudrate auf 115200 einstellen) und warten Sie auf eine Meldung, z.B. eine erfolgreiche Verbindung.



```
#StandWithUkraine https://bit.ly/swua

[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"192.168.1.175"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

Bemerkung: Wenn die Meldung `ESP is not responding` erscheint, befolgen Sie bitte diese Schritte:

- Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.
- Setzen Sie das ESP8266-Modul zurück, indem Sie den Pin RST für 1 Sekunde mit GND verbinden, dann trennen Sie ihn.
- Drücken Sie den Reset-Knopf auf dem R4-Board.

Manchmal müssen Sie den oben genannten Vorgang 3-5 Mal wiederholen, bitte haben Sie Geduld.

5. Nun zeigt Blynk den Status Ihrer Türen und Fenster an. Wenn Ihre Türen und Fenster geschlossen sind, wird das LED-Widget grün sein, sonst grau.
6. Wenn Sie Blynk auf mobilen Geräten nutzen möchten, beachten Sie bitte [Wie verwendet man Blynk auf einem Mobilgerät?](#).

Wie funktioniert das?

Für dieses Beispiel sollten Sie sich auf die folgenden Zeilen konzentrieren. „Daten jede Sekunde an den Blynk Cloud V1 Datastream senden“ wird durch diese Zeilen definiert.

```
BlynkTimer timer;

void myTimerEvent()
{
    Blynk.virtualWrite(V1, pinValue);
}

void setup()
{
    timer.setInterval(1000L, myTimerEvent);
}

void loop()
{
    timer.run(); // Initiates BlynkTimer
}
```

Die Blynk-Bibliothek stellt einen eingebauten Timer zur Verfügung, zuerst erstellen wir ein Timer-Objekt.

```
BlynkTimer timer;
```

Legen Sie das Timer-Intervall in `setup()` fest. Hier setzen wir es so, dass die Funktion `myTimerEvent()` alle 1000ms ausgeführt wird.

```
timer.setInterval(1000L, myTimerEvent);
```

Führen Sie `BlynkTimer` in `loop()` aus.

```
timer.run();
```

Bearbeiten Sie die benutzerdefinierte Funktion `myTimerEvent()`. Der Code `Blynk.virtualWrite(V1, pinValue)` dient dazu, den Datenpin-Wert für V1 zu schreiben.

```
void myTimerEvent()
{
    Blynk.virtualWrite(V1, pinValue);
}
```

6.4 4. Cloud-Musikspieler

Das Ziel dieses Projekts ist es, einen Musikspieler mit Blynk zu erstellen. Die Musik wird auf dieselbe Weise wie in *5.7 Tone() oder noTone()* abgespielt, indem das Lied im Programm geschrieben und mit einem passiven Summer abgespielt wird. In diesem Beispiel können wir jedoch den Schalter klicken, um die Wiedergabe zu starten/anhalten und den Schieberegler verschieben, um den Wiedergabefortschritt zu ändern.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

SunFounder 3in1 Kit

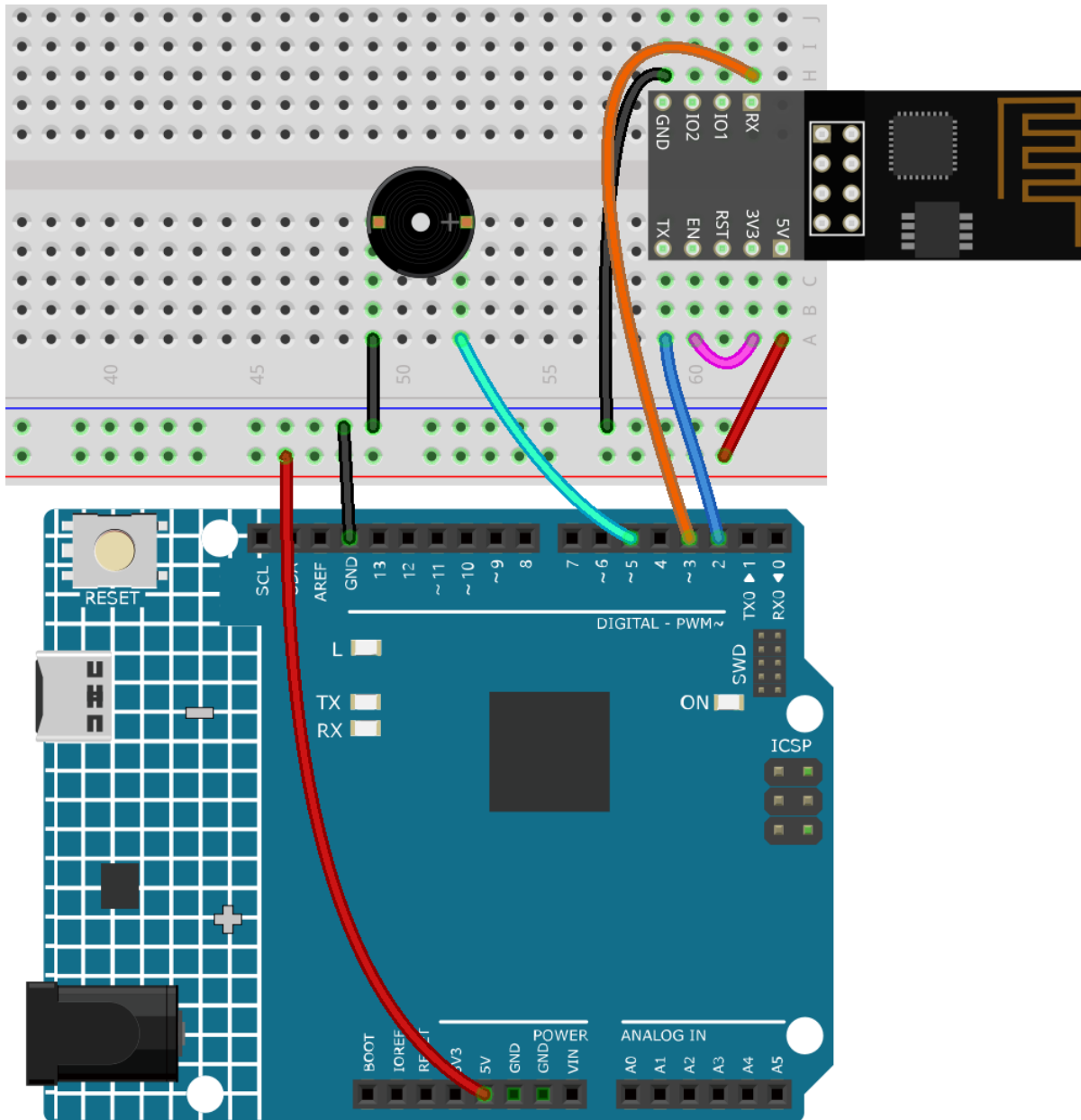
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>ESP8266 Modul</i>	
<i>Jumper-Kabel</i>	
<i>Summer</i>	

1. Schaltkreis aufbauen

Bemerkung: Das ESP8266-Modul erfordert einen hohen Strom, um eine stabile Betriebsumgebung zu gewährleisten. Stellen Sie daher sicher, dass die 9V-Batterie angeschlossen ist.



2. Dashboard bearbeiten

1. Erstellen Sie einen **Datastream** vom Typ **Virtual Pin** auf der **Datastream**-Seite als den vom später hinzugefügten Slider-Widget oder Code geänderten Wert. Setzen Sie den DATENTYP auf **Integer** und MIN und MAX auf **0** und **30**.

Virtual Pin Datastream

NAME: Slider ALIAS: Slider

PIN: V2 DATA TYPE: Integer

UNITS: None

MIN: 0 MAX: 30 DEFAULT VALUE: 0

+ ADVANCED SETTINGS

Cancel Create

Region: ny3 Privacy Policy

- Erstellen Sie auch einen anderen **Datastream** vom Typ **Virtual Pin**, um den Musiknamen anzuzeigen, und setzen Sie den DATENTYP auf String.

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATA TYPE:

DEFAULT VALUE:

Region: ny3 [Privacy Policy](#)

3. Gehen Sie zur **Wed Dashboard**-Seite, ziehen Sie ein **Switch**-Widget und setzen Sie **Datastream** auf V0 (V0 ist bereits in 2. *Daten von Blynk abrufen* eingestellt); ziehen Sie ein **Label**-Widget und setzen Sie es auf V3; ziehen Sie ein **Slider**-Widget und setzen Sie es auf V2.

Widget Box
3 of 30 widgets

Chart

Map

Device name Online
Device Owner Company Name
Tag

Dashboard

Last Hour 6 Hours 1 Day 1 Week 1 Month

Button Co... (V0)

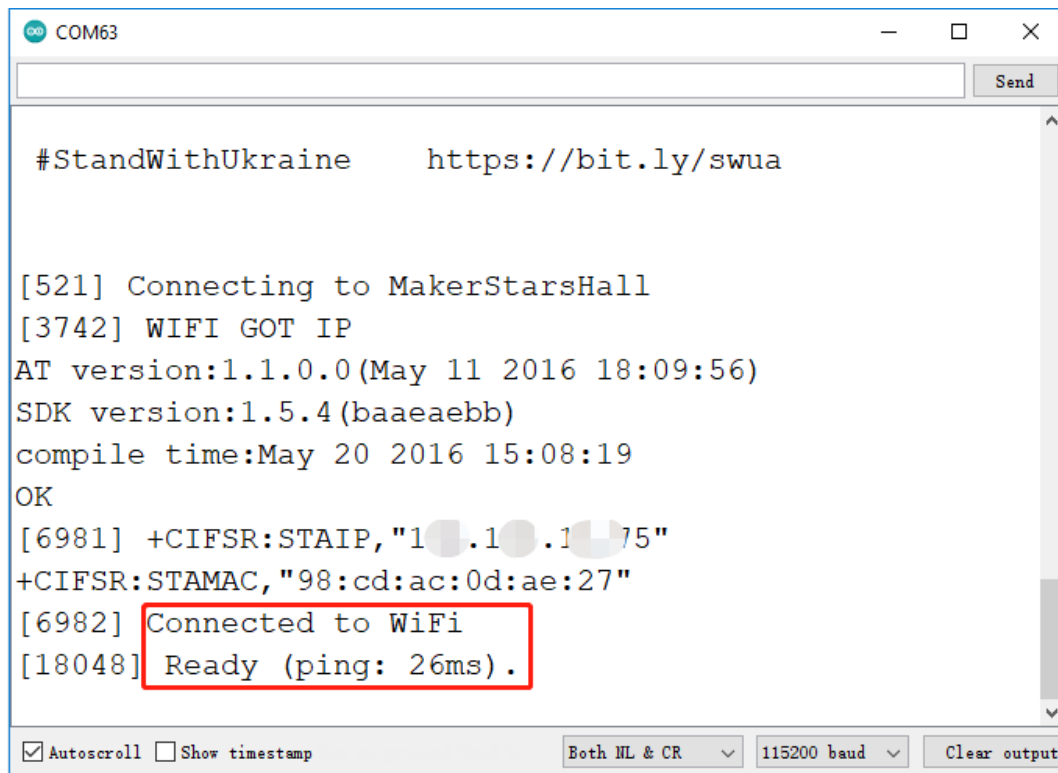
Slider (V2)

Song Name (V3)
String

Bemerkung: Ihre virtuellen Pins könnten sich von meinen unterscheiden, Ihre haben Vorrang, aber Sie müssen die entsprechende Pin-Nummer im Code ändern.

3. Code ausführen

1. Öffnen Sie die Datei `4.cloud_music_player.ino` unter dem Pfad `3in1-kit\iot_project\4.cloud_music_player`.
2. Ersetzen Sie die Template ID, Device Name und Auth Token durch Ihre eigenen Angaben. Sie müssen auch die ssid und das password des von Ihnen verwendeten WLANs eingeben. Für detaillierte Anleitungen verweisen Sie bitte auf [1.4 Verbindung des R4-Boards mit Blynk herstellen](#).
3. Wählen Sie das richtige Board und den Port aus und klicken Sie auf die **Upload**-Taste.
4. Öffnen Sie den Seriellen Monitor (Baudrate auf 115200 einstellen) und warten Sie auf eine Meldung, wie eine erfolgreiche Verbindung.



```
#StandWithUkraine https://bit.ly/swua

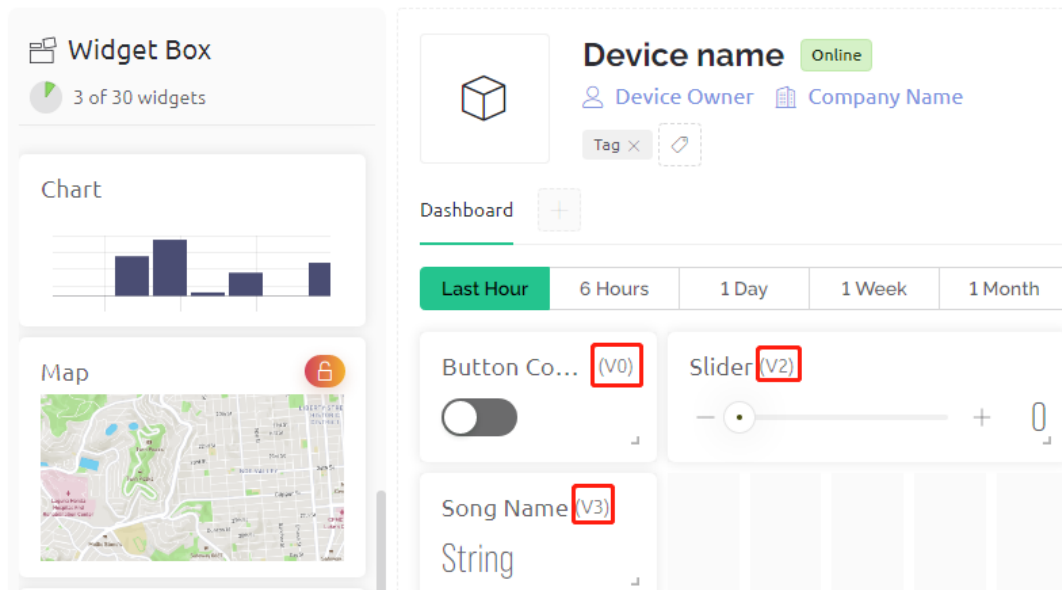
[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"1...1...75"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

Bemerkung: Wenn die Meldung `ESP is not responding` beim Verbinden erscheint, befolgen Sie bitte diese Schritte.

- Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.
- Setzen Sie das ESP8266-Modul zurück, indem Sie den Pin RST 1 Sekunde lang auf GND legen und dann abziehen.
- Drücken Sie die Reset-Taste auf dem R4-Board.

Manchmal müssen Sie den obigen Vorgang 3-5 Mal wiederholen. Bitte haben Sie Geduld.

5. Nun können Sie Blynks Button Control Widget verwenden, um die Musik zu starten/pausieren und den Slider, um den Wiedergabefortschritt anzupassen. Sie sehen auch den Namen des Musikstücks.



6. Wenn Sie Blynk auf mobilen Geräten verwenden möchten, verweisen Sie bitte auf [Wie verwendet man Blynk auf einem Mobilgerät?](#).

Wie funktioniert das?

Der Datenstrom **V0** wird verwendet, um den Status des Switch-Widgets abzurufen und der Variable **musicPlayFlag** zuzuweisen, die das Pausieren und Abspielen der Musik steuert.

```
int musicPlayFlag=0;

BLYNK_WRITE(V0)
{
    musicPlayFlag = param.asInt(); // START/PAUSE MUSIC
}
```

Der Datenstrom **V2** wird verwendet, um den Wert des Slider-Widgets abzurufen und ihn der Variable **scrubBar** zuzuweisen, wenn der Slider verschoben wird.

```
int scrubBar=0;

BLYNK_WRITE(V2)
{
    scrubBar=param.asInt();
}
```

Wenn das Gerät mit der **Blynk Cloud** verbunden ist, schreiben Sie den Musiknamen für den Datenstrom **V3** und zeigen Sie ihn dann mit dem **Label**-Widget an.

```
BLYNK_CONNECTED() {
    String songName = "Ode to Joy";
    Blynk.virtualWrite(V3, songName);
}
```

Der **Blynk Timer** wird jede Sekunde ausgeführt. Die Musik wird abgespielt, wenn **musicPlayFlag** nicht 0 ist, d.h. das

Switch-Widget ist EIN. Sobald zwei Noten gespielt werden, wird die Variable für die Fortschrittsleiste `scrubBar` um 2 erhöht, und der Wert wird dann in die **Blynk Cloud** geschrieben, die den Wert des **Slider**-Widgets synchronisiert.

```
void myTimerEvent()
{
    if(musicPlayFlag!=0)
    {
        tone(buzzerPin,melody[scrubBar],250);
        scrubBar=(scrubBar+1)%(sizeof(melody)/sizeof(int));
        delay(500);
        tone(buzzerPin,melody[scrubBar],250);
        scrubBar=(scrubBar+1)%(sizeof(melody)/sizeof(int));
        Serial.println(scrubBar);
        Blynk.virtualWrite(V2, scrubBar);
    }
}
```

6.5 5. Überwachung des häuslichen Umfelds

In diesem Kapitel werden wir Blynk verwenden, um einen Umweltmonitor für das Zuhause zu erstellen. Sie können mit dem DHT11 und dem Fotowiderstand die Temperatur, die Luftfeuchtigkeit und die Lichtintensität eines Raumes messen. Indem Sie diese Werte an Blynk senden, können Sie die Umgebung Ihres Zuhauses über das Internet überprüfen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein gesamtes Set zu kaufen, hier ist der Link:

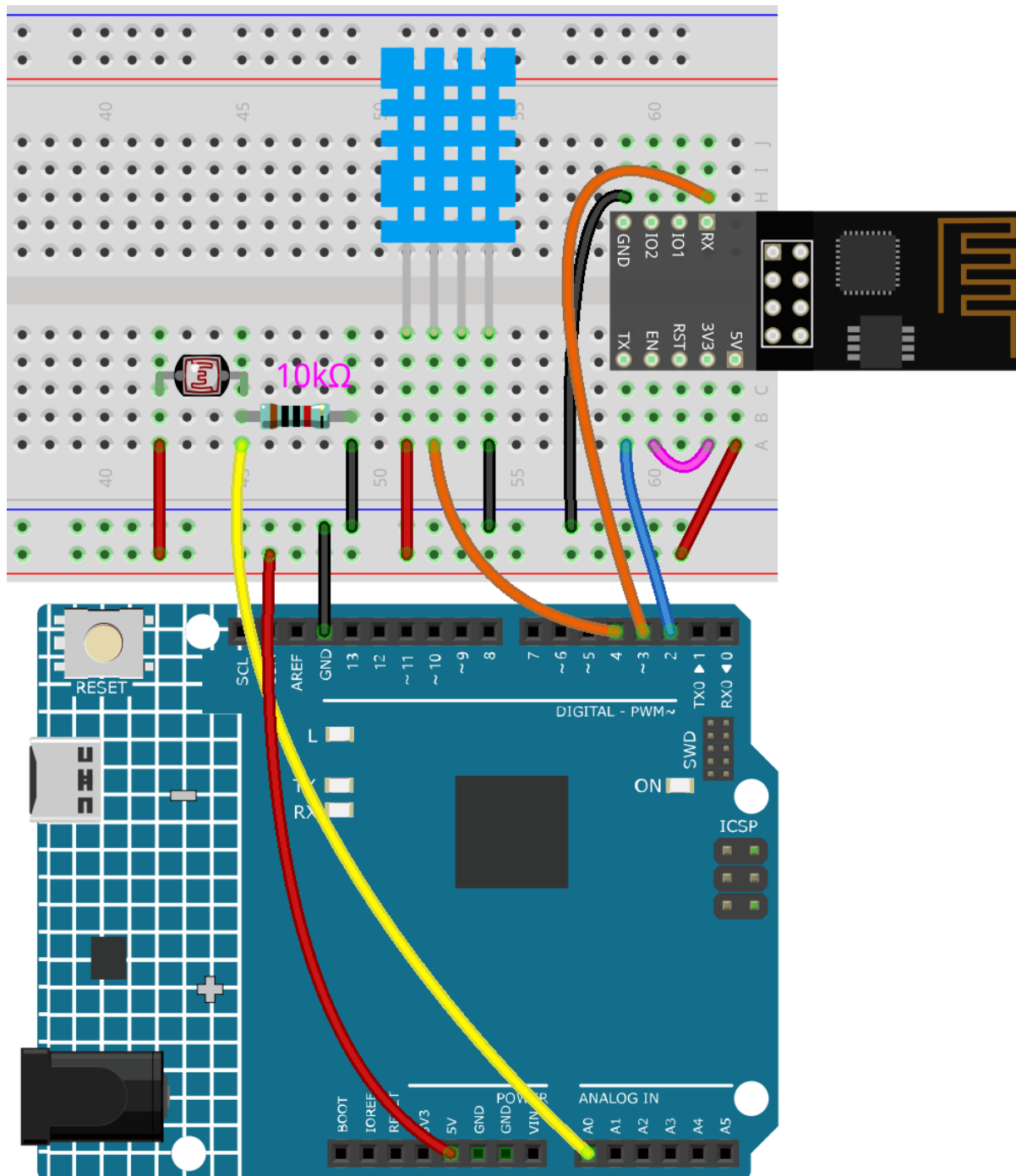
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können diese auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>ESP8266 Modul</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Fotowiderstand</i>	
<i>DHT11 Feuchtigkeits- und Temperatursensor</i>	-

1. Schaltung aufbauen

Bemerkung: Das ESP8266-Modul benötigt einen hohen Strom, um einen stabilen Betrieb zu gewährleisten. Stellen Sie also sicher, dass die 9V-Batterie angeschlossen ist.



2. Dashboard bearbeiten

1. Für die Aufzeichnung von Luftfeuchtigkeitswerten erstellen Sie einen **Datastream** vom Typ **Virtual Pin** auf der **Datastream**-Seite. Setzen Sie den DATENTYP auf **Double** und MIN und MAX auf **0** und **100**. Setzen Sie auch die Einheiten auf **Percentage, %**.

Quickstart Template

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATA TYPE:

UNITS:

MIN: MAX: DECIMALS: DEFAULT VALUE:

[+ ADVANCED SETTINGS](#)

Region: ny3 [Privacy Policy](#)

- Erstellen Sie dann einen **Datastream** vom Typ **Virtual Pin** für die Temperaturaufzeichnung. Setzen Sie DATENTYP auf Double, MIN und MAX auf -30 und 50, und die Einheiten auf **Celsius, °C**.

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATA TYPE:

UNITS:

MIN: MAX: DECIMALS: DEFAULT VALUE:

Region: ny3 [Privacy Policy](#)

- Erstellen Sie auch einen **Datastream** vom Typ **Virtual Pin** zur Aufzeichnung der Lichtintensität. Verwenden Sie den Standarddatentyp - **Integer**, mit MIN und MAX auf 0 und 1024 eingestellt.

Quickstart Templa...

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATA TYPE:

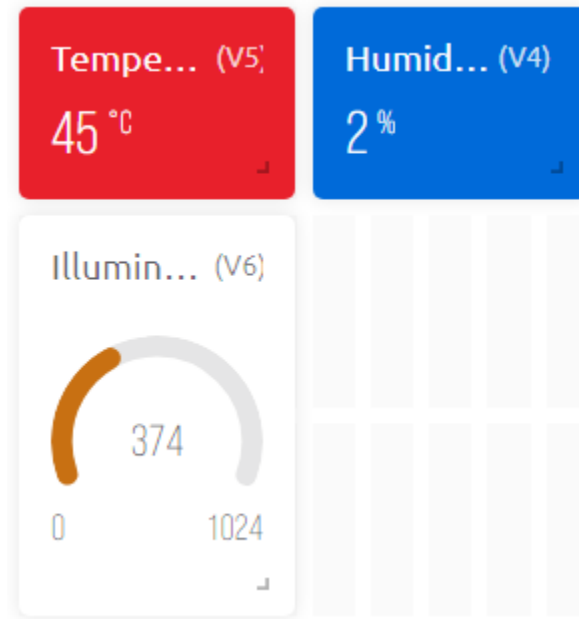
UNITS:

MIN: MAX: DEFAULT VALUE:

[+ ADVANCED SETTINGS](#)

Region: ny3 [Privacy Policy](#)

4. Gehen Sie zur **Wed Dashboard**-Seite, ziehen Sie zwei **Label**-Widgets und setzen Sie ihre Datenströme jeweils auf **V4** und **V5**. Ziehen Sie ein **Gauge**-Widget und setzen Sie den Datenstrom auf **V6**. In den Widget-Einstellungen können Sie **Farbänderung basierend auf Wert** aktivieren und die entsprechende Farbe auswählen, um das Widget ansprechender und intuitiver zu gestalten.

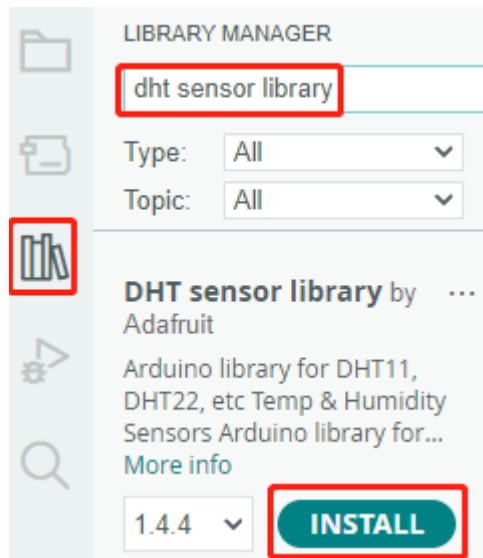


3. Den Code ausführen

1. Öffnen Sie die Datei `5.home_environment_monitoring.ino` unter dem Pfad `3in1-kit\iot_project\5.home_environment_monitoring` oder kopieren Sie diesen Code in die **Arduino IDE**.

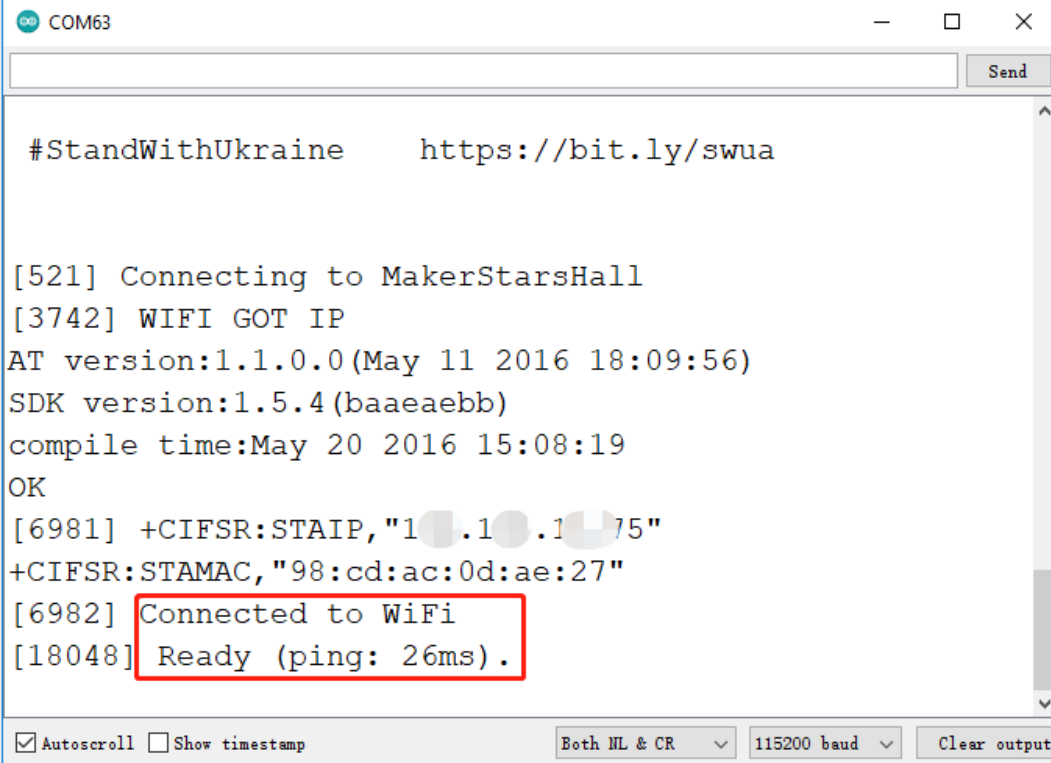
Bemerkung:

- Hier wird die `DHT sensor library` verwendet. Sie können diese aus dem **Library Manager** installieren.



2. Ersetzen Sie die `Template ID`, den `Device Name` und den `Auth Token` durch Ihre eigenen. Sie müssen auch die `ssid` und das `Passwort` des von Ihnen verwendeten WLANs eingeben. Für detaillierte Anleitungen siehe *1.4 Verbindung des R4-Boards mit Blynk herstellen*.
3. Wählen Sie das richtige Board und den richtigen Port aus und klicken Sie auf den **Upload**-Button.

4. Öffnen Sie den seriellen Monitor (Baudrate auf 115200 einstellen) und warten Sie, bis eine Meldung, wie z.B. eine erfolgreiche Verbindung, erscheint.



```
COM63

#StandWithUkraine    https://bit.ly/swua

[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"192.168.1.75"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

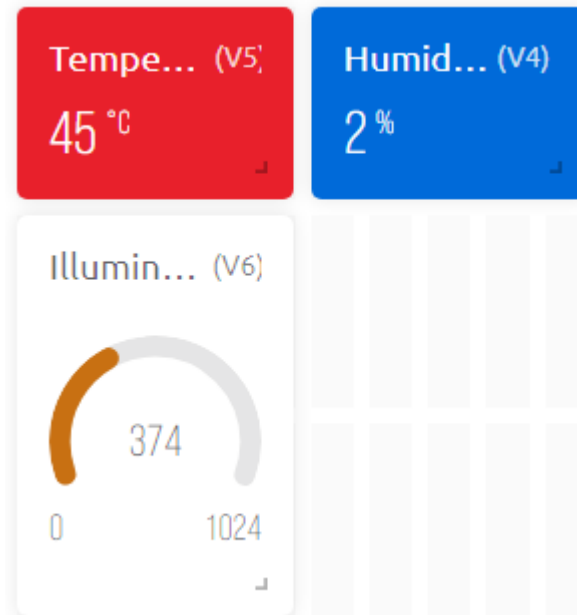
☒ Autoscroll ☐ Show timestamp Both NL & CR 115200 baud Clear output

Bemerkung: Wenn die Meldung ESP is not responding beim Verbinden erscheint, folgen Sie bitte diesen Schritten.

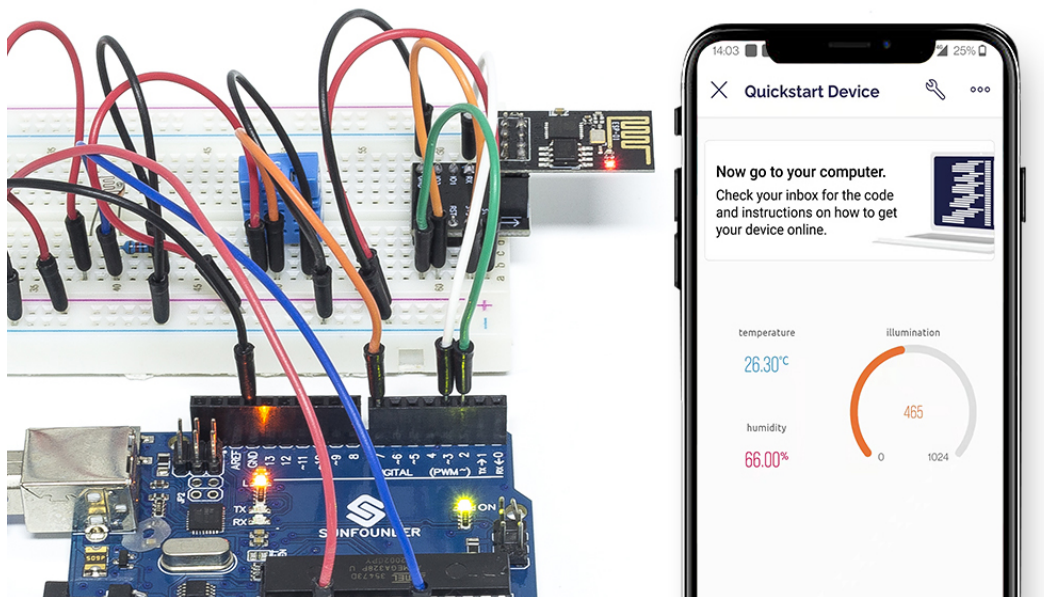
- Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.
- Setzen Sie das ESP8266-Modul zurück, indem Sie den RST-Pin für 1 Sekunde mit GND verbinden und dann trennen.
- Drücken Sie den Reset-Knopf auf dem R4-Board.

Manchmal müssen Sie den obigen Vorgang 3-5 Mal wiederholen, bitte haben Sie Geduld.

5. Nun sehen Sie die aktuelle Raumtemperatur, Luftfeuchtigkeit und Lichtintensität in Blynk.



6. Wenn Sie Blynk auf mobilen Geräten verwenden möchten, verweisen Sie auf *Wie verwendet man Blynk auf einem Mobilgerät?*.



Wie funktioniert das?

Diese beiden Funktionen werden verwendet, um die Temperatur, Luftfeuchtigkeit und Lichtintensität des Raumes zu erhalten.

```
int readLight(){
    return analogRead(lightPin);
}

bool readDHT() {
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
// Reading temperature or humidity takes about 250 milliseconds!  
// Sensor readings may also be up to 2 seconds 'old' (it's a very slow sensor)  
humidity = dht.readHumidity();  
// Read temperature as Celsius (the default)  
temperature = dht.readTemperature();  
  
// Check if any reads failed and exit early (to try again).  
if (isnan(humidity) || isnan(temperature)) {  
    Serial.println("Failed to read from DHT sensor!");  
    return false;  
}  
return true;  
}
```

Mit dem Blynk Timer werden die Raumtemperatur, Luftfeuchtigkeit und Lichtintensität jede Sekunde abgerufen und an den Datenstrom in der Blynk Cloud gesendet, von wo aus die Widgets die Daten anzeigen.

```
void myTimerEvent()  
{  
    bool chk = readDHT();  
    int light = readLight();  
    if(chk){  
        Blynk.virtualWrite(V4,humidity);  
        Blynk.virtualWrite(V5,temperature);  
    }  
    Blynk.virtualWrite(V6,light);  
}
```

6.6 6. Pflanzenüberwachung

Das Ziel dieses Projekts ist es, ein intelligentes Bewässerungssystem zu erstellen, das die aktuelle Temperatur, Luftfeuchtigkeit, Lichtintensität und Bodenfeuchtigkeit erfasst und diese auf Blynk anzeigt.

Sobald Sie den Schalter in Blynk Cloud aktivieren, beginnt die Pumpe zu arbeiten und die Pflanzen werden bewässert.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

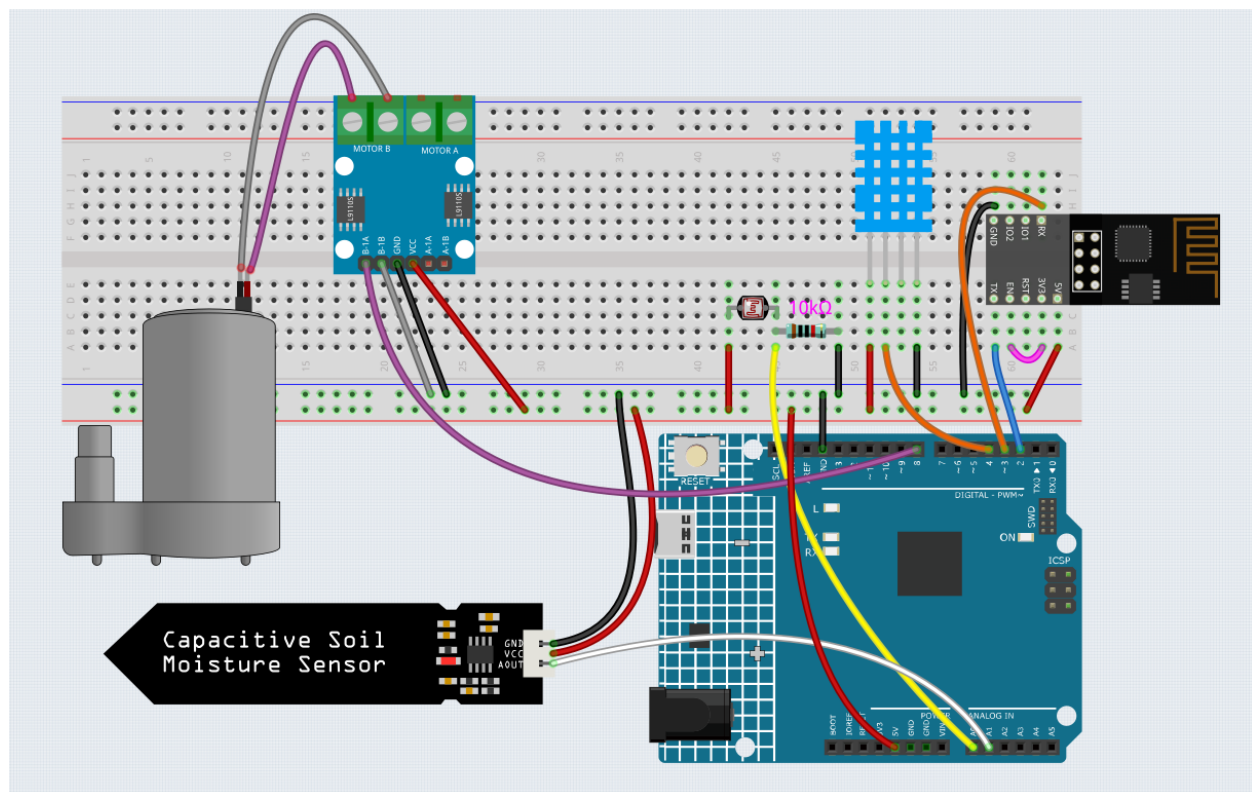
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
Arduino Uno R4 Minima	-
Breadboard	
ESP8266 Modul	
Jumper-Kabel	
Widerstand	
Fotowiderstand	
DHT11 Feuchtigkeits- und Temperatursensor	-
Bodenfeuchtigkeitsmodul	
L9110 Motortreibermodul	-
Kreiselpumpe	-

1. Bauen Sie den Schaltkreis

Bemerkung: Das ESP8266-Modul benötigt einen hohen Strom, um eine stabile Betriebsumgebung zu gewährleisten. Stellen Sie daher sicher, dass die 9V-Batterie angeschlossen ist.



2. Dashboard bearbeiten

- Die in den vorherigen Projekten erstellten Datenströme müssen gespeichert werden, und sie werden auch in diesem Projekt verwendet.

- Um die Bodenfeuchtigkeit aufzuzeichnen, erstellen Sie einen weiteren **Datastream** vom Typ **Virtual Pin** auf der **Datastream**-Seite. Setzen Sie den DATENTYP auf **Integer** und MIN und MAX auf **0** und **1024**.

Virtual Pin Datastream

NAME: ALIAS:

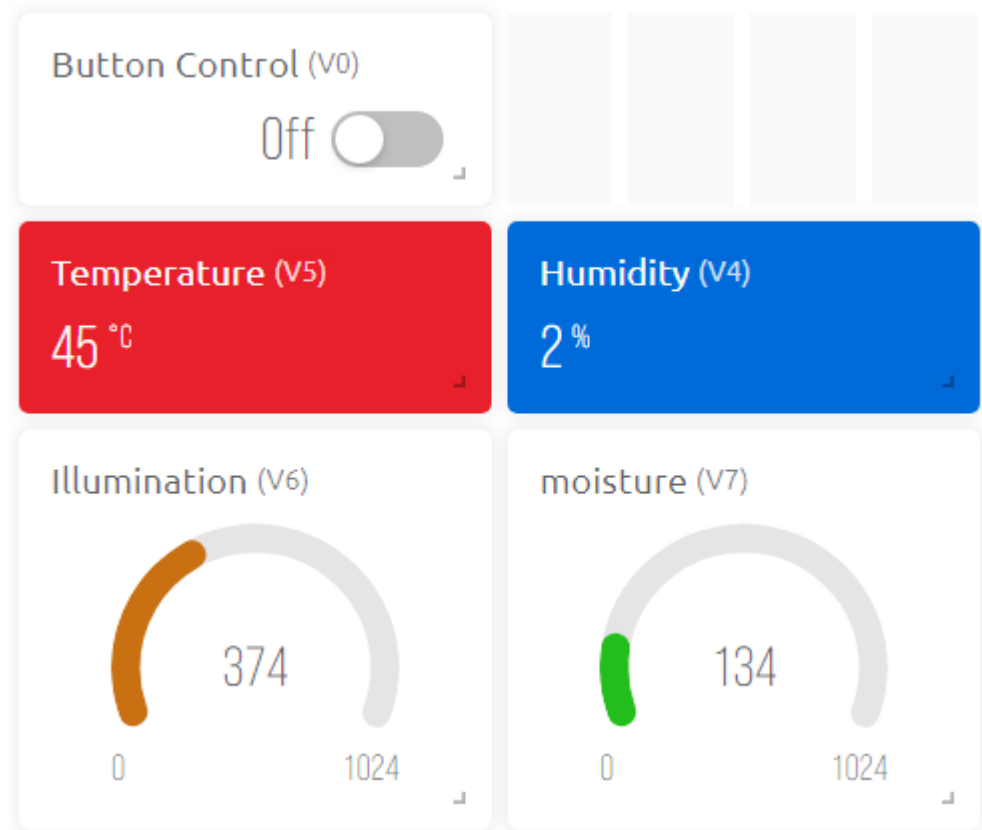
PIN: DATA TYPE:

UNITS:

MIN: MAX: DEFAULT VALUE:

Region: ny3 [Privacy Policy](#)

- Gehen Sie nun zur **Wed Dashboard**-Seite, ziehen Sie 2 **Label**-Widgets und setzen Sie ihre Datenströme jeweils auf **V4** und **V5**; ziehen Sie 2 **Gauge**-Widgets und setzen Sie ihre Datenströme so, dass sie **V6** und **V7** anzeigen; und ziehen Sie schließlich ein **Switch**-Widget und setzen Sie seinen Datenstrom auf **V0**.

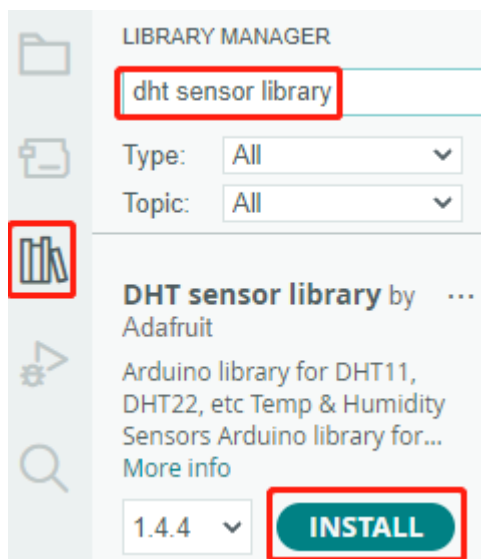


3. Den Code ausführen

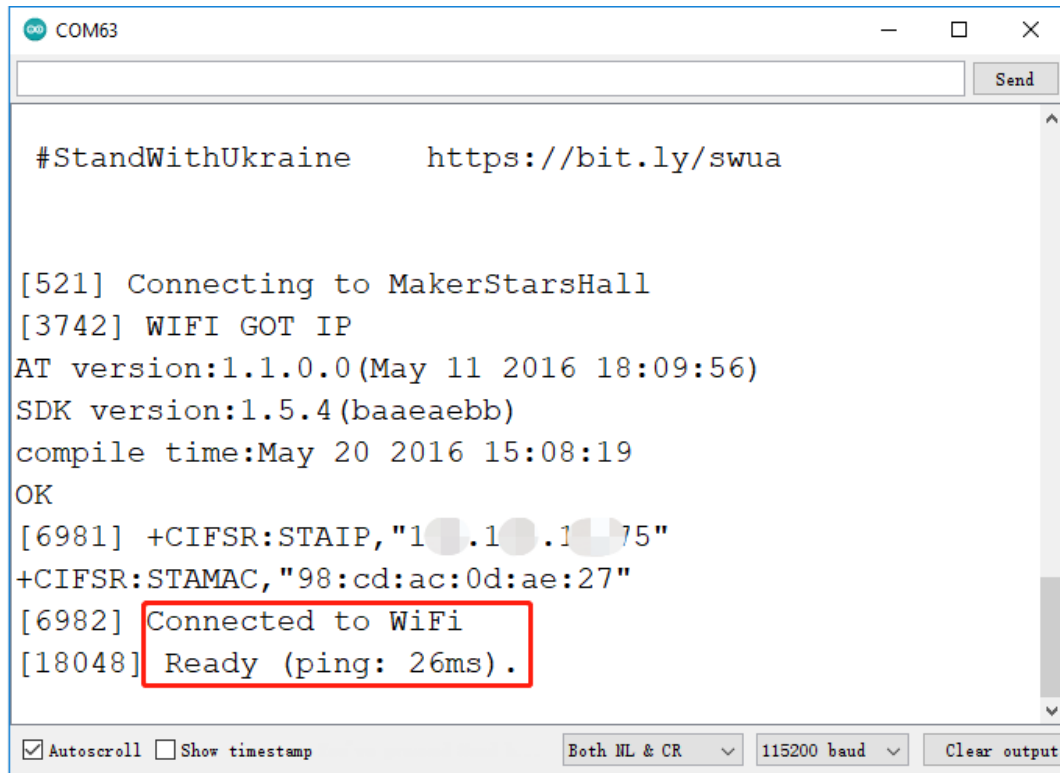
Öffnen Sie die Datei `6.plant_monitoring.ino` im Pfad `3in1-kit\iot_project\6.plant_monitoring`, oder kopieren Sie diesen Code in die **Arduino IDE**.

Bemerkung:

- Hier wird die `DHT sensor library` verwendet. Sie können diese über den **Library Manager** installieren.



1. Ersetzen Sie die Template ID, den Device Name und das Auth Token durch Ihre eigenen Angaben. Sie müssen auch die ssid und das password des von Ihnen verwendeten WLANs eingeben. Für detaillierte Anleitungen verweisen Sie bitte auf [1.4 Verbindung des R4-Boards mit Blynk herstellen](#).
2. Wählen Sie das richtige Board und den Port aus und klicken Sie auf die Schaltfläche **Upload**.
3. Öffnen Sie den Seriellen Monitor (Baudrate auf 115200 einstellen) und warten Sie auf eine Meldung, z. B. dass eine erfolgreiche Verbindung hergestellt wurde.



```
#StandWithUkraine https://bit.ly/swua

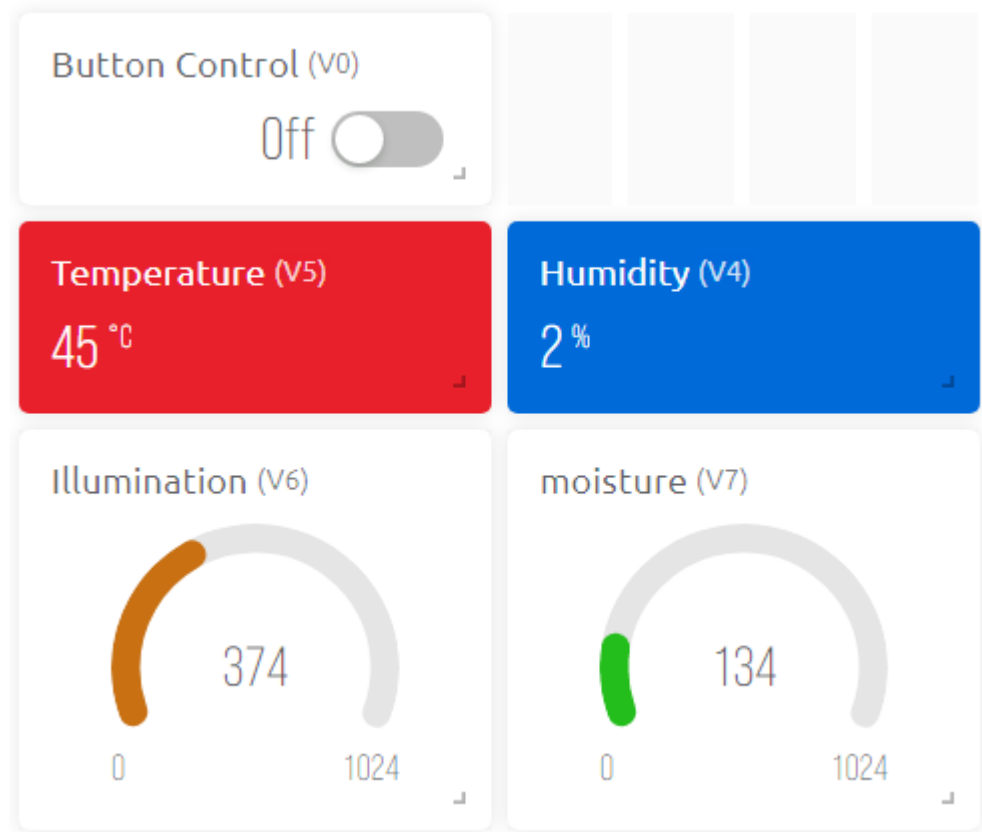
[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"1.1.1.175"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

Bemerkung: Wenn beim Verbinden die Meldung `ESP is not responding` erscheint, befolgen Sie bitte diese Schritte:

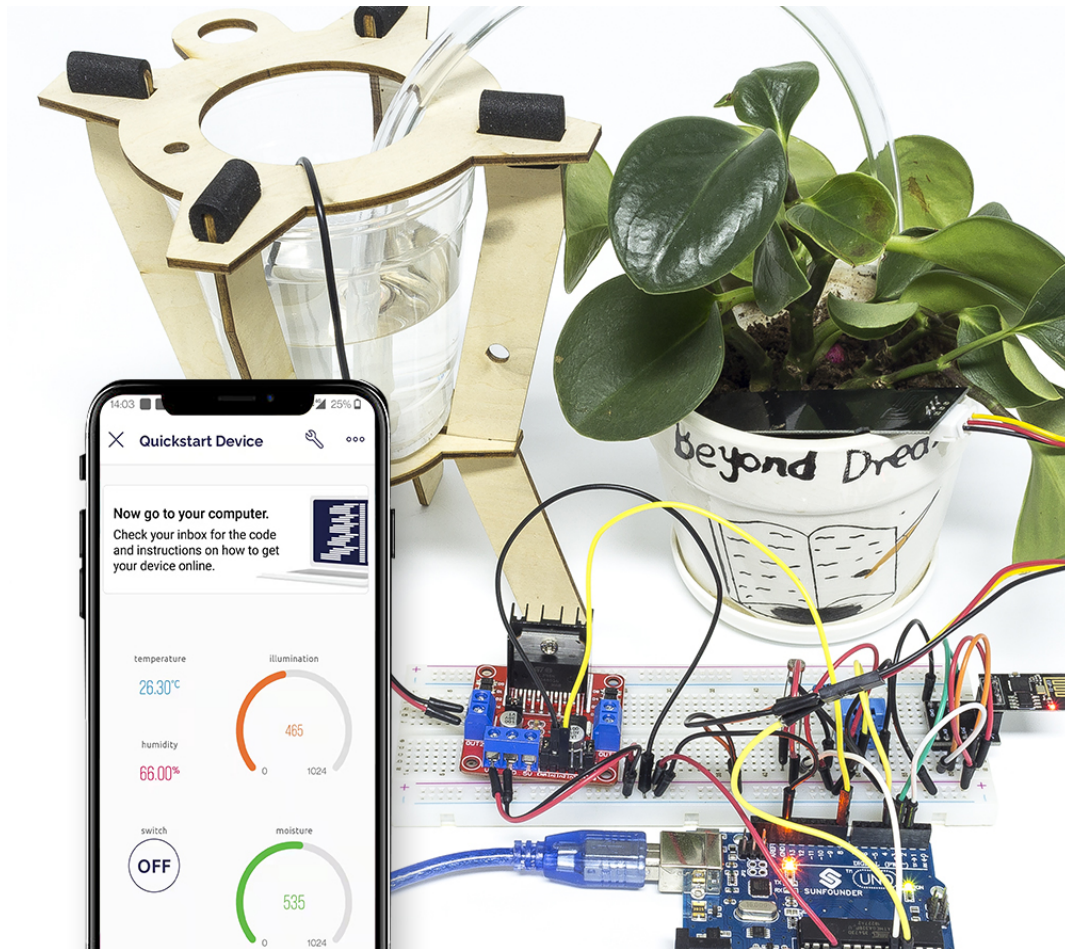
- Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.
- Setzen Sie das ESP8266-Modul zurück, indem Sie den RST-Pin für 1 Sekunde mit GND verbinden und dann trennen.
- Drücken Sie den Reset-Knopf auf dem R4-Board.

Manchmal müssen Sie den obigen Vorgang 3-5 Mal wiederholen. Bitte haben Sie Geduld.

4. Zurück in Blynk werden die aktuelle Temperatur, Luftfeuchtigkeit, Lichtintensität und Bodenfeuchtigkeit angezeigt. Mit dem Button Control Widget können Sie, falls nötig, die Pumpe aktivieren, um die Pflanzen zu bewässern.



5. Wenn Sie Blynk auf mobilen Geräten nutzen möchten, verweisen Sie bitte auf *Wie verwendet man Blynk auf einem Mobilgerät?*.



Wie funktioniert das?

Dieser BLYNK_WRITE bewirkt, dass das **Switch**-Widget von Blynk die Pumpe startet, wenn es eingeschaltet ist und ausschaltet, wenn es ausgeschaltet ist.

```
BLYNK_WRITE(V0)
{
  if(param.asInt()==1){
    digitalWrite(pumpA,HIGH);
  }else{
    digitalWrite(pumpA,LOW);
  }
}
```

Diese drei Funktionen werden verwendet, um die aktuelle Umgebungstemperatur, Luftfeuchtigkeit, Lichtintensität und Bodenfeuchtigkeit zu messen.

```
int readMoisture(){
  return analogRead(moisturePin);
}

int readLight(){
  return analogRead(lightPin);
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

bool readDHT() {
    // Reading temperature or humidity takes about 250 milliseconds!
    // Sensor readings may also be up to 2 seconds 'old' (it's a very slow sensor)
    humidity = dht.readHumidity();
    // Read temperature as Celsius (the default)
    temperature = dht.readTemperature();

    // Check if any reads failed and exit early (to try again).
    if (isnan(humidity) || isnan(temperature)) {
        Serial.println("Failed to read from DHT sensor!");
        return false;
    }
    return true;
}

```

Mit dem Blynk Timer werden jede Sekunde die Umgebungstemperatur, Luftfeuchtigkeit, Lichtintensität und Bodenfeuchtigkeit ermittelt und an den Datenstrom auf dem **Blynk Cloud** gesendet, von wo aus die Widgets die Daten anzeigen.

```

void myTimerEvent()
{
    bool chk = readDHT();
    int light = readLight();
    int moisture = readMoisture();
    if(chk){
        Blynk.virtualWrite(V4,humidity);
        Blynk.virtualWrite(V5,temperature);
    }
    Blynk.virtualWrite(V6,light);
    Blynk.virtualWrite(V7,moisture);
}

```

6.7 7. Strombegrenztes Tor

In manchen Situationen, wie zum Beispiel auf Parkplätzen, ist eine Mengenverwaltung erforderlich.

Hier erstellen wir ein intelligentes Tor: Ein Servo dient als Tor und vor ihm wird ein IR-Hinderniserkennungssensor platziert; wenn ein Objekt (z.B. ein Auto) erkannt wird, öffnet sich das Tor und die Anzahl erhöht sich um 1. Die Zählung wird auf einer 7-Segment-Anzeige dargestellt und auch in die Blynk Cloud hochgeladen, sodass Sie sie aus der Ferne einsehen können. Schließlich verfügt Blynk über ein Schalter-Widget, um dieses intelligente Torsystem zu aktivieren oder zu deaktivieren.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>Arduino Uno R4 Minima</i>	-
<i>Breadboard</i>	
<i>ESP8266 Modul</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Servo</i>	
<i>Hindernisvermeidungsmodul</i>	
<i>7-Segment-Anzeige</i>	
<i>74HC595</i>	

1. Den Schaltkreis aufbauen

Bemerkung: Das ESP8266-Modul benötigt einen hohen Strom, um eine stabile Betriebsumgebung zu gewährleisten. Stellen Sie daher sicher, dass die 9V-Batterie angeschlossen ist.

Virtual Pin Datastream

NAME: ALIAS:

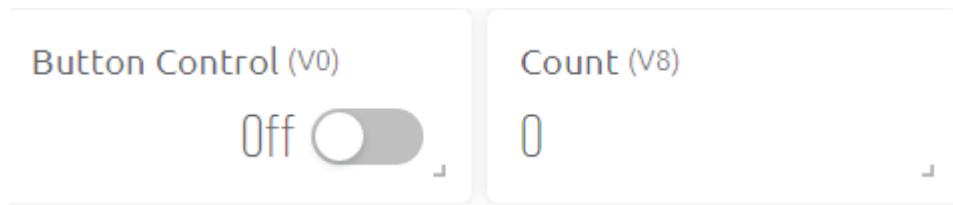
PIN: DATA TYPE:

UNITS:

MIN: MAX: DEFAULT VALUE:

Region: ny3 [Privacy Policy](#)

2. Navigieren Sie nun zur **Wed Dashboard**-Seite, ziehen Sie ein **Switch**-Widget, um seinen Datenstrom auf **V0** und ein **Label**-Widget, um seinen Datenstrom auf **V8** zu setzen.



3. Den Code ausführen

1. Öffnen Sie die Datei `7.current_limiting_gate.ino` unter dem Pfad `3in1-kit\iot_project\7.current_limiting_gate`, oder kopieren Sie diesen Code in die **Arduino IDE**.
2. Ersetzen Sie die `Template ID`, `Device Name` und `Auth Token` durch Ihre eigenen. Sie müssen auch die `ssid` und das `password` des von Ihnen verwendeten WLANs eingeben. Für detaillierte Anleitungen verweisen Sie bitte auf [1.4 Verbindung des R4-Boards mit Blynk herstellen](#).
3. Wählen Sie das korrekte Board und den Port aus und klicken Sie auf die Schaltfläche **Upload**.
4. Öffnen Sie den Seriellen Monitor (Baudrate auf 115200 einstellen) und warten Sie, bis eine Aufforderung wie eine erfolgreiche Verbindung erscheint.


```

COM63

#StandWithUkraine https://bit.ly/swua

[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"192.168.1.175"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).

☒ Autoscroll ☐ Show timestamp
Both NL & CR 115200 baud Clear output

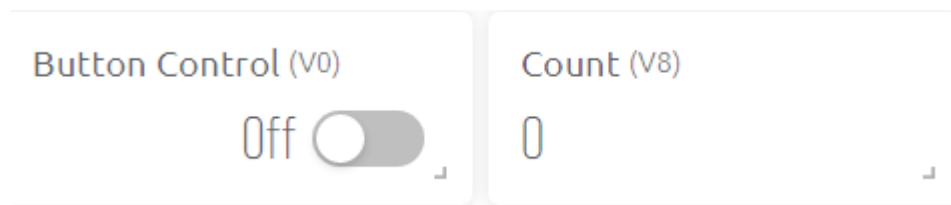
```

Bemerkung: Wenn die Meldung ESP is not responding erscheint, wenn Sie sich verbinden, folgen Sie bitte diesen Schritten.

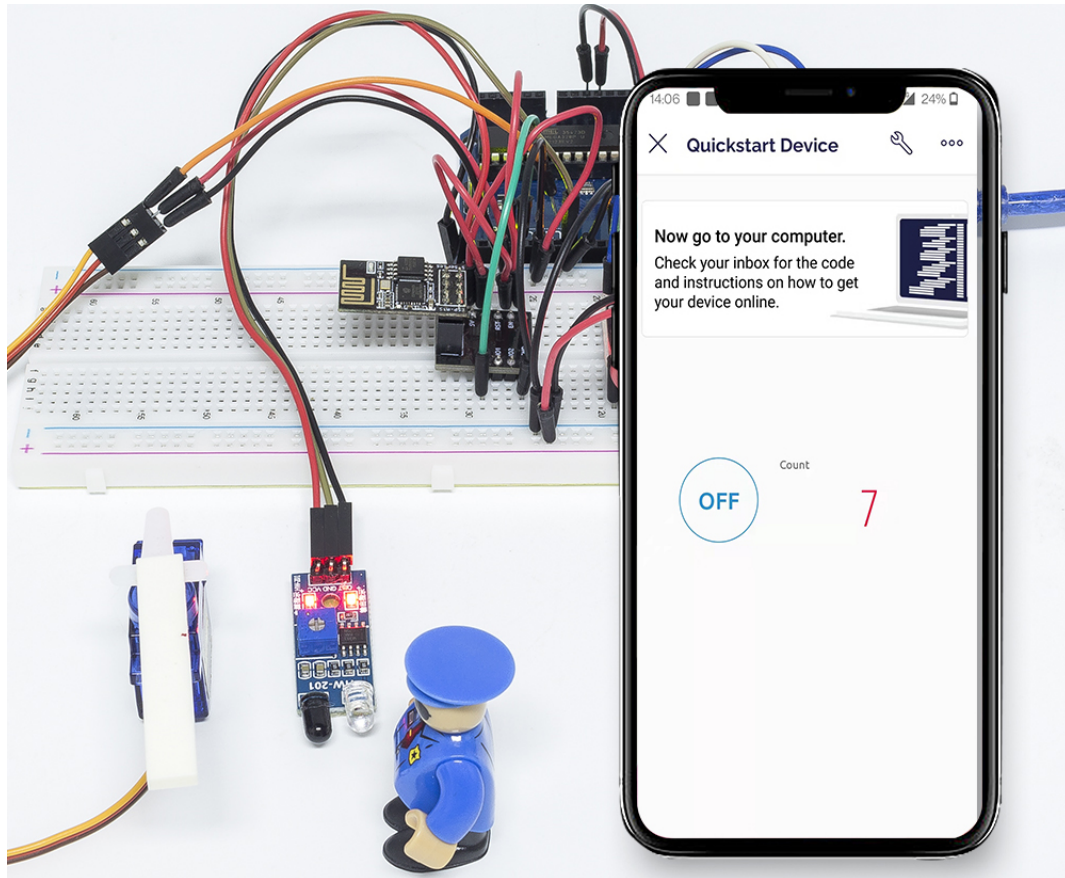
- Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.
- Setzen Sie das ESP8266-Modul zurück, indem Sie den Pin RST für 1 Sekunde mit GND verbinden, dann ziehen Sie ihn ab.
- Drücken Sie den Reset-Knopf auf dem R4-Board.

Manchmal müssen Sie den obigen Vorgang 3-5 Mal wiederholen, bitte haben Sie Geduld.

5. Klicken Sie nun auf das Button Control-Widget in Blynk, um das intelligente Torsystem zu aktivieren. Wenn das IR-Hindernisvermeidungsmodul ein Hindernis erkennt, öffnet sich das Tor und die 7-Segment-Anzeige sowie das Zählwidget in Blynk erhöhen sich um 1.



6. Wenn Sie Blynk auf mobilen Geräten verwenden möchten, verweisen Sie bitte auf *Wie verwendet man Blynk auf einem Mobilgerät?*.



Wie funktioniert das?

Die Funktion `BLYNK_WRITE(V0)` erhält den Status des **Switch**-Widgets und weist ihn der Variable `doorFlag` zu, die verwendet wird, um zu bestimmen, ob das intelligente Torsystem aktiviert ist oder nicht.

```
BLYNK_WRITE(V0)
{
    doorFlag = param.asInt(); // Enable Gat
}
```

Im Blynk Timer wird `doorFlag` jede Sekunde überprüft und, falls aktiviert, die Hauptfunktion des Tors ausgeführt.

```
void myTimerEvent()
{
    if (doorFlag)
    {
        channelEntrance();
    }
}
```

Die Hauptfunktion des Tors ist `channelEntrance()`. Wenn ein Objekt sich dem Tor nähert (der Sensor erkennt ein Hindernis), wird `count` um 1 erhöht. Schreiben Sie `count` in den Datenstrom `V8` der Blynk Cloud und die 7-Segment-Anzeige im Schaltkreis und öffnen Sie das Tor. Wenn das Objekt von vorhanden zu abwesend wechselt, was bedeutet, dass das Objekt das Tor betreten hat, schließen Sie das Tor.

```
void channelEntrance()
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

{
  int currentState = digitalRead(irPin); // 0:obstacle 1:no-obstacle
  if (currentState == 0 && lastState == 1) {
    count=(count+1)%10;
    Blynk.virtualWrite(V8, count);
    showNumber(count);
    operateGate(true);
  } else if ((currentState == 1 && lastState == 0)) {
    operateGate(false);
  }
  lastState = currentState;
}

```

Die Funktion `showNumber(int num)` wird verwendet, um die 7-Segment-Anzeige den Wert anzeigen zu lassen.

```

void showNumber(int num)
{
  digitalWrite(STcp, LOW); //ground ST_CP and hold low for as long as you are
  ↳transmitting
  shiftOut(DS, SHcp, MSBFIRST, dataArray[num]);
  digitalWrite(STcp, HIGH); //pull the ST_CPST_CP to save the data
}

```

Die Funktion `operateGate(bool openGate)` öffnet das Tor langsam, wenn die Referenz True ist, und schließt das Tor langsam, wenn die Referenz False ist.

```

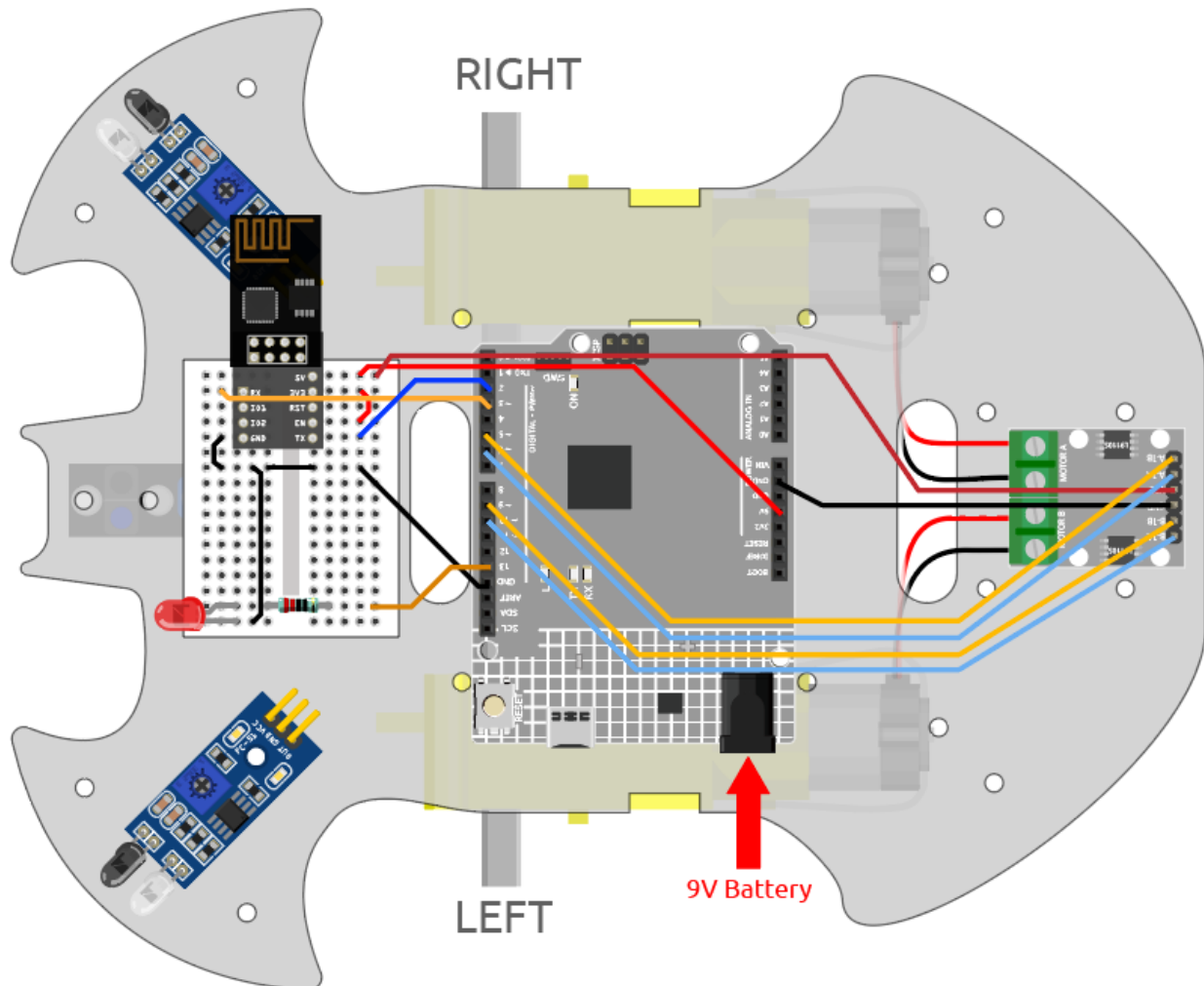
void operateGate(bool openGate) {
  if (openGate == true)
  {
    // open gate
    while (angle <= 90) {
      angle++;
      myservo.write(angle);
      delay(5);
    }
  } else {
    // close gate
    while (angle >= 0){
      angle--;
      myservo.write(angle);
      delay(5);
    }
  }
}
}

```

6.8 8. IoT-Auto

Für dieses Projekt haben wir die Blynk-App auf dem Smartphone genutzt, um das Auto zu steuern. Allerdings müssen Sie zuerst den Abschnitt *Auto-Projekte* konsultieren, um das Auto zusammenzubauen und ein grundlegendes Verständnis dafür zu bekommen. In Zeiten der 5G-Netzwerkverbreitung könnte diese Methode in vielen Industrien zu einer der Hauptproduktionsweisen werden. Daher lohnt es sich, diese Technologie vorab zu erleben.

1. Schaltkreis aufbauen



2. Dashboard bearbeiten

Blynk für Mobilgeräte bietet keine Möglichkeit zur Bearbeitung von Datastreams, daher müssen diese Schritte weiterhin über die Web-Oberfläche erledigt werden.

1. Erstellen Sie einen **Datastream** vom Typ **Virtual Pin** auf der **Datastream**-Seite, um den X-Achsenwert des Joysticks zu speichern. Legen Sie den NAMEN auf Xvalue, den DATENTYP auf Integer und MIN sowie MAX auf -10 und 10 fest.

Virtual Pin Datastream

NAME: Xvalue ALIAS: Xvalue

PIN: V9 DATA TYPE: Integer

UNITS: None

MIN: -10 MAX: 10 DEFAULT VALUE: 0

+ ADVANCED SETTINGS Cancel Create

2. Erstellen Sie einen weiteren **Datastream** vom Typ **Virtual Pin**, um den Y-Achsenwert des Joysticks zu erfassen. Setzen Sie den NAMEN auf Yvalue, den DATENTYP auf Integer, sowie MIN und MAX auf -10 und 10.

Virtual Pin Datastream

NAME: Yvalue ALIAS: Yvalue

PIN: V10 DATA TYPE: Integer

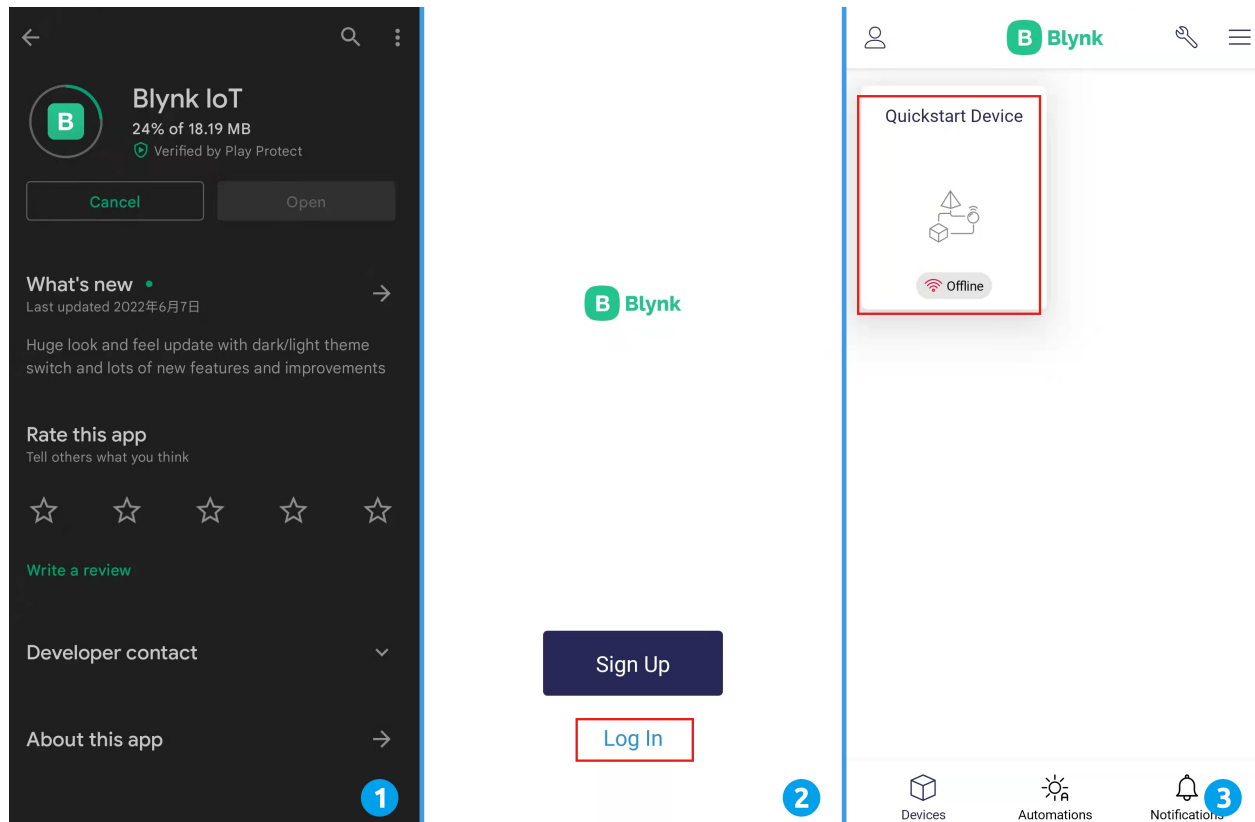
UNITS: None

MIN: -10 MAX: 10 DEFAULT VALUE: 0

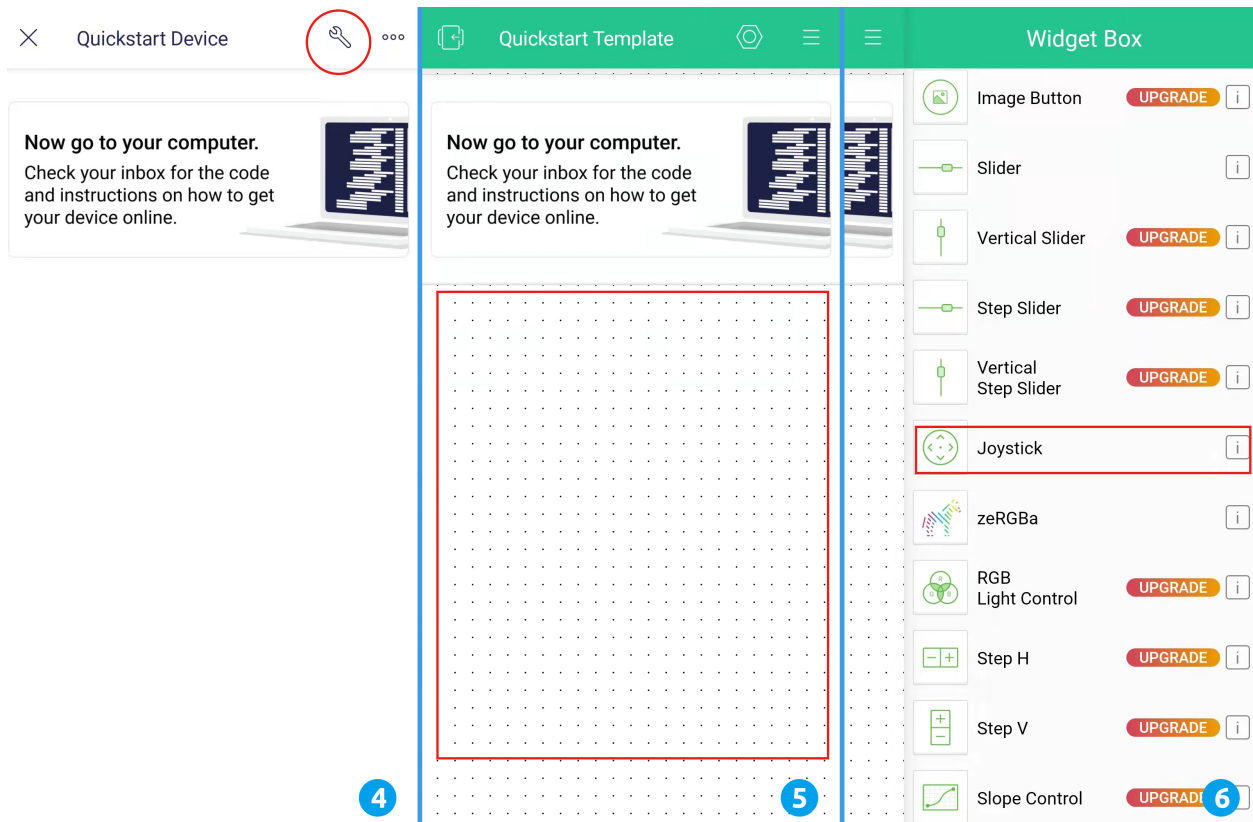
ADVANCED SETTINGS Cancel Create

Führen Sie anschließend die folgenden Schritte auf Ihrem Smartphone durch:

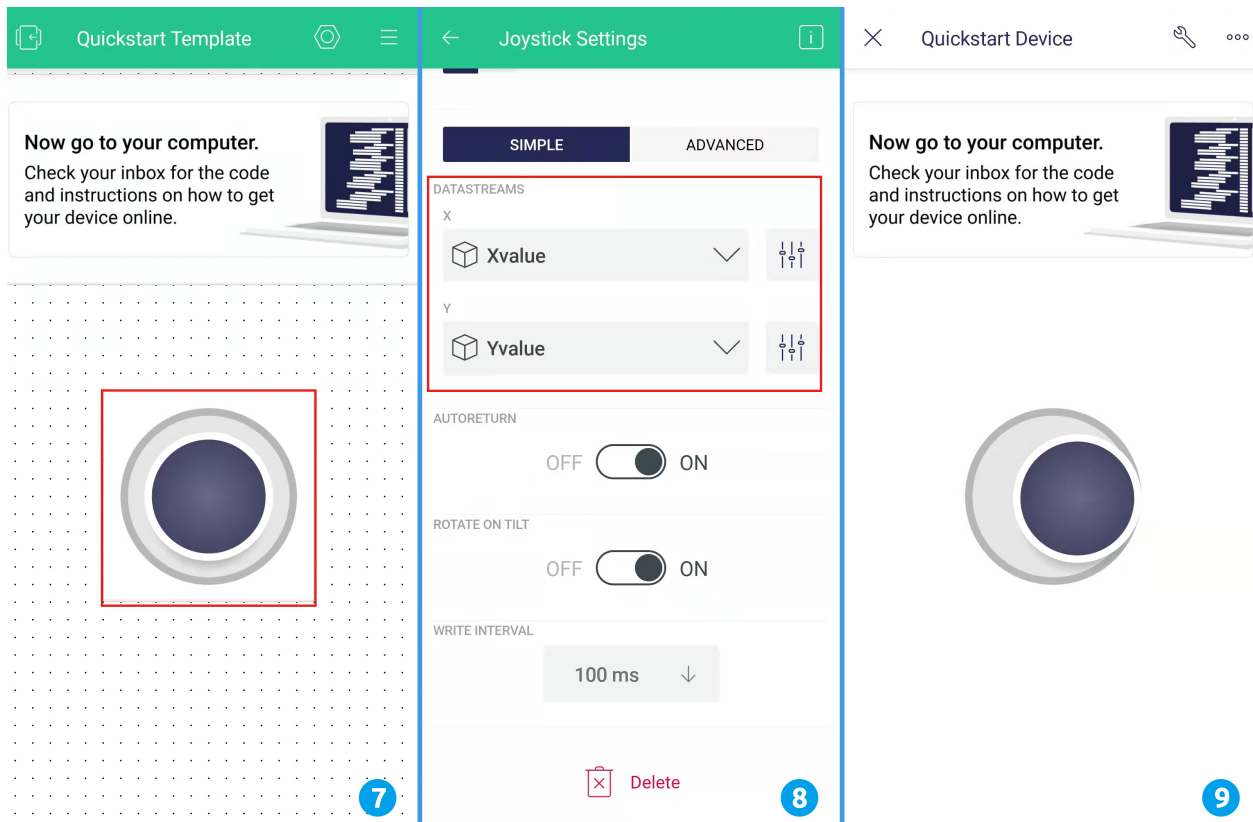
1. Suchen Sie im GOOGLE Play oder im APP Store nach „Blynk IoT“ (nicht Blynk(legacy)) und laden Sie die App herunter.
2. Nach dem Öffnen der App melden Sie sich an. Dieses Konto sollte mit dem Konto übereinstimmen, das Sie auf der Web-Oberfläche verwenden.
3. Navigieren Sie dann zum Dashboard (falls noch keines vorhanden ist, erstellen Sie eines). Sie werden feststellen, dass die Dashboards für Mobilgeräte und Web unabhängig voneinander sind.



4. Klicken Sie auf das Bearbeitungssymbol.
5. Tippen Sie auf einen freien Bereich.
6. Wählen Sie ein Joystick-Widget aus.

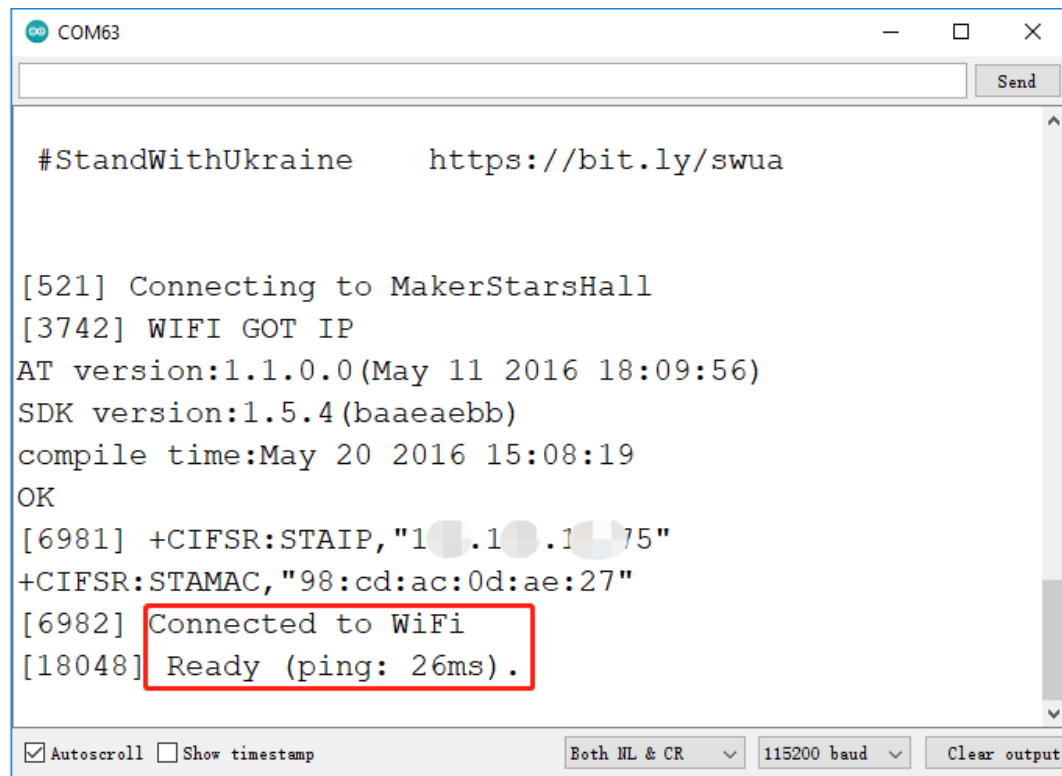


7. Nun sollten Sie ein Joystick-Widget im freien Bereich sehen, tippen Sie darauf.
8. Die Joystick-Einstellungen werden angezeigt. Wählen Sie die zuvor in den Datastreams festgelegten Xvalue und Yvalue aus.
9. Kehren Sie zur Dashboard-Seite zurück. Nun können Sie den Joystick nach Belieben bedienen.



3. Code ausführen

1. Öffnen Sie die Datei `8.iot_car.ino` unter dem Pfad `3in1-kit\iot_project\8.iot_car` oder kopieren Sie diesen Code in die **Arduino IDE**.
2. Ersetzen Sie die **Template ID**, den **Device Name** und den **Auth Token** durch Ihre eigenen Angaben. Geben Sie zudem die **ssid** und das **password** des von Ihnen genutzten WLANs ein. Für detaillierte Anleitungen siehe [1.4 Verbindung des R4-Boards mit Blynk herstellen](#).
3. Wählen Sie das richtige Board und den passenden Port aus und klicken Sie dann auf den **Upload**-Button.
4. Öffnen Sie den Seriellen Monitor (Baudrate auf 115200 einstellen) und warten Sie, bis eine Meldung über eine erfolgreiche Verbindung erscheint.



```
COM63

#StandWithUkraine  https://bit.ly/swua

[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"1.1.1.1"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).

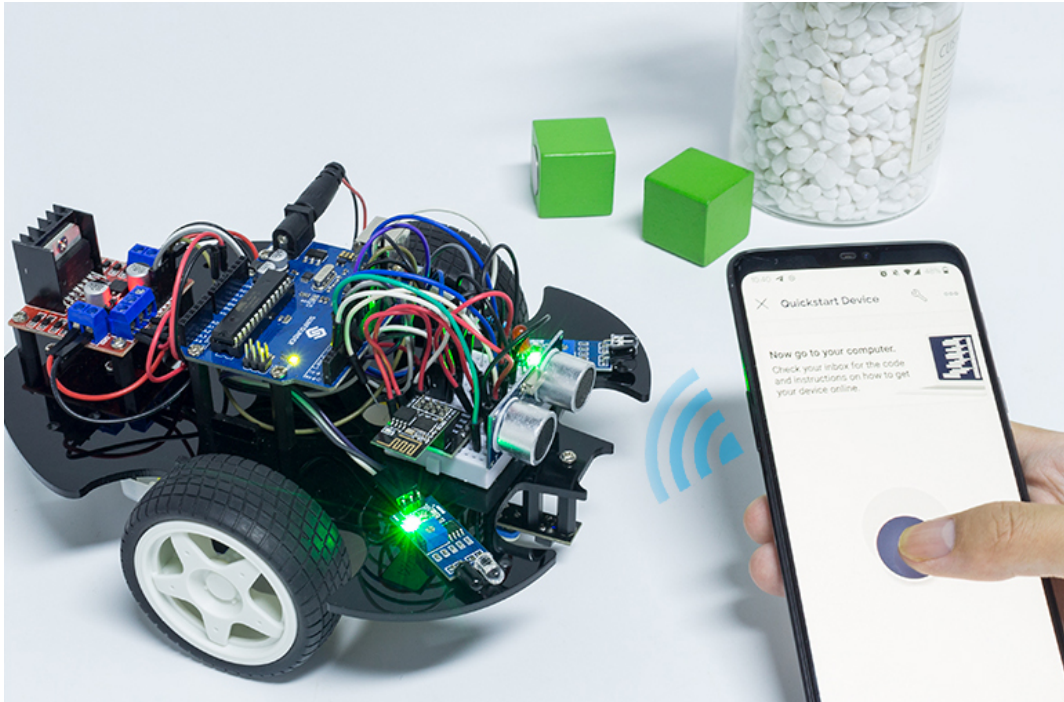
☒ Autoscroll ☐ Show timestamp
Both NL & CR 115200 baud Clear output
```

Bemerkung: Wenn die Meldung ESP is not responding beim Verbinden erscheint, folgen Sie bitte diesen Schritten.

- Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.
- Setzen Sie das ESP8266-Modul zurück, indem Sie den Pin RST für 1 Sekunde mit GND verbinden, danach trennen.
- Drücken Sie die Reset-Taste auf dem R4-Board.

Manchmal müssen Sie die obigen Schritte 3-5 Mal wiederholen, bitte haben Sie Geduld.

5. Trennen Sie jetzt das USB-Kabel und versorgen Sie den Wagen nur mit einer 9V-Batterie. Warten Sie, bis die LED leuchtet, was darauf hinweist, dass das Auto mit Blynk verbunden ist.
6. Öffnen Sie Blynk auf Ihrem Handy und steuern Sie das Auto mithilfe des Joystick-Widgets.



Wie funktioniert das?

Diese Funktionen werden zur Steuerung der Bewegungen des Autos verwendet.

```
void moveForward(int speed) {...}
void moveBackward(int speed) {...}
void turnRight(int speed) {...}
void turnLeft(int speed) {...}
void stopMove() {...}
```

Der IoT-Bereich liest die Werte des Joystick-Widgets aus und weist sie den Variablen Xvalue und Yvalue zu.

```
int Xvalue = 0;
int Yvalue = 0;

BLYNK_WRITE(V9)
{
    Xvalue = param.asInt();
}

BLYNK_WRITE(V10)
{
    Yvalue = param.asInt();
}
```

Im loop() führt das Auto unterschiedliche Aktionen basierend auf Xvalue und Yvalue durch.

```
if (Yvalue >= 5) {
    moveForward(255);
} else if (Yvalue <= -5) {
    moveBackward(255);
} else if (Xvalue >= 5) {
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
    turnRight(150);
} else if (Xvalue <= -5) {
    turnLeft(150);
} else {
    stopMove();
}
```

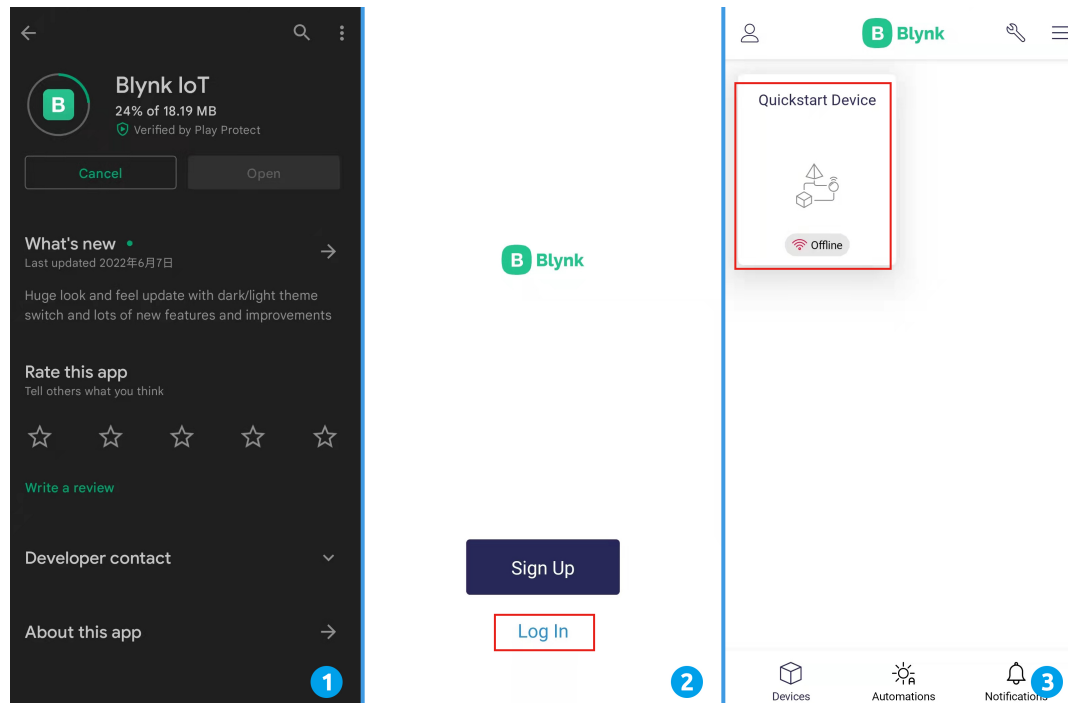
Fügen Sie außerdem eine Netzwerkstatusbestimmung zum `loop()` hinzu, die eine LED aufleuchten lässt, wenn eine Verbindung zur Blynk Cloud besteht.

```
if (!Blynk.connected()) {
    digitalWrite(ledPin, LOW);
    Serial.print("offline!");
    bool result = Blynk.connect();
    Serial.println(result);
} else {
    digitalWrite(ledPin, HIGH);
}
```

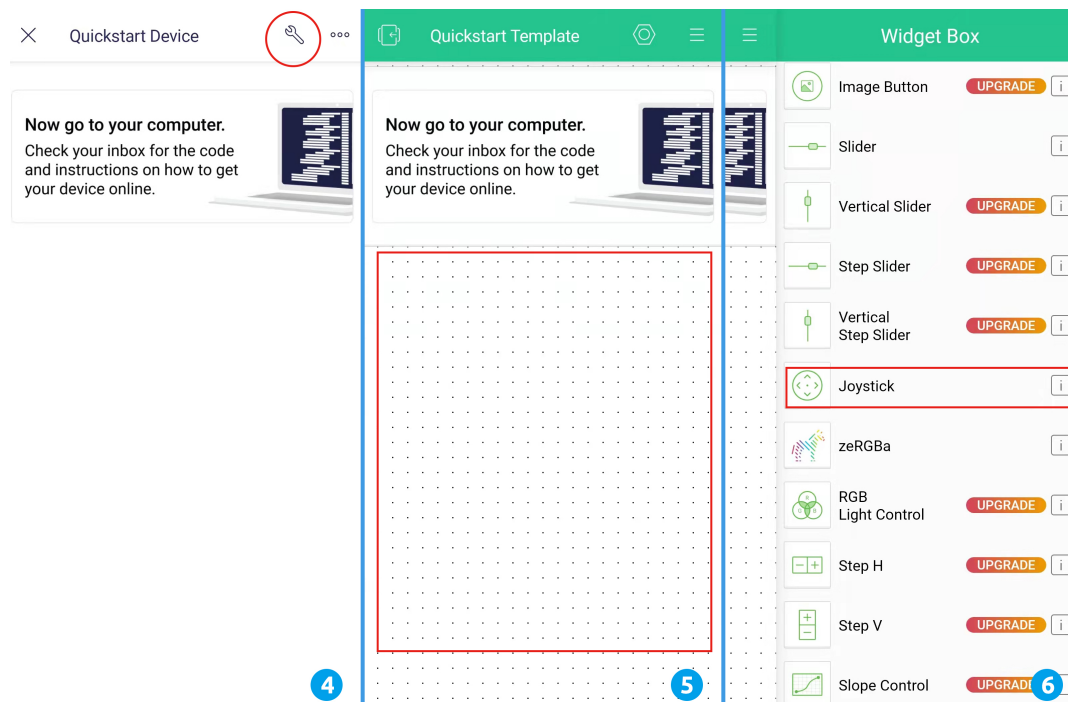
7.1 Wie verwendet man Blynk auf einem Mobilgerät?

Bemerkung: Da Datenströme nur im Web-Blynk erstellt werden können, müssen Sie verschiedene Projekte heranziehen, um Datenströme im Web zu erstellen. Anschließend folgen Sie dem untenstehenden Tutorial, um Widgets in Blynk auf Ihrem Mobilgerät zu erstellen.

1. Öffnen Sie Google Play oder den APP Store auf Ihrem Mobilgerät und suchen Sie nach „Blynk IoT“ (nicht Blynk(legacy)) zum Herunterladen.
2. Nach dem Öffnen der App melden Sie sich an. Dieses Konto sollte dasselbe sein wie das Konto, das auf dem Web-Client verwendet wird.
3. Gehen Sie dann zum **Dashboard** (falls Sie keines haben, erstellen Sie eins) und Sie werden sehen, dass das **Dashboard** für Mobilgeräte und Web voneinander unabhängig sind.

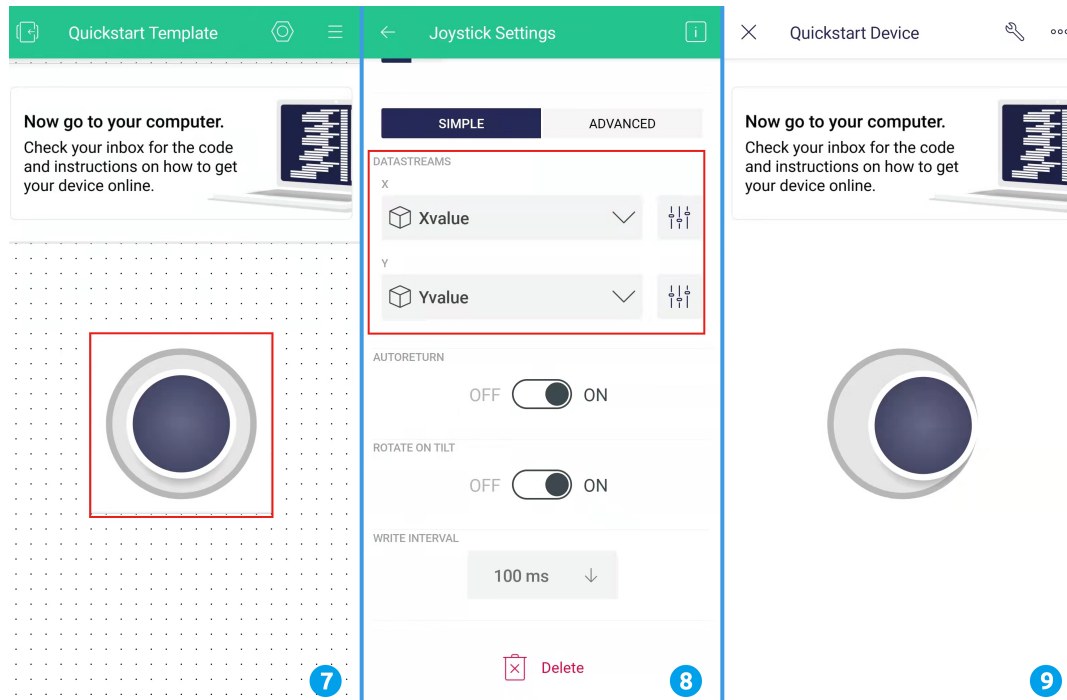


4. Klicken Sie auf das **Bearbeiten**-Symbol.
5. Klicken Sie auf den leeren Bereich.
6. Wählen Sie das gleiche Widget wie auf der Webseite, zum Beispiel ein **Joystick**-Widget.



7. Jetzt sehen Sie ein **Joystick**-Widget im leeren Bereich, klicken Sie darauf.
8. Die **Joystick**-Einstellungen werden angezeigt. Wählen Sie die **Xvalue**- und **Yvalue**-Datenströme aus, die Sie gerade auf der Webseite eingerichtet haben. Beachten Sie, dass jedes Widget in jedem Projekt einem anderen Datenstrom entspricht.

9. Gehen Sie zurück zur **Dashboard**-Seite und Sie können den **Joystick** bei Bedarf bedienen.

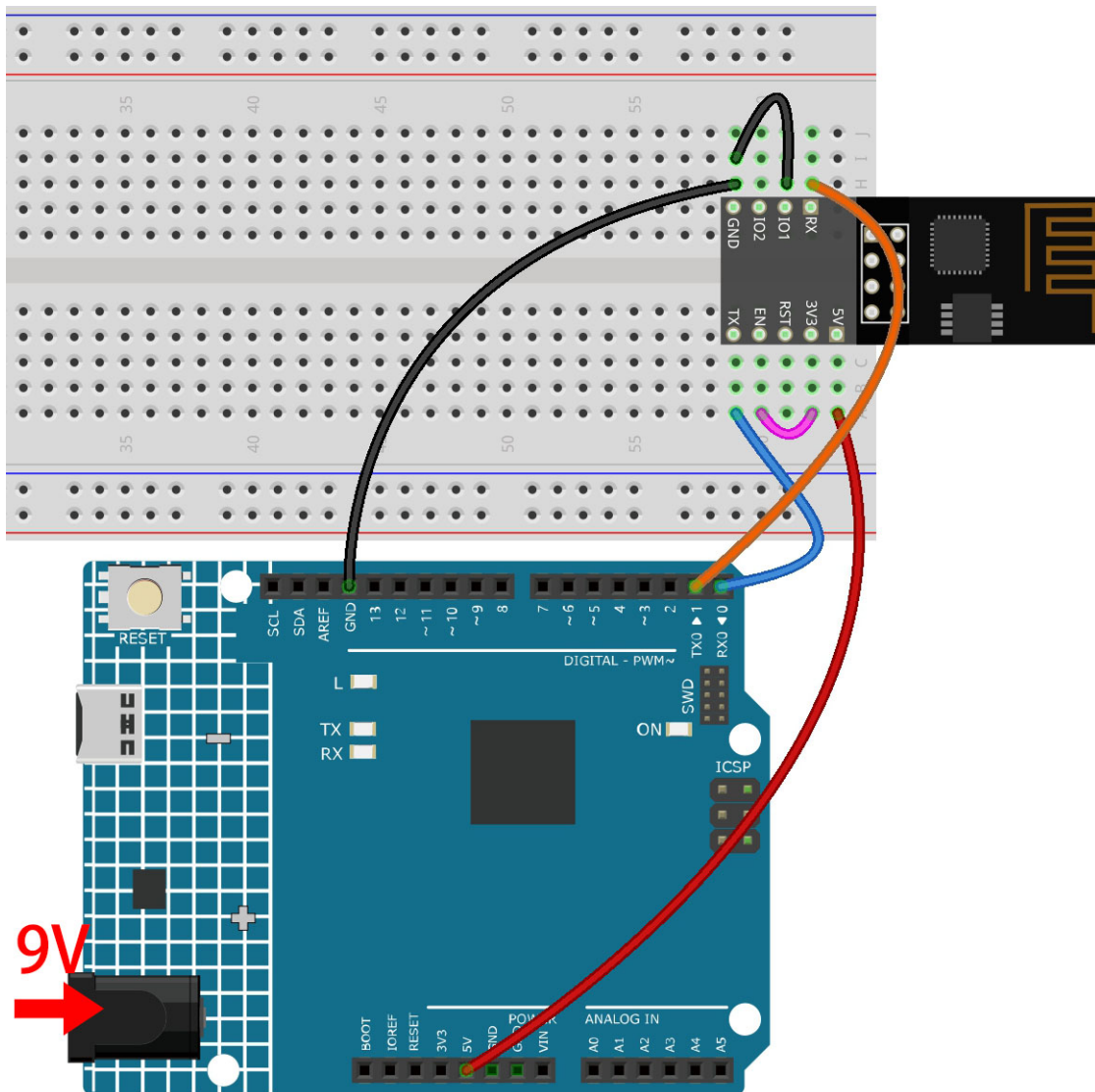


7.2 Wie lädt man die Firmware für das ESP8266-Modul erneut hoch?

7.2.1 Firmware mit R4 erneut hochladen

1. Schaltkreis aufbauen

Verbinden Sie ESP8266 und das Arduino UNO R4 Board.



2. Folgenden Code auf R4 hochladen

```
void setup() {
  Serial.begin(115200);
  Serial1.begin(115200);
}

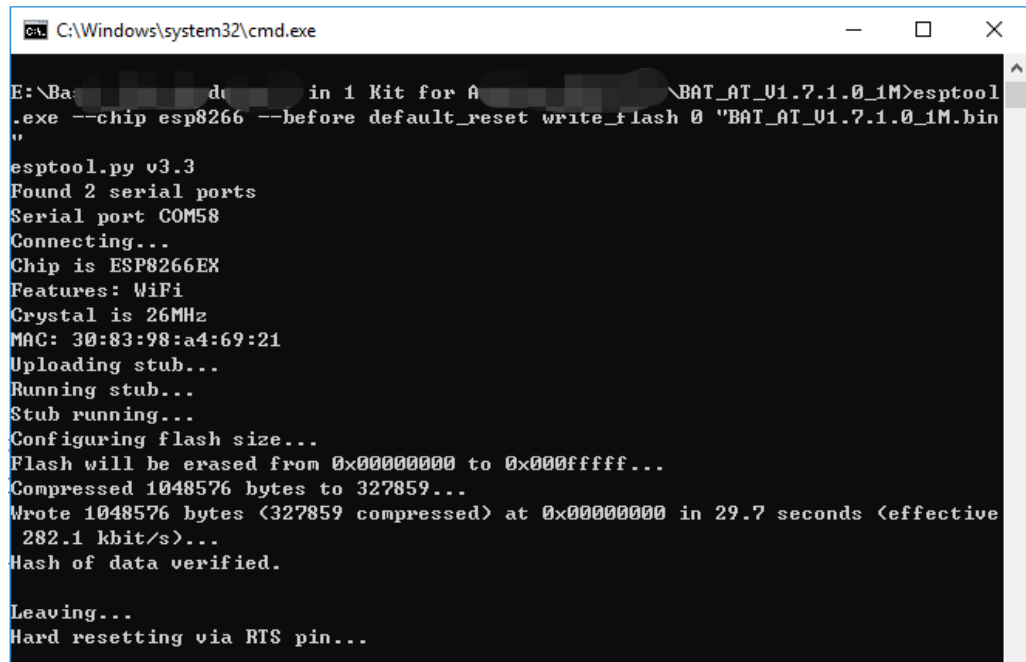
void loop() {
  if (Serial.available()) {      // If anything comes in Serial (USB),
    Serial1.write(Serial.read()); // read it and send it out Serial1 (pins 0 & 1)
  }

  if (Serial1.available()) {    // If anything comes in Serial1 (pins 0 & 1)
    Serial.write(Serial1.read()); // read it and send it out Serial (USB)
  }
}
```

3. Firmware hochladen

- Folgen Sie den untenstehenden Schritten zum Hochladen der Firmware, wenn Sie **Windows** verwenden.

1. Firmware und Brenntool herunterladen.
 - ESP8266 Firmware
2. Nach dem Entpacken sehen Sie 4 Dateien.
 - BAT_AT_V1.7.1.0_1M.bin: Die Firmware, die auf das ESP8266-Modul gebrannt wird.
 - esptool.exe: Ein Befehlszeilen-Utility für Windows.
 - install_r3.bat: Dies ist das Befehlspaket für das Windows-System.
 - install_r4.bat: Ähnlich wie install_r3.bat, aber speziell für das UNO R4 Board.
3. Doppelklicken Sie auf install_r4.bat, um das Brennen der Firmware zu starten. Wenn Sie die folgende Aufforderung sehen, wurde die Firmware erfolgreich installiert.



```

C:\Windows\system32\cmd.exe

E:\Ba... in 1 Kit for A... \BAT_AT_V1.7.1.0_1M>esptool
.exe --chip esp8266 --before default_reset write_flash 0 "BAT_AT_V1.7.1.0_1M.bin
"
esptool.py v3.3
Found 2 serial ports
Serial port COM58
Connecting...
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 30:83:98:a4:69:21
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash will be erased from 0x00000000 to 0x000fffff...
Compressed 1048576 bytes to 327859...
Wrote 1048576 bytes (327859 compressed) at 0x00000000 in 29.7 seconds (effective
282.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
  
```

Bemerkung: Wenn das Brennen fehlschlägt, überprüfen Sie bitte die folgenden Punkte.

- Setzen Sie das ESP8266-Modul zurück, indem Sie den RST am ESP8266-Adapter an GND anschließen und dann entfernen.
- Überprüfen Sie, ob die Verdrahtung korrekt ist.
- Stellen Sie sicher, dass Ihr Computer Ihr Board erkannt hat und der Port nicht belegt ist.
- Öffnen Sie die install.bat-Datei erneut.

- Zum Brennen der Firmware befolgen Sie diese Schritte, wenn Sie das **Mac OS** System verwenden.


1. Installieren Sie Esptool mit den folgenden Befehlen. Esptool ist ein Python-basiertes, Open-Source- und plattformunabhängiges Hilfsprogramm, um mit dem ROM-Bootloader in Espressif-Chips zu kommunizieren.

```
python3 -m pip install --upgrade pip
python3 -m pip install esptool
```

2. Wenn esptool ordnungsgemäß installiert ist, gibt es eine Meldung wie [usage: esptool] aus, wenn Sie `python3 -m esptool` ausführen.
3. Firmware herunterladen.
 - ESP8266 Firmware
4. Nach dem Entpacken sehen Sie 4 Dateien.
 - `BAT_AT_V1.7.1.0_1M.bin`: Die Firmware, die auf das ESP8266-Modul gebrannt wird.
 - `esptool.exe`: Ein Befehlszeilen-Utility für Windows.
 - `install_r3.bat`: Dies ist das Befehlspaket für das Windows-System.
 - `install_r4.bat`: Ähnlich wie `install_r3.bat`, aber speziell für das UNO R4 Board.
5. Öffnen Sie ein Terminal und navigieren Sie mit dem `cd`-Befehl in den Ordner mit der heruntergeladenen Firmware. Führen Sie dann den folgenden Befehl aus, um die vorhandene Firmware zu löschen und die neue Firmware zu brennen.

```
python3 -m esptool --chip esp8266 --before no_reset_no_sync erase_flash
python3 -m esptool --chip esp8266 --before no_reset_no_sync write_flash_
↪ 0 "BAT_AT_V1.7.1.0_1M.bin"
```

6. Wenn Sie die folgende Aufforderung sehen, wurde die Firmware erfolgreich installiert.



```
BAT_AT_V1.7.1.0_1M -- -bash -- 99x22
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$ python3 -m esptool --chip esp8266 --before default_reset w
rite_flash 0 "BAT_AT_V1.7.1.0_1M.bin"
esptool.py v4.3
[Found 3 serial ports
Serial port /dev/cu.usbmodem143401
Connecting...
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 30:83:98:a4:69:21
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash will be erased from 0x00000000 to 0x000fffff...
Compressed 1048576 bytes to 327859...
Wrote 1048576 bytes (327859 compressed) at 0x00000000 in 29.6 seconds (effective 283.0 kbit/s)...
Hash of data verified.

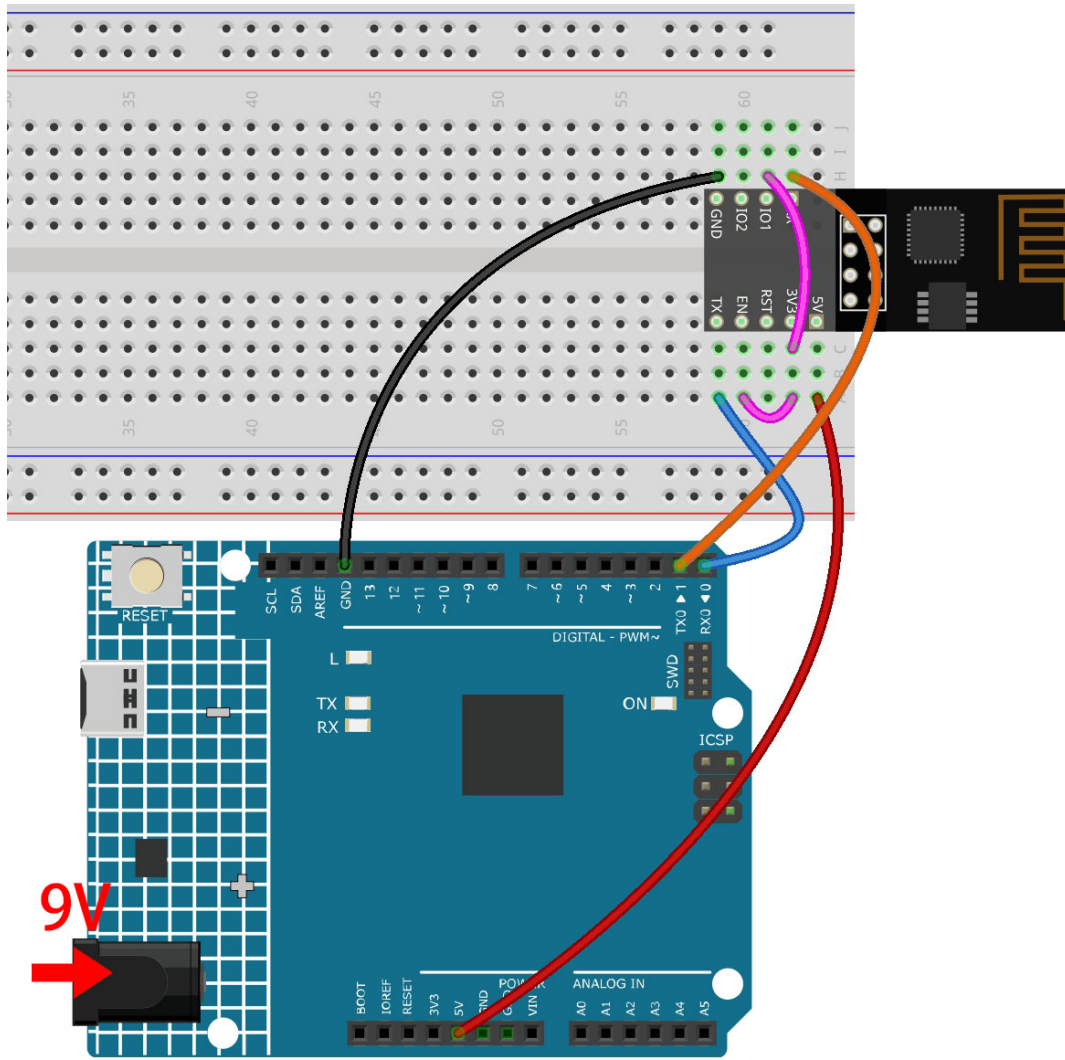
Leaving...
Hard resetting via RTS pin...
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$
```

Bemerkung: Wenn das Brennen fehlschlägt, überprüfen Sie bitte die folgenden Punkte.

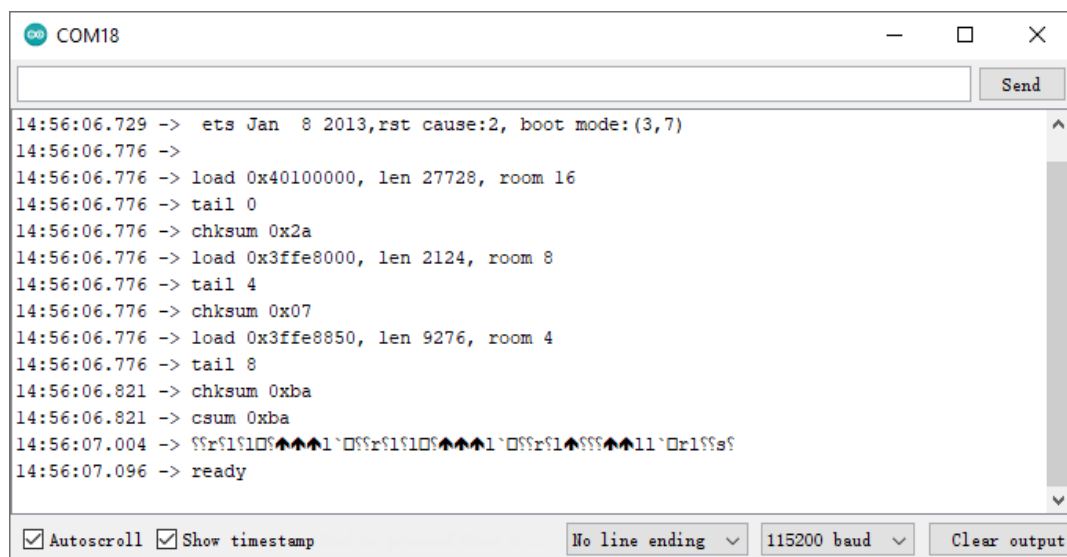
- Setzen Sie das ESP8266-Modul zurück, indem Sie den RST am ESP8266-Adapter an GND anschließen und dann entfernen.
- Überprüfen Sie, ob die Verdrahtung korrekt ist.
- Stellen Sie sicher, dass Ihr Computer Ihr Board erkannt hat und der Port nicht belegt ist.
- Öffnen Sie die `install.bat`-Datei erneut.

4. Test

1. Auf Grundlage der ursprünglichen Verkabelung verbinden Sie IO1 mit 3V3.



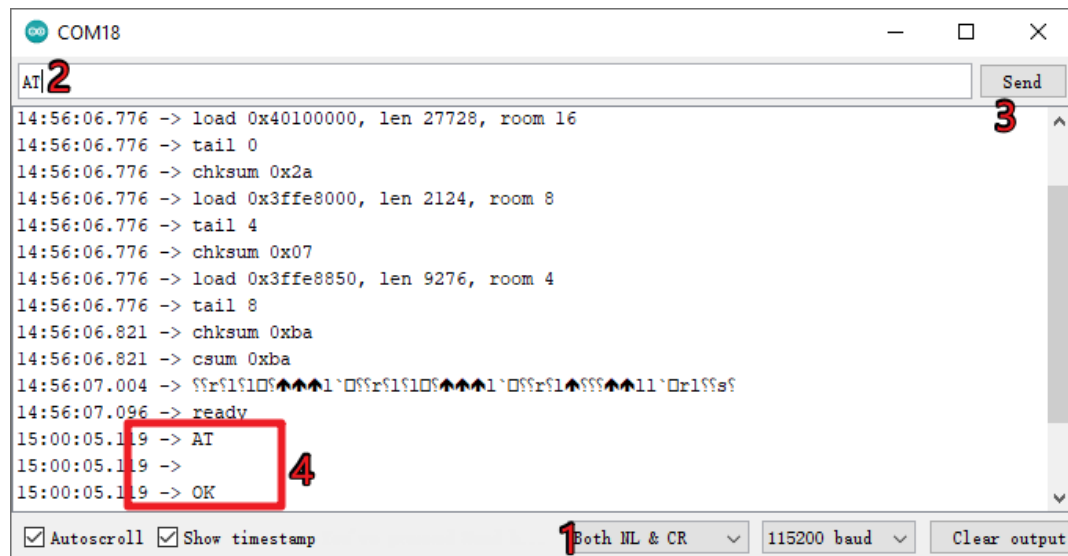
2. Sie können Informationen über das ESP8266-Modul sehen, wenn Sie auf das Lupensymbol (Serial Monitor) in der oberen rechten Ecke klicken und die Baudrate auf **115200** einstellen.



Bemerkung:

- Wenn ready nicht erscheint, können Sie versuchen, das ESP8266-Modul zurückzusetzen (verbinden Sie RST mit GND) und den Serial Monitor erneut zu öffnen.

3. Klicken Sie auf **NEWLINE DROPDOWN BOX**, wählen Sie im Dropdown-Menü both NL & CR, geben Sie AT ein. Wenn OK zurückgegeben wird, bedeutet dies, dass ESP8266 erfolgreich eine Verbindung mit Ihrem Board hergestellt hat.

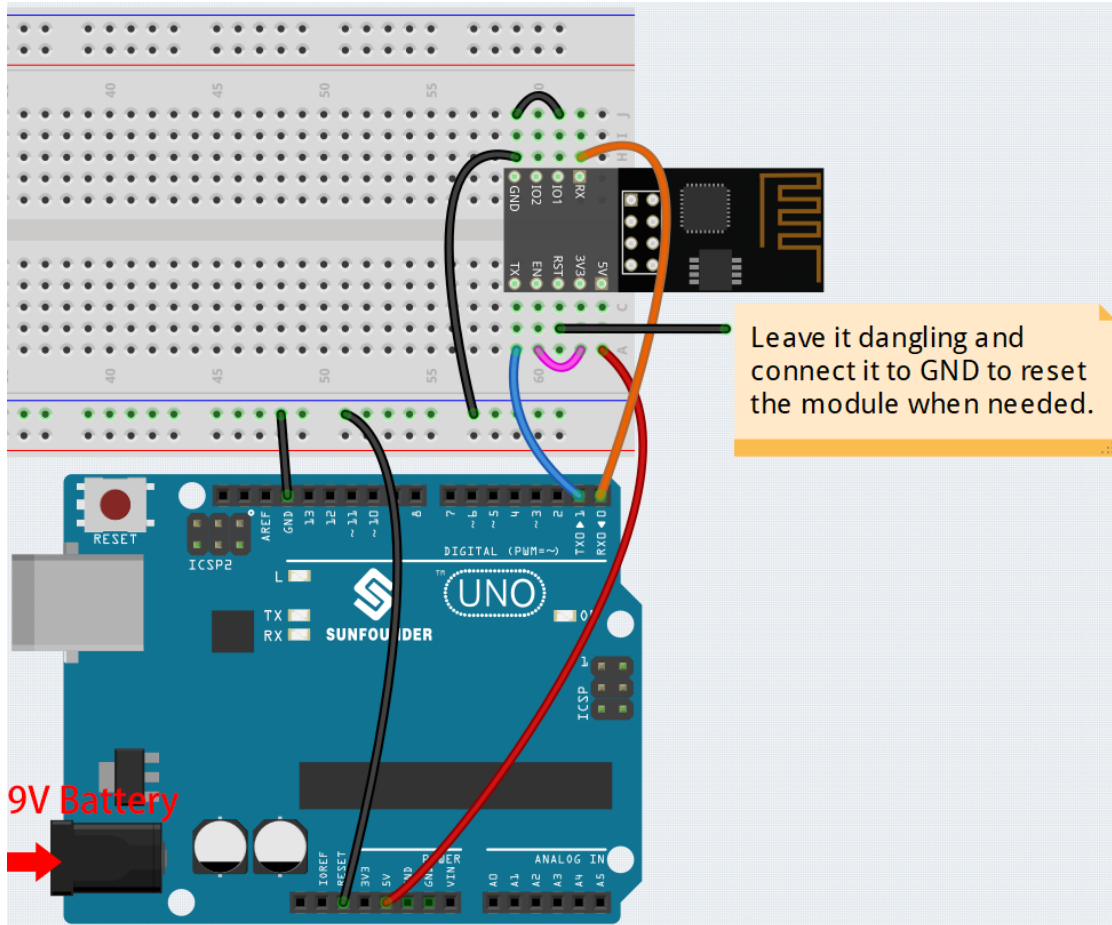


Nun können Sie weiterhin [1.1 Konfiguration des ESP8266](#) folgen, um den Arbeitsmodus und die Baudrate des ESP8266-Moduls einzustellen.

7.2.2 Firmware mit R3 erneut aufspielen

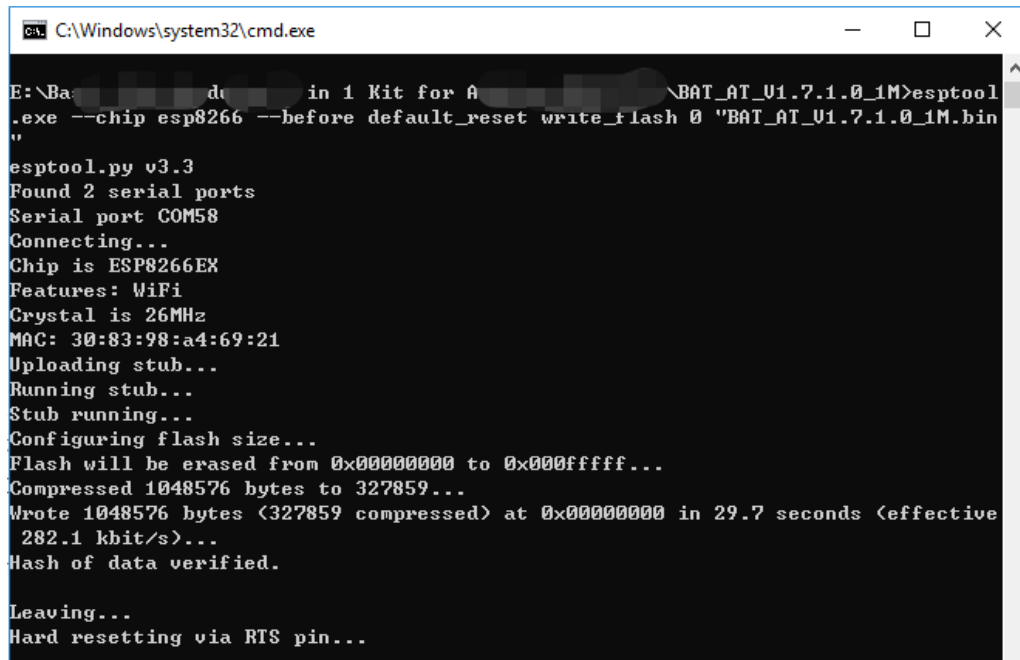
1. Schaltung aufbauen

Verbinden Sie ESP8266 mit dem SunFounder R3-Board.



2. Firmware brennen

- Befolgen Sie die folgenden Schritte, um die Firmware unter **Windows** zu brennen.
 1. Firmware und Brenn-Tool herunterladen.
 - ESP8266 Firmware
 2. Nach dem Entpacken sehen Sie 4 Dateien.
 - BAT_AT_V1.7.1.0_1M.bin: Die Firmware, die auf das ESP8266-Modul gebrannt wird.
 - esptool.exe: Ein Befehlszeilen-Tool für Windows.
 - install_r3.bat: Ein Befehlspaket für das Windows-System. Ein Doppelklick auf diese Datei führt alle Befehle in der Datei aus.
 - install_r4.bat: Wie install_r3.bat, aber speziell für das UNO R4-Board.
 3. Doppelklicken Sie auf install_r3.bat, um den Brennvorgang der Firmware zu starten. Wenn Sie die folgende Aufforderung sehen, wurde die Firmware erfolgreich installiert.



```

C:\Windows\system32\cmd.exe

E:\Ba... du... in 1 Kit for A... \BAT_AT_V1.7.1.0_1M>esptool
.exe --chip esp8266 --before default_reset write_flash 0 "BAT_AT_V1.7.1.0_1M.bin
"
esptool.py v3.3
Found 2 serial ports
Serial port COM58
Connecting...
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 30:83:98:a4:69:21
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash will be erased from 0x00000000 to 0x000fffff...
Compressed 1048576 bytes to 327859...
Wrote 1048576 bytes (327859 compressed) at 0x00000000 in 29.7 seconds (effective
282.1 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

```

Bemerkung: Wenn das Brennen fehlschlägt, überprüfen Sie bitte folgende Punkte:




- Setzen Sie das ESP8266-Modul zurück, indem Sie RST am ESP8266-Adapter an GND anschließen und dann abziehen.
- Überprüfen Sie, ob die Verkabelung korrekt ist.
- Stellen Sie sicher, dass Ihr Computer Ihr Board korrekt erkannt hat und der Port nicht belegt ist.
- Öffnen Sie die install.bat-Datei erneut.

- Um die Firmware unter **Mac OS** zu brennen, folgen Sie diesen Schritten:

1. Installieren Sie Esptool mit den folgenden Befehlen. Esptool ist ein in Python geschriebenes, Open-Source, plattformunabhängiges Tool, um mit dem ROM-Bootloader in Espressif-Chips zu kommunizieren.

```
python3 -m pip install --upgrade pip
python3 -m pip install esptool
```

2. Wenn esptool ordnungsgemäß installiert ist, gibt es eine Nachricht wie [usage: esptool] aus, wenn Sie `python3 -m esptool` ausführen.
3. Firmware herunterladen.
 - ESP8266 Firmware
4. Nach dem Entpacken sehen Sie 4 Dateien.


 BAT_AT_V1.7.1.0_1M.bin
 esptool.exe
 install.bat

- BAT_AT_V1.7.1.0_1M.bin: Die Firmware, die auf das ESP8266-Modul gebrannt wird.

- esptool.exe: Ein Befehlszeilen-Tool für Windows.
 - install_r3.bat: Ein Befehlspaket für das Windows-System.
 - install_r4.bat: Wie install_r3.bat, aber speziell für das UNO R4-Board.
5. Öffnen Sie ein Terminal und verwenden Sie den cd-Befehl, um in den gerade heruntergeladenen Firmware-Ordner zu wechseln. Führen Sie dann den folgenden Befehl aus, um die vorhandene Firmware zu löschen und die neue Firmware neu zu brennen.

```
python3 -m esptool --chip esp8266 --before default_reset erase_flash
python3 -m esptool --chip esp8266 --before default_reset write_flash 0
↪ "BAT_AT_V1.7.1.0_1M.bin"
```

6. Wenn Sie die folgende Aufforderung sehen, wurde die Firmware erfolgreich installiert.



```
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$ python3 -m esptool --chip esp8266 --before default_reset w
rite_flash 0 "BAT_AT_V1.7.1.0_1M.bin"
esptool.py v4.3
[Found 3 serial ports
Serial port /dev/cu.usbmodem143401
Connecting...
Chip is ESP8266EX
Features: WiFi
Crystal is 26MHz
MAC: 30:83:98:a4:69:21
Uploading stub...
Running stub...
Stub running...
Configuring flash size...
Flash will be erased from 0x00000000 to 0x000fffff...
Compressed 1048576 bytes to 327859...
Wrote 1048576 bytes (327859 compressed) at 0x00000000 in 29.6 seconds (effective 283.0 kbit/s)...
Hash of data verified.

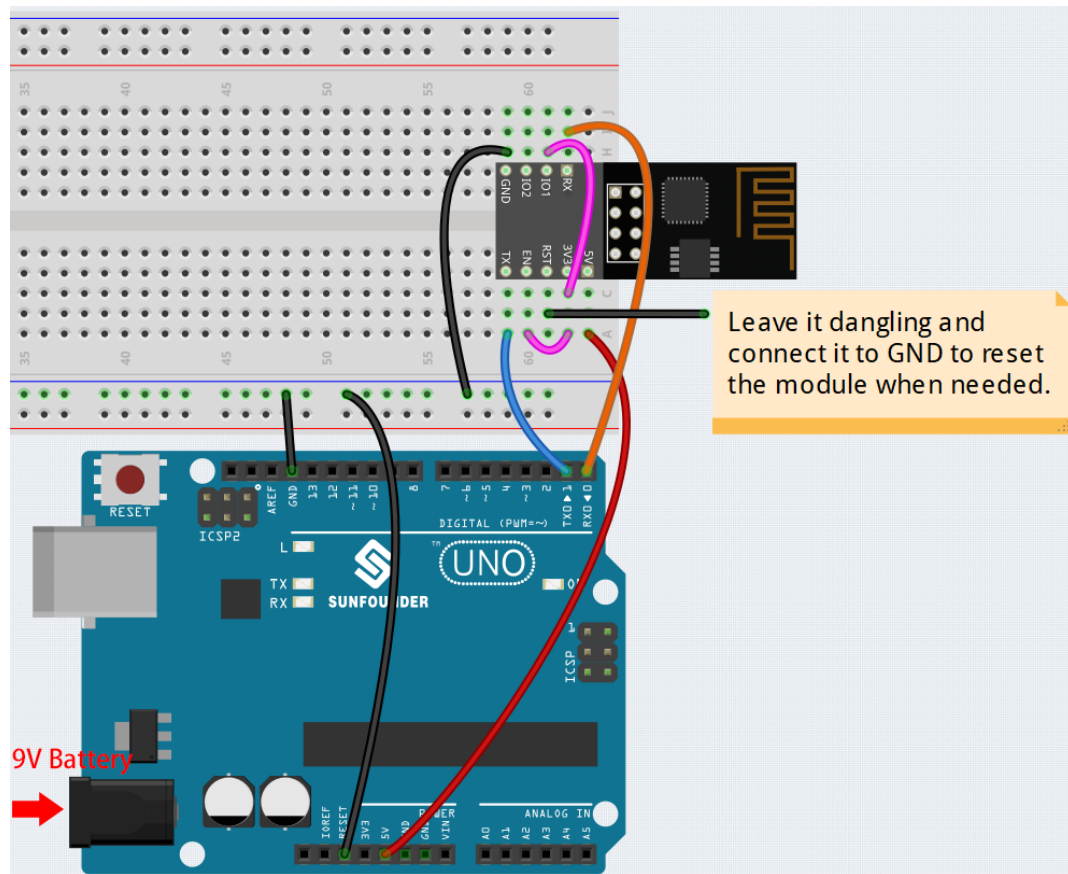
Leaving...
Hard resetting via RTS pin...
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$
```

Bemerkung: Wenn das Brennen fehlschlägt, überprüfen Sie bitte folgende Punkte:

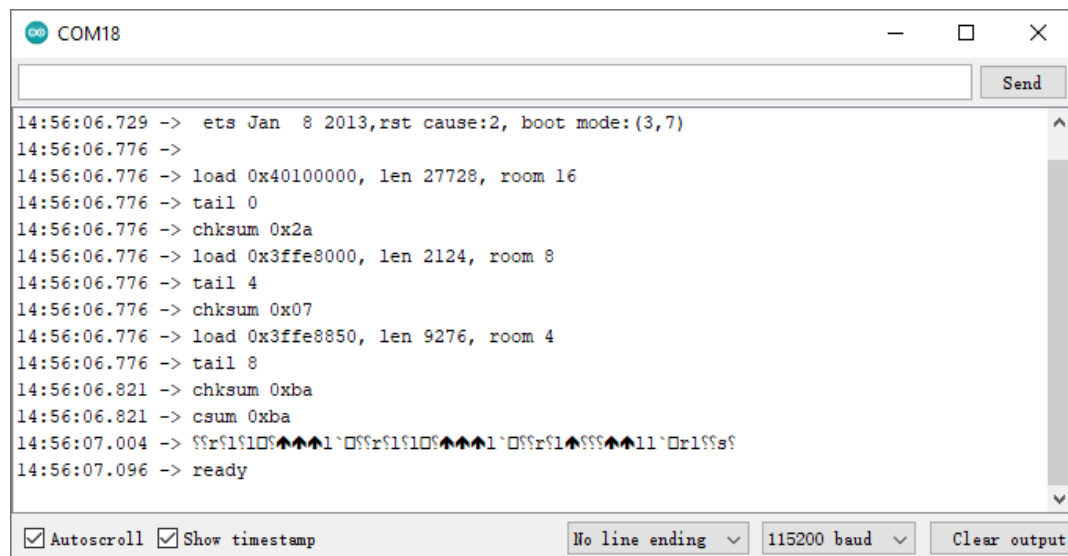
- Setzen Sie das ESP8266-Modul zurück, indem Sie RST am ESP8266-Adapter an GND anschließen und dann abziehen.
- Überprüfen Sie, ob die Verkabelung korrekt ist.
- Stellen Sie sicher, dass Ihr Computer Ihr Board korrekt erkannt hat und der Port nicht belegt ist.
- Öffnen Sie die install.bat-Datei erneut.

3. Test

1. Auf Basis der ursprünglichen Verkabelung verbinden Sie IO1 mit 3V3.



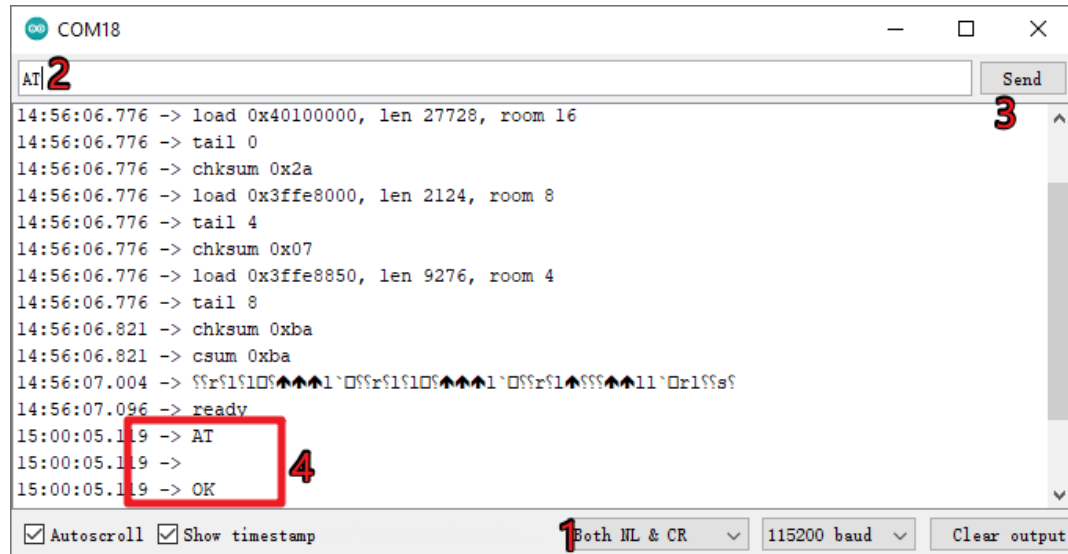
2. Sie können Informationen über das ESP8266-Modul sehen, wenn Sie auf das Lupensymbol (Serial Monitor) in der oberen rechten Ecke klicken und die Baudrate auf **115200** einstellen.



Bemerkung:

- Wenn ready nicht angezeigt wird, können Sie versuchen, das ESP8266-Modul zurückzusetzen (RST mit GND verbinden) und den Serial Monitor erneut zu öffnen.

3. Klicken Sie auf **NEWLINE DROPDOWN BOX**, wählen Sie im Dropdown-Menü both NL & CR, geben Sie AT ein. Wenn OK zurückgegeben wird, bedeutet dies, dass das ESP8266 erfolgreich eine Verbindung zum R3-Board hergestellt hat.



Nun können Sie weiterhin *1.1 Konfiguration des ESP8266* folgen, um den Arbeitsmodus und die Baudrate des ESP8266-Moduls einzustellen.

Danke

Ein Dankeschön an die Gutachter, die unsere Produkte bewertet haben, an die erfahrenen Nutzer, die Vorschläge für das Tutorial gemacht haben, und an die Anwender, die uns kontinuierlich gefolgt sind und uns unterstützt haben. Ihre wertvollen Anregungen sind unsere Motivation, bessere Produkte anzubieten!

Besonderer Dank

- Len Davisson
- Kalen Daniel
- Juan Delacosta

Würden Sie sich etwas Zeit nehmen, um diesen Fragebogen auszufüllen?

Bemerkung: Nachdem Sie den Fragebogen eingereicht haben, kehren Sie bitte zum Anfang zurück, um die Ergebnisse anzusehen.

Urheberrechtshinweis

Alle Inhalte, einschließlich, aber nicht beschränkt auf Texte, Bilder und Code in dieser Anleitung, gehören der SunFounder Company. Sie sollten es nur für persönliches Studium, Untersuchung, Vergnügen oder andere nicht kommerzielle oder gemeinnützige Zwecke verwenden, gemäß den einschlägigen Vorschriften und Urheberrechtsgesetzen, ohne die rechtlichen Rechte des Autors und der betreffenden Rechteinhaber zu verletzen. Für jede Einzelperson oder Organisation, die diese ohne Erlaubnis zu kommerziellem Gewinn verwendet, behält sich die Firma das Recht vor, rechtliche Schritte einzuleiten.