
SunFounder 3in1 Kit

www.sunfounder.com

03.01.2024

Inhaltsverzeichnis

1	Erfahren Sie mehr über die Komponenten in Ihrem Kit	3
1.1	SunFounder R3 Platine	4
1.2	ESP8266 Modul	6
1.3	Breadboard	9
1.4	Widerstand	10
1.5	Kondensator	13
1.6	Jumper-Kabel	15
1.7	74HC595	15
1.8	LED	17
1.9	RGB LED	18
1.10	7-Segment-Anzeige	20
1.11	I2C LCD1602	23
1.12	Summer	25
1.13	TT Motor	26
1.14	Servo	28
1.15	Kreiselpumpe	30
1.16	L9110 Motortreibermodul	31
1.17	Knopf	33
1.18	Reedschalter	34
1.19	Potentiometer	35
1.20	Joystick Modul	37
1.21	IR-Empfänger	39
1.22	Fotowiderstand	41
1.23	Thermistor	42
1.24	DHT11 Feuchtigkeits- und Temperatursensor	43
1.25	Linienverfolgungsmodul	45
1.26	Bodenfeuchtigkeitsmodul	46
1.27	Hinderniserkennungsmodul	48
1.28	Ultraschall-Modul	49
2	Einstieg in Arduino	51
2.1	Was ist Arduino?	51
2.2	Was kann Arduino tun?	52
2.3	Wie erstellt man ein Arduino-Projekt	52
3	Arduino-Videokurs	83
3.1	Video 1 - Vorstellung des Kits	83

3.2	Video 2 - Einführung in die Arduino IDE	84
3.3	Video 3: Grundlagen der Programmierung und LED-Projekte	84
3.4	Video 4: Zahlensysteme und Serieller Monitor	84
3.5	Video 5: Datentypen und Variablen	85
3.6	Video 6: Steuerung von Summer, Motor und Wasserpumpe	85
3.7	Video 7: Taster und Reed-Schalter	86
3.8	Video 8: PWM und Schleifenstrukturen	86
3.9	Video 9: Spannungsmessung	87
3.10	Video 10: Bedingte Anweisungen und Arrays	87
3.11	Video 11: DHT11-Sensor	87
3.12	Video 12: Funktionen und Switch-Case	88
3.13	Video 13: Joystick	88
3.14	Video 14: millis() und map()	89
3.15	Video 15: Automatisierung der Pflanzenbewässerung	89
3.16	Video 16: Infrarot-Hindernisvermeidung	90
3.17	Video 17: Unterbrechungen und LDR	90
3.18	Video 18: Servomotor	91
3.19	Video 19: Ultraschallsensor	91
3.20	Video 20: LCD 1602	92
3.21	Video 21: IR-Fernbedienung	92
3.22	Video 22: 7-Segment-Anzeige	93
3.23	Video 23: Montage eines Smart Cars	93
3.24	Video 24: Steuerung eines Smart Cars	94
3.25	Video 25: Hindernisvermeidung mit dem Smart Car	94
3.26	Video 26: Smart Car Ultraschallsensor	95
3.27	Video 27: Handverfolgung mit dem Smart Car	95
3.28	Video 28: Selbstfahrendes Smart Car	96
3.29	Video 29: Fernsteuerung des Smart Cars	96
3.30	Video 30: Linienverfolgung mit dem Smart Car	96
3.31	Video 31: WLAN-Temperaturüberwachung	97
4	Code herunterladen¶	99
5	Grundprojekte	101
5.1	1. Digitaler Schreibvorgang	101
5.2	2. Analoges Schreiben	111
5.3	3. Digitaler Lesevorgang	119
5.4	4. Analoges Auslesen	132
5.5	5. Weitere Syntax	147
5.6	6. Lustiges Projekt	210
6	Auto Projekte	235
6.1	Auto zusammenbauen	235
6.2	1. Bewegung	247
6.3	2. Bewegung durch Code	252
6.4	3. Beschleunigung	256
6.5	4. Die Linie verfolgen	257
6.6	5. Mit dem Hindernisvermeidungs-Modul spielen	261
6.7	6. Spielen mit dem Ultraschallmodul	265
6.8	7. Folge Deiner Hand	268
6.9	8. Autonomes Fahren	270
6.10	9. Fernbedienung	273
6.11	10. Start mit einem Tastendruck	278
6.12	11. Geschwindigkeitskalibrierung	280

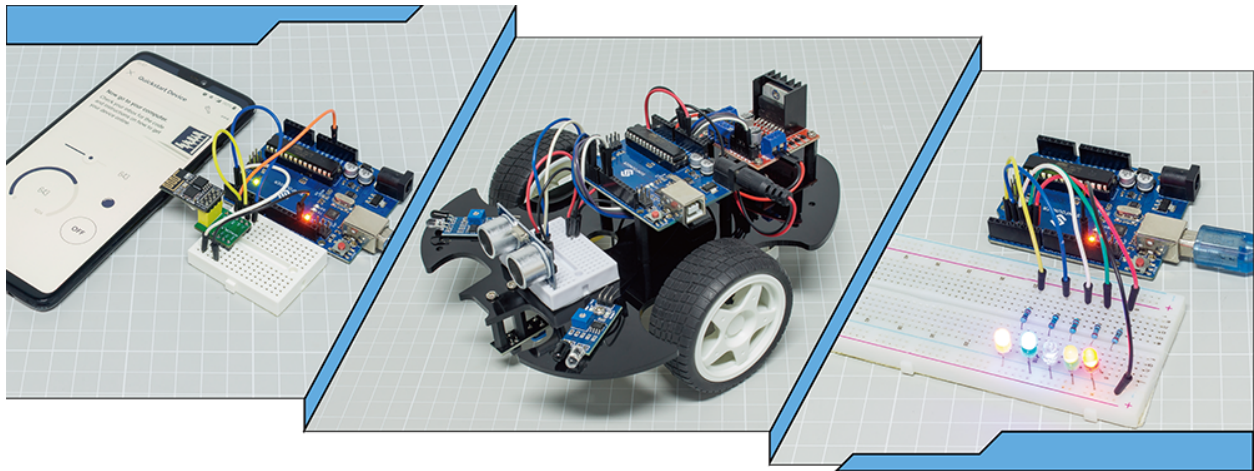
7	IoT-Projekte	283
7.1	1. Erste Schritte mit Blynk	284
7.2	2. Daten von Blynk abrufen	298
7.3	3. Daten an Blynk senden	309
7.4	4. Cloud-Musikplayer	313
7.5	5. Überwachung des häuslichen Umfelds	319
7.6	6. Pflanzenüberwachung	327
7.7	7. Strombegrenzendes Tor	334
7.8	8. IoT Auto	341
8	Mit Scratch spielen	351
8.1	1.1 PictoBlox installieren	352
8.2	1.2 Einführung in die Benutzeroberfläche	353
8.3	1.3 Schnellanleitung zu PictoBlox	354
8.4	2.1 Tischlampe	372
8.5	2.2 Atmende LED	378
8.6	2.3 Bunte Kugeln	385
8.7	2.4 LCD1602	393
8.8	2.5 Bewegende Maus	402
8.9	2.6 Türklingel	409
8.10	2.7 Niedrigtemperaturalarm	416
8.11	2.8 Lichtwecker	423
8.12	2.9 Temperatur und Feuchtigkeit auslesen	430
8.13	2.10 Pendel	436
8.14	2.11 Drehender Ventilator	445
8.15	2.12 Lichtempfindlicher Ball	452
8.16	2.13 SPIEL - Schießen	465
8.17	2.14 SPIEL - Den Ballon aufblasen	479
8.18	2.15 SPIEL - Sternengekreuzt	489
8.19	2.16 SPIEL - Apfel essen	498
8.20	2.17 SPIEL - Flappy Papagei	509
8.21	2.18 SPIEL - Breakout-Klon	518
8.22	2.19 SPIEL - Angeln	530
8.23	2.20 SPIEL - Klicke nicht auf das weiße Feld	539
8.24	2.21 SPIEL - Schütze dein Herz	560
8.25	2.22 SPIEL - Drachen töten	577
8.26	3.1 Das Auto testen	601
8.27	3.2 Bewegung	606
8.28	3.3 Der Linie folgen	613
8.29	3.4 Folge Deiner Hand	617
8.30	3.5 Hindernisvermeidung	621
8.31	3.6 Deiner Hand folgen 2	629
8.32	3.7 Hindernisvermeidung 2	633
9	Videokurse	639
10	FAQ	643
10.1	Wie verwendet man Blynk auf einem mobilen Gerät?	643
10.2	Wie kann die Firmware für das ESP8266-Modul neu geflasht werden?	645
11	Danke	657
12	Urheberrechtshinweis	659

Danke, dass Sie sich für unser 3-in-1 Starterset entschieden haben.

Bemerkung: Dieses Dokument ist in den folgenden Sprachen verfügbar.

-
-
-
-
-

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



Haben Sie online ein Lernset gekauft, das nur mit einem einfachen PDF oder einem Heft mit den Schritten zum Aufbau des Projekts geliefert wurde?

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

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

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

Im 3-in-1 Starterset finden Sie einen vollständigen Arduino-Kurs, der Anfängern hilft, Arduino zu lernen, sowie eine Vielzahl interessanter Projekte, die andere Lernsets nicht bieten, wie Smart Car-Projekte und IoT-Projekte. Sie werden Arduino meistern, solange Sie dem Kurs des Sets Schritt für Schritt folgen. Statt nur Code zu kopieren und einzufügen, werden Sie Ihren eigenen Code schreiben und Ihr Arduino-Projekt nach Ihren Wünschen umsetzen.

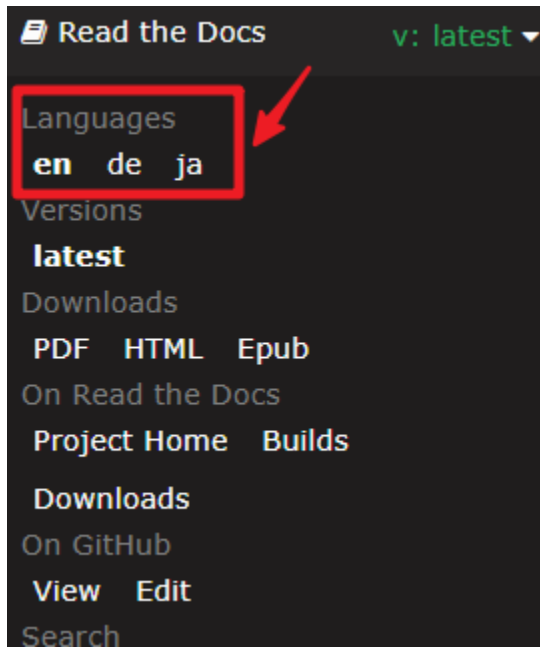
Darüber hinaus bietet das Set auch 30+ Scratch-Programmierprojekte für jüngere Schüler und Enthusiasten, sodass Anfänger keine Programmiererfahrung benötigen, um ihre eigenen Kreationen zu schreiben und zu realisieren!

Los geht's! Fangen Sie mit Vertrauen an, Arduino von Null auf Held zu programmieren!

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

Über die Anzeigesprache

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



Contents

KAPITEL 1

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.

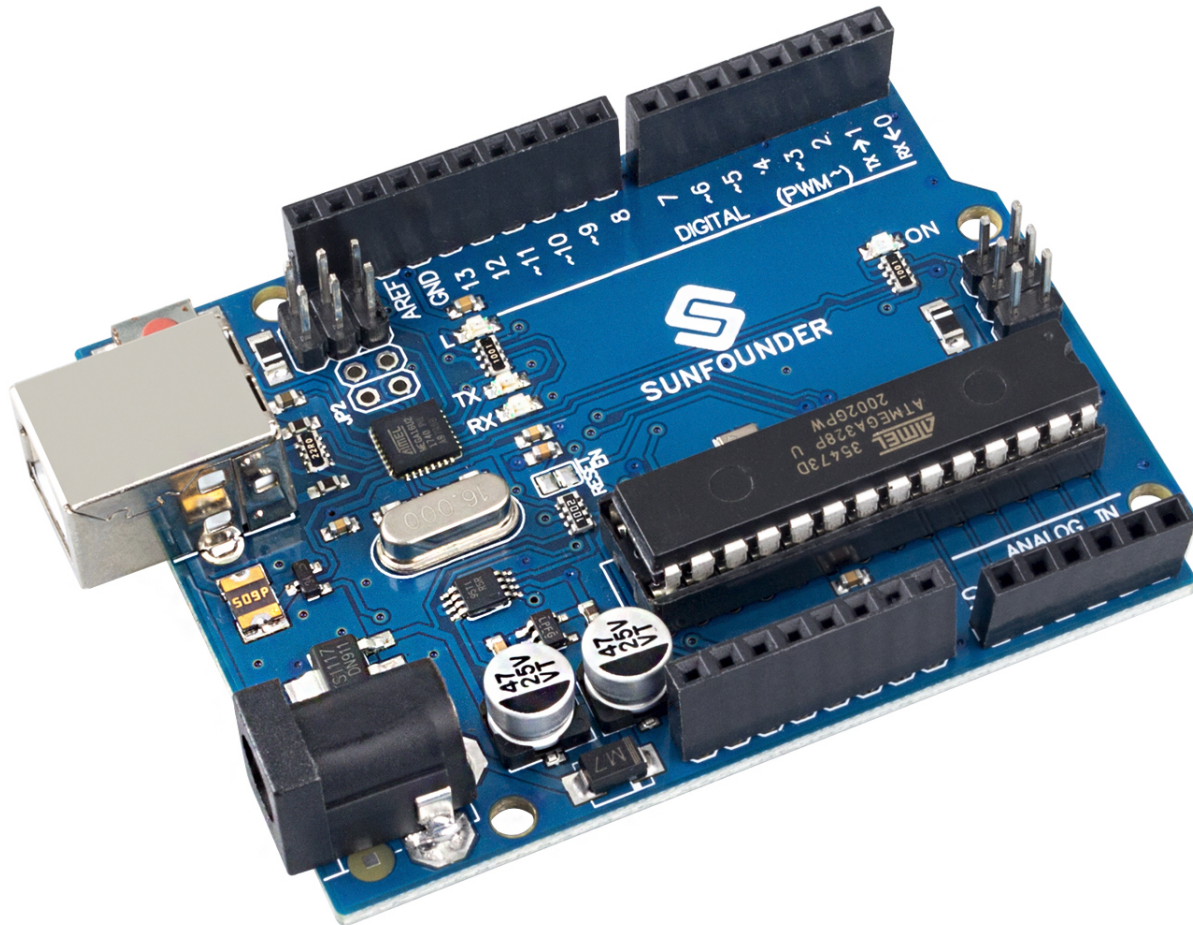
Packing List

 Resistor(10Ω) 10 pcs	 Resistor(100Ω) 10 pcs	 Resistor(220Ω) 30 pcs	 Resistor(330Ω) 10 pcs	 Resistor(1KΩ) 10 pcs	 Resistor(2KΩ) 10 pcs	 Resistor(5.1KΩ) 10 pcs	 Resistor(10KΩ) 10 pcs	 Resistor(100KΩ) 10 pcs
 Resistor(1MΩ) 10 pcs	 Reed Switch 2 pcs	 Button 5 pcs	 Active/Passive Buzzer 1+1 pcs	 Capacitor 104 pF 5 pcs	 Potentiometer 1 pc	 74HC595 1 pc	 7-segment Display 1 pc	 IR Receiver 1 pc
 Green LED 5 pcs	 Red LED 5 pcs	 White LED 5 pcs	 Blue LED 5 pcs	 Yellow LED 5 pcs	 RGB LED 1 pc	 Photoresistor/ Thermistor 1+1 pcs	 M2.5x6 Screw 10 pcs	 M3 Nut 8 pcs
 M3x10 Screw 4 pcs	 M3x6 Screw 30 pcs	 M3x30 Screw 6 pcs	 M2.5x11 Standoff 6 pcs	 M3x24 Standoff 7 pcs	 M3x12 Standoff 10 pcs	 Screwdriver 1 pc	 DHT11 1 pc	 ESP8266 Module 1 pc
 Soil Moisture Module 1 pc	 Joystick Module 1 pc	 Ultrasonic Module 1 pc	 IR Obstacle Avoidance Module 2 pcs	 I2C LCD 1602 1 pc	 R3 Board 1 pc	 Line Tracking Module 1 pc	 L9110 Module 1 pc	 ESP8266 Adapter Module 1 pc
 USB Cable 1 pc	 TT Wheel 2 pcs	 TT Motor 2 pcs	 1" Universal Wheel 1 pc	 Remote Control 1 pc	 Pump 1 pc	 Acrylic Plate 1 pc	 9G Servo 1 pc	 Mini Breadboard 1 pc
 Breadboard 1 pc	 Velcro 4 pcs	 Tube 1 pc	 Jump Wire F/M 20 pcs	 F/F DuPont Wire 20 pcs	 9V Battery Cable 1 pc	 Jump Wire M/M 45 pcs	 9V Battery 1 pc	Online Tutorials: https://3in1-kit-v2.rtfid.io

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

Steuerplatine

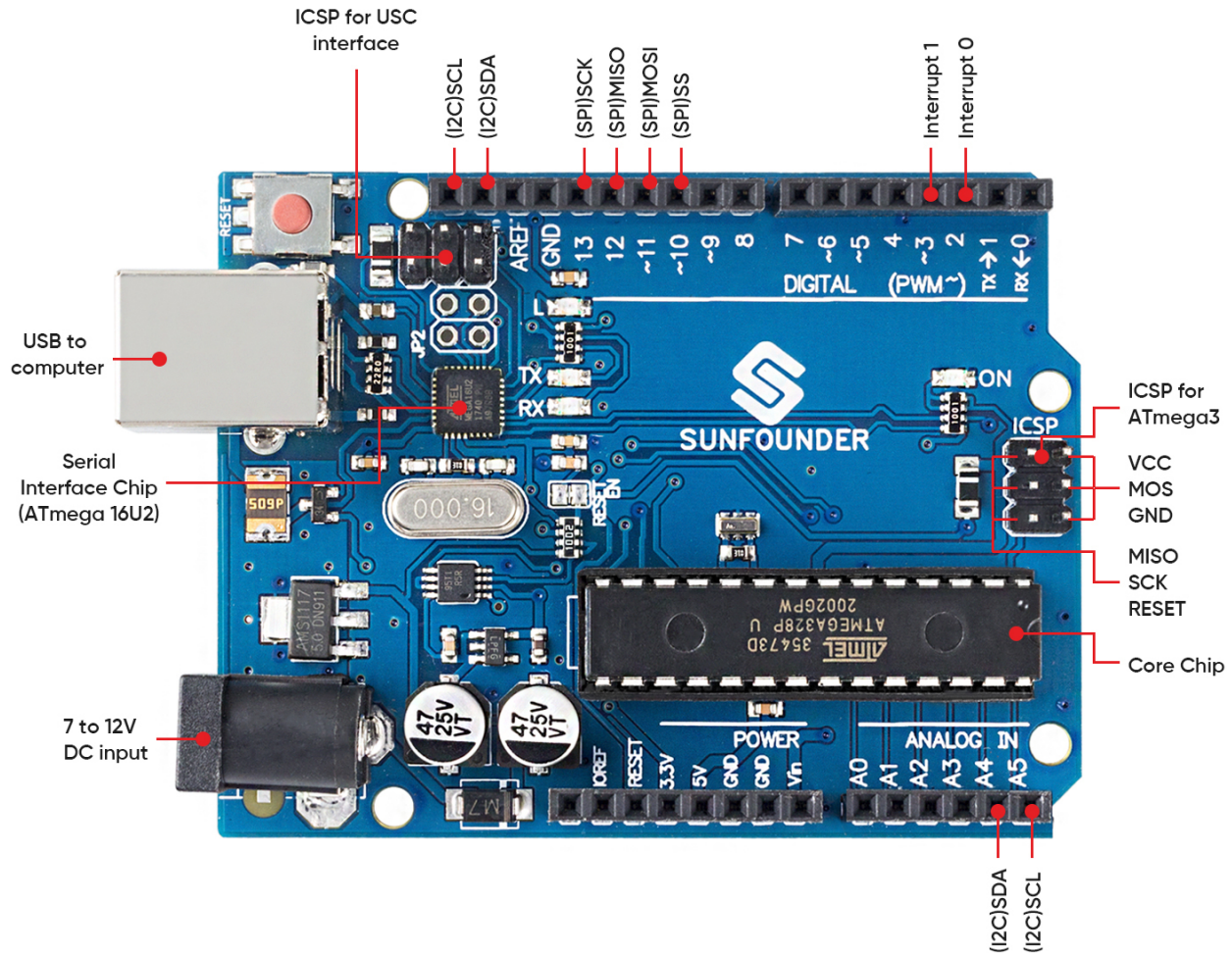
1.1 SunFounder R3 Platine



Bemerkung: Die SunFounder R3 Platine ist eine Hauptplatine mit fast denselben Funktionen wie das [Arduino Uno](#). Beide Platinen können austauschbar verwendet werden.

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

Technische Parameter



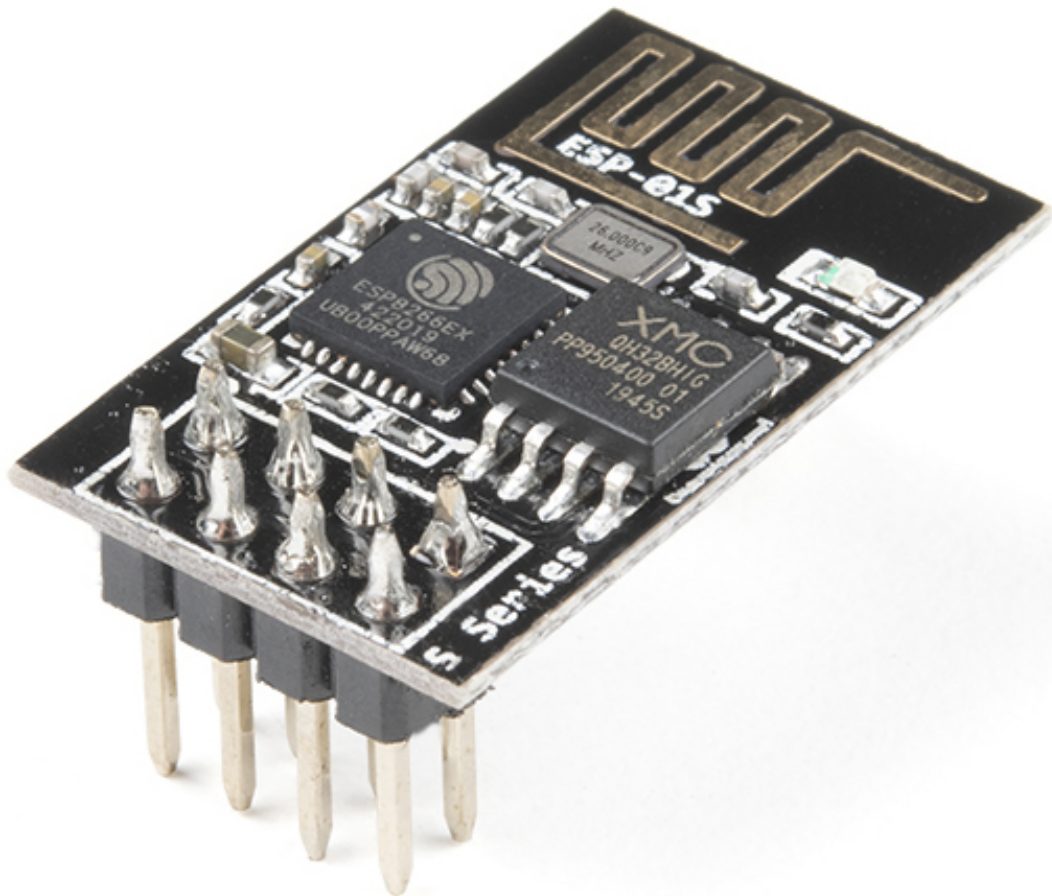
- MIKROCONTROLLER: ATmega328P
- BETRIEBSSPANNUNG: 5V
- EINGANGSSPANNUNG (EMPFOHLEN): 7-12V
- EINGANGSSPANNUNG (GRENZWERTE): 6-20V
- DIGITALE I/O-PINS: 14 (0-13, von denen 6 PWM-Ausgabe bieten (3, 5, 6, 9-11))
- PWM DIGITALE I/O-PINS: 6 (3, 5, 6, 9-11)
- ANALOGE EINGANGSPINS: 6 (A0-A5)
- GLEICHSTROM PRO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

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

Weiterführende Informationen

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

1.2 ESP8266 Modul

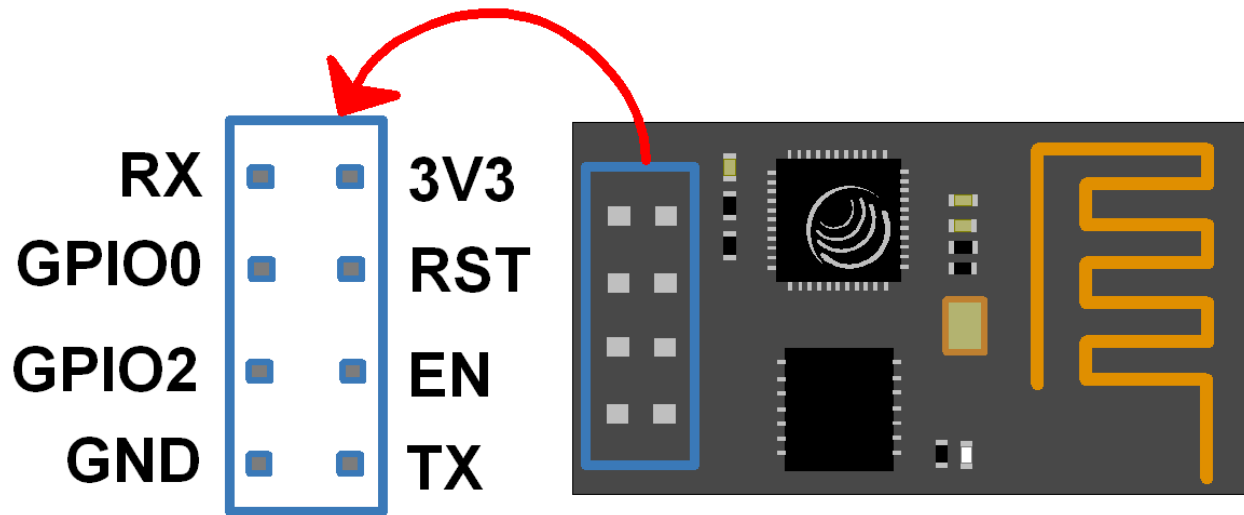


Das ESP8266 ist ein kostengünstiger Wi-Fi-Mikrochip mit integrierter TCP/IP-Netzwerksoftware und Mikrocontroller-Fähigkeit, produziert von Espressif Systems in Shanghai, China.

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:

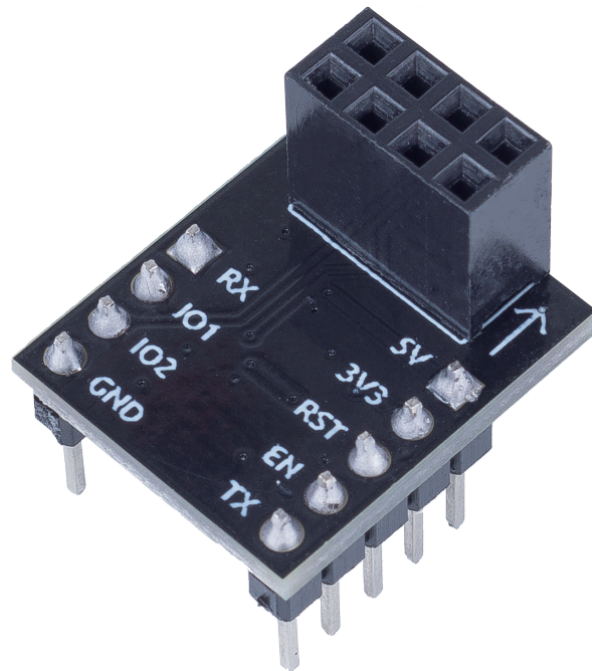


Tab. 1: ESP8266-01 Pins

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](#)

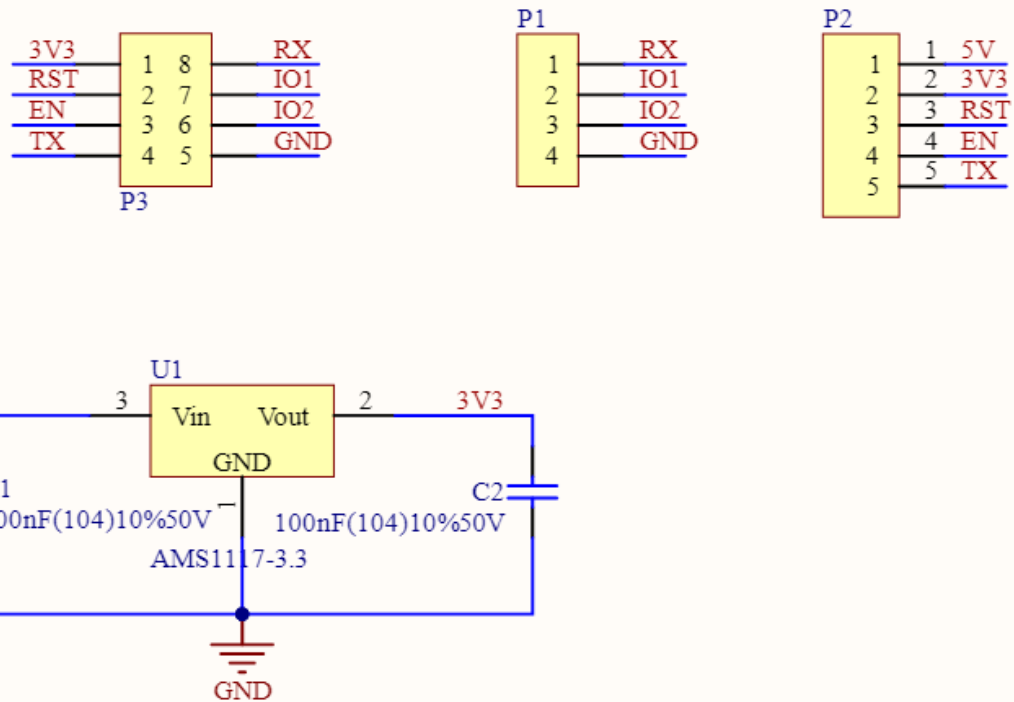
1.2.1 ESP8266 Adapter



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:

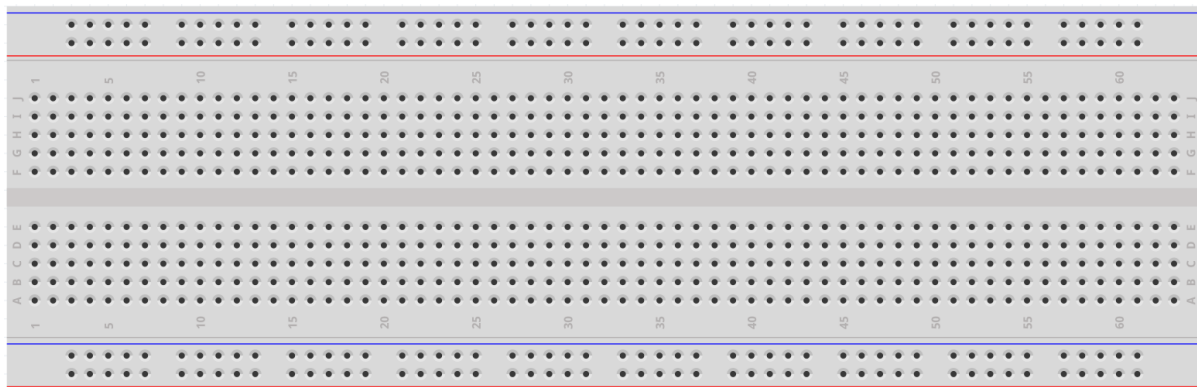


Beispiel

- *IoT-Projekte* (IoT-Projekt)

Grundlagen

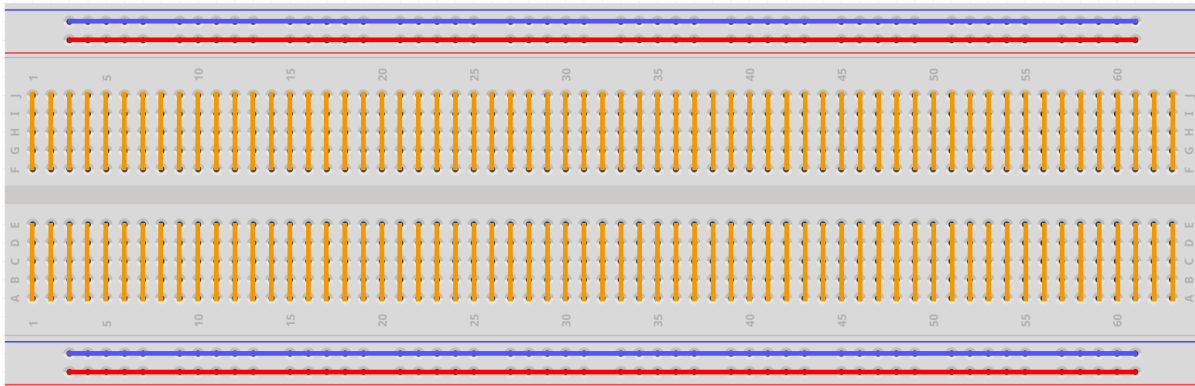
1.3 Breadboard



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



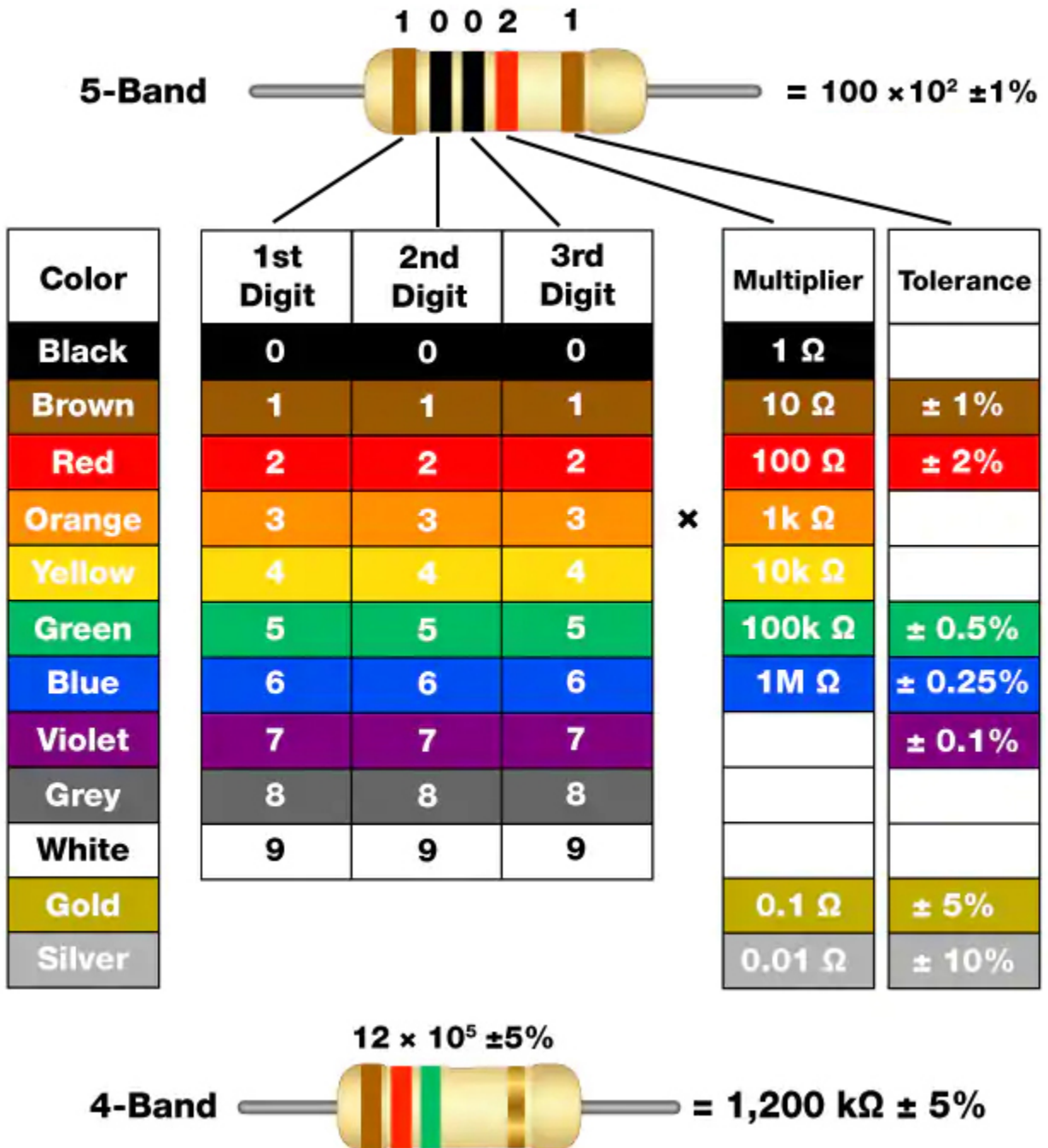
Ein Widerstand ist ein elektronisches Element, das den Strom in einem Zweig begrenzen kann. Ein fester Widerstand ist eine Art von Widerstand, dessen Widerstandswert nicht verändert werden kann, während der eines Potentiometers oder eines variablen Widerstands einstellbar ist.

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.

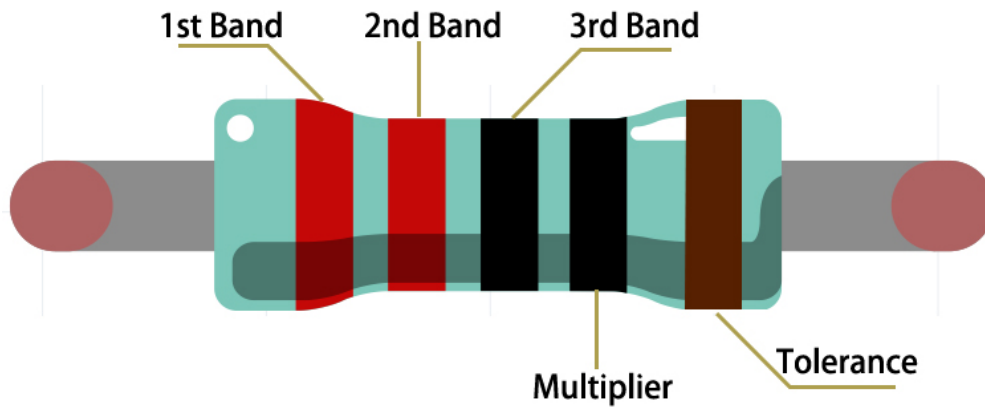
Schwarz	Braun	Rot	Orange	Gelb	Grün	Blau	Violett	Grau	Weiß	Gold	Silber
0	1	2	3	4	5	6	7	8	9	0.1	0.01

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 Keramik Kondensatoren und Elektrolytkondensatoren verwendet.

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

Auf den Keramik Kondensatoren 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

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

Beispiel

- [2.6 Türklingel](#) (Scratch-Projekt)
- [2.16 SPIEL - Apfel essen](#) (Scratch-Projekt)
- [2.19 SPIEL - Angeln](#) (Scratch-Projekt)

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.

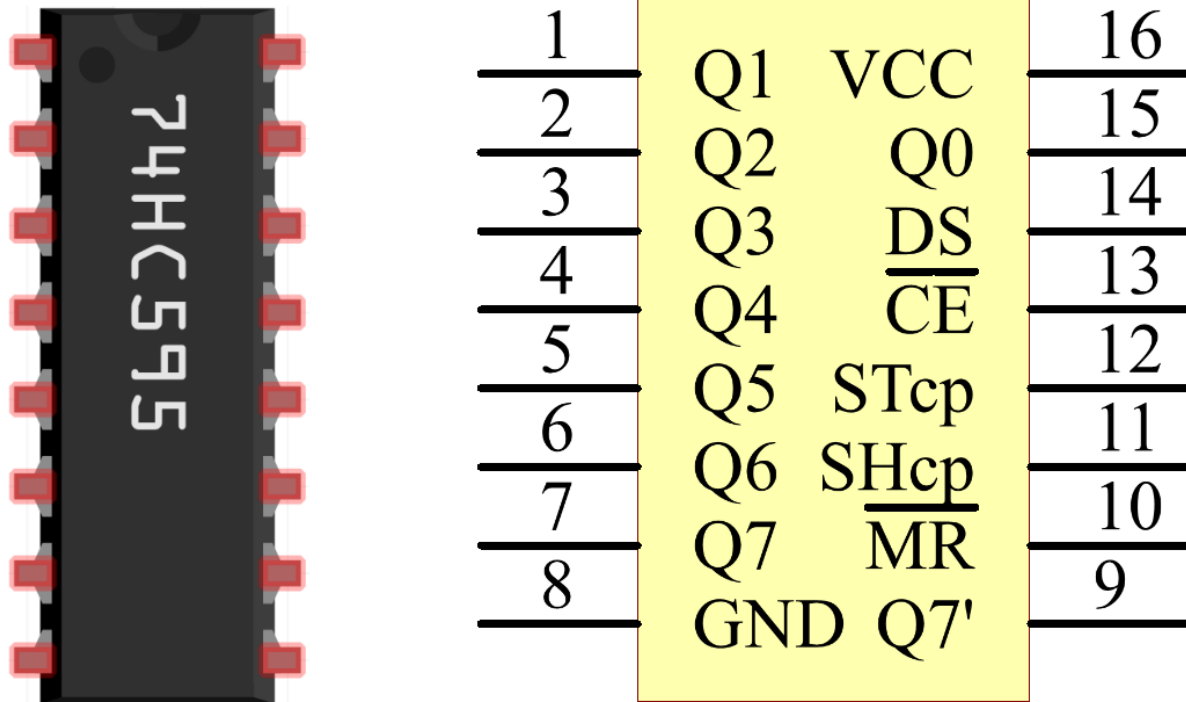
Chip

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:

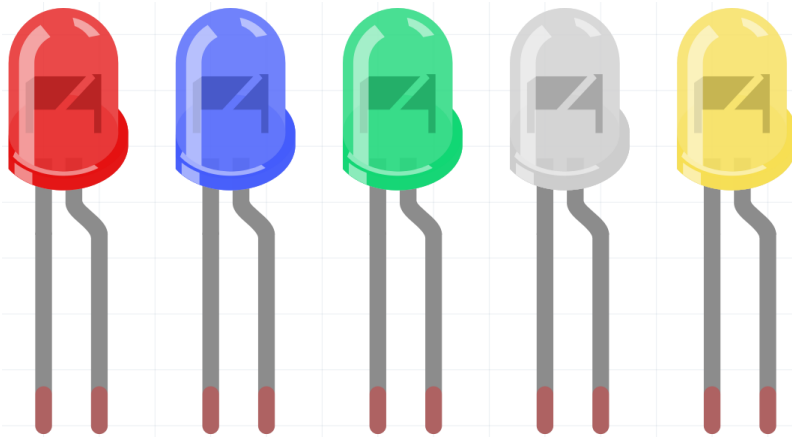
- **Q0-Q7**: 8-Bit-parallele Daten-Ausgangspins, können 8 LEDs oder 8 Pins einer 7-Segment-Anzeige direkt steuern.
- **Q7'**: Serieller Ausgangspin, verbunden mit DS eines anderen 74HC595, um mehrere 74HC595 in Serie zu verbinden.
- **MR**: Reset-Pin, aktiv auf niedrigem Pegel.
- **SHcp**: Zeitsequenzeingabe 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. Strombegrenzendes Tor](#) (IoT-Projekt)

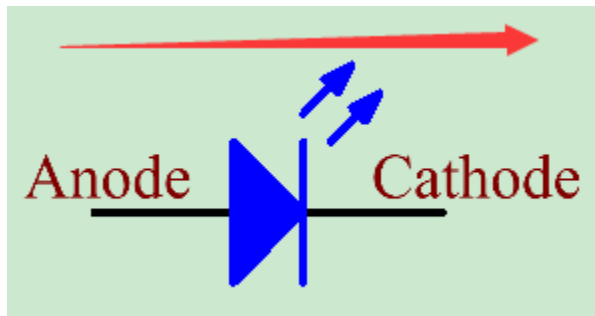
Anzeige

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.

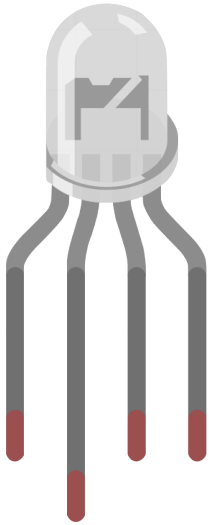
Hier finden Sie eine detaillierte Einführung in die LED: [LED - Wikipedia](#).

Beispiel

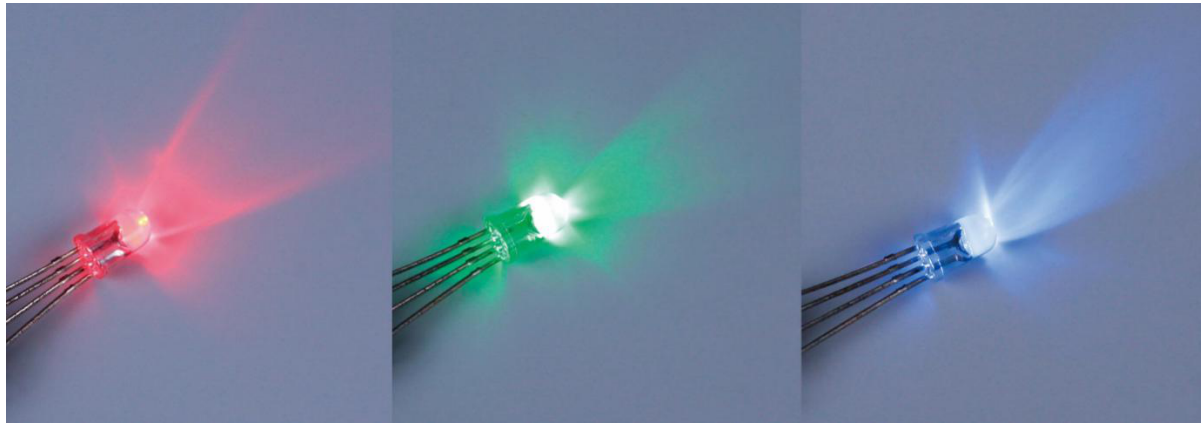
- [1.1 Hallo, LED!](#) (Grundlegendes Projekt)
- [2.1 Überblenden](#) (Grundlegendes Projekt)
- [2. Daten von Blynk abrufen](#) (IoT-Projekt)

- 2.2 *Atmende LED* (Scratch-Projekt)
- 2.1 *Tischlampe* (Scratch-Projekt)

1.9 RGB LED

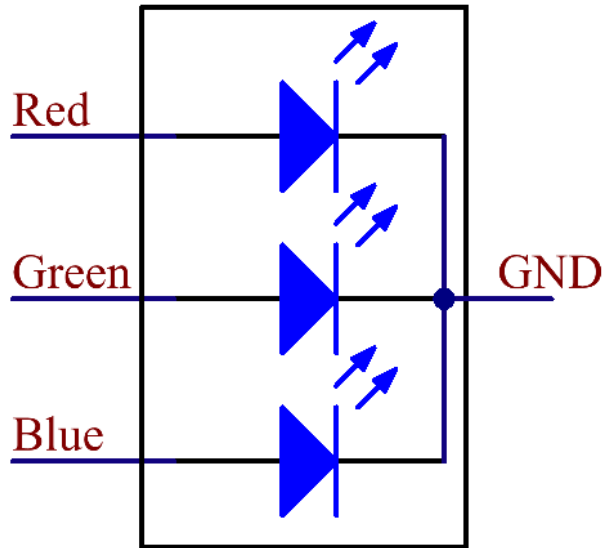


RGB-LEDs emittieren Licht in verschiedenen Farben. Eine RGB-LED kombiniert drei LEDs in den Farben Rot, Grün und Blau in einer transparenten oder halbtransparenten Kunststoffhülle. Durch das Ändern der Eingangsspannung der drei Pins und deren Überlagerung kann sie, statistischen Angaben zufolge, 16.777.216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

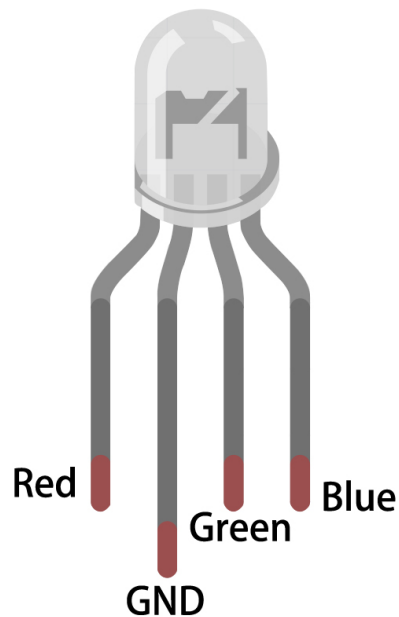


RGB-LEDs können in gemeinsame Anoden und gemeinsame Kathoden unterteilt werden. In diesem Bausatz wird die letztere verwendet. Die **gemeinsame Kathode**, oder GK, bedeutet, dass die Kathoden der drei LEDs verbunden sind. Wenn man sie mit GND verbindet und die drei Pins anschließt, wird die LED die entsprechende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

Ihr Schaltungssymbol ist im folgenden Bild dargestellt.



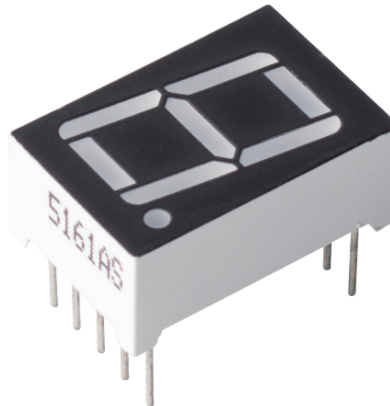
Eine RGB-LED hat 4 Pins: Der längste ist GND; die anderen sind Rot, Grün und Blau. Wenn man die Kunststoffhülle berührt, findet man eine Kerbe. Der dem Einschnitt am nächsten gelegene Pin ist der erste Pin, gekennzeichnet als Rot, dann GND, Grün und anschließend Blau.



Beispiel

- [2.2 Bunttes Licht](#) (Grundlegendes Projekt)
- [5.2 Schwellenwert](#) (Grundlegendes Projekt)
- [2.3 Bunte Kugeln](#) (Scratch-Projekt)

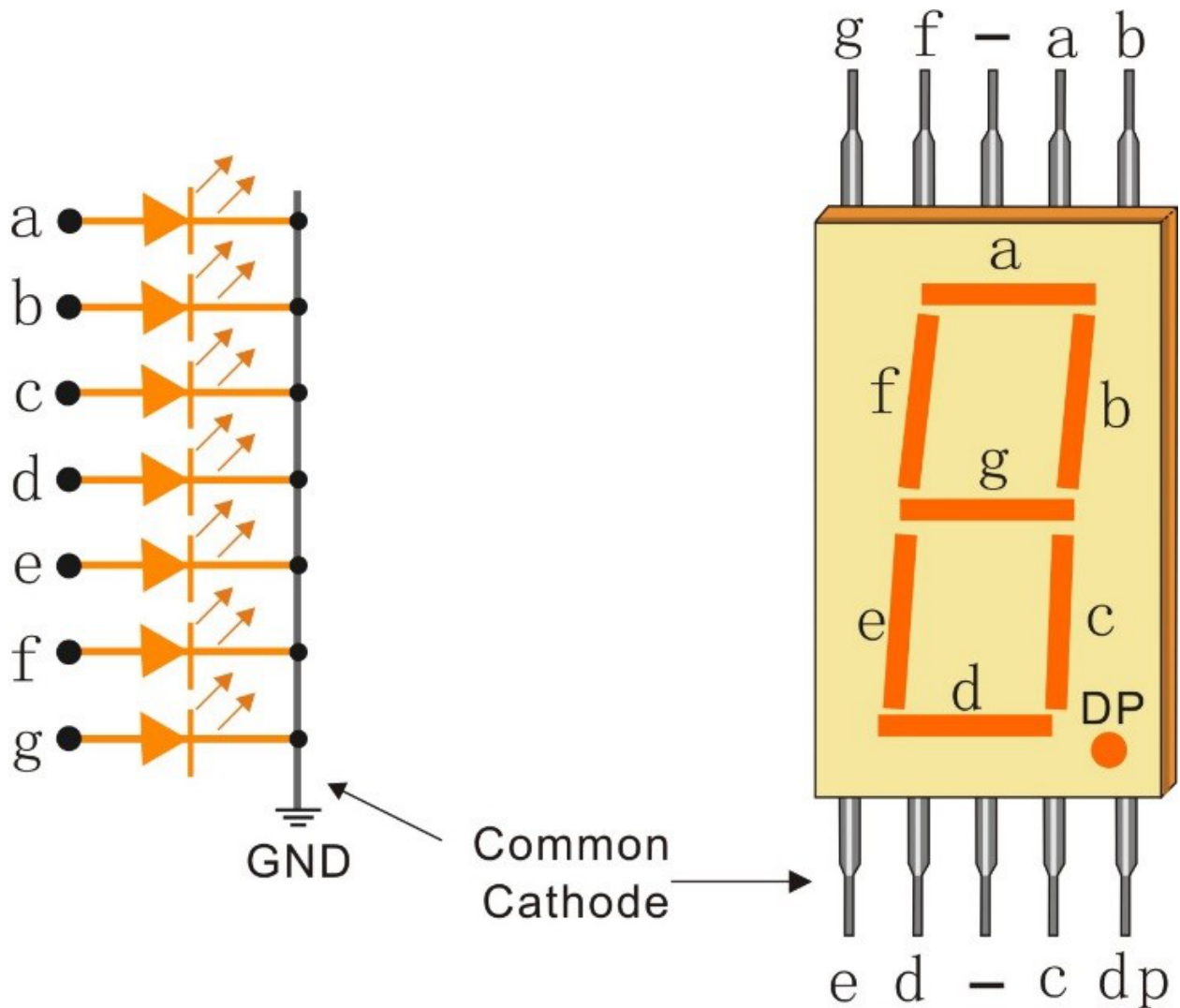
1.10 7-Segment-Anzeige



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

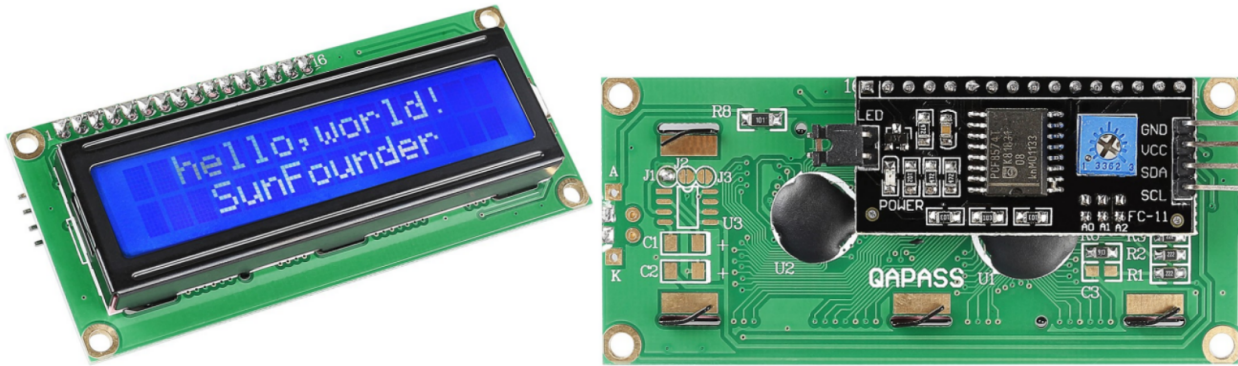
Um Ihnen zu helfen zu verstehen, wie 7-Segment-Anzeigen (Gemeinsame Kathode) Zahlen darstellen, haben wir die folgende Tabelle gezeichnet. Zahlen sind die Zahlen 0-F, die auf der 7-Segment-Anzeige dargestellt werden; (DP) GFEDCBA bezieht sich auf die entsprechende LED, die auf 0 oder 1 eingestellt ist. 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

- *5.10 ShiftOut(Segmentanzeige)* (Grundlegendes Projekt)
- *7. Strombegrenzendes 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.

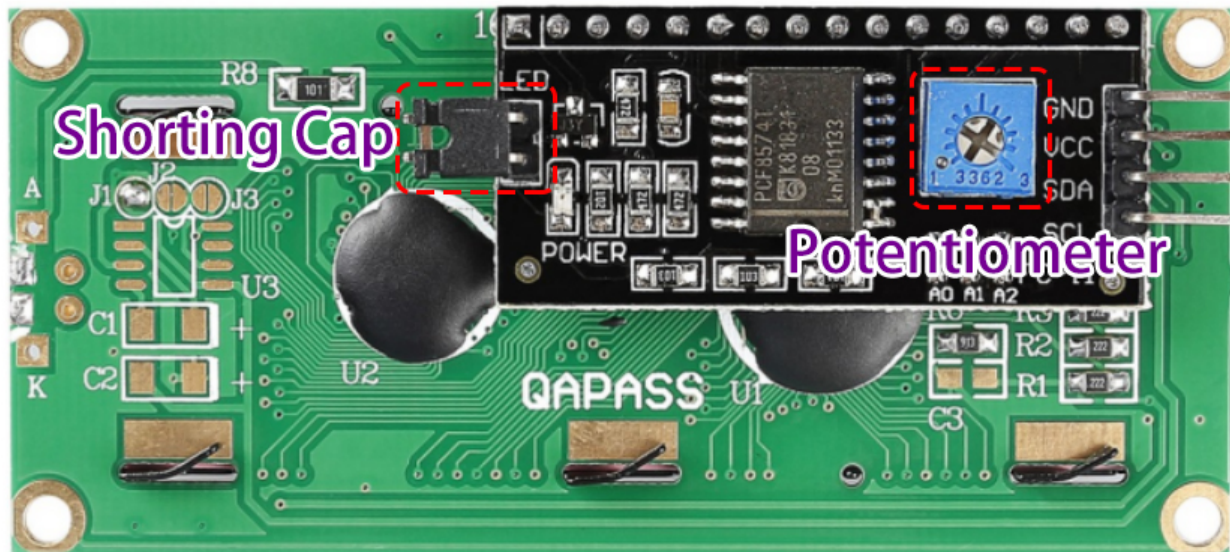
Nimmt man die Standardadresse von 0x27 als Beispiel, kann die Geräteadresse durch Kurzschließen der A0/A1/A2-Pads geändert werden; im Standardzustand sind A0/A1/A2 auf 1, und wenn das Pad kurzgeschlossen ist, sind A0/A1/A2 auf 0.

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 Serielle Lektüre](#) (Grundlegendes Projekt)

Ton

1.12 Summer



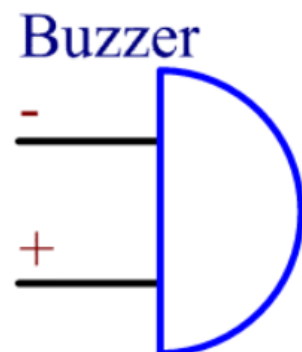
Als eine Art von elektronischem Summer mit integrierter Struktur, die mit Gleichstrom versorgt werden, werden Summer häufig in Computern, Druckern, Fotokopierern, Alarmen, elektronischen Spielzeugen, automobilen Elektronikgeräten, Telefonen, Timern und anderen elektronischen Produkten oder Sprachgeräten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

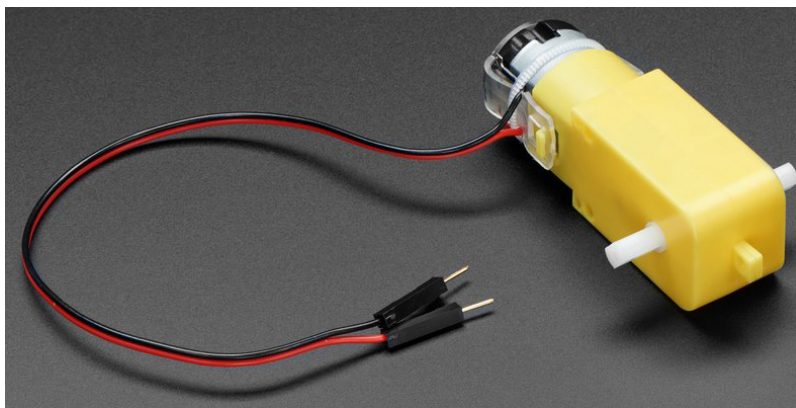
[Summer - Wikipedia](#)

Beispiel

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

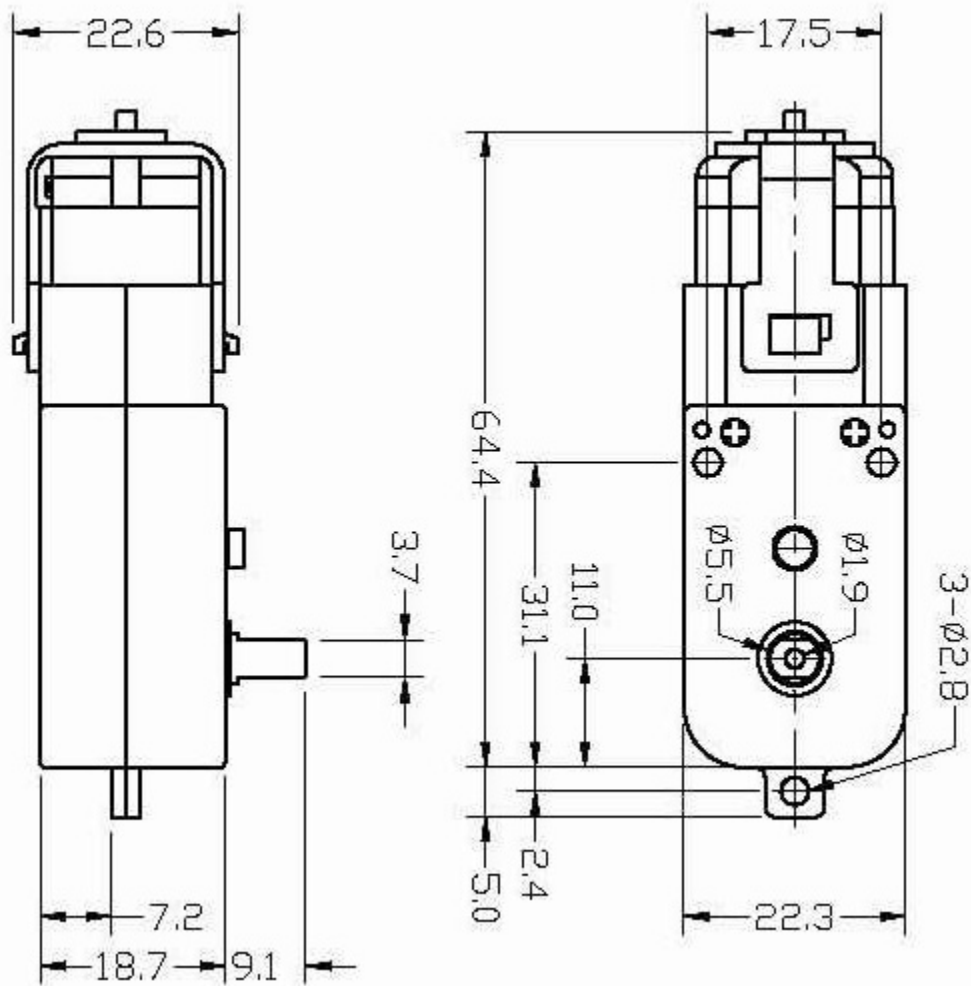
Diese Motoren können mit einer Spannung von 3 ~ 6VDC betrieben werden. Bei höheren Spannungen werden sie natürlich etwas schneller laufen.

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

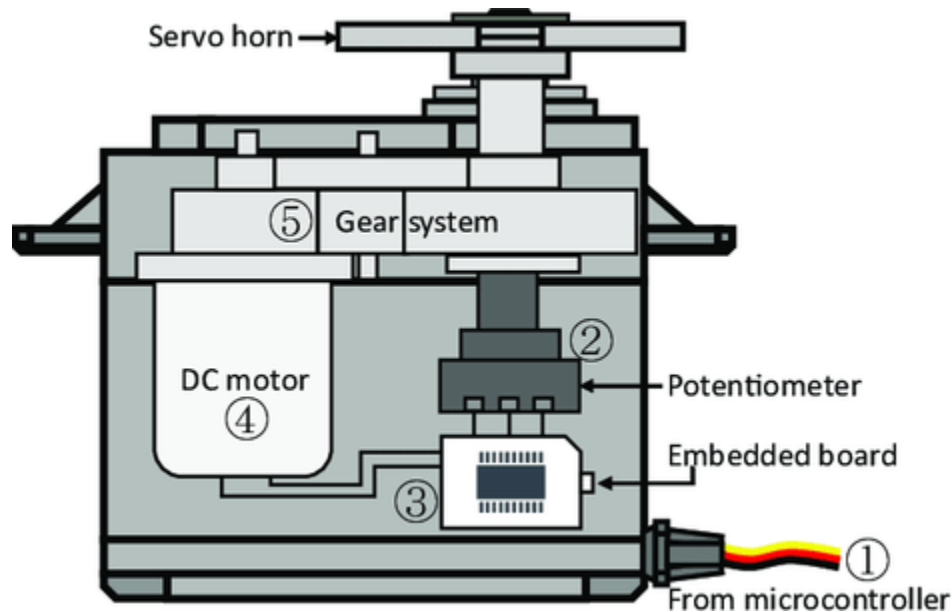
- [1.3 Das Rad drehen](#) (Grundlegendes Projekt)
- [1. Bewegung](#) (Auto-Projekt)
- [3. Beschleunigung](#) (Auto-Projekt)
- [8. IoT Auto](#) (IoT-Projekt)
- [3.1 Das Auto testen](#) (Scratch-Projekt)

1.14 Servo

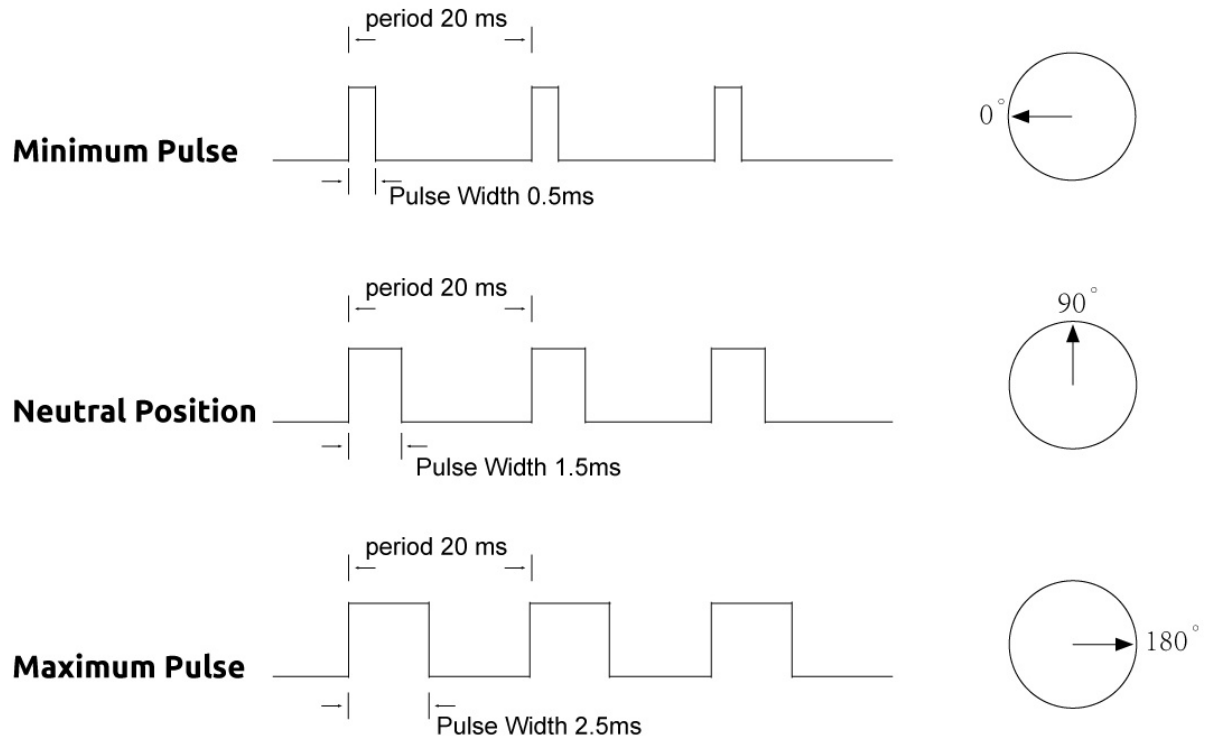


Ein Servo besteht in der Regel aus folgenden Teilen: Gehäuse, Welle, Getriebesystem, Potentiometer, Gleichstrommotor und eingebettetem Board.

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.

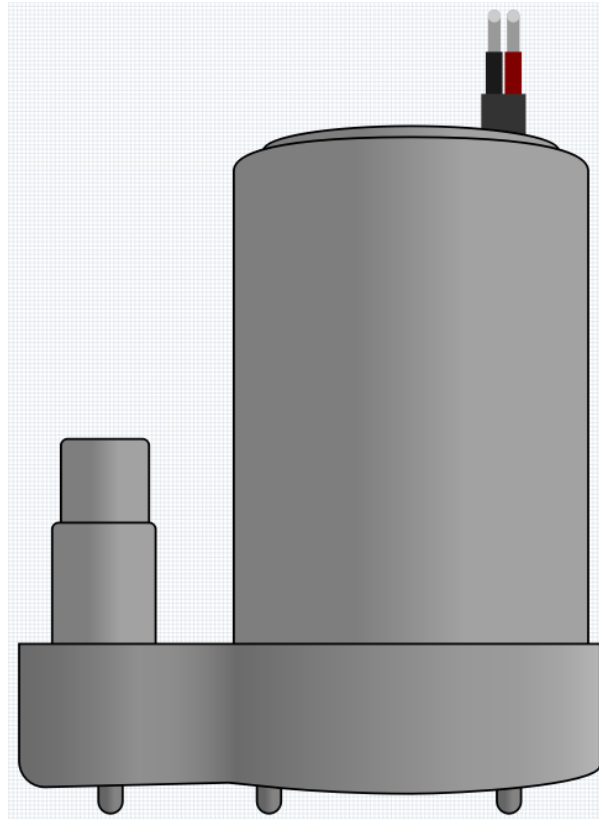


Der Winkel wird durch die Dauer eines Impulses bestimmt, der auf das Steuerkabel gegeben wird. Dies wird als Pulsweitenmodulation bezeichnet. Das Servo erwartet alle 20 ms einen Impuls. Die Länge des Impulses bestimmt, wie weit der Motor dreht. Ein 1,5ms-Impuls lässt den Motor beispielsweise auf die 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 verwenden](#) (Grundlegendes Projekt)
- [7. Strombegrenzendes Tor](#) (IoT-Projekt)
- [2.10 Pendel](#) (Scratch-Projekt)

1.15 Kreispumpe



Die Kreispumpe 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 Kreispumpen sind das Pumpen von Wasser, Abwasser, in der Landwirtschaft, für Erdöl und Petrochemie.

- [Kreispumpe - 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

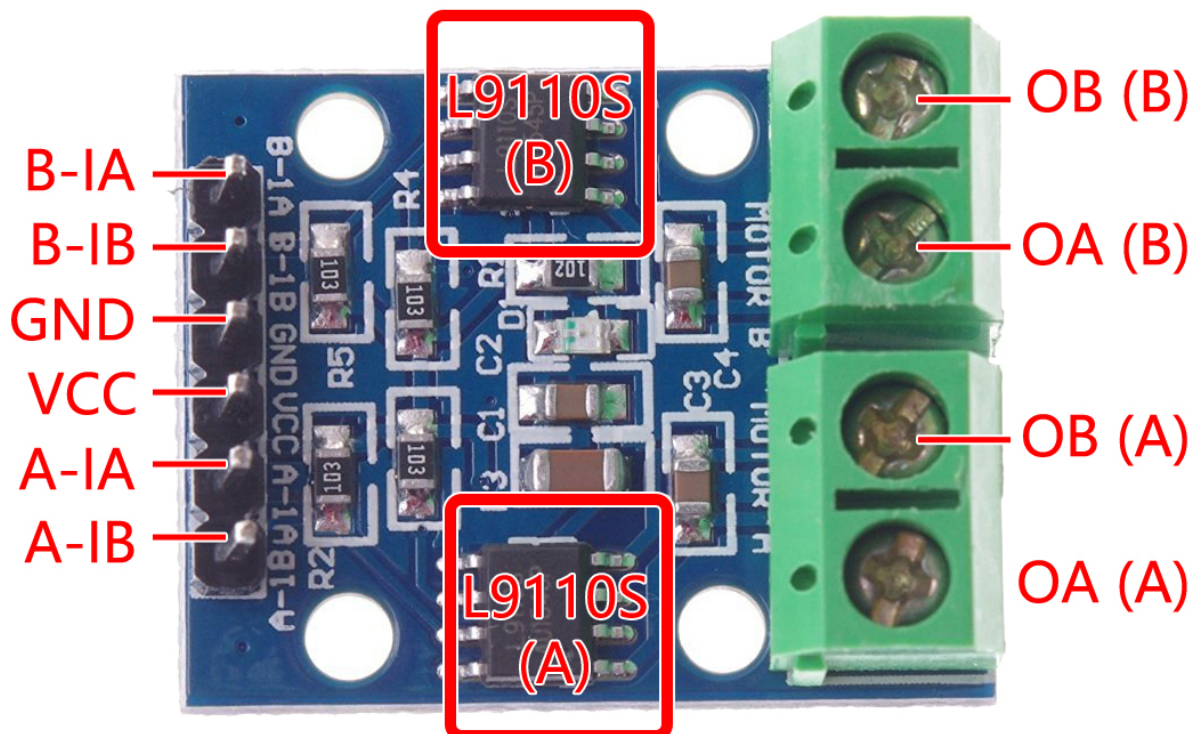
Das L9110 Motortreibermodul eignet sich hervorragend zum Ansteuern von zwei Motoren gleichzeitig. Es beherbergt ein Paar unabhängiger L9110S-Treiberchips, wobei jeder Kanal einen konstanten Stromausgang von bis zu 800mA bietet.

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 B-1B(B-2A) gibt die verschiedenen Zustände von Motor B basierend auf den Werten der Eingangspins B-1A und B-1B. 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:

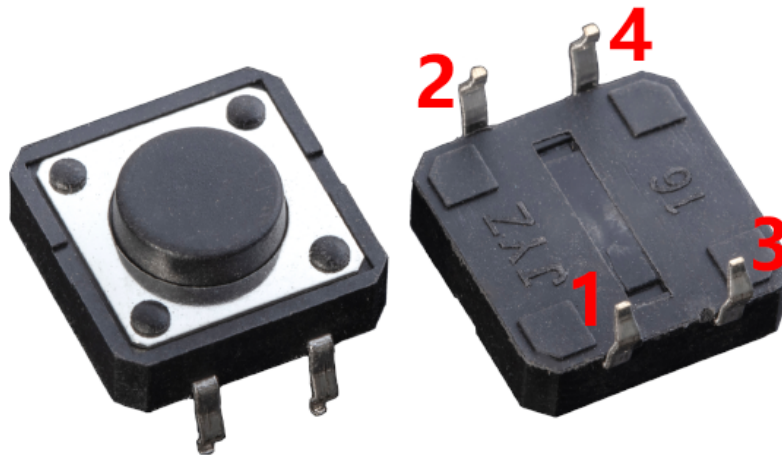
Diese Wahrheitstabelle zeigt die verschiedenen Zustände von Motor A basierend auf den Werten der Eingangspins A-1A und A-1B. Sie gibt die Drehrichtung (im Uhrzeigersinn oder gegen den Uhrzeigersinn), Bremsen oder Anhalten von Motor A an.

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)

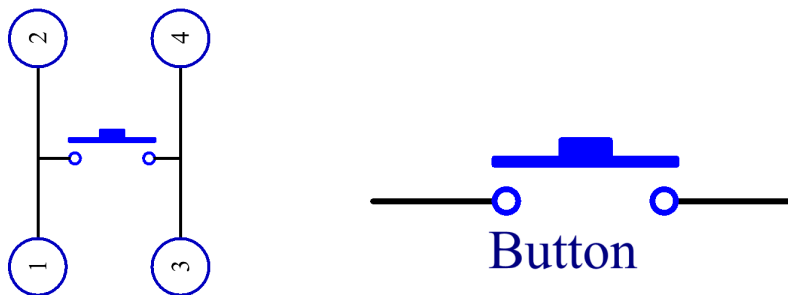
Controller

1.17 Knopf

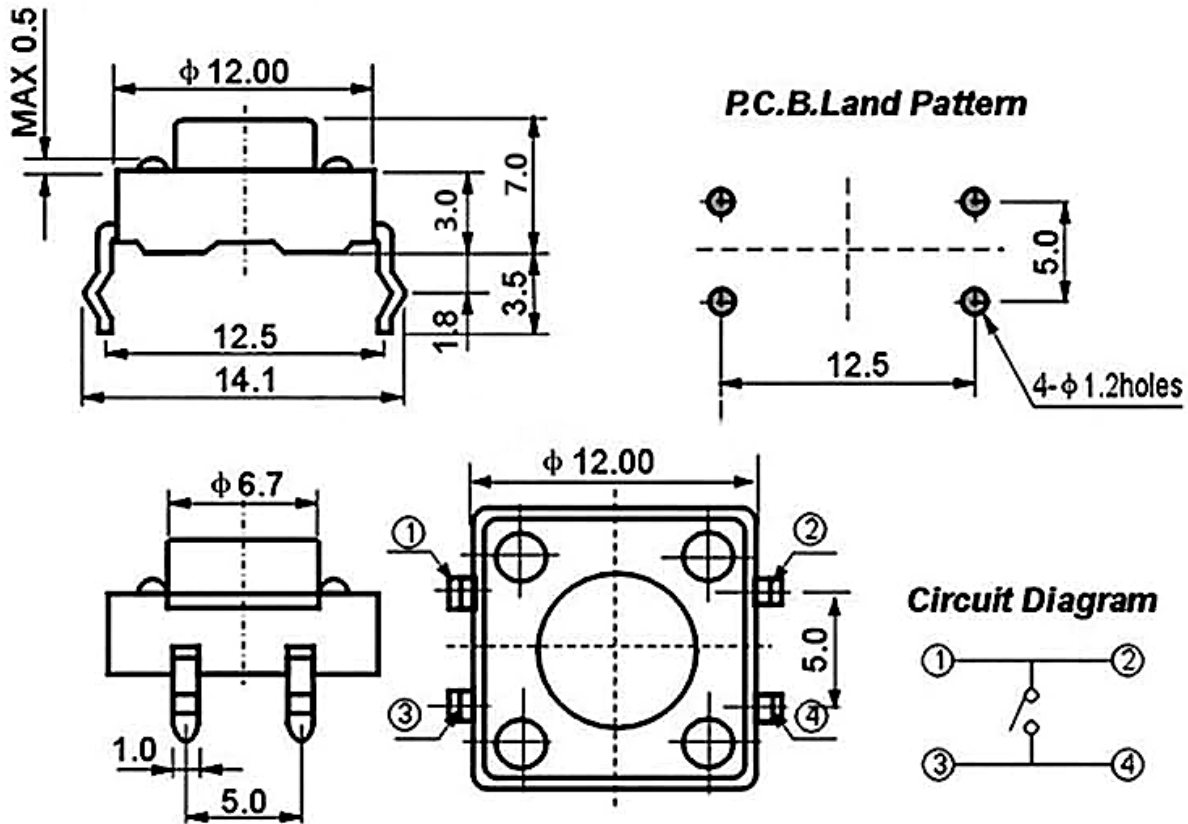


Knöpfe sind eine gängige Komponente zur Steuerung von elektronischen Geräten. Sie werden üblicherweise als Schalter verwendet, um Schaltkreise zu verbinden oder zu unterbrechen. Obwohl es Knöpfe in verschiedenen Größen und Formen gibt, handelt es sich bei dem hier verwendeten um einen 6mm Mini-Knopf, 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)
- 2.6 Türklingel (Scratch-Projekt)
- 2.16 SPIEL - Apfel essen (Scratch-Projekt)
- 2.19 SPIEL - Angeln (Scratch-Projekt)

1.18 Reedschalter



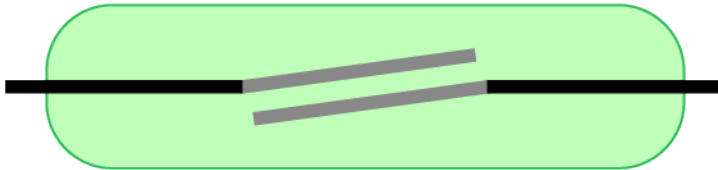
Der Reedschalter ist ein elektrischer Schalter, der mittels eines angelegten Magnetfeldes funktioniert. Er wurde 1936 von Walter B. Ellwood von den Bell Telephone Laboratories erfunden und am 27. Juni 1940 in den USA unter der Patentnummer 2264746 patentiert.

Das Funktionsprinzip eines Reedswitchers 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 Reedswitcher werden so hergestellt, dass sie im Inneren ein Vakuum aufweisen, um ihre Hochspannungsleistung zu verbessern.

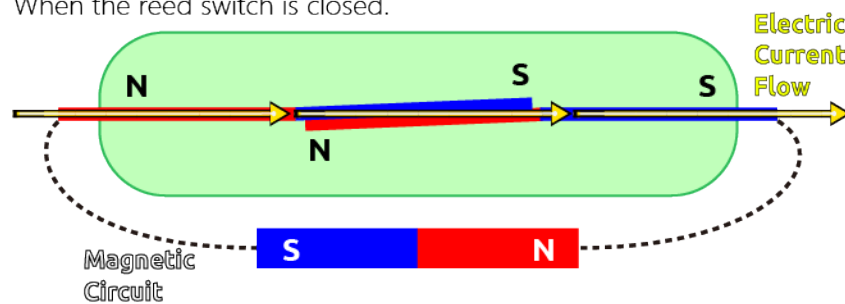
Das Reed fungiert als Magnetflussleiter. Die beiden Reeds sind im Ruhezustand nicht in Kontakt; wenn sie durch ein von einem Permanentmagneten oder einer elektromagnetischen Spule erzeugtes Magnetfeld hindurchgehen, führt das angelegte Magnetfeld dazu, dass die beiden Reeds an ihren Endpunkten unterschiedliche 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.

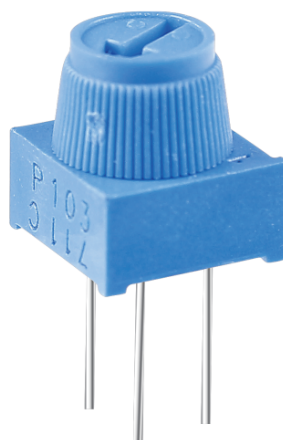


- [Reedschalter – Wikipedia](#)

Beispiel

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

1.19 Potentiometer

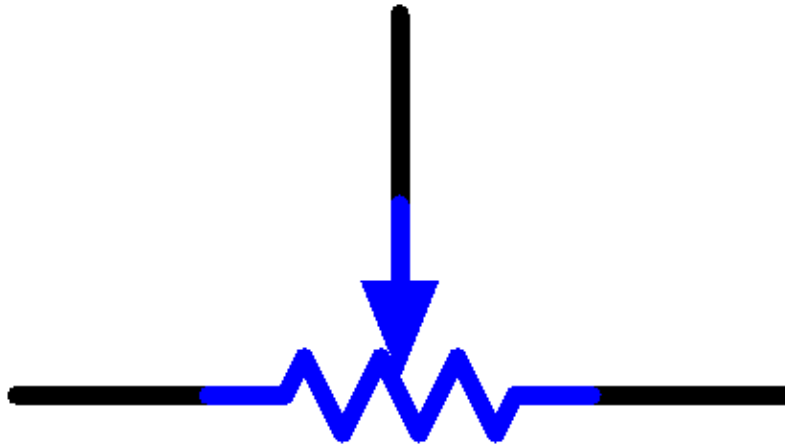


Ein Potentiometer ist ebenfalls ein Widerstandsbauelement mit drei Anschlüssen, dessen Widerstandswert nach einer bestimmten Regelung variiert werden kann.

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:

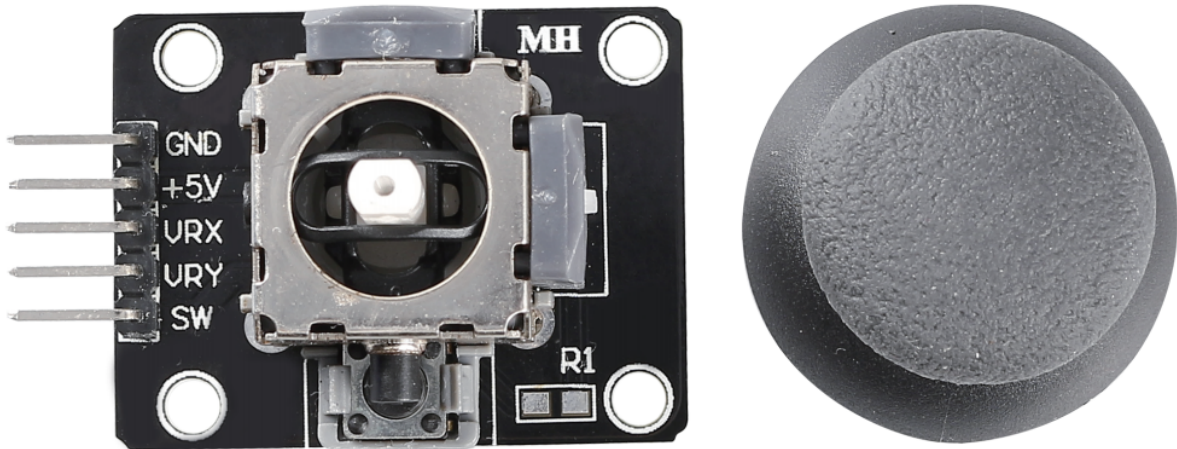
Wenn das Potentiometer als Stromregler fungiert, muss der Schiebekontaktanschluss als einer der Ausgangsanschlüsse verbunden werden.

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

Beispiel

- [4.1 Den Drehknopf drehen](#) (Grundlegendes Projekt)
- [2.5 Bewegende Maus](#) (Scratch-Projekt)
- [2.18 SPIEL - Breakout-Klon](#) (Scratch-Projekt)

1.20 Joystick Modul

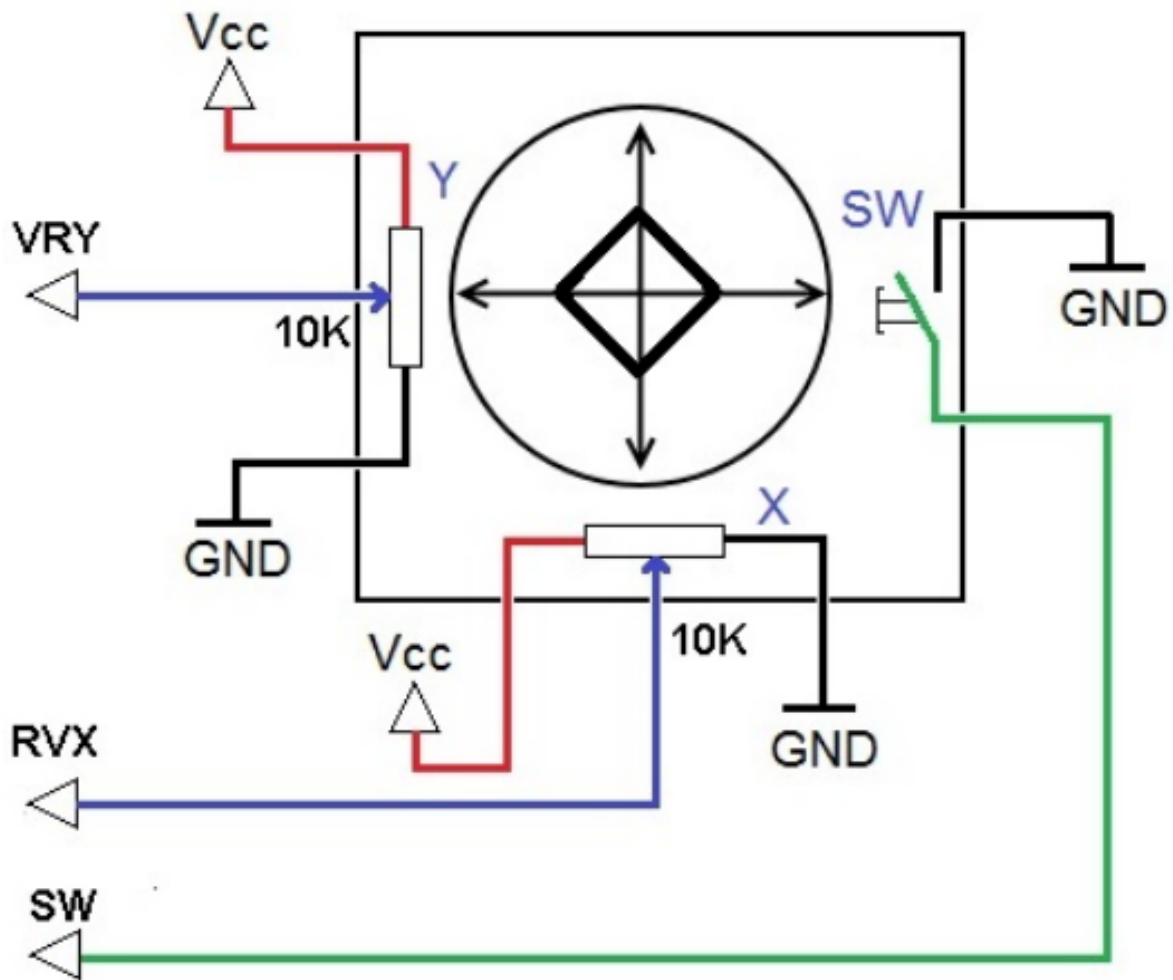


Das Grundkonzept eines Joysticks besteht darin, die Bewegung eines Sticks in elektronische Informationen umzusetzen, die ein Computer verarbeiten kann.

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.

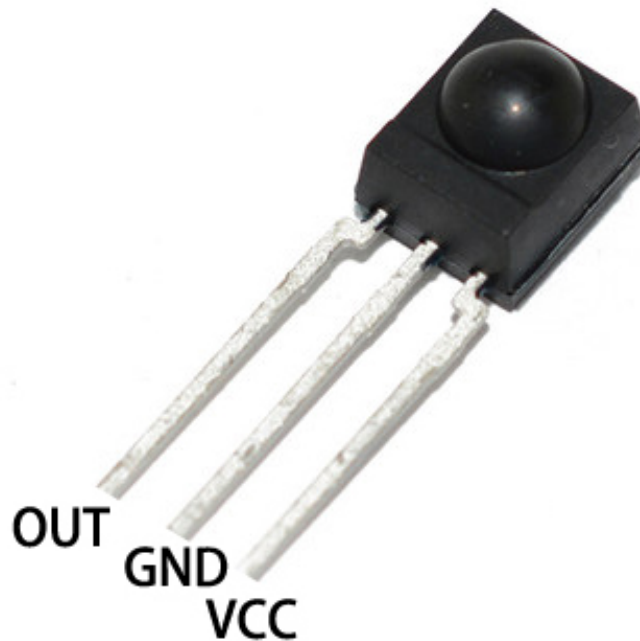


Beispiel

- 4.3 Den Joystick betätigen (Grundlegendes Projekt)
- 2.15 SPIEL - Sternengekreuzt (Scratch-Projekt)
- 2.22 SPIEL - Drachen töten (Scratch-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.

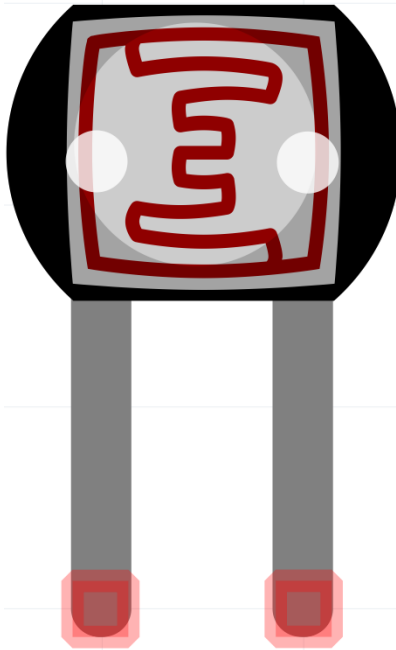
- Größe: 85x39x6mm
- Reichweite der Fernbedienung: 8-10m
- Batterie: 3V Knopfzellen-Lithium-Mangan-Batterie
- Infrarot-Trägerfrequenz: 38KHz
- Oberflächenklebematerial: 0.125mm PET
- Effektive Lebensdauer: mehr als 20.000 Betätigungen

Beispiel

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

Sensor

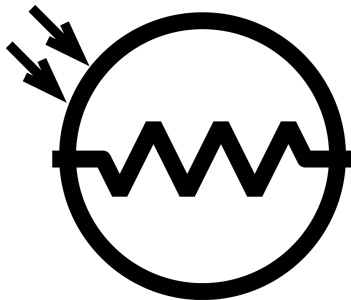
1.22 Fotowiderstand



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.

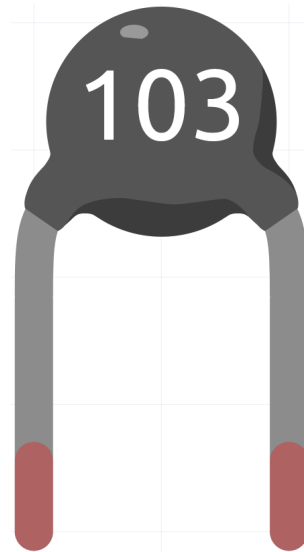


- [Fotowiderstand - Wikipedia](#)

Beispiel

- [4.2 Licht wahrnehmen](#) (Grundlegendes Projekt)
- [5. Überwachung des häuslichen Umfelds](#) (IoT-Projekt)
- [6. Pflanzenüberwachung](#) (IoT-Projekt)

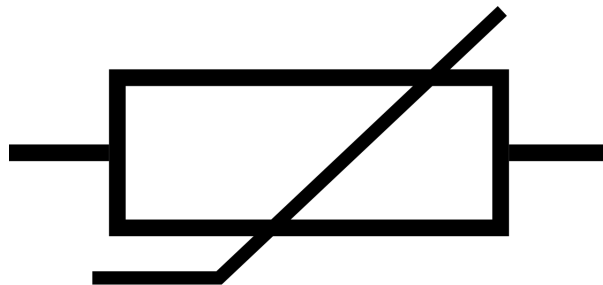
1.23 Thermistor



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

- [2.7 Niedrigtemperaturalarm](#) (Scratch-Projekt)
- [6.3 Hochtemperaturalarm](#) (Grundlegendes Projekt)
- [4.5 Thermometer](#) (Grundlegendes Projekt)

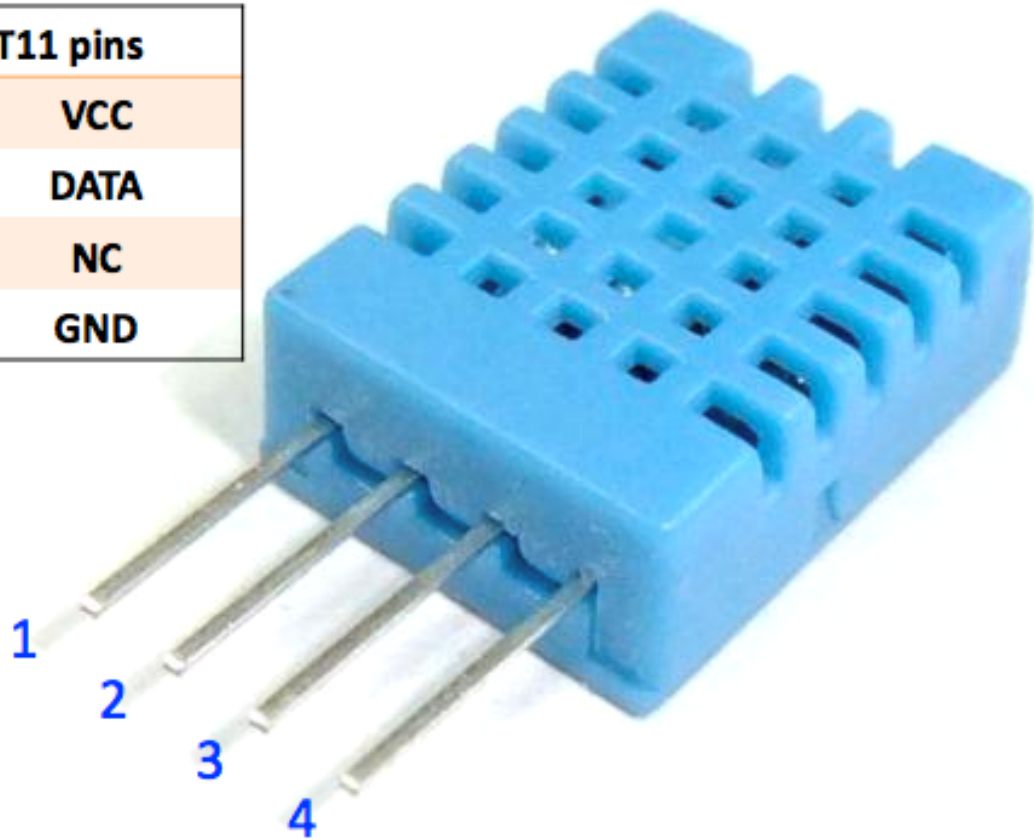
1.24 DHT11 Feuchtigkeits- und Temperatursensor

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

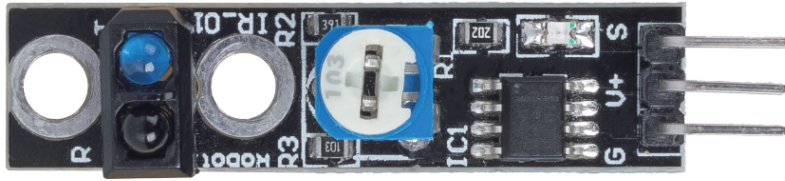
1. Messbereich für Feuchtigkeit: 20 - 90%RH
2. Temperaturmessbereich: 0 - 60°C
3. Ausgabe digitaler Signale für Temperatur und Feuchtigkeit
4. Betriebsspannung: DC 5V; PCB-Größe: 2,0 x 2,0 cm
5. Messgenauigkeit für Feuchtigkeit: $\pm 5\%$ RH
6. Messgenauigkeit für Temperatur: $\pm 2^\circ\text{C}$

- [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 Linie erkennen* (Grundlegendes Projekt)
- *4. Die Linie verfolgen* (Auto-Projekt)
- *2.21 SPIEL - Schütze dein Herz* (Scratch-Projekt)

1.26 Bodenfeuchtigkeitsmodul

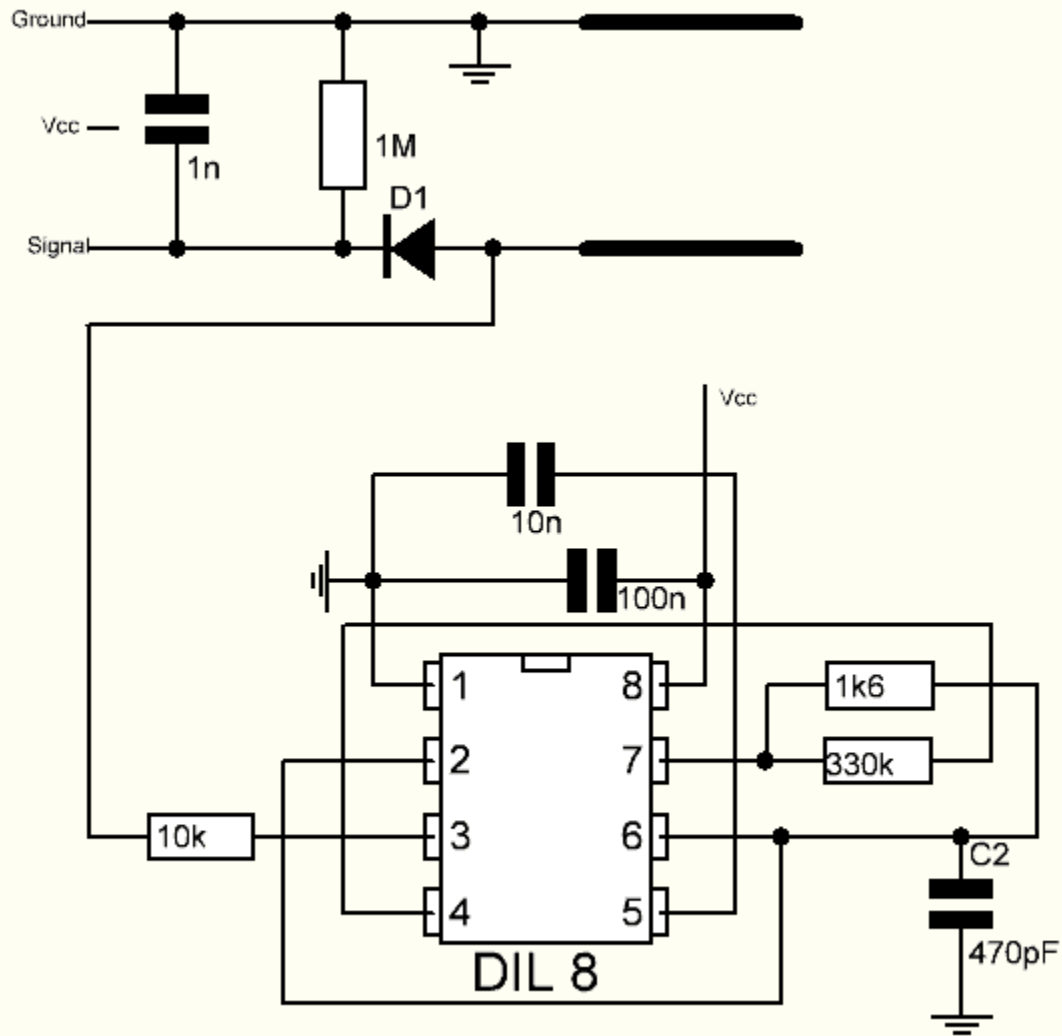


- GND: Masse
- VCCStromversorgung, 3.3v~5V
- AUOT: Gibt den Bodenfeuchtigkeitswert aus, je feuchter der Boden, desto geringer ist sein Wert.

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.

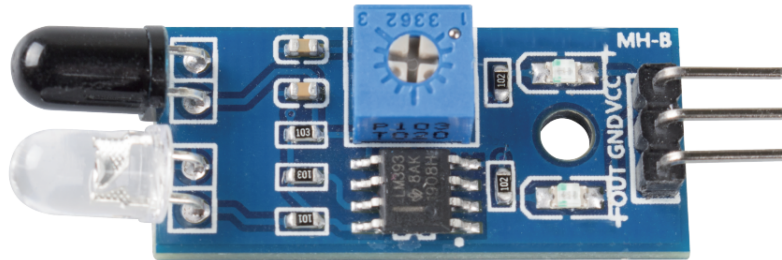
Spezifikation

- 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

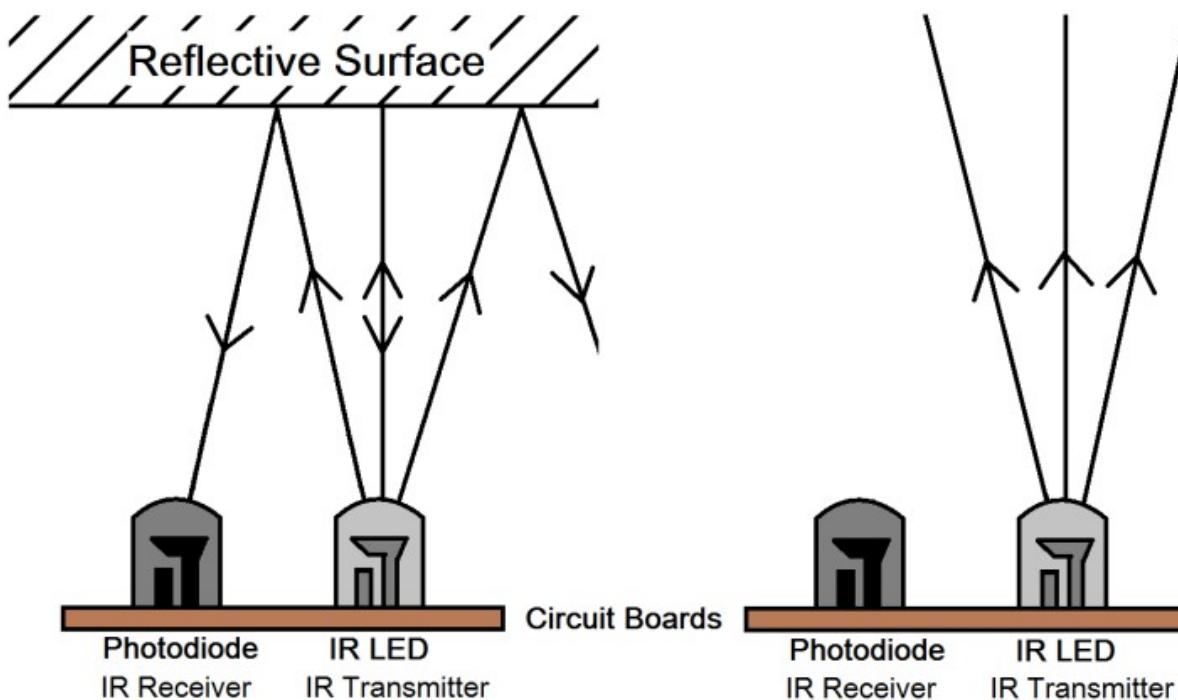


- **VCC:** Stromversorgung, 3,3 ~ 5V DC.
- **GND:** Masse
- **OUT:** Signalausgang, normalerweise im hohen Zustand, schaltet auf niedrig, wenn ein Hindernis erkannt wird.

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

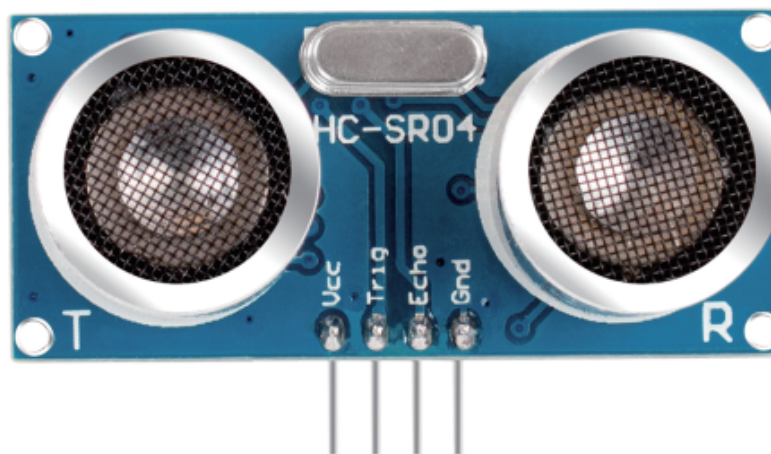
Die Sendetube gibt eine Infrarotfrequenz ab. Trifft die Erkennungsrichtung auf ein Hindernis, wird die Infrarotstrahlung von der Empfangsröhre aufgenommen. Nach der Verarbeitung im Komparator-Schaltkreis leuchtet die Anzeige auf und gibt ein niedriges Signal aus.

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



Beispiel

- *3.3 Hindernisse erkennen* (Grundlegendes Projekt)
- *5. Mit dem Hindernisvermeidungs-Modul spielen* (Auto-Projekt)
- *8. Autonomes Fahren* (Auto-Projekt)
- *7. Strombegrenzendes Tor* (IoT-Projekt)
- *2.13 SPIEL - Schießen* (Scratch-Projekt)
- *2.20 SPIEL - Klicke nicht auf das weiße Feld* (Scratch-Projekt)

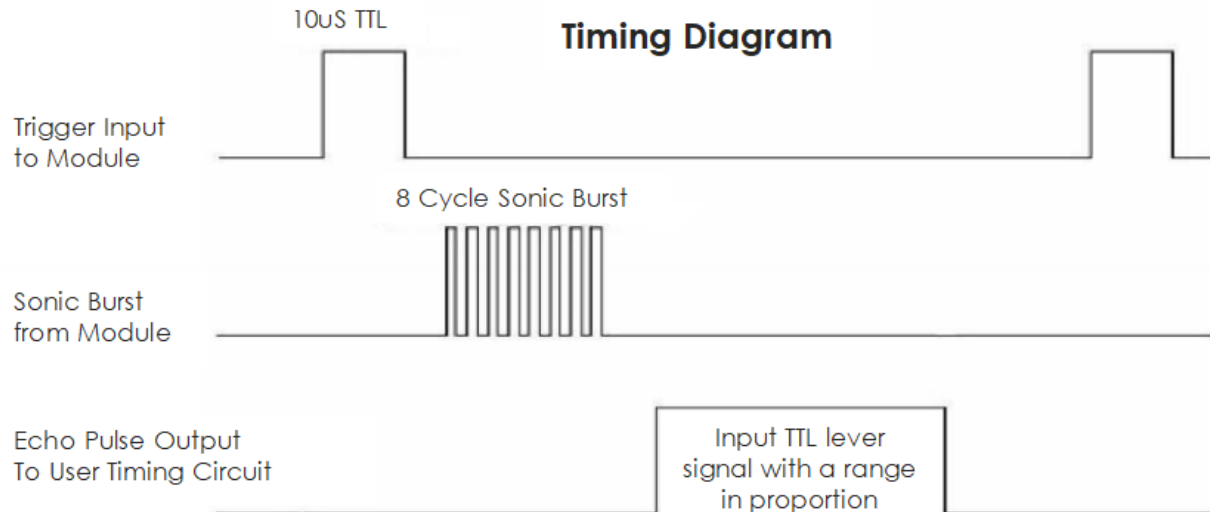
1.28 Ultraschall-Modul

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: $\text{us} / 58 = \text{Zentimeter}$ oder $\text{us} / 148 = \text{Inch}$; 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. Spielen mit dem Ultraschallmodul* (Auto-Projekt)
- *2.17 SPIEL - Flappy Papagei* (Scratch-Projekt)

Einstieg in Arduino

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

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

2.1 Was ist Arduino?

Zunächst gebe ich Ihnen eine kurze Einführung in Arduino.

Arduino ist eine bequeme, flexible und benutzerfreundliche Open-Source-Plattform für elektronische Prototypen, einschließlich verschiedener Hardware-Arduino-Boards und der Software Arduino IDE. Es eignet sich nicht nur für Ingenieure für schnelle Prototypen, sondern auch für Künstler, Designer, Hobbyisten und ist fast ein unverzichtbares Werkzeug für moderne Maker.

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

2.2 Was kann Arduino tun?

Sie fragen sich vielleicht, was Arduino eigentlich kann. Genug 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 und eignen sich für Schüler, Maker und sogar professionelle Programmierer.

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

2.3 Wie erstellt man ein Arduino-Projekt

Folgen Sie diesen Schritten, um zu lernen, wie Sie Arduino von Grund auf verwenden!

2.3.1 Download und Installation von Arduino IDE 2.0

Die Arduino IDE, bekannt als Arduino Integrierte Entwicklungsumgebung, bietet alle erforderliche 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 diese auf das Arduino-Board hochzuladen.

Arduino IDE 2.0 ist ein Open-Source-Projekt. Es stellt einen großen Fortschritt gegenüber seinem robusten Vorgänger, Arduino IDE 1.x, dar und bietet eine überarbeitete Benutzeroberfläche, verbesserten Board- & Bibliotheksmanager, Debugger, Autocomplete-Funktion und vieles mehr.

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

Voraussetzungen

- 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](#).

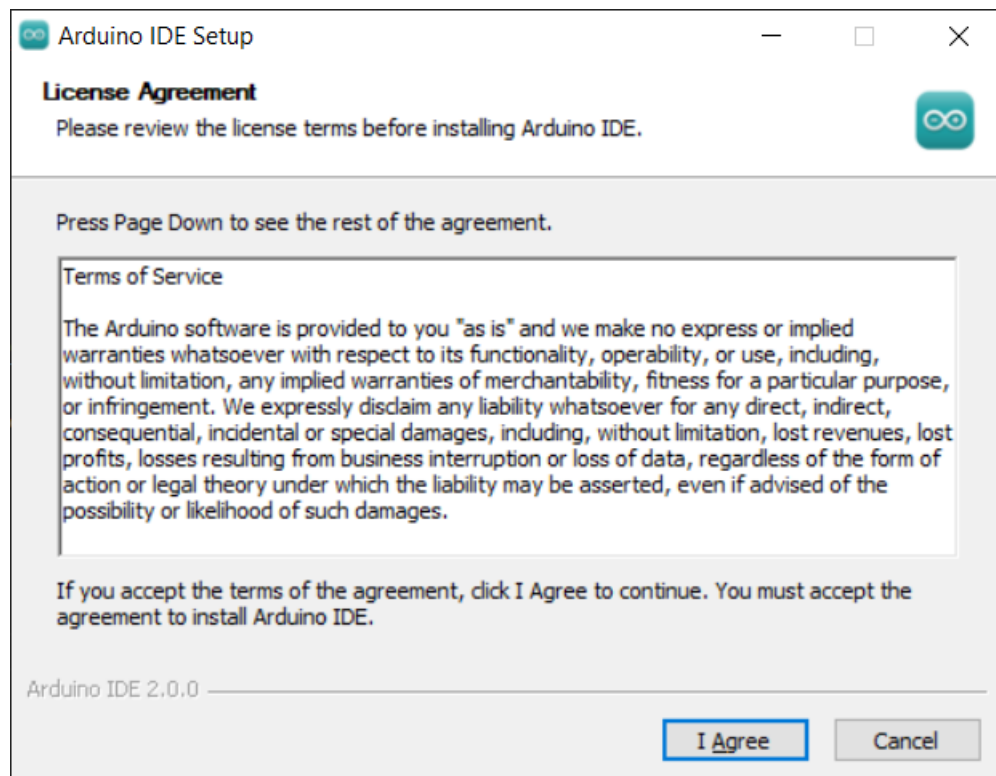
DOWNLOAD OPTIONS

- 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

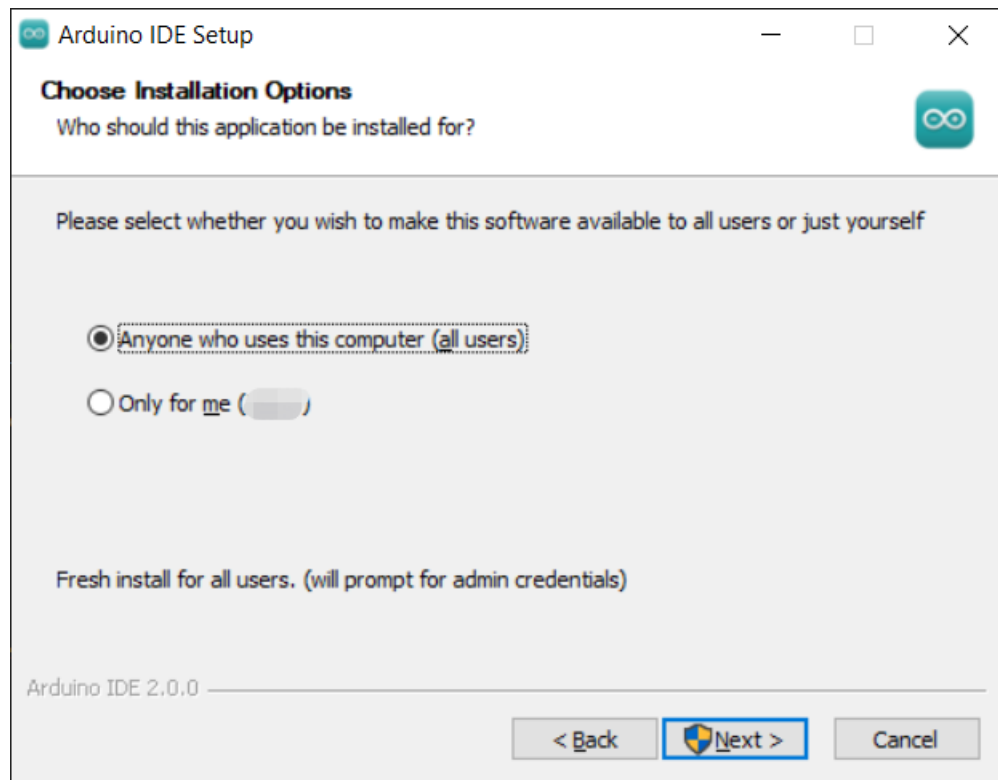
Installation

Windows

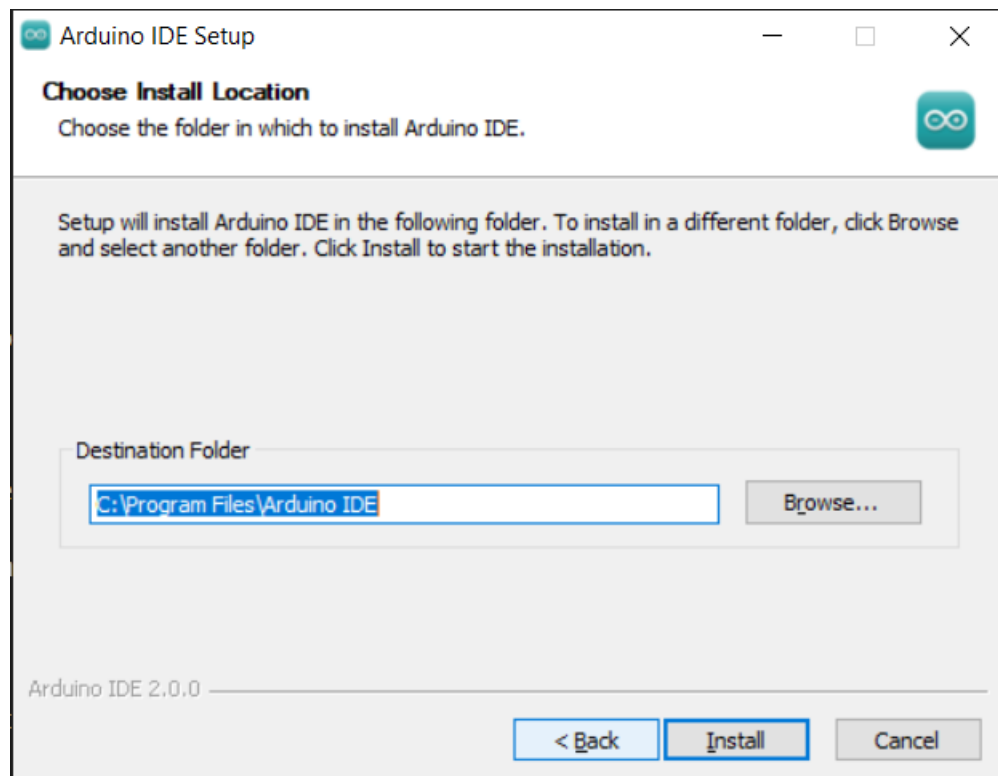
1. Doppelklicken Sie auf die Datei `arduino-ide_xxxx.exe`, um die heruntergeladene Datei auszuführen.
2. Lesen Sie die Lizenzvereinbarung und stimmen Sie dieser zu.



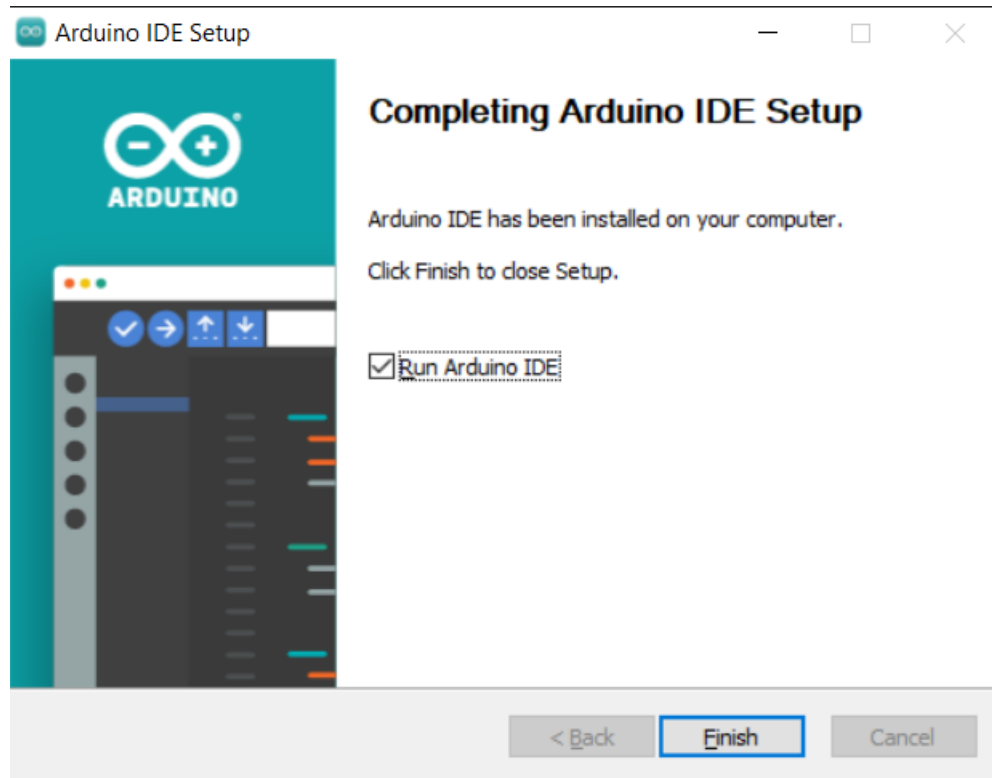
3. Wählen Sie die Installationsoptionen.



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

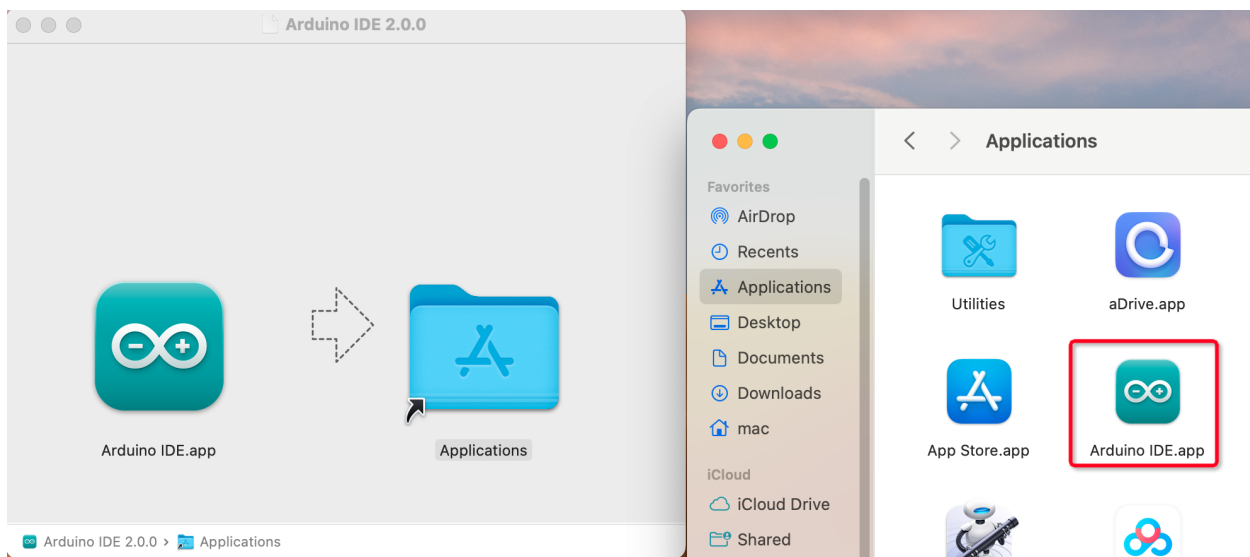


5. Abschließen.



macOS

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

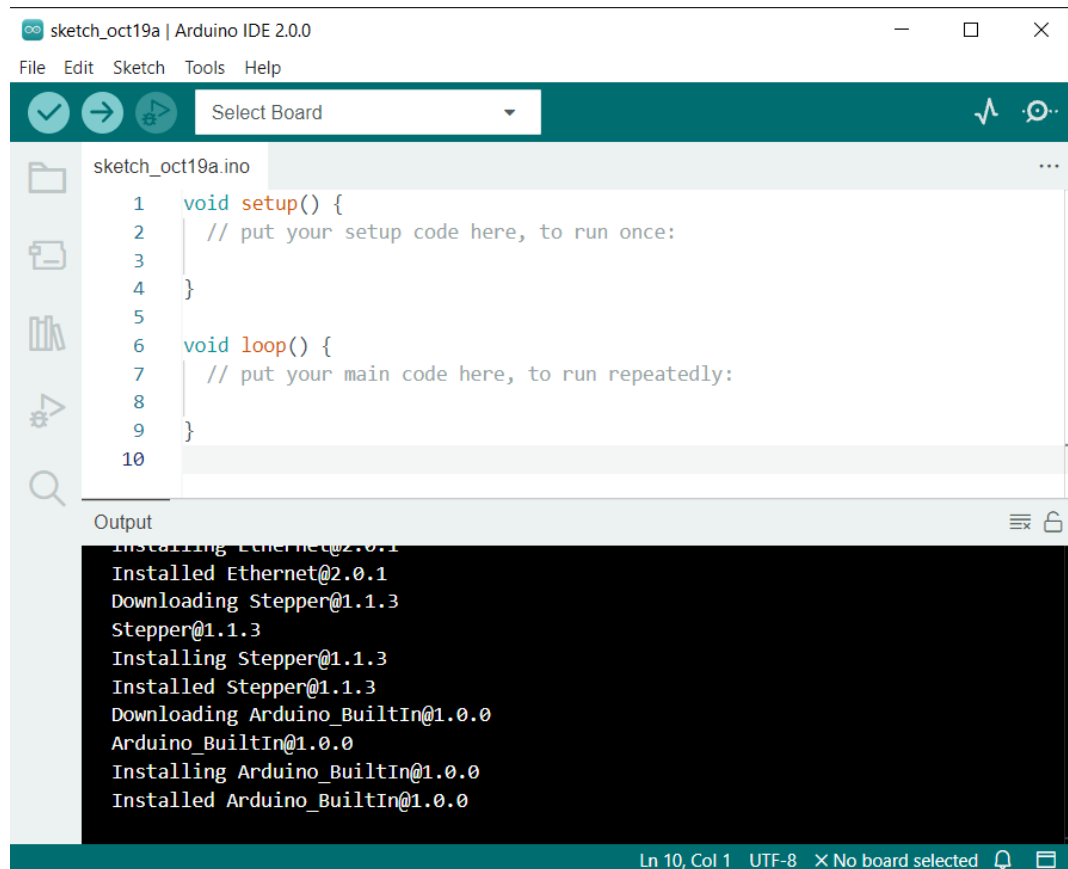


Linux

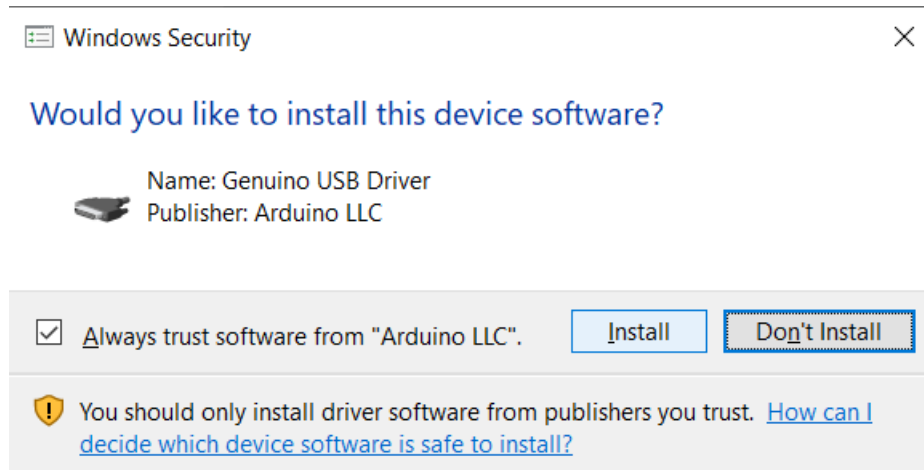
Für das Tutorial zur Installation der Arduino IDE 2.0 auf einem Linux-System, verweisen Sie bitte auf: <https://docs.arduino.cc/software/ide-v2/tutorials/getting-started/ide-v2-downloading-and-installing#linux>

Öffnen der IDE

1. Beim ersten Öffnen der Arduino IDE 2.0 werden automatisch die Arduino AVR Boards, eingebaute Bibliotheken und andere erforderliche Dateien installiert.



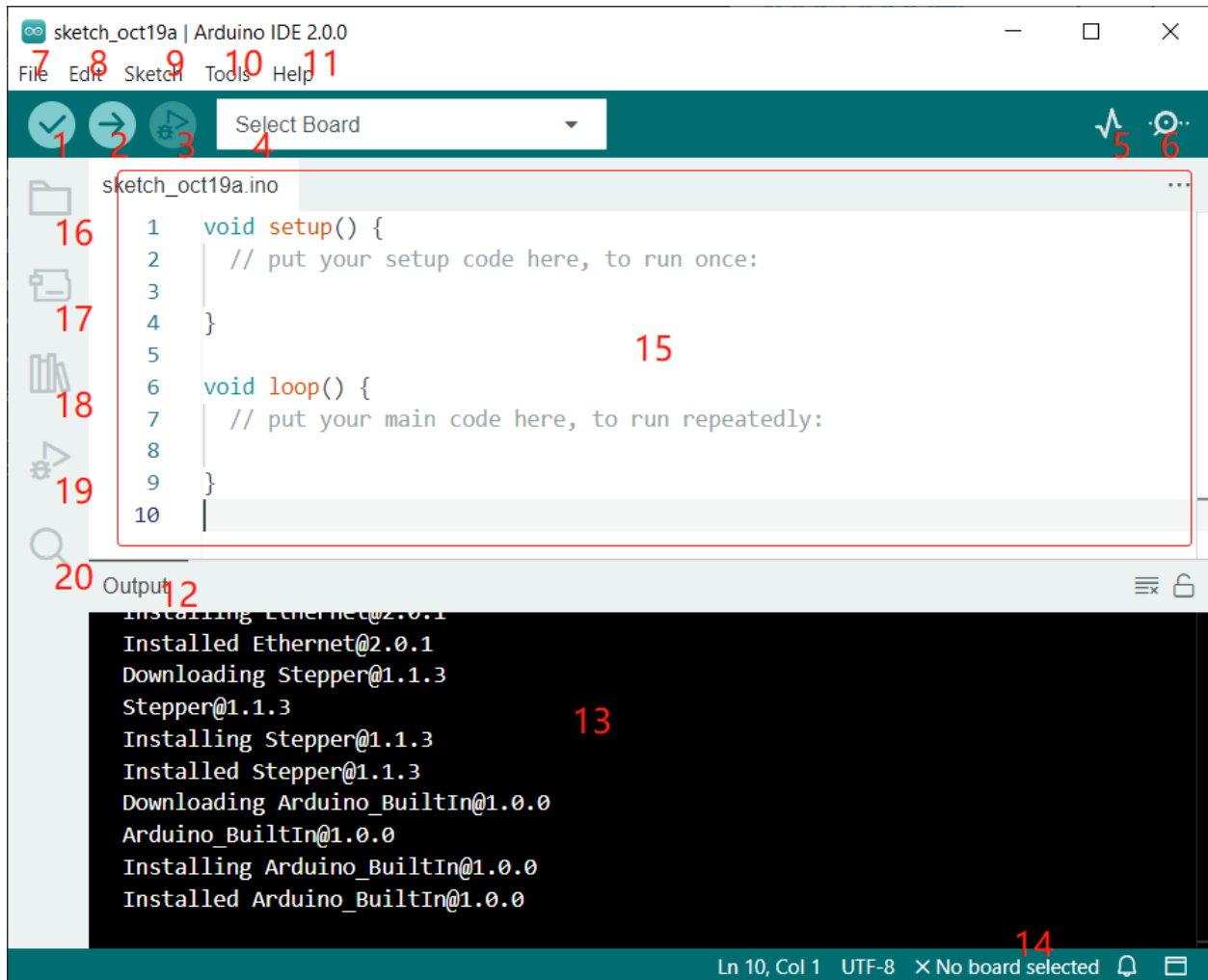
2. Zusätzlich können Ihre Firewall oder Ihr Sicherheitscenter einige Male fragen, ob Sie einen Gerätetreiber installieren möchten. Bitte installieren Sie alle.



3. Jetzt ist Ihre Arduino IDE einsatzbereit!

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

2.3.2 Einführung in die Arduino IDE



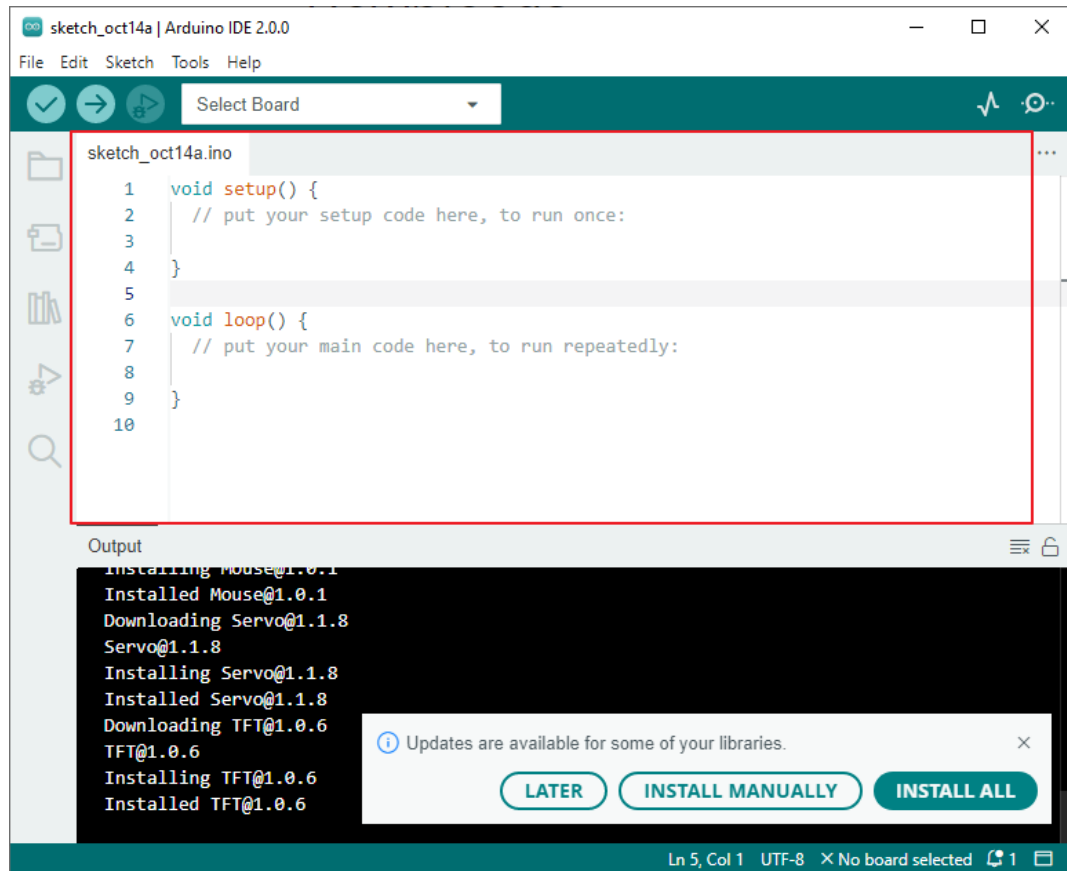
1. **Überprüfen (Verify):** Kompilieren Sie Ihren Code. Bei jeglichen Syntaxproblemen werden Fehlermeldungen angezeigt.
2. **Hochladen (Upload):** Laden Sie den Code auf Ihr Board. Wenn Sie den Button klicken, blinken die RX- und TX-LEDs auf dem Board schnell und hören erst auf, wenn das Hochladen abgeschlossen ist.
3. **Debuggen (Debug):** Für die Fehlerüberprüfung Zeile für Zeile.
4. **Board auswählen (Select Board):** Schnelleinstellung für Board und Port.
5. **Serieller Plotter (Serial Plotter):** Überprüfen Sie die Änderung des Auslesewerts.
6. **Serieller Monitor (Serial Monitor):** Klicken Sie auf den Button und ein Fenster erscheint. Es empfängt die Daten, die von Ihrem Steuerungsboard gesendet werden. Sehr nützlich für das Debuggen.
7. **Datei (File):** Klicken Sie auf das Menü und eine Dropdown-Liste erscheint, einschließlich Dateierstellung, Öffnen, Speichern, Schließen, Konfiguration einiger Parameter usw.
8. **Bearbeiten (Edit):** Klicken Sie auf das Menü. In der Dropdown-Liste gibt es Bearbeitungsoperationen wie **Ausschneiden (Cut)**, **Kopieren (Copy)**, **Einfügen (Paste)**, **Suchen (Find)** usw. mit den entsprechenden Tastenkombinationen.
9. **Skizze (Sketch):** Beinhaltet Operationen wie **Überprüfen (Verify)**, **Hochladen (Upload)**, **Dateien hinzufügen**

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

10. **Werkzeug (Tool)**: Beinhaltet einige Werkzeuge – am häufigsten verwendet werden das Board (das von Ihnen verwendete Board) und der 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 Bedienung in der IDE, Einführungsinformationen, Fehlerbehebung, Codeerklärung usw.
12. **Ausgabeleiste (Output Bar)**: Wechseln Sie hier den Ausgabe-Tab.
13. **Ausgabefenster (Output Window)**: Informationen ausgeben.
14. **Board und Port**: Hier können Sie das ausgewählte Board und den Port für das Hochladen des Codes vorab ansehen. Sie können sie erneut über **Werkzeug (Tools)** -> **Board / Port** auswählen, falls etwas nicht korrekt ist.
15. Der Bearbeitungsbereich der IDE. Hier können Sie Code schreiben.
16. **Skizzenbuch (Sketchbook)**: Zum Verwalten von Skizzendateien.
17. **Board-Manager (Board Manager)**: Zum Verwalten von Board-Treibern.
18. **Library Manager (Library Manager)**: Zum Verwalten Ihrer Bibliotheksdateien.
19. **Debuggen (Debug)**: Hilft beim Debuggen des Codes.
20. **Suchen (Search)**: Suchen Sie die Codes aus Ihren Skizzen.

2.3.3 Wie erstellt, öffnet oder speichert man die Skizze?

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



Diese Skizzendateien haben einen regelmäßigen temporären Namen, aus dem Sie das Erstellungsdatum der Datei ableiten können. `sketch_oct14a.ino` bedeutet die erste Skizze vom 14. Oktober, `.ino` ist das Dateiformat dieser Skizze.

2. Jetzt versuchen wir, eine neue Skizze zu erstellen. Kopieren Sie den folgenden Code in die Arduino IDE, um den ursprünglichen Code zu ersetzen.



```
void setup() {
  // Geben Sie hier Ihren Setup-Code ein, der einmal ausgeführt wird:
  pinMode(13,OUTPUT);
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

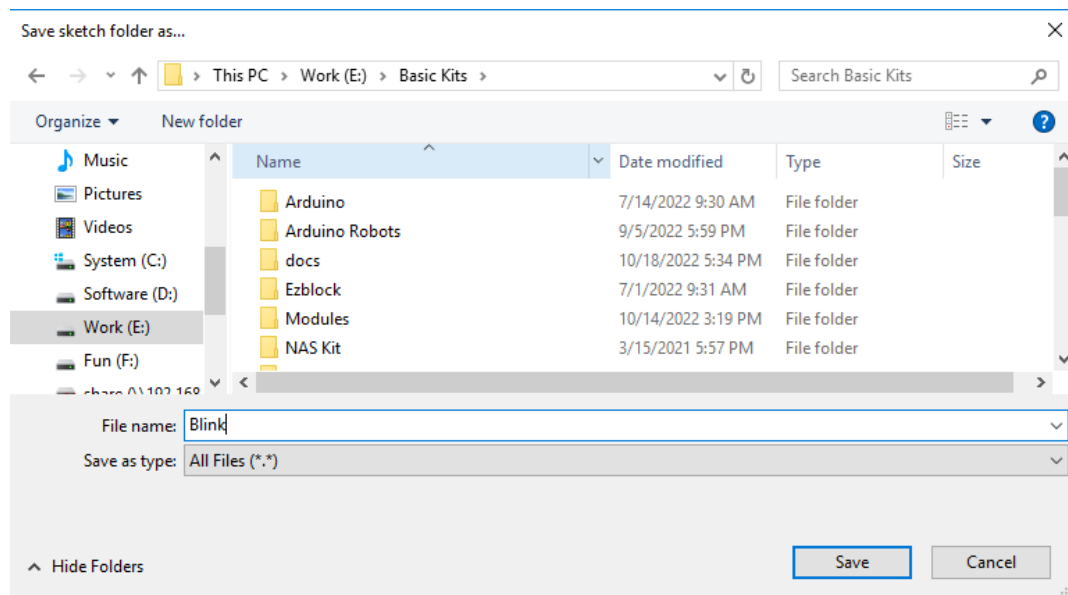
```

}

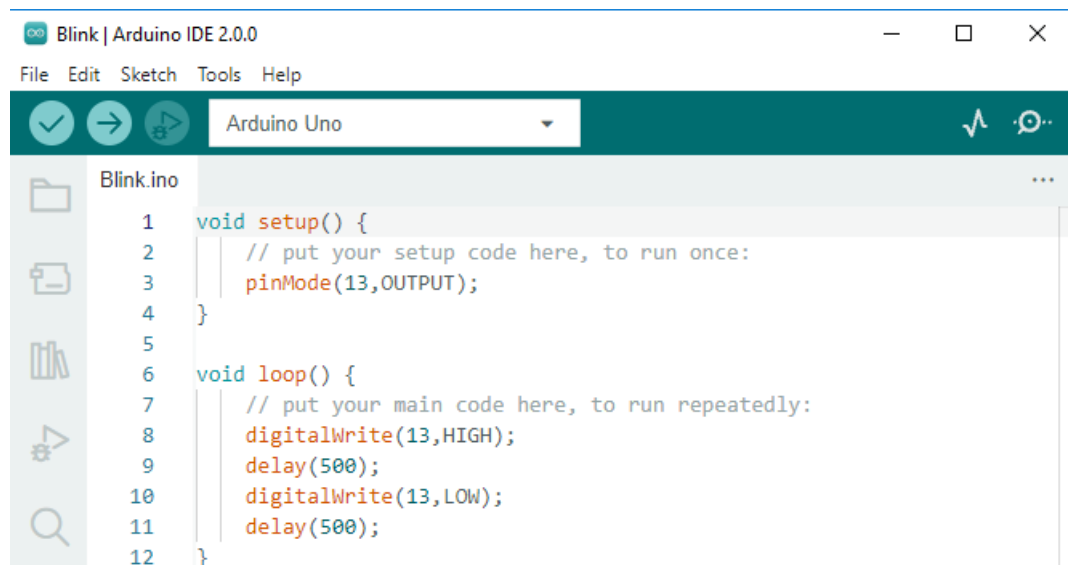
void loop() {
  // Geben Sie hier Ihren Hauptcode ein, der wiederholt ausgeführt wird:
  digitalWrite(13,HIGH);
  delay(500);
  digitalWrite(13,LOW);
  delay(500);
}

```

3. Drücken Sie Ctrl+S oder klicken Sie auf **Datei (File) -> Speichern (Save)**. Die Skizze wird standardmäßig unter C:\Users\{Ihr_Benutzername}\Dokumente\Arduino gespeichert. Sie können sie umbenennen oder einen neuen Pfad zum Speichern auswählen.



4. Nachdem Sie erfolgreich gespeichert haben, werden Sie feststellen, dass der Name in der Arduino IDE aktualisiert wurde.



Bitte fahren Sie mit dem nächsten Abschnitt fort, um zu lernen, wie Sie diese erstellte Skizze auf Ihr Arduino-Board hochladen.

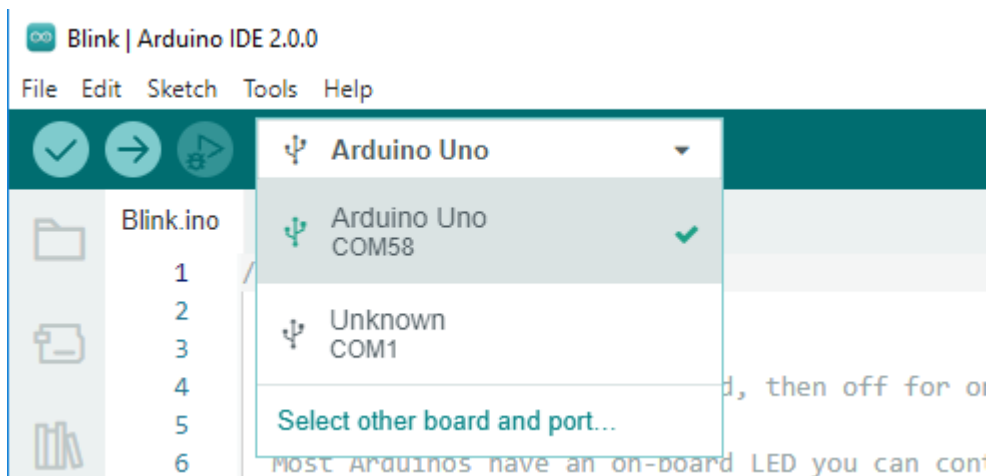
2.3.4 Wie lädt man die Skizze auf das Board hoch?

In diesem Abschnitt erfahren Sie, wie Sie die zuvor erstellte Skizze auf das Arduino-Board hochladen und einige Überlegungen dazu kennen lernen.

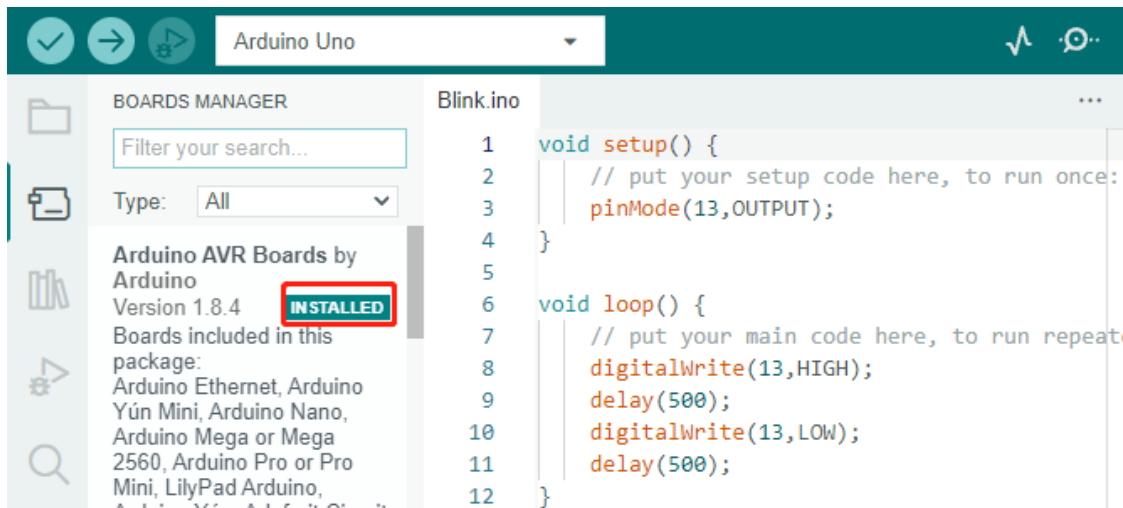
1. Board und Port auswählen

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



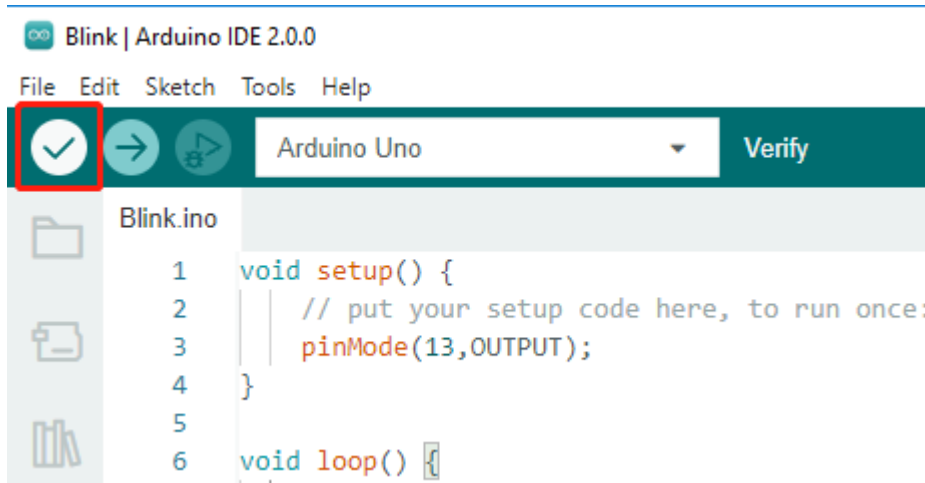
Wenn Ihr Board bereits angeschlossen ist, aber nicht erkannt wird, überprüfen Sie, ob das **INSTALLED**-Logo im Abschnitt **Arduino AVR Boards** des **Boards-Manager** erscheint. Wenn nicht, scrollen Sie bitte etwas nach unten und klicken auf **INSTALL**.



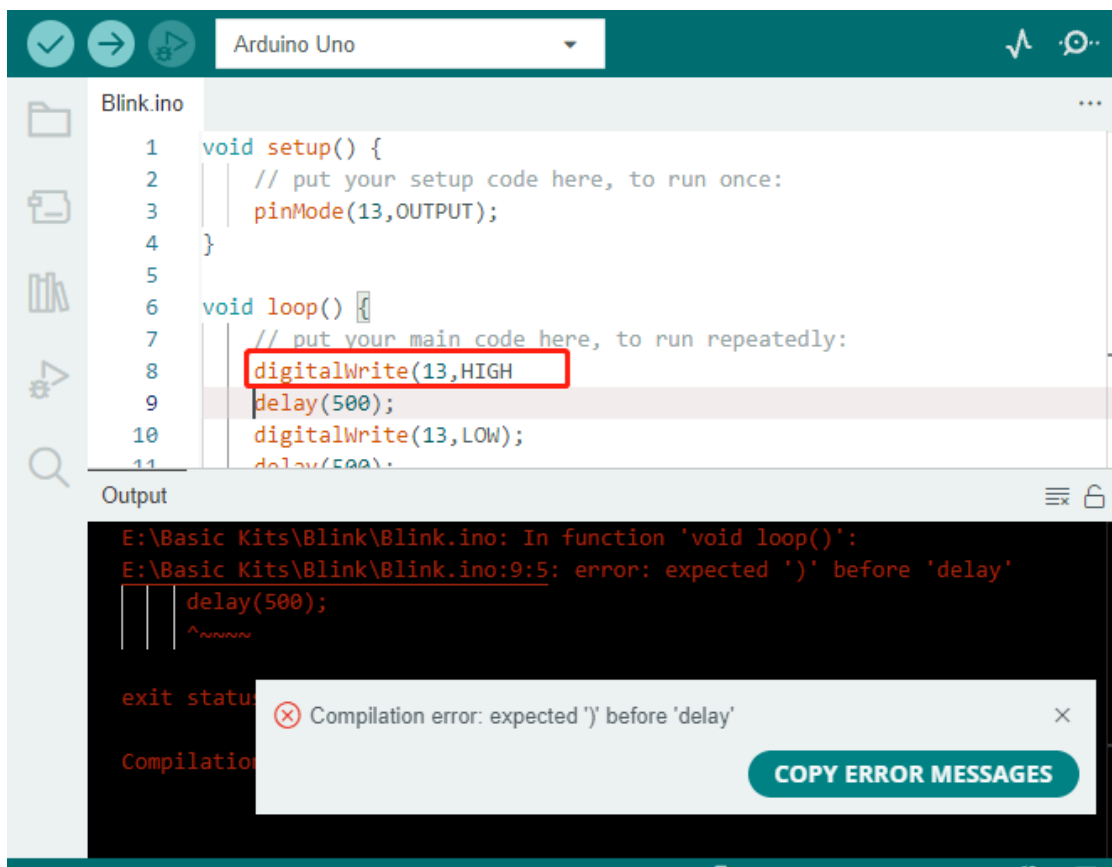
Ein erneutes Öffnen der Arduino IDE und das erneute Anstecken des Arduino-Boards beheben die meisten Probleme. Sie können auch auf **Tools** -> **Board** oder **Port** klicken, um sie auszuwählen.

2. Überprüfen der Skizze

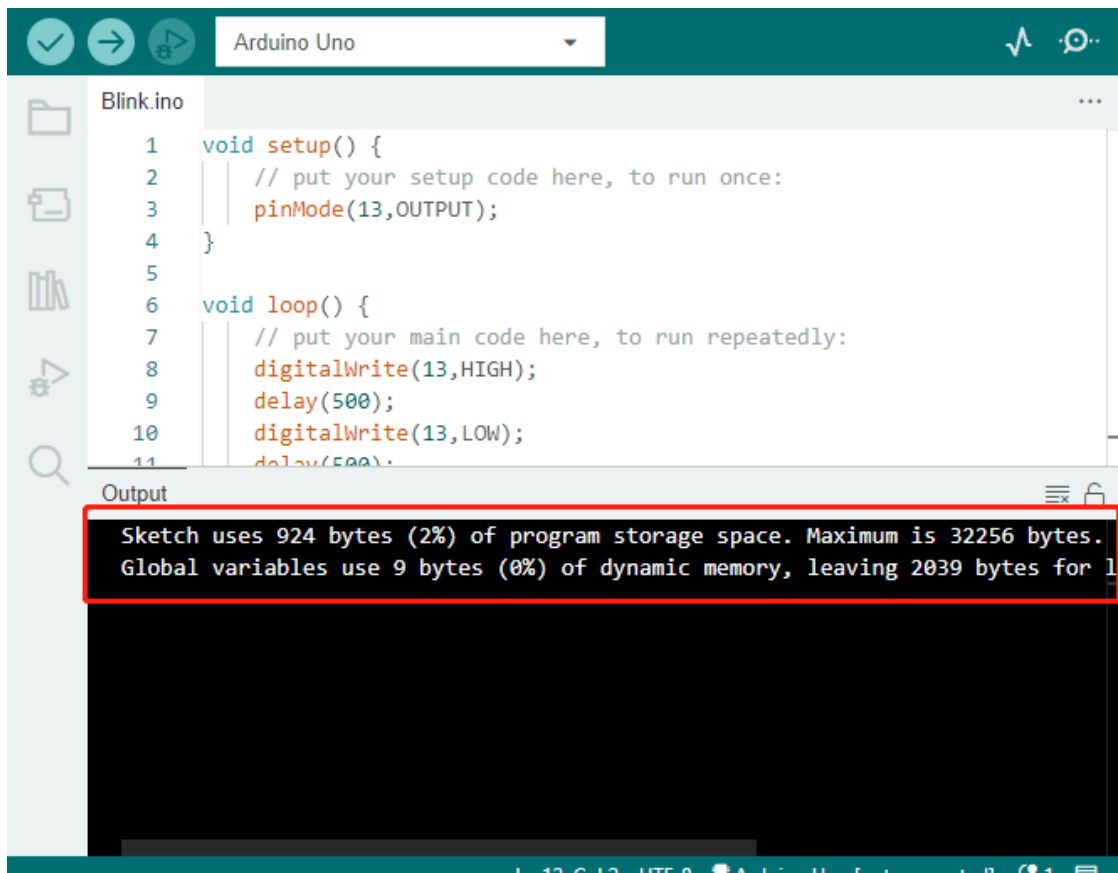
Nachdem Sie auf die Überprüfen-Schaltfläche geklickt haben, wird die Skizze kompiliert, um festzustellen, ob Fehler vorliegen.



Sie können es verwenden, um Fehler zu finden, wenn Sie einige Zeichen löschen oder versehentlich einige Buchstaben eingeben. In der Nachrichtenleiste können Sie sehen, wo und welche Art von Fehlern aufgetreten sind.

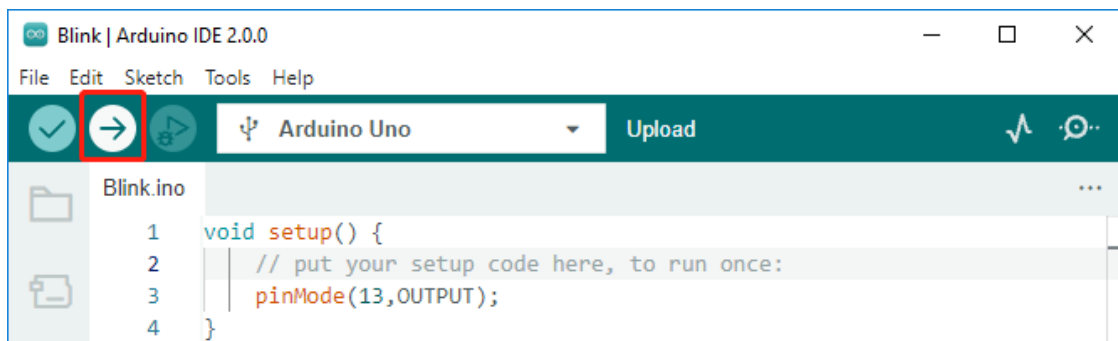


Wenn keine Fehler vorliegen, sehen Sie eine Nachricht wie die unten stehende.

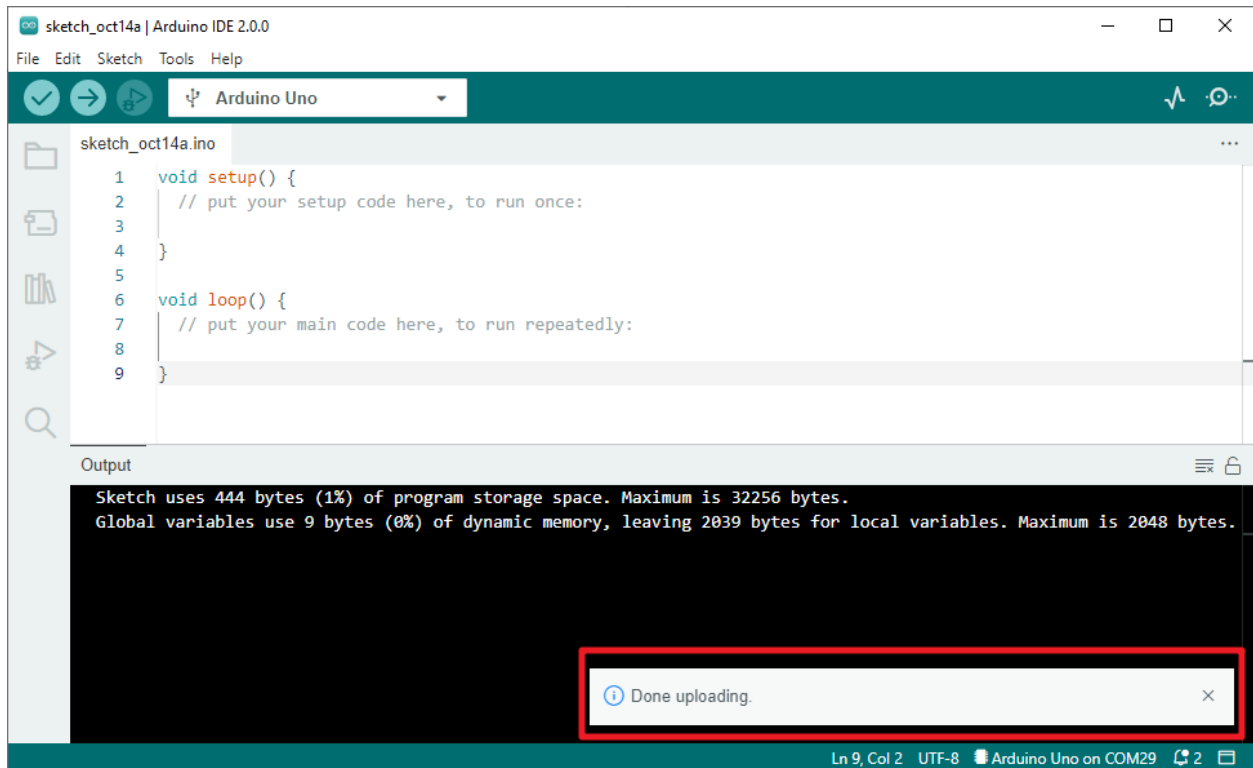


3. Skizze hochladen

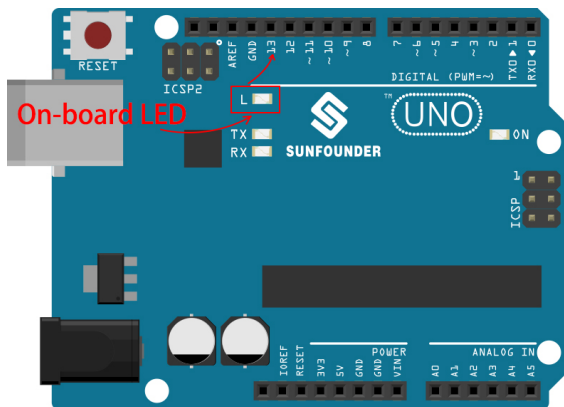
Nach Abschluss der oben genannten Schritte klicken Sie auf die **Upload**-Schaltfläche, um diese Skizze auf das Board hochzuladen.



Wenn der Vorgang erfolgreich war, sehen Sie den folgenden Hinweis.



Gleichzeitig blinkt die LED auf dem Board.



Das Arduino-Board führt die Skizze automatisch aus, sobald nach dem Hochladen der Skizze Strom angelegt wird. Das laufende Programm kann durch Hochladen einer neuen Skizze überschrieben werden.

2.3.5 Struktur eines Arduino-Programms

Werfen wir einen Blick auf die neue Skizze. Obwohl sie nur wenige Codezeilen enthält, handelt es sich tatsächlich um eine „leere“ Skizze. Das Hochladen dieser Skizze auf das Entwicklungsboard wird keine Aktion auslösen.

```

void setup() {
  // Fügen Sie Ihren Setup-Code hier ein, der einmal ausgeführt wird:
}

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
void loop() {  
  // Fügen Sie Ihren Hauptcode hier ein, der wiederholt ausgeführt wird:  
}
```

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

Während des Skizzierens wird `setup()` zuerst ausgeführt und der Code darin (innerhalb von `{}`) wird nach dem Einschalten oder Zurücksetzen des Boards einmal ausgeführt. `loop()` dient zum Schreiben der Hauptfunktion, und der darin enthaltene Code wird in einer Schleife nach der Ausführung von `setup()` wiederholt.

Um `setup()` und `loop()` besser zu verstehen, betrachten wir vier Skizzen. Ihr Ziel ist es, die integrierte LED des Arduino blinken zu lassen. Bitte führen Sie jedes Experiment nacheinander aus und notieren Sie die spezifischen Effekte.

- Skizze 1: Lassen Sie die integrierte LED kontinuierlich blinken.

```
void setup() {  
  // Fügen Sie Ihren Setup-Code hier ein, der einmal ausgeführt wird:  
  pinMode(13,OUTPUT);  
}  
  
void loop() {  
  // Fügen Sie Ihren Hauptcode hier ein, der wiederholt ausgeführt wird:  
  digitalWrite(13,HIGH);  
  delay(500);  
  digitalWrite(13,LOW);  
  delay(500);  
}
```

- Skizze 2: Lassen Sie die integrierte LED nur einmal blinken.

```
void setup() {  
  // Fügen Sie Ihren Setup-Code hier ein, der einmal ausgeführt wird:  
  pinMode(13,OUTPUT);  
  digitalWrite(13,HIGH);  
  delay(500);  
  digitalWrite(13,LOW);  
  delay(500);  
}  
  
void loop() {  
  // Fügen Sie Ihren Hauptcode hier ein, der wiederholt ausgeführt wird:  
}
```

- Skizze 3: Lassen Sie die integrierte LED einmal langsam und dann schnell blinken.

```
void setup() {  
  // Fügen Sie Ihren Setup-Code hier ein, der einmal ausgeführt wird:  
  pinMode(13,OUTPUT);  
  digitalWrite(13,HIGH);  
  delay(1000);  
  digitalWrite(13,LOW);  
  delay(1000);  
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

}

void loop() {
    // Fügen Sie Ihren Hauptcode hier ein, der wiederholt ausgeführt wird:
    digitalWrite(13,HIGH);
    delay(200);
    digitalWrite(13,LOW);
    delay(200);
}

```

- Skizze 4: Meldet einen Fehler.

```

void setup() {
    // Fügen Sie Ihren Setup-Code hier ein, der einmal ausgeführt wird:
    pinMode(13,OUTPUT);
}

digitalWrite(13,HIGH);
delay(1000);
digitalWrite(13,LOW);
delay(1000);

void loop() {
    // Fügen Sie Ihren Hauptcode hier ein, der wiederholt ausgeführt wird:
}

```

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 dann `loop()` ausgeführt.
- Der Code muss innerhalb des `{}`-Bereichs von `setup()` oder `loop()` geschrieben werden, außerhalb des Rahmens tritt ein Fehler auf.

Bemerkung: Anweisungen wie `digitalWrite(13,HIGH)` dienen zur Steuerung der integrierten LED, und ihre Verwendung wird in späteren Kapiteln ausführlich besprochen.

2.3.6 Skizze Schreibregel

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

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

In diesem Kapitel werden die Grundregeln der Arduino-Sprache vorgestellt. Dies wird Ihnen helfen zu verstehen, wie man natürliche Sprache in Code umsetzt.

Natürlich ist dies ein Prozess, mit dem man sich erst mit der Zeit vertraut macht und der für Anfänger am fehleranfälligsten ist. Wenn Sie also oft Fehler machen, ist das in Ordnung. Versuchen Sie es einfach noch ein paar Mal.

Semikolon ;

Genauso wie Sie in einem Brief am Ende eines Satzes einen Punkt setzen, verlangt die Arduino-Sprache, dass Sie ; verwenden, um dem Board das Ende des Befehls mitzuteilen.

Betrachten wir das bekannte Beispiel „Onboard-LED blinkt“. Eine korrekte Skizze sollte so aussehen:

Beispiel:

```
void setup() {  
    // Geben Sie hier Ihren Setup-Code ein, der einmal ausgeführt wird:  
    pinMode(13,OUTPUT);  
}  
  
void loop() {  
    // Geben Sie hier Ihren Hauptcode ein, der wiederholt ausgeführt wird:  
    digitalWrite(13,HIGH);  
    delay(500);  
    digitalWrite(13,LOW);  
    delay(500);  
}
```

Betrachten wir als Nächstes die folgenden beiden Skizzen und raten, ob sie von Arduino korrekt erkannt werden können, bevor wir sie ausführen.

Skizze A:

```
void setup() {  
    // Geben Sie hier Ihren Setup-Code ein, der einmal ausgeführt wird:  
    pinMode(13,OUTPUT);  
}  
  
void loop() {  
    // Geben Sie hier Ihren Hauptcode ein, der wiederholt ausgeführt wird:  
    digitalWrite(13,HIGH)  
    delay(500)  
    digitalWrite(13,LOW)  
    delay(500)  
}
```

Skizze B:

```
void setup() {  
    // Geben Sie hier Ihren Setup-Code ein, der einmal ausgeführt wird:  
    pinMode(13,OUTPUT);  
}  
  
void loop() {  
    // Geben Sie hier Ihren Hauptcode ein, der wiederholt ausgeführt wird:  
    digitalWrite(13,  
HIGH); delay  
    (500  
    );  
    digitalWrite(13,
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
LOW);
    delay(500)
;
}
```

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

- Die Fehler in **Skizze A** fehlen ; und obwohl es normal aussieht, kann es Arduino nicht lesen.
- **Skizze B** sieht zwar für den Menschen merkwürdig aus, jedoch sind Einrückungen, Zeilenumbrüche und Leerzeichen in Aussagen Dinge, die in Arduino-Programmen nicht existieren. Daher sieht es für den Arduino-Compiler genauso so aus wie im Beispiel.

Bitte schreiben Sie Ihren Code jedoch nicht wie **Skizze B**, denn normalerweise sind es Menschen, die den Code schreiben und ansehen, also bringen Sie sich nicht selbst in Schwierigkeiten.

Geschweifte Klammern {}

{ } sind die Hauptkomponenten der Arduino-Programmiersprache und sie müssen paarweise auftreten. Eine bessere Programmierkonvention ist es, eine Struktur, die geschweifte Klammern benötigt, einzufügen, indem man direkt nach dem Tippen der linken geschweiften Klammer die rechte geschweifte Klammer eintippt und dann den Cursor zwischen die geschweiften Klammern bewegt, um die Anweisung einzufügen.

Kommentar //

Ein Kommentar ist der Teil der Skizze, den der Compiler ignoriert. Sie werden normalerweise verwendet, um anderen zu erklären, wie das Programm funktioniert.

Wenn wir in einer Codezeile zwei benachbarte Schrägstriche schreiben, wird der Compiler alles bis zum Ende der Zeile ignorieren.

Erstellen wir eine neue Skizze, wird sie mit zwei Kommentaren geliefert. Wenn wir diese beiden Kommentare entfernen, wird die Skizze in keiner Weise beeinflusst.

```
void setup() {
    // Geben Sie hier Ihren Setup-Code ein, der einmal ausgeführt wird:
}

void loop() {
    // Geben Sie hier Ihren Hauptcode ein, der wiederholt ausgeführt wird:
}
```

Kommentare sind in der Programmierung sehr nützlich und einige häufige Anwendungen sind unten aufgeführt.

- Verwendung A: Erklären Sie sich selbst oder anderen, was dieser Abschnitt des Codes macht.

```
void setup() {
    pinMode(13,OUTPUT); // Pin 13 auf Ausgangsmodus setzen, er steuert die Onboard-LED
}

void loop() {
    digitalWrite(13,HIGH); // Aktivieren der Onboard-LED durch Setzen von Pin 13 auf HIGH
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
delay(500); // Status quo für 500 ms
digitalWrite(13,LOW); // Ausschalten der Onboard-LED
delay(500); // Status quo für 500 ms
}
```

- Verwendung B: Temporäres Deaktivieren einiger Anweisungen (ohne sie zu löschen) und erneutes Kommentieren, wenn sie gebraucht werden, so dass sie nicht neu geschrieben werden müssen. Dies ist sehr nützlich, wenn man Code debuggt und versucht, Programmfehler zu lokalisieren.

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

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 den Kommentar zu entfernen.

Kommentar /**/

Gleich wie // für Kommentare. Diese Art von Kommentar kann mehr als eine Zeile lang sein. Sobald der Compiler /* liest, ignoriert er alles, was danach kommt, bis er auf */ stößt.

Beispiel 1:

```
/* Blinken */

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

void loop() {
  /*
   Der folgende Code lässt die integrierte LED blinken.
   Sie können die Zahl in delay() ändern, um die Blinkfrequenz zu ändern.
  */
  digitalWrite(13,HIGH);
  delay(500);
  digitalWrite(13,LOW);
  delay(500);
}
```

#define

Dies ist ein nützliches C++ Werkzeug.

```
#define Bezeichner Token-Zeichenfolge
```

Beim Lesen ersetzt der Compiler automatisch **Bezeichner** durch **Token-Zeichenfolge**, was normalerweise für Konstantendefinitionen verwendet wird.

Als Beispiel hier ein Sketch, der `define` verwendet, was die Lesbarkeit des Codes verbessert.

```
#define BOARD_LED 13
#define VERZÖGERUNGSZEIT 500

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

void loop() {
    digitalWrite(BOARD_LED, HIGH);
    delay(VERZÖGERUNGSZEIT);
    digitalWrite(BOARD_LED, LOW);
    delay(VERZÖGERUNGSZEIT);
}
```

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);
}
```

Wir können sehen, dass der **Bezeichner** ersetzt wurde und im Programm nicht existiert. Deshalb gibt es einige Vorichtsmaßnahmen bei der Verwendung.

1. Eine Token-Zeichenfolge kann nur manuell geändert und nicht durch Arithmetik im Programm in andere Werte umgewandelt werden.
2. Vermeiden Sie die Verwendung von Symbolen wie `;`. Zum Beispiel:

```
#define BOARD_LED 13;

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

void loop() {
    digitalWrite(BOARD_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 Benennungskonvention für #define besteht darin, den Bezeichner zu groß zu schreiben, um Verwirrung mit Variablen zu vermeiden.

2.3.7 Variable

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

Im folgenden Skizzenbeispiel werden Variablen verwendet. Es speichert die Pin-Nummern der an Bord befindlichen 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);  
}
```

Warte mal, ist das nicht das Gleiche, was #define tut? Die Antwort ist NEIN.

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

Die folgende Skizze addiert selbst zur Variable und lässt die an Bord befindliche LED nach jedem Blinken länger blinken.

```
int ledPin = 13;  
int delayTime = 500;  
  
void setup() {  
    pinMode(ledPin,OUTPUT);  
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
void loop() {
    digitalWrite(ledPin,HIGH);
    delay(delayTime);
    digitalWrite(ledPin,LOW);
    delay(delayTime);
    delayTime = delayTime+200; //Bei jeder Ausführung wird der Wert um 200 erhöht
}
```

Eine Variable deklarieren

Eine Variable zu deklarieren bedeutet, eine Variable zu erstellen.

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

Betrachten wir diese Variable als Beispiel.

```
int delayTime;
```

Datentyp

Hier ist `int` ein Datentyp, der als Integer-Typ bezeichnet wird und verwendet werden kann, um Ganzzahlen von -32768 bis 32766 zu speichern. Es kann auch nicht verwendet werden, um Dezimalzahlen zu speichern.

Variablen können neben Ganzzahlen auch andere Arten von Daten speichern. Die Arduino-Sprache (die, wie wir uns erinnern, C++ ist) hat eingebauten Support für einige von ihnen (hier sind nur die am häufigsten verwendeten und nützlichsten aufgeführt):

- `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. Da es als `char` markiert ist, wird der Compiler versuchen, es einem Zeichen aus dem zuzuordnen.
- `string`: Kann eine Zeichenkette speichern, z.B. Halloween.

Variablenname

Sie können der Variable einen beliebigen Namen geben, wie `i`, `apple`, `Bruce`, `R2D2`, `Sectumsempra`, aber es gibt einige grundlegende Regeln zu beachten.

1. Beschreiben Sie, wofür es verwendet wird. Hier habe ich die Variable `delayTime` genannt, damit man leicht verstehen kann, was sie tut. Es wäre in Ordnung, wenn ich die Variable `barryAllen` nenne, aber es verwirrt die Person, die den Code anschaut.
2. Verwenden Sie eine gebräuchliche Schreibweise. Sie können CamelCase verwenden, wie ich es getan habe, mit dem anfänglichen T in `delayTime`, damit man leicht sehen kann, dass die Variable aus zwei Wörtern besteht. Sie können auch UnderScoreCase verwenden, um die Variable als `delay_time` zu schreiben. Es hat keinen Einfluss auf das Laufen des Programms, aber es würde dem Programmierer helfen, den Code zu lesen, wenn Sie die bevorzugte Schreibweise verwenden.
3. Verwenden Sie keine Schlüsselwörter. Ähnlich wie beim Tippen von „`int`“ färbt die Arduino IDE es ein, 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 eingefärbt ist.

4. Spezialsymbole sind nicht erlaubt. Zum Beispiel: Leerzeichen, #, \$, /, +, %, usw. Die Kombination aus englischen Buchstaben (Groß-/Kleinschreibung beachten), Unterstrichen und Zahlen (aber Zahlen dürfen nicht als erstes Zeichen eines Variablennamens verwendet werden) ist reichhaltig genug.

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 einen Wert zuweisen, sobald wir sie deklarieren.

```
int delayTime = 500;
```

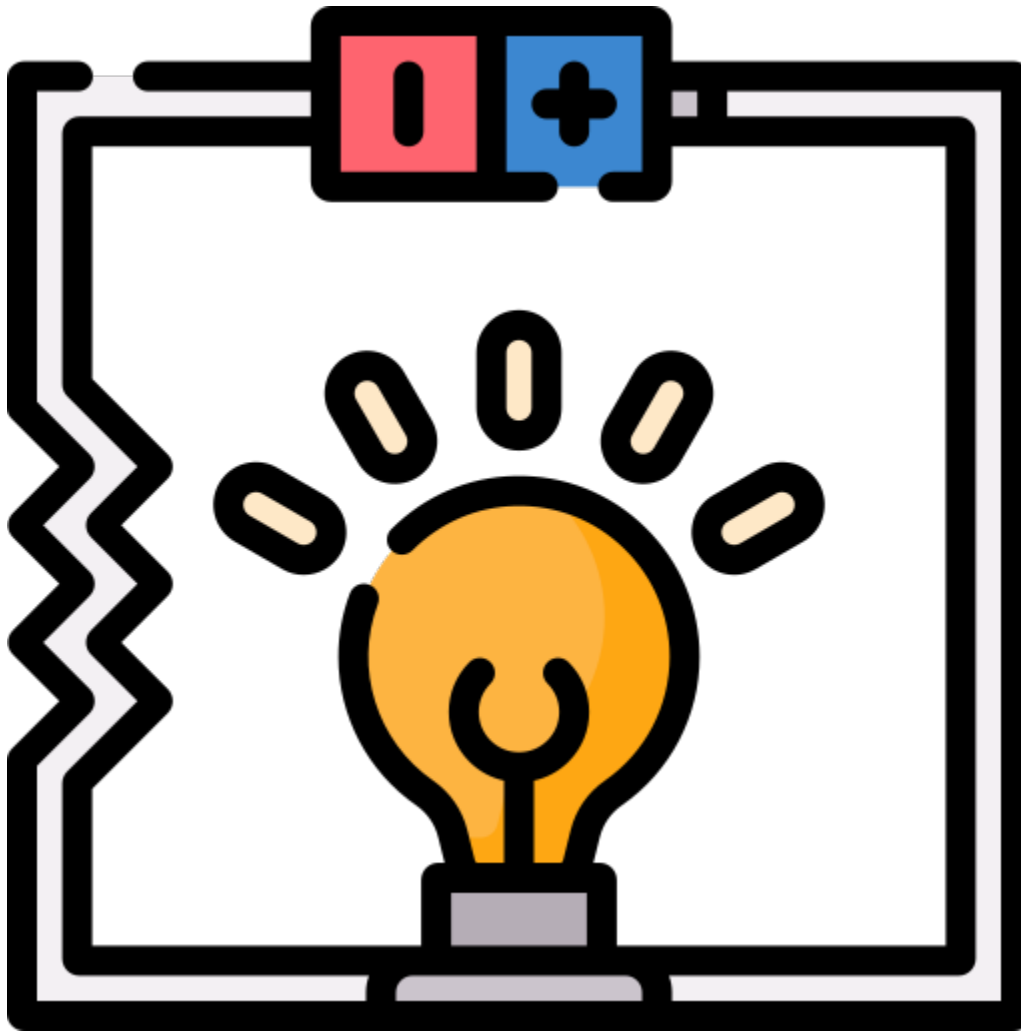
Es ist auch möglich, ihr zu einem späteren Zeitpunkt einen neuen Wert zuzuweisen.

```
int delayTime; // kein Wert
delayTime = 500; // Wert ist 500
delayTime = delayTime + 200; // Wert ist 700
```

2.3.8 Wie man den Stromkreis aufbaut

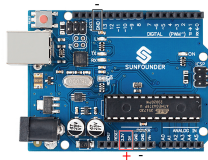
Viele der Dinge, die Sie täglich verwenden, werden mit Strom betrieben, wie die Lichter in Ihrem Haus oder der Computer, auf dem Sie gerade lesen.

Um Strom zu nutzen, müssen Sie einen elektrischen Stromkreis aufbauen. Grundsätzlich ist ein Stromkreis ein Pfad, durch den Strom fließt, oder ein elektronischer Stromkreis, und besteht aus elektrischen Geräten und Bauteilen, die auf eine bestimmte Weise miteinander verbunden sind, wie Widerstände, Kondensatoren, Stromquellen und Schalter.



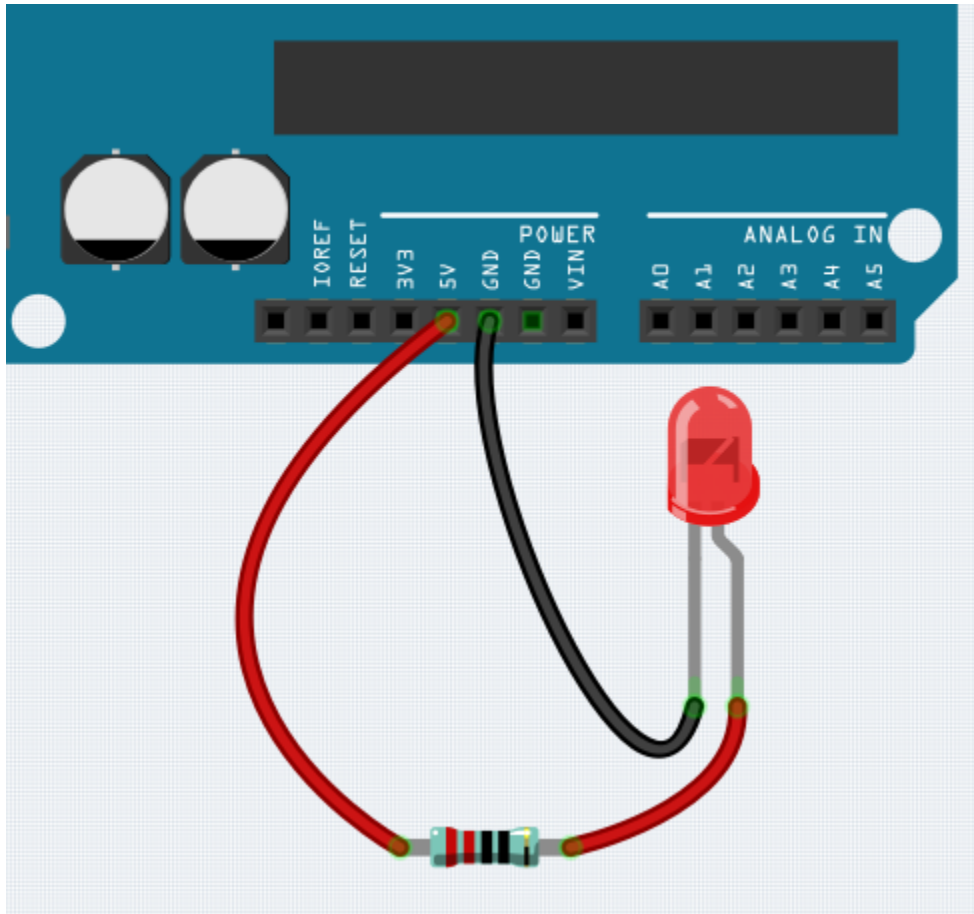
Ein Stromkreis ist ein geschlossener Pfad, in dem Elektronen sich bewegen, um einen elektrischen Strom zu erzeugen. Damit der Strom fließt, muss ein leitender Pfad zwischen dem positiven Anschluss der Stromquelle und dem negativen Anschluss bestehen. Dies wird als geschlossener Stromkreis bezeichnet (wenn er unterbrochen ist, spricht man von einem offenen Stromkreis).

Das Arduino Board verfügt über einige Stromausgangspins (positiv) und einige Erdungspins (negativ). Diese Pins können als positive und negative Seite der Stromversorgung verwendet werden, indem Sie die Stromquelle in das Board stecken.



Mit Strom können Sie Werke mit Licht, Ton und Bewegung erschaffen. Sie können eine LED zum Leuchten bringen, indem Sie den langen Pin an den positiven Anschluss und den kurzen Pin an den negativen Anschluss anschließen. Die LED wird sehr schnell kaputtgehen, wenn Sie dies tun, also müssen Sie einen 220* Widerstand in den Stromkreis einfügen, um ihn zu schützen.

Der von ihnen gebildete Stromkreis ist unten dargestellt.



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

In dieser Situation werden Lötfreie Steckplatinen Ihre besten Verbündeten sein.

Hallo, Steckplatine!

Eine Steckplatine ist eine rechteckige Kunststoffplatte mit vielen kleinen Löchern. Diese Löcher ermöglichen es uns, elektronische Bauteile einfach einzufügen und elektronische Schaltkreise aufzubauen. Steckplatinen fixieren elektronische Bauteile nicht dauerhaft, so dass wir einen Schaltkreis bei Fehlern einfach reparieren und von vorne beginnen können.

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

Im Internet finden wir viele Informationen über Steckplatinen.

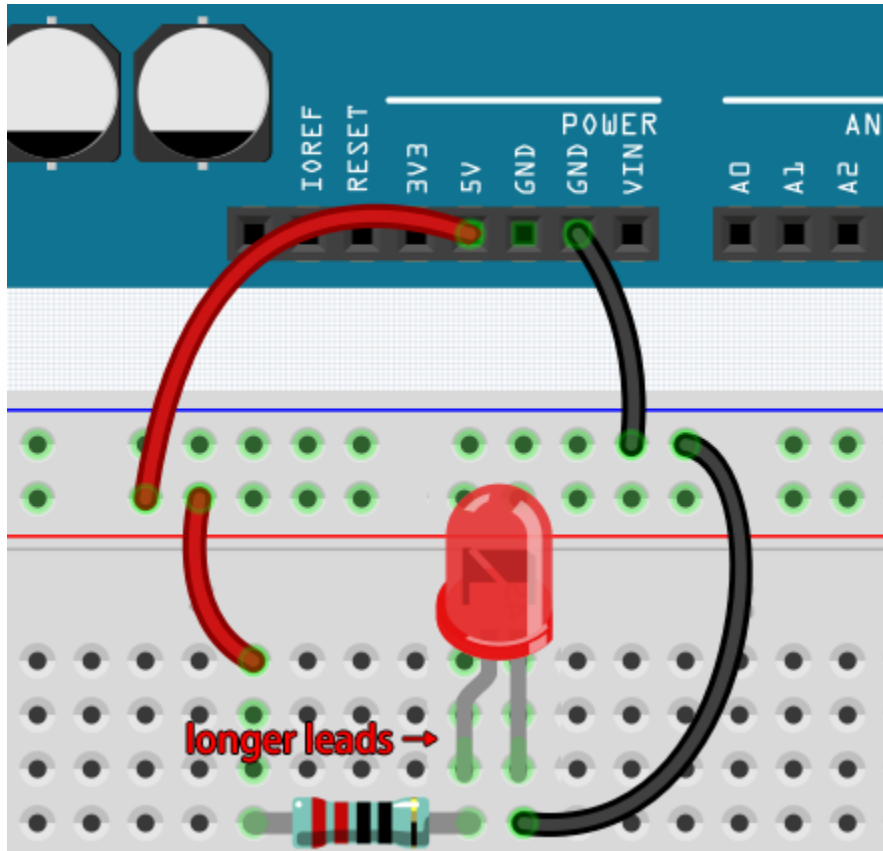
- [Wie man eine Steckplatine verwendet - Science Buddies](#)
- [Was ist eine STECKPLATINE? - Makezine](#)

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

1. Jede halbe Reihengruppe (wie Spalte A-E in Reihe 1 oder Spalte F-J in Reihe 3) ist verbunden. Wenn also ein elektrisches Signal von A1 einfließt, kann es von B1, C1, D1, E1, aber nicht von F1 oder A2 herausfließen.

2. In den meisten Fällen werden beide Seiten der Steckplatine als Stromschienen verwendet, und die Löcher in jeder Spalte (etwa 50 Löcher) sind miteinander verbunden. Als Faustregel werden positive Stromquellen an die Löcher in der Nähe des roten Drahts angeschlossen, und negative Stromquellen an die Löcher in der Nähe des blauen Drahts.
3. In einem Schaltkreis fließt der Strom von der positiven zur negativen Polklemme, nachdem er durch die Last geflossen ist. In diesem Fall kann ein Kurzschluss auftreten.

Lassen Sie uns die Richtung des Stroms verfolgen, 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 männlich-zu-männlich (M2M) Verbindungskabel, um es mit der roten Stromschiene zu verbinden.
2. Um die LED zu schützen, muss der Strom durch einen 220-Ohm-Widerstand fließen. Verbinden Sie ein Ende (beliebiges Ende) des Widerstands mit der roten Stromschiene und das andere Ende mit der freien Reihe der Steckplatine.

Bemerkung: Der Farbring des 220-Ohm-Widerstands ist rot, rot, schwarz, schwarz und braun.

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

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

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 männlich-zu-männlich (M2M) Verbindungskabel, um den kurzen LED-Pin mit der negativen Stromschiene der Steckplatine zu verbinden.
5. Verbinden Sie den GND-Pin des Boards mit der negativen Stromschiene mithilfe eines Jumpers.

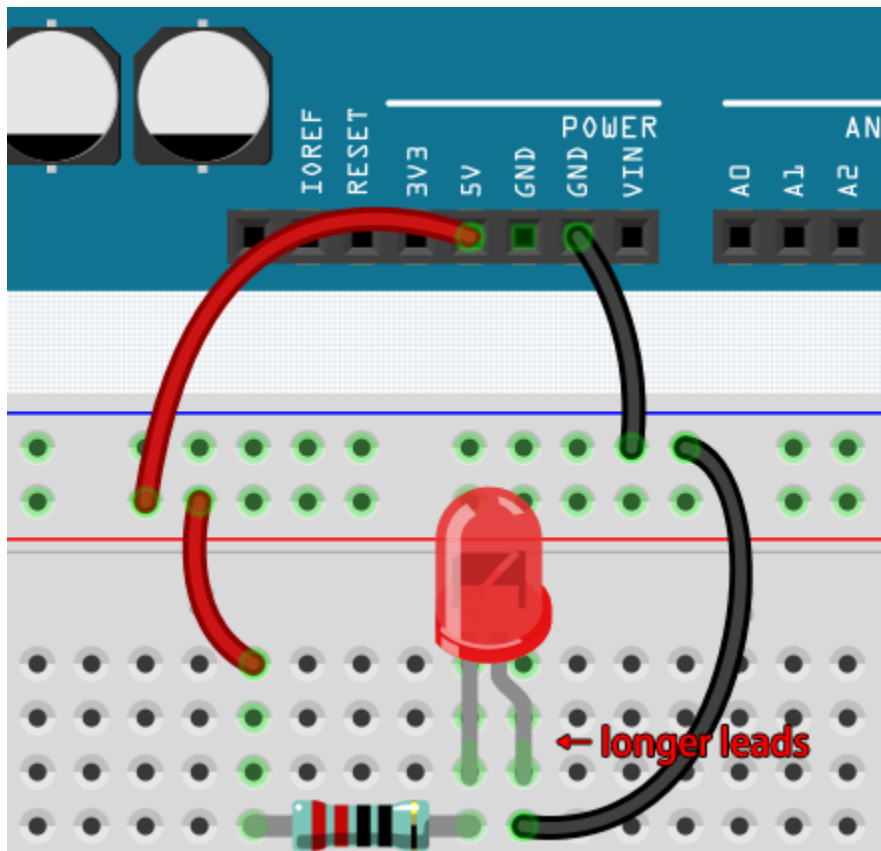
Vorsicht vor Kurzschlüssen

Kurzschlüsse können auftreten, wenn zwei Komponenten, die nicht verbunden sein sollten, „versehentlich“ verbunden werden. Dieses Kit enthält Widerstände, Transistoren, Kondensatoren, LEDs usw., die lange Metallpins haben, die aneinander stoßen und einen Kurzschluss verursachen können. Einige Schaltungen funktionieren einfach nicht richtig, wenn ein Kurzschluss auftritt. Gelegentlich kann ein Kurzschluss Bauteile dauerhaft beschädigen, insbesondere zwischen der Stromquelle und der Erdungsschiene, wodurch der Schaltkreis sehr heiß wird, das Plastik auf der Steckplatine schmilzt und sogar die Bauteile verbrennt!

Stellen Sie daher immer sicher, dass die Pins aller Elektronikbauteile auf der Steckplatine sich nicht berühren.

Richtung des Schaltkreises

Schaltkreise haben eine Orientierung und diese spielt bei bestimmten elektronischen Komponenten eine bedeutende Rolle. Es gibt Bauteile mit Polarität, was bedeutet, dass sie basierend auf ihren positiven und negativen Polen korrekt verbunden werden müssen. Schaltkreise mit falscher Orientierung 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 manche Bauteile keine Richtung, wie zum Beispiel die Widerstände in diesem Schaltkreis. Daher können Sie diese umkehren, ohne die normale Funktion der LEDs zu beeinflussen.

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

Schutz des Schaltkreises

Strom ist die Geschwindigkeit, mit der Elektronen an einem Punkt in einem geschlossenen elektrischen Schaltkreis vorbeifließen. Im Grundlegendsten ist Strom = Fluss. Ein Ampere (kurz: Amp) ist die internationale Einheit zur Messung von Strom. Es drückt die Menge an Elektronen (manchmal als „elektrische Ladung“ bezeichnet) aus, die über einen Zeitraum an einem Punkt im Schaltkreis vorbeifließen.

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

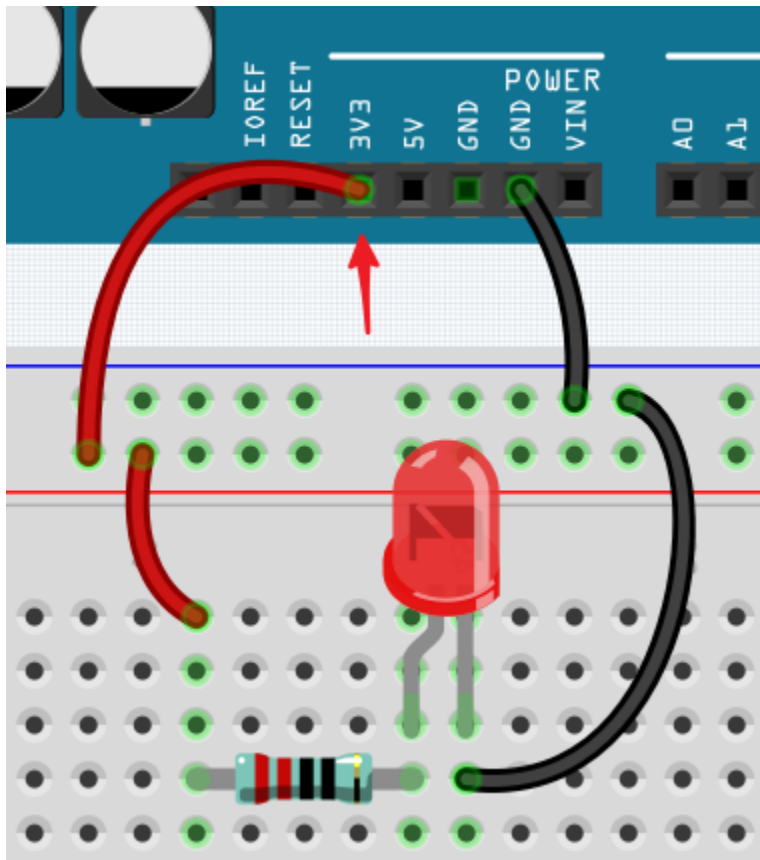
Widerstand (R) ist die Eigenschaft des Materials, die den Stromfluss einschränkt und wird in Ohm (Ω) gemessen.

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

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

- [Ohmsches Gesetz - Wikipedia](#)

Zum Ohmschen Gesetz können wir ein einfaches Experiment durchführen.



Wenn Sie das Kabel, das 5V mit 3,3V verbindet, wechseln, wird die LED dunkler. Wenn Sie den Widerstand von 220 Ohm auf 1000 Ohm ändern (Farbring: braun, schwarz, schwarz, braun und braun), werden Sie feststellen, dass die LED dunkler als zuvor wird. Je größer der Widerstand, desto dunkler die LED.

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

Die meisten verpackten Module benötigen nur Zugang zur richtigen Spannung (normalerweise 3,3V oder 5V), wie zum Beispiel das Ultraschallmodul.

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

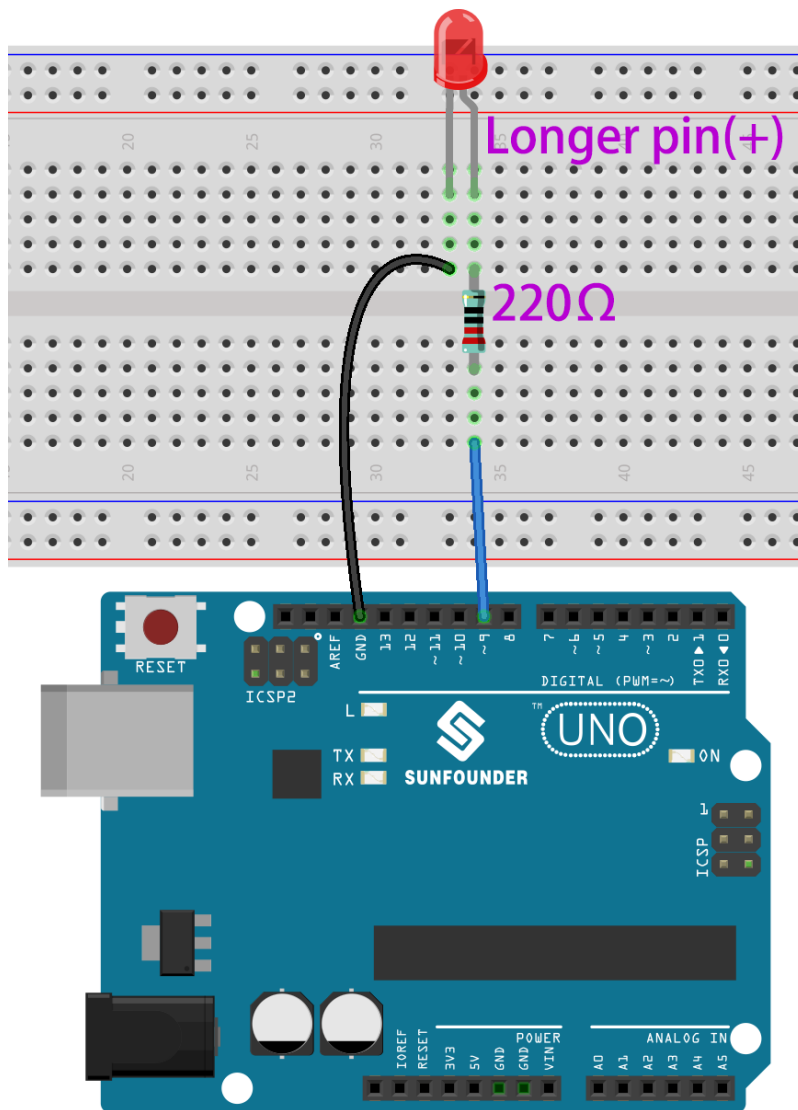
Als Beispiel: LEDs verbrauchen normalerweise 20mA Strom und ihr Spannungsabfall beträgt etwa 1,8V. Laut Ohm'schem Gesetz benötigen wir bei einer 5V-Stromversorgung einen Widerstand von mindestens 160 Ohm $((5-1,8)/20\text{mA})$, um die LED nicht zu zerstören.

Schaltkreissteuerung mit Arduino

Jetzt, da wir ein grundlegendes Verständnis für Arduino-Programmierung und elektronische Schaltungen haben, ist es Zeit für die wichtigste Frage: Wie steuert man Schaltungen mit Arduino?

Einfach ausgedrückt, steuert Arduino einen Schaltkreis, indem es den Pegel der Pins auf dem Board ändert. Zum Beispiel steuert es eine an Bord befindliche LED, indem es ein hohes oder niedriges Signal an Pin 13 sendet.

Jetzt versuchen wir, das Arduino-Board zu programmieren, um die blinkende LED auf dem Steckbrett zu steuern. 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;

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 an Bord befindlichen LED zu steuern. Der Unterschied ist, dass der Wert von ledPin dieses Mal auf 9 geändert wurde.

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

Arduino-Videokurs

Begeben Sie sich mit dem umfassenden Arduino-Videokurs auf eine Entdeckungsreise durch die Arduino-Welt, unter Verwendung des 3-in-1-Starterkits von SunFounder. Diese Serie startet mit einer Einführung in das Arduino-Ökosystem und die Fähigkeiten des Uno R3 Boards, was den Grundstein für eine intensive Auseinandersetzung mit praktischen Anwendungen und Programmierungstechniken legt. Sie erlernen die Grundlagen der Steuerung von LEDs, des Verständnisses für serielle Kommunikation und der Manipulation verschiedener Komponenten wie RGB-LEDs, Tasten und Schieberegistern. Der Kurs bietet auch eine Vielzahl von spannenden Projekten wie Smart-Car-Projekte und IoT-Projekte. Wenn Sie dem Kurs Schritt für Schritt folgen, werden Sie Arduino meistern – nicht nur Code kopieren und einfügen, sondern Ihren eigenen Code schreiben und Ihre Arduino-Projekte nach Ihren Vorstellungen umsetzen.

Projekte

3.1 Video 1 - Vorstellung des Kits

Dieses anfangersfreundliche Tutorial führt Arduino als Open-Source-Plattform ein und beschreibt die Komponenten und Anwendungsmöglichkeiten des dreiteiligen Arduino-Kits von SunFounder.

- **Kurseinführung:** Einführung in den Robojax Arduino-Kurs, zugeschnitten auf absolute Anfänger.
- **Arduino-Grundlagen:** Erläuterung von Arduino als zugängliche und Open-Source-Elektronikplattform.
- **Kursanforderungen:** Übersicht der notwendigen Software- und Hardwareanforderungen zum Start mit Arduino.
- **Arduino-Boards erklärt:** Detaillierte Betrachtung verschiedener Arduino-Boards, einschließlich Uno, Mega und Wi-Fi-Varianten.
- **Unboxing des Kits:** Umfassendes Auspacken des SunFounder-Kits, Vorstellung seiner vielfältigen Komponenten.

Video

3.2 Video 2 - Einführung in die Arduino IDE

Dieses Tutorial bietet eine Anleitung für Anfänger zur Einrichtung der Arduino-Software und zur Erkundung des dreiteiligen Arduino-Kits von SunFounder, essentiell für Heim- und Schulautomationsprojekte.

- **Einrichtung der Arduino IDE:** Anleitung zum Herunterladen und Installieren der Arduino IDE, einschließlich Systemanforderungen.
- **Navigation in der Arduino IDE:** Detaillierte Einführung in die Benutzeroberfläche und Funktionen der Arduino IDE.
- **Anleitung zur Board-Verbindung:** Tutorial zur Verbindung und Einrichtung eines Arduino-Boards mit einem Computer.
- **Tutorial zur Arduino-Dokumentation:** Nutzung der offiziellen Arduino-Dokumentation und der Community-Foren für Programmierunterstützung.
- **Erkundung des SunFounder Kits:** Übersicht über die Dokumentation des SunFounder Kits und die Projektressourcen.

Video

3.3 Video 3: Grundlagen der Programmierung und LED-Projekte

Dieses Tutorial bietet eine umfassende Einführung in die Arduino-Programmierung, einschließlich der grundlegenden Code-Struktur, der Board-Komponenten und eines anfangergefreundlichen Projekts zum Blinken einer LED.

- **Code-Struktur:** Detaillierte Erklärung der Funktionen `setup()` und `loop()` in der Arduino-Programmierung.
- **Arduino Uno Board:** Tiefgehende Betrachtung des Arduino Uno Boards, einschließlich der Funktionen und Verwendung der Pins.
- **Verständnis von Widerständen:** Erläuterung von Widerständen, einschließlich des Lesens von Farbcodes und der Messung des Widerstands.
- **Grundlagen der LEDs:** Übersicht über Leuchtdioden (LEDs), einschließlich der Identifikation von Anode und Kathode.
- **Praktisches LED-Projekt:** Schritt-für-Schritt-Anleitung zur Einrichtung eines einfachen LED-Blinkprojekts auf einem Steckbrett.

Video

Verwandte Online-Tutorials

- [1.1 Hallo, LED!](#)

3.4 Video 4: Zahlensysteme und Serieller Monitor

Dieses Video vertieft sich in die Grundlagen verschiedener Zahlensysteme, die in Arduino verwendet werden, und zeigt praktische Anwendungen des Seriellen Monitors für Programmierung und Debugging.

- **Erklärung von Zahlensystemen:** Detaillierte Erklärung der dezimalen, binären, oktalen und hexadezimalen Zahlensysteme und ihrer Bedeutung in Arduino.
- **Techniken der Datenkonvertierung:** Methoden zur Umwandlung von Daten zwischen verschiedenen Zahlensystemen.

- **Verwendung des Seriellen Monitors:** Anleitung zur Nutzung des Arduino Seriellen Monitors für Debugging und Datenvisualisierung.
- **Daten im Seriellen Monitor ausgeben:** Techniken zum Ausdrucken von ASCII-Werten, Zeichen und numerischen Daten im Seriellen Monitor.
- **Verarbeitung von Benutzereingaben:** Demonstrationen des Lesens und Interpretierens von Benutzereingaben über den Seriellen Monitor.

Video

3.5 Video 5: Datentypen und Variablen

Dieses Video bietet eine ausführliche Anleitung zu verschiedenen Datentypen und Variablen in Arduino, mit Schwerpunkt auf ihrer Deklaration, Speichernutzung und praktischen Anwendungen in Skizzen.

- **Verständnis von Datentypen:** Umfassende Erläuterung von Datentypen wie Ganzzahlen, Zeichen, Fließkommazahlen und Booleschen Werten in Arduino.
- **Deklaration und Verwendung von Variablen:** Detaillierte Anleitung zur Deklaration, Zuweisung und Verwendung von Variablen in Arduino-Skizzen.
- **Darstellung numerischer Daten:** Einblicke in die Darstellung numerischer Daten als Binär-, Hexadezimal- und Dezimalwerte.
- **String-Manipulation:** Techniken zur String-Konkatenation, Konvertierung und Manipulation in Arduino.
- **Überblick über spezielle Datentypen:** Erkundung spezieller Datentypen wie Bytes, vorzeichenlose Zeichen und Wörter, einschließlich ihrer Speicherung und Reichweite.

Video

3.6 Video 6: Steuerung von Summer, Motor und Wasserpumpe

Dieses Video bietet praktische Anleitungen zur Steuerung eines aktiven Summers, Motors und einer Wasserpumpe mit Arduino, was essentielle Fähigkeiten für Robotik und automatisierte Systeme vermittelt.

- **Überblick über das SunFounder Kit:** Einführung in das dreiteilige Arduino-Kit von SunFounder und seine Nützlichkeit in Robotik- und Heimautomationsprojekten.
- **Steuerung des aktiven Summers:** Schritt-für-Schritt-Anleitung zum Anschließen und Programmieren eines aktiven Summers mit Arduino.
- **Tutorial zur Motorsteuerung:** Detaillierte Anweisungen zur Verwendung eines L298N-Motortreibers für die Motorsteuerung in Smart-Car-Anwendungen.
- **Wasserpumpenprojekt:** Demonstration der Steuerung einer Wasserpumpe für automatisierte Bewässerungssysteme.
- **Einrichtung der Arduino-Umgebung:** Wie man Arduino-Skizzen für jedes Projekt einrichtet und modifiziert.
- **Praktische Projektausführung:** Praktische Durchführung jedes Projekts mit Tipps zur Fehlersuche und Optimierung.

Video

Verwandte Online-Tutorials

- [1.2 Piepton](#)

- [1.3 Das Rad drehen](#)
- [1.4 Pumpen](#)

3.7 Video 7: Taster und Reed-Schalter

Dieses Arduino-Tutorial untersucht die Grundlagen des digitalen Lesens, mit einem Fokus auf das Erkennen von Tastendrücken und das Sensieren von Magnetfeldern mit Reed-Schaltern.

- **Grundlagen des Digitalen Lesens:** Verständnis des Konzepts des digitalen Lesens in Arduino, Unterscheidung zwischen hohen und niedrigen Signalen.
- **Erkennung von Tastendrücken:** Detaillierte Anleitung zum Anschließen eines Tasters und zum Lesen seines Zustands mit Arduino.
- **Sensieren mit Reed-Schaltern:** Demonstration der Verwendung eines Reed-Schalters zur Erkennung von Magnetfeldern, einschließlich Verdrahtung und Code-Einrichtung.
- **Einrichtung der Arduino IDE:** Schritt-für-Schritt-Anleitung zur Vorbereitung der Arduino IDE für Projekte des digitalen Lesens.
- **Verwendung interner Pull-Up-Widerstände:** Techniken zur Verwendung eines Tasters ohne externen Widerstand durch Nutzung der eingebauten Pull-Up-Widerstände von Arduino.
- **Fehlerbehebungstipps:** Lösungen für gängige Probleme beim digitalen Lesen, wie beispielsweise floating pins.

Video

Verwandte Online-Tutorials

- [3.1 Tastenwert auslesen](#)
- [3.2 Magnetismus spüren](#)

3.8 Video 8: PWM und Schleifenstrukturen

Dieses Arduino-Tutorial behandelt die Verwendung von PWM zur Steuerung von Geräten wie Motoren und LEDs, sowie einen umfassenden Leitfaden zur Programmierung von Schleifen wie for, while und do-while.

- **PWM-Grundlagen:** Verständnis von PWM und Tastverhältnissen zur Steuerung des Verhaltens von Geräten.
- **Analoges Schreiben mit Arduino:** Einsatz von *analogWrite* zur Modulation der Signalstärke für Motoren und LEDs.
- **For-Schleifen-Demonstrationen:** Praktische Beispiele von for-Schleifen für das allmähliche Verblässen von LEDs.
- **Verwendung der While-Schleife:** Implementierung von while-Schleifen für eine effiziente Programmausführung.
- **Mechanik der Do-While-Schleife:** Erforschung der einzigartigen Struktur und Anwendungen von do-while-Schleifen in Arduino-Projekten.
- **Praktische LED-Steuerung:** Schritt-für-Schritt-Einrichtung und Code zur Anpassung der LED-Helligkeit mit PWM.

Video

Verwandte Online-Tutorials

- [2.1 Überblenden](#)

3.9 Video 9: Spannungsmessung

Lernen Sie, wie man Gleichspannung mit Arduinos Funktion `analogRead` liest und misst, verstehen Sie ADC und erkunden Sie die Verwendung von Potentiometern für präzise Spannungsmessungen.

- **AnalogRead-Funktion:** Eine Einführung in die Verwendung von Arduinos `analogRead` zur Messung von Gleichspannung.
- **Überblick über analoge Pins:** Verständnis der Rolle analoger Pins (A0 - A5) bei der Spannungslesung.
- **Signaltypen:** Unterscheidung zwischen digitalen und analogen Signalen sowie deren Spannungspegel.
- **Grundlagen des ADC:** Erläuterung der Analog-Digital-Umwandlung und die Bedeutung von 10-Bit- und 12-Bit-Auflösungen.
- **Potentiometertypen:** Demonstration verschiedener Potentiometer zur Spannungsmessung.
- **Methoden der Spannungsmessung:** Detaillierte Methoden zur genauen Ablesung und Anzeige von Spannungswerten.

Video

Verwandte Online-Tutorials

- [4.1 Den Drehknopf drehen](#)

3.10 Video 10: Bedingte Anweisungen und Arrays

Entdecken Sie die Grundlagen von bedingten Anweisungen und Arrays in Arduino, einschließlich praktischer Anwendungen in der LED-Steuerung und Datenverwaltung.

- **Bedingte Anweisungen:** Erlernen der Verwendung von `if`-, `else`- und `else-if`-Anweisungen in Arduino für Entscheidungsfindungen.
- **Praktische Beispiele:** Anwendung bedingter Logik in Szenarien wie Temperaturkontrolle und Verkehrsregelung.
- **Arrays in Arduino:** Demonstration der Erstellung, Modifikation und Nutzung von Arrays zur Speicherung und Verwaltung von Daten.
- **LED-Steuerung mit Arrays:** Wie man Arrays für die effiziente Steuerung mehrerer LEDs verwendet.
- **Verständnis von Floats:** Einführung in den Datentyp `float` und Verwaltung der Präzision in numerischen Operationen.
- **Verwendung von Schleifen mit Arrays:** Einsatz von `for`-Schleifen zur Iteration über Array-Elemente, Vereinfachung des Codes für Aufgaben wie LED-Sequenzierung.

Video

3.11 Video 11: DHT11-Sensor

Dieses Video bietet eine ausführliche Anleitung zu Arduino-Bibliotheken und der praktischen Verwendung des DHT11-Temperatur- und Feuchtigkeitssensors, einschließlich Installation, Programmierung und einem Anwendungsbeispiel.

- **Einführung in Arduino-Bibliotheken:** Erklärung des Konzepts, Zwecks und der Installationsmethoden von Arduino-Bibliotheken.

- **Fähigkeiten des DHT11-Sensors:** Technische Spezifikationen, einschließlich Temperatur- und Feuchtigkeitsbereich, Genauigkeit und Betriebsspannung.
- **Anleitung zur Sensorverdrahtung:** Schritt-für-Schritt-Anweisungen, wie der DHT11-Sensor korrekt an ein Arduino-Board angeschlossen wird.
- **Sensordaten lesen:** Techniken zur Verwendung von Arduino-Code zum Lesen und Interpretieren von Daten des DHT11-Sensors.
- **Programmierdurchlauf:** Detaillierte Erläuterung des für den Betrieb des DHT11-Sensors notwendigen Arduino-Codes.
- **Praktische Demonstration:** Umsetzung einer realen Anwendung durch Auslösen eines Summers basierend auf Temperaturmessungen des DHT11-Sensors.

Video

Verwandte Online-Tutorials

- *5.11.3 Temperatur - Luftfeuchtigkeit*

3.12 Video 12: Funktionen und Switch-Case

Erforschen Sie die Grundlagen der Definition und Verwendung von Funktionen in der Arduino-Programmierung, einschließlich verschiedener Arten von Funktionen, Parameterhandhabung, Rückgabewerten und der Verwendung von Switch-Case-Anweisungen.

- **Verständnis von Funktionen in Arduino:** Einführung in die grundlegende Struktur und den Zweck von Funktionen in der Arduino-Programmierung.
- **Arten von Funktionen:** Erklärt den Unterschied zwischen Void-Funktionen und solchen, die spezifische Datentypen wie Ganzzahlen oder Fließkommazahlen zurückgeben.
- **Verwendung von Parametern und Rückgabewerten:** Demonstriert, wie Parameter an Funktionen übergeben werden und wie die von ihnen zurückgegebenen Werte im Hauptprogramm genutzt werden können.
- **Praktische Funktionsbeispiele:** Bietet reale Beispiele für Funktionen für verschiedene Berechnungen und bedingte Operationen.
- **Syntax und Verwendung von Switch-Case:** Bietet eine Alternative zu If-Else-Anweisungen und zeigt, wie Switch-Case die Code-Struktur vereinfachen kann.
- **Anwendung von Funktionen in Projekten:** Zeigt praktische Anwendungen von Funktionen in Arduino-Projekten, wie z.B. die Steuerung von Geräten oder die Verarbeitung von Sensordaten.

Video

3.13 Video 13: Joystick

Dieses Tutorial behandelt die Verwendung eines XY-Joysticks mit Arduino, einschließlich seiner Struktur, Verkabelung, Positionserfassung, Tastendruckererkennung und Programmierung von Arduino zur Reaktion auf Joystick-Bewegungen.

- **Mechanik und Struktur des Joysticks:** Ein Überblick über die Komponenten des XY-Joysticks, einschließlich seiner Potentiometer für die Richtungssteuerung und des integrierten Tasters.
- **Arduino-Joystick-Verbindung:** Detaillierte Anweisungen zur korrekten Verkabelung des Joysticks mit einem Arduino-Board für genaue Signallerfassung.

- **Interpretation von Joystick-Bewegungen:** Techniken zur Erfassung der Bewegungen der X- und Y-Achse des Joysticks durch Arduino-Programmierung.
- **Erkennung von Tastendrücken:** Anleitung zur Erkennung und Programmierung von Reaktionen auf die Aktivierung des Joystick-Tasters.
- **Arduino-Code für Joystick-Eingaben:** Schritt-für-Schritt-Anleitung zur Erstellung von Arduino-Code zur Interpretation und Anzeige von Joystick-Bewegungen auf einem seriellen Monitor.
- **Reaktive Aktionen auf Joystick-Eingaben:** Demonstration, wie man Arduino programmiert, um spezifische Aktionen, wie das Auslösen eines Buzzers, basierend auf der Joystick-Position durchzuführen.

Video

Verwandte Online-Tutorials

- [4.3 Den Joystick betätigen](#)

3.14 Video 14: millis() und map()

Dieses Tutorial konzentriert sich auf die Verwendung von `millis()` zur Zeitverfolgung und `map()` zur Wertumwandlung in Arduino, mit praktischen Anwendungen wie zeitgesteuerten Tastenreaktionen und LED-Helligkeitssteuerung.

- **Funktionalität von Millis:** `millis()` als Zeitmessungsfunktion in Arduino, beginnend bei Null beim Programmstart und sich jede Millisekunde erhöhend.
- **Zeitgesteuerte Ereignisse mit millis():** Wie `millis()` zur Ausführung von Ereignissen in bestimmten Zeitintervallen genutzt wird, ohne das gesamte Programm anzuhalten, im Gegensatz zu `delay()`.
- **Zeitmessung bei Tastendrücken:** Beispiel für die Verwendung von `millis()` zur Erkennung eines Tastendrucks und Ausführung einer Aktion nach einer festgelegten Dauer.
- **Einführung in die Map-Funktion:** Vorstellung der `map()`-Funktion, die zur Umwandlung numerischer Werte von einem Bereich in einen anderen verwendet wird.
- **Einstellung der LED-Helligkeit:** Praktische Demonstration der Verwendung von `map()` zur Anpassung der LED-Helligkeitsstufen, indem Prozentwerte in den PWM-Bereich abgebildet werden.
- **Effizientes Programmieren mit millis() und map():** Vorführung effizienter Arduino-Programmierpraktiken durch die Kombination von `millis()` für nicht-blockierende Zeitmessung und `map()` für intuitive Wertumwandlung.

Video

3.15 Video 15: Automatisierung der Pflanzenbewässerung

Dieses Tutorial behandelt die Verwendung eines Bodenfeuchtigkeitssensors mit Arduino zur Überwachung der Bodenfeuchtigkeit und automatisiert die Bewässerung durch Code- und Hardware-Demonstrationen.

- **Grundlagen des Bodenfeuchtigkeitssensors:** Einführung in den Bodenfeuchtigkeitssensor, seine Anschlüsse und wie er die Bodenfeuchtigkeit misst.
- **Sensor-Arduino-Verkabelung:** Schritt-für-Schritt-Anleitung zum Anschließen des Feuchtigkeitssensors an den Arduino.
- **Arduino-Code für die Feuchtigkeitsmessung:** Erklärung des Arduino-Codes zur Lesung und Interpretation der Feuchtigkeitswerte vom Sensor.

- **Verständnis der Sensorausgabe:** Analyse der Ausgabewerte des Sensors zur Bestimmung der Bodenfeuchtigkeitsbedingungen.
- **Automatisierte Bewässerungslogik:** Entwicklung eines fortgeschrittenen Arduino-Codes zur Automatisierung der Bewässerung basierend auf den Bodenfeuchtigkeitswerten.
- **Praktische Umsetzung und Tests:** Demonstrationen des Sensors im Einsatz im Boden, die zeigen, wie das System einen Summer auslösen oder eine Wasserpumpe für die automatisierte Bewässerung steuern kann.

Video

Verwandte Online-Tutorials

- [4.4 Bodenfeuchtigkeit messen](#)

3.16 Video 16: Infrarot-Hindernisvermeidung

Lernen Sie, einen Infrarot-Hindernisvermeidungssensor mit Arduino einzurichten und zu programmieren, um Hindernisse zu erkennen und einen Summer zu aktivieren.

- **Verständnis der IR-Sensormechanik:** Überblick über die Komponenten des Infrarotsensors und wie sie funktionieren, um Hindernisse zu erkennen.
- **Sensor-Arduino-Verbindung:** Schritt-für-Schritt-Anleitung zur Verkabelung des Infrarotsensors mit dem Arduino-Board.
- **Programmierung zur Hinderniserkennung:** Einführung in das Schreiben und Verstehen von Arduino-Code zur Hinderniserkennung mit dem Sensor.
- **Echtzeit-Demonstration der Hinderniserkennung:** Demonstration der Fähigkeit des Sensors, Objekte in verschiedenen Entfernungen und Bedingungen zu erkennen.
- **Empfindlichkeitseinstellung:** Detaillierte Anleitung zur Anpassung der Empfindlichkeit des Sensors für unterschiedliche Erkennungsbereiche.
- **Integration eines Buzzers:** Modifikation der Einrichtung und des Codes, um einen Summer einzuschließen, der bei Hinderniserkennung aktiviert wird.

Video

Verwandte Online-Tutorials

- [3.3 Hindernisse erkennen](#)

3.17 Video 17: Unterbrechungen und LDR

Erforschen Sie die Funktionalitäten von Arduino-Unterbrechungen und lichtabhängigen Widerständen (LDRs), um intelligente Tages-/Nacht-Erkennungsprojekte zu erstellen.

- **Erklärung von Arduino-Unterbrechungen:** Tauchen Sie ein in die Mechanik von Arduino-Unterbrechungen für effizienteres Programmieren.
- **Effektive Nutzung von Interrupt-Pins:** Lernen Sie, welche Arduino-Pins Unterbrechungen unterstützen und wie sie verwendet werden.
- **Praktische Einrichtung von Interrupts:** Demonstration der Einrichtung eines Tasters und Summers zur Verwendung mit Arduino-Unterbrechungen.
- **Einführung in LDRs:** Entdecken Sie, wie LDRs ihren Widerstand basierend auf Licht verändern und welche Anwendungsmöglichkeiten sie bieten.

- **LDR-Messverfahren:** Lernen Sie, wie der Widerstand von LDRs unter verschiedenen Lichtbedingungen mit einem Multimeter gemessen wird.
- **Tages-/Nacht-Erkennungsprojekte:** Implementierung von LDR-basierten Projekten für das automatische Umschalten von Geräten wie Summern oder Lichtern basierend auf der Lichtintensität.

Video

Verwandte Online-Tutorials

- [5.13 Unterbrechung](#)

3.18 Video 18: Servomotor

Dieses Tutorial behandelt die Grundlagen der Verwendung von Servomotoren mit Arduino, von der Verkabelung und Programmierung bis hin zu praktischen Projektvorführungen.

- **Überblick über Servomotoren:** Einführung in Servomotoren, mit besonderem Augenmerk auf ihre Rolle in verschiedenen ferngesteuerten und robotischen Anwendungen.
- **Typen und Eigenschaften:** Untersuchung verschiedener Arten von Servomotoren und ihrer Drehmomentfähigkeiten.
- **Verständnis der Servokomponenten:** Aufschlüsselung der Teile eines Servomotors, einschließlich Arme und Zahnräder.
- **Arduino-Verkabelungsanleitung:** Schritt-für-Schritt-Anweisungen zum Anschließen von Servomotoren an Arduino-Boards.
- **Programmierung für Präzision:** Demonstration, wie man Arduino-Code verwendet, um Servowinkel und -bewegungen zu steuern.
- **Praktische Projektbeispiele:** Anwendungen und Modifikationen von Servomotoren in Arduino-Projekten.

Video

Verwandte Online-Tutorials

- [5.5 Interne Bibliothek verwenden](#)

3.19 Video 19: Ultraschallsensor

Dieses Tutorial erforscht die Verwendung eines Ultraschallsensors mit Arduino und zeigt, wie man diesen verdrahtet, programmiert und Entfernungen genau misst.

- **Grundlagen des Ultraschallsensors:** Einführung in Ultraschallsensoren und ihre Funktionalität bei der Entfernungsmessung.
- **Funktionsmechanismus:** Verständnis der Ultraschallwellen und ihres Reflexionsprinzips zur Entfernungsmessung.
- **Sensorverdrahtung zu Arduino:** Schritt-für-Schritt-Anleitung zum Anschließen des Ultraschallsensors an Arduino.
- **Bibliothekseinrichtung:** Wie man die notwendige Bibliothek für den Sensor herunterlädt und installiert.
- **Programmierung zur Messung:** Detaillierte Erklärung des Arduino-Codes zum Betrieb des Ultraschallsensors.
- **Praktische Messtests:** Demonstration der Fähigkeit des Sensors, verschiedene Entfernungen präzise zu messen.

Video

Verwandte Online-Tutorials

- [5.8 Benutzerdefinierte Funktion](#)

3.20 Video 20: LCD 1602

Dieses umfassende Tutorial behandelt die Verwendung des Arduino LCD 1602 zur Anzeige verschiedener Sensordaten, einschließlich Temperatur, Luftfeuchtigkeit und Entfernung.

- **Grundlagen des LCD 1602:** Einführung in die Funktionen und Fähigkeiten des LCD 1602.
- **Verwendung des I2C-Moduls:** Vereinfachung der LCD-Verbindungen mit dem I2C-Modul und dessen Verdrahtungsaufbau.
- **LCD-Adresskonfiguration:** Wie man die I2C-Adresse für das LCD korrekt einstellt und identifiziert.
- **Einrichtung der Arduino-Bibliothek:** Anleitung zur Installation der notwendigen Bibliothek für die LCD-Integration.
- **Anzeigen von Text und Zeichen:** Demonstration der grundlegenden Textanzeige und der Erstellung benutzerdefinierter Zeichen auf dem LCD.
- **Sensorintegration:** Vorführung der Integration eines DHT-Temperatursensors, eines Potentiometers und eines Ultraschallsensors mit dem LCD zur Echtzeitdatenanzeige.

Video

Verwandte Online-Tutorials

- [5.11.1 Flüssigkristallanzeige](#)
- [5.12 Serielle Lektüre](#)

3.21 Video 21: IR-Fernbedienung

Dieses Tutorial zeigt, wie man eine Infrarot-Fernbedienung mit Arduino zur Steuerung verschiedener Funktionen verwendet, wobei der Fokus auf der Signaldekodierung und praktischen Anwendungen wie der Aktivierung eines Buzzers liegt.

- **Verständnis der Infrarot-Technologie:** Einblick, wie Infrarot-Fernbedienungen und -Empfänger mit Arduino funktionieren.
- **Verdrahtung des IR-Empfängers:** Detaillierte Anleitung zur korrekten Verdrahtung des Infrarot-Empfängers an das Arduino-Board.
- **Installation notwendiger Bibliotheken:** Schritt-für-Schritt-Prozess zur Installation der IR-Fernbedienungsbibliothek in der Arduino IDE.
- **Dekodierung von IR-Signalen:** Erklärung, wie Signale einer IR-Fernbedienung mit Arduino dekodiert werden.
- **Aktion bei Tastendruck:** Programmierung von Arduino zur Ausführung spezifischer Aktionen basierend auf Eingaben der Fernbedienung.
- **Einblick in zukünftige Projekte:** Vorschau auf kommende fortgeschrittene Projekte mit Arduino und Infrarot-Technologie.

Video

Verwandte Online-Tutorials

- [5.11.2 IR-Empfänger](#)

3.22 Video 22: 7-Segment-Anzeige

Dieses Tutorial führt Sie durch die Erstellung eines digitalen Zählers und eines elektronischen Würfels mit Arduino, wobei sowohl Montage- als auch Programmieraspekte vorgestellt werden.

- **Montage der 7-Segment-Anzeige:** Anleitung zur Montage und zum Verständnis einer 7-Segment-Anzeige.
- **Einblicke in das Datenblatt der Anzeige:** Erläuterung des Datenblatts für das Modell 5161AS, mit Schwerpunkt auf Spezifikationen und optimaler Nutzung.
- **Verdrahtungsanleitung für den Zähler:** Detaillierte Anleitung zur Verdrahtung des digitalen Zählers, mit Fokus auf den 74HC595-Chip und die Anzeigeverbindungen.
- **Aufschlüsselung des Zähler-Codes:** Ausführliche Erklärung des Arduino-Codes zum Betrieb des digitalen Zählers.
- **Projekt Elektronischer Würfel:** Schritt-für-Schritt-Anleitung zum Bau und zur Programmierung eines elektronischen Würfels, einschließlich der Implementierung eines Taster-Interrupts.

Video

Verwandte Online-Tutorials

- [5.10 ShiftOut\(Segmentanzeige\)](#)

3.23 Video 23: Montage eines Smart Cars

Lernen Sie, wie man ein Smart Car unter Verwendung der Arduino-Plattform montiert und verkabelt, detailliert von Anfang bis Ende, einschließlich Motor- und Treibereinrichtung.

- **Montage des Smart Cars:** Anleitung zur Montage der Grundplatte, Räder und Motoren des Autos, um eine korrekte Einrichtung zu gewährleisten.
- **Grundlagen der Gleichstrommotoren:** Erläuterung des Funktionsprinzips von Gleichstrommotoren und ihrer Rolle in der Funktionalität des Smart Cars.
- **Verkabelung des Motortreibers:** Detaillierte Anleitung zur Verkabelung des L298N Dual Full Bridge Motor Drivers für optimale Steuerung.
- **Test der Motorverkabelung:** Durchführung eines grundlegenden Tests zur Überprüfung der korrekten Verkabelung und Funktionalität der Motoren.
- **Bewegungs demonstration:** Vorführung der grundlegenden Bewegungsfähigkeiten des Autos, Demonstration einer erfolgreichen Montage und Verkabelung.

Video

Verwandte Online-Tutorials

- [Auto zusammenbauen](#)
- [1. Bewegung](#)
- [2. Bewegung durch Code](#)

3.24 Video 24: Steuerung eines Smart Cars

Dieses Tutorial leitet die Zuschauer an, wie man ein Smart Car mit Arduino steuert, mit Schwerpunkt auf Richtungs-
bewegungen, Anhalten und Geschwindigkeitsanpassungen.

- **Grundlegende Auto-Bewegungen:** Einführung in die Programmierung des Smart Cars für grundlegende Bewegungen wie vorwärts, rückwärts, links, rechts und stoppen.
- **Einrichtung und Konfiguration der Verkabelung:** Schritt-für-Schritt-Anleitung zum Aufbau der Verkabelung zwischen dem Arduino und dem Smart Car.
- **Arduino-Code für Bewegung:** Detaillierte Erklärung des für die Richtungssteuerung des Smart Cars erforderlichen Arduino-Codes.
- **Techniken zur Geschwindigkeitskontrolle:** Techniken zur Programmierung des Autos für Geschwindigkeitsvariationen, einschließlich Beschleunigung und Verzögerung.
- **Praktische Bewegungsdemonstrationen:** Live-Demonstrationen, die die Umsetzung von Auto-Bewegungen und Geschwindigkeitskontrolle zeigen.

Video

Verwandte Online-Tutorials

- [*3. Beschleunigung*](#)

3.25 Video 25: Hindernisvermeidung mit dem Smart Car

Dieses Tutorial lehrt, wie man ein Smart Car mit Infrarotsensoren zur Hindernisvermeidung programmiert, Teil des Robojax Arduino-Kurses mit dem SunFounder-Kit.

- **Grundlagen der Hindernisvermeidung:** Überblick über die Verwendung von Infrarotsensoren zur Erkennung und Vermeidung von Hindernissen bei Smart Cars.
- **Erforderliches Wissen:** Betonung des notwendigen Vorwissens zur Steuerung der Auto-Bewegungen und der Grundlagen von Infrarotsensoren.
- **Sensoraufbau und -ausrichtung:** Details zur korrekten Ausrichtung der Sensoren für eine optimale Hinderniserkennung.
- **Verdrahtung für die Hinderniserkennung:** Schritt-für-Schritt-Anleitung zum Anschließen der Sensoren an den Arduino und das Auto.
- **Code-Durchlauf für die Hindernisvermeidung:** Umfassende Erklärung des Arduino-Codes zur Hinderniserkennung und -reaktion.
- **Live-Demonstration der Hindernisvermeidung:** Echtzeitdemonstration, die zeigt, wie das Smart Car Hindernisse umfährt.

Video

Verwandte Online-Tutorials

- [*5. Mit dem Hindernisvermeidungs-Modul spielen*](#)

3.26 Video 26: Smart Car Ultraschallsensor

In diesem Tutorial lernen Sie, wie man ein Smart Car mit einem Arduino und Ultraschallsensoren für fortgeschrittene Hinderniserkennung und Navigationsfunktionen programmiert.

- **Einführung in Ultraschallsensoren:** Erkundung der Fähigkeiten von Ultraschallsensoren in der Navigation von Smart Cars.
- **Anleitungen zur Sensorverkabelung:** Schritt-für-Schritt-Anleitung zum Anschließen des Ultraschallsensors an das Arduino-Board.
- **Aufschlüsselung des Arduino-Codes:** Ausführliche Erklärung des Codes, der zur Verarbeitung von Sensordaten und zur Steuerung des Autos verwendet wird.
- **Motorsteuerung über Sensordaten:** Wie Ultraschallsensormessungen zur Steuerung der Motoren des Autos verwendet werden.

Video

Verwandte Online-Tutorials

- *6. Spielen mit dem Ultraschallmodul*

3.27 Video 27: Handverfolgung mit dem Smart Car

In diesem Tutorial lernen Sie, wie man ein Smart Car programmiert, um mit Arduino, Infrarot- und Ultraschallsensoren Handbewegungen zu folgen.

- **Handverfolgungs-Funktion des Smart Cars:** Überblick über die Fähigkeit des Autos, Handbewegungen zu verfolgen und zu folgen.
- **Sensorintegration zur Erkennung:** Nutzung von Infrarot- und Ultraschallsensoren zur Erkennung der Position und Entfernung der Hand.
- **Detaillierte Verkabelungsanleitung:** Schritt-für-Schritt-Anweisungen zur korrekten Verkabelung der Sensoren an das Arduino-Board.
- **Umfassender Code-Durchlauf:** Erklärung des Arduino-Codes zur Verarbeitung von Sensoreingaben und zur Steuerung des Autos.
- **Reaktive Motorsteuerung:** Anpassung der Motorreaktionen des Autos basierend auf Sensordaten für präzise Bewegungen.
- **Code-Optimierung für Genauigkeit:** Feinabstimmung des Codes, um zuverlässige und konsistente Sensordaten zu gewährleisten.

Video

Verwandte Online-Tutorials

- *7. Folge Deiner Hand*

3.28 Video 28: Selbstfahrendes Smart Car

Lernen Sie, wie man mit Arduino ein einfaches selbstfahrendes Auto erstellt, komplett mit Ultraschall- und Infrarotsensoren für Navigation und Hindernisvermeidung.

- **Projekteinführung:** Überblick über die Erstellung eines selbstfahrenden Autos mit Arduino.
- **Sensorintegration:** Verwendung von Ultraschall- und Infrarotsensoren für Navigation und Hinderniserkennung.
- **Detaillierte Code-Erklärung:** Erläuterung des Arduino-Codes, der die Bewegungen des Autos in Reaktion auf Sensordaten steuert.
- **Motor- und Sensor-Setup:** Konfiguration der Arduino-Pins für Motorsteuerung und Sensoreingänge.
- **Bewegungslogik per Code:** Implementierung bedingter Anweisungen im Code für intelligente Auto-Bewegungen.
- **Test der Code-Funktionalität:** Beobachtung der Reaktion des Autos auf Hindernisse und seiner Navigationsfähigkeiten über den seriellen Monitor.

Video

Verwandte Online-Tutorials

- [8. Autonomes Fahren](#)

3.29 Video 29: Fernsteuerung des Smart Cars

Dieses Tutorial lehrt, wie man eine Infrarot-Fernbedienung zur Steuerung eines Arduino Smart Cars verwendet, einschließlich Bewegungsrichtungen und Geschwindigkeitsanpassungen.

- **Einführung in die Fernsteuerung:** Verständnis der Grundlagen zur Steuerung eines Arduino-Autos mit einer Infrarot-Fernbedienung.
- **Funktionen der Fernbedienungstasten:** Detaillierte Erklärung der Funktion jeder Taste auf der Fernbedienung für die Autosteuerung.
- **Verkabelungsanleitung:** Schritt-für-Schritt-Einrichtung für den Infrarotempfänger und eine zusätzliche LED.
- **Code-Erklärung und Demonstration:** Ausführlicher Durchgang durch den Arduino-Code und seine Auswirkungen auf die Autosteuerung.

Video

Verwandte Online-Tutorials

- [9. Fernbedienung](#)

3.30 Video 30: Linienverfolgung mit dem Smart Car

Dieses Tutorial zeigt, wie man ein Arduino Smart Car für die Linienverfolgung mit Infrarotsensoren programmiert.

- **Einführung in die Linienverfolgung:** Verständnis der Grundlagen der Linienverfolgung bei Arduino Smart Cars.
- **Details zum Linienverfolgungssensor:** Einblicke in das Infrarot-Sender- und Empfängermodul, das für die Linienverfolgung verwendet wird.
- **Sensoranpassung für Genauigkeit:** Tipps zur Einstellung des Sensors für präzise Linien-Erkennung.

- **Durchgang durch den Arduino-Code:** Detaillierte Erklärung des Codes, der die Linienverfolgungsfähigkeit des Autos steuert.
- **Demonstration der Sensorenwerte:** Wie Arduino mit dem Sensor zwischen schwarzen und weißen Oberflächen unterscheidet.
- **Praktische Demonstration der Linienverfolgung:** Vorführung der praktischen Anwendung der Linienverfolgung in einem Arduino-Auto.

Video

Verwandte Online-Tutorials

- [*4. Die Linie verfolgen*](#)

3.31 Video 31: WLAN-Temperaturüberwachung

Ein umfassender Leitfaden zur Einrichtung des ESP8266 ESP-01 mit einem DHT11-Sensor zur Temperaturüberwachung über WLAN und deren Anzeige auf verschiedenen Geräten.

- **Einführung in ESP8266 und DHT11:** Erkundung der Komponenten, die für die WLAN-basierte Temperaturüberwachung verwendet werden.
- **Grundlagen des Web-Servers:** Verständnis der Funktionsweise von Web-Servern im Kontext des Fernzugriffs auf Daten.
- **Arduino IDE-Einrichtung für ESP8266:** Konfiguration der Arduino IDE zur Programmierung des ESP8266-Moduls.
- **Sensorverkabelungsanleitung:** Detaillierte Anleitung zum Anschließen des DHT11-Sensors an den ESP8266.
- **Temperaturdaten auf dem seriellen Monitor:** Programmierung des ESP8266 zur Anzeige von Temperaturmessungen auf einem seriellen Monitor.
- **Fernanzeige von Temperaturdaten:** Demonstration, wie man Temperaturdaten auf mobilen und Desktop-Geräten über WLAN anzeigt.

Video

Verwandte Online-Tutorials

- [*5. Überwachung des häuslichen Umfelds*](#)

KAPITEL 4

Code herunterladen¶

Laden Sie den entsprechenden Code über den untenstehenden Link herunter.

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

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

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

- *1. Digitaler Schreibvorgang*: Setzen Sie die Ausgangsspannung des Pins auf Hoch oder Niedrig, um das Licht ein- und auszuschalten.
- *2. Analoges Schreiben*: Schreiben Sie den analogen Wert (**PWM-Welle**) auf den Pin, um die Helligkeit des Lichts anzupassen.
- *3. Digitaler Lesevorgang*: Lesen Sie das Pegelsignal des digitalen Pins, um den Arbeitszustand des Schalters zu ermitteln.
- *4. Analoges Auslesen*: Lesen Sie die Spannung des analogen Pins, um den Arbeitszustand des Drehknopfs zu ermitteln.

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

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

5.1 1. Digitaler Schreibvorgang

Digitaler Schreibvorgang dient dazu, ein digitales Signal an einen digitalen Pin auszugeben oder zu schreiben. Das digitale Signal hat nur zwei Zustände, 0 oder 1, 0V oder 5V, so dass einige Komponenten, wie die LED und der Summer, ein- oder ausgeschaltet werden können.

Auf dem Arduino R3-Board gibt es 14 digitale I/O-Pins von 0 bis 13. Verwenden Sie nun die Funktionen `pinMode()` und `digitalWrite()` um diesen digitalen Pins ein hohes oder niedriges Signal zu schreiben.

- `pinMode(pin, mode)`: Konfiguriert den spezifischen 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 gesetzt werden soll.
- mode: INPUT, OUTPUT oder INPUT_PULLUP.
- digitalWrite(pin, value): Schreiben Sie ein hohes Signal (5V) oder ein niedriges Signal (0V) auf 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 gesetzt: 5V (oder 3,3V auf 3,3V-Boards) für HIGH, 0V (Erdung) für LOW.

Syntax

digitalWrite(pin, value)

Parameter

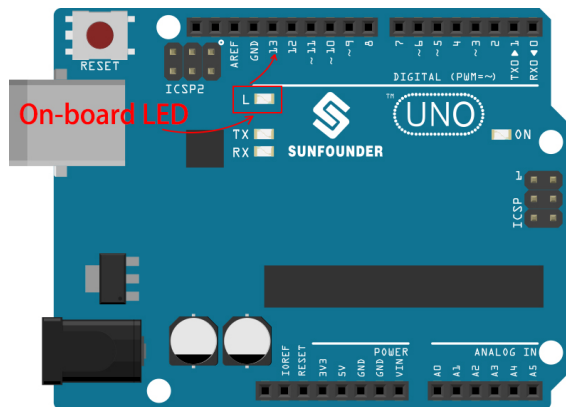
- pin: Die Pin-Nummer von Arduino.
- value: HIGH oder LOW.

Beispiel für Digitaler Schreibvorgang:

```
const int pin = 13;

void setup() {
  pinMode(pin, OUTPUT);    // setzt den digitalen Pin als Ausgang
}

void loop() {
  digitalWrite(pin, HIGH); // schaltet den digitalen Pin ein
  delay(1000);             // wartet eine Sekunde
  digitalWrite(pin, LOW);  // schaltet den digitalen Pin aus
  delay(1000);             // wartet eine Sekunde
}
```



Anmerkungen 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. Ein Anschluss an diese Pins stört die Kommunikation, einschließlich der Verhinderung des Hochladens auf das Board.
- Wenn die digitalen Pins aufgebraucht sind, können die analogen Pins (A0-A5) auch als digitale Pins verwendet werden.

Zugehörige Komponenten

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

5.1.1 1.1 Hallo, LED!

Genau wie das Ausdrucken von „Hello, world!“ der erste Schritt beim Erlernen der Programmierung ist, so ist die Anwendung eines Programms zum Steuern einer LED die traditionelle Einführung in die physische Programmierung.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

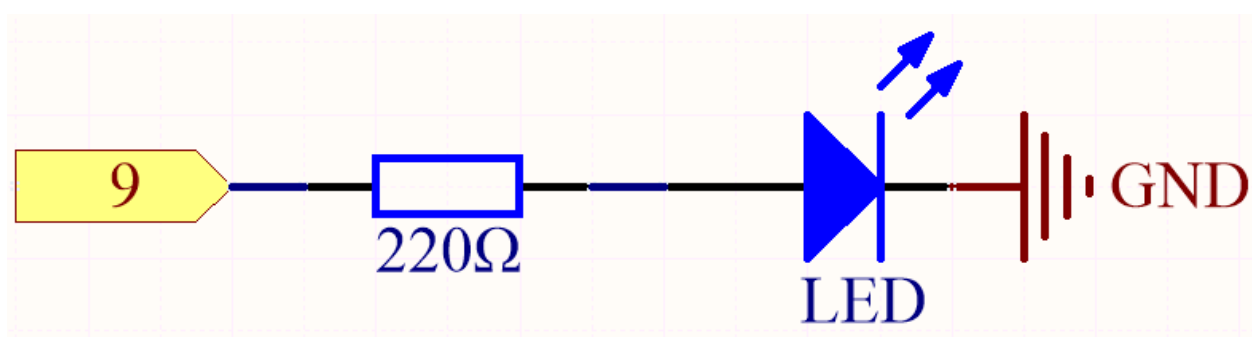
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

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

Sie können diese auch einzeln über die untenstehenden Links kaufen.

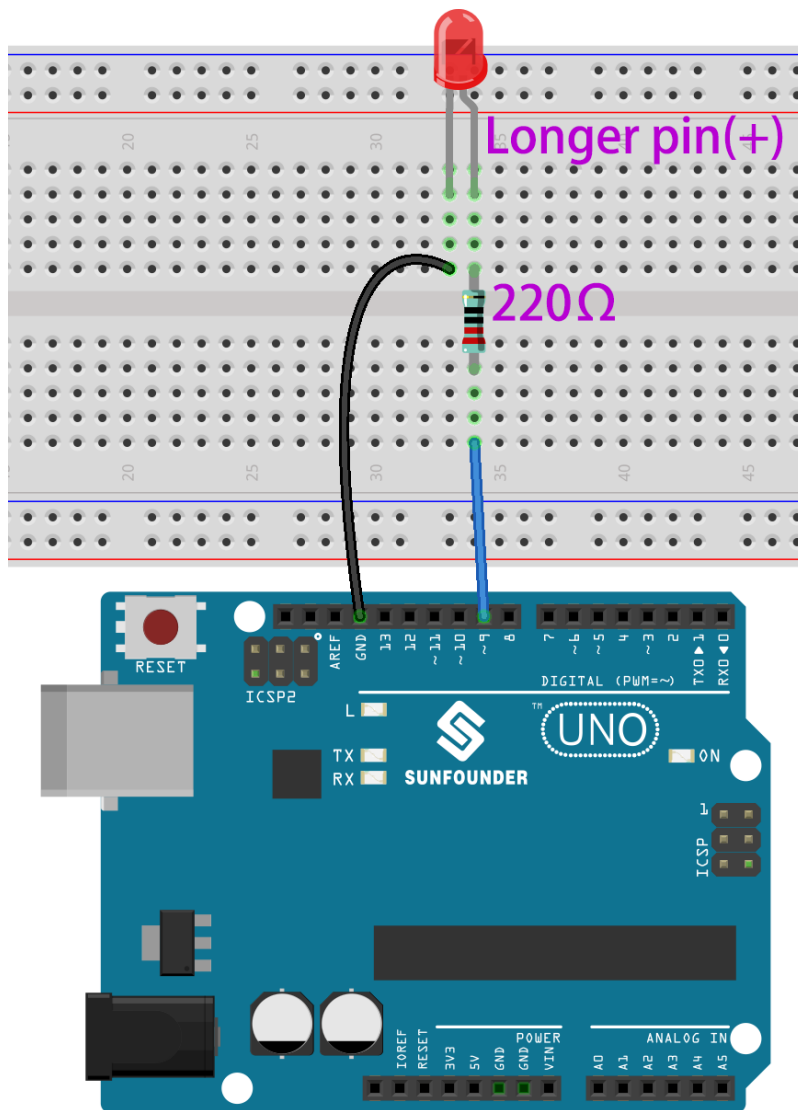
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	

Schaltplan



Das Prinzip dieser Schaltung ist einfach und die Stromrichtung wird in der Abbildung angezeigt. Wenn Pin 9 ein hohes Signal (5V) ausgibt, leuchtet die LED nach dem 220-Ohm-Strombegrenzungswiderstand auf. Wenn Pin 9 ein niedriges Signal (0V) ausgibt, schaltet sich die LED aus.

Verdrahtung



Code

Bemerkung:

- Sie können die Datei 1.1.hello_led.ino unter dem Pfad 3in1-kit\basic_project\1.1.hello_led öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nach erfolgreichem Hochladen des Codes sehen Sie die LED blinken.

Wie funktioniert das?

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

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

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

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

In `loop()`, wird `digitalWrite()` verwendet, um ein 5V-High-Level-Signal für `ledPin` bereitzustellen, was einen Spannungsunterschied zwischen den LED-Pins verursacht und die LED einschaltet.

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

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

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

Ein Intervall zwischen Ein- und Ausschalten ist erforderlich, damit Menschen die Änderung sehen können, deshalb verwenden wir einen `delay(1000)`-Code, um den Controller 1000 ms lang nichts tun zu lassen.

```
delay(1000);
```

5.1.2 1.2 Piepton

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

Zwei Arten von Summern sind im Kit enthalten. Wir müssen den aktiven Summer verwenden. Drehen Sie sie um, die versiegelte Rückseite (nicht die freiliegende PCB) ist die, die wir wollen.



Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

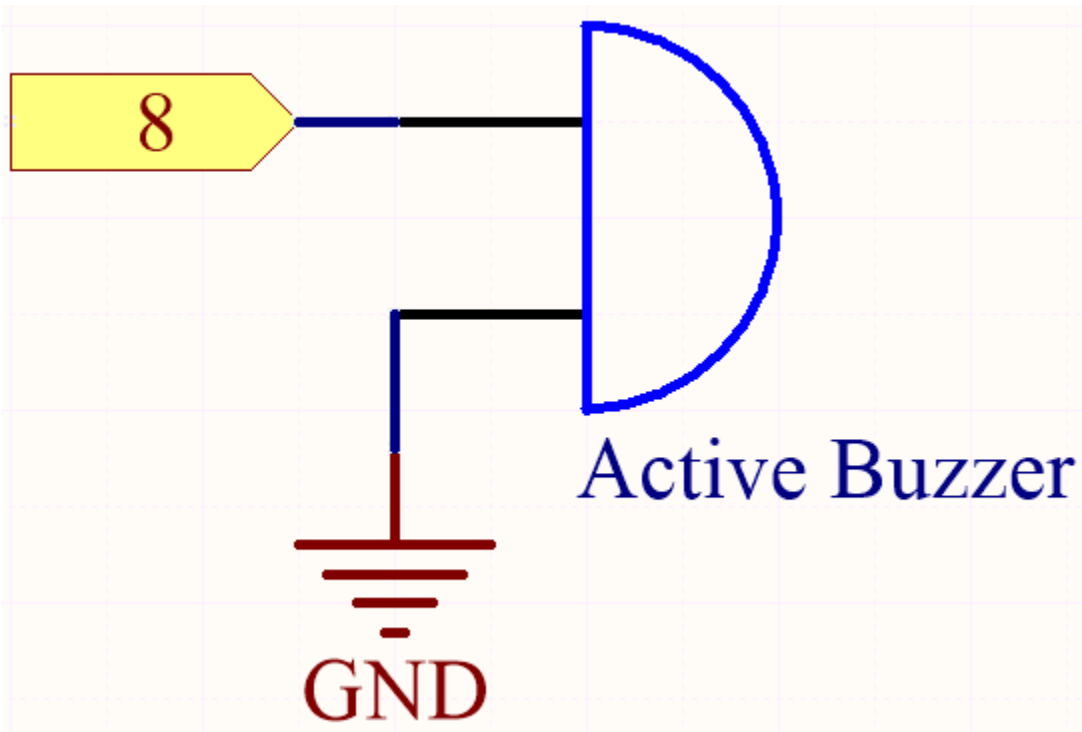
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

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

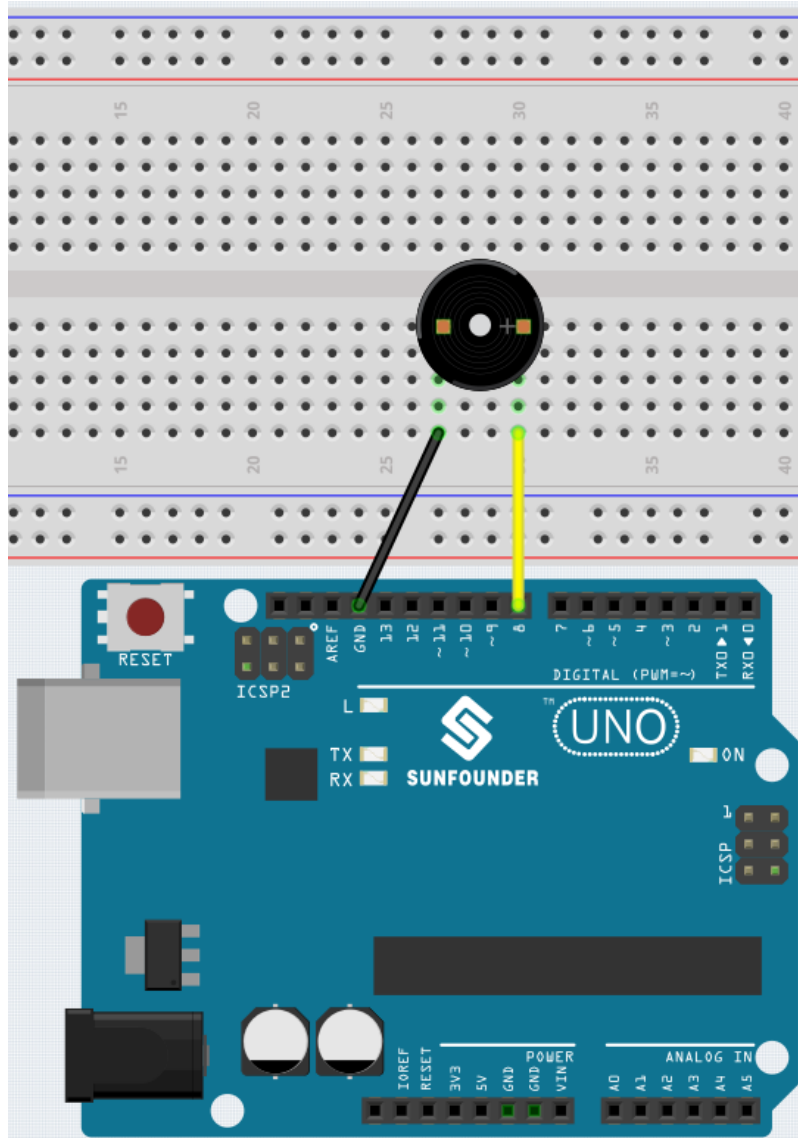
Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</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\basic_project\1.2.beep` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nach erfolgreichem Hochladen des Codes hören Sie alle Sekunde einen Piepton.

5.1.3 1.3 Das Rad drehen

Ein Motor ist ein typisches digitales Ausgabegerät und wird genauso verwendet wie eine LED. Allerdings benötigt der Motor einen großen Strom, und der hohe Strom kann das Hauptsteuerungsboard wie z.B. das R3-Board beschädigen. Deshalb wird in diesem Fall ein L298N-Modul verwendet, welches ein guter Helfer für das R3-Board ist, um den Motor sicher zu steuern.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

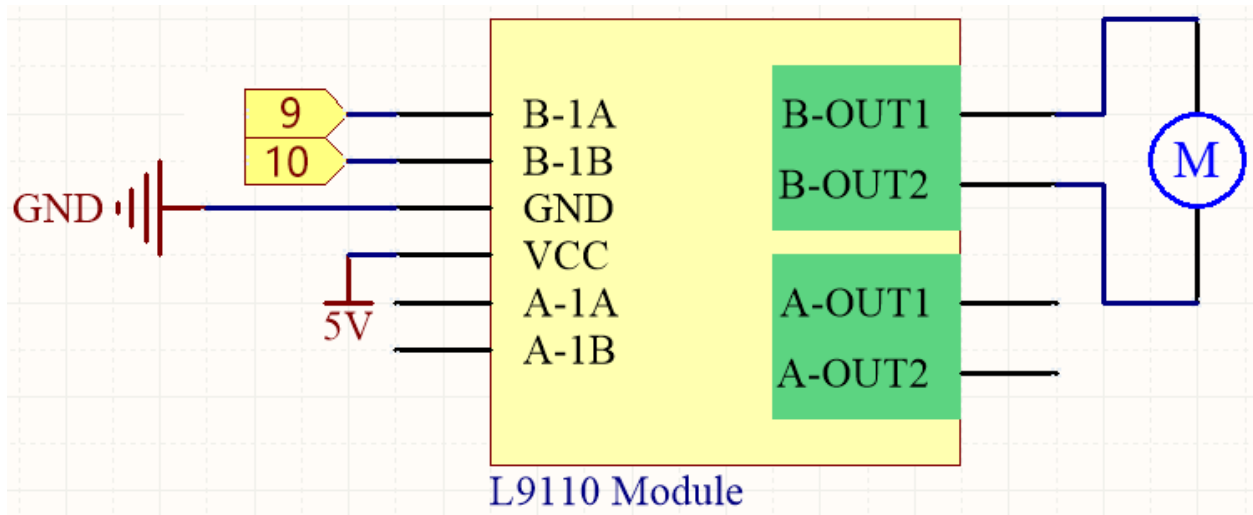
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

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

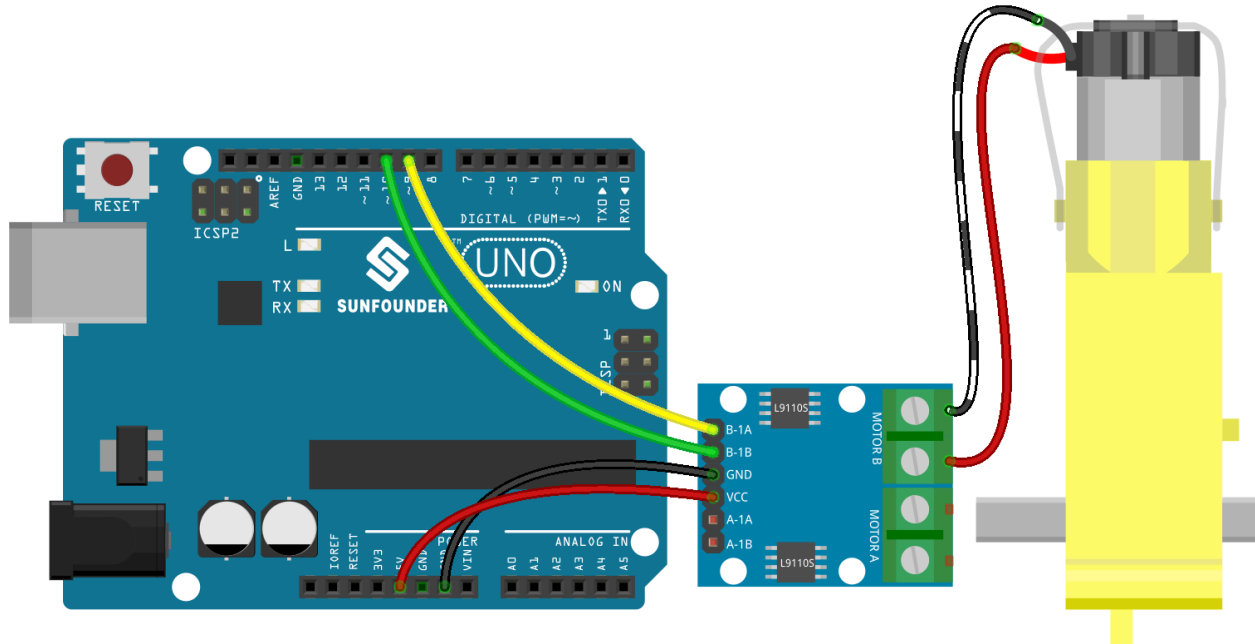
Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>TT Motor</i>	-
<i>L9110 Motortreibermodul</i>	-

Schaltplan



Verdrahtung



Code

Bemerkung:

- Sie können die Datei 1.3.turn_the_wheel.ino unter dem Pfad 3in1-kit\basic_project\1.3.turn_the_wheel öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

5.1.4 1.4 Pumpen

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

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

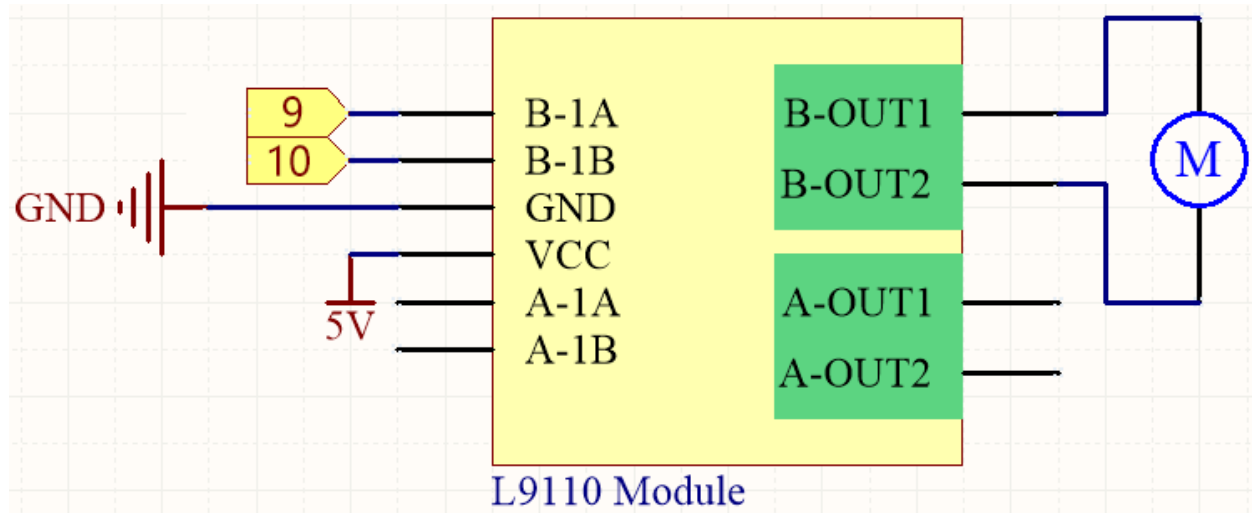
Es ist sicherlich praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

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

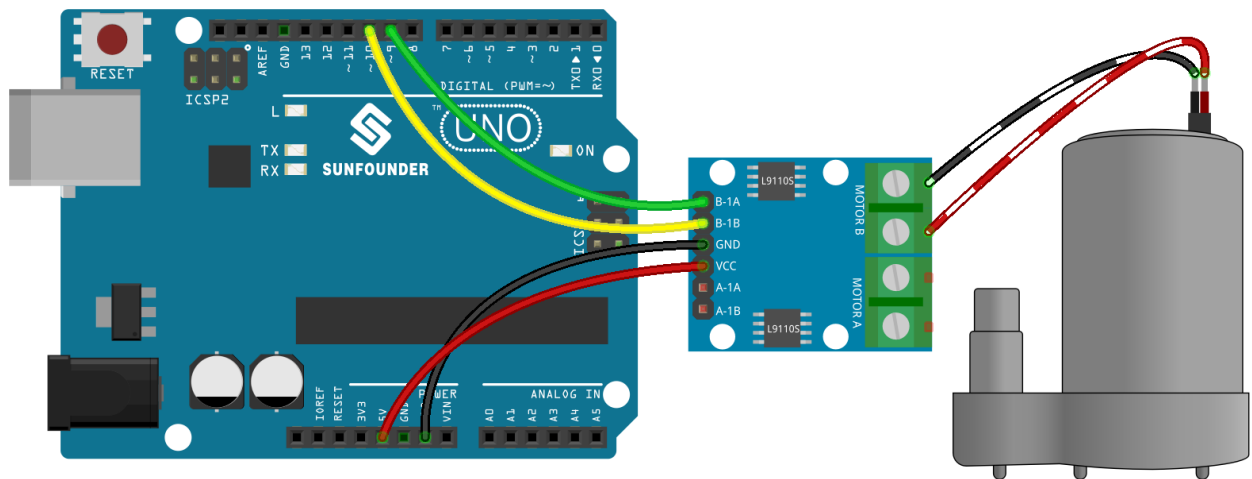
Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>L9110 Motortreibermodul</i>	-
<i>Kreiselpumpe</i>	-

Schaltplan



Verdrahtung



Code

Bemerkung:

- Sie können die Datei 1.4.pumping.ino unter dem Pfad 3in1-kit/basic_project/1.4.pumping öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

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

5.2 2. Analoges Schreiben

6 der 14 digitalen Pins von Arduino verfügen auch über eine PWM-Ausgabefunktion. Daher können Sie neben dem Schreiben von digitalen Signalen an diese 6 Pins auch analoge Signale (PWM-Wellensignale) an sie senden. Auf diese Weise können Sie die LEDs in unterschiedlicher Helligkeit 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 die wörtliche Bedeutung schwer zu fassen sein mag, folgt hier ein Beispiel zur Steuerung der Intensität einer LED, um das Prinzip 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 für eine kurze Zeit (wie 20ms, die visuelle Verweildauer der meisten Menschen) ein-, aus- und wieder einschalten, werden wir nicht bemerken, dass sie ausgeschaltet war, aber die Helligkeit des Lichts wird etwas schwächer sein. In diesem Zeitraum gilt: Je länger die LED eingeschaltet ist, desto heller wird sie sein. Das heißt, innerhalb eines Zeitraums, je breiter der Puls, desto größer ist die „elektrische Signalstärke“, die vom Mikrocontroller ausgegeben wird.

Dies ist die benötigte Funktion, um die PWM-Welle zu schreiben:

- `analogWrite(pin, value)`

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

Syntax

`analogWrite(pin, value)`

Parameter

- `pin`: Der Arduino-Pin, an den geschrieben wird. Erlaubte Datentypen: `int`.
- `value`: Das Tastverhältnis: zwischen 0 (immer aus) und 255 (immer an). Erlaubte Datentypen: `int`.

Beispiel für Analoges Schreiben

```
int pin = 9;           // Verbindung zum PWM-Pin

void setup() {
  pinMode(pin, OUTPUT); // Definiert den Pin als Ausgang
}

void loop() {
  for (int i = 0 ; i<255 ; i++){
    analogWrite(pin, i); //analogWrite Werte von 0 bis 255
    delay(30);
  }
}
```

Anmerkungen und Warnungen

- Wenn Sie genau auf das R3-Board schauen, haben die mit dem Symbol „~“ gekennzeichneten Pins eine analoge Ausgabefunktion.
- Die an den Pins 5 und 6 erzeugten PWM-Ausgänge haben höhere Tastverhältnisse als erwartet. Dies liegt an Wechselwirkungen mit den Funktionen `millis()` und `delay()`, die denselben internen Timer verwenden, der zum Erzeugen dieser PWM-Ausgänge genutzt wird. Dies wird vor allem bei niedrigen Tastverhältniseinstellungen (z. B. 0 - 10) bemerkbar und kann dazu führen, dass ein Wert von 0 den Ausgang an den Pins 5 und 6 nicht vollständig abschaltet.

Zugehörige Komponenten

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

5.2.1 2.1 Überblenden

Dieses Projekt ähnelt [1.1 Hallo, LED!](#). Der Unterschied liegt in der Signalart. Während das frühere Projekt die LED durch Ausgabe eines digitalen Signals (0&1) ein- oder ausschaltet, steuert dieses Projekt die Helligkeit der LED durch Ausgabe eines analogen Signals.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

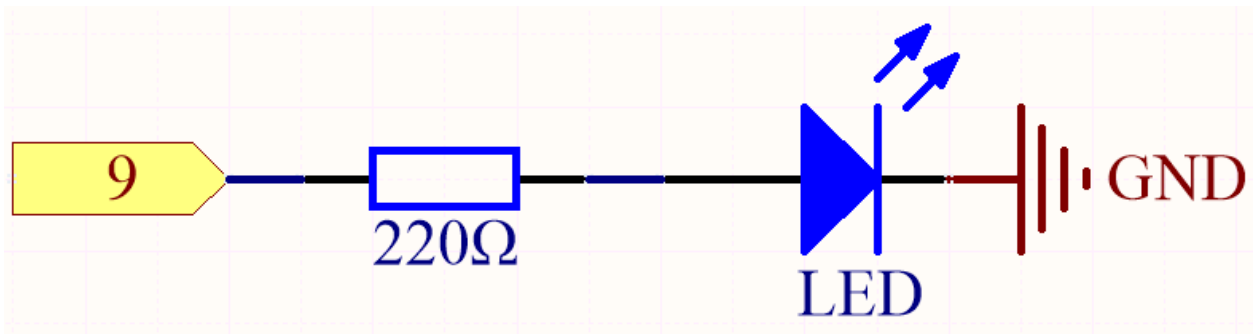
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

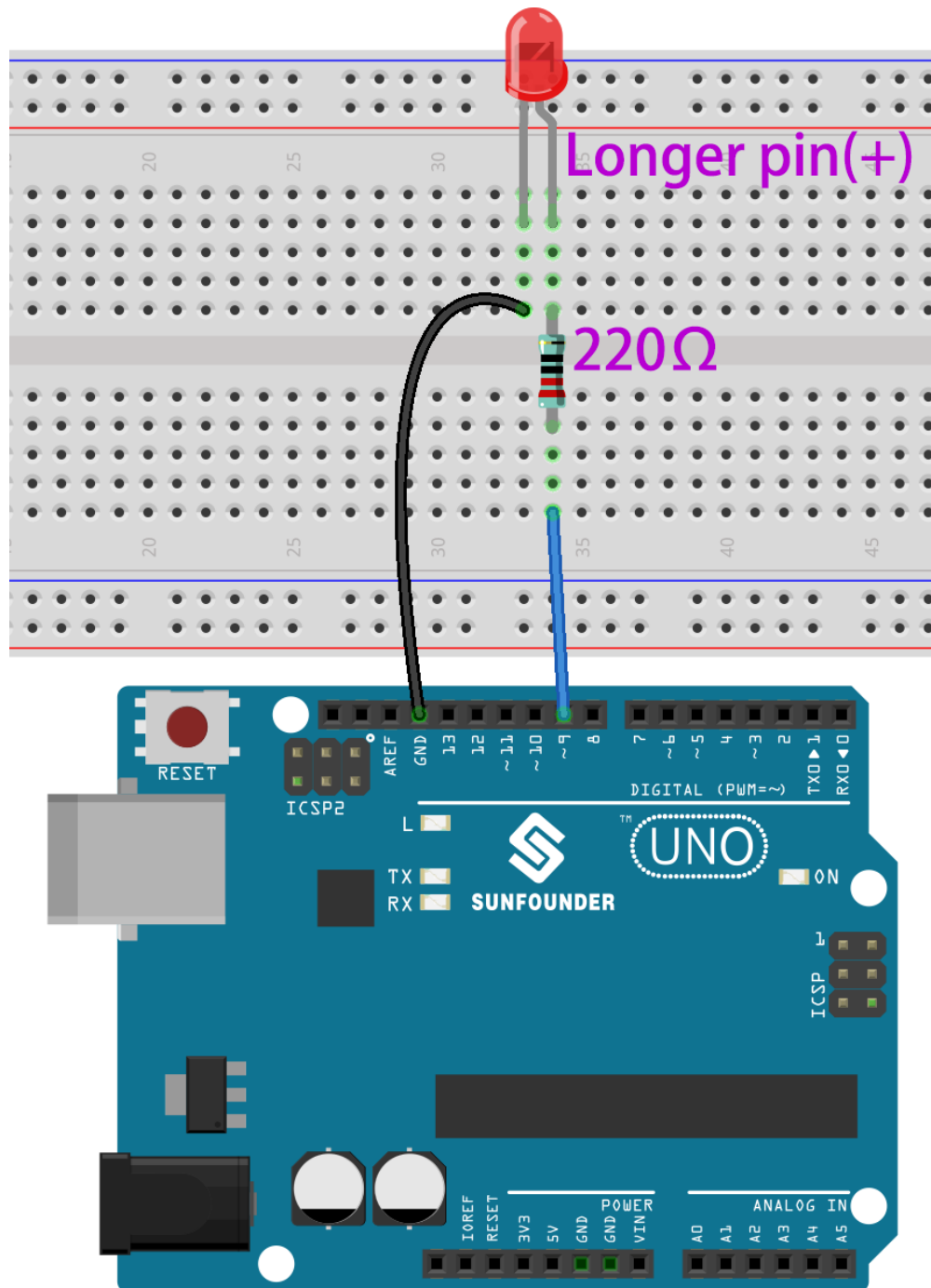
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</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\basic_project\2.analogWrite\2.1.fading öffnen.
- 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, können Sie sehen, wie die LED atmet.

5.2.2 2.2 Buntres Licht

Wie wir wissen, kann Licht überlagert werden. Beispielsweise ergibt die Kombination von blauem und grünem Licht ein türkises Licht und die von rotem und grünem Licht ein gelbes Licht. Dies wird als „Additive Farbmischung“ bezeichnet.

- [Additive Farbe - Wikipedia](#)

Auf Basis dieser Methode können wir durch Mischen der drei Grundfarben nach verschiedenen spezifischen Gewich-tungen jedes sichtbare Licht erzeugen. Beispielsweise entsteht Orange aus viel Rot und wenig Grün.

In diesem Kapitel werden wir die Geheimnisse der additiven Farbmischung mit einer RGB-LED erkunden!

Eine RGB-LED entspricht dem Kapseln einer roten, einer grünen und einer blauen LED unter einer Lampe, wobei die drei LEDs einen gemeinsamen Kathodenpin teilen. Da das elektrische Signal für jeden Anodenpin bereitgestellt wird, kann das Licht der entsprechenden Farbe angezeigt werden. Durch Änderung der elektrischen Signalintensität jeder Anode kann sie dazu gebracht werden, verschiedene Farben zu erzeugen.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

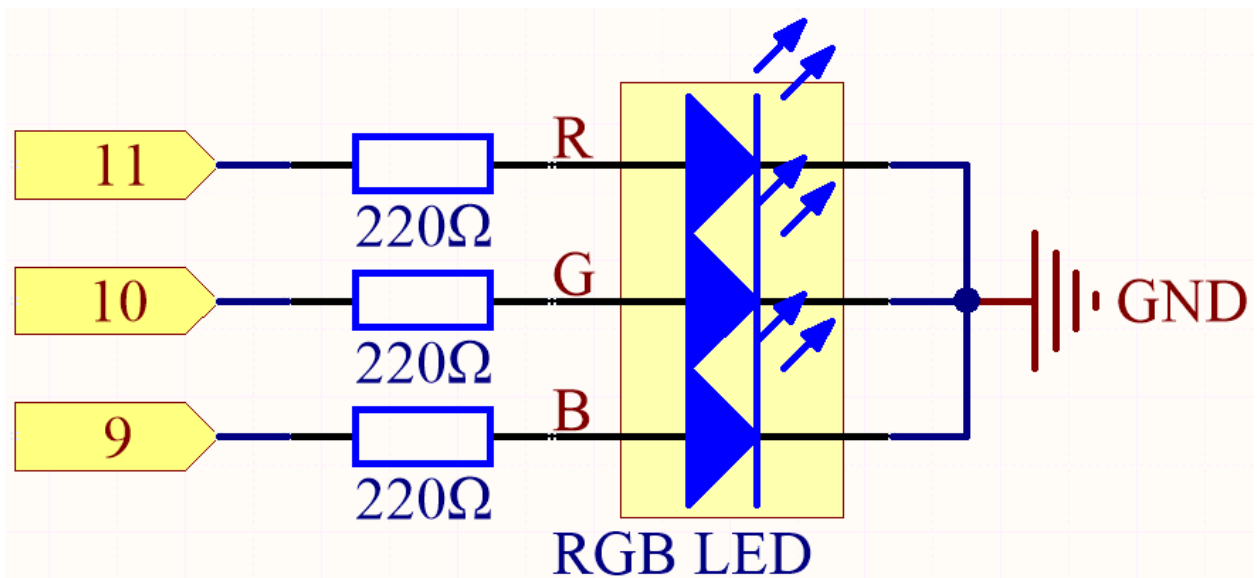
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

Sie können sie auch separat über die untenstehenden Links kaufen.

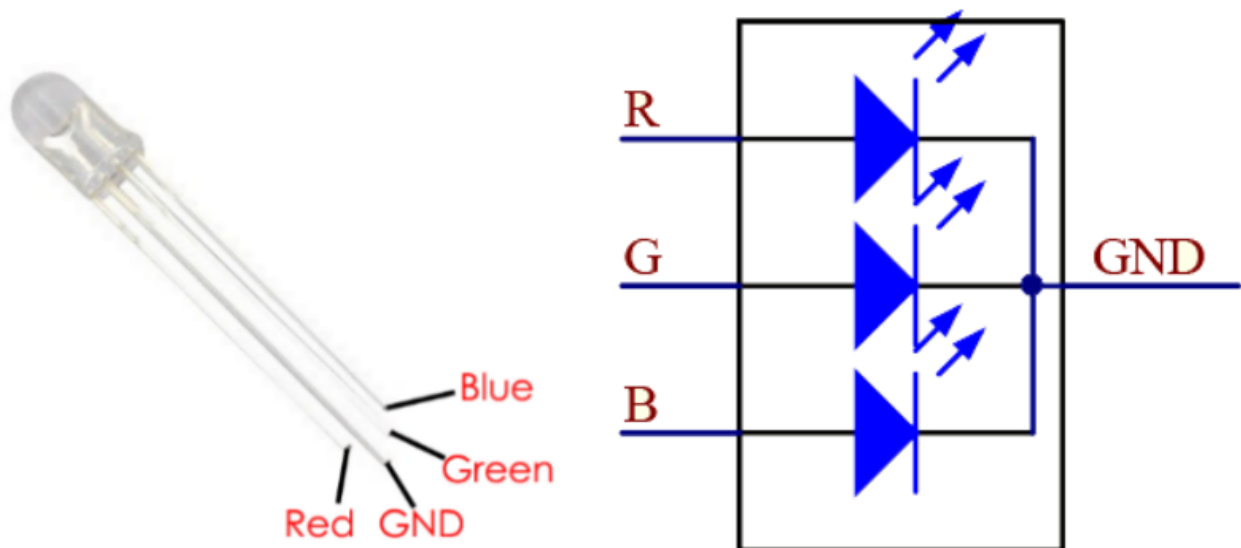
KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>RGB LED</i>	

Schaltplan

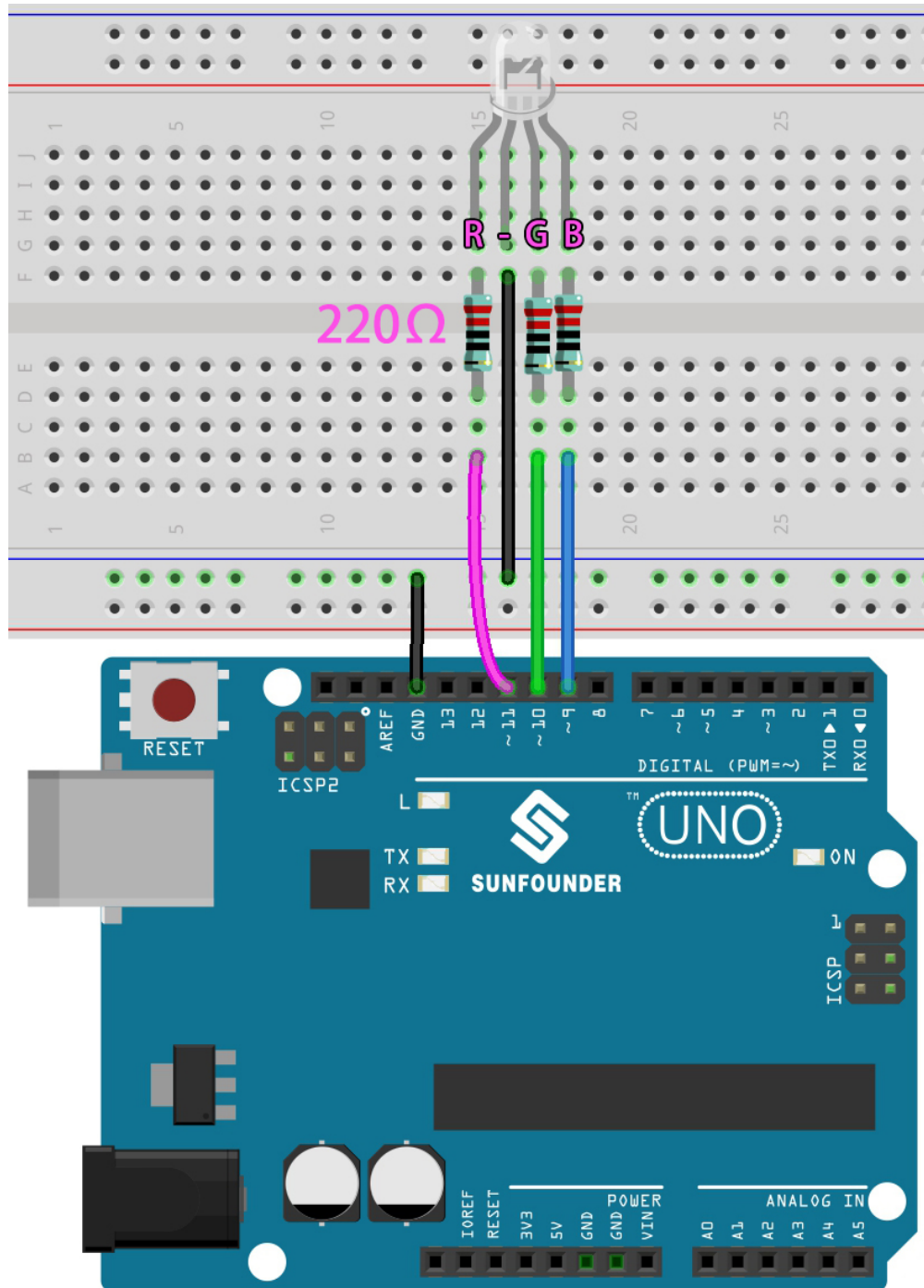


Die PWM-Pins 11, 10 und 9 steuern die Rot-, Grün- und Blau-Pins der RGB-LED jeweils an und verbinden den gemeinsamen Kathodenpin mit GND. Dadurch kann die RGB-LED durch Überlagerung von Licht auf diesen Pins mit verschiedenen PWM-Werten eine bestimmte Farbe anzeigen.

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.



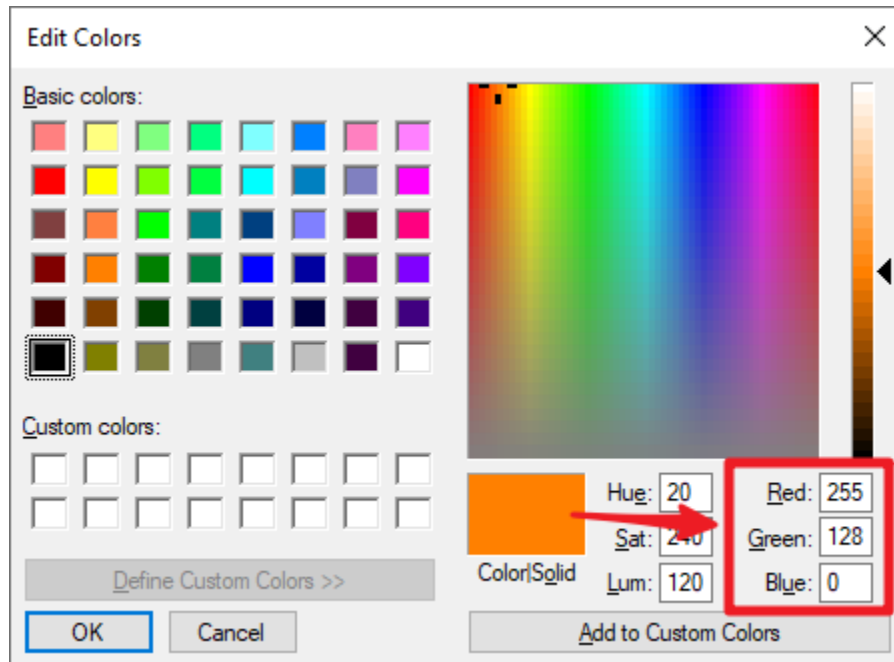
Code

Hier können wir unsere Lieblingsfarbe in 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/basic_project/2.analogWrite/2.2.colorful_light` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.



Schreiben Sie den RGB-Wert in `color_set()`, dann werden Sie sehen, wie die RGB-LED die gewünschten Farben anzeigt.

Wie funktioniert das?

In diesem Beispiel wird die Funktion, die Werte den drei Pins von RGB zuweist, 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()` dient der RGB-Wert als Eingabeargument, um die Funktion `color()` aufzurufen und somit die RGB-LED verschiedene Farben ausstrahlen zu lassen.

```
void loop()
{
    color(255, 0, 0); // rot
    delay(1000);
    color(0, 255, 0); // grün
    delay(1000);
    color(0, 0, 255); // blau
    delay(1000);
}
```

5.3 3. Digitaler Lesevorgang

Sensoren erfassen Informationen aus der realen Welt, welche über Pins (manche digital, manche analog) an das Hauptboard weitergegeben werden, sodass der Computer die tatsächliche Situation erkennen kann.

Daher kann das Arduino-Board den Arbeitsstatus von digitalen Sensoren erkennen, indem es den Wert digitaler Pins wie Tasten oder IR-Vermeidungsmodule liest.

Hier sind die benötigten Funktionen:

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

Syntax

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

Parameter

- `pin`: Die Arduino-Pin-Nummer, für die der Modus gesetzt werden soll.
- `mode`: INPUT, OUTPUT oder INPUT_PULLUP.

- `digitalRead(pin)`: Liest den Wert (Zustand) vom angegebenen digitalen Pin.

Syntax

```
digitalRead(pin)
```

Parameter

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

Rückgabewert

HIGH oder LOW

Beispiel für Digitalen Lesevorgang

```
int ledPin = 13; // LED verbunden mit digitalem Pin 13
int inPin = 7;   // Taster verbunden mit digitalem Pin 7
int val = 0;     // Variable zum Speichern des gelesenen Wertes

void setup() {
  pinMode(ledPin, OUTPUT); // setzt den digitalen Pin 13 als Ausgang
  pinMode(inPin, INPUT);   // setzt den digitalen Pin 7 als Eingang
}

void loop() {
  val = digitalRead(inPin); // liest den Eingangspin
  digitalWrite(ledPin, val); // setzt die LED auf den Wert des Tasters
}
```

Anmerkungen und Warnungen

1. Pull-Up & Pull-Down.

`digitalRead()` kann zufällige, unbestimmte Werte produzieren, wenn der Pin kein klares Signal erhält. Das Steuern der Eingangspins in einen bekannten Zustand kann das Projekt zuverlässiger machen. Bei der Verwendung einer Eingangskomponente wie einem Taster ist es normalerweise notwendig, einen Pull-Up- oder Pull-Down-Widerstand parallel zum digitalen Eingangspin anzuschließen.

Abgesehen vom Anschluss eines Pull-up-Widerstands können Sie auch den Pin-Modus im Code auf INPUT_PULLUP setzen, zum Beispiel `pinMode(pin, INPUT_PULLUP)`. In diesem Fall wird der Pin

über die Software auf den internen Pull-up-Widerstand von Atmega zugreifen, und es hat den gleichen Effekt wie das Anschließen eines Pull-up-Widerstands.

2. Über Pin13.

Alle digitalen Pins (1-13) auf dem R3-Board können mit `digitalRead()` verwendet werden. Aber der digitale Pin 13 ist schwieriger als Eingang zu verwenden als andere digitale Pins. Da er mit einer LED und einem Widerstand verbunden ist, ist er auf den meisten Boards verlötet. Wenn Sie seinen internen 20k Pull-up-Widerstand aktivieren, wird er statt der erwarteten 5V bei etwa 1,7V liegen, da die onboard LED und der Serienwiderstand das Spannungsniveau senken. Das bedeutet, er gibt immer LOW zurück. Wenn Sie Pin 13 als digitalen Eingang verwenden müssen, setzen Sie seinen `pinMode()` auf INPUT und verwenden Sie einen externen Pull-down-Widerstand.

3. Analoge Pins.

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

Verwandte Komponenten

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

5.3.1 3.0 Serieller Monitor

In der Arduino IDE gibt es einen seriellen Monitor, der es ermöglicht, Nachrichten von Ihrem Computer zum Arduino-Board (über USB) zu senden und auch Nachrichten vom Arduino zu empfangen.

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 zur Kommunikation mit dem Computer verwendet. Das Anschließen von Geräten an diese Pins kann die Kommunikation stören, einschließlich fehlgeschlagener Uploads zum Board.

Verwendung des seriellen Monitors

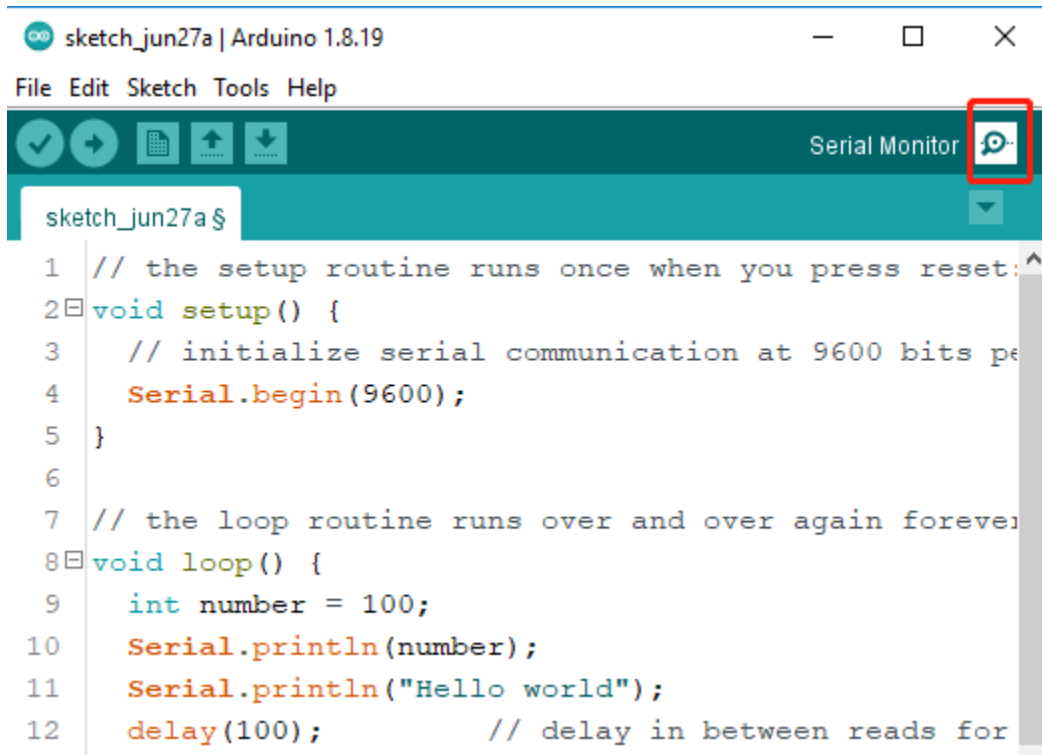
1. Öffnen Sie die Arduino IDE und fügen Sie den folgenden Code ein.

```
// Die Setup-Routine wird einmal ausgeführt, wenn Sie Reset drücken:
void setup() {
    // Initialisiere die serielle Kommunikation mit 9600 Baud:
    Serial.begin(9600);
}

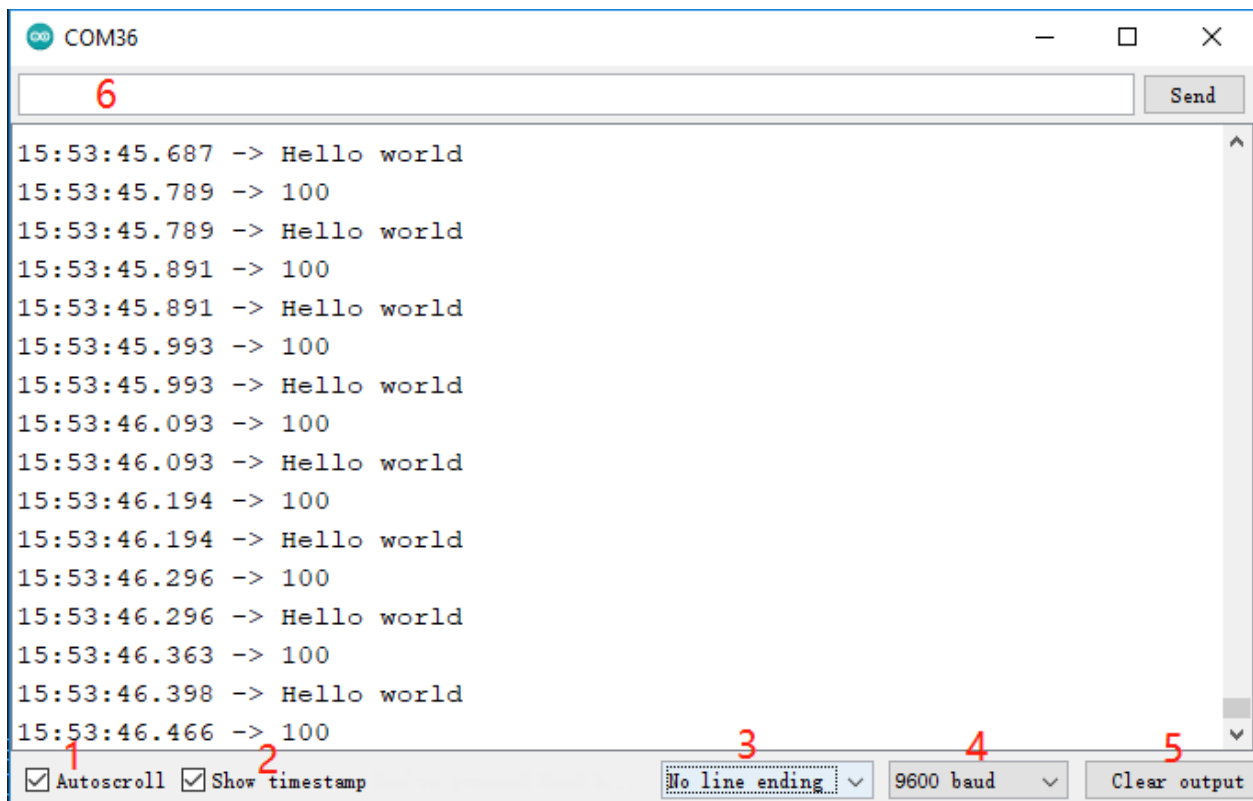
// Die Loop-Routine wird immer und immer wieder ausgeführt:
void loop() {
    int number = 100;
    Serial.println(number);
    Serial.println("Hello world");
    delay(100);           // Verzögerung zwischen den Abfragen für Stabilität
}
```

- `Serial.begin()`: Legt die Datenrate in Bits pro Sekunde (Baud) für die serielle Datenübertragung fest, hier auf 9600 eingestellt.
- `Serial.println()`: Sendet Daten zum seriellen Port als menschenlesbaren ASCII-Text, gefolgt von einem Wagenrücklauf (ASCII 13 oder `\r`) und einem Zeilenumbruch (ASCII 10 oder `\n`). Dieser Befehl entspricht `Serial.print()`.

2. Wählen Sie das richtige Board und den richtigen Port aus, um den Code hochzuladen.
3. Klicken Sie in der Symbolleiste auf das Lupensymbol, um den seriellen Monitor zu aktivieren.



4. Hier sehen Sie den seriellen Monitor.



- **1:** Option, um automatisch zu scrollen oder nicht zu scrollen.
- **2:** Option, um einen Zeitstempel vor den im seriellen Monitor angezeigten Daten anzuzeigen.
- **3:** Auswahl des Zeilenendes, wählt die Zeichen aus, die den an Arduino gesendeten Daten hinzugefügt werden. Auswahl beinhaltet:
 - **Kein Zeilenumbruch** sendet nur, was Sie eingeben;
 - **Zeilenumbruch** ist `\n` und sendet nach Ihrer Eingabe einen ASCII-Zeilenumbruchcode;
 - **Wagenrücklauf** ist `\r` und sendet nach Ihrer Eingabe ein ASCII-Wagenrücklaufzeichen;
 - **Beides, NL & CR** ist `\r\n` und sendet nach Ihrer Eingabe sowohl ein Wagenrücklauf- als auch ein Zeilenumbruchzeichen.
- **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 Ausgabekonsole.
- **6:** Ein Textfeld, um Zeichen an das Arduino-Board zu senden, siehe [5.12 Serielle Lektüre](#) für ein Tutorial.

5.3.2 3.1 Tastenwert auslesen

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

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

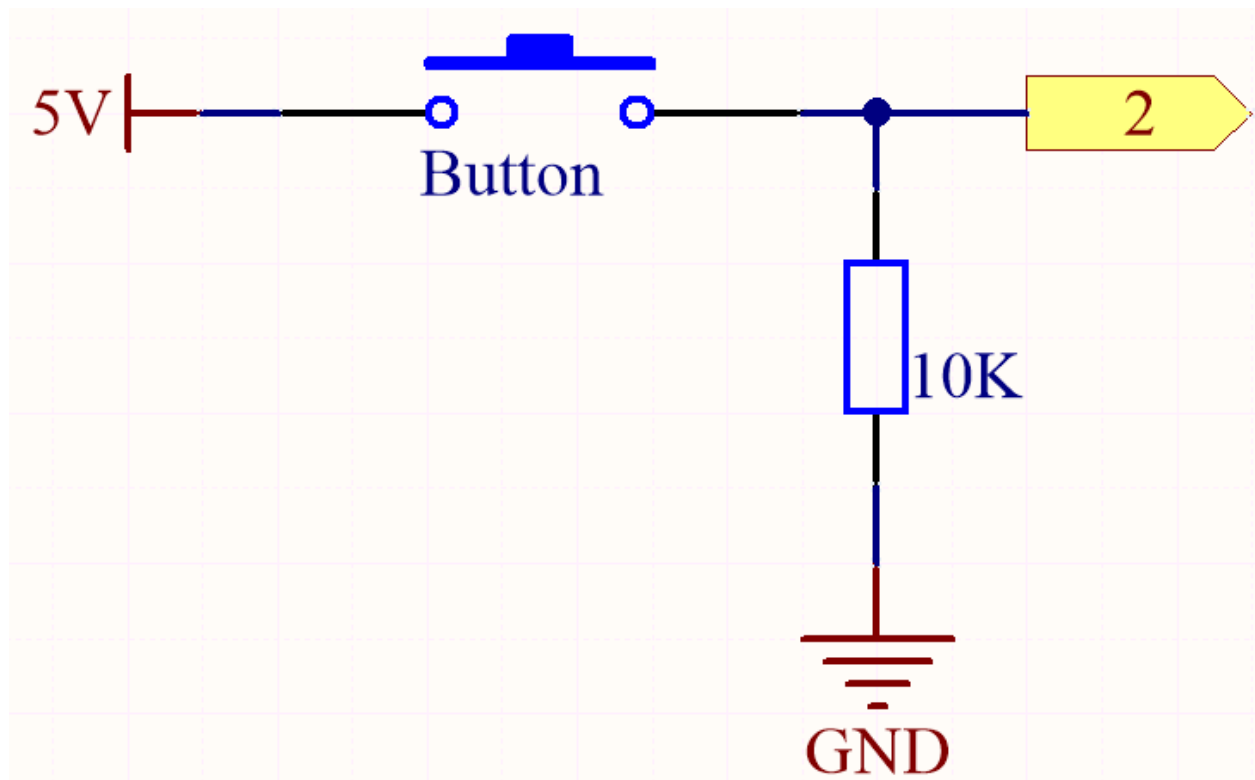
Es ist definitiv praktisch, ein ganzes Kit zu kaufen. Hier ist der Link:

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

Sie können diese auch separat über die untenstehenden Links kaufen.

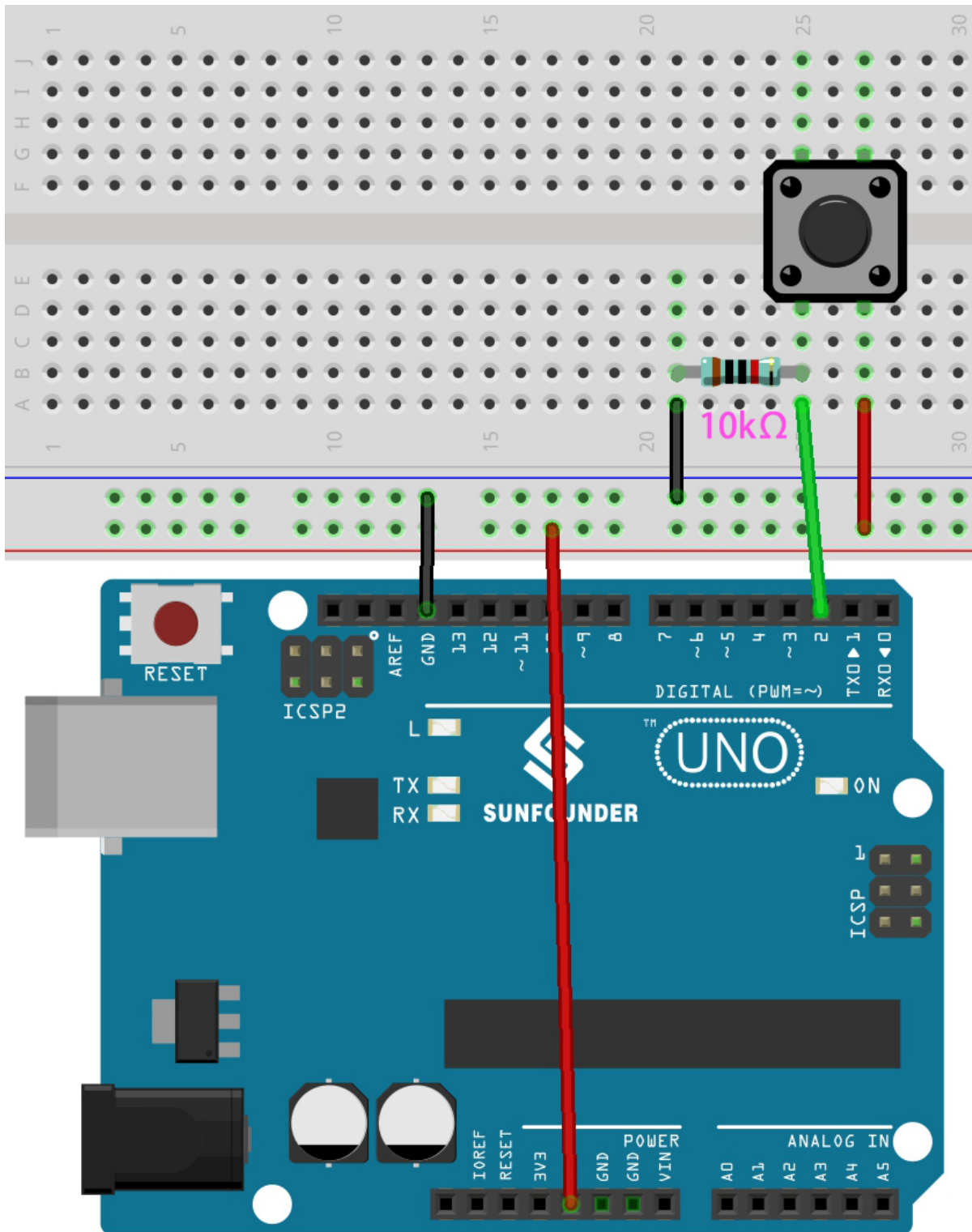
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Knopf</i>	

Schaltplan



Ein Pin der Taste ist mit 5V verbunden, der andere Pin mit Pin 2. Wenn die Taste gedrückt wird, wird Pin 2 hochgeschaltet. Wenn die Taste jedoch nicht gedrückt wird, befindet sich Pin 2 in einem Schwebezustand und kann hoch oder niedrig sein. Um ein stabiles niedriges Signal zu erhalten, wenn die Taste nicht gedrückt ist, muss Pin 2 über einen 10K Pull-down-Widerstand erneut mit GND verbunden werden.

Verkabelung



Code

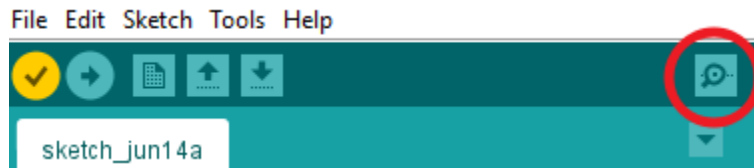
Bemerkung:

- Sie können die Datei 3.1.read_button_value.ino unter dem Pfad 3in1-kit\basic_project\3.1.

read_button_value öffnen.

- 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, klicken Sie auf das Lupensymbol in der oberen rechten Ecke der Arduino IDE (Serieller Monitor).



Wenn Sie die Taste drücken, wird im Seriellen Monitor „1“ angezeigt.

5.3.3 3.2 Magnetismus spüren

Der gebräuchlichste Typ eines Reed-Schalters enthält ein Paar magnetisierbare, flexible Metallzungen, deren Enden durch eine kleine Lücke getrennt sind, wenn der Schalter offen ist.

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

Ein gängiges Beispiel für eine Anwendung eines Reed-Schalters ist die Erkennung des Öffnens einer Tür oder eines Fensters für eine Sicherheitsalarmanlage.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

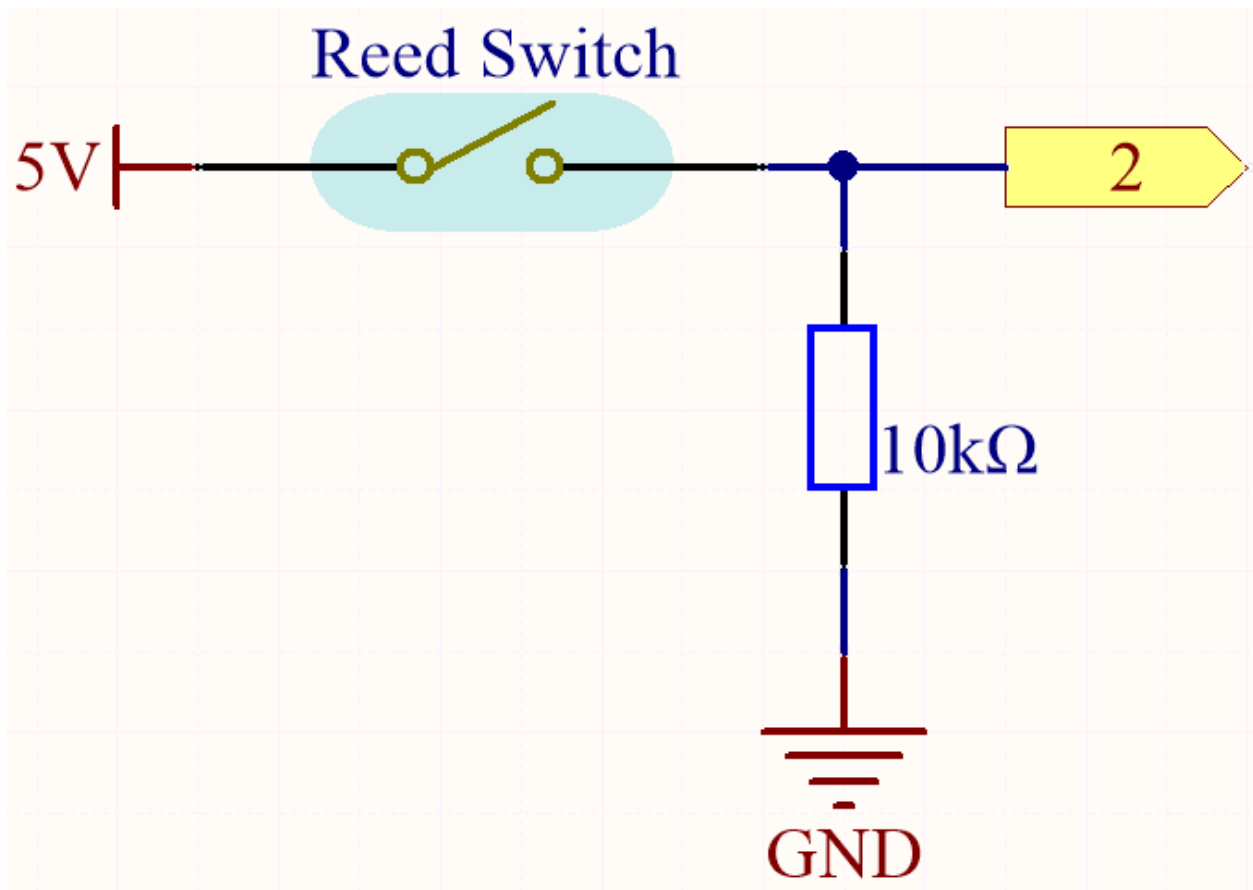
Es ist definitiv praktisch, ein ganzes Kit zu kaufen. Hier ist der Link:

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>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Reedschalter</i>	-

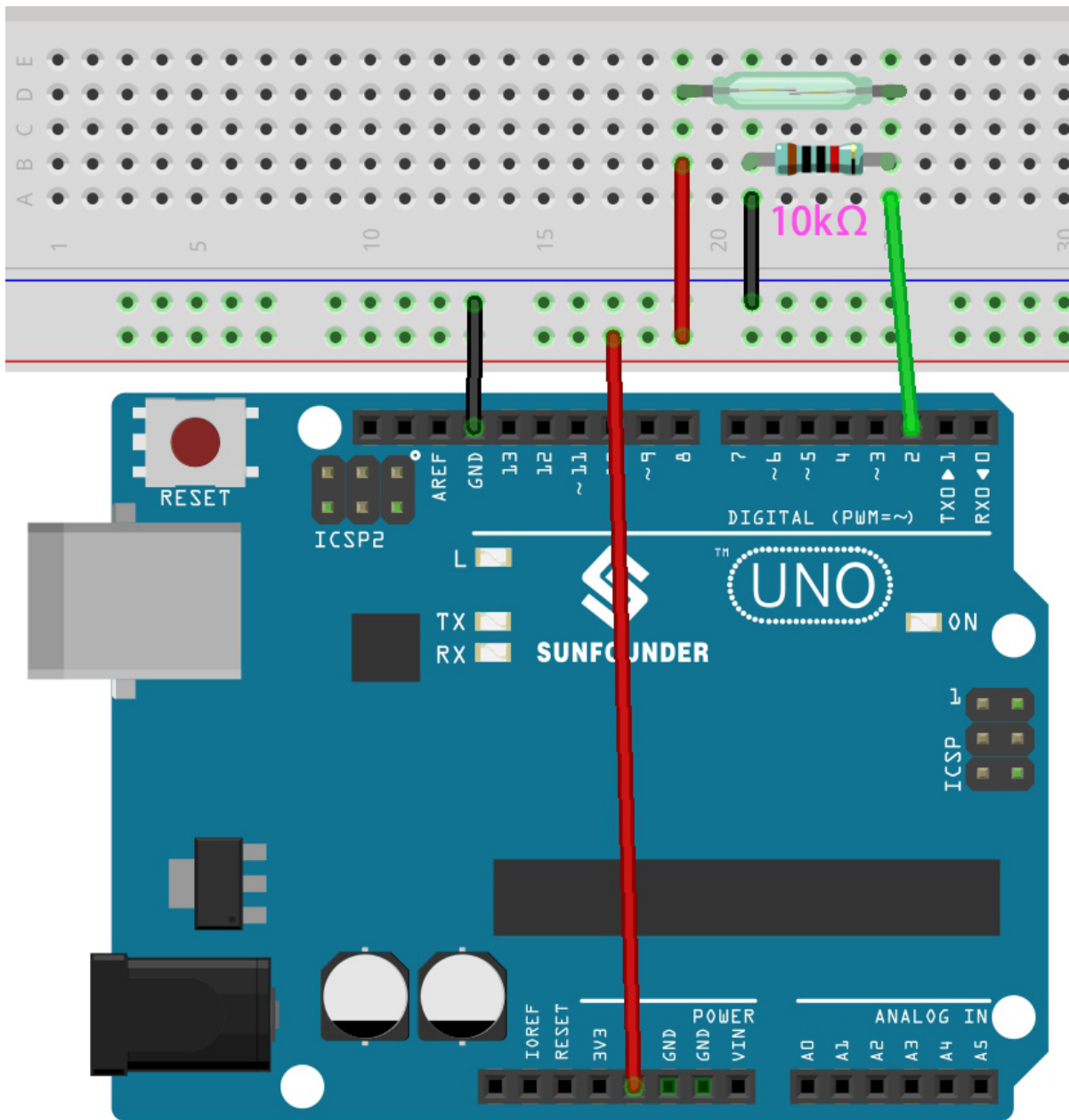
Schaltplan



Standardmäßig ist Pin 2 niedrig; er wird hochschalten, wenn der Magnet in der Nähe des Reed-Schalters ist.

Der Zweck des 10K Widerstands besteht darin, Pin 2 auf einem stabilen niedrigen Niveau zu halten, wenn kein Magnet in der Nähe ist.

Verkabelung



Code

Bemerkung:

- Sie können die Datei `3.2.feel_the_magnetism.ino` unter dem Pfad `3in1-kit\basic_project\3.2.feel_the_magnetism` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nach erfolgreichem Hochladen des Codes wird, wenn ein Magnet in der Nähe des Reed-Schalters ist, im seriellen Monitor eine 1 angezeigt.

5.3.4 3.3 Hindernisse erkennen

Dieses Modul wird häufig in Autos und Robotern installiert, um das Vorhandensein von Hindernissen vor ihnen zu erkennen. Es findet auch breite Anwendung in Handgeräten, Wasserhähnen und weiteren.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

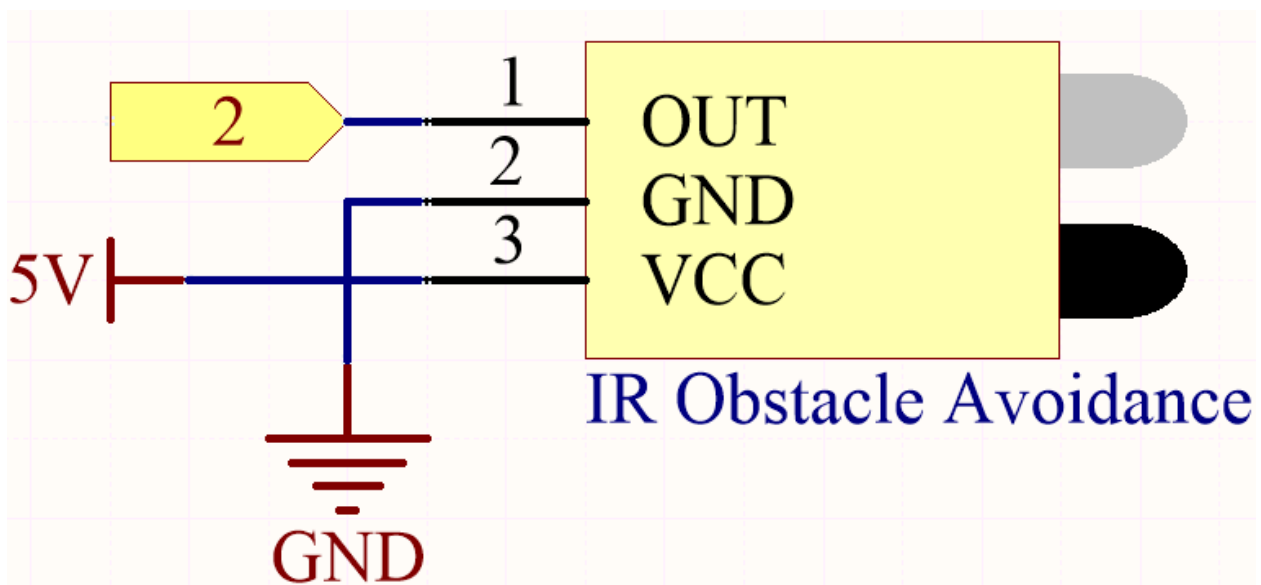
Es ist definitiv praktisch, ein komplettes Set zu kaufen. Hier ist der Link:

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

Sie können diese auch separat über die untenstehenden Links kaufen.

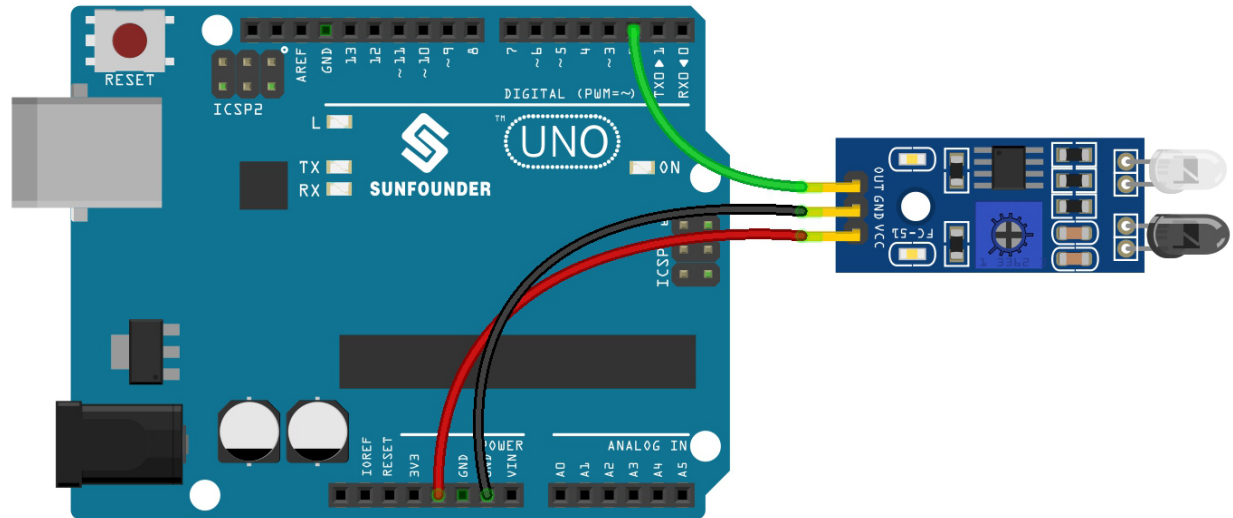
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>Hindernisvermeidungsmodul</i>	

Schaltplan



Der digitale Pin 2 wird verwendet, um das Signal des IR-Hindernisvermeidungsmoduls zu lesen. Das VCC des IR-Sensormoduls ist mit 5V verbunden, GND mit GND und OUT mit dem digitalen Pin 2.

Verkabelung



Code

Bemerkung:

- Sie können die Datei `3.3.detect_the_obstacle.ino` unter dem Pfad `3in1-kit\basic_project\3.3.detect_the_obstacle` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Wenn das IR-Hindernisvermeidungsmodul ein Blockieren vor sich erkennt, wird auf dem seriellen Monitor [0] angezeigt. Andernfalls wird [1] angezeigt.

5.3.5 3.4 Linie erkennen

Das Linienverfolgungsmodul wird verwendet, um festzustellen, ob es schwarze Bereiche auf dem Boden gibt, wie zum Beispiel schwarze Linien, die mit Isolierband geklebt wurden.

Eine seiner LEDs sendet infrarotes Licht zum Boden, und die schwarze Oberfläche hat eine relativ starke Fähigkeit, Licht zu absorbieren und eine schwächere Reflexionsfähigkeit. Weiße Oberflächen verhalten sich genau umgekehrt. Wenn reflektiertes Licht erkannt wird, bedeutet dies, dass der Boden aktuell weiß ist. Wird keines erkannt, ist er schwarz.

So funktioniert es.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

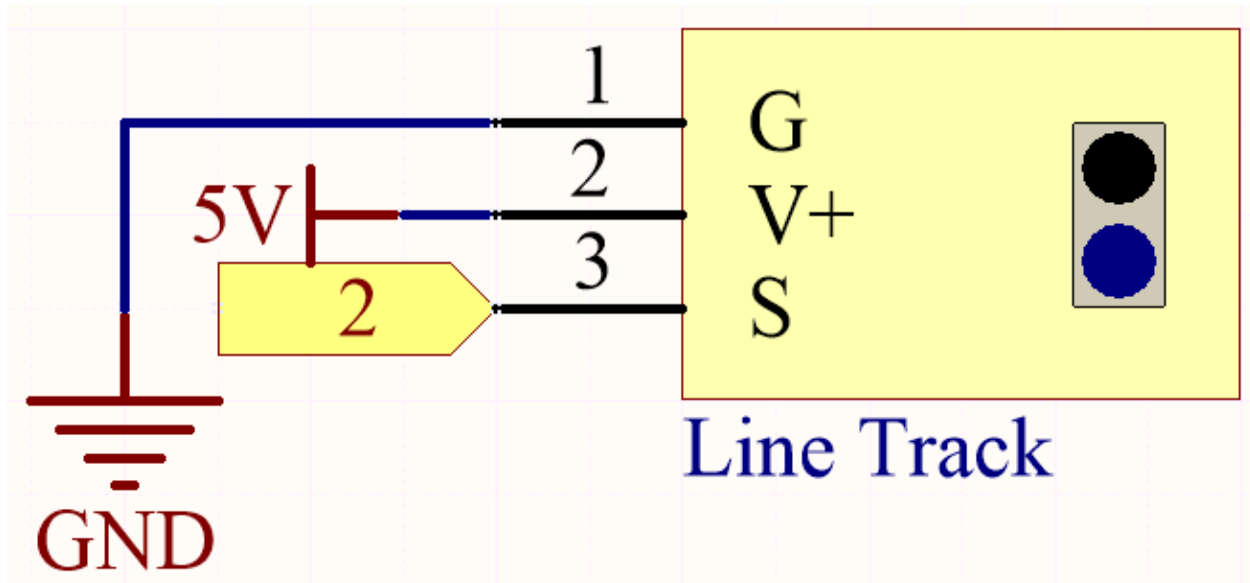
Es ist definitiv praktisch, ein komplettes Set zu kaufen. Hier ist der Link:

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

Sie können diese auch separat über die untenstehenden Links kaufen.

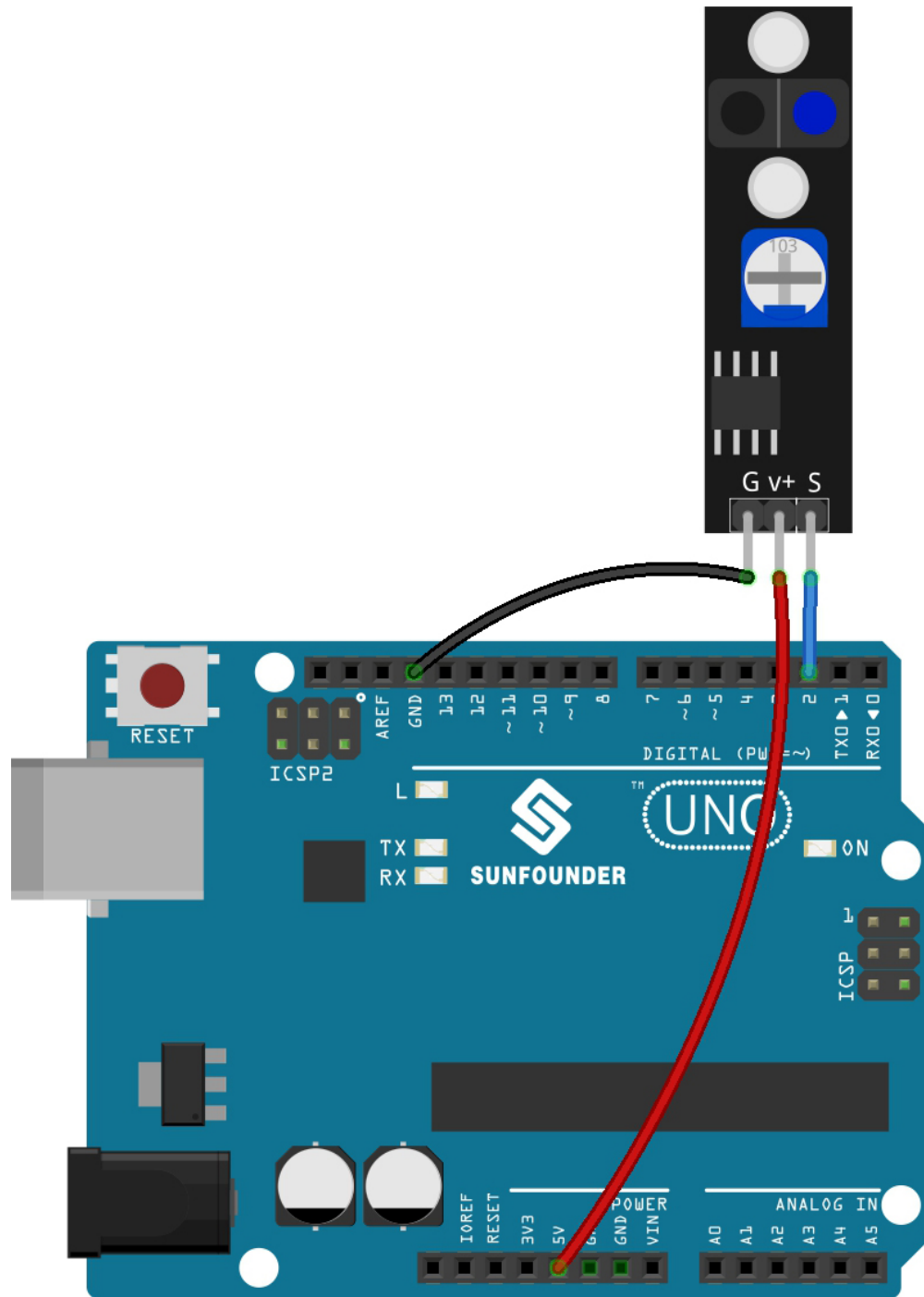
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>Linienverfolgungsmodul</i>	

Schaltplan



Der digitale Pin 2 wird verwendet, um das Signal des Linienverfolgungsmoduls zu lesen. Das VCC des Moduls ist mit 5V verbunden, GND mit GND und OUT mit dem digitalen Pin 2.

Verkabelung



Code

Bemerkung:

- Sie können die Datei 3.4.detect_the_line.ino unter dem Pfad 3in1-kit\basic_project\3.4.detect_the_line öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Wenn das Linienverfolgungsmodul eine schwarze Linie erkennt, wird [1] auf dem seriellen Monitor angezeigt. Andernfalls wird [0] dargestellt.

5.4 4. Analoges Auslesen

Das Arduino kann die angeschlossenen analogen Sensoren über die analogen Pins auslesen.

Das R3-Board enthält einen mehrkanaligen, 10-Bit-Analog-Digital-Wandler. Das bedeutet, dass es die Eingangsspannung zwischen 0 und der Betriebsspannung (5V oder 3,3V) einem Ganzzahlwert zwischen 0 und 1023 zuordnet.

Die Funktion `analogRead(pin)` wird benötigt, um den Wert des analogen Pins auszulesen.

- `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).

Rückgabewerte

0-1023. Datentyp: `int`.

Beispiel für analoges Auslesen

```
int analogPin = A0; // Gerät an den analogen Pin A0 angeschlossen
                    // Äußere Leitungen führen zu Ground und +5V
int val = 0; // Variable zum Speichern des gelesenen Werts

void setup() {
    Serial.begin(9600); // Serielle Schnittstelle einrichten
}

void loop() {
    val = analogRead(analogPin); // Eingangspin auslesen
    Serial.println(val);         // Wert zur Fehlersuche ausgeben
}
```

Hinweise und Warnungen

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

Zugehörige Komponenten

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

5.4.1 4.1 Den Drehknopf drehen

Ein Potentiometer ist ein Widerstandskomponente mit 3 Anschlüssen, dessen Widerstandswert nach einer bestimmten Regelung variiert werden kann.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

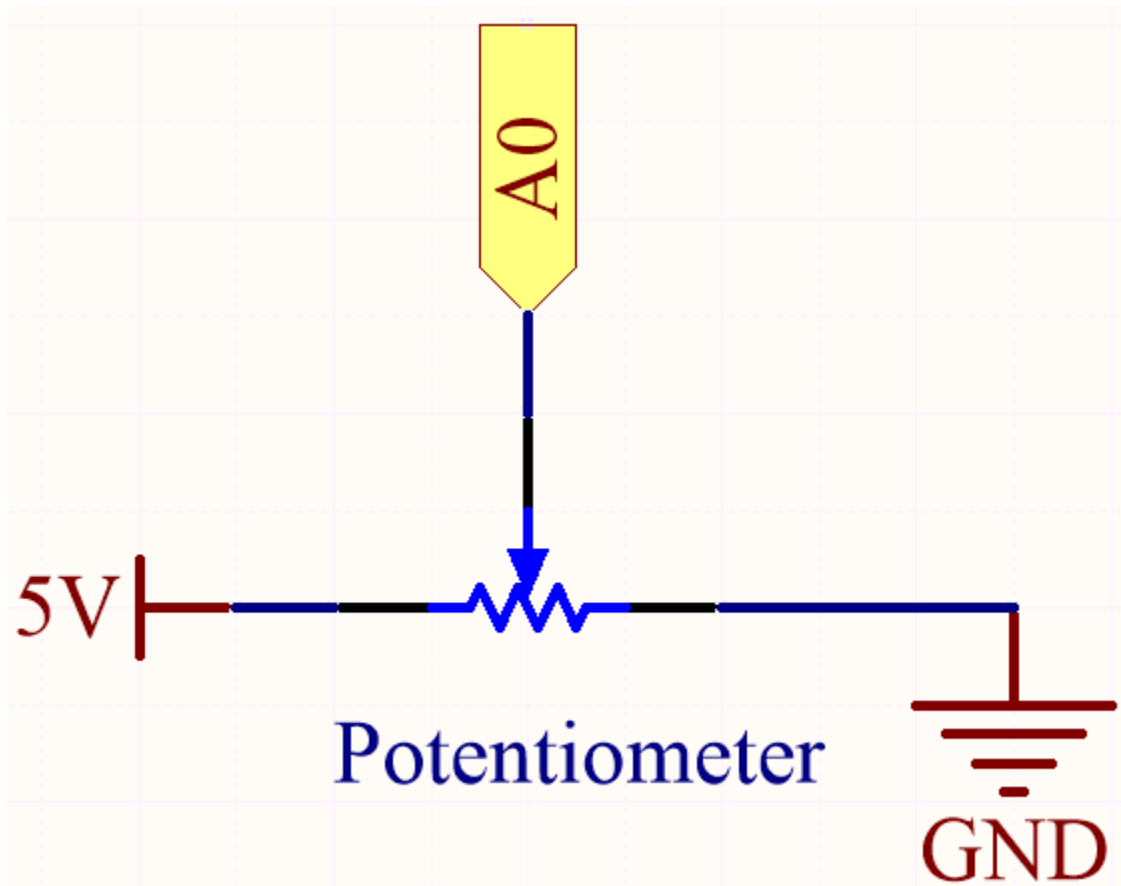
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

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

Sie können diese auch einzeln über die untenstehenden Links kaufen.

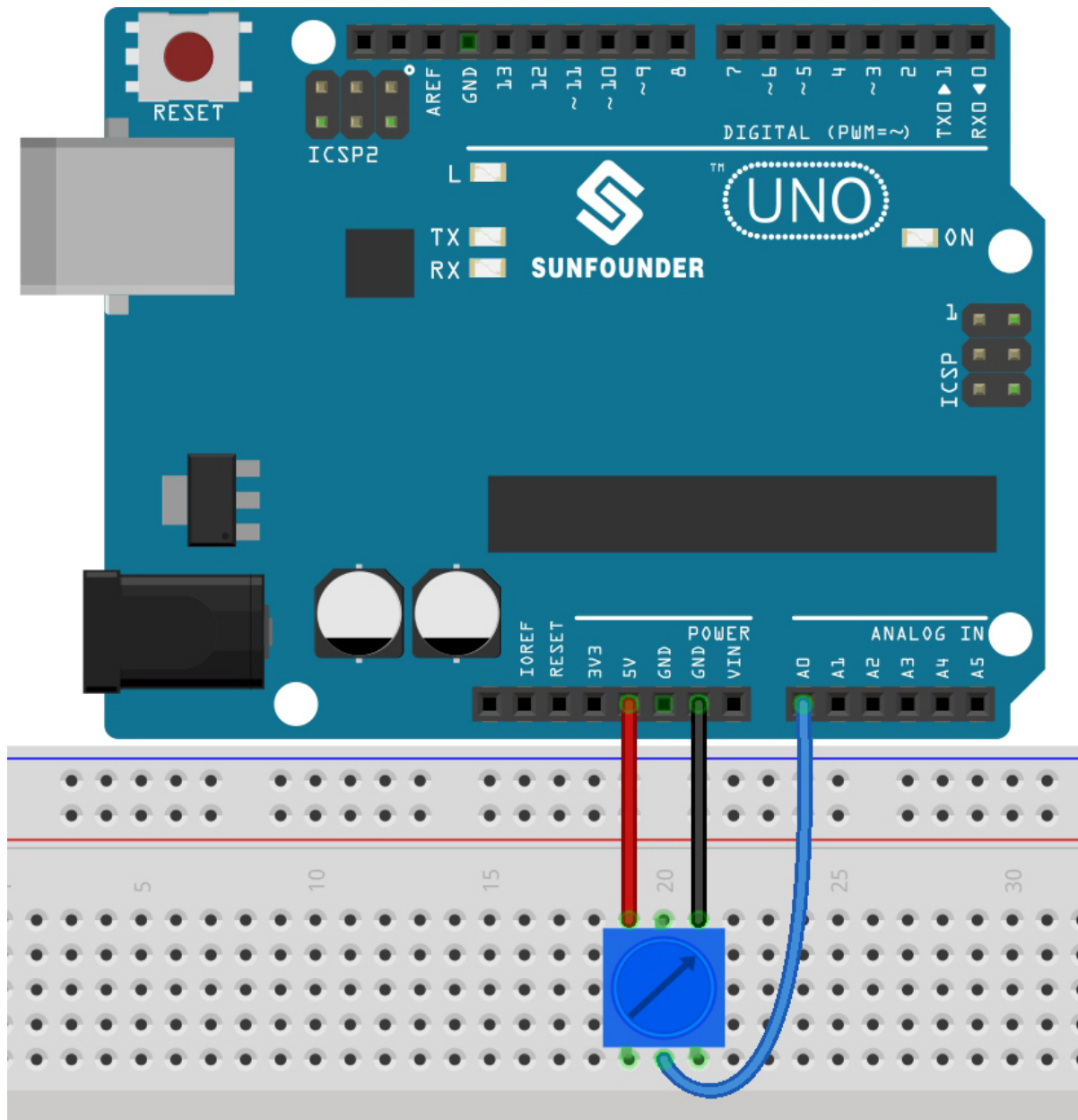
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Potentiometer</i>	

Schaltplan



In diesem Beispiel verwenden wir den analogen Pin (A0), um den Wert des Potentiometers auszulesen. Durch das Drehen der Achse des Potentiometers können Sie die Widerstandsverteilung zwischen diesen drei Pins ändern und so die Spannung am mittleren Pin verändern. Wenn der Widerstand zwischen dem mittleren und einem äußeren, mit 5V verbundenen Pin nahezu null ist (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 (Widerstand zwischen dem mittleren und einem äußeren, mit 5V verbundenen Pin ist nahe 10k) führt dazu, dass die Spannung am mittleren Pin nahe 0V liegt.

Verkabelung



Code

Bemerkung:

- Sie können die Datei `4.1.turn_the_knob.ino` im Pfad `3in1-kit/basic_project/4.1.turn_the_knob` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nachdem Sie die Codes auf das Board geladen haben, können Sie den seriellen Monitor öffnen, um den gelesenen Wert des Pins zu sehen. Wenn Sie die Achse des Potentiometers drehen, wird der serielle Monitor den Wert zwischen 0 und 1023 anzeigen.

5.4.2 4.2 Licht wahrnehmen

Der Fotowiderstand ist ein typisches Gerät für analoge Eingänge und wird in einer sehr ähnlichen Weise wie ein Potentiometer verwendet. Sein Widerstandswert hängt von der Intensität des Lichts ab: Je stärker das einfallende Licht, desto geringer ist sein Widerstandswert und umgekehrt.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

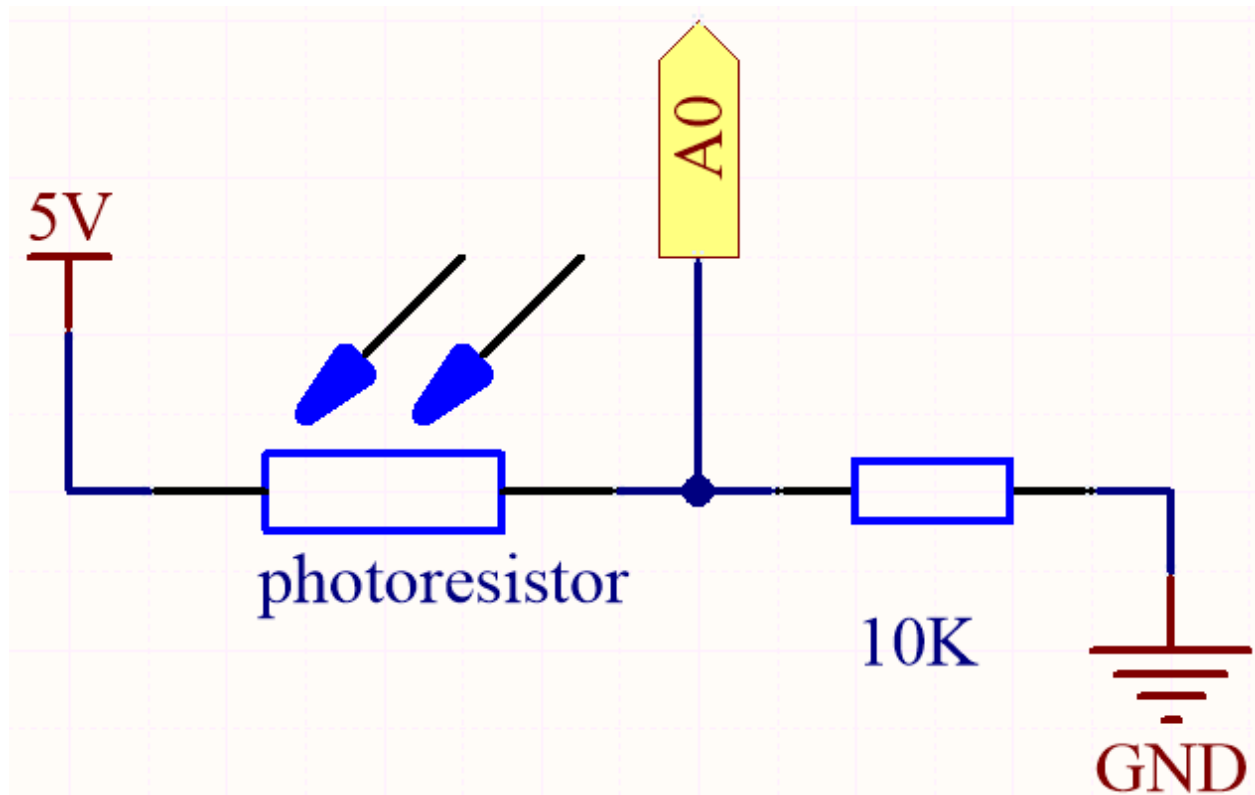
Es ist sicherlich praktisch, ein komplettes Set zu kaufen, hier ist der Link:

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

Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Fotowiderstand</i>	

Schaltplan

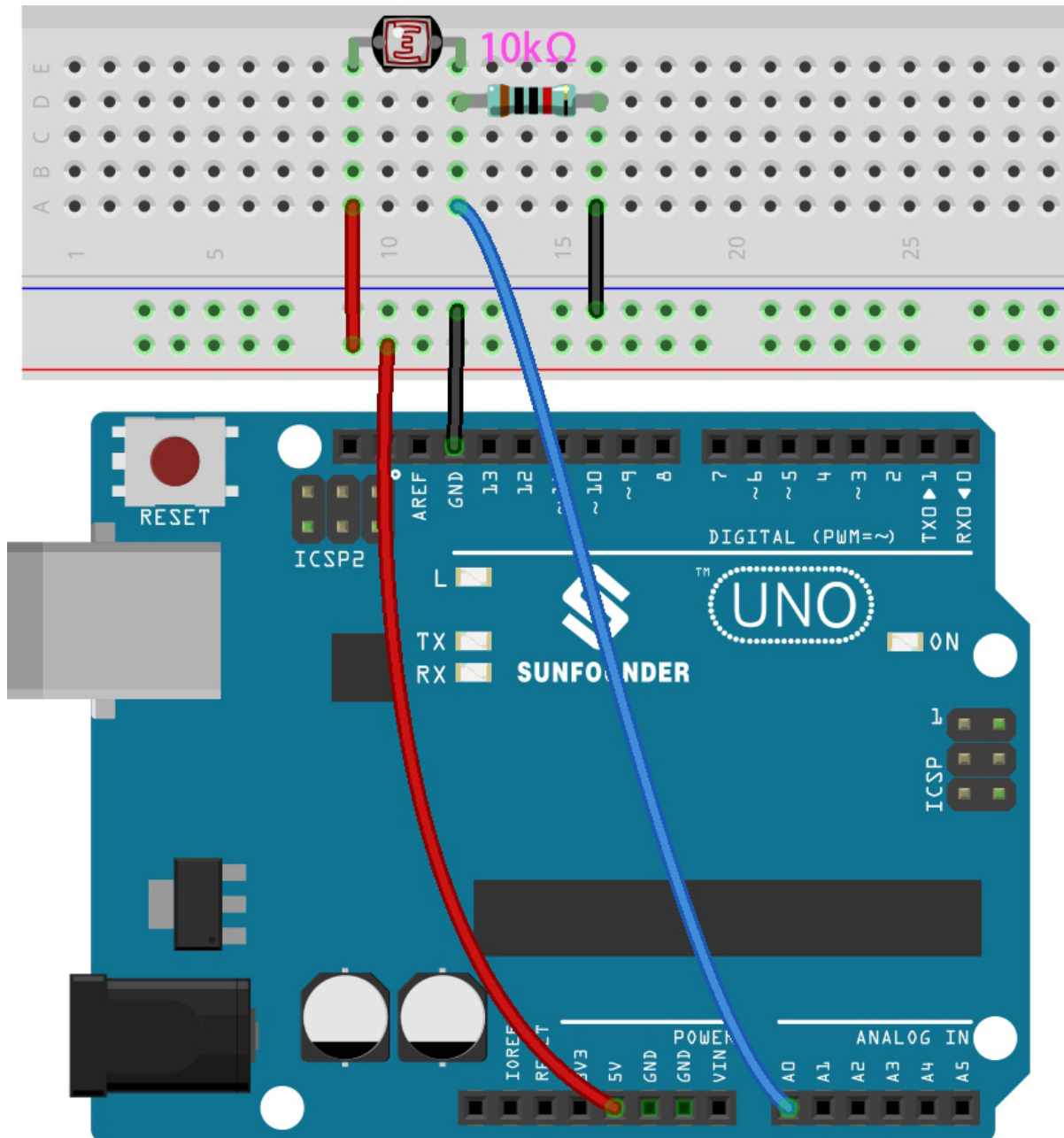


In dieser Schaltung sind der 10K Widerstand und der Fotowiderstand in Reihe geschaltet und der durch sie fließende Strom ist derselbe. Der 10K Widerstand dient als Schutz und der Pin A0 liest den Wert nach der Spannungsumwandlung des Fotowiderstands.

Wenn das Licht verstärkt wird, verringert sich der Widerstand des Fotowiderstands, seine Spannung sinkt und der Wert von Pin A0 steigt. Wenn das Licht stark genug ist, wird der Widerstand des Fotowiderstands nahezu 0 sein und der Wert von Pin A0 wird nahe 1023 liegen. Der 10K Widerstand spielt hierbei eine schützende Rolle, sodass 5V und GND nicht miteinander verbunden werden und es zu einem Kurzschluss kommt.

Platziert man den Fotowiderstand in einer dunklen Umgebung, wird der Wert von Pin A0 sinken. In einer ausreichend dunklen Umgebung wird der Widerstand des Fotowiderstands unendlich groß und seine Spannung nahe 5V liegen (der 10K Widerstand ist vernachlässigbar), der Wert von Pin A0 wird nahe 0 liegen.

Verkabelung



Code

Bemerkung:

- Öffnen Sie die Datei `4.2.feel_the_light.ino` unter dem Pfad `3in1-kit\basic_project\4.2.feel_the_light`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

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

5.4.3 4.3 Den Joystick betätigen

Jeder, der regelmäßig Videospiele spielt, sollte den Joystick gut kennen. Er wird normalerweise verwendet, um Charaktere zu bewegen oder den Bildschirm zu drehen.

Unsere Bewegungen können vom Joystick erkannt 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 Set beinhaltet auch einen Joystick mit digitalem Eingang. Er wird aktiviert, wenn der Joystick gedrückt wird.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

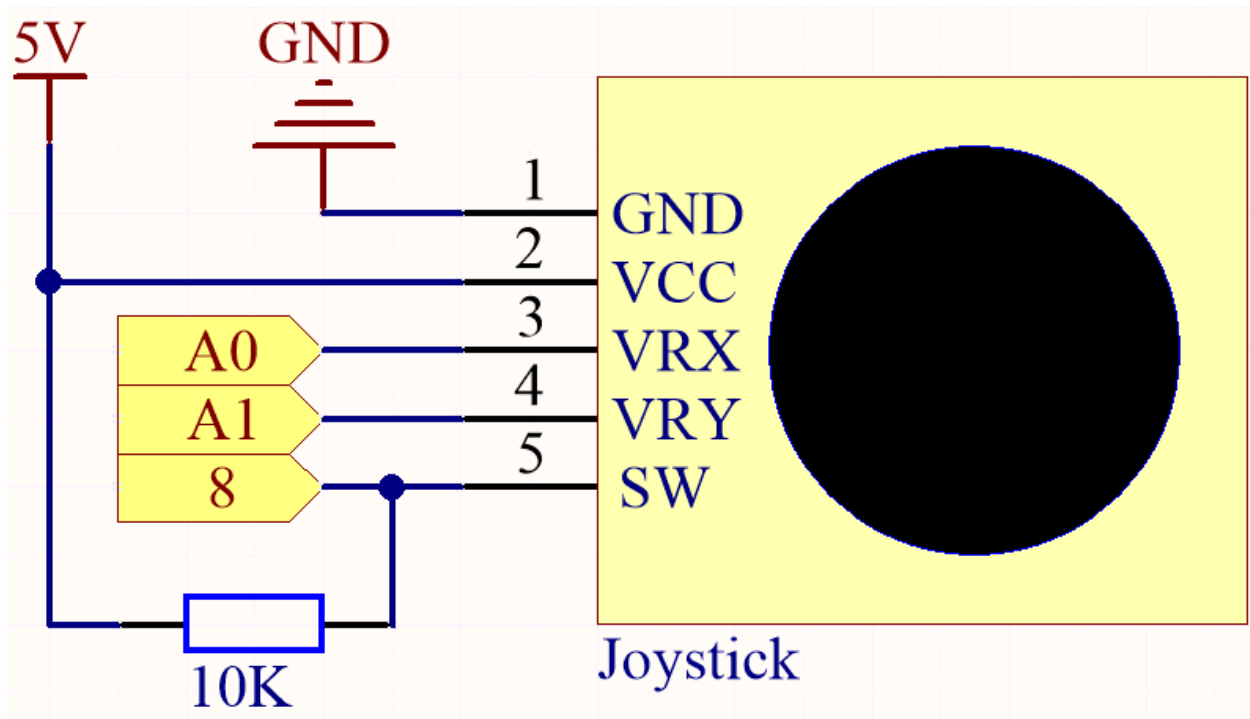
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

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

Sie können sie auch separat über die untenstehenden Links kaufen.

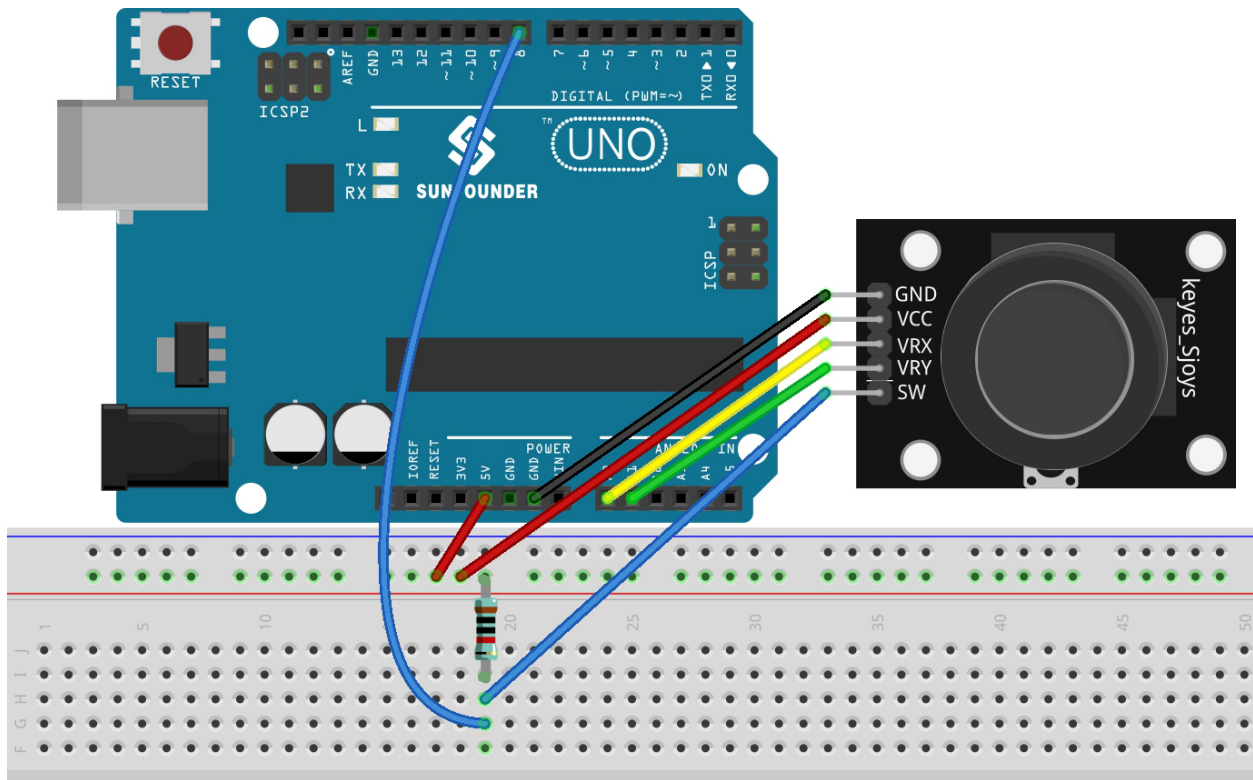
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Joystick Modul</i>	-

Schaltplan



Bemerkung: Der SW-Pin ist mit einem 10K Pull-up-Widerstand verbunden, um ein stabiles Hochpegelsignal am SW-Pin (Z-Achse) zu erhalten, wenn der Joystick nicht gedrückt ist; sonst befindet sich der SW in einem schwebenden Zustand, und der Ausgabewert kann zwischen 0/1 variieren.

Verkabelung



Code

Bemerkung:

- Öffnen Sie die Datei `4.3.toggle_the_joystick.ino` im Pfad `3in1-kit\basic_project\4.3.toggle_the_joystick`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Öffnen Sie den seriellen Monitor, nachdem der Code erfolgreich hochgeladen wurde, um die x,y,z-Werte des Joysticks zu sehen.

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

5.4.4 4.4 Bodenfeuchtigkeit messen

In der Pflanzenzucht können die Pflanzen die anorganischen Elemente im Boden nicht direkt aufnehmen. Das Wasser im Boden dient als Lösungsmittel, um diese anorganischen Elemente zu lösen.

Die Pflanzen nehmen die Bodenfeuchtigkeit über das Wurzelsystem auf, erhalten Nährstoffe und fördern das Wachstum.

Im Prozess des Pflanzenwachstums und der Entwicklung unterscheiden sich auch die Anforderungen an die Bodentemperatur. Daher wird ein Bodenfeuchtigkeitssensor benötigt.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

SunFounder 3in1 Kit

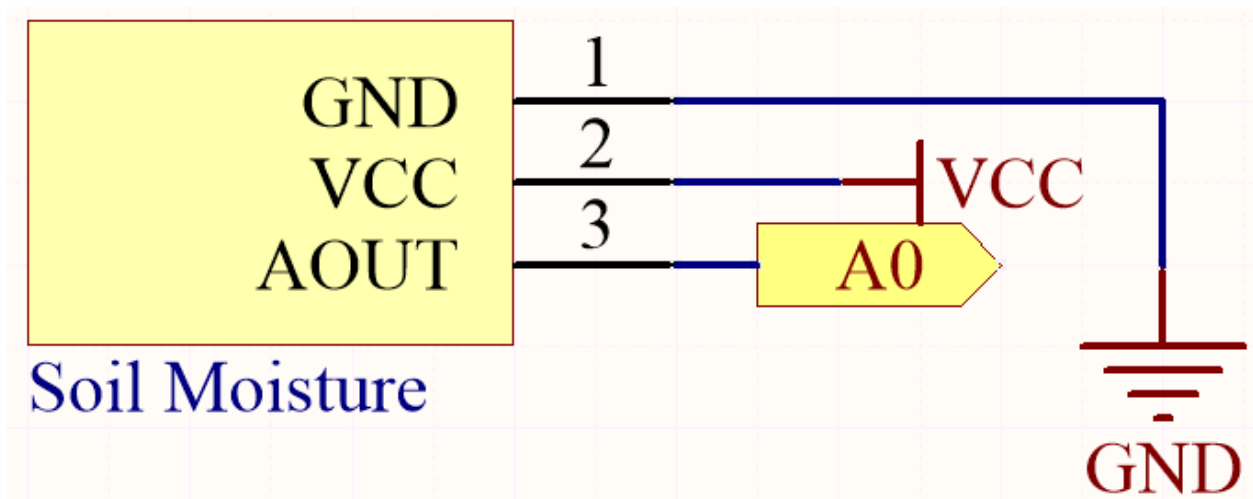
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

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

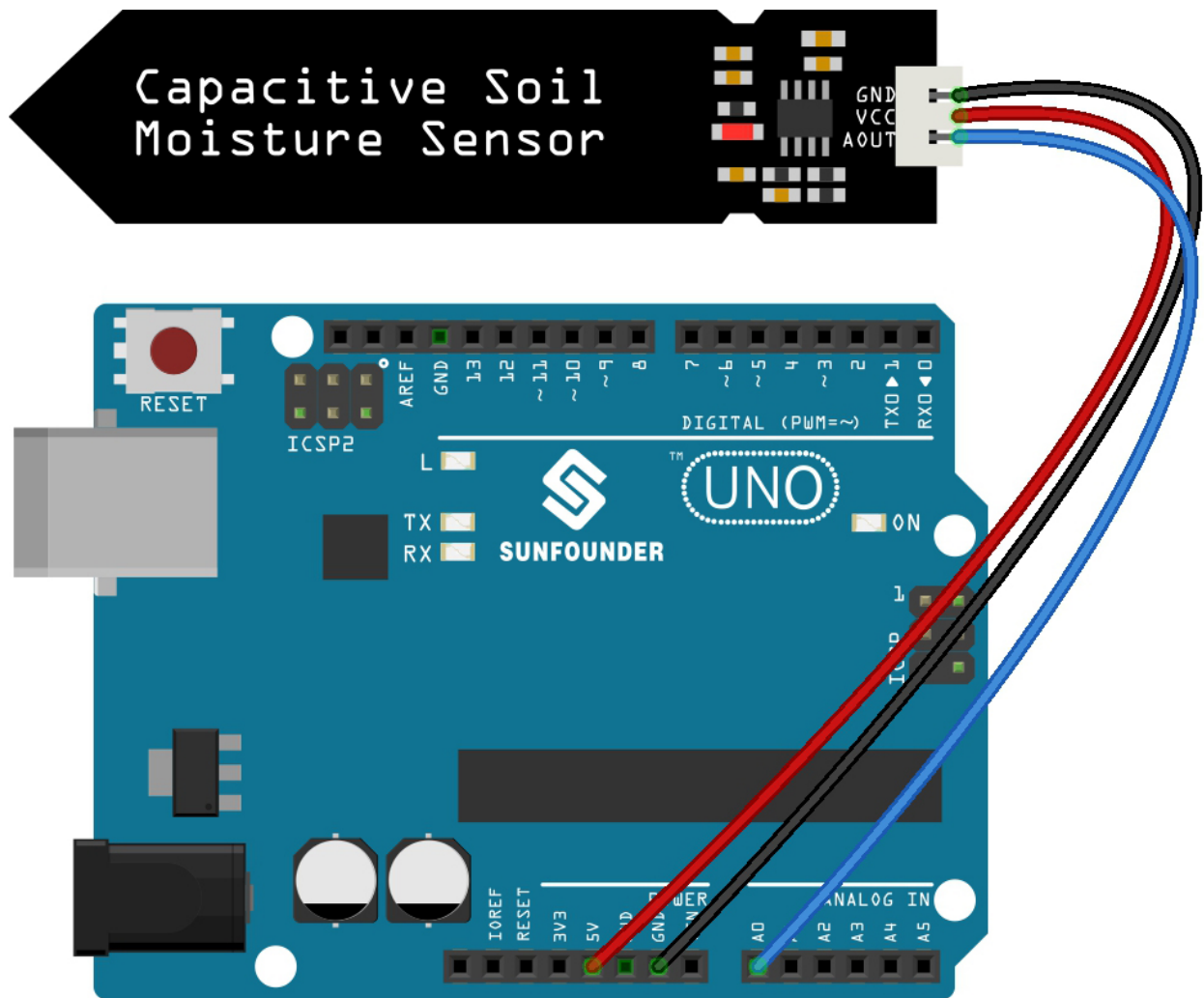
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>Bodenfeuchtigkeitsmodul</i>	

Schaltplan



Verkabelung



Code

Bemerkung:

- Öffnen Sie die Datei `4.4.measure_soil_moisture.ino` im Pfad `3in1-kit\basic_project\4.4.measure_soil_moisture`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nach erfolgreichem Hochladen des Codes wird der serielle Monitor den Wert der Bodenfeuchtigkeit ausgeben.

Wenn Sie das Modul in den Boden stecken und ihn bewässern, wird der Wert des Bodenfeuchtigkeitssensors kleiner.

5.4.5 4.5 Thermometer

Ein Thermometer ist ein Gerät, das die Temperatur oder einen Temperaturgradienten misst (das Maß für die Hitze oder Kälte eines Objekts). Ein Thermometer hat zwei wichtige Elemente: (1) einen Temperatursensor (z.B. die Glühbirne eines Quecksilber-Thermometers oder den pyrometrischen Sensor in einem Infrarot-Thermometer), bei dem eine Änderung mit einer Temperaturänderung eintritt; und (2) eine Möglichkeit, diese Änderung in einen Zahlenwert umzuwandeln (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 Prozessüberwachung, in der Meteorologie, in der Medizin und in der wissenschaftlichen Forschung weit verbreitet verwendet.

Ein Thermistor ist eine Art Temperatursensor, dessen Widerstand stark temperaturabhängig ist und er hat 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 zu bauen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

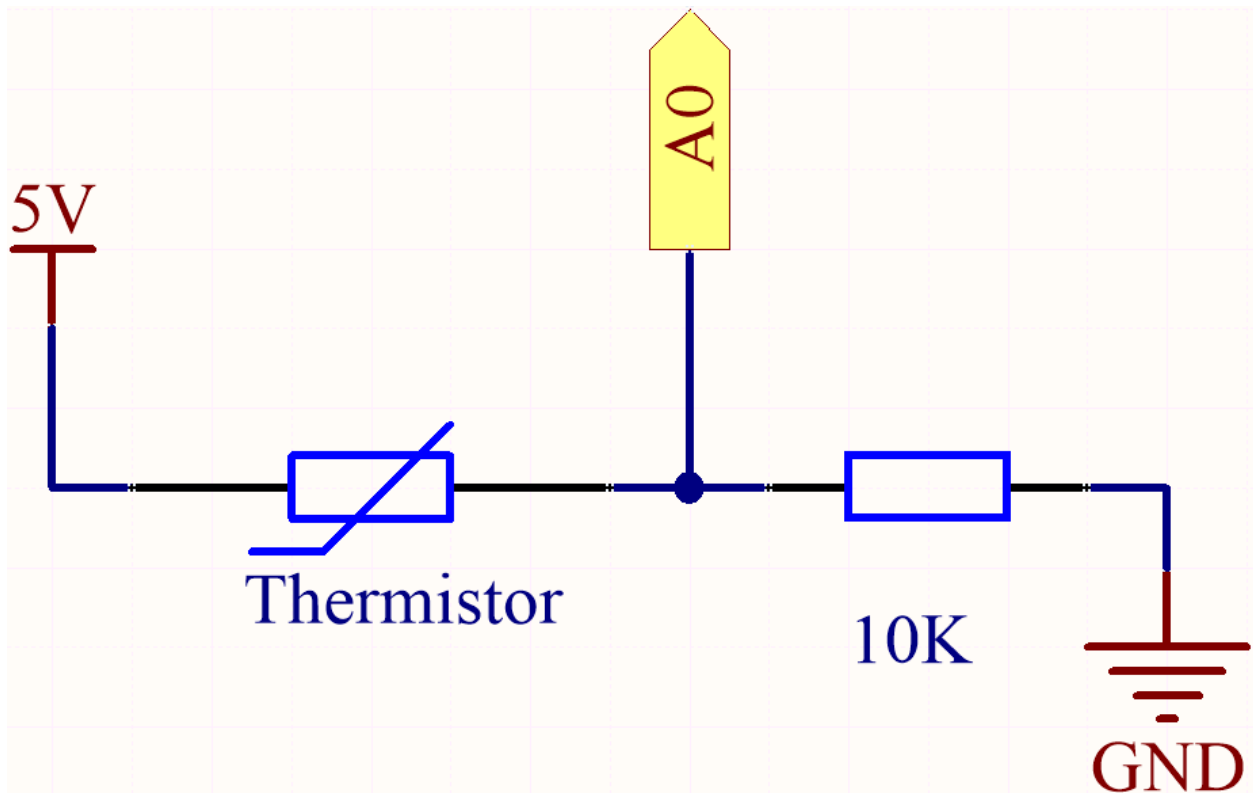
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

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>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Thermistor</i>	

Schaltplan



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

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

Die Temperatur in Celsius oder Fahrenheit wird mittels 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 Nenn-Temperatur T_N . Hier beträgt der Wert von R_N 10k.
- **T_K** ist eine Kelvin-Temperatur und die Einheit ist K. Der Wert von **T_K** beträgt hier $273,15 + \text{Grad Celsius}$.
- **T_N** ist eine bewertete Kelvin-Temperatur; die Einheit ist auch 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 einen Wert von 3950.
- **\exp** steht für Exponentialfunktion, und die Basiszahl e ist eine natürliche Zahl und beträgt ungefähr 2,7.

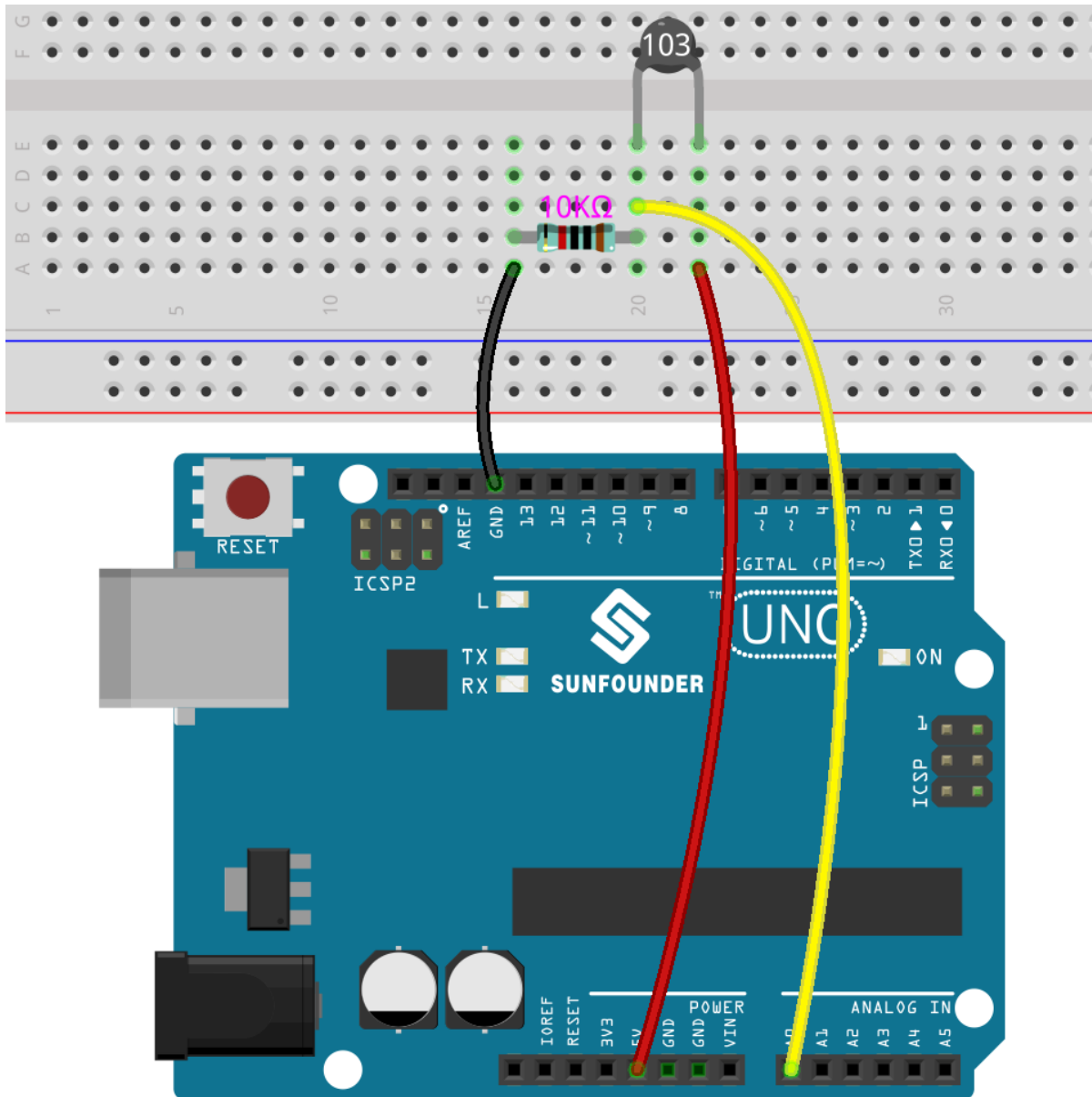
Wandeln Sie diese Formel $T_K = 1 / (\ln(R_T/R_N) / B + 1/T_N)$ um, um die Kelvin-Temperatur zu erhalten, die minus 273,15 Grad Celsius entspricht.

Diese Beziehung ist eine empirische Formel. Sie ist nur dann genau, wenn Temperatur und Widerstand im effektiven Bereich liegen.

Verkabelung

Bemerkung:

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



Code

Bemerkung:

- Sie können die Datei `4.5_thermometer.ino` im Pfad `euler-kit/arduino/4.5_thermometer` öffnen.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Vergessen Sie nicht, das Raspberry Pi Pico Board und den richtigen Port auszuwählen, bevor Sie auf den Upload-Button klicken.

Nach erfolgreichem Hochladen des Codes wird der Serial Monitor die Temperaturen in Celsius und Fahrenheit ausgegeben.

5.5 5. Weitere 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.

5.5.1 5.1 If else

Normalerweise nutzen wir bedingte Beurteilungen, um die einfachsten Interaktionsprojekte mit der Realität abzuschließen. Hier erstellen wir ein Türerkennungssystem mit einem Reed-Kontakt und einer LED, um diese Logik zu zeigen.

Befestigen Sie den Magneten auf einer Seite der Tür und den Reed-Kontakt (mit Schaltung) auf der anderen Seite der Tür. Wenn die Tür geschlossen ist, ist der Magnet dem Reed-Kontakt nahe und schaltet diesen ein.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

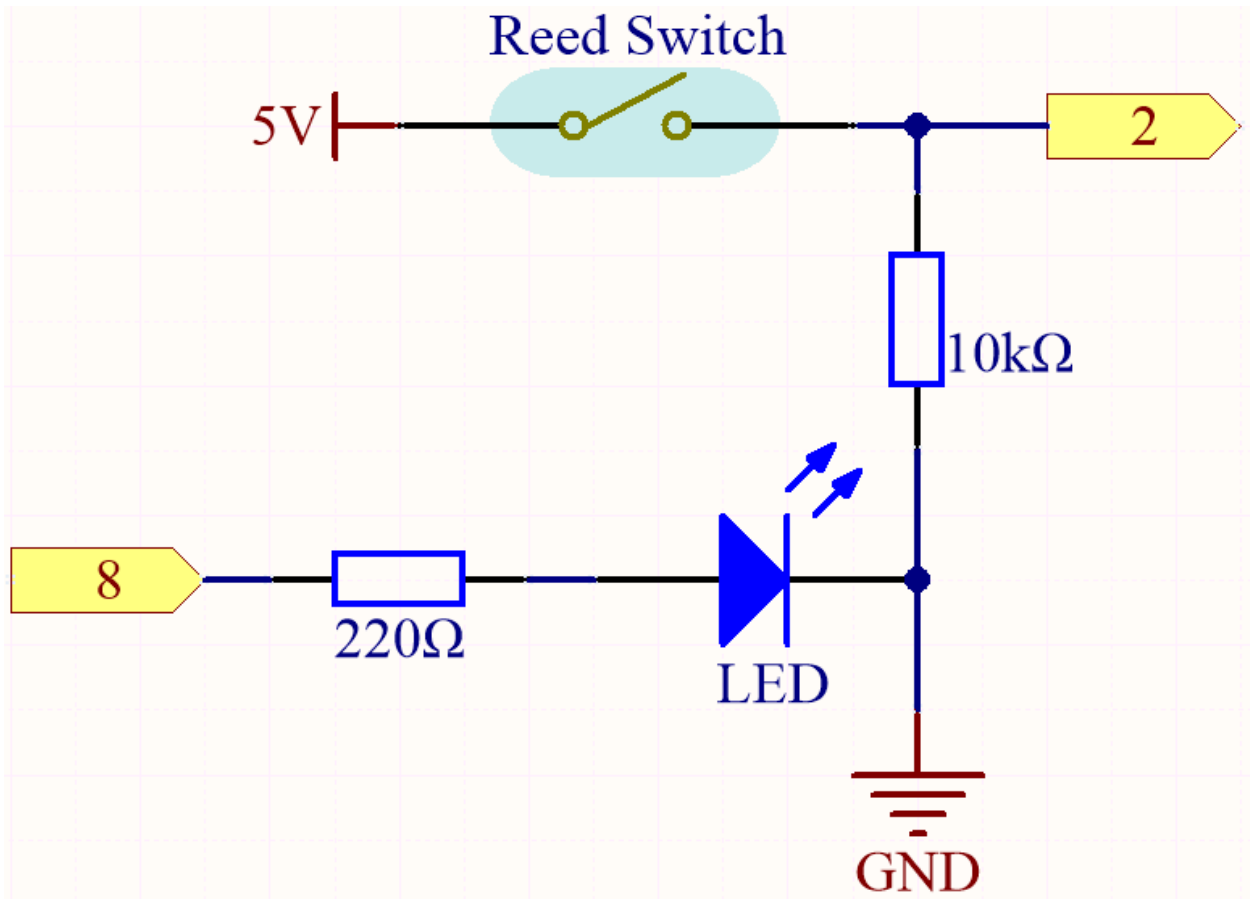
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

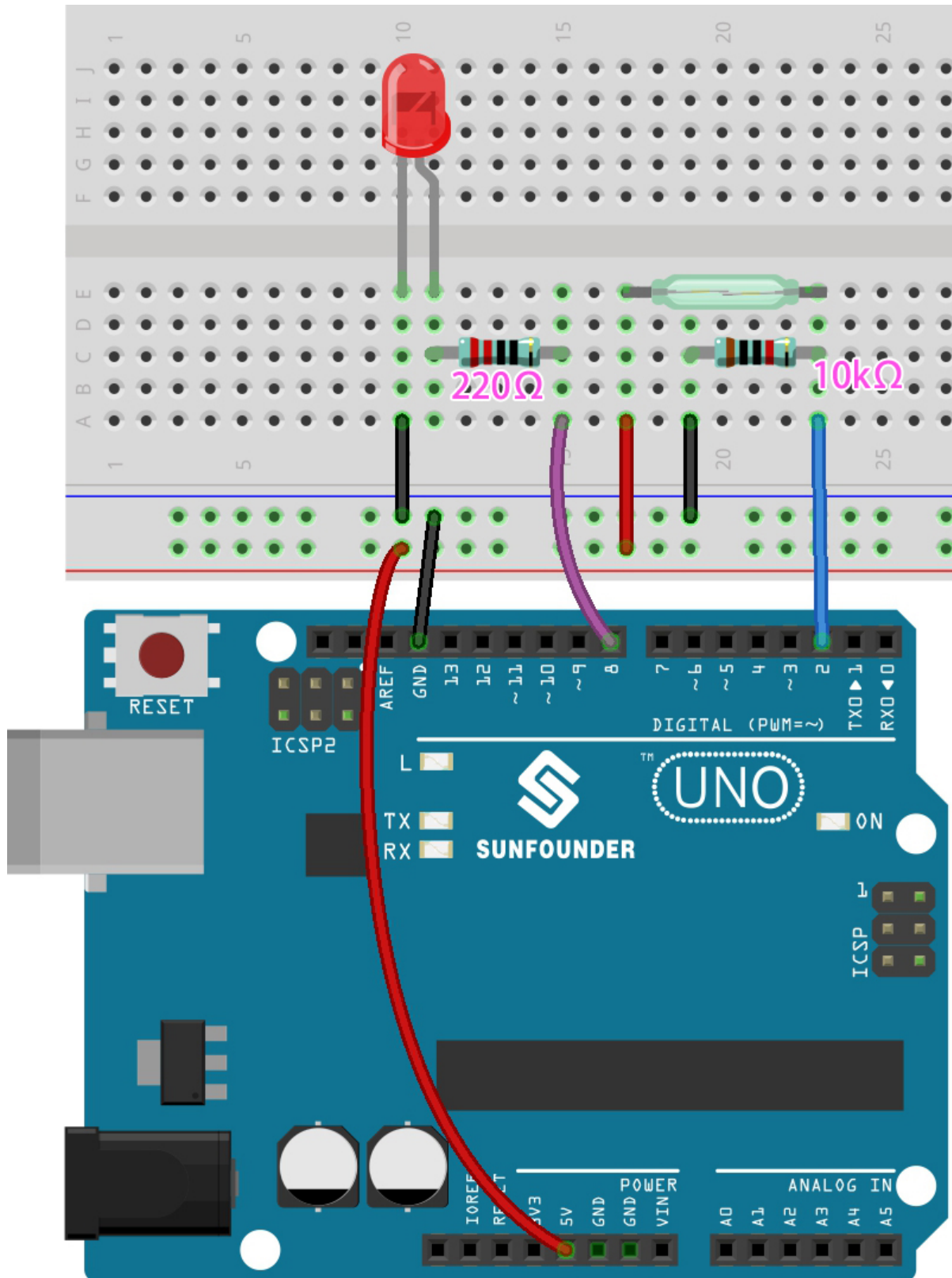
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Reedschalter</i>	-

Schaltplan



Verkabelung



Code

Bemerkung:

- Öffnen Sie die Datei `5.1.if_else.ino` unter dem Pfad `3in1-kit/basic_project/5.1.if_else`.
- 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 und wenn Sie die Tür nicht schließen, leuchtet die LED auf und fordert Sie auf, die Tür zu schließen.

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

- `if else`

Das `if else` ermöglicht eine bessere Kontrolle über den Code-Fluss als die grundlegende `if`-Anweisung, indem mehrere Tests gruppiert werden können.

5.5.2 5.2 Schwellenwert

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

Zum Beispiel in einem Smart Home: Wenn die Lichtintensität unter 50 Lux fällt, schalte das Licht ein; Ein weiteres Beispiel ist auf dem Computer-Motherboard: Wenn die Betriebstemperatur des CPUs über 65 Grad Celsius steigt, 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 besser an individuelle Bedürfnisse anzupassen. Zum Beispiel: Wenn ich eine hellere Wohnatmosphäre bevorzuge, kann ich den Schwellenwert für das automatische Licht im Smart Home auf 80 Lux erhöhen. Ein weiteres Beispiel: Das Lüftungsumfeld meines Studios ist nicht sehr gut und der Kühlbedarf höher, dann kann der Schwellenwert für das automatische Lüftereinschalten auf 50 Grad Celsius angepasst werden.

Hier verwenden wir einen Bodenfeuchtigkeitssensor und 2 LEDs, um einen Pflanzentopf-Monitor 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 manuell anpassen, um die Trockenheit und Feuchtigkeit des Bodens zu bestimmen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

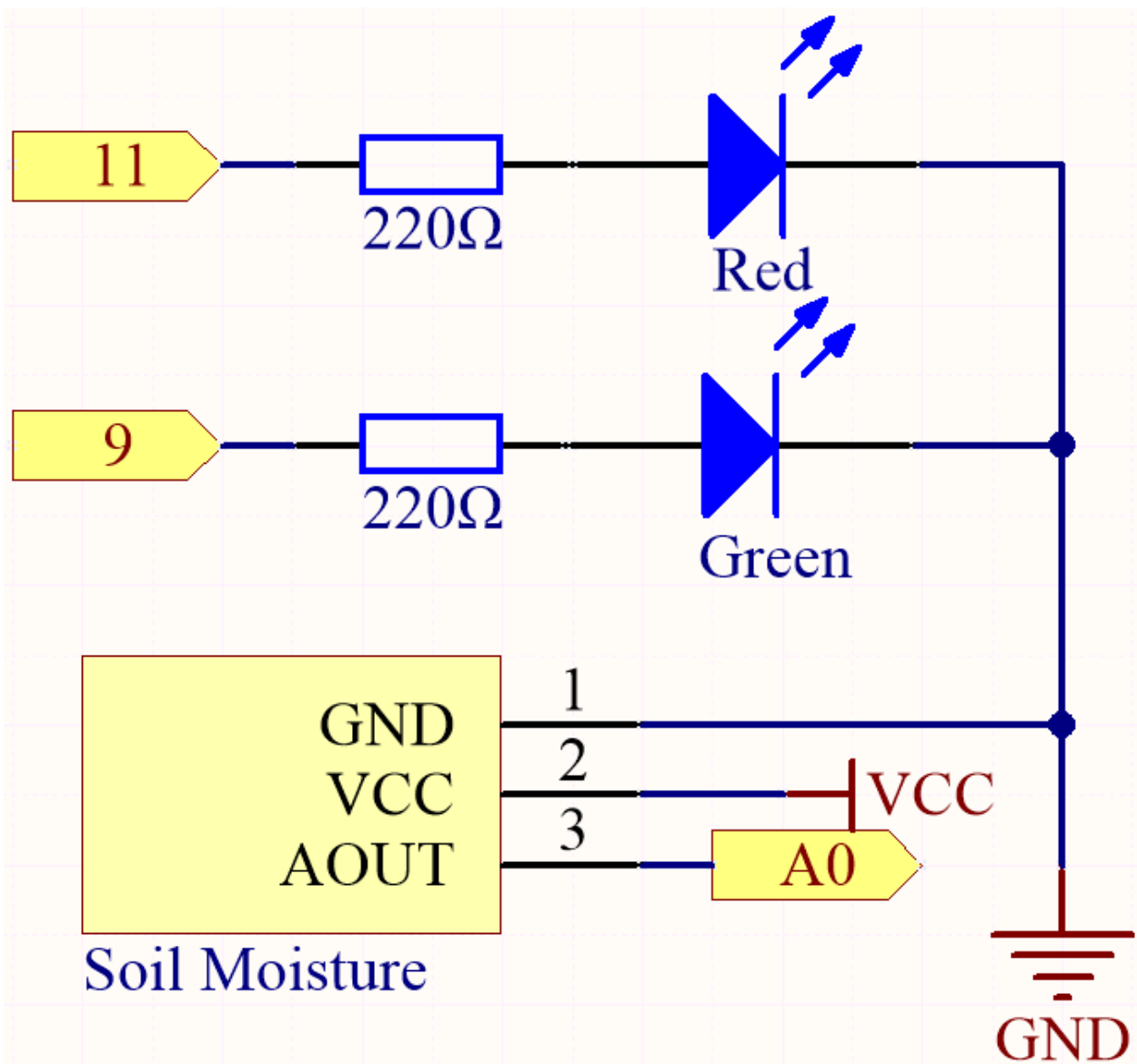
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

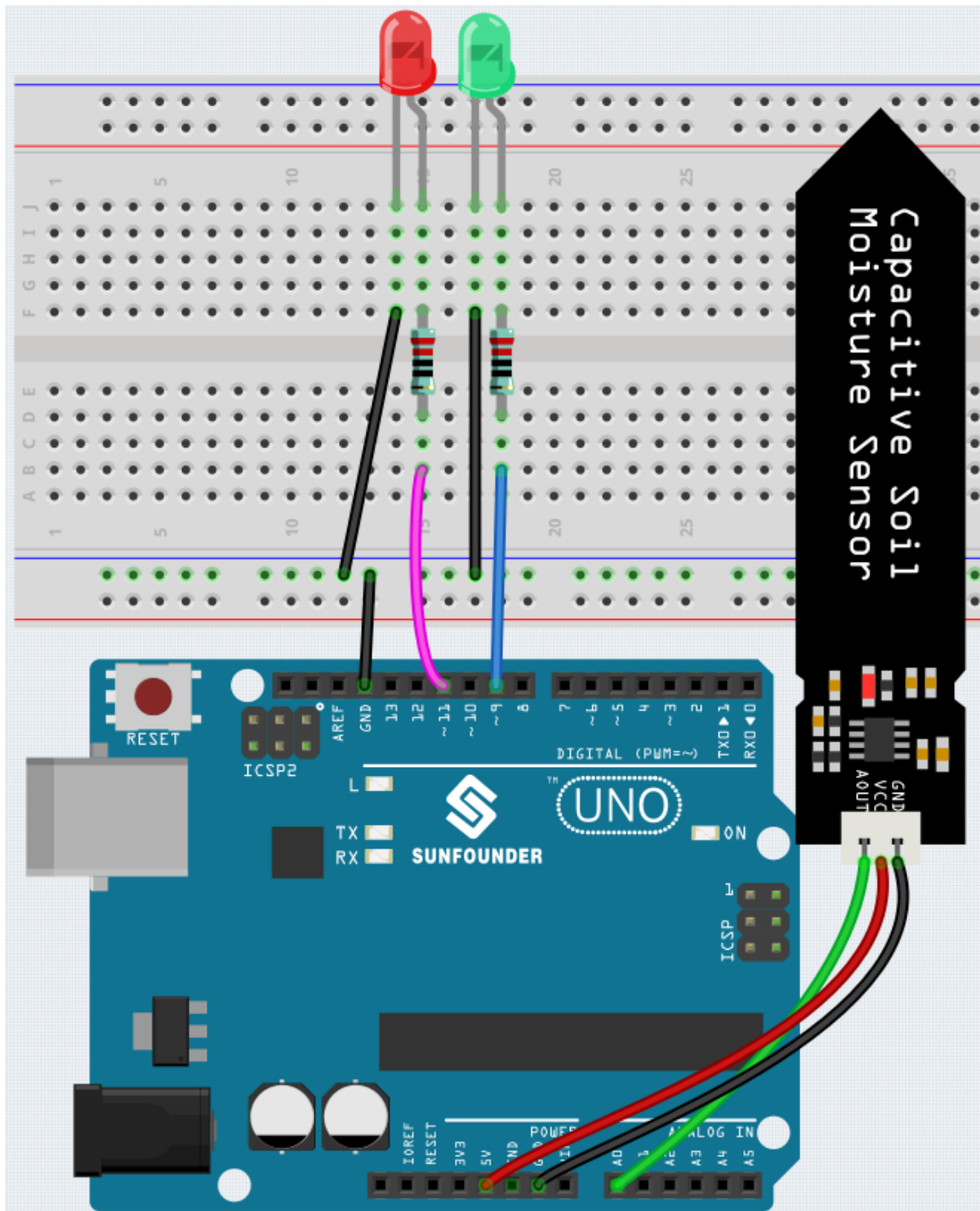
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Bodenfeuchtigkeitsmodul</i>	

Schaltplan



Verkabelung



Code

Bemerkung:

- Öffnen Sie die Datei `5.2.threshold.ino` unter dem Pfad `3in1-kit/basic_project/5.2.threshold`.

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

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

Wie funktioniert das?

```
...

void loop() {
    int sensorValue = analogRead(soilMoisture);
    Serial.println(sensorValue);
    if (sensorValue > threshold) {
        digitalWrite(redPin, HIGH); // Rote LED einschalten
        digitalWrite(greenPin, LOW); // Grün ausschalten
    } else {
        digitalWrite(greenPin, HIGH); // Grüne LED einschalten
        digitalWrite(redPin, LOW); // Rot ausschalten
    }
}

...
```

Zunächst setzen Sie einen Schwellenwert und lesen dann den Wert des Bodenfeuchtigkeitsmoduls aus, dessen Wert mit steigendem Feuchtigkeitsgrad abnimmt. Wenn der aktuell gelesene Wert größer als der gesetzte Schwellenwert ist, lassen Sie die rote LED leuchten, andernfalls schalten Sie die grüne LED ein.

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

5.5.3 5.3 Zustandswechsel-Erkennung

Wenn der Knopf andere Geräte steuert, kann er nicht nur bei Betätigung arbeiten und bei Loslassen stoppen. Es ist auch möglich, bei jedem Tastendruck den Arbeitszustand zu wechseln.

Um diesen Effekt zu erzielen, müssen Sie wissen, wie Sie den Arbeitszustand beim Drücken des Knopfes zwischen aus und an wechseln können. Das nennt man „Zustandswechsel-Erkennung“.

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

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

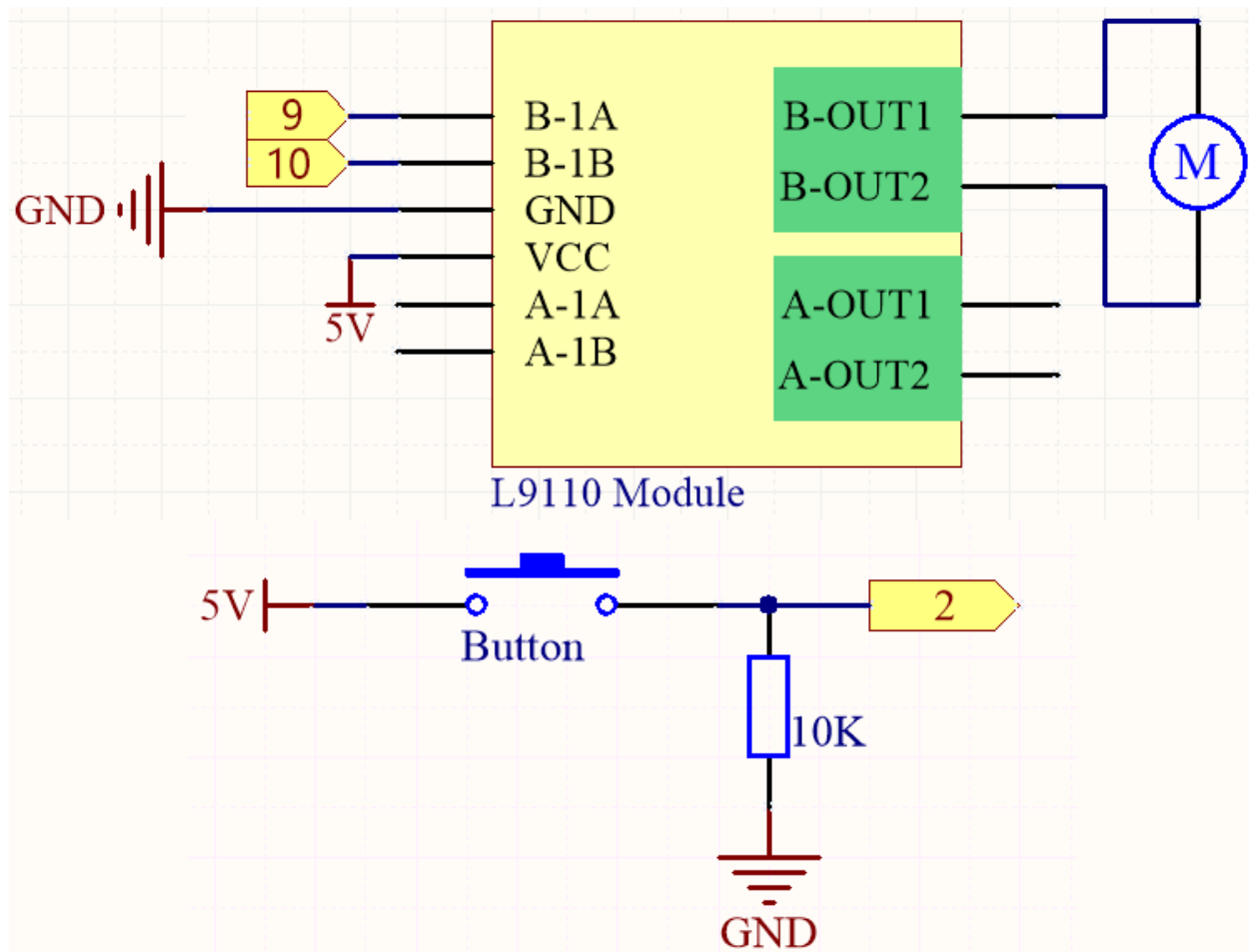
Es ist definitiv praktisch, ein komplettes Set zu kaufen. Hier ist der Link:

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

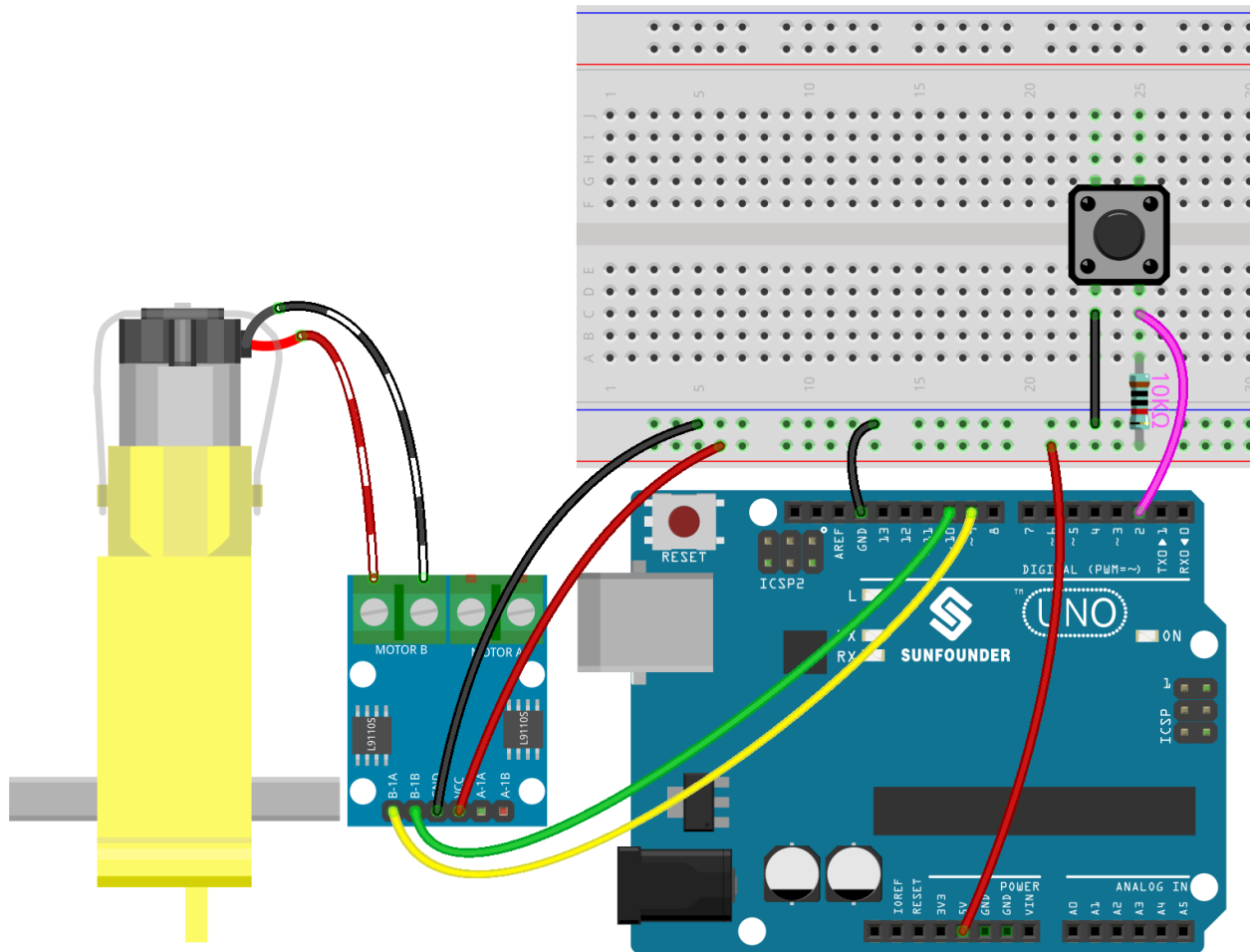
Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Knopf</i>	
<i>TT Motor</i>	-
<i>L9110 Motortreibermodul</i>	-

Schaltplan



Verkabelung



Code

Bemerkung:

- Öffnen Sie die Datei 5.3.state_change_detection.ino im Pfad 3in1-kit\basic_project\5.3.state_change_detection.
- 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, drücken Sie den Knopf und der Motor wird sich drehen; bis Sie den Knopf erneut drücken, wird der Motor stoppen.

Wie funktioniert das?

1. Variablen erstellen und Pins für den Motor und den Knopf definieren.

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

- detectionState ist ein Flag, dessen Wert bei jedem Tastendruck geändert wird, z.B. 0 dieses Mal, 1 beim nächsten Mal, und so weiter im Wechsel.

- `buttonState` und `lastButtonState` werden verwendet, um den aktuellen und den letzten Zustand des Knopfes zu speichern, um zu vergleichen, ob der Knopf gedrückt oder losgelassen wurde.

2. Jeden Pin initialisieren und die Baudrate des seriellen Monitors festlegen.

```
void setup() {
  pinMode(buttonPin, INPUT);
  Serial.begin(9600);
  pinMode(B_1A, OUTPUT);
  pinMode(B_1B, OUTPUT);
}
```

3. Zuerst den Zustand des Knopfes lesen, und wenn der Knopf gedrückt wird, wechselt die Variable `detectionState` ihren Wert von 0 auf 1 oder von 1 auf 0. Wenn `detectionState` 1 ist, wird der Motor eingeschaltet. Das hat den Effekt, dass der Motor beim ersten Drücken des Knopfes startet, beim nächsten Drücken stoppt und so weiter im Wechsel.

```
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(B_1A,HIGH);
    digitalWrite(B_1B,LOW);
  }else{
    digitalWrite(B_1A,LOW);
    digitalWrite(B_1B,LOW);
  }
}
```

Der gesamte Arbeitsablauf ist wie folgt.

- Den Knopfwert lesen.

```
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 der Variable `lastButtonState`. `delay(50)` wird verwendet, um Jitter zu eliminieren.

```
if (buttonState != lastButtonState) {
  ...
  delay(50);
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
}
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. nach einer Operation von 0 auf 1.

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

- Wenn die Variable `detectionState` 1 ist, lässt den Motor drehen, ansonsten stoppen.

```
if(detectionState==1){
  digitalWrite(B_1A,HIGH);
  digitalWrite(B_1B,LOW);
}else{
  digitalWrite(B_1A,LOW);
  digitalWrite(B_1B,LOW);
}
```

5.5.4 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()` anhält. Wenn die Taste gedrückt wird, während Arduino auf das Ende des `delay()` wartet, wird Ihr Programm den Tastendruck übersehen.

Ein Vergleich 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. Das Analogon zur Verwendung von `delay()` wäre, vor der Mikrowelle zu sitzen und den Timer von 10 Minuten herunterzählen zu sehen, bis der Timer null erreicht. Wenn die wichtige E-Mail in dieser Zeit ankommt, werden Sie sie verpassen.

Was Sie im wirklichen Leben tun würden, wäre die Pizza einzuschalten, dann Ihre E-Mails zu überprüfen und dann vielleicht etwas anderes zu tun (das nicht zu lange dauert!) und von Zeit zu Zeit würden Sie zur Mikrowelle zurückkommen, um zu sehen, ob der Timer null erreicht hat, was bedeutet, dass Ihre Pizza fertig ist.

Dieses Sketch zeigt, wie man einen Summer ohne Verwendung von `delay()` tönt. Es schaltet den Summer ein und notiert dann die Zeit. Dann überprüft es bei jedem Durchlauf durch `loop()`, ob das gewünschte Intervall vergangen ist. Wenn dies der Fall ist, tönt es den Summer und notiert die neue Zeit. Auf diese Weise tönt der Summer kontinuierlich, während die Sketch-Ausführung nie an einer einzigen Anweisung hängen bleibt.

Aufgrund dieses Zustands können wir den Code der Taste hinzufügen, um die LED zu steuern, sie wird nicht durch den Summer, der Musik spielt, gestört.

Erforderliche Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

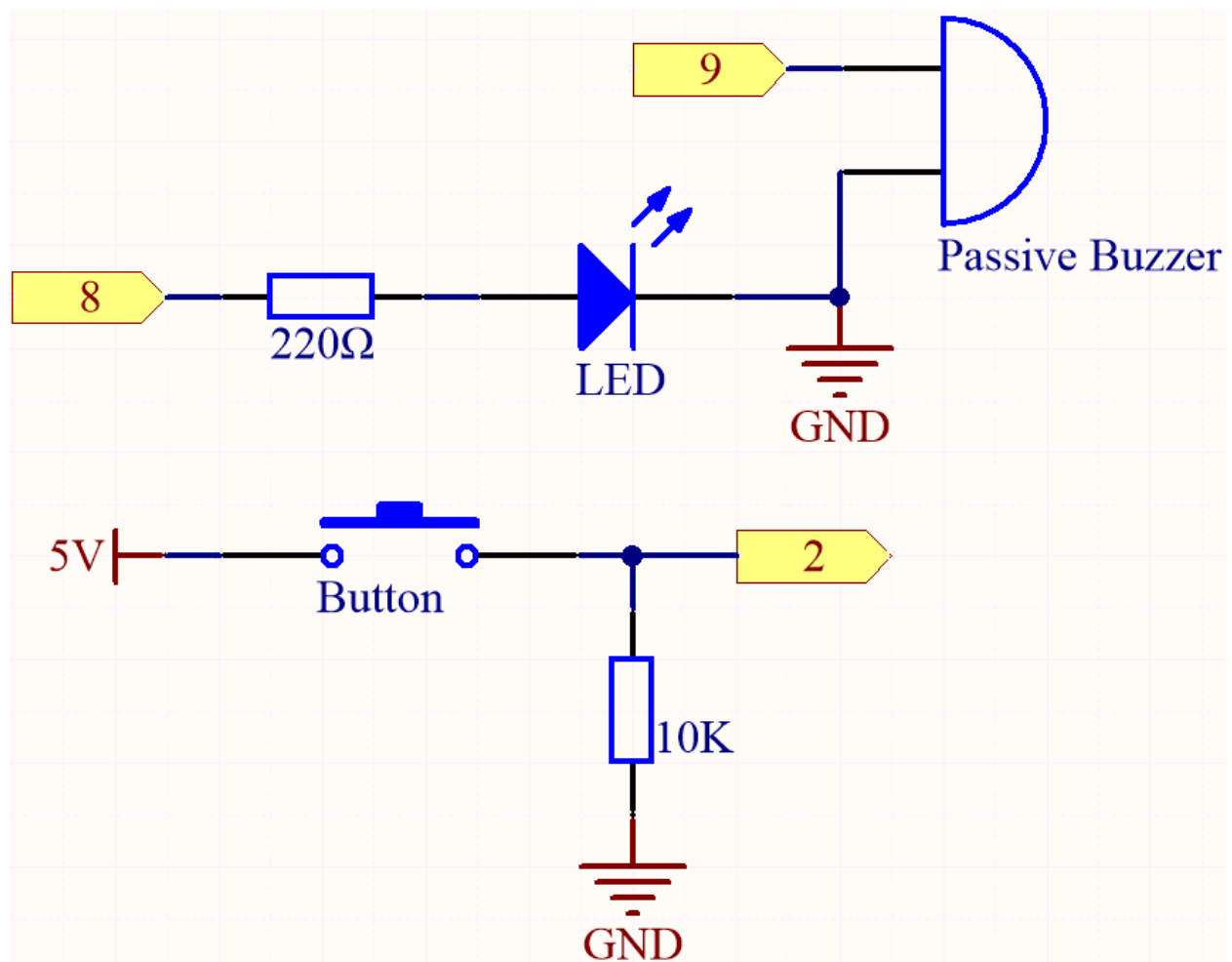
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

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

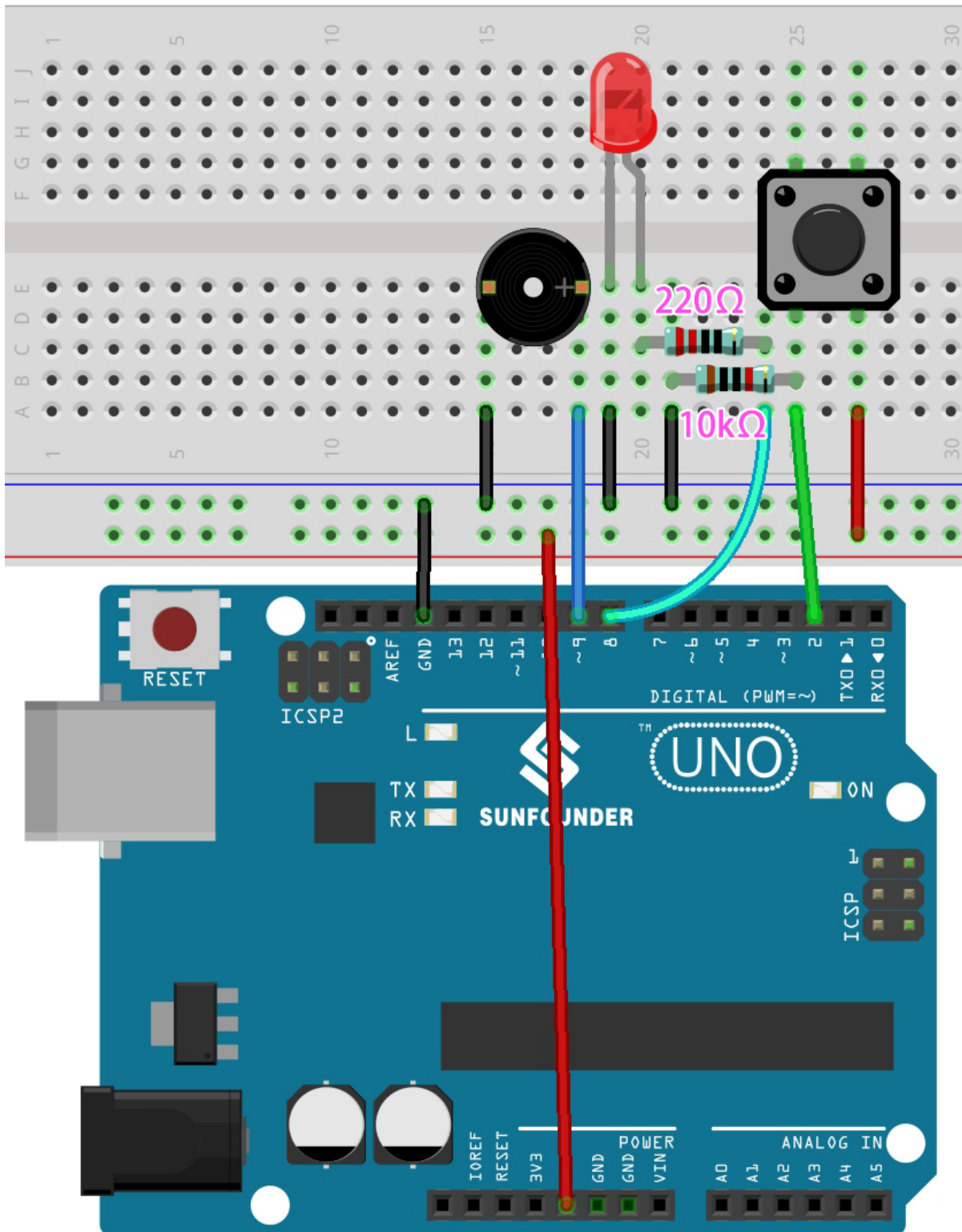
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Knopf</i>	
<i>Summer</i>	

Schaltplan



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei `5.4.interval.ino` unter dem Pfad `3in1-kit/basic_project/5.4.interval`.
 - 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, wird der Summer Musik abspielen; jedes Mal, wenn Sie den Knopf drücken, leuchtet die LED auf. Die Funktion von LED und Summer stört sich nicht gegenseitig.

Wie funktioniert es?

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;
```

Das Intervall 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. Aktualisieren Sie gleichzeitig die `previousMillis` auf die aktuelle Zeit für das nächste Auslösen, das 1 Sekunde später erfolgen soll.

```
if (currentMillis - previousMillis >= interval) {  
    previousMillis = currentMillis; // speichern Sie die letzte Zeit des letzten Tons  
    //...  
}
```

Spielen Sie die Noten der Melodie nacheinander ab.

```
tone(buzzerPin,melody[thisNote],100);  
interval=1000/noteDurations[thisNote]; // Intervall zum Tönen  
thisNote=(thisNote+1)%(sizeof(melody)/2); // Iterieren über die Noten der Melodie
```

Der Knopf steuert die LED.

```
// Knopf & LED steuern  
digitalWrite(ledPin,digitalRead(buttonPin));
```

5.5.5 5.5 Interne Bibliothek verwenden

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

In diesem Projekt wird die Servo Bibliothek verwendet, um das Servo anzusteuern, damit es sich zwischen 0° und 180° drehen kann.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

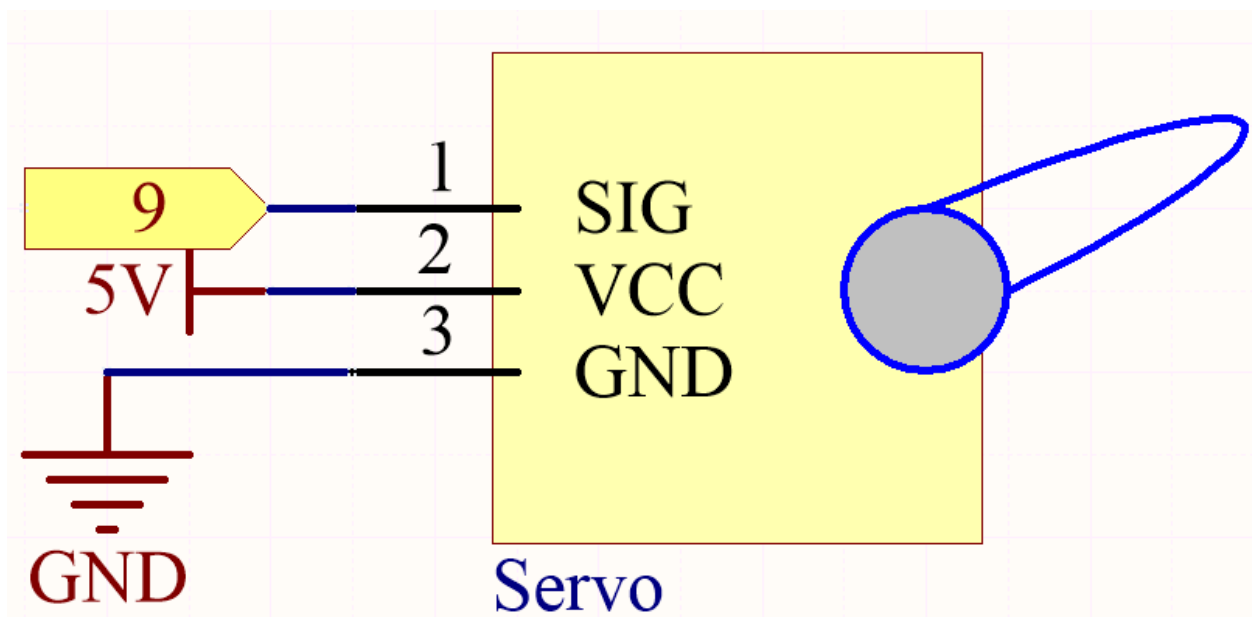
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

Sie können sie auch einzeln über die untenstehenden Links kaufen.

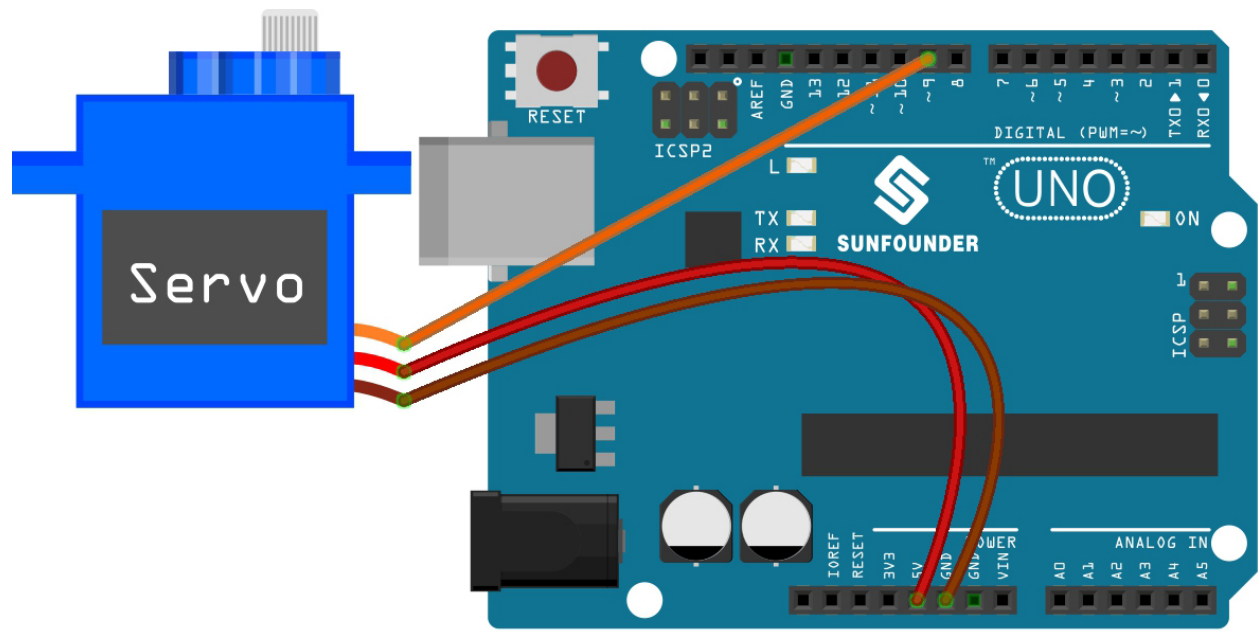
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>Servo</i>	

Schaltplan



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

Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.5.use_internal_library.ino unter dem Pfad 3in1-kit\basic_project\5.5.use_internal_library.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nachdem Sie die Codes auf das R3-Board hochgeladen haben, können Sie sehen, wie der Servoarm sich im Bereich von 0° bis 180° dreht.

Wie funktioniert das?

Durch den Aufruf der Bibliothek Servo.h können Sie das Servo einfach steuern.

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

Bibliotheksfunktionen

Servo

Erstellen Sie ein **Servo** Objekt, um ein Servo zu steuern.

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

Rufen Sie pinMode() auf, um einen Pin zu einem Servotreiber zu machen und geben Sie 0 bei Fehlern zurück.

```
void detach();
```

Geben Sie einen Pin vom Servo-Antrieb frei.

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

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

```
int read();
```

Geben Sie den mit dem letzten `write()` gesetzten Wert zurück.

```
bool attached();
```

Gibt 1 zurück, wenn das Servo aktuell angeschlossen ist.

5.5.6 5.6 Karte (Map)

Wenn Sie genau hinsehen, 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 Ausgabewinkel des Servos liegt bei (0~180).

Das bedeutet, wenn wir das Potentiometer verwenden möchten, um die Helligkeit der LED oder den Winkel des Servos zu steuern, müssen wir eine Zuordnungsoperation durchführen.

Lassen Sie uns nun sehen, wie das erreicht werden kann.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

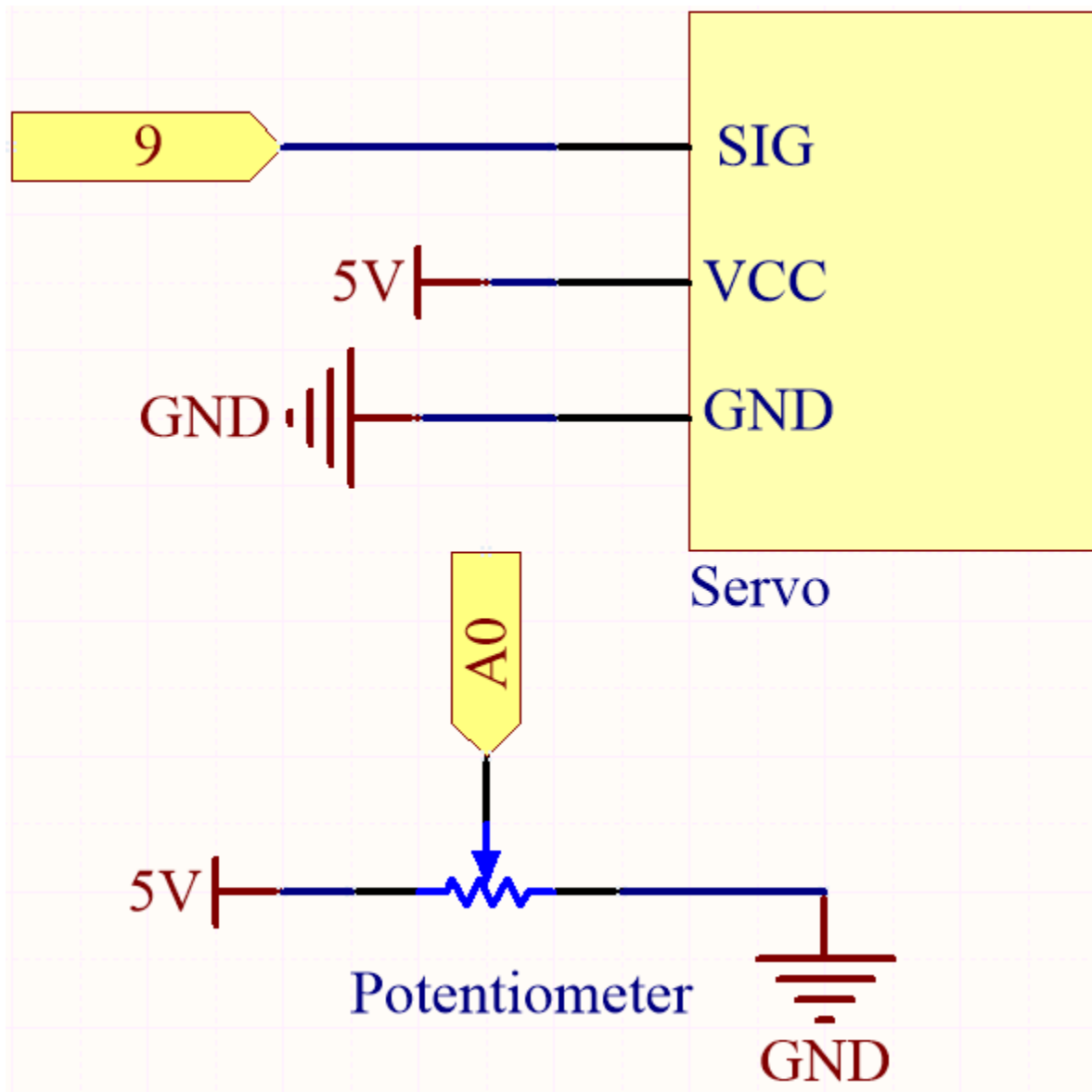
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

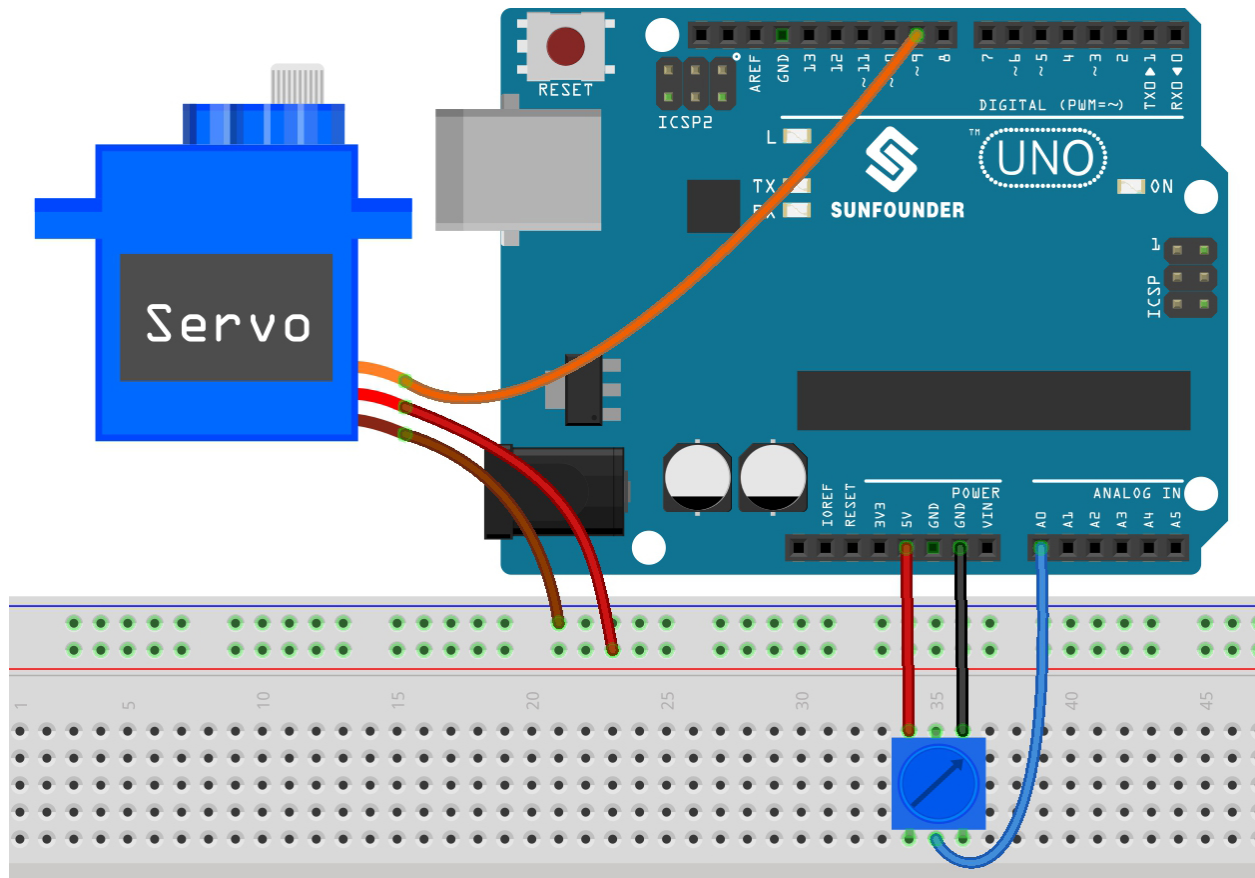
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENEINFÜHRUNG	KAUF-LINK
<i>SunFounder R3 Platine</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\basic_project\5.6.map.
- 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, können Sie das Potentiometer hin und her drehen, und die Ausgangswelle des Servos wird hin und her drehen.

Wie funktioniert das?

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

Syntax

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

Parameter

- `value`: die zuzuordnende Zahl.
- `fromLow`: die untere Grenze des aktuellen Wertebereichs.
- `fromHigh`: die obere Grenze des aktuellen Wertebereichs.
- `toLow`: die untere Grenze des Zielwertebereichs.

- toHigh: die obere Grenze des Zielwertebereichs.

Wenn das Potentiometer die LED steuert, können Sie auch die Zuordnung 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 als die „obere Grenze“ sein, was bedeutet, dass die map() Funktion verwendet werden kann, um einen Zahlenbereich umzukehren.

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

- Die Zuordnung funktioniert auch gut für negative Zahlen.

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

- Die Zuordnung verwendet Ganzzahlen und die Dezimalstellen von Fließkommazahlen werden verworfen.

5.5.7 5.7 Tone() oder noTone()

Tone() wird verwendet, um eine Rechteckwelle der angegebenen Frequenz (und 50% Tastverhältnis) an einem Pin zu erzeugen. Es kann eine Dauer angegeben werden, ansonsten wird die Welle fortgesetzt, bis ein Aufruf von noTone() erfolgt.

In diesem Projekt verwenden Sie diese beiden Funktionen, um den passiven Summer zum Schwingen und somit zum Tönen zu bringen. Wie der aktive Summer nutzt auch der passive Summer das Phänomen der elektromagnetischen Induktion für seine Funktion. Der Unterschied besteht darin, dass ein passiver Summer keine Oszillationsquelle besitzt, weshalb er bei Verwendung von Gleichstromsignalen nicht piepst. Dies ermöglicht es jedoch dem passiven Summer, seine eigene Oszillationsfrequenz anzupassen und verschiedene Töne wie „doh, re, mi, fa, sol, la, ti“ auszugeben.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

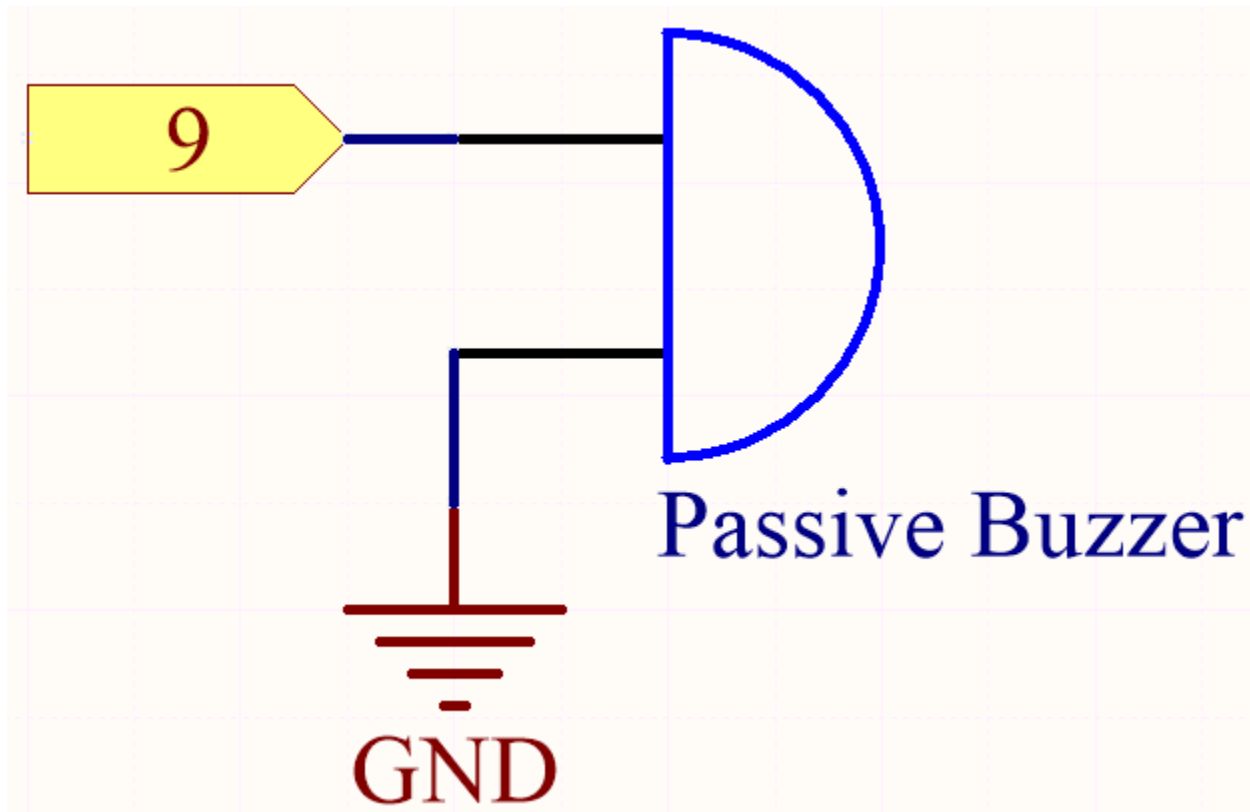
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

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

Sie können sie auch einzeln über die untenstehenden Links kaufen.

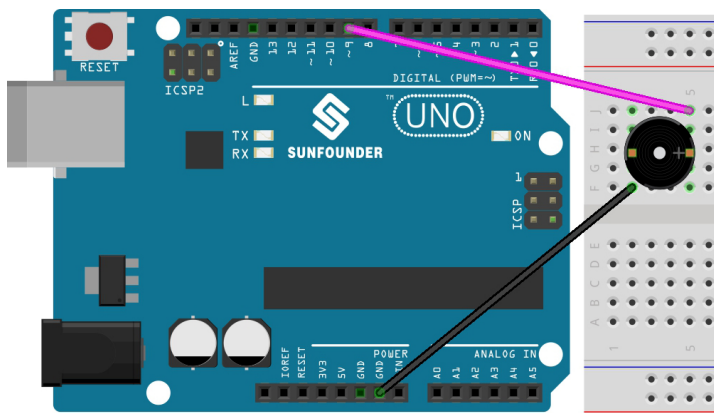
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Summer</i>	

Schaltplan



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

Verkabelung



Code

Bemerkung:

- Öffnen Sie die Datei 5.7.tone_notone.ino im Pfad 3in1-kit\basic_project\5.7.tone_notone.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Wenn Sie den Code erfolgreich auf das R3-Board hochgeladen haben, können Sie eine Melodie aus sieben Tönen hören.

Wie funktioniert das?

Es gibt zwei Punkte, die Sie beachten sollten:

1. `tone()` & `noTone()`: Diese Funktion wird verwendet, um den Ton des passiven Summers direkt zu steuern. Der Prototyp lautet:

Syntax

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

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

Parameter

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

Erzeugt eine Rechteckwelle der angegebenen Frequenz (und 50% Tastverhältnis) an einem Pin, damit der passive Summer klingt. Die Welle geht weiter, bis ein Aufruf von `noTone()` erfolgt. Der Pin kann mit einem Piezo-Summer oder einem anderen Lautsprecher verbunden werden, um Töne zu erzeugen.

Es kann nur ein Ton gleichzeitig erzeugt werden. Wenn bereits ein Ton auf einem anderen Pin abgespielt wird, hat der Aufruf von `tone()` keinen Effekt. Wenn der Ton auf demselben Pin abgespielt wird, wird seine Frequenz eingestellt.

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

Es ist nicht möglich, Töne unter 31Hz zu erzeugen.

Syntax

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

Parameter

pin: Der Arduino-Pin, an dem der Ton erzeugt wird.

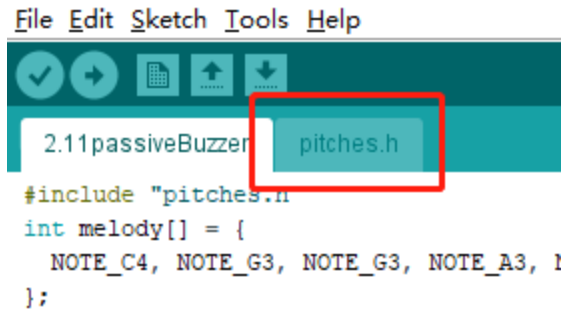
Stoppt die Erzeugung einer durch `tone()` ausgelösten Rechteckwelle. Hat keinen Effekt, wenn kein Ton erzeugt wird.

Nachdem Sie die beiden Funktionen kennen, sollten Sie den Code verstehen können. Die Installation der Arrays `melody[]` und `noteDurations[]` bereitet die anschließenden Aufrufe der Funktion `tone()` und die Änderung von Ton und Dauer in der Schleife für einen besseren Musikeffekt vor.

2. **pitches.h**: Der Code verwendet eine zusätzliche Datei namens `pitches.h`. Diese Datei enthält alle Tonwerte 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 verfasst, auf dessen Arbeit der Befehl `tone()` basiert. Sie könnte nützlich sein, wenn Sie musikalische Noten erzeugen möchten.

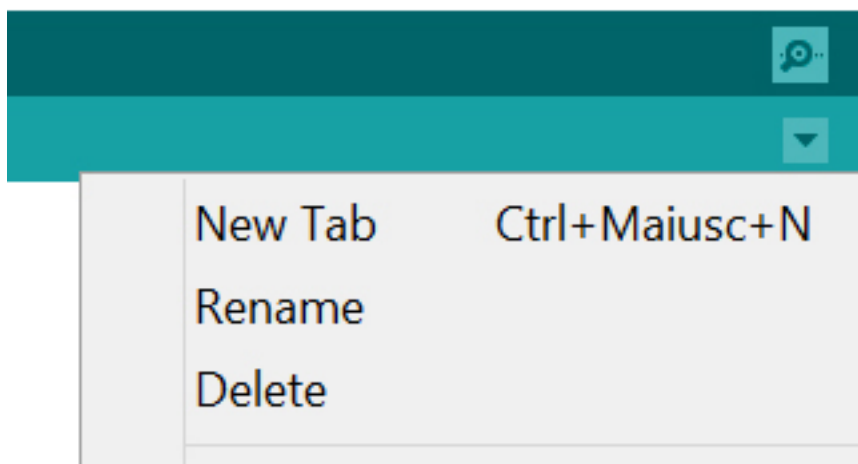
```
#include "pitches.h"
```

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 zur Installation von `pitches.h` weggelassen werden.



Wenn Sie die Code-Datei öffnen und die `pitches.h` Datei nicht öffnen können, können Sie sie 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 **Neuer Tab** oder verwenden Sie **Ctrl+Shift+N**.



Dann fügen Sie den folgenden Code ein und speichern 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
#define NOTE_B1  62
#define NOTE_C2  65
#define NOTE_CS2 69
#define NOTE_D2  73
#define NOTE_DS2 78

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
#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
#define NOTE_DS6 1245
#define NOTE_E6 1319
#define NOTE_F6 1397
#define NOTE_FS6 1480
#define NOTE_G6 1568
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
#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
```

5.5.8 5.8 Benutzerdefinierte Funktion

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

Die folgenden Vorteile ergeben sich durch die Nutzung 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 überall in einem Programm aufrufen.
- Ein großes C-Programm lässt sich leichter verfolgen, wenn es in mehrere Funktionen unterteilt ist.
- Wiederverwendbarkeit ist das Hauptziel von C-Funktionen.
- Dennoch stellt der Aufruf von Funktionen immer eine Belastung in einem C-Programm dar.

In der C-Programmierung gibt es zwei Arten von Funktionen:

- **Bibliotheksfunktionen:** Funktionen, die in den C-Headerdateien deklariert sind.
- **Benutzerdefinierte Funktionen:** Funktionen, die vom C-Programmierer erstellt werden, so dass er/sie sie mehrmals verwenden kann. Dies reduziert die Komplexität eines großen Programms und optimiert den Code.

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

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

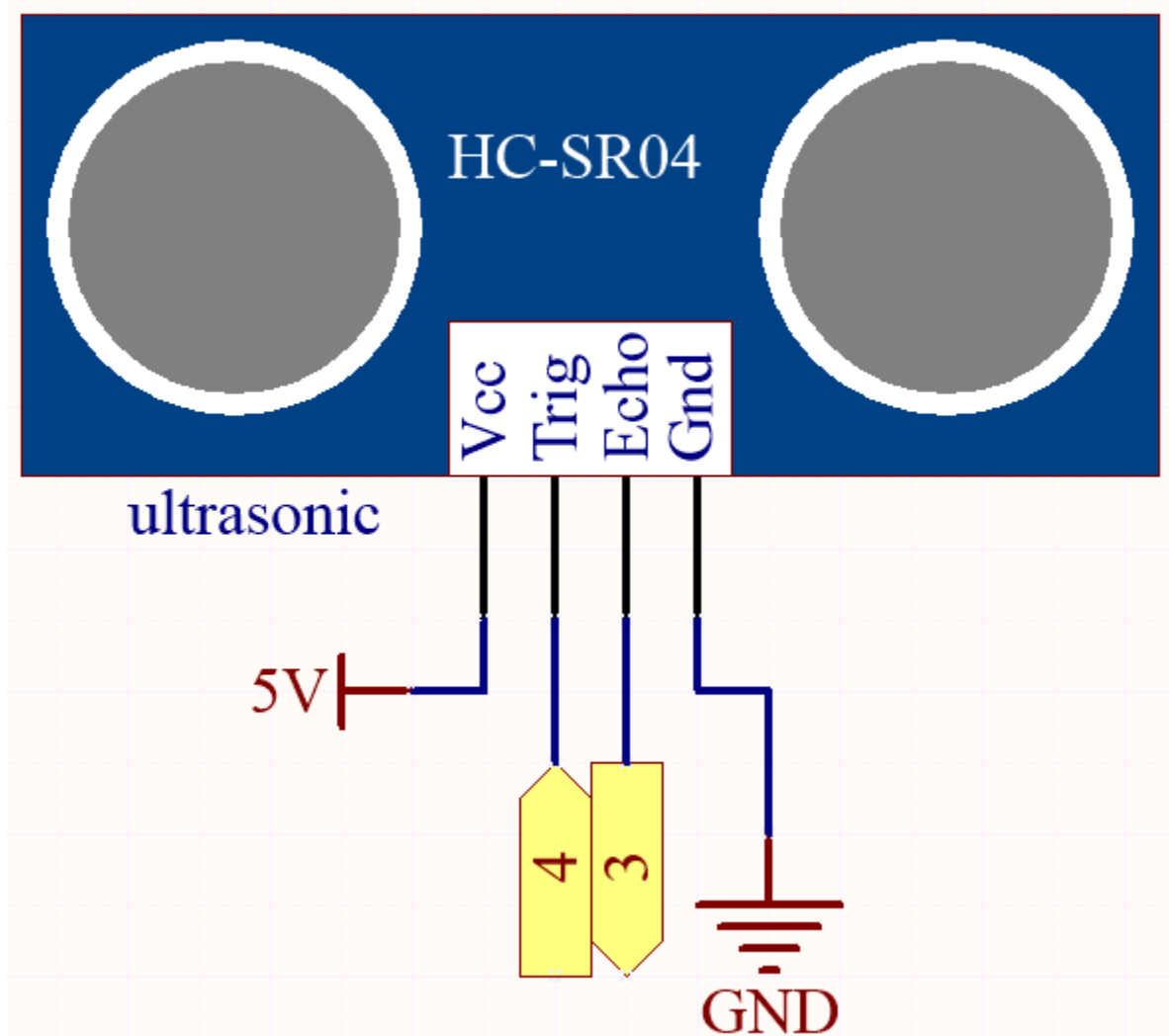
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

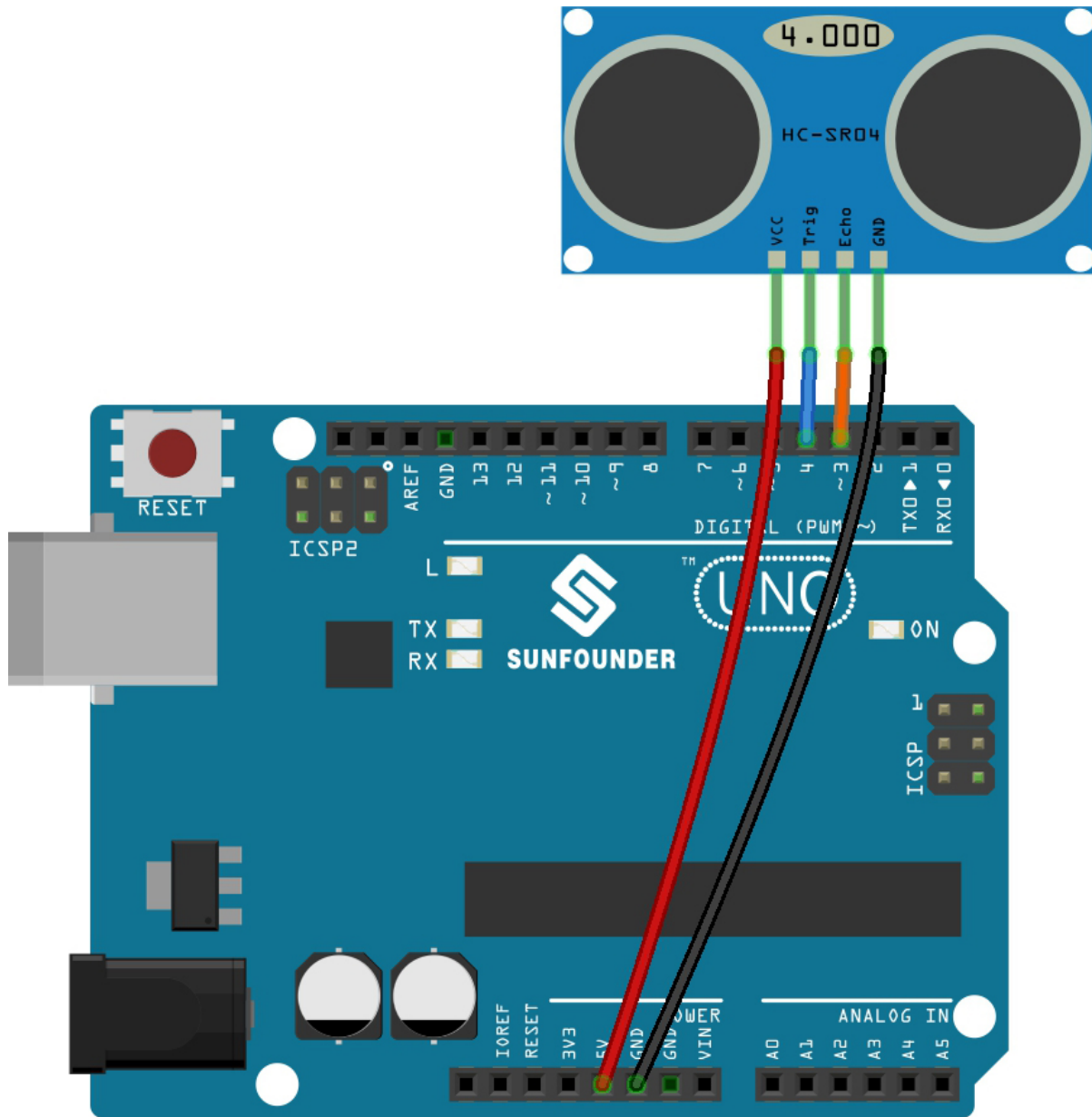
Die Komponenten können auch einzeln über die folgenden Links erworben werden:

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</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\basic_project\5.8.user_function`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nach erfolgreichem Hochladen des Codes wird der serielle Monitor die Entfernung zwischen dem Ultraschallsensor und dem Hindernis vor ihm anzeigen.

Wie funktioniert das?

Zur Anwendung des Ultraschallsensors können wir direkt die Unterfunktion überprüfen.

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

Das trigPin des Ultraschallmoduls sendet alle 2us ein 10us Quadratsignal.

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

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

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

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

Das ergibt die vom Quadratsignal zurückgelegte Entfernung, sowohl hin als auch zurück, daher teilen wir durch 2, um die Entfernung des Hindernisses zu ermitteln.

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

Beachten Sie, dass der Ultraschallsensor das Programm anhält, wenn er arbeitet, was bei komplexen Projekten zu Verzögerungen führen kann.

5.5.9 5.9 ShiftOut(LED)

shiftOut() lässt 74HC595 8 digitale Signale ausgeben. Das letzte Bit der Binärzahl wird an Q0 und das erste Bit an Q7 ausgegeben. In anderen Worten, wenn man die Binärzahl „00000001“ schreibt, wird Q0 auf hohem Niveau ausgegeben und Q1~Q7 auf niedrigem Niveau.

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

Insbesondere kann der 74hc595 durch Schreiben einer 8-Bit-Binärzahl 8 Pins für die digitale Signalabgabe ersetzen.

- [Binärzahl - Wikipedia](#)

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

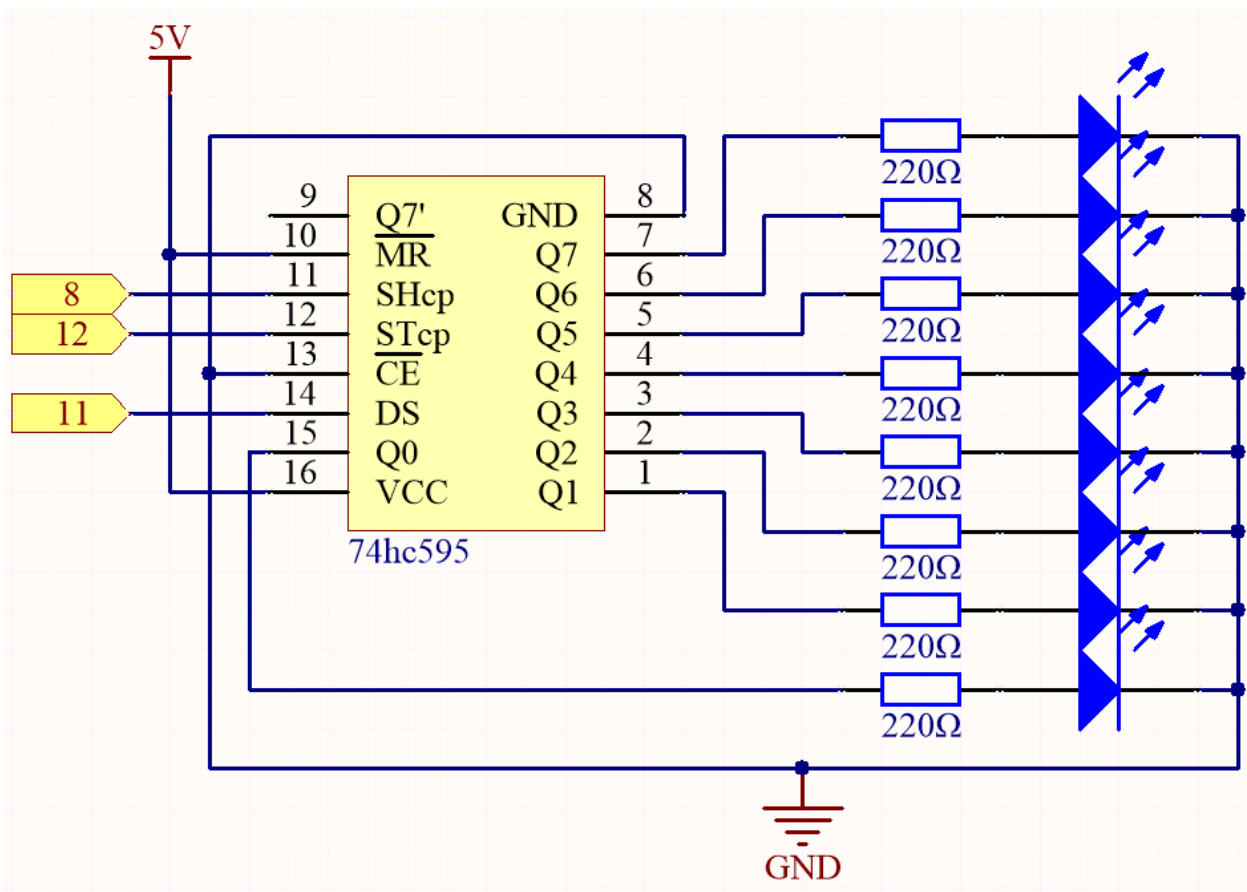
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

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

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENEINFÜHRUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>74HC595</i>	

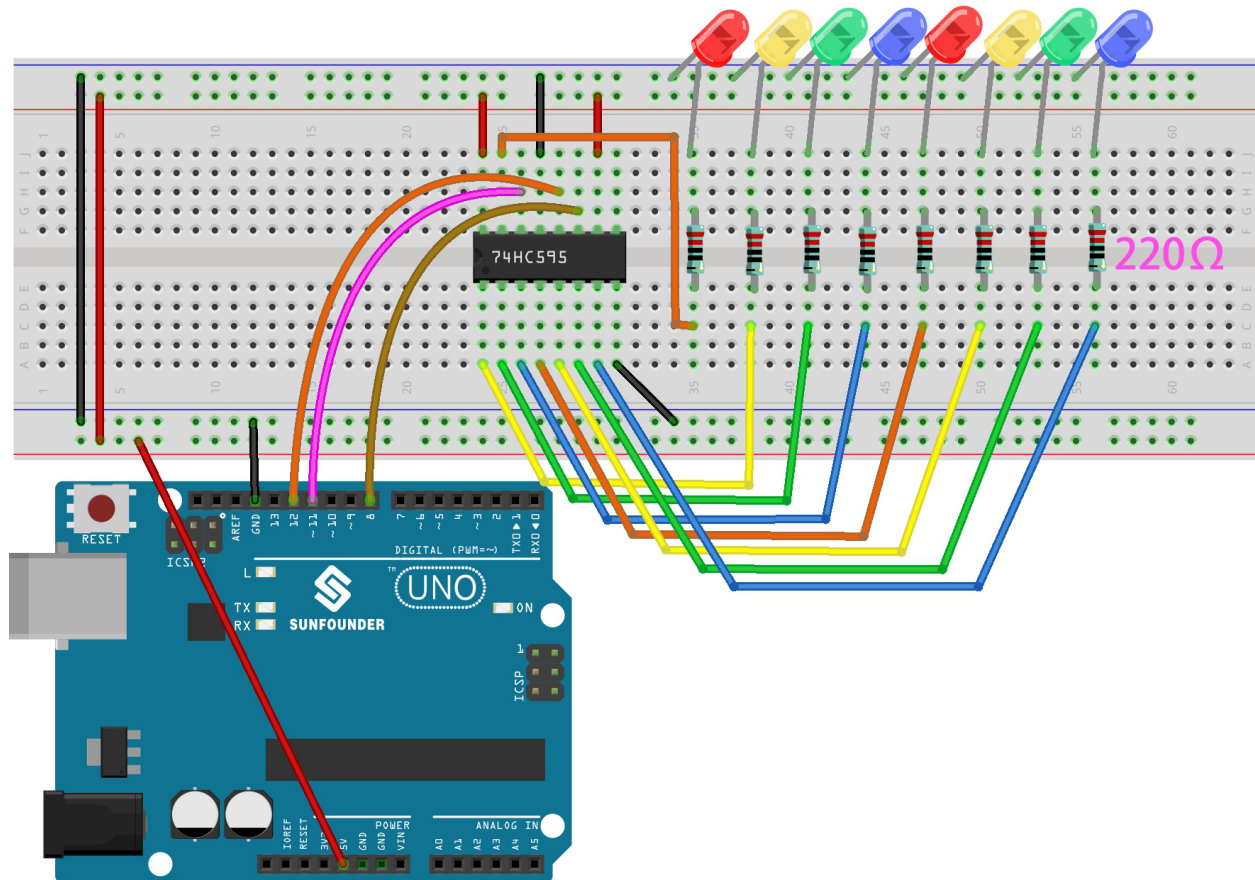
Schaltplan



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

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

Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.9.shiftout_led.ino im Pfad 3in1-kit\basic_project\5.9.shiftout_led.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nachdem Sie den Code erfolgreich auf das R3-Board hochgeladen haben, sehen Sie, dass die LEDs nacheinander leuchten.

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. Dies erzeugt eine ansteigende Flanke von STcp.

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

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

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

Nach digitalWrite(STcp,HIGH) wird STcp im ansteigenden Takt sein. Zu diesem Zeitpunkt werden die Daten im Schieberegister ins Speicherregister verschoben.

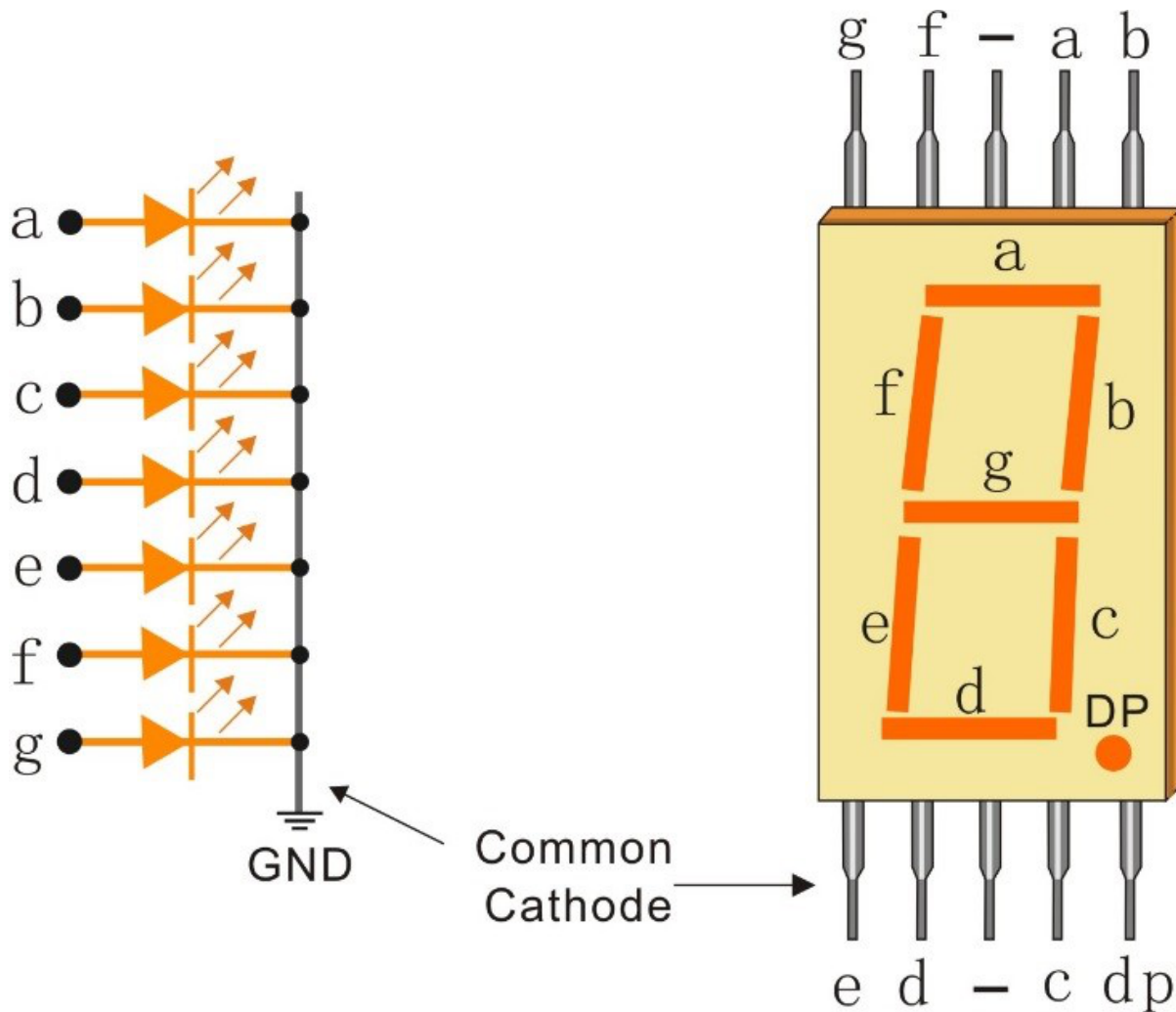
```
digitalWrite(STcp,HIGH);
```

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

5.5.10 5.10 ShiftOut(Segmentanzeige)

In einem früheren Beispiel haben wir die Funktion shiftout() verwendet, um acht LEDs zu beleuchten. Hier verwenden wir sie, um die Zahlen 0-9 auf der 7-Segment-Anzeige darzustellen.

Die 7-Segment-Anzeige ist im Grunde ein Gerät, das aus 8 LEDs besteht. 7 davon sind streifenförmige LEDs, die eine „8“ formen, und es gibt eine etwas kleinere LED als Dezimalpunkt. Diese LEDs sind als a, b, c, d, e, f, g und dp markiert. Sie verfügen über eigene Anoden-Pins und teilen sich Kathoden. Ihre Pin-Positionen sind in der untenstehenden Abbildung dargestellt.



Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Bauteile.

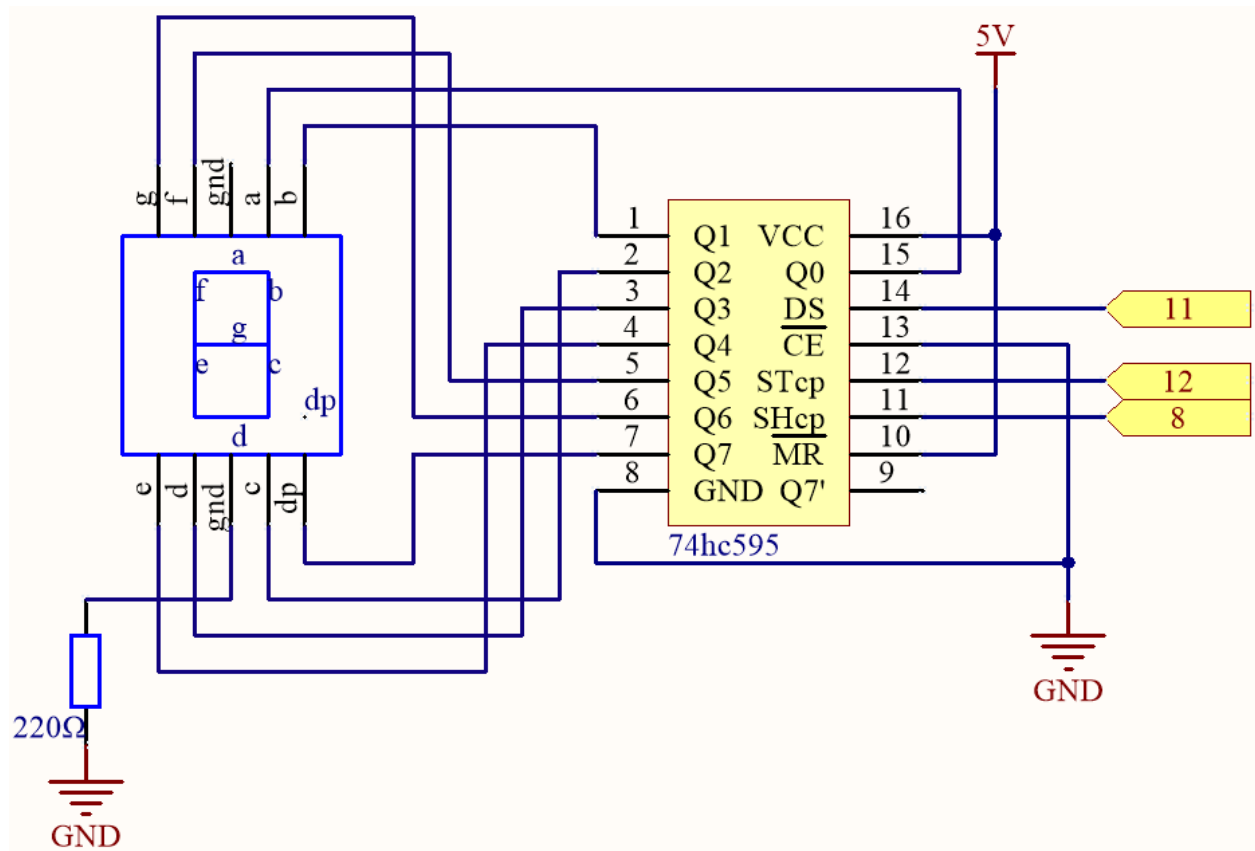
Es ist durchaus praktisch, ein komplettes Set zu kaufen. Hier ist der Link:

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

Sie können diese auch einzeln über die folgenden Links erwerben.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>7-Segment-Anzeige</i>	
<i>74HC595</i>	

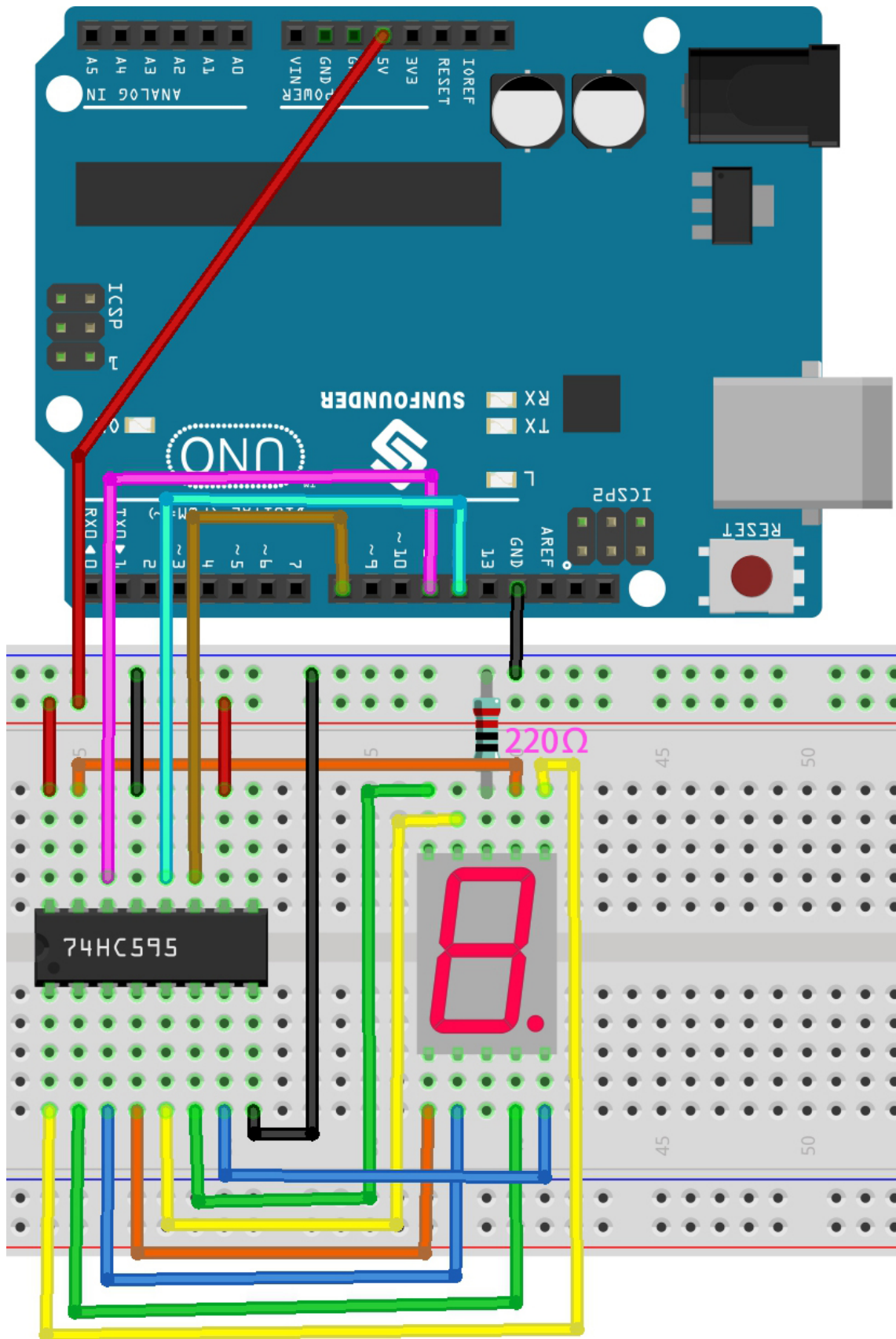
Schaltplan



Verdrahtung

Tab. 1: Verdrahtung

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



Code

Bemerkung:

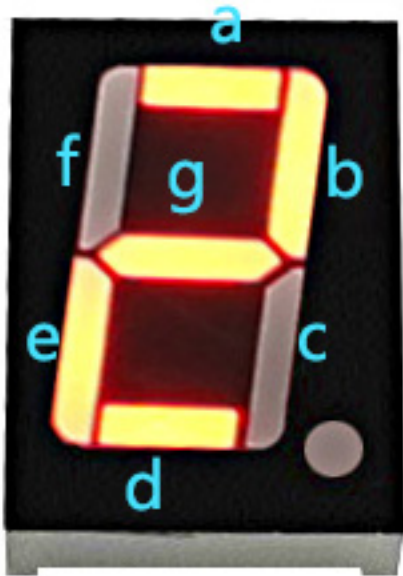
- Öffnen Sie die Datei `5.10.shiftout_segment.ino` im Pfad `3in1-kit/basic_project/5.10.shiftout_segment`.
 - Oder kopieren Sie diesen Code in die **Arduino IDE**.
 - Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.
-

Nachdem der Code erfolgreich hochgeladen wurde, können Sie beobachten, wie die LED Segmentanzeige die Zahlen 0~9 nacheinander anzeigt.

Wie funktioniert das?

`shiftOut()` lässt den 74HC595 8 digitale Signale ausgeben. Es gibt das letzte Bit der Binärzahl an Q0 aus und das erste Bit an Q7. Das bedeutet, dass das Schreiben der Binärzahl „00000001“ dazu führt, dass Q0 ein hohes Signal und Q1~Q7 ein niedriges Signal ausgibt.

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



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

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

Tab. 2: Zeichen Code

Zahlen	Binär 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.

5.5.11 5.11 Externe Bibliotheken installieren

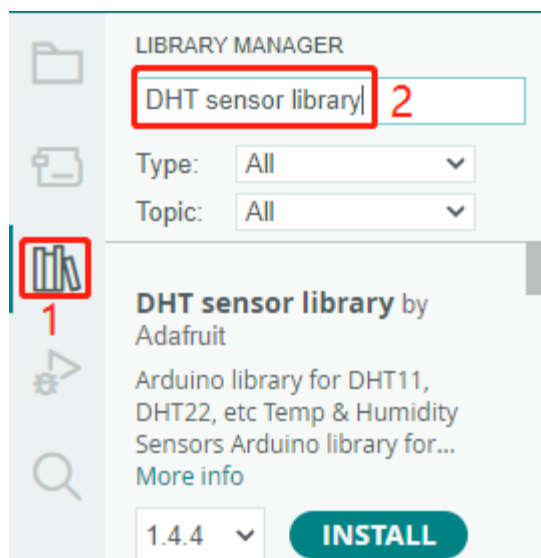
Eine Bibliothek ist eine Sammlung von vorgeschriebenem Code oder Funktionen, die die Fähigkeiten der Arduino IDE erweitern. Bibliotheken bieten sofort einsatzbereiten Code für verschiedene Funktionen und ermöglichen es Ihnen, Zeit und Aufwand beim Programmieren komplexer Features zu sparen.

Es gibt zwei Hauptwege, Bibliotheken zu installieren:

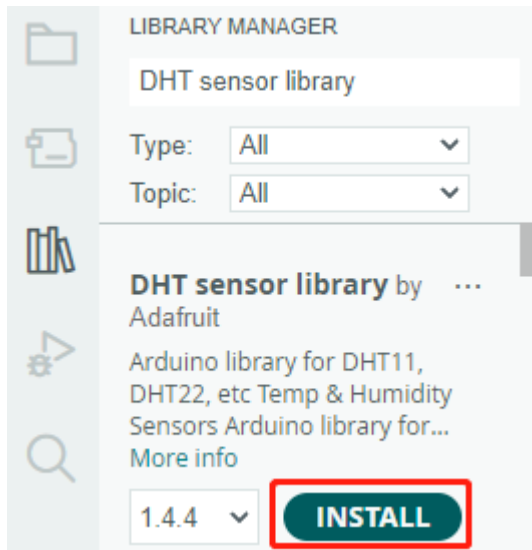
Viele Bibliotheken sind direkt über den Arduino **Library Manager** verfügbar. Sie können den **Library Manager** mit den folgenden Schritten aufrufen:

1. Im **Library Manager** können Sie nach der gewünschten Bibliothek suchen, indem Sie den Namen eingeben oder durch verschiedene Kategorien blättern.

Bemerkung: In Projekten, bei denen die Installation von Bibliotheken erforderlich ist, gibt es Hinweise, welche Bibliotheken zu installieren sind. Befolgen Sie die bereitgestellten Anweisungen, wie zum Beispiel „Die DHT-Sensor-Bibliothek wird hier verwendet, Sie können sie aus dem **Library Manager** installieren.“ Installieren Sie einfach die empfohlenen Bibliotheken wie angegeben.



2. Sobald Sie die zu installierende Bibliothek gefunden haben, klicken Sie darauf und anschließend auf den **Install**-Button.



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

Verwandte Komponenten

Im Folgenden finden Sie die verwandten Komponenten. Sie können darauf klicken, um zu erfahren, wie Sie diese verwenden.

5.11.1 Flüssigkristallanzeige

Ein I2C LCD1602 setzt sich aus einem LCD1602 und einem I2C-Modul zusammen. Das LCD1602 kann zur Anzeige von Zeichen, Zahlen usw. verwendet werden, benötigt jedoch viele Pins des Hauptcontrollers. Nach der Konfiguration eines I2C-Moduls sind nur noch 2 I/O-Pins erforderlich, um dieses LCD1602 anzusteuern.

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

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten:

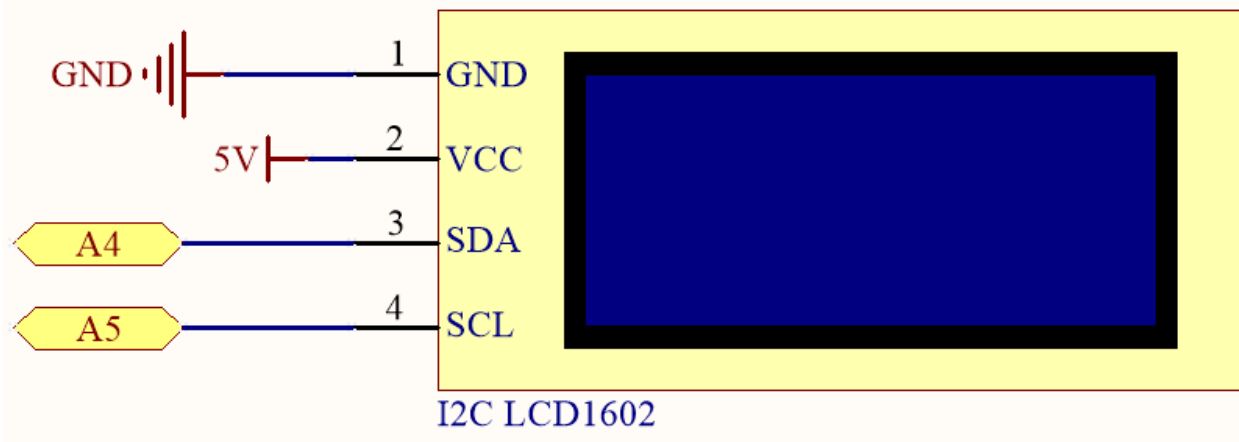
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

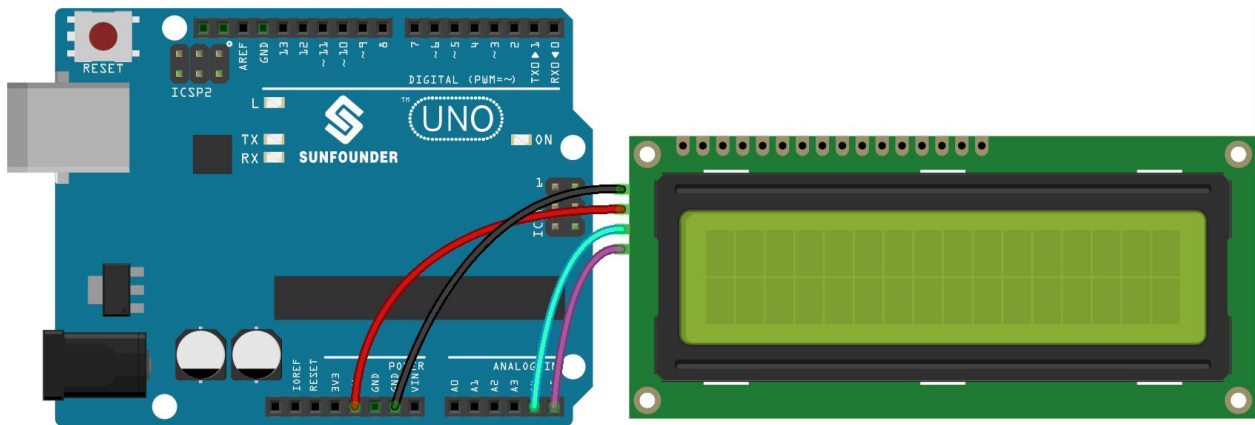
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>I2C LCD1602</i>	

Schaltplan



Verdrahtung

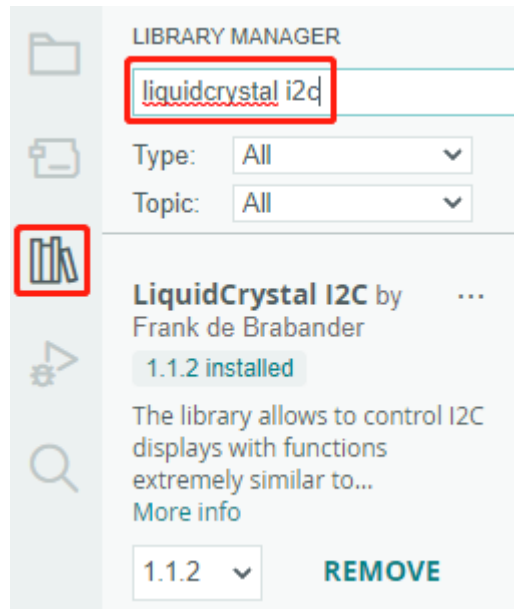


Bemerkung: Die SDA- und SCL-Pins des R3-Boards entsprechen den Pins A4 und A5.

Code

Bemerkung:

- Öffnen Sie die Datei `5.11.liquid_crystal_display.ino` unter dem Pfad `3in1-kit/basic_project/5.11.liquid_crystal_display`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Hier wird die `LiquidCrystal I2C`-Bibliothek verwendet, die Sie aus dem **Library Manager** installieren können.



Nach erfolgreichem Hochladen des Codes werden Sie „SunFounder“, „Hallo Welt“ auf dem I2C LCD1602 sehen.

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

Wie funktioniert das?

Durch den Aufruf der Bibliothek `LiquidCrystal_I2C.h` können Sie das LCD einfach ansteuern.

```
#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 LCD ist standardmäßig auf 0x27 eingestellt.
- `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 Cursorposition auf Null.

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

Setzt die Cursorposition auf col,row.

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

Gibt Text auf dem LCD aus.

- **data:** Die auszugebenden 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, wie Sie einen IR-Empfänger verwenden.

Ein Infrarot-Empfänger ist eine Komponente, die Infrarotsignale empfängt. Er kann Infrarotstrahlen eigenständig empfangen und Signale ausgeben, die mit dem TTL-Level kompatibel sind. Er ähnelt von der Größe her einem normal verpackten Transistor und eignet sich für alle Arten von Infrarot-Fernbedienungen und Infrarotübertragungen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

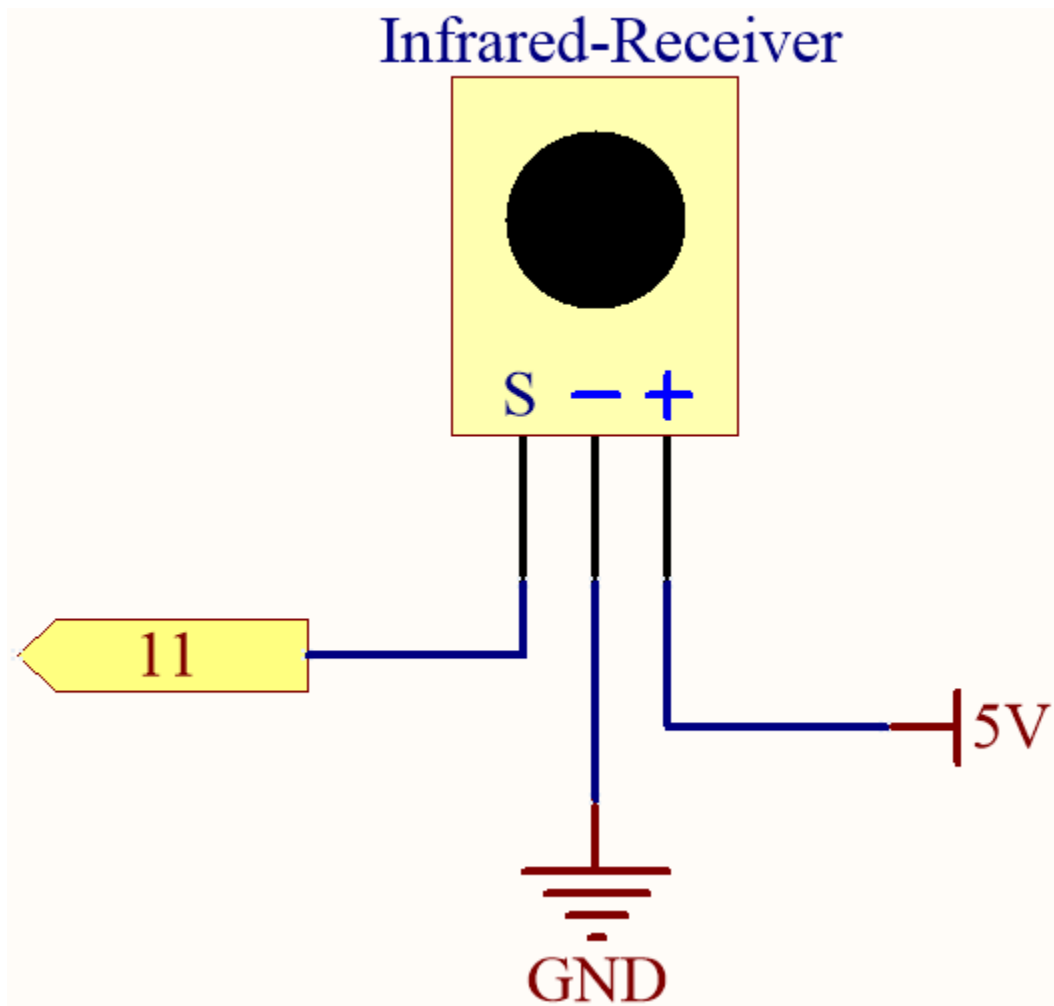
Es ist definitiv praktisch, ein ganzes Kit zu kaufen. Hier ist der Link:

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

Sie können diese auch separat über die untenstehenden Links kaufen.

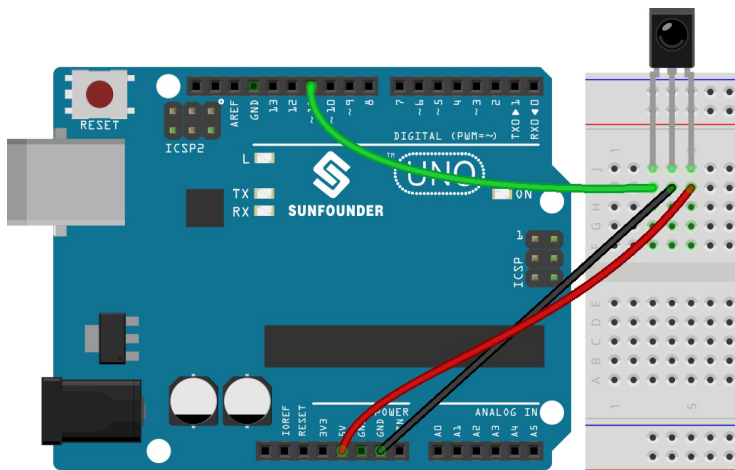
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>IR-Empfänger</i>	-

Schaltplan



Verkabelung

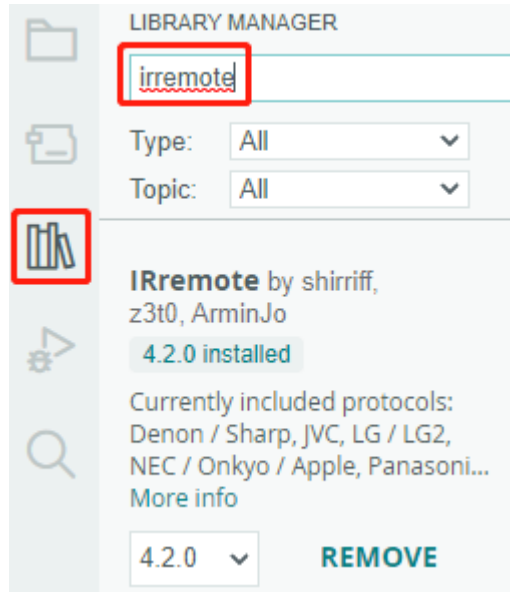
In diesem Beispiel verbinden wir den linken Pin des IR-Empfängers mit Pin 11, den mittleren Pin mit GND und den rechten Pin mit 5V.



Code

Bemerkung:

- Öffnen Sie die Datei 5.11.ir_receiver.ino unter dem Pfad 3in1-kit\basic_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 R3-Board hochgeladen haben, können Sie den aktuellen Wert der gedrückten Taste der IR-Fernbedienung auf dem Seriellen Monitor anzeigen lassen.

Wie funktioniert das?

Dieser Code ist für die Verwendung mit einer Infrarot (IR) Fernbedienung und der IRremote Bibliothek entwickelt. Hier ist die Erklärung:

1. Einbindung von Bibliotheken: Dies beinhaltet die IRremote Bibliothek, welche Funktionen zur Arbeit mit IR-Fernbedienungen bereitstellt.

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

2. Definiert den Arduino-Pin, an den der Signal-Pin des IR-Sensors angeschlossen ist.

```
const int IR_RECEIVE_PIN = 11; // Pinnummer für den IR-Sensor definieren
```

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

```
void setup() {
  Serial.begin(9600); // Serielle
  // Kommunikation mit 9600 Baudrate starten
  IrReceiver.begin(IR_RECEIVE_PIN, ENABLE_LED_FEEDBACK); // IR-Empfänger
  // starten
}
```

4. Die Schleife läuft kontinuierlich, 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 decodiert wurde.
- Decodiert den IR-Befehl und speichert ihn in decodedValue mit einer benutzerdefinierten Funktion decodeKeyValue().
- Überprüft, ob der decodierte Wert kein Fehler ist.
- Gibt den decodierten IR-Wert auf dem seriellen Monitor aus.
- Setzt den IR-Signalempfang für das nächste Signal fort.

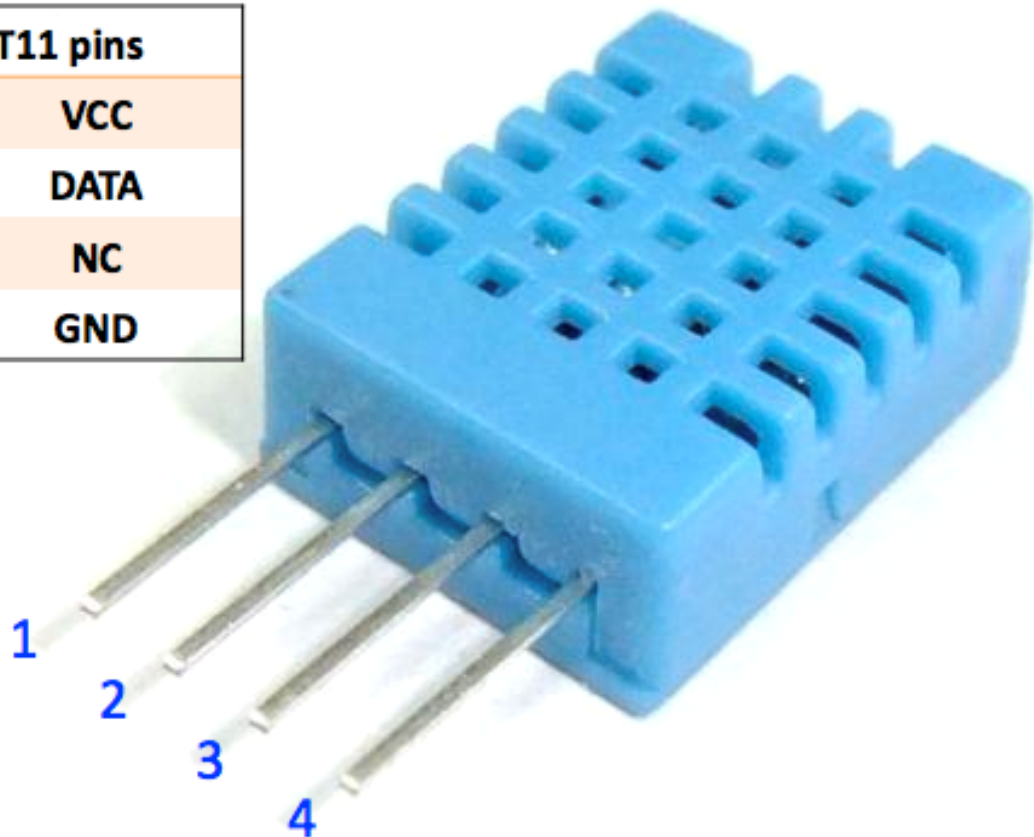
5.11.3 Temperatur - Luftfeuchtigkeit

Temperatur und Luftfeuchtigkeit sind sowohl von der physikalischen Größe her als auch im täglichen Leben der Menschen eng miteinander verbunden. Die Temperatur und Luftfeuchtigkeit unserer Umgebung beeinflusst direkt die thermoregulatorische Funktion und den Wärmeübertragungseffekt des menschlichen Körpers. Dies kann weiterhin die Denkaktivität und den mentalen Zustand beeinflussen und somit unsere Lern- und Arbeitsleistung beeinträchtigen.

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

Die Luftfeuchtigkeit gibt die Konzentration des in der Luft enthaltenen Wasserdampfs an. Die relative Luftfeuchtigkeit, häufig im Alltag verwendet, wird in %RH ausgedrückt. Sie hängt eng mit der Temperatur zusammen. Für ein bestimmtes Volumen eingeschlossenen Gases gilt: Je höher die Temperatur, desto niedriger die relative Luftfeuchtigkeit und umgekehrt.

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



Das im Kit enthaltene dht11 ist ein digitaler Temperatur- und Luftfeuchtigkeitssensor. Er nutzt einen kapazitiven Feuchtigkeitssensor und einen Thermistor, um die Umgebungsluft zu messen und gibt ein digitales Signal am Datenpin aus.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

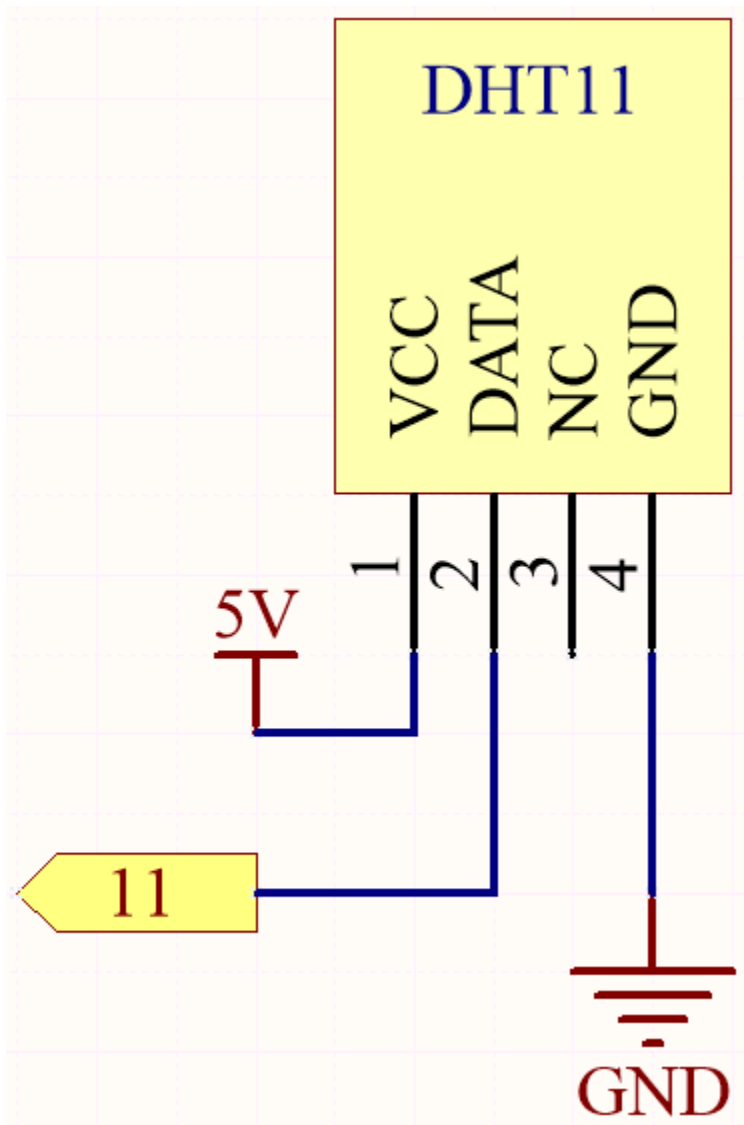
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

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

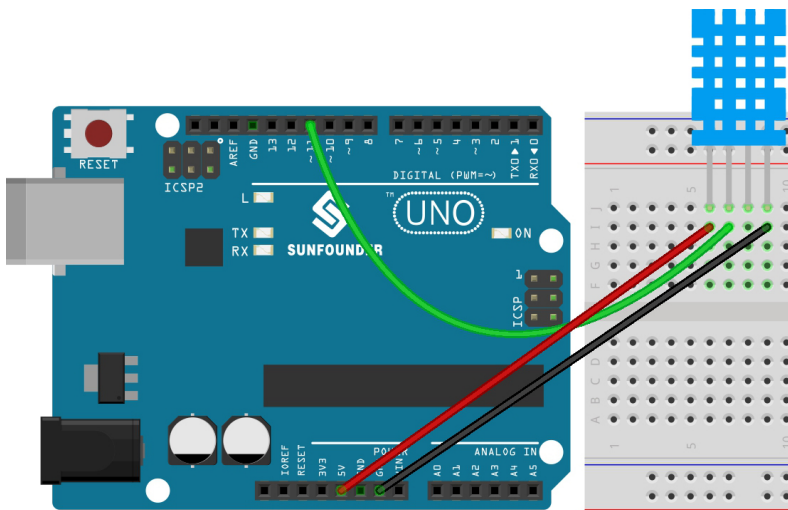
Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>DHT11 Feuchtigkeits- und Temperatursensor</i>	-

Schaltplan



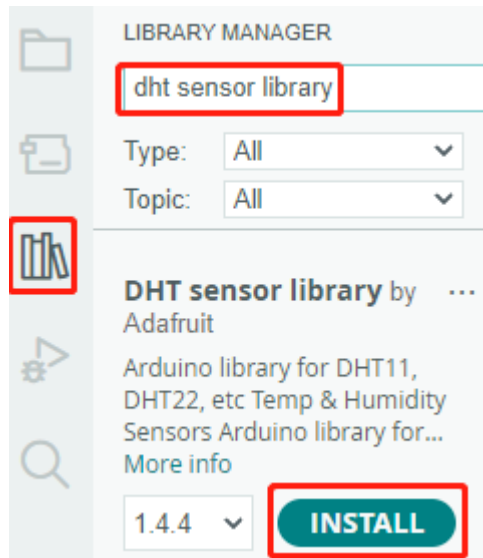
Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.11.temperature_humidity.ino im Pfad 3in1-kit\basic_project\5.11.temperature_humidity.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Hier wird die DHT Sensorbibliothek verwendet, die Sie im **Library Manager** installieren können.



Nachdem der Code erfolgreich hochgeladen wurde, werden Sie sehen, dass der Serielle Monitor kontinuierlich Temperatur und Luftfeuchtigkeit ausgibt. Mit fortschreitender Laufzeit des Programms werden diese Werte immer genauer.

Wie funktioniert das?

1. Fügt die DHT.h-Bibliothek hinzu, die Funktionen zur Interaktion mit den DHT-Sensoren bereitstellt. Anschließend legen Sie den Pin und den Typ für den DHT-Sensor fest.

```
#include "DHT.h"

#define DHTPIN 11 // Legt den mit dem DHT11-Datenpin verbundenen Pin fest
#define DHTTYPE DHT11 // DHT 11

DHT dht(DHTPIN, DHTTYPE);
```

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

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

3. In der Funktion loop() werden die Temperatur- und Luftfeuchtigkeitswerte vom DHT11-Sensor gelesen und auf dem seriellen Monitor ausgegeben.

```

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");
}

```

- Die Funktion `dht.readHumidity()` wird aufgerufen, um den Luftfeuchtigkeitswert vom DHT-Sensor zu lesen.
- Die Funktion `dht.readTemperature()` wird aufgerufen, um den Temperaturwert vom DHT-Sensor zu lesen.
- Die Funktion `isnan()` wird verwendet, um zu überprüfen, ob die Messwerte gültig sind. Wenn entweder der Luftfeuchtigkeits- oder Temperaturwert NaN (keine Zahl) ist, zeigt dies ein fehlerhaftes Lesen des Sensors an, und eine Fehlermeldung wird ausgegeben.

5.5.12 5.12 Serielle Lektüre

Vielleicht haben Sie dies bemerkt, als Sie die Funktion `Serial.print()` verwendet haben. Da es eine Ausgabe gibt, gibt es auch eine Eingabe? Wofür ist das Textfeld im Seriellen Monitor? Ja, Sie haben richtig geraten. Es gibt Möglichkeiten, Programme und Schaltungen durch Eingabe von Informationen über das Textfeld des seriellen Monitors zu steuern.

In diesem Projekt verwenden wir das I2C LCD1602, um den im Seriellen Monitor eingegebenen Text anzuzeigen und die Verwendung von `Serial.read()` kennenzulernen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

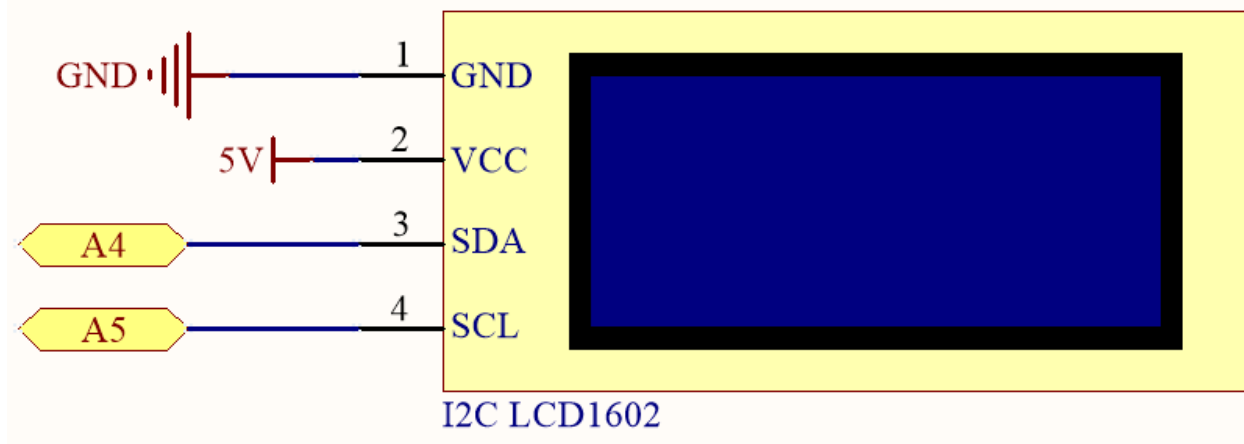
Es ist definitiv praktisch, ein komplettes Kit zu kaufen, hier ist der Link:

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

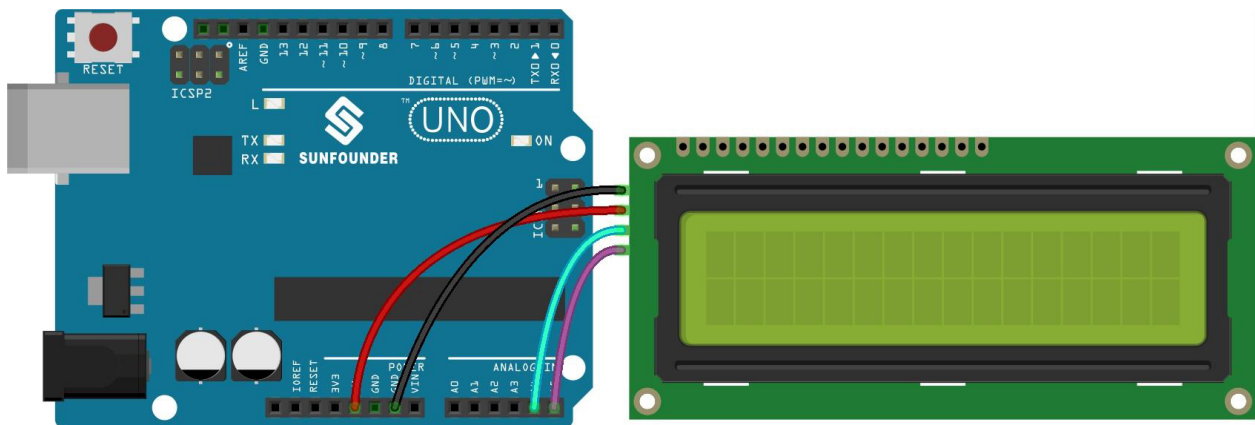
Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>I2C LCD1602</i>	

Schaltplan



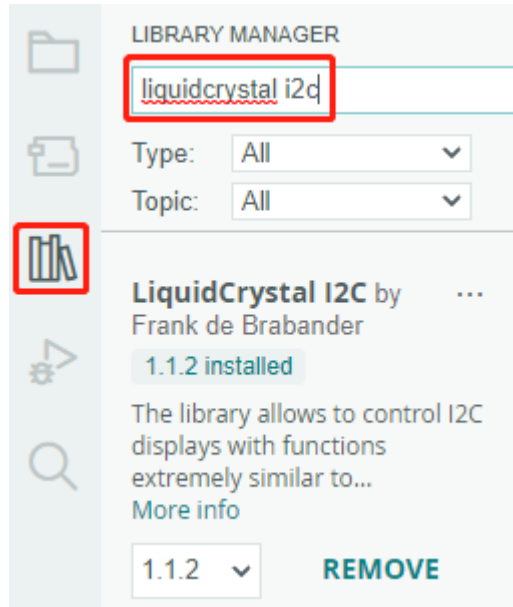
Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.12.serial_read.ino unter dem Pfad 3in1-kit\basic_project\5.12.serial_read.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Hier wird die Bibliothek LiquidCrystal I2C verwendet. Sie können sie aus dem **Library Manager** installieren.



Nachdem der Code erfolgreich hochgeladen wurde, können Sie Text in das Textfeld des seriellen Monitors eingeben, und das LCD zeigt die Informationen an.

Wie funktioniert das?

```
void loop()
{
  // wenn Zeichen über die serielle Schnittstelle ankommen...
  if (Serial.available()) {
    // warten Sie kurz, damit die gesamte Nachricht ankommt
    delay(100);
    // den Bildschirm leeren
    lcd.clear();
    // lesen Sie alle verfügbaren Zeichen
    while (Serial.available() > 0) {
      // zeigt jedes Zeichen auf dem LCD an
      lcd.write(Serial.read());
    }
  }
}
```

- `Serial.available()` kann die Anzahl der Zeichen in dem eingehenden Stream erhalten, wenn Sie etwas aus dem Textfeld eingeben. Da es in der Eingabe zwei Abschlusszeichen 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 zum Beispiel AB eingeben 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; das Aufrufen dieser Funktion, wenn keine Zeichen im Eingabestream verfügbar sind, führt zu einem Fehler.

Zusammenfassend ist es üblich, die beiden oben genannten Funktionen zu kombinieren und mit einer `while`-Schleife alle bei jedem Eingabevorgang eingegebenen Zeichen zu lesen.

```
while (Serial.available() > 0) {
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```
Serial.print(Serial.read());
}
```

Übrigens, wenn Sie `Serial.read()` nicht verwenden, um Zeichen aus dem eingehenden Stream zu erhalten, werden die Zeichen des eingehenden Streams aufgestapelt. Zum Beispiel, wenn Sie A gefolgt von AB eingeben, wird der eingehende Stream 7 Zeichen anhäufen.

5.5.13 5.13 Unterbrechung

Wenn Sie in einem Projekt, das Sensoren verwendet, einige `delay()`-Befehle nutzen, werden Sie vielleicht feststellen, dass bei Aktivierung dieser Sensoren das Programm eventuell keine Wirkung zeigt. Dies liegt daran, dass die Verzögerungsanweisung das Programm anhält und das Programm das vom Sensor an die Hauptsteuerplatine gesendete Signal nicht erfassen kann.

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 Prozess der Verwendung von Unterbrechungen zu erleben.

In der `loop()`-Funktion wird `delay(1000)` verwendet, um Sekunden zu zählen. Legen Sie den Knopf, um den Summer zu steuern, in den ISR, sodass er nicht durch die Verzögerung gestört wird und die Aufgabe reibungslos ausgeführt werden kann.

Bemerkung: ISRs sind spezielle Funktionen mit einigen einzigartigen Einschränkungen, die die meisten anderen Funktionen nicht haben. Ein ISR kann keine Parameter haben und sollte nichts zurückgeben. Im Allgemeinen sollte ein ISR so kurz und schnell wie möglich sein. Wenn Ihr Skript mehrere ISRs verwendet, kann immer nur einer gleichzeitig ausgeführt werden. Andere Unterbrechungen werden nach Beendigung des aktuellen in einer Reihenfolge ausgeführt, die von ihrer Priorität abhängt.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

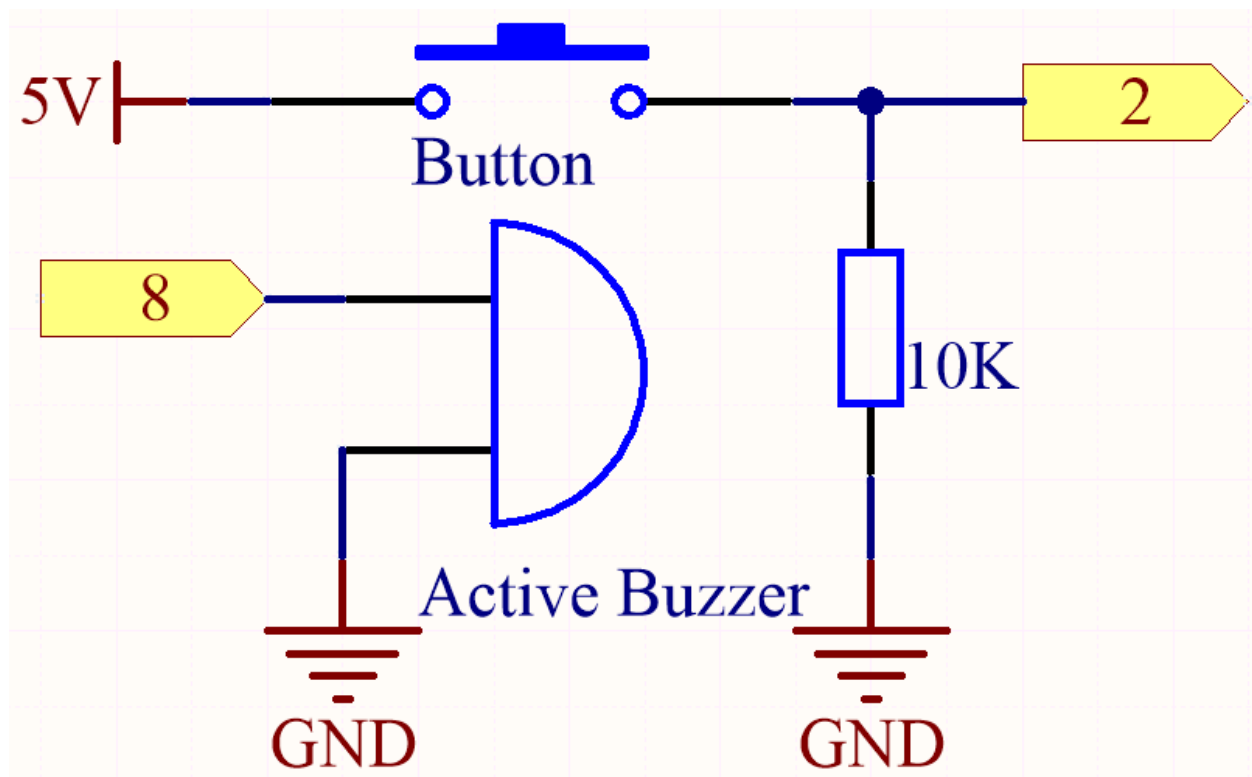
Es ist definitiv praktisch, ein ganzes Kit zu kaufen, hier ist der Link:

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

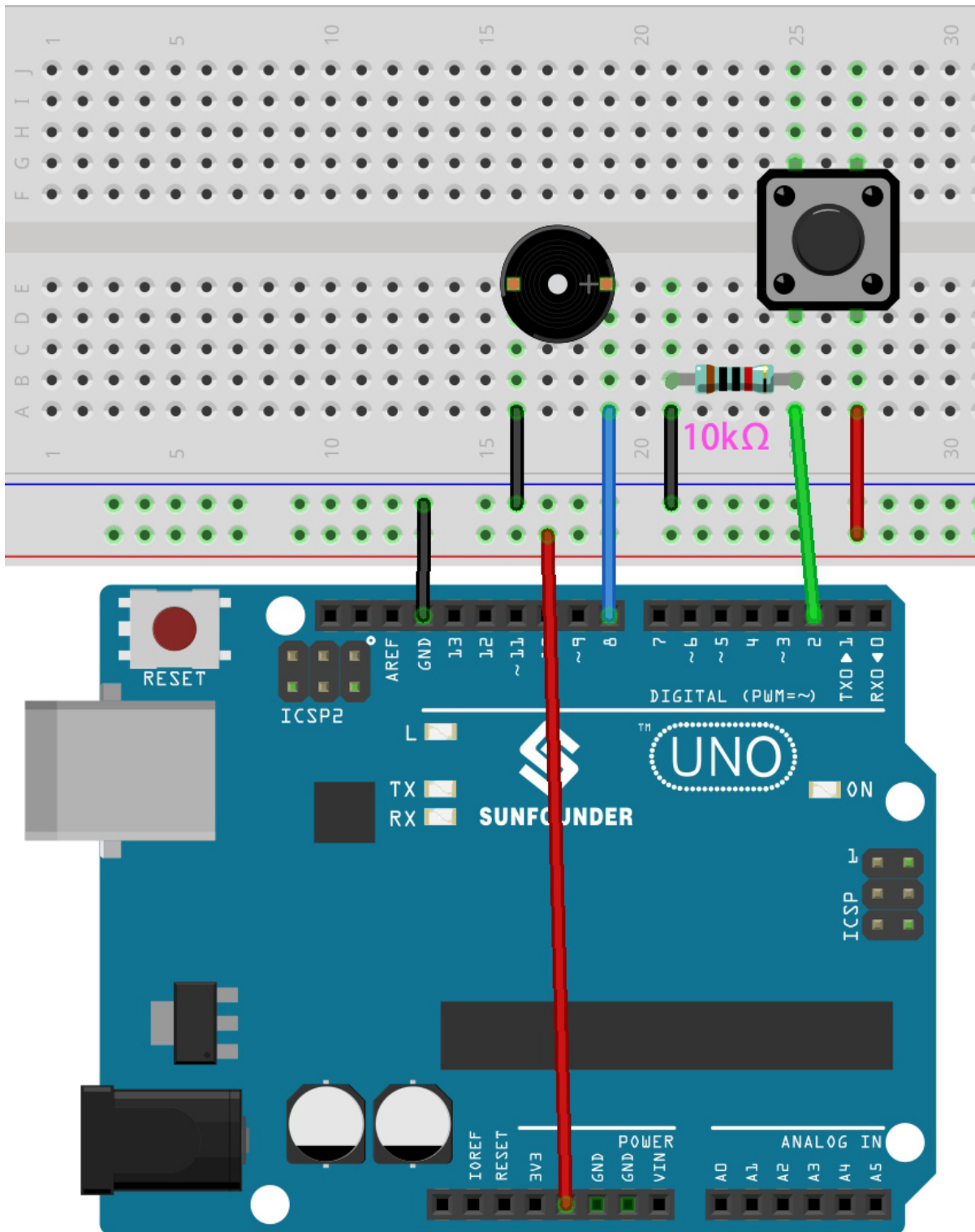
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Knopf</i>	
<i>Summer</i>	-

Schaltplan



Verkabelung



Code

Bemerkung:

- Öffnen Sie die Datei `5.13.interrupt.ino` im Pfad `3in1-kit/basic_project/5.13.interrupt`.

- 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, schalten Sie den Serial Monitor ein und Sie werden sehen, dass jede Sekunde eine automatisch hochzählende Zahl ausgegeben wird. Wenn Sie den Knopf drücken, gibt der Summer einen Ton aus. Die Knopf-gesteuerte Summerfunktion und die Zeitfunktion beeinflussen einander nicht.

Wie funktioniert das?

- `attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)`: Fügt einen Interrupt hinzu.

Syntax

```
attachInterrupt(digitalPinToInterrupt(pin), ISR, mode)
```

Parameter

- `pin`: Die Arduino-Pinnummer. Sie sollten `digitalPinToInterrupt(pin)` verwenden, um den eigentlichen Digitalpin in eine spezifische Interrupt-Nummer umzuwandeln. Wenn Sie zum Beispiel an Pin 3 anschließen, verwenden Sie `digitalPinToInterrupt(3)` als ersten Parameter.
- `ISR`: Der ISR, der aufgerufen wird, wenn der Interrupt auftritt; diese Funktion darf keine Parameter haben und nichts zurückgeben. Diese Funktion wird manchmal als Unterbrechungsbehandlungsroutine bezeichnet.
- `mode`: definiert, wann der Interrupt ausgelöst werden soll. Vier Konstanten sind als gültige Werte vordefiniert:
 - * `LOW` um den Interrupt auszulösen, wenn der Pin niedrig ist.
 - * `CHANGE` um den Interrupt auszulösen, wenn sich der Pinwert ändert.
 - * `RISING` um ihn auszulösen, wenn der Pin von niedrig zu hoch wechselt.
 - * `FALLING` um ihn auszulösen, wenn der Pin von hoch zu niedrig wechselt.

Bemerkung: Verschiedene Hauptsteuerkarten können Interrupt-Pins unterschiedlich verwenden. Auf der R3-Platine können nur Pin 2 und Pin 3 Unterbrechungen verwenden.

5.5.14 5.14 Kalibrierung

Wenn Sie analoge Eingangskomponenten wie Fotowiderstände, Bodenfeuchtigkeitssensoren usw. verwenden, stellen Sie möglicherweise fest, dass ihr Messbereich nicht 0 bis 1023 beträgt, sondern eher Bereiche wie 0 bis 800 oder 600 bis 1000, da es bei normalem Gebrauch unmöglich ist, die Grenzen dieser Geräte zu erreichen.

In solchen Fällen kann eine Technik zur Kalibrierung der Sensoreingänge angewendet werden. Beim Start sollte das Steuerbrett die Sensorwerte für fünf Sekunden messen und die höchsten und niedrigsten Werte aufzeichnen. Dieser fünfsekündige Messwert definiert die erwarteten Mindest- und Höchstwerte der Messwerte während des Zyklus.

In diesem Projekt verwenden wir einen Fotowiderstand und einen passiven Summer, um ein Spiel ähnlich dem [Theremin](#) unter Verwendung der oben beschriebenen Kalibrierungstechnik zu implementieren.

Bemerkung: Das [Theremin](#) ist ein elektronisches Musikinstrument, das keinen physischen Kontakt erfordert. Es erzeugt unterschiedliche Töne, indem es die Position der Hände des Spielers erfasst.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

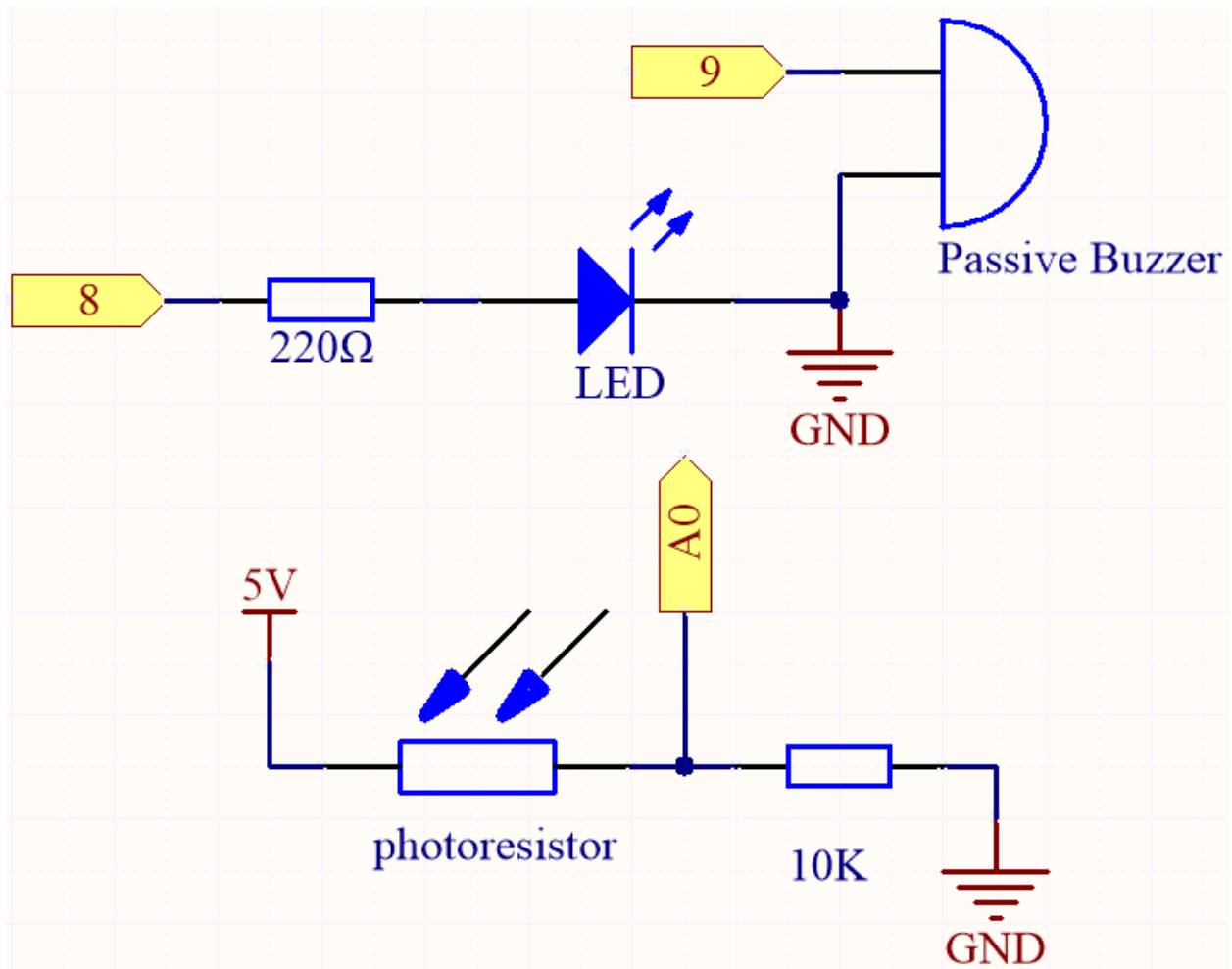
Es ist sicherlich praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

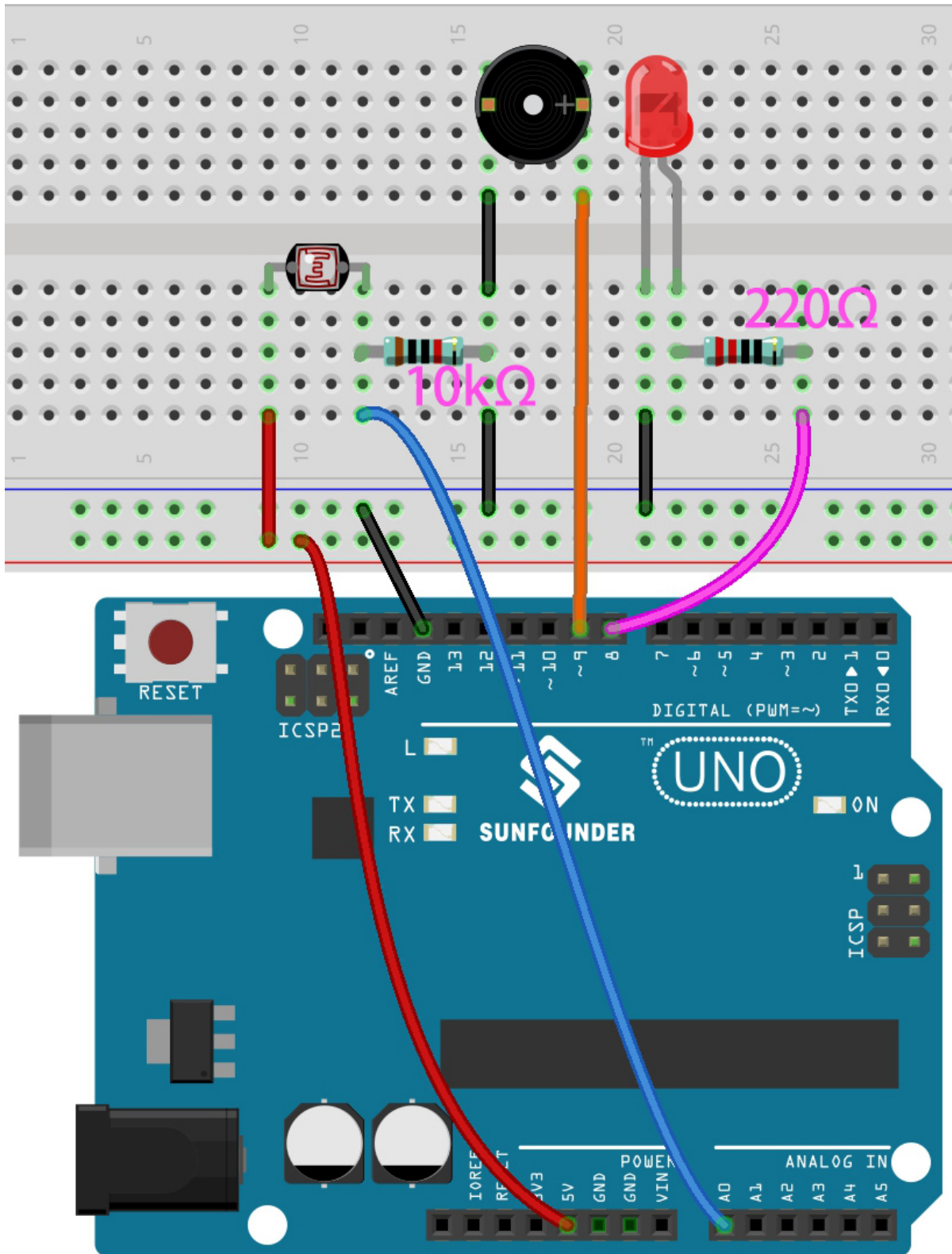
Sie können diese auch separat über die folgenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Summer</i>	
<i>LED</i>	
<i>Fotowiderstand</i>	

Schaltbild



Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei 5.14.calibration.ino im Pfad 3in1-kit\basic_project\5.14.calibration.
 - Oder kopieren Sie diesen Code in die **Arduino IDE**.
 - Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.
-

Nach erfolgreichem Hochladen des Codes 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).

In dieser Zeit sollten wir unsere Hände über dem Fotowiderstand auf und ab bewegen, und der Bewegungsbereich der Hand wird auf den Spielbereich dieses Instruments kalibriert.

Nach 5 Sekunden erlischt die LED und wir können mit unseren Händen über dem Fotowiderstand schwingen, um zu spielen.

Wie funktioniert das?

1. Legen Sie die Anfangswerte und Pins aller Komponenten fest.

```
const int buzzerPin = 9;
const int ledPin = 8;
const int photocellPin = A0; // Fotowiderstand ist an A2 angeschlossen

int lightLow = 1023;
int lightHigh = 0;

int sensorValue = 0; // vom Sensor gelesener Wert
int pitch = 0; // Sensorwert in LED-'Balken' umgewandelt

unsigned long previousMillis = 0;
const long interval = 5000;
```

2. Richten Sie einen Kalibrierungsprozess in setup() ein.

```
void setup()
{
  pinMode(buzzerPin, OUTPUT); // machen Sie alle LED-Pins zu Outputs
  pinMode(ledPin, OUTPUT); // Den LED-Pin-Ausgang erstellen

  /* kalibriere die Maximal- und Minimalwerte des Fotowiderstands */
  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 Arbeitsablauf ist wie folgt:

- Verwendung von `millis()` für die Zeitmessung mit einem Intervall von 5000 ms.

```
previousMillis = millis();
...
while (millis() - previousMillis <= interval) {
...
}
```

- Während dieser fünf Sekunden, bewegen Sie Ihre Hand um den Fotowiderstand. Die maximalen und minimalen Werte des erkannten Lichts werden erfasst und den Variablen `lightHigh` und `lightLow` entsprechend zugewiesen.

```
sensorValue = analogRead(photocellPin);
if (sensorValue > lightHigh) {
    lightHigh = sensorValue;
}
if (sensorValue < lightLow) {
    lightLow = sensorValue;
}
```

3. Jetzt können Sie dieses Theremin spielen. Lesen Sie den Wert des Fotowiderstands in `sensorValue` und übertragen Sie ihn von einem kleinen Bereich in einen großen Bereich, um ihn als Frequenz des Buzzers zu verwenden.

```
void loop()
{
    /* spielen */
    sensorValue = analogRead(photocellPin); // Wert von A0 lesen
    pitch = map(sensorValue, lightLow, lightHigh, 50, 6000); // auf die Buzzer-
    ↪ Frequenz abbilden
    if (pitch > 50) {
        tone(buzzerPin, pitch, 20);
    }
    delay(10);
}
```

5.5.15 5.15 EEPROM

EEPROM ist ein Speicher, dessen Daten nicht gelöscht werden, wenn das Hauptsteuerungsboard ausgeschaltet wird. Sie können es verwenden, um einige Daten zu speichern und diese beim nächsten Einschalten wieder abzurufen.

Als Beispiel könnten Sie einen Sportzähler erstellen, der verfolgt, wie oft Sie täglich Seil springen.

Sie können auch in einem Programm Daten darauf schreiben und in einem anderen Programm daraus lesen. Wenn Sie beispielsweise an einem Autoprojekt arbeiten und die Geschwindigkeiten der beiden Motoren nicht übereinstimmen, können Sie ein Kalibrierungsprogramm schreiben, um den Kompensationswert der Motorgeschwindigkeit zu speichern.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

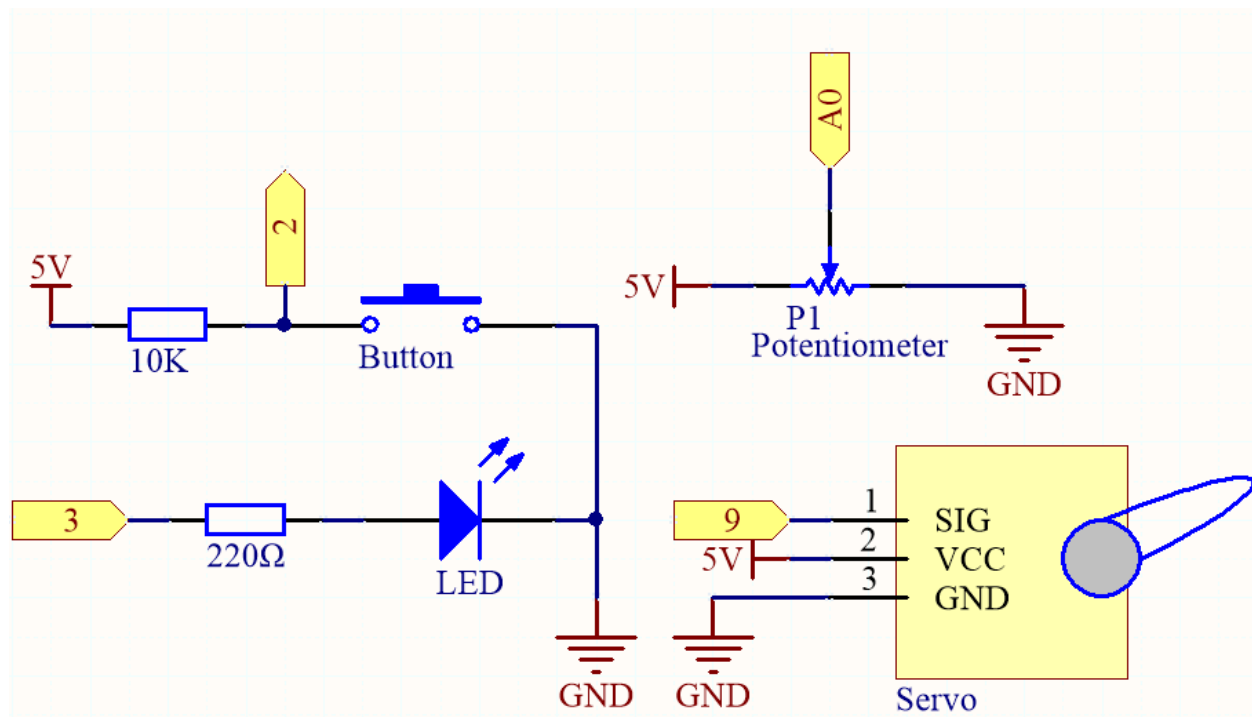
Es ist sicherlich praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

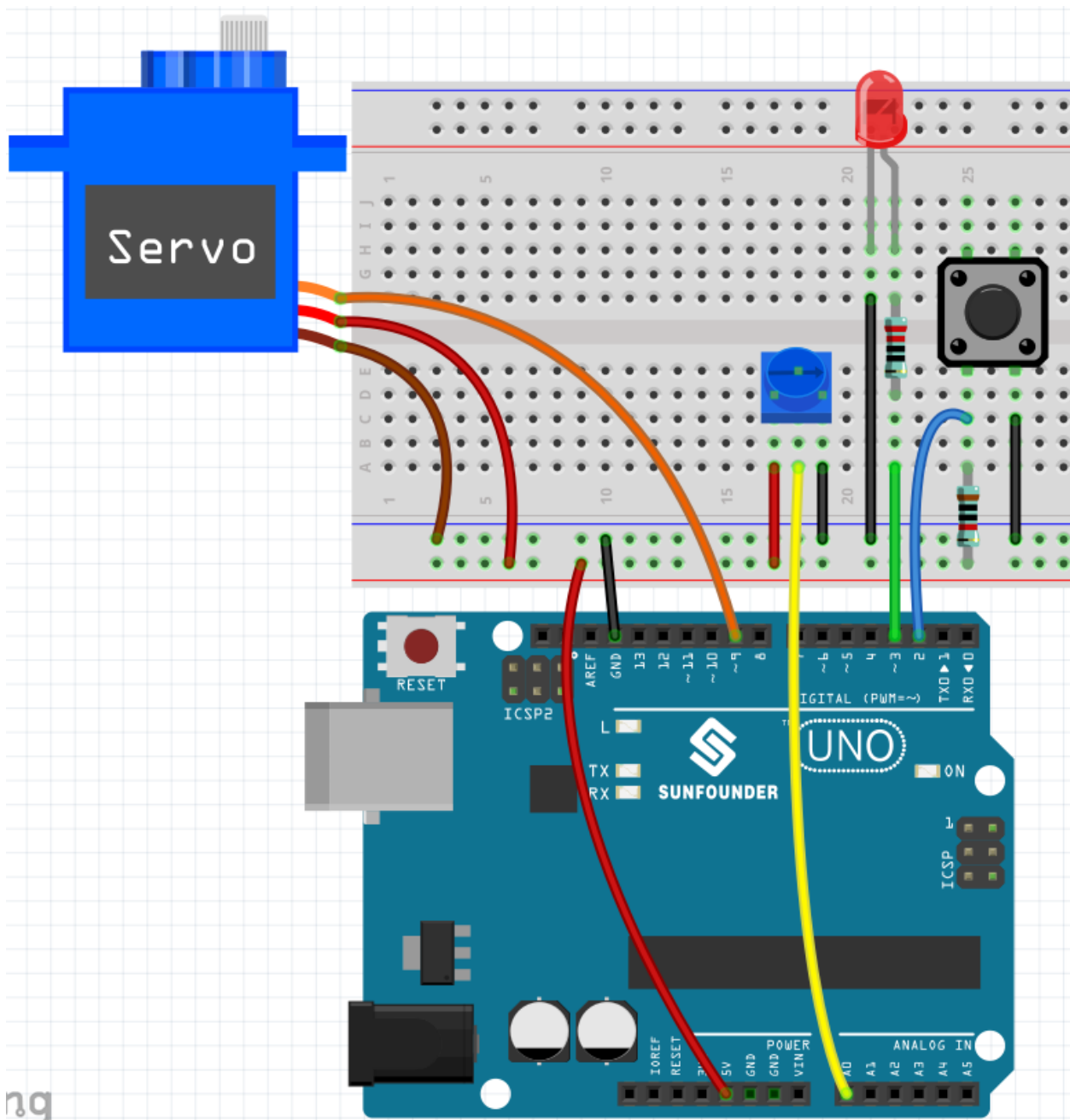
Sie können diese auch separat über die folgenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</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/basic_project/5.15.eeprom.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Um diese Schaltung zu verwenden, drücken Sie einfach den Knopf, um die Aufnahme zu starten, und geben die gewünschten Informationen über ein Potentiometer ein. Nun wird das Board Ihre Aktionen endlos wiederholen (und bei

jeder Iteration blinkt eine LED), bis Sie den Knopf erneut drücken, um neue Aktionen aufzunehmen. Sie können auch die Aufnahmezeit variieren, indem Sie die Werte von Auflösung und Aufnahmezeit ändern.

Wie funktioniert das?

1. Importieren Sie die Bibliothek `EEPROM.h` und initialisieren Sie den EEPROM-Speicher.

```
...
#include <EEPROM.h> //wird verwendet, um aufgezeichnete Werte zu speichern
...
float resolution = 1000; //MUSS kleiner sein als EEPROM.length()
float recordTime = 5; //Verzögerungszeit
bool recording = false;
...
```

Beachten Sie, dass `/MUST kleiner sein als EEPROM.length()`, in `setup()` wird der Speicher Ihres Boards EEPROM angezeigt, welcher für das SunFounder R3 Board 1024 sein sollte. Wenn Sie ein anderes Board verwenden, können Sie den Wert der Variable `resolution` anpassen.

2. Zeigen Sie den EEPROM-Speicher Ihres Boards an.

```
void setup() {
    ...
    Serial.begin(9600);
    //Serial.println(EEPROM.length());
}
```

Um die Größe des EEPROM-Speichers Ihres Boards zu ermitteln, kommentieren Sie die Zeile `Serial.println(EEPROM.read(i))` aus. Dies wird die Größe des EEPROM im seriellen Monitor anzeigen, und Sie können den Wert der Variable `resolution` entsprechend ändern.

3. Sobald ein Tastendruck erkannt wird, beginnt die Aufzeichnung und die erforderlichen Informationen werden über ein Potentiometer eingegeben. Jetzt wiederholt das Board Ihre Aktion endlos (und eine LED blinkt bei jeder Wiederholung), bis Sie die Taste erneut drücken und eine neue Aktion aufzeichnen.

```
void loop() {
    if (recording == true) { //aufzeichnen
        for (int i = 1; i <= resolution; i++) {
            digitalWrite(ledPin, HIGH); //Lichtstatus-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); //Status-LED ausschalten
        delay(1000); //Zeit für den Menschen geben
        recording = false;
    }
    else {
        for (int i = 1; i <= resolution; i++) { //Wiedergabe
            if (digitalRead(buttonPin) == 0) { //Stoppen Sie die Wiedergabe
                //und zeichnen Sie neue Werte auf
                recording = true;
                break;
            }
        }
    }
}
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

    }
    int readval = EEPROM.read(i);
    myServo.write(readval);
    //Serial.println(readval);
    delay(recordTime);
  }
  digitalWrite(ledPin, HIGH); //eine neue Wiederholung anzeigen
  delay(100);
  digitalWrite(ledPin, LOW);
}
}

```

- Setzen Sie die Variable `recording` auf `true`, wenn die Taste gedrückt wird.
- Wenn die Variable `recording` `true` ist, beginnen Sie mit der Aufzeichnung der Aktion im Speicherbereich.
- Lesen Sie den Wert des Potentiometers und mappen Sie ihn auf 0-180, um ihn im EEPROM zu speichern und die Drehung des Servos zu steuern.
- Die LED leuchtet zu Beginn der Aufzeichnung und erlischt am Ende.
- Wiederholen Sie die aufgezeichnete Aktion mit einem kurzen Aufleuchten der LED, um Sie an eine neue Wiederholung zu erinnern.

4. Über die EEPROM-Bibliothek.

Hier sind einige ihrer Funktionen.

- `write(address, value)`: Schreibt ein Byte in das EEPROM.
 - `address`: die Adresse, an die geschrieben werden soll, beginnend bei 0 (int)
 - `value`: der zu schreibende Wert, von 0 bis 255 (byte)
 - Ein EEPROM-Schreibvorgang dauert 3,3 ms. Der EEPROM-Speicher 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 in das EEPROM. Der Wert wird nur geschrieben, wenn er sich von dem bereits an derselben Adresse gespeicherten unterscheidet.
 - Ein EEPROM-Schreibvorgang dauert 3,3 ms. Der EEPROM-Speicher hat eine angegebene Lebensdauer von 100.000 Schreib-/Löschzyklen, daher 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 in das EEPROM.
 - `address`: die Adresse, von der gelesen werden soll, beginnend bei 0 (int).
 - `data`: die zu lesenden Daten, können ein primitiver Typ (z. B. `float`) oder eine benutzerdefinierte Struktur sein.
 - Diese Funktion verwendet `EEPROM.update()`, um den Schreibvorgang auszuführen, 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`: die Adresse, von der gelesen werden soll, beginnend bei 0 (int).

- `data`: die zu lesenden Daten, können ein primitiver Typ (z. B. `float`) oder eine benutzerdefinierte Struktur sein.

5.6 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.

5.6.1 6.1 Lichtgesteuerte fließende LEDs

Ein Fotowiderstand oder Fotodiode ist ein lichtgesteuerter variabler Widerstand. Der Widerstand eines Fotowiderstandes nimmt mit zunehmender einfallender Lichtintensität ab; anders ausgedrückt, zeigt er Photoleitfähigkeit. Ein Fotowiderstand kann in lichtempfindlichen Detektorschaltungen und licht- und dunkelaktivierten Schaltkreisen eingesetzt werden.

Der Widerstand eines Fotowiderstandes ändert sich mit der einfallenden Lichtintensität. Wenn die Lichtintensität steigt, nimmt der Widerstand ab; sinkt sie, steigt der Widerstand. In diesem Experiment verwenden wir acht LEDs, um die Lichtintensität darzustellen. Je höher die Lichtintensität ist, desto mehr LEDs leuchten auf. Bei ausreichend hoher Lichtintensität leuchten alle LEDs. Bei Dunkelheit erlöschen alle LEDs.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

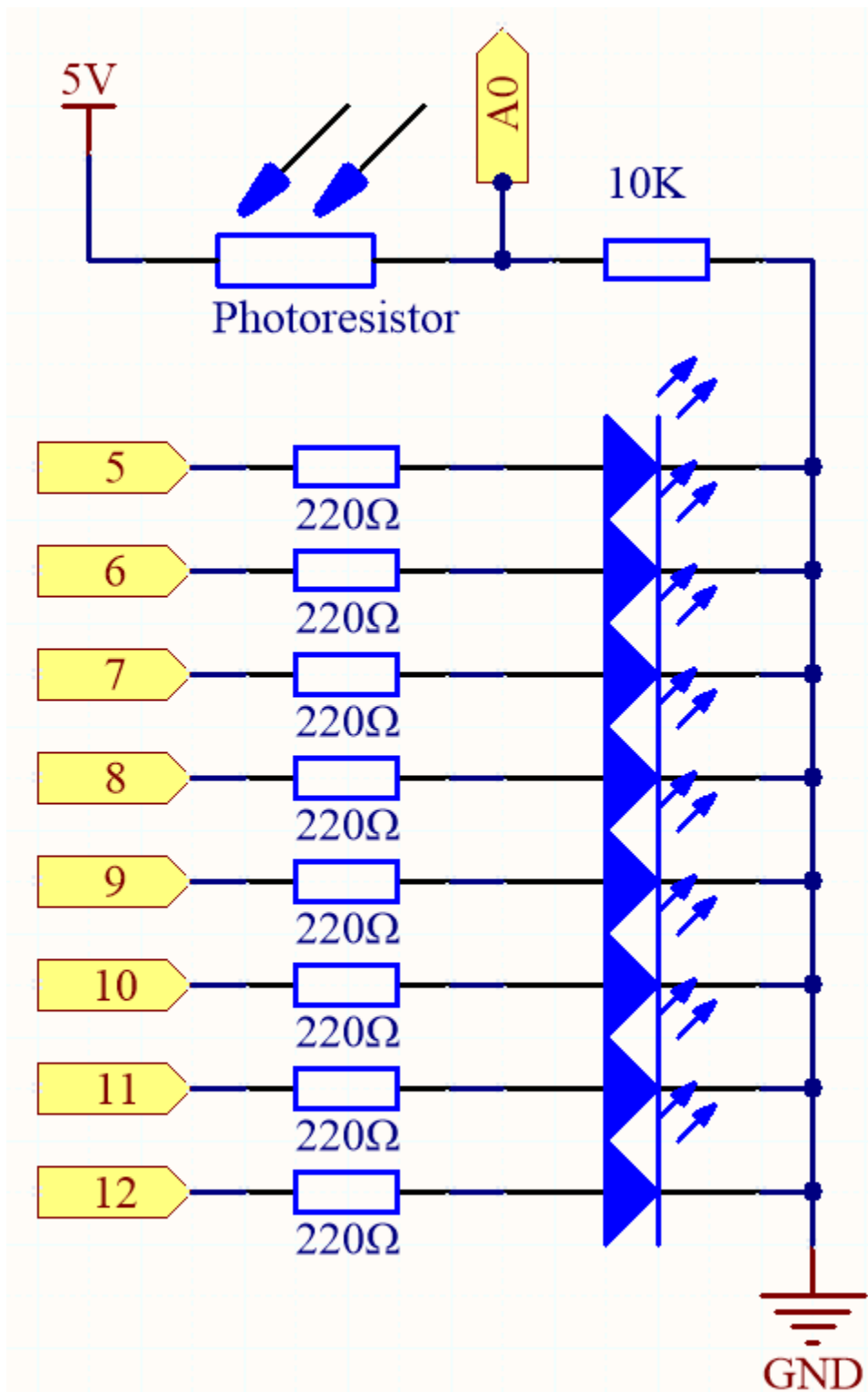
Es ist sicherlich praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

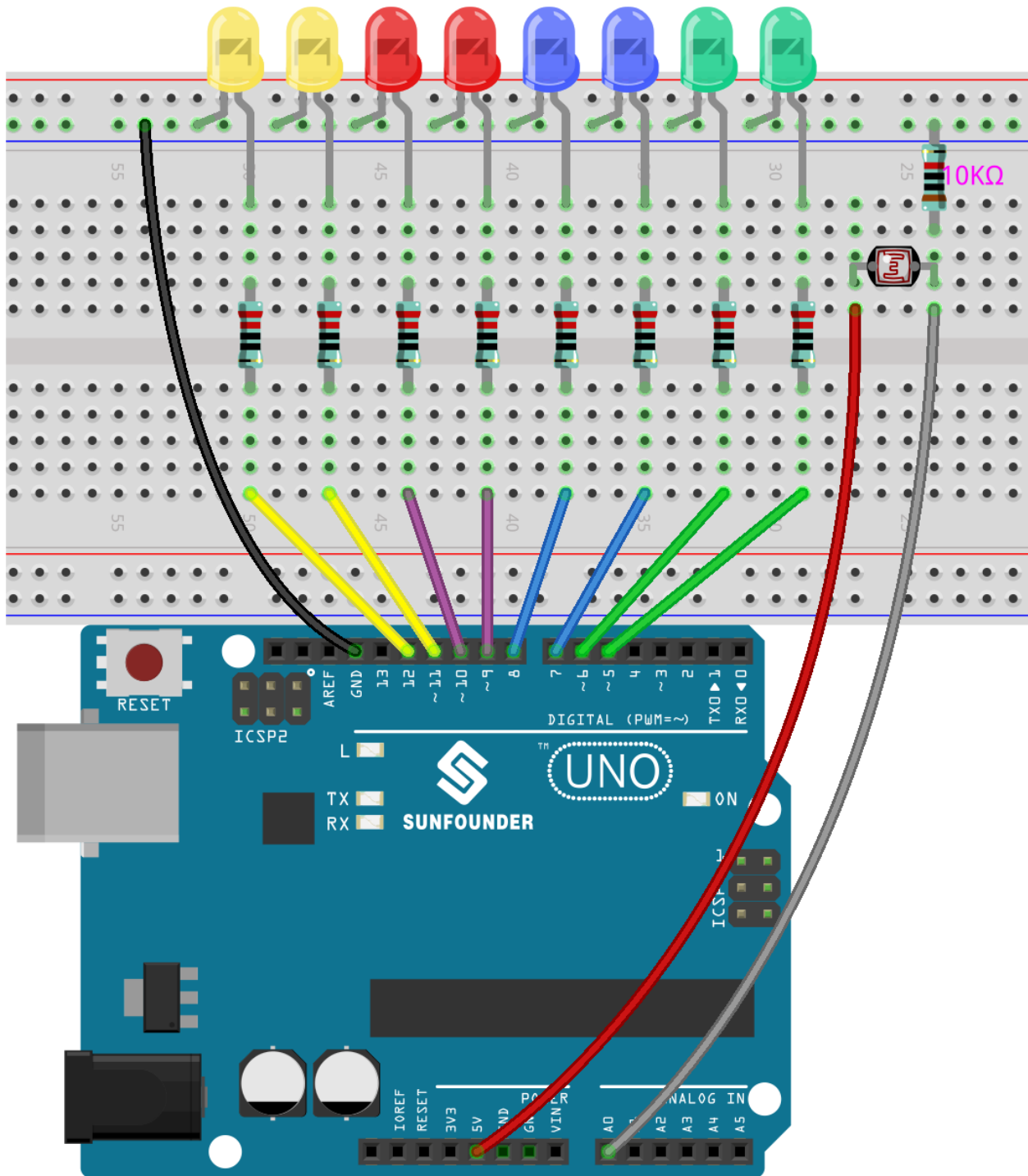
Sie können diese auch einzeln über die folgenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Fotowiderstand</i>	

Schaltplan



Verkabelung



Code

Bemerkung:

- Öffnen Sie die Datei `6.1.light_control_led.ino` im Pfad `3in1-kit\basic_project\6.1.light_control_led`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.

- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Jetzt leuchten Sie mit einer Lichtquelle auf den Fotowiderstand, und Sie werden sehen, dass mehrere LEDs aufleuchten. Bei noch mehr Licht leuchten noch mehr LEDs. In einer dunklen Umgebung erlöschen alle LEDs.

Wie funktioniert das?

```
void loop()
{
    sensorValue = analogRead(photocellPin); //liest den Wert von A0 aus
    ledLevel = map(sensorValue, 300, 1023, 0, NbrLEDs); // mappt auf die Anzahl der LEDs
    for (int led = 0; led < NbrLEDs; led++)//
    {
        if (led < ledLevel ) //Wenn led kleiner als ledLevel ist, wird der folgende Code
        ↳ausgeführt.
        {
            digitalWrite(ledPins[led], HIGH); // schaltet Pins unterhalb des Levels ein
        }
        else
        {
            digitalWrite(ledPins[led],LOW); // schaltet Pins oberhalb aus
        }
    }
}
```

Mit der Funktion map() können Sie den Wert des Fotowiderstands auf die 8 LEDs abbilden, zum Beispiel, wenn sensorValue 560 ist, dann ist ledLevel 4, also sollten in diesem Moment ledPins[0] bis ledPins[4] leuchten und ledPins[5] bis ledPins[7] aus sein.

5.6.2 6.2 Elektronischer Würfel

Hier verwenden wir einen Knopf, eine 7-Segment-Anzeige und den 74hc595, um einen elektronischen Würfel zu erstellen. Jedes Mal, wenn der Knopf gedrückt wird, wird eine zufällige Zahl zwischen 1 und 6 generiert und auf der 7-Segment-Anzeige dargestellt.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

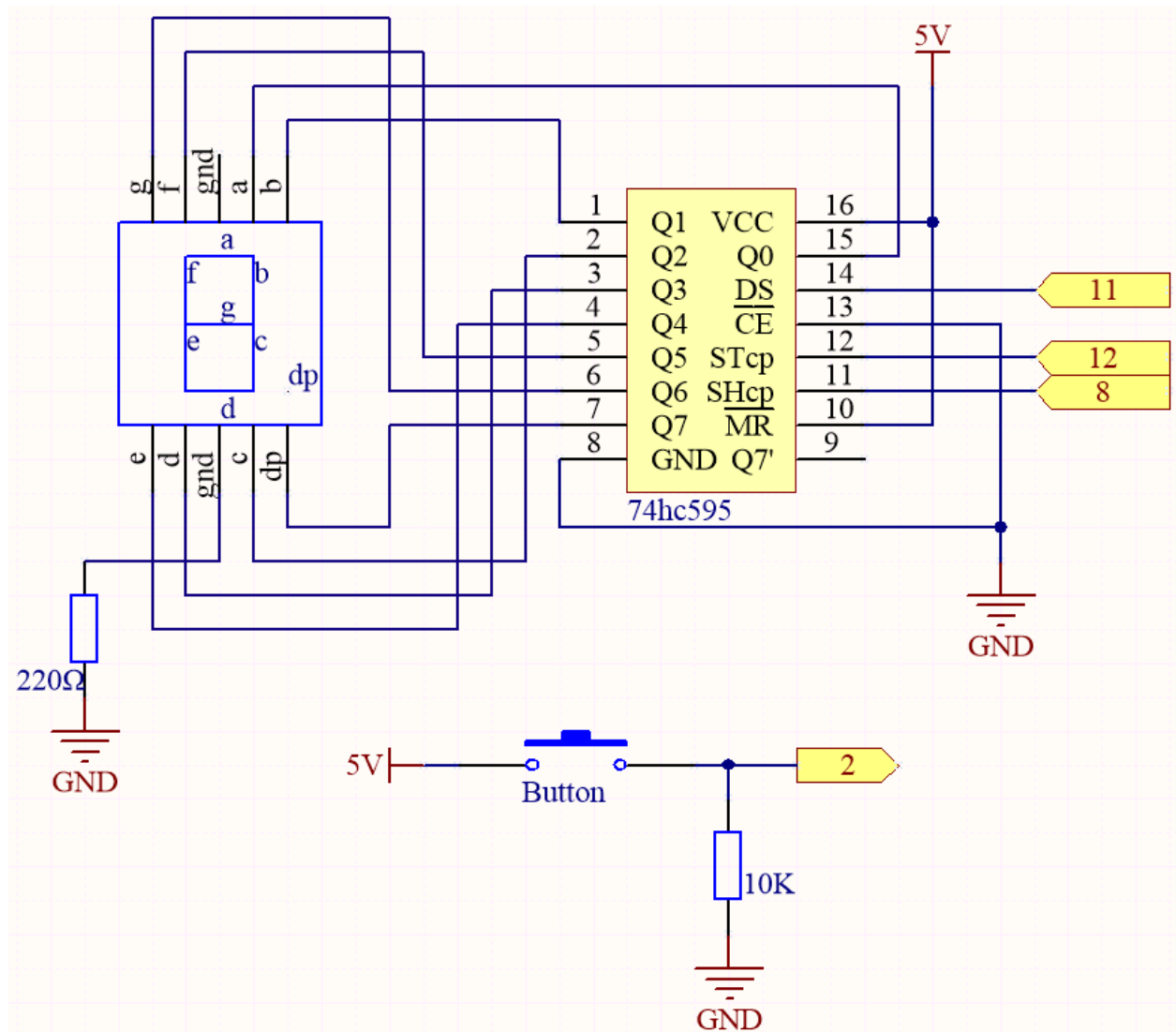
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

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

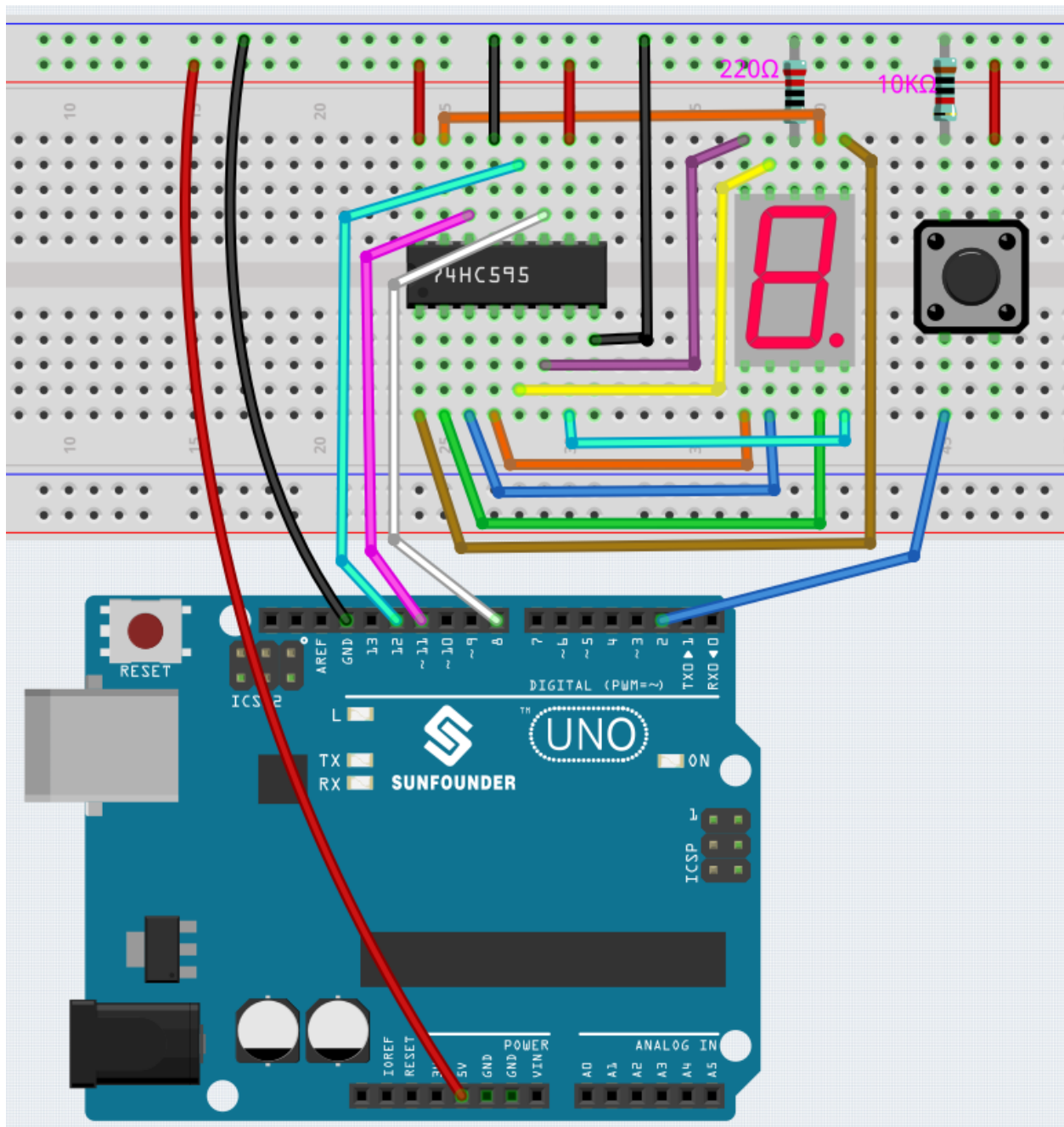
Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Knopf</i>	
<i>74HC595</i>	
<i>7-Segment-Anzeige</i>	

Schaltplan



Verkabelung



Code

Bemerkung:

- Öffnen Sie die Datei 6.2.electronic_dice.ino im Pfad 3in1-kit\basic_project\6.2.electronic_dice.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Oder laden Sie den Code über den [Arduino Web Editor](#) hoch.

Nach erfolgreichem Hochladen des Codes zeigt die 7-Segment-Anzeige schnell die Zahlen 0-7 an. Wenn Sie den Knopf drücken, zeigt sie eine zufällige Zahl an und stoppt das Scrollen. Das Scrollen beginnt erneut, wenn Sie den Knopf wieder drücken.

Wie funktioniert das?

Dieses Projekt basiert auf [5.10 ShiftOut\(Segmentanzeige\)](#) mit einem Knopf, um das Scrollen auf der 7-Segment-Anzeige zu starten/pausieren.

1. Initialisieren Sie jeden Pin und lesen Sie den Wert des Knopfes.

```
void setup ()
{
    ...
    attachInterrupt(digitalPinToInterrupt(buttonPin), rollDice, FALLING);
}
```

- Der Interrupt wird hier verwendet, um den Zustand des Knopfes zu lesen. Der Standardwert von buttonPin ist niedrig und ändert sich von niedrig zu hoch, wenn der Knopf gedrückt wird.
 - rollDice stellt die Funktion dar, die aufgerufen wird, wenn der Interrupt ausgelöst wird. Sie dient zum Umschalten des Werts der Variablen state.
 - FALLING bedeutet, dass der Interrupt ausgelöst wird, wenn der buttonPin von niedrig zu hoch geht.
2. Wenn die Variable state 0 ist, wird die Funktion showNumber() aufgerufen, um die 7-Segment-Anzeige eine zufällige Zahl zwischen 1 und 7 anzuzeigen.

```
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 von 1 beim letzten Mal zu 0 dieses Mal.

4. Über die Funktion showNumber().

```
void showNumber(int num) {
    digitalWrite(STcp, LOW); //ST_CP erden und solange niedrig halten, wie
    ↪ Daten übertragen werden
    shiftOut(DS, SHcp, MSBFIRST, dataArray[num]);
    //setzen Sie den Latch-Pin zurück, um dem Chip zu signalisieren, dass er
    //nicht mehr auf Informationen hören muss
    digitalWrite(STcp, HIGH); //ST_CPST_CP hochziehen, um die Daten zu
    ↪ speichern
}
```

Dies ist der Code innerhalb von `loop()` im Projekt *5.10 ShiftOut(Segmentanzeige)* in der Funktion `showNumber()`.

5.6.3 6.3 Hochtemperaturalarm

Als Nächstes werden wir mithilfe von Thermistor, Druckknopf, Potentiometer und LCD ein Hochtemperaturalarmgerät bauen. Das LCD1602 zeigt die vom Thermistor erfasste Temperatur sowie den Hochtemperaturschwellenwert an, der mit einem Potentiometer eingestellt werden kann. Der Schwellenwert wird gleichzeitig auf dem EEPROM gespeichert. Überschreitet die aktuelle Temperatur den Schwellenwert, ertönt der Summer.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

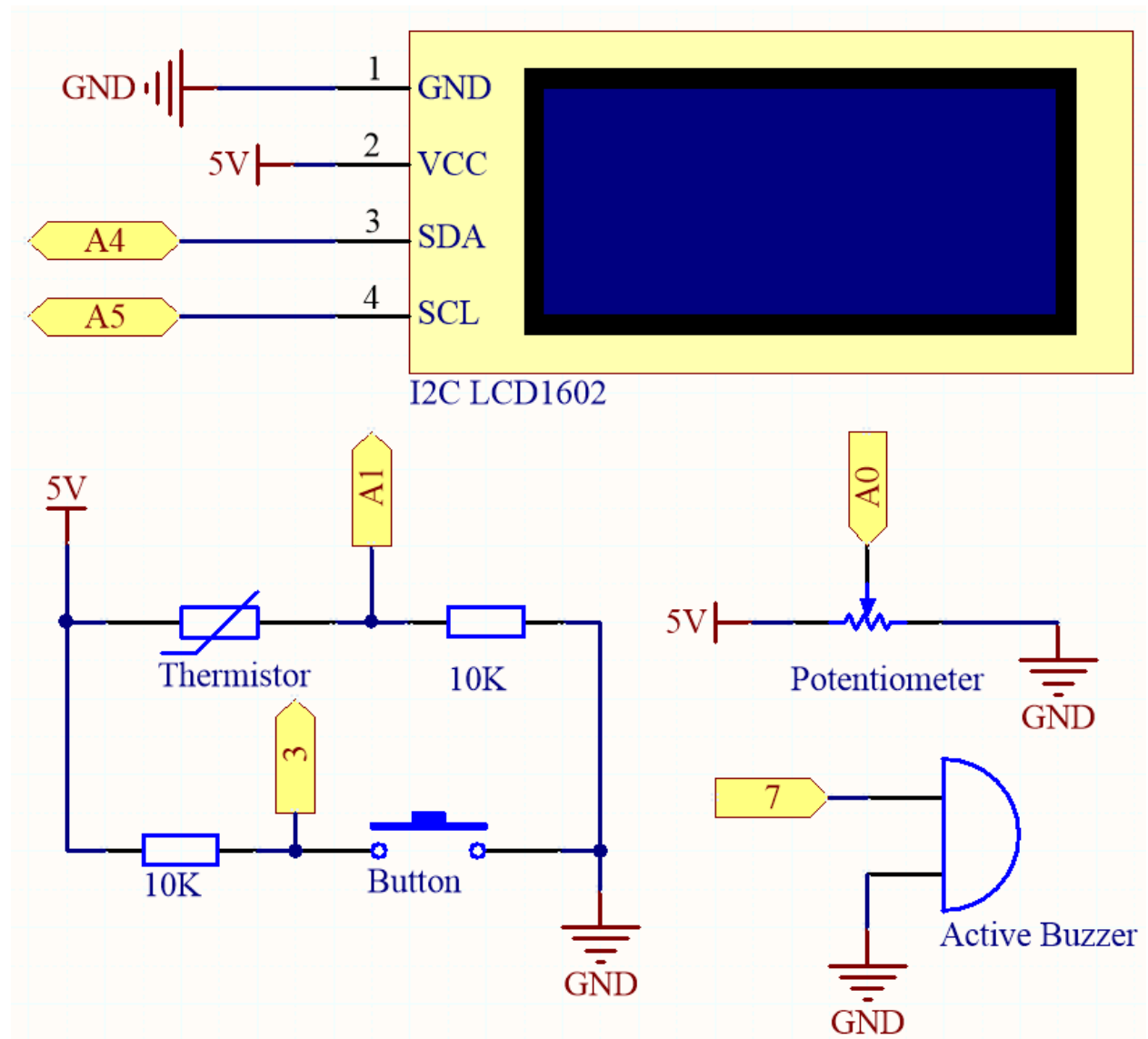
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

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

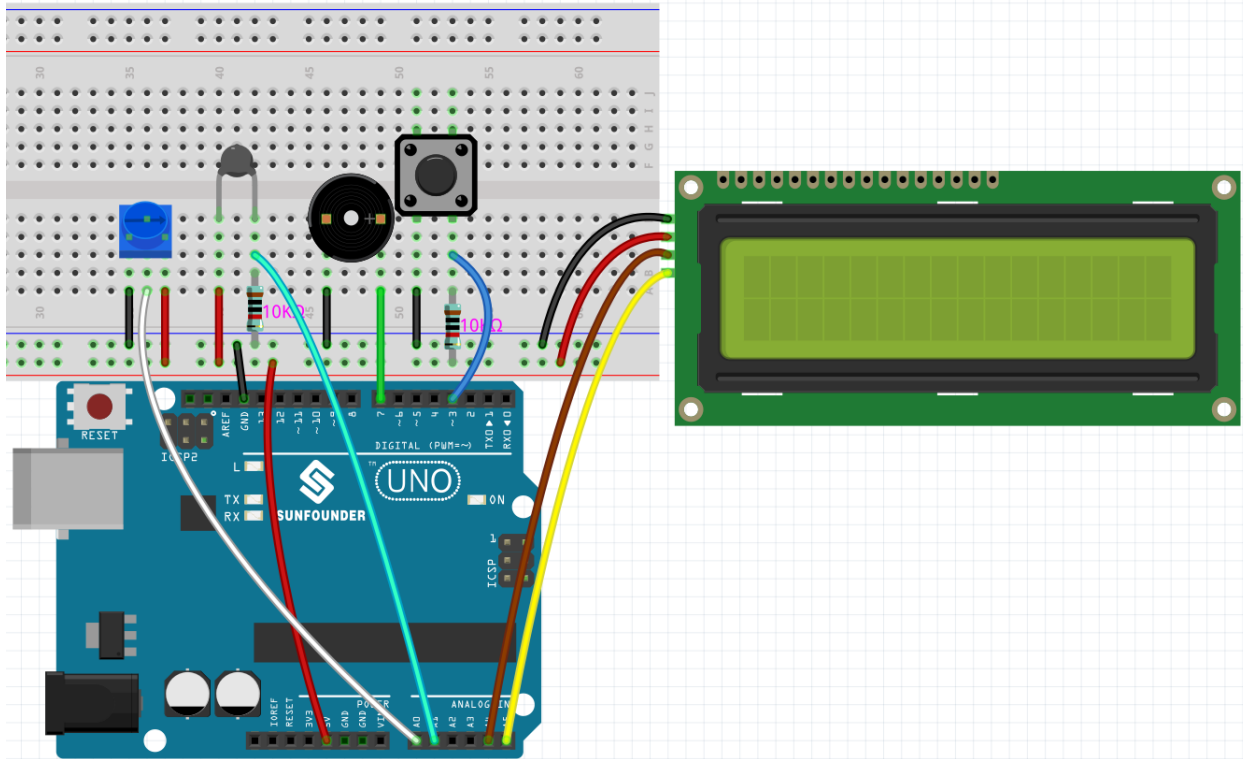
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</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



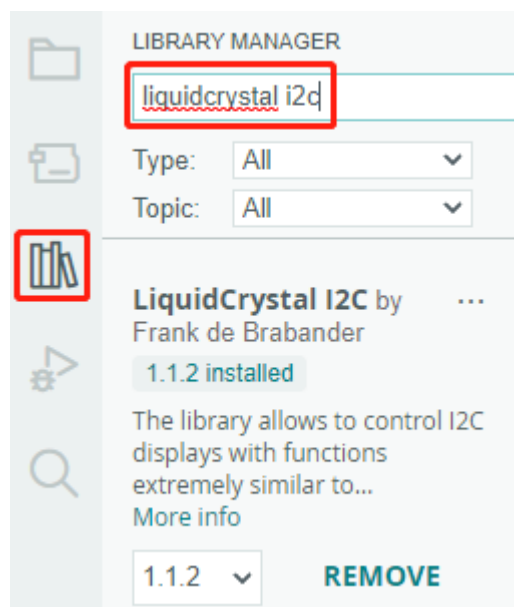
Verdrahtung



Code

Bemerkung:

- Sie können die Datei 6.3.high_tem_alarm.ino direkt im Pfad 3in1-kit\basic_project\6.3.high_tem_alarm öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE 1/2.
- Hier wird die Bibliothek LiquidCrystal I2C verwendet. Sie können sie aus dem **Library Manager** installieren.



Nachdem der Code erfolgreich hochgeladen wurde, zeigt das LCD1602 die vom Thermistor erfasste Temperatur sowie den Hochtemperaturschwellenwert an, der mit einem Potentiometer eingestellt werden kann. Der Schwellenwert wird gleichzeitig auf dem EEPROM gespeichert. Überschreitet die aktuelle Temperatur den Schwellenwert, ertönt der Summer.

Bemerkung: Wenn der Code und die Verkabelung korrekt sind, das LCD jedoch immer noch keinen Inhalt anzeigt, können Sie das Potentiometer auf der Rückseite drehen.

Wie funktioniert das?

1. Initialisieren Sie den Knopf, den Summer und das I2C LCD1602 und lesen Sie die EEPROM-Werte aus. Hier wird auch ein Interrupt verwendet, um den Status des Knopfes 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 Zustand des Knopfes zu lesen. Wenn der Knopf gedrückt wird, wechselt buttonPin von niedrig auf hoch.
 - Die Funktion buttonState wird aufgerufen, wenn der Interrupt ausgelöst wird, und sie wechselt den Wert der Variable state.
 - FALLING bedeutet, dass der Interrupt auftritt, wenn buttonPin von niedrig auf hoch geht.
2. Um den Hochtemperaturschwellenwert festzulegen, wird die Funktion upperTemSetting() aufgerufen, wenn state 1 ist (state wechselt zwischen 0 und 1 bei Knopfdruck) im Hauptprogramm, ansonsten 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;
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

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. Betreten Sie diese Funktion, zeigt das LCD1602 den aktuellen Schwellenwert an, der mit dem Potentiometer geändert werden kann. Dieser Schwellenwert wird im EEPROM gespeichert und beim erneuten Drücken des Knopfes verlassen.

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: ");
    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
    {

```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

    digitalWrite(buzzerPin, LOW);
}
}

```

- Mit dieser Funktion können Sie die Temperatur anzeigen und einen Alarm einstellen.
- Der Wert des Thermistors wird gelesen und dann mittels der Formel in die Celsius-Temperatur umgerechnet und auf dem LCD1602 angezeigt.
- Der eingestellte Schwellenwert wird ebenfalls auf dem LCD angezeigt.
- Wenn die aktuelle Temperatur höher als der Schwellenwert ist, gibt der Summer einen Alarm aus.

5.6.4 6.4 Einparkhilfe

Mit der Entwicklung von Wissenschaft und Technik wurden viele High-Tech-Produkte in Autos eingebaut, unter denen das Einparkhilfesystem eines der bekanntesten ist. In diesem Projekt nutzen wir ein Ultraschallmodul, LCD, LED und Buzzer, um ein einfaches Ultraschall-Einparkhilfesystem zu erstellen.

Benötigte Bauteile

Für dieses Projekt benötigen wir die folgenden Komponenten.

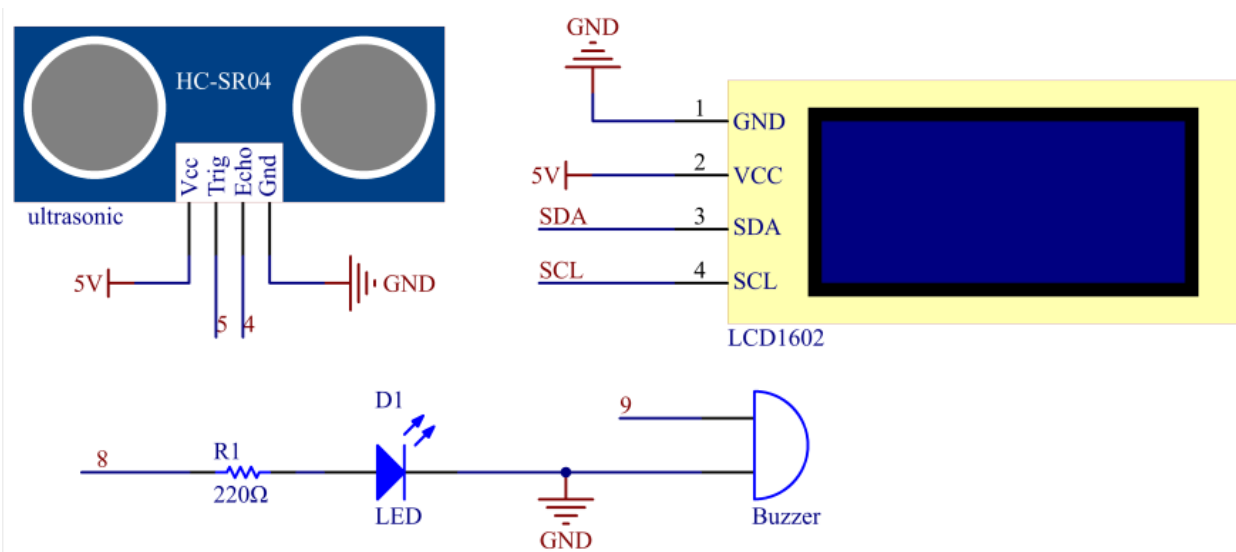
Es ist sicherlich praktisch, ein komplettes Set zu kaufen, hier ist der Link:

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

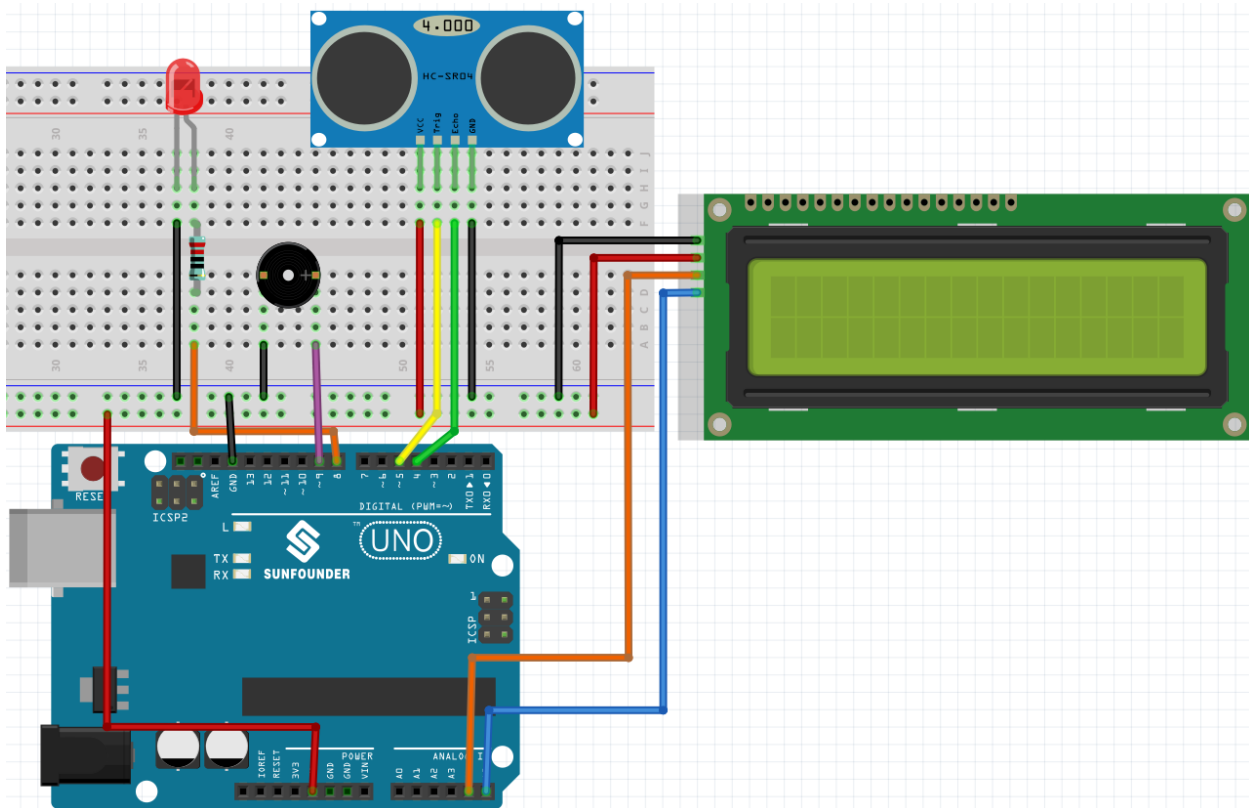
Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Summer</i>	
<i>I2C LCD1602</i>	
<i>Ultraschall-Modul</i>	

Schaltplan



Verdrahtung

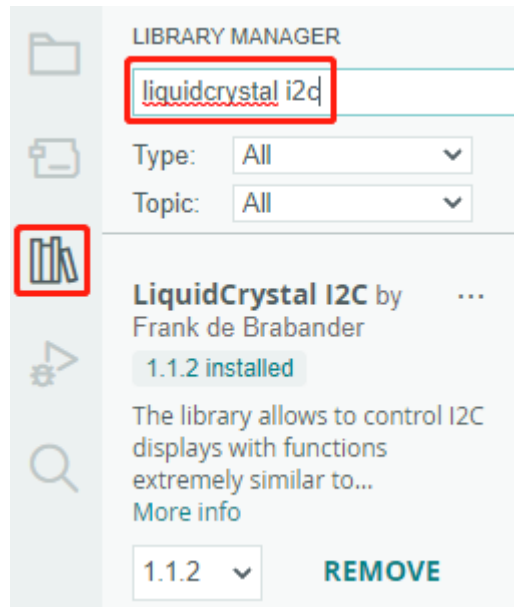


Code

Bemerkung:

- Sie können die Datei 6.4_reversingAid.ino direkt im Pfad 3in1-kit/basic_project/6.4_reversingAid öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE 1/2.

- Hier wird die Bibliothek LiquidCrystal I2C verwendet. Sie können sie aus dem **Library Manager** installieren.



Nachdem der Code erfolgreich hochgeladen wurde, wird die aktuell erkannte Entfernung auf dem LCD angezeigt. Der Buzzer wird dann die Klangfrequenz je nach Entfernung ändern.

Bemerkung: Wenn der Code und die Verdrahtung korrekt sind, aber das LCD 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 kann und Feedback über ein LCD-Display und einen Summer liefert.

Die Funktion `loop()` enthält die Hauptlogik des Programms und läuft kontinuierlich. Schauen wir uns die Funktion `loop()` genauer an.

1. Schleife zum Auslesen der Entfernung und Aktualisieren von Parametern

In der `loop` liest der Code zuerst die vom Ultraschallmodul gemessene Entfernung aus und aktualisiert den Intervall-Parameter basierend auf dieser Entfernung.

```
// Entfernung aktualisieren
distance = readDistance();

// Intervalle basierend auf der Entfernung aktualisieren
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 für einen Signalton ist

Der Code berechnet die Differenz zwischen der aktuellen Zeit und der letzten Signalton-Zeit. Wenn die Differenz größer oder gleich der Intervallzeit ist, löst er den Summer aus und aktualisiert die letzte Signalton-Zeit.

```
unsigned long currentMillis = millis();
if (currentMillis - previousMillis >= intervals) {
  Serial.println("Beeping!");
  beep();
  previousMillis = currentMillis;
}
```

3. LCD-Anzeige aktualisieren

Der Code löscht das LCD-Display und zeigt dann „Dis:“ und die aktuelle Entfernung in Zentimetern in der ersten Zeile an.

```
lcd.clear();
lcd.setCursor(0, 0);
lcd.print("Dis: ");
lcd.print(distance);
lcd.print(" cm");

delay(100);
```

5.6.5 6.5 Reaktionsspiel

Unser Körper hat viele Reaktionszeiten, wie zum Beispiel auditive RT, visuelle RT, taktile RT usw.

Reaktionszeiten haben viele Auswirkungen auf unser tägliches Leben. Ein langsamer als normaler Reaktionszeit beim Fahren kann beispielsweise zu schwerwiegenden Folgen führen.

In diesem Projekt verwenden wir 3 Tasten und 2 LEDs, um unsere visuelle Reaktionszeit zu messen.

Der serielle Monitor des Arduino zeigt die Nachricht „Warten...“ an. Nach dem Drücken der „Bereit“-Taste muss eine der beiden LEDs nach einem zufälligen Zeitintervall zufällig aufleuchten. Es ist wichtig, dass der Proband den entsprechenden Knopf so schnell wie möglich drückt. Der Arduino erfasst die Zeitdifferenz zwischen dem Aufleuchten der LED und dem Drücken des entsprechenden Knopfes durch die Person und gibt die gemessene Reaktionszeit auf dem Arduino-Seriellen-Monitor aus.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

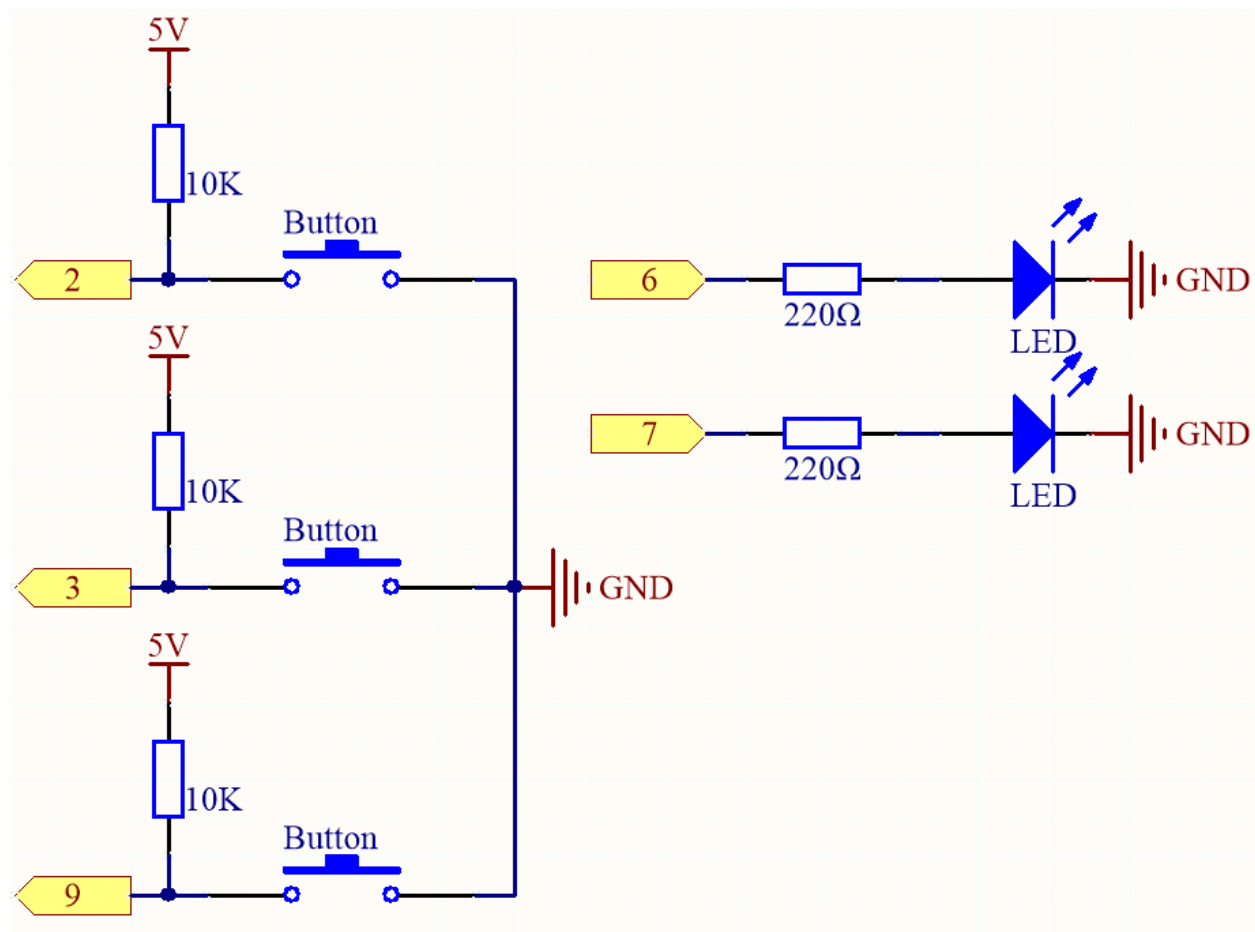
Es ist definitiv praktisch, ein gesamtes Set zu kaufen. Hier ist der Link:

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>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	
<i>Knopf</i>	

Schaltplan



Verdrahtung

Code

Bemerkung:

- Sie können die Datei 6.5_reaction_time.ino direkt im Pfad 3in1-kit\basic_project\6.5_reversingAid öffnen.
 - Oder kopieren Sie diesen Code in die Arduino IDE 1/2.
 - Stellen Sie bitte sicher, dass Sie die LiquidCrystal_I2C Bibliothek hinzugefügt haben. Detaillierte Anleitungen finden Sie unter [5.11 Externe Bibliotheken installieren](#).
-

Wie funktioniert das?

1. Initialisierung der 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, startet das Spiel erneut. Nach einer zufälligen Zeit zwischen 2 und 5ms leuchtet eine der LEDs auf.

```
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 zwischen 2-5s zu generieren.
 - Diese Zeit wird dann verwendet, um das Leuchten der LEDs zu steuern.
 - Auch das Leuchten der 2 LEDs wird zufällig durch randomTime % 2 mit 0 und 1 generiert. Wenn flag 0 ist, dann leuchtet LED1; wenn 1, dann leuchtet LED2.
3. Über die pressed1() Funktion


```

void pressed1() {
    if (flag == -1) {
        return;
    }
    if (flag == 0) {
        int currentTime = millis();
        Serial.print("Correct! Your reaction time is: ");
        Serial.print(currentTime - timer);
        Serial.println(" ms");
    } else if (flag == 1) {
        Serial.println("Wrong Click!");
    }
    flag = -1;
}

```

Diese Funktion wird ausgelöst, wenn Taste 1 gedrückt wird. Wird Taste 1 gedrückt und flag ist zu diesem Zeitpunkt 0, wird die Reaktionszeit angezeigt. Andernfalls wird ein Druckfehler angezeigt.

4. Über die pressed2() Funktion

```

void pressed2() {
    if (flag == -1) {
        return;
    }
    if (flag == 1) {
        int currentTime = millis();
        Serial.print("Correct! Your reaction time is: ");
        Serial.print(currentTime - timer);
        Serial.println(" ms");
    } else if (flag == 0) {
        Serial.println("Wrong Click!");
    }
    flag = -1;
}

```

Diese Funktion wird ausgelöst, wenn Taste 2 gedrückt wird. Wird Taste 2 gedrückt und flag ist zu diesem Zeitpunkt 1, wird die Reaktionszeit angezeigt. Andernfalls wird ein Druckfehler angezeigt.

5.6.6 6.6 Zahl erraten

Das Erraten von Zahlen ist ein unterhaltsames Partyspiel, bei dem Sie und Ihre Freunde abwechselnd eine Zahl (0~99) eingeben. Der Bereich wird mit der Eingabe der Zahl kleiner, bis ein Spieler das Rätsel richtig beantwortet. 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 Nummernbereich zu 50~99; wenn Spieler 2 70 eingibt, kann der Nummernbereich 50~70 sein; wenn Spieler 3 51 eingibt, ist er oder sie der Pechvogel. Hier verwenden wir die IR-Fernbedienung zur Eingabe von Zahlen und das LCD zur Ausgabe von Ergebnissen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

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

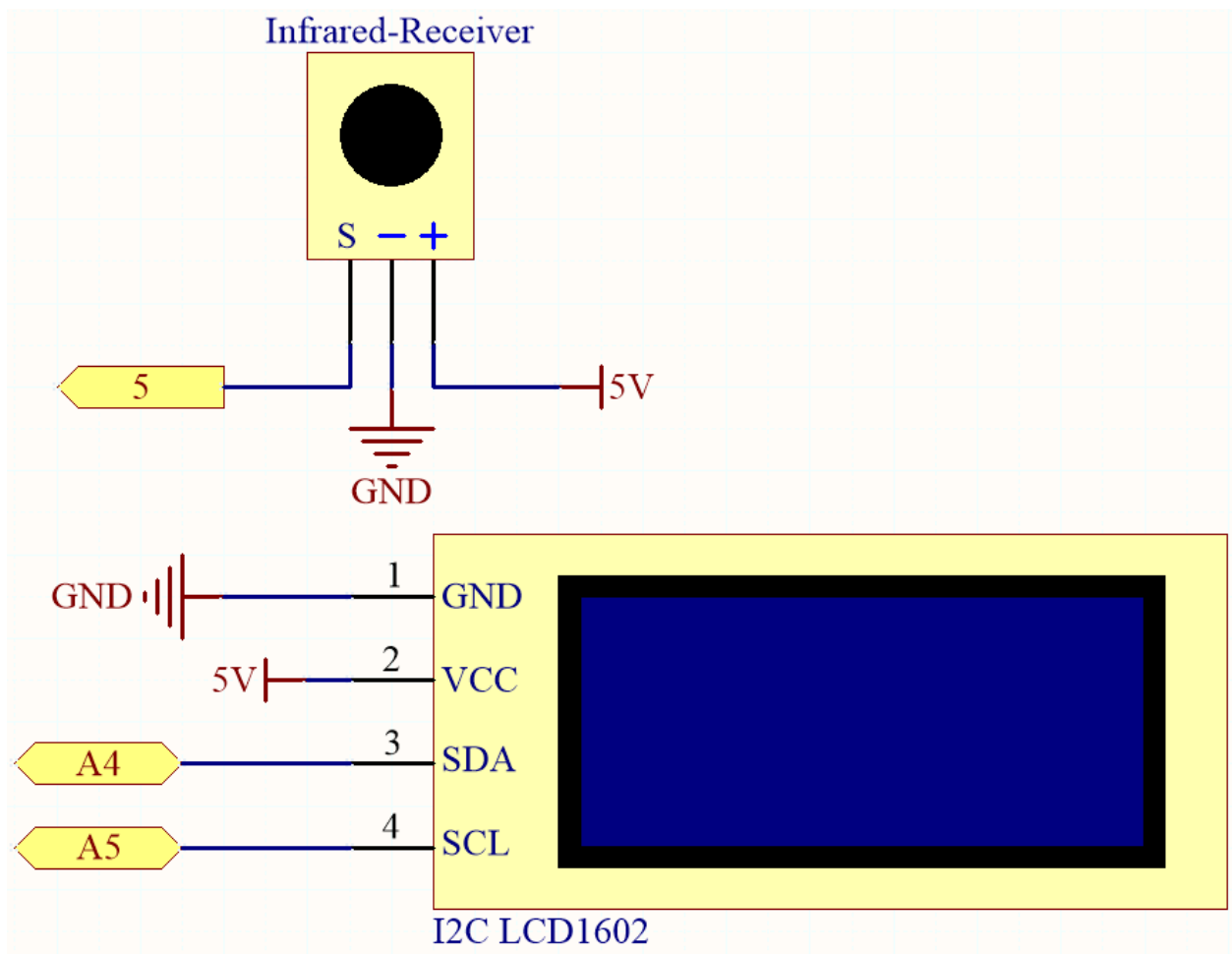
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

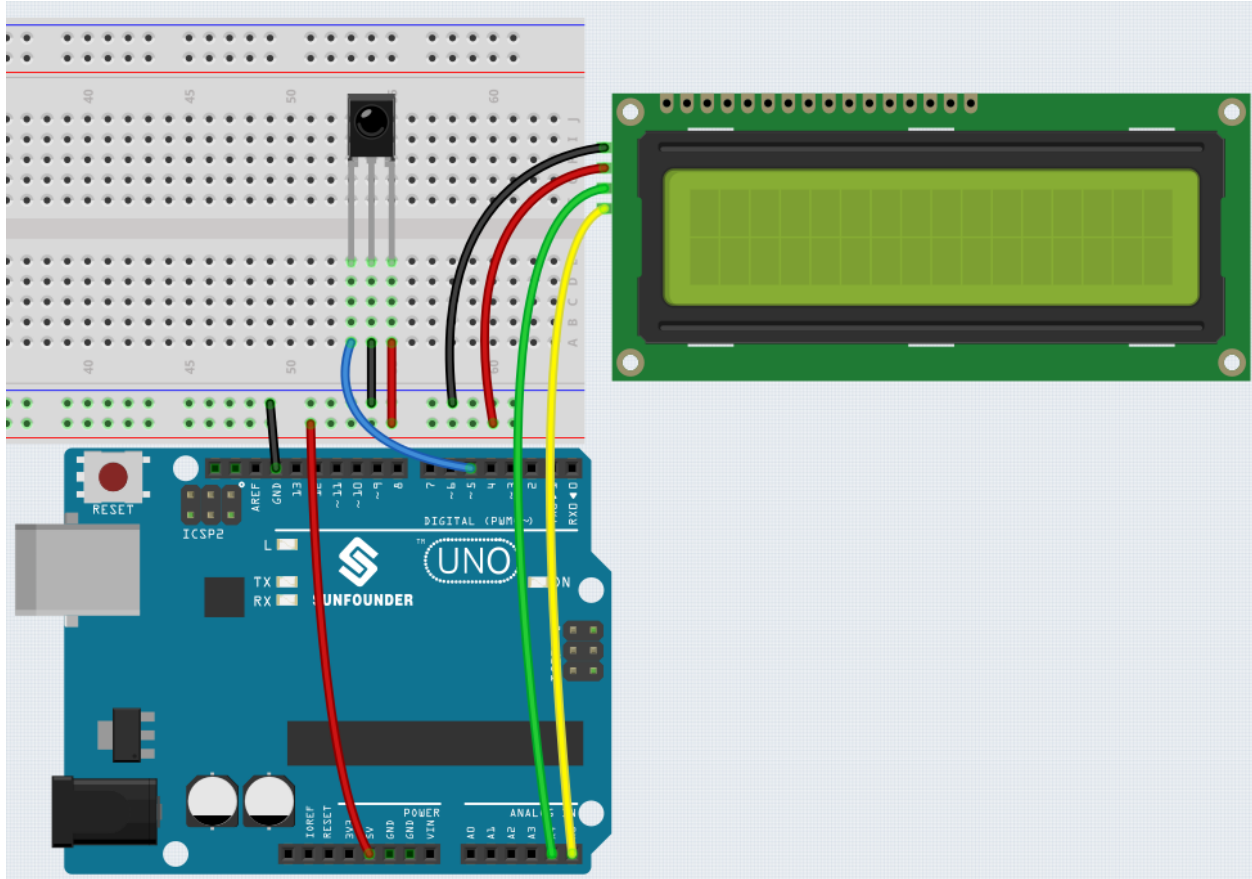
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>I2C LCD1602</i>	
<i>IR-Empfänger</i>	-

Schaltplan

In diesem Beispiel erfolgt die Verdrahtung des LCD1602 und des Infrarot-Empfangsmoduls wie folgt.



Verdrahtung



Code

Bemerkung:

- Sie können die Datei 6.6.guess_number.ino direkt unter dem Pfad 3in1-kit\basic_project\6.6.guess_number öffnen.
- Oder kopieren Sie diesen Code in die Arduino IDE 1/2.
- Die Bibliotheken LiquidCrystal I2C und IRremote werden hier verwendet. Sie können sie aus dem **Library Manager** installieren.

Nach erfolgreichem Hochladen des Codes erscheinen die Willkommenszeichen auf dem LCD1602. Drücken Sie nun die Nummer entsprechend dem Bereichshinweis auf dem Bildschirm. Die Anzeige wird immer kleiner, es sei denn, Sie erraten die Glückszahl.

Bemerkung: Wenn der Code und die Verdrahtung in Ordnung sind, das LCD aber trotzdem keinen Inhalt anzeigt, können Sie das Potentiometer auf der Rückseite drehen, um den Kontrast zu erhöhen.

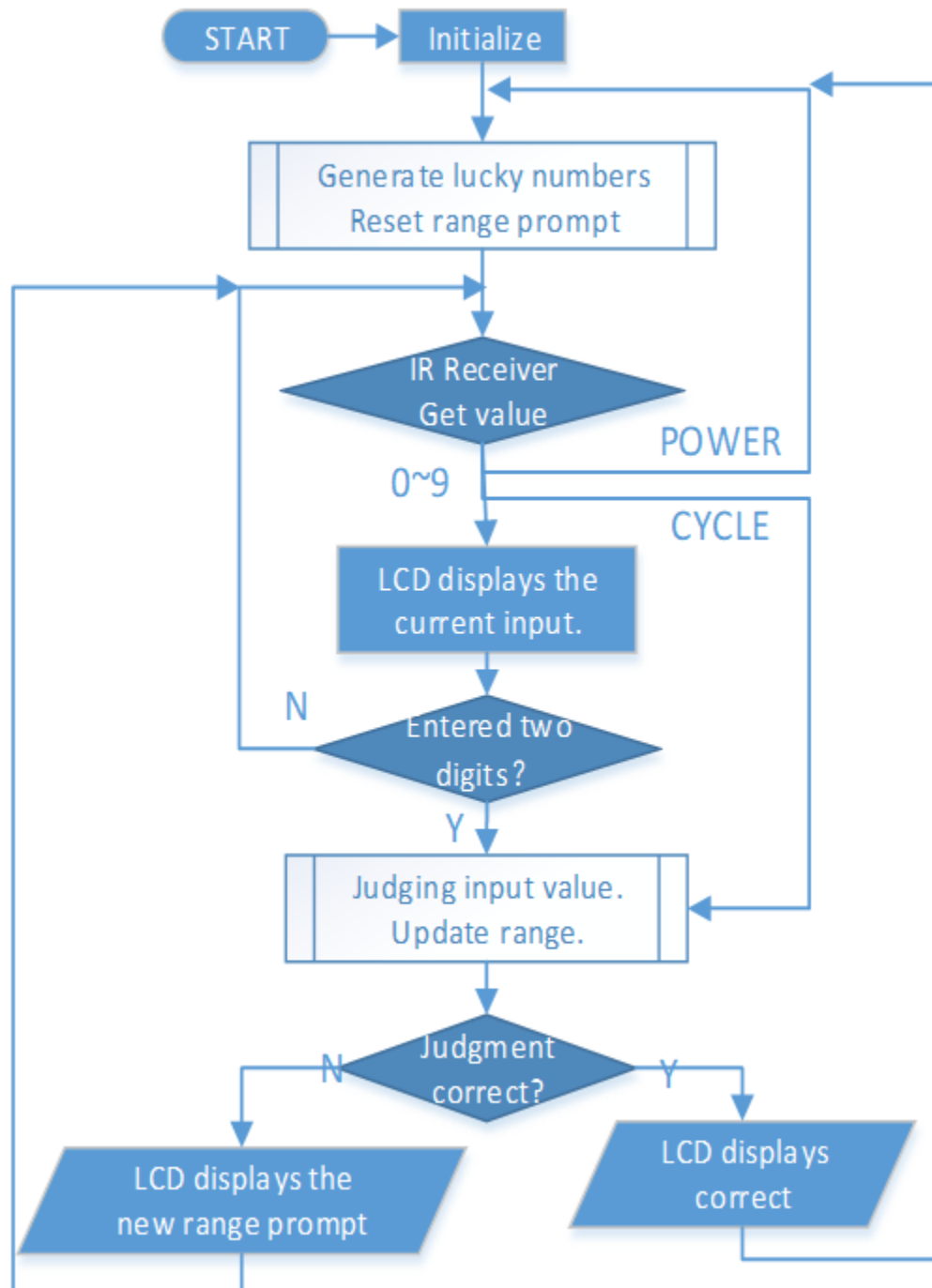
Wie funktioniert das?

Um das Zahlerraten-Spiel 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 Nummernbereichshinweis wird auf 0 ~ 99 zurückgesetzt.
2. Das LCD zeigt die eingetragene Zahl und den Nummernbereichshinweis an.

3. Nach Eingabe von zwei Ziffern erscheint automatisch ein Ergebnisurteil.
4. Wenn Sie eine einzelne Ziffer eingeben, können Sie die CYCLE-Taste (die Taste in der Mitte der Fernbedienung) drücken, um das Ergebnisurteil zu starten.
5. Wenn die Antwort nicht erraten wird, wird der neue Nummernbereichshinweis angezeigt (wenn die Glückszahl 51 ist und Sie 50 eingeben, ändert sich der Nummernbereichshinweis zu 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.

Zusammenfassend zeigt der Ablauf des Projekts der Ablaufdiagramm.



Ich bin mir sicher, dass Sie bereits viele verschiedene intelligente Roboter-Autos gesehen haben. Ihre Grundfunktionen sind ähnlich: grundlegende Bewegung, Hindernisvermeidung, Linienverfolgung, Nachfahren und Steuerung per Fernbedienung usw.

Hier verwenden wir die einfachste Struktur, um ein intelligentes Roboter-Auto zu bauen, das ebenfalls alle oben genannten Funktionen ausführen kann. Zusätzlich können Sie es mit Ihrem Handy steuern. Bitte beziehen Sie sich für das Tutorial auf [8. IoT Auto](#).

Montageanleitung

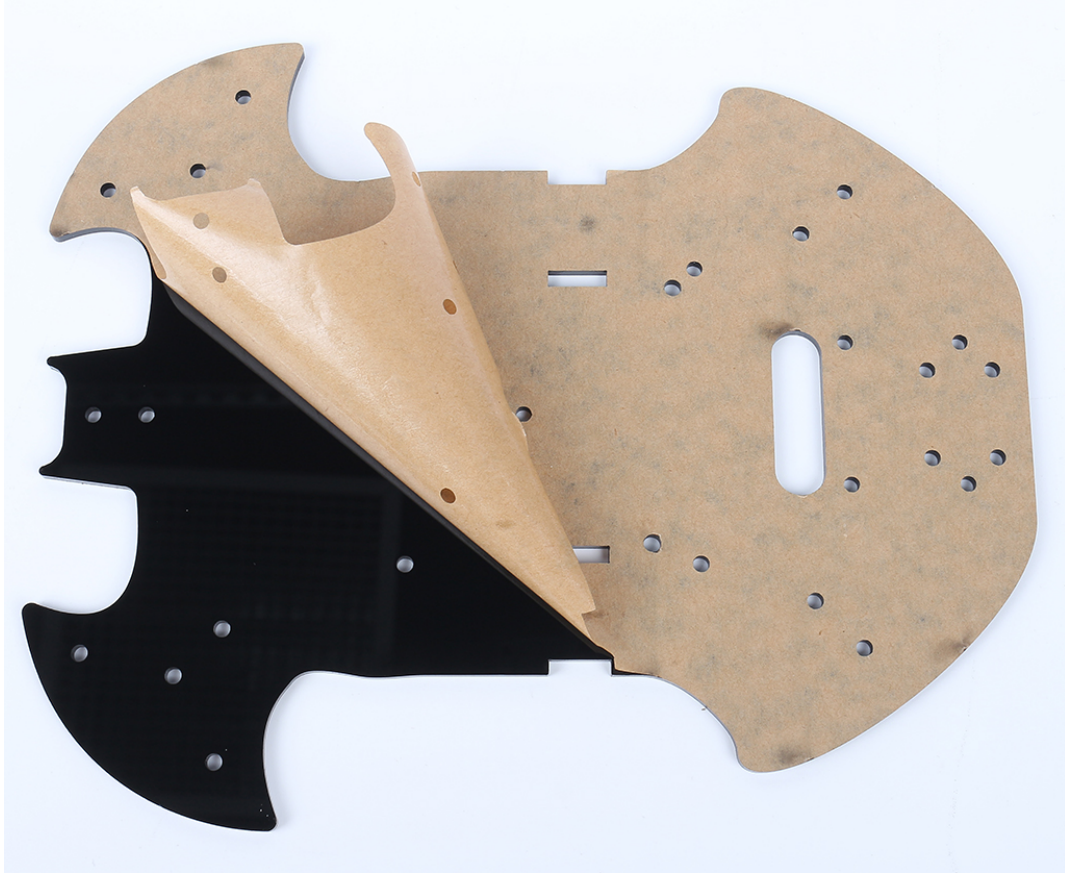
6.1 Auto zusammenbauen

Video

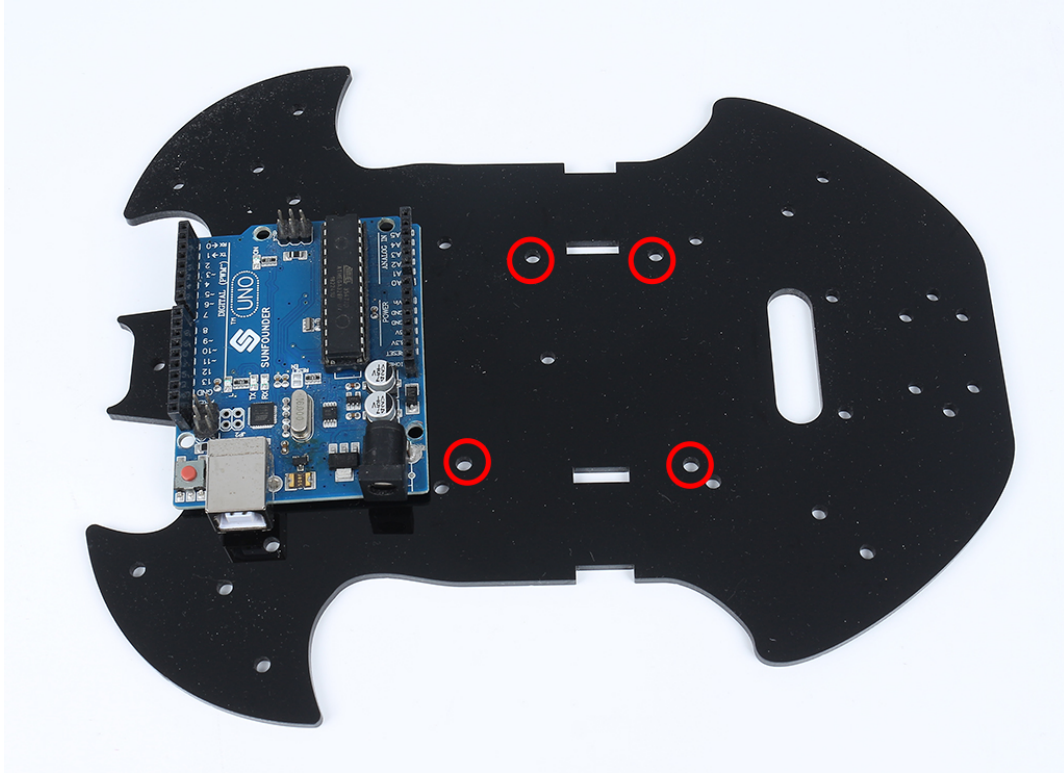
Schritte

Bitte befolgen Sie die untenstehenden Schritte, um den Zusammenbau des Autos abzuschließen.

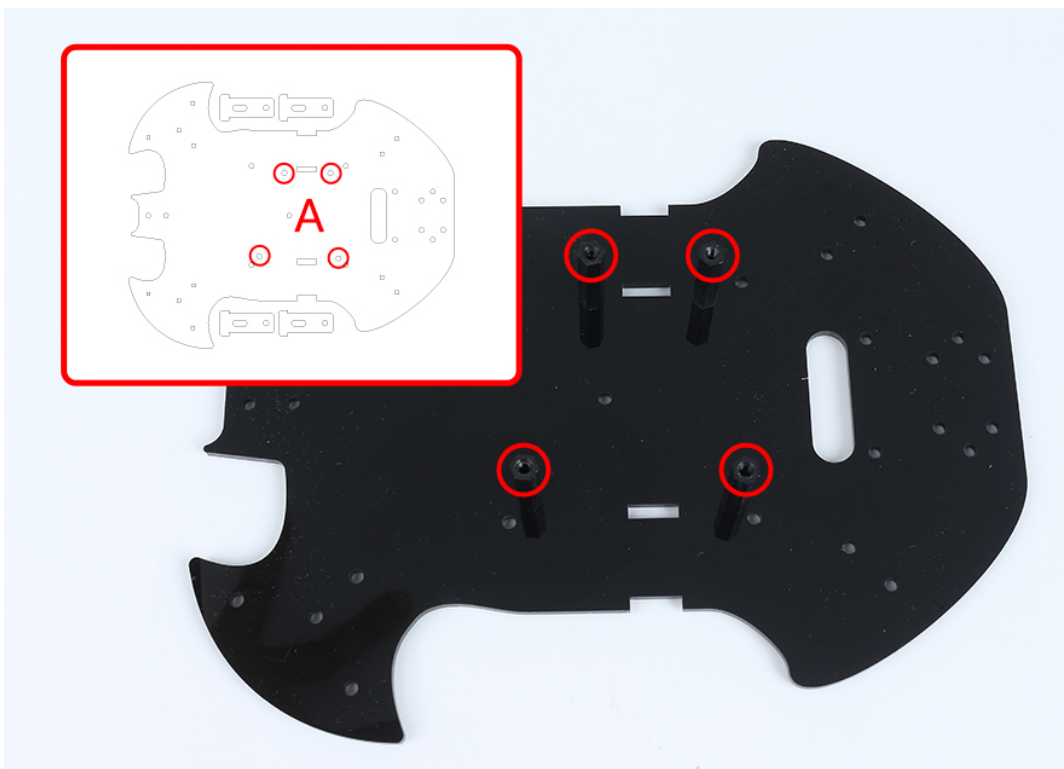
1. Entfernen Sie die Schutzfolie vom Acryl.



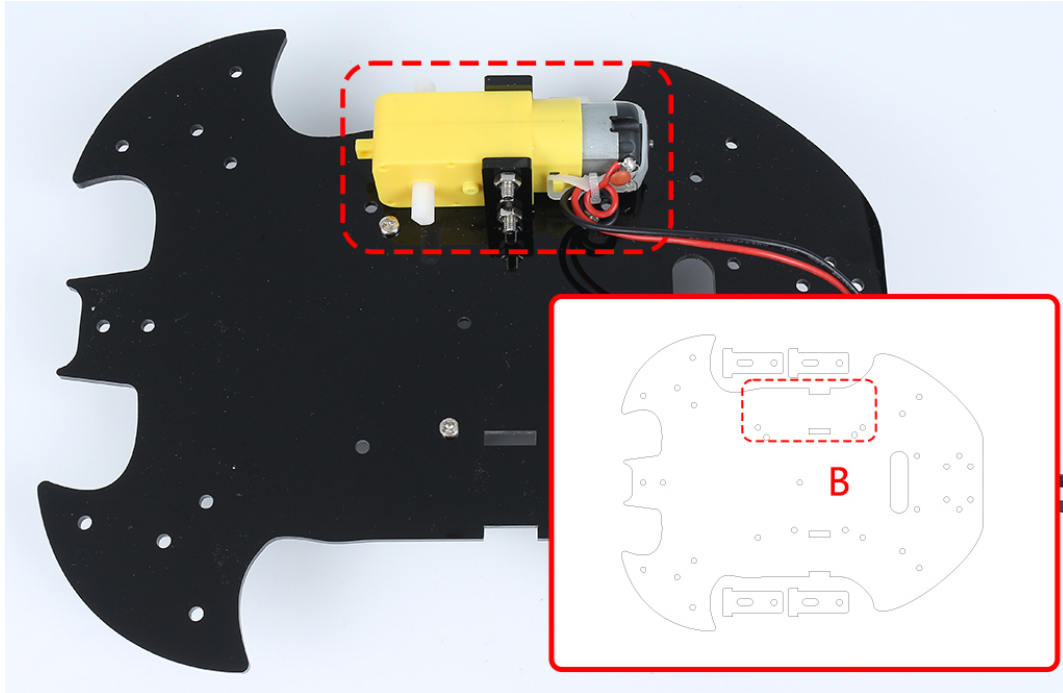
2. Legen Sie die Platine wie im Bild gezeigt auf den Tisch. Die Seite, die dasselbe Loch wie das R3-Board hat, nennen wir A; die Rückseite ist B. Dies wird Ihnen helfen, Fehler während des Zusammenbaus zu vermeiden.



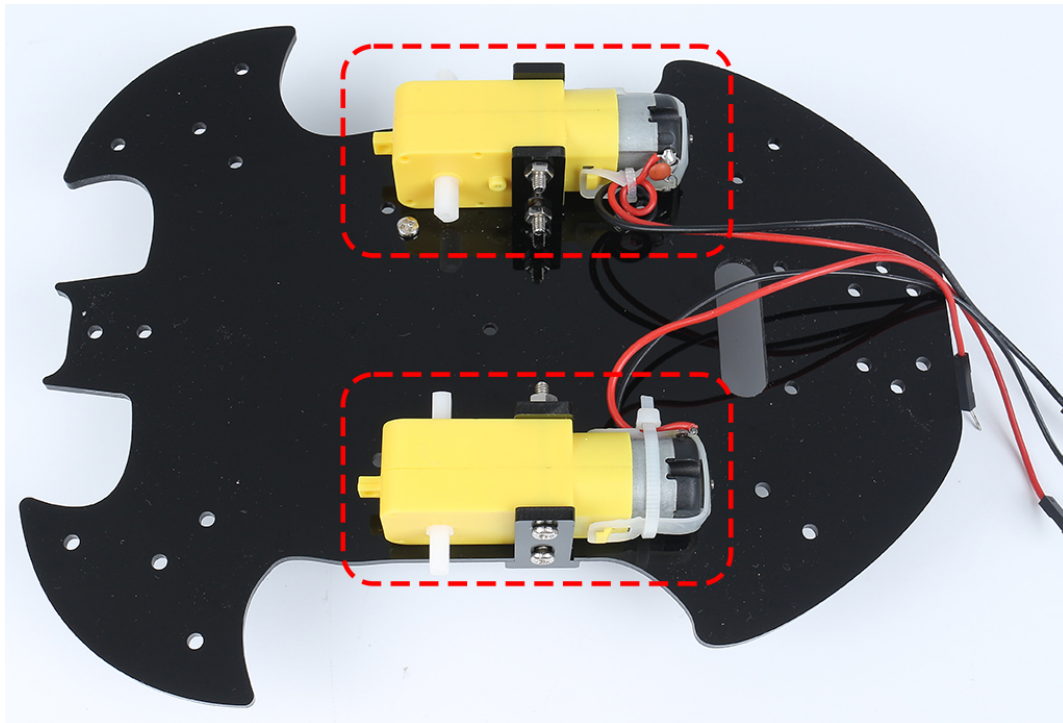
3. Montieren Sie den **M3x24mm Abstandshalter** mit **M3x6mm Schrauben** wie unten gezeigt.



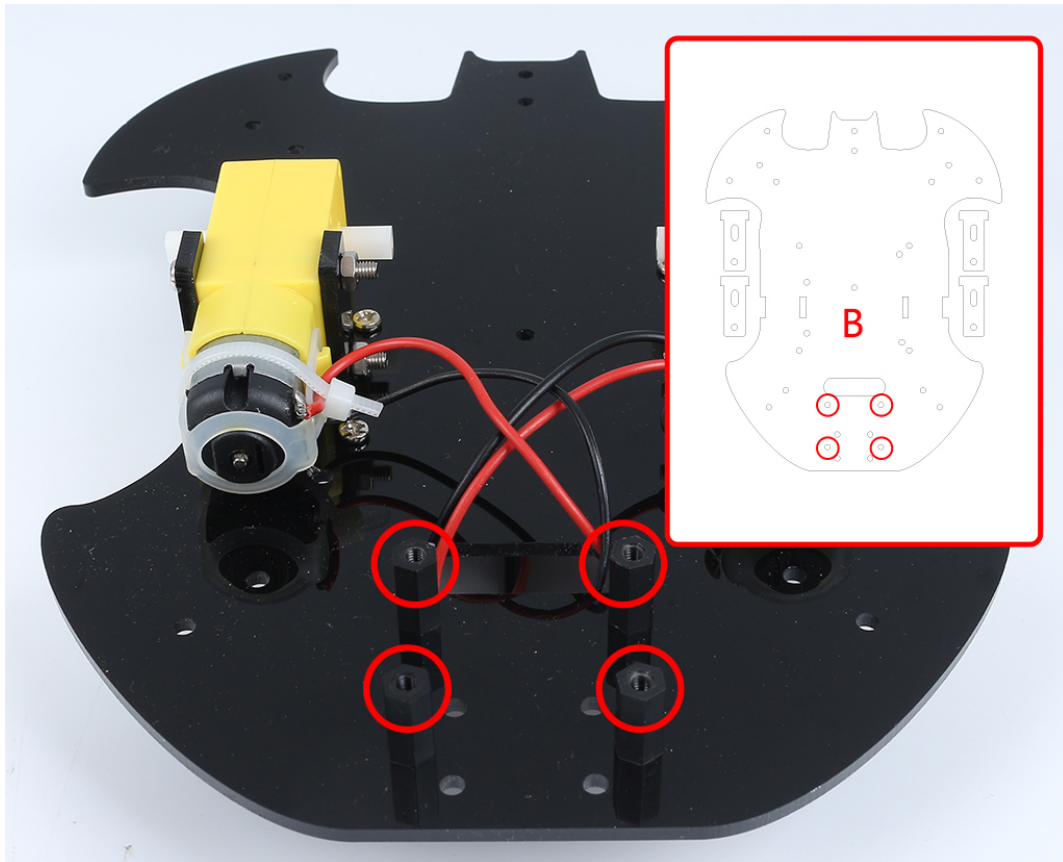
4. Drehen Sie die Platte zu Seite B und verwenden Sie **M3x30mm Schrauben** und **M3 Muttern**, um den TT Motor anzubringen. Zwei Details hier: 1 - die Ausgangswelle zeigt zur fledermausförmigen Seite; 2 - das Motorkabel zeigt nach innen.



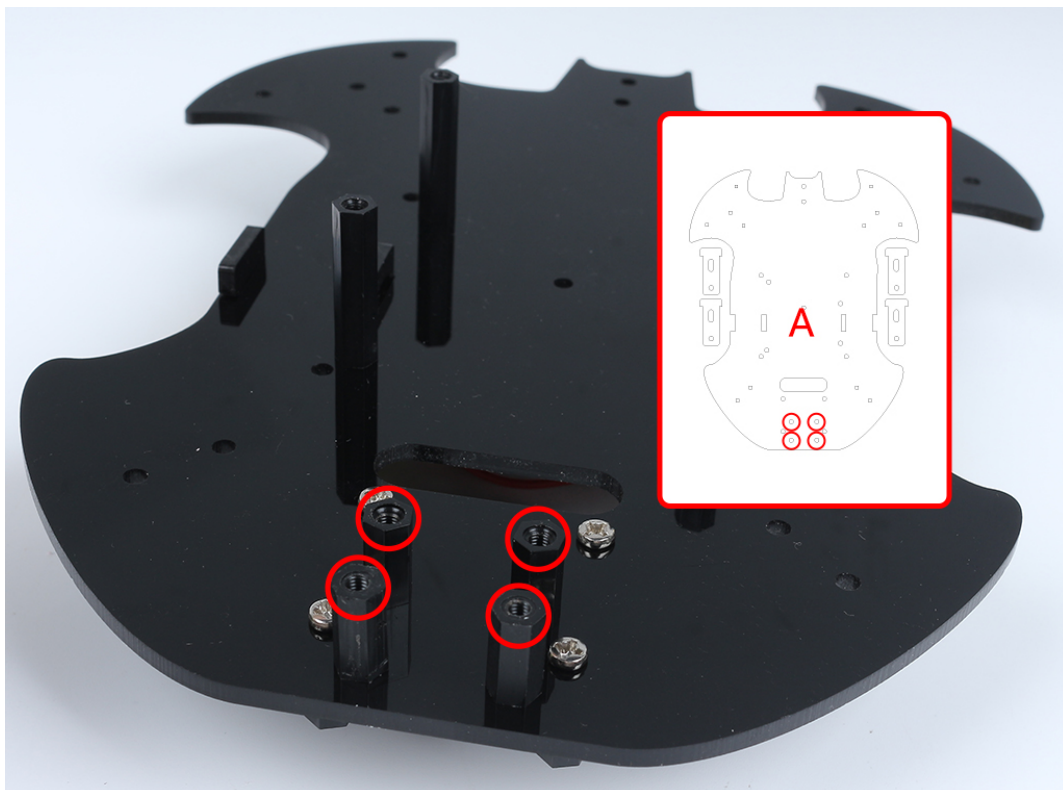
5. Montieren Sie einen weiteren TT Motor. Achten Sie dabei auf die gleiche Ausrichtung der Ausgangswelle und die Richtung des Kabels.



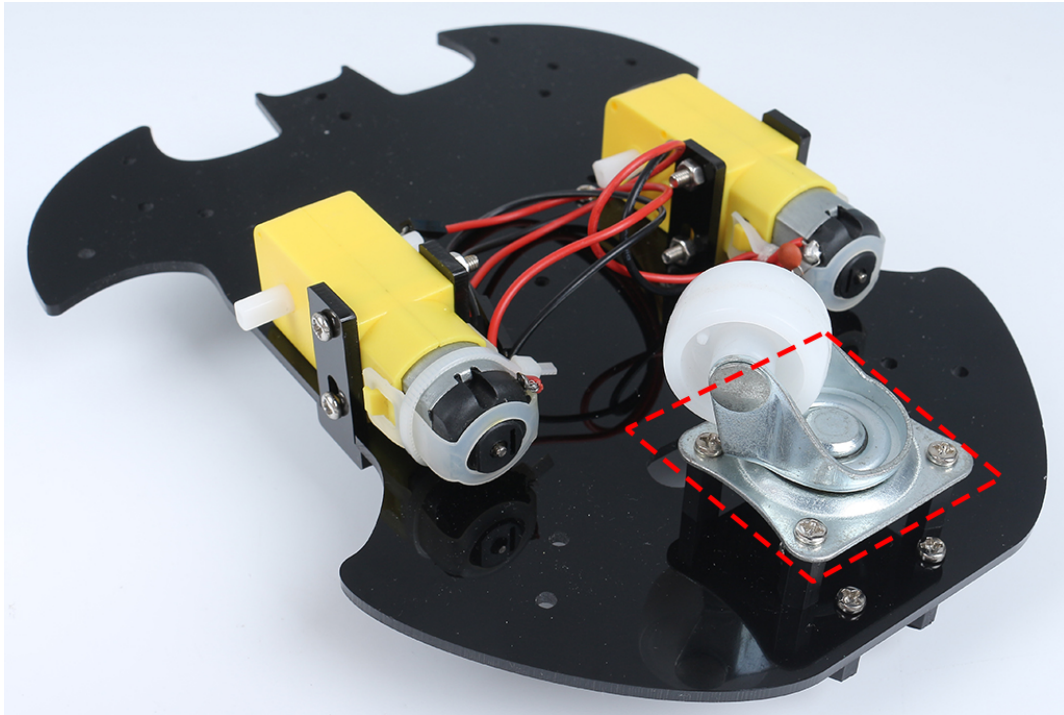
6. Verwenden Sie **M3x6mm Schrauben**, um den **M3x10mm Abstandshalter** wie unten gezeigt zu montieren.



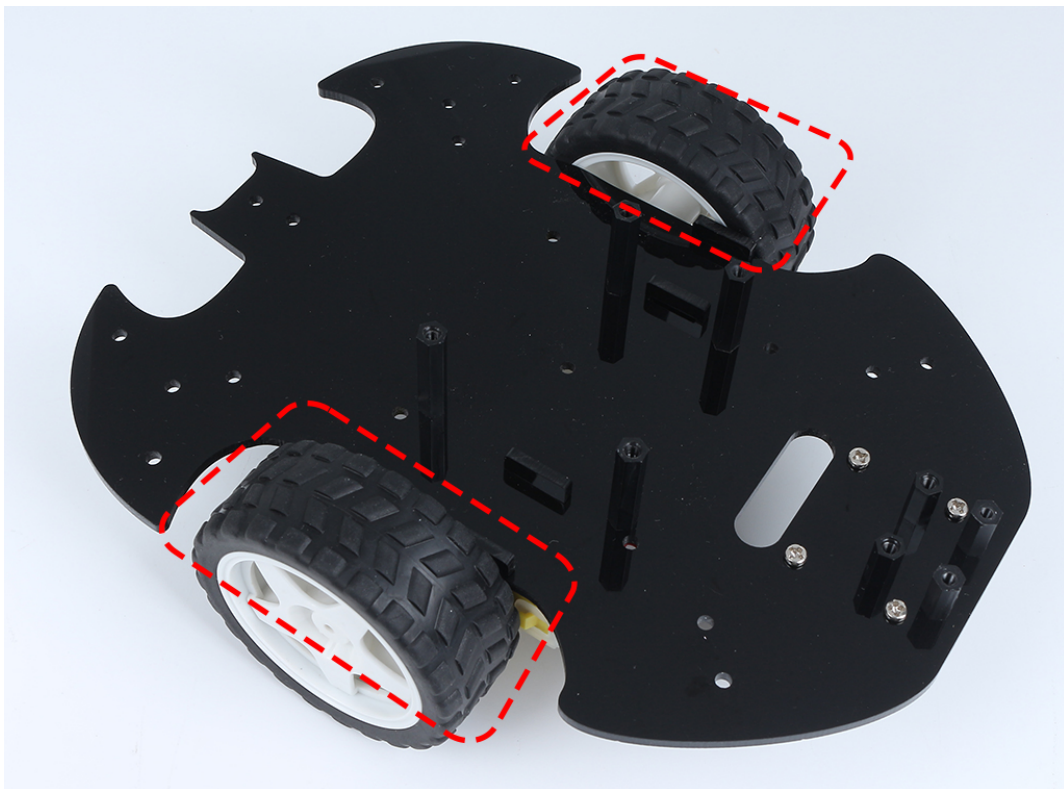
7. Befestigen Sie den **M2.5x11mm Abstandshalter** mit **M2.5x6mm Schrauben** an der Rückseite des Autos.



8. Verwenden Sie **M3x6mm Schrauben**, um das Universalrad zu montieren.



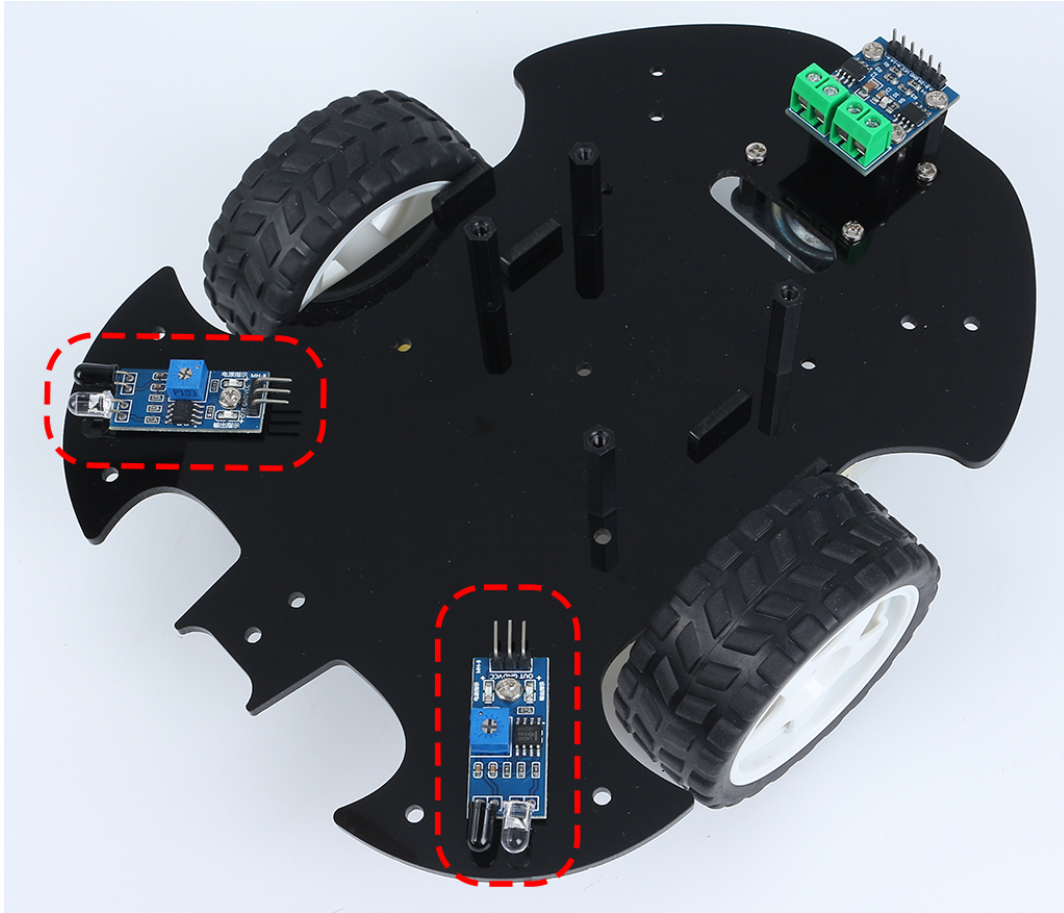
9. Setzen Sie die beiden Räder auf und die Grundstruktur des Autos ist fertig.



10. Befestigen Sie das L9110 Modul mit **M2.5x6mm Schrauben**.



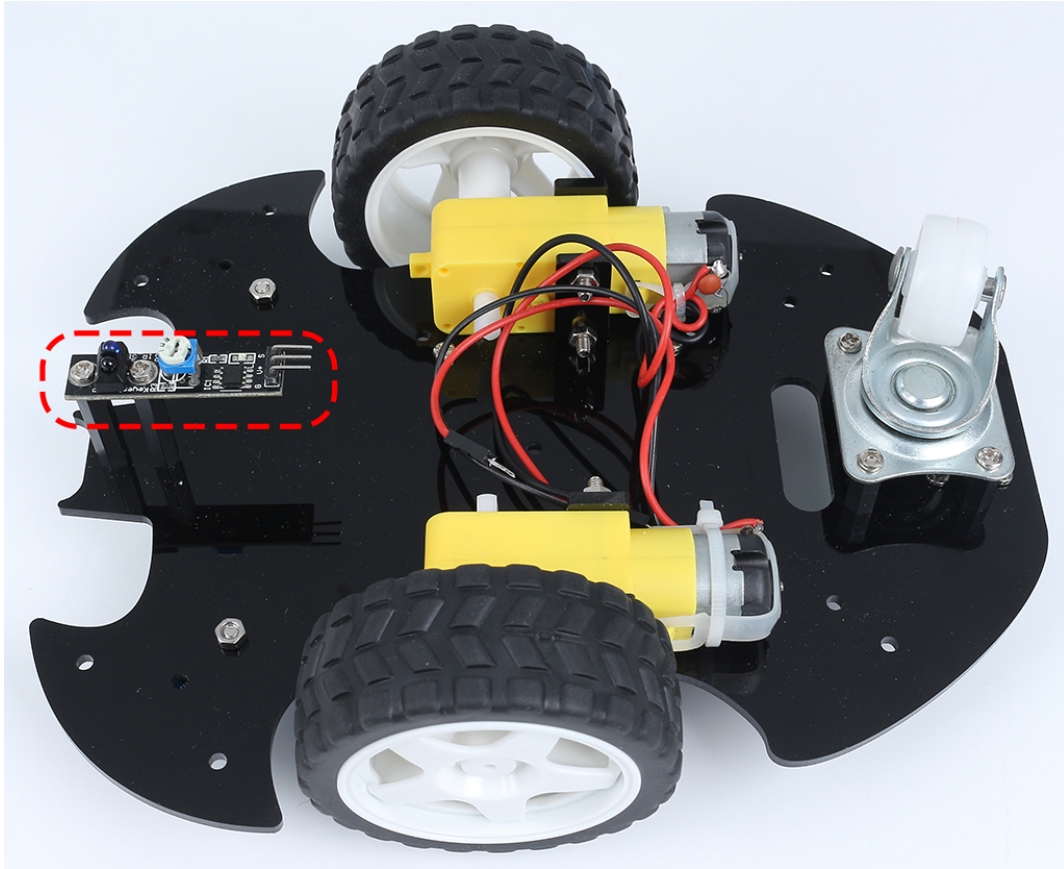
11. Montieren Sie die beiden IR-Hindernismodule mit **M3x10mm Schrauben** und **M3 Muttern**.



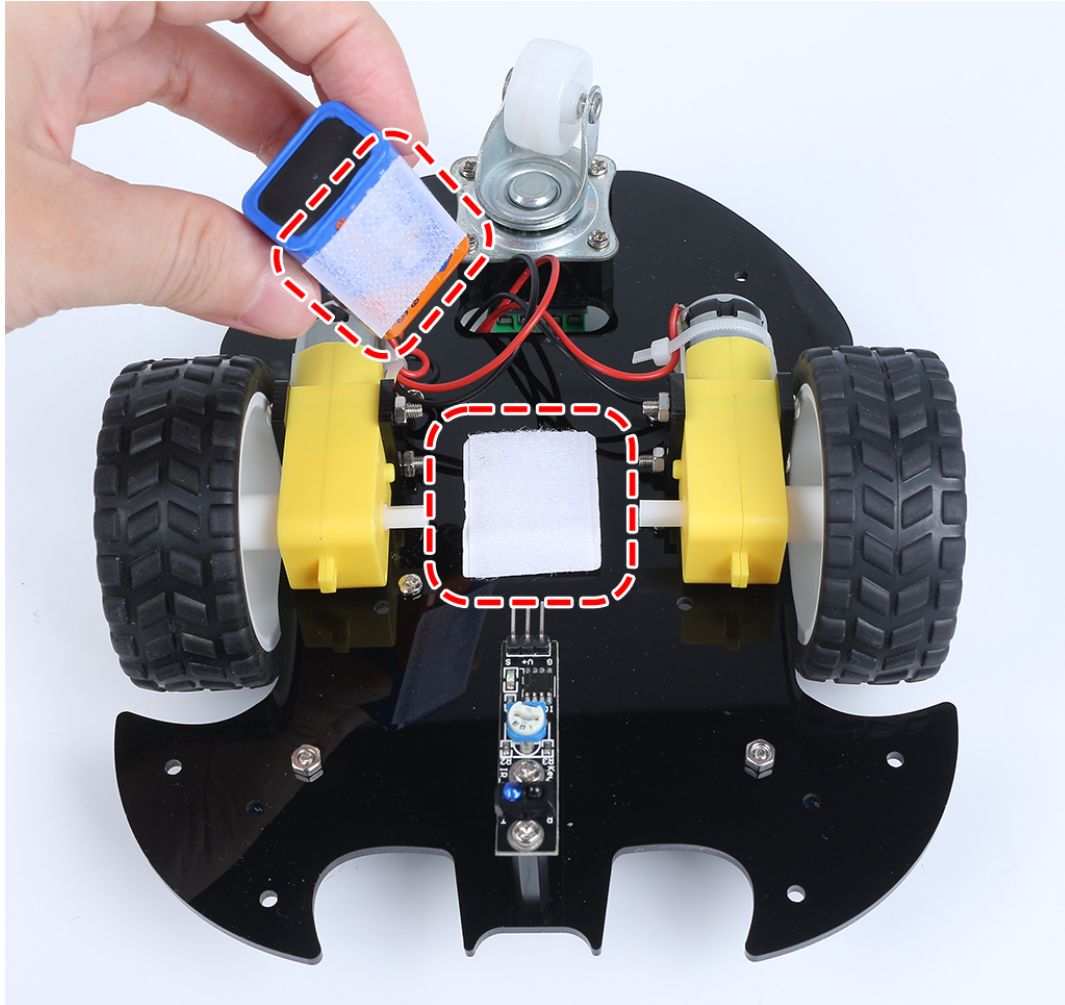
12. Drehen Sie zu Seite B und befestigen Sie das Linienverfolgungsmodul mit vier **M3x6mm Schrauben** und zwei **M3x24mm Abstandshaltern**.

Bemerkung: Es ist ratsam, zuerst die **M3x24mm Abstandshalter** am Linienverfolgungsmodul zu befestigen.

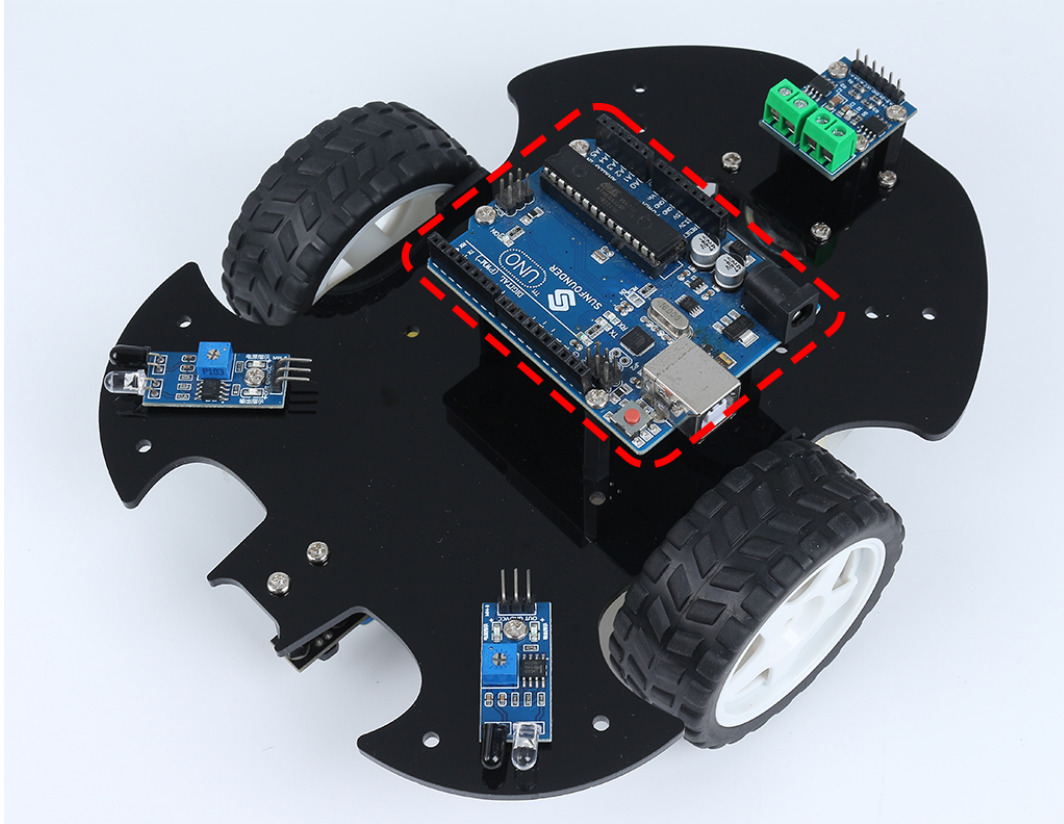
Ein wichtiger Hinweis: Die Pins des Liniensensors sind etwas weich und ragen leicht in Richtung der Löcher. Beim Anbringen der **M3x24mm Abstandshalter** drücken Sie die Sensorenpins vorsichtig beiseite.



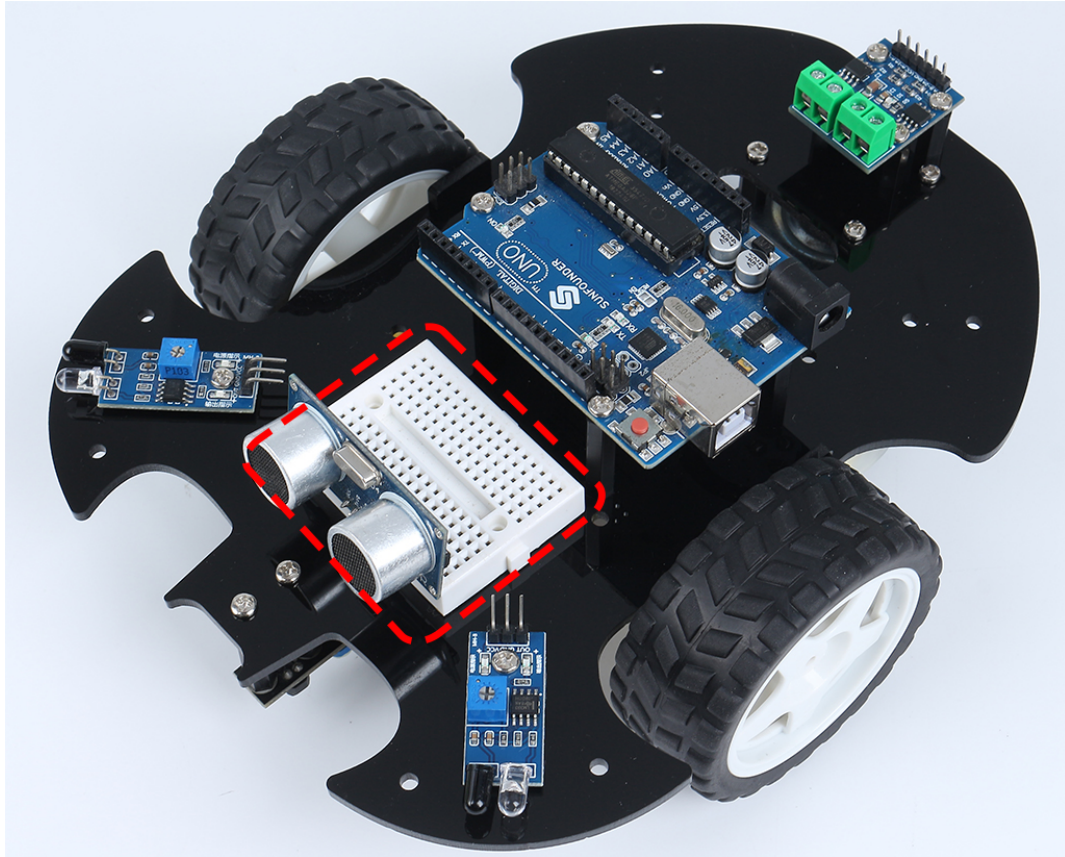
13. Kleben Sie den Klettverschluss auf die 9V-Batterie und setzen Sie den Batterieclip darauf. Kleben Sie den anderen Teil des Klettverschlusses auf das Auto, um die Batterie zu fixieren.



14. Drehen Sie das Auto zu Seite A und montieren Sie das R3-Board mit **M3x6mm Schrauben**.



15. Befestigen Sie das Steckbrett vorne am Auto. Danach können Sie je nach Projektanforderung verschiedene Komponenten (z. B. Ultraschallmodul) auf dem Steckbrett hinzufügen.



16. Um das Auto in Betrieb zu nehmen, müssen Sie es auch verdrahten und Code schreiben, was in den folgenden Abschnitten beschrieben wird.

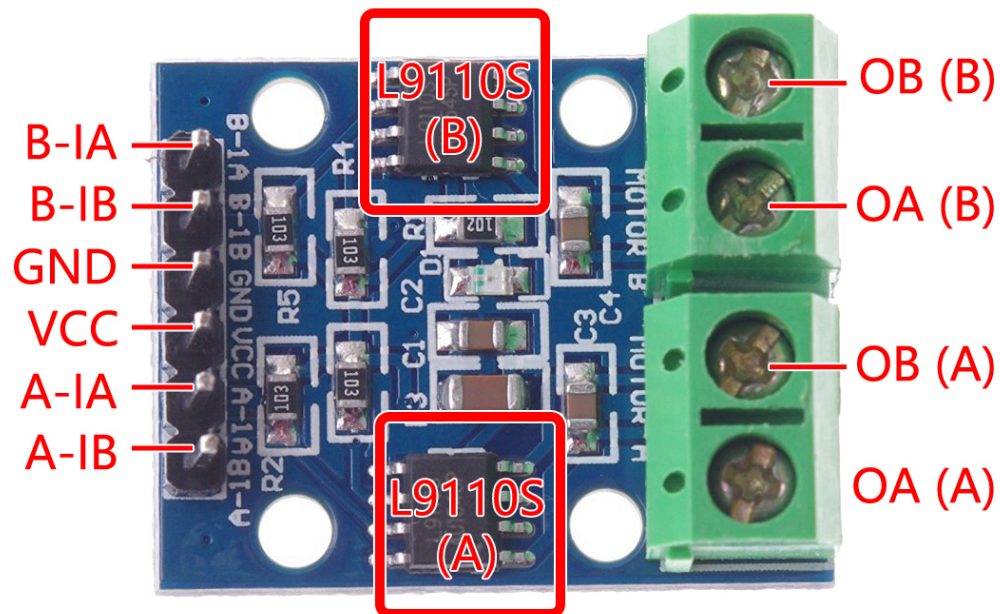
Projekte

Hier sind einige Projekte für das Auto, programmiert in C mit der Arduino IDE. Wenn Sie nicht besonders geübt im Umgang mit Arduino sind, können Sie sich auf [Einstieg in Arduino](#) beziehen.

Die folgenden Projekte sind in der Reihenfolge des Programmierschwierigkeitsgrades geschrieben. Es wird empfohlen, diese Bücher in der angegebenen Reihenfolge zu lesen.

Wenn Sie ein Auto mit Scratch programmieren möchten, beziehen Sie sich bitte auf: [Mit Scratch spielen](#).

6.2 1. Bewegung



Bevor wir mit der Programmierung beginnen, überprüfen wir das Arbeitsprinzip des L9110-Moduls.

Hier ist die Wahrheitstabelle des Motor B:

B-1A	B-1B(B-2A)	Zustand des Motor B
1	0	Dreht im Uhrzeigersinn
0	1	Dreht gegen den Uhrzeigersinn
0	0	Bremse
1	1	Stop

Hier ist die Wahrheitstabelle des Motor A:

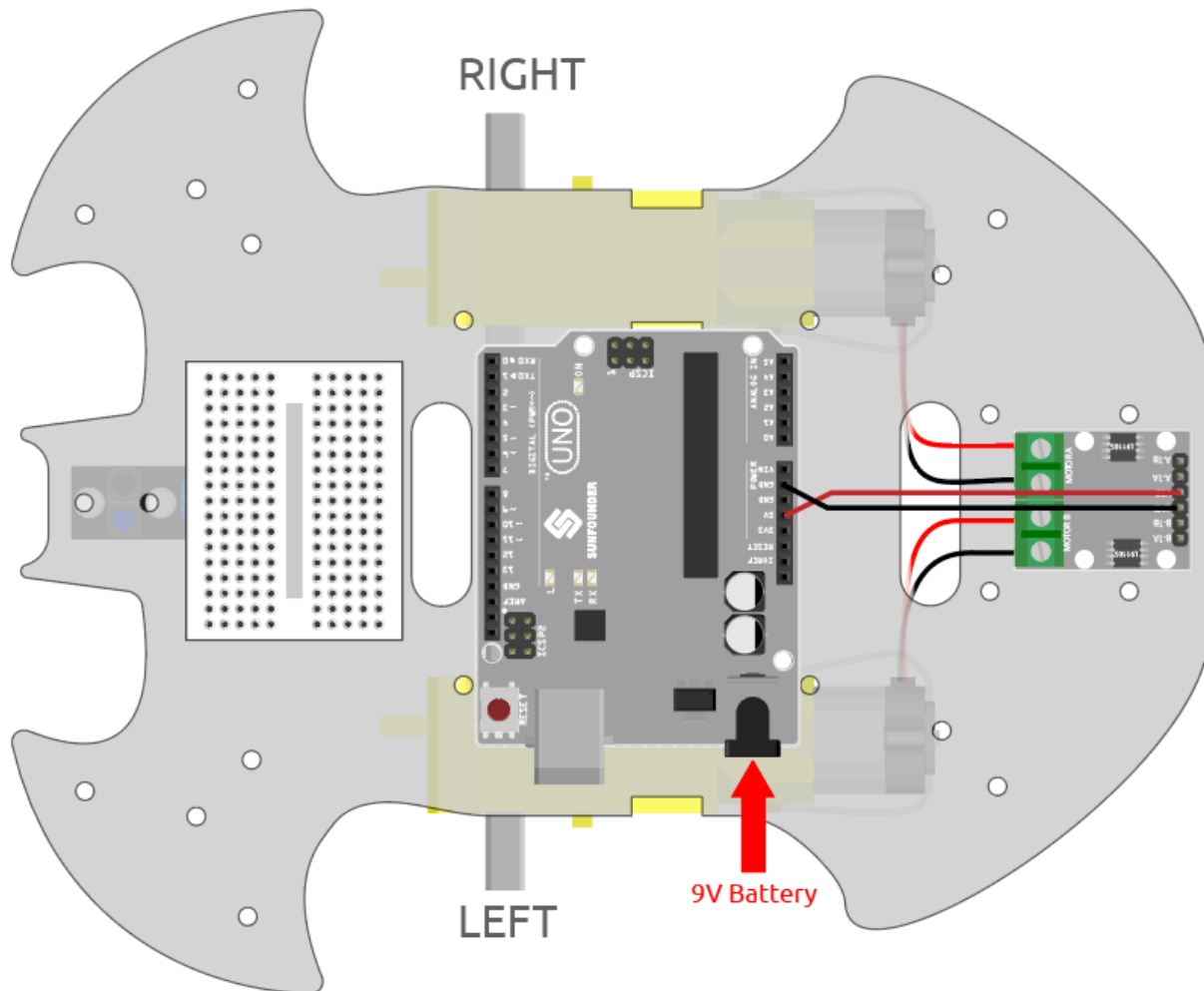
A-1A	A-1B	Zustand des Motor A
1	0	Dreht im Uhrzeigersinn
0	1	Dreht gegen den Uhrzeigersinn
0	0	Bremse
1	1	Stop

- *L9110 Motortreibermodul*

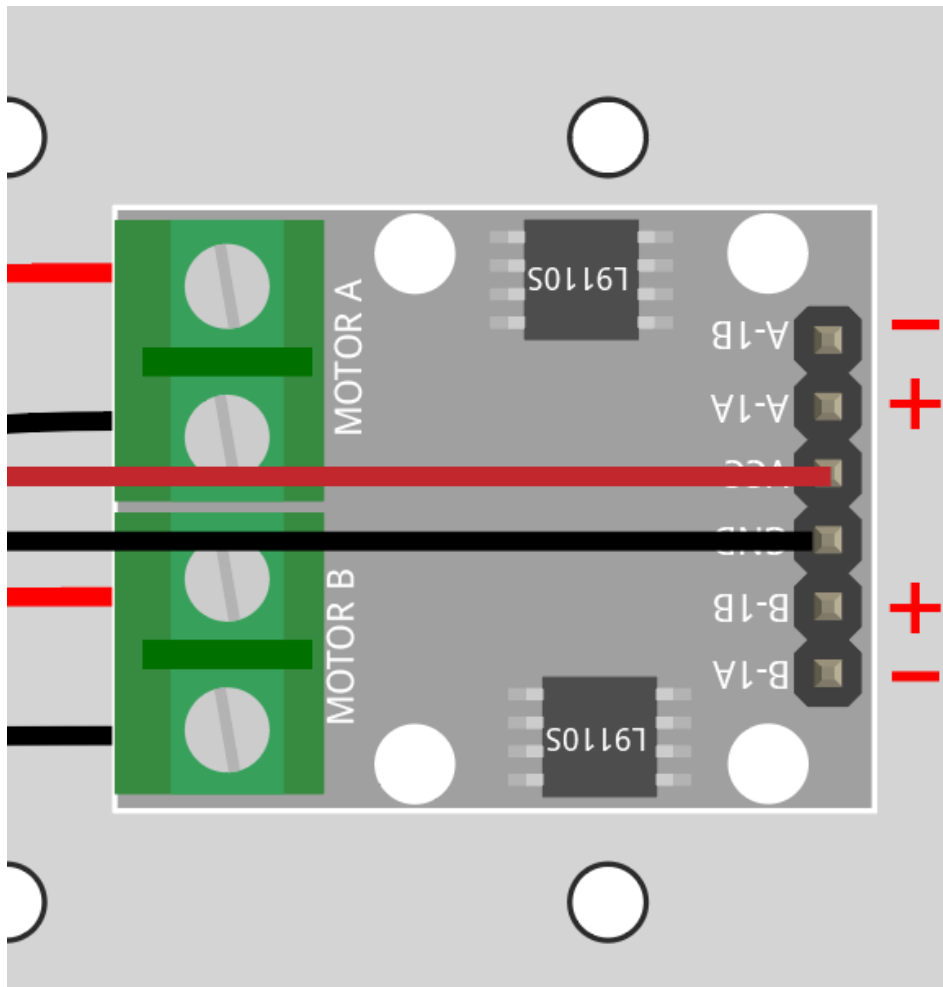
Vorwärts

Verbinden Sie nun den Eingang des L9110-Moduls direkt mit 12V bzw. GND, um das Auto zu bewegen.

1. Verbinden Sie das R3-Board, das L9110-Modul und die 2 Motoren.



2. Verbinden Sie B-1B(B-2A) und A-1A mit VCC und B-1A sowie A-1B mit GND. Dann sollten Sie das Auto vorwärts fahren sehen.

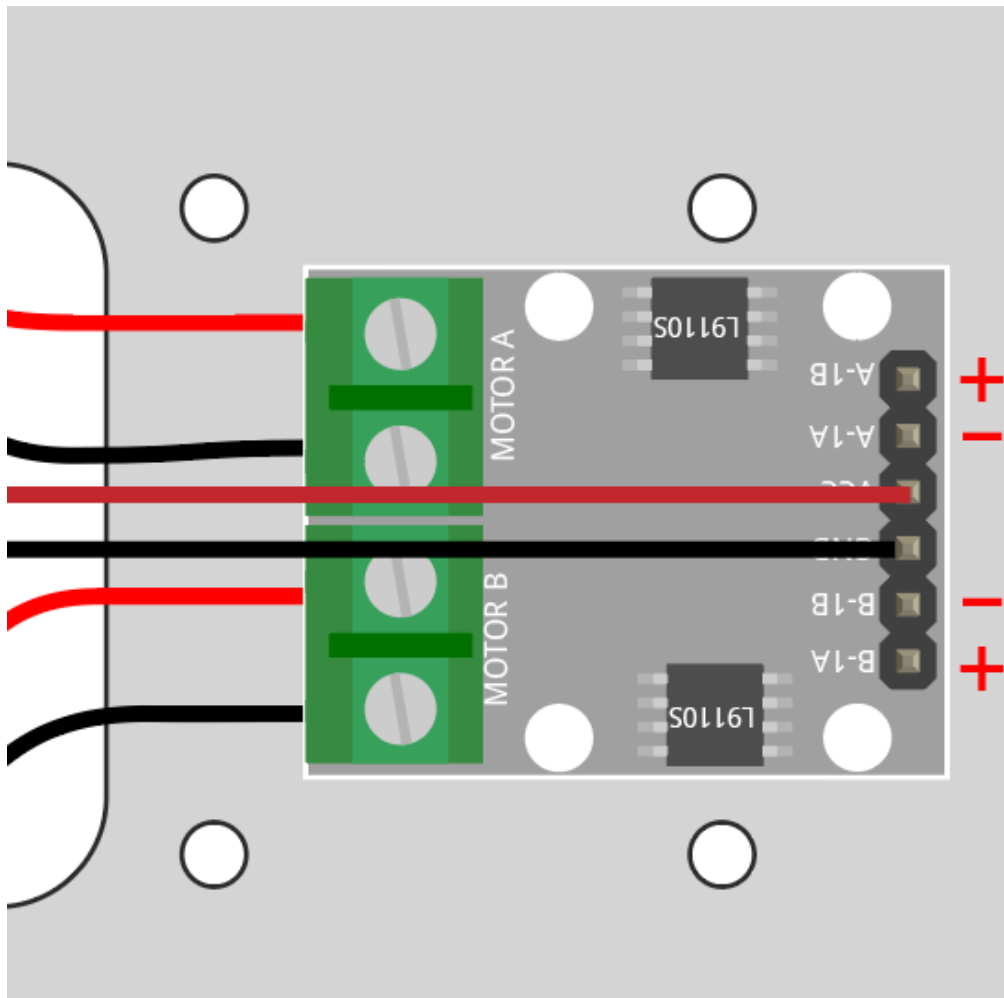


Wenn beide nicht vorwärts drehen, aber die folgenden Situationen auftreten, müssen Sie die Verkabelung der beiden Motoren neu justieren.

- Wenn beide Motoren gleichzeitig rückwärts drehen (linker Motor dreht 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 sich der linke Motor rückwärts dreht (Drehung im Uhrzeigersinn), tauschen Sie die Verkabelung von OA(B) und OB(B) des linken Motors.
- Wenn sich der rechte Motor rückwärts dreht (Drehung gegen den Uhrzeigersinn), tauschen Sie die Verkabelung von OA(A) und OB(A) des rechten Motors.

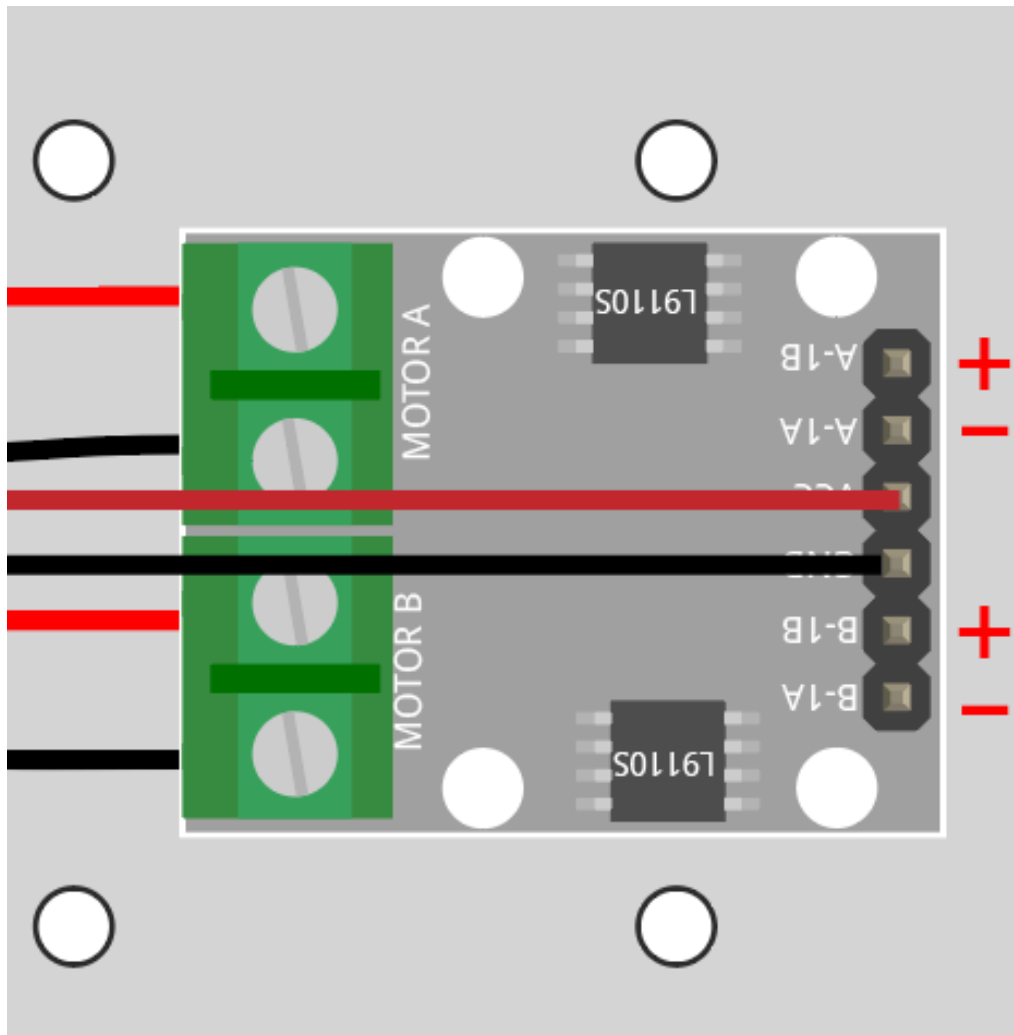
Rückwärts

Verbinden Sie B-1B(B-2A) und A-1A mit GND und B-1A sowie A-1B mit VCC. Dann sollten Sie das Auto rückwärts fahren sehen.



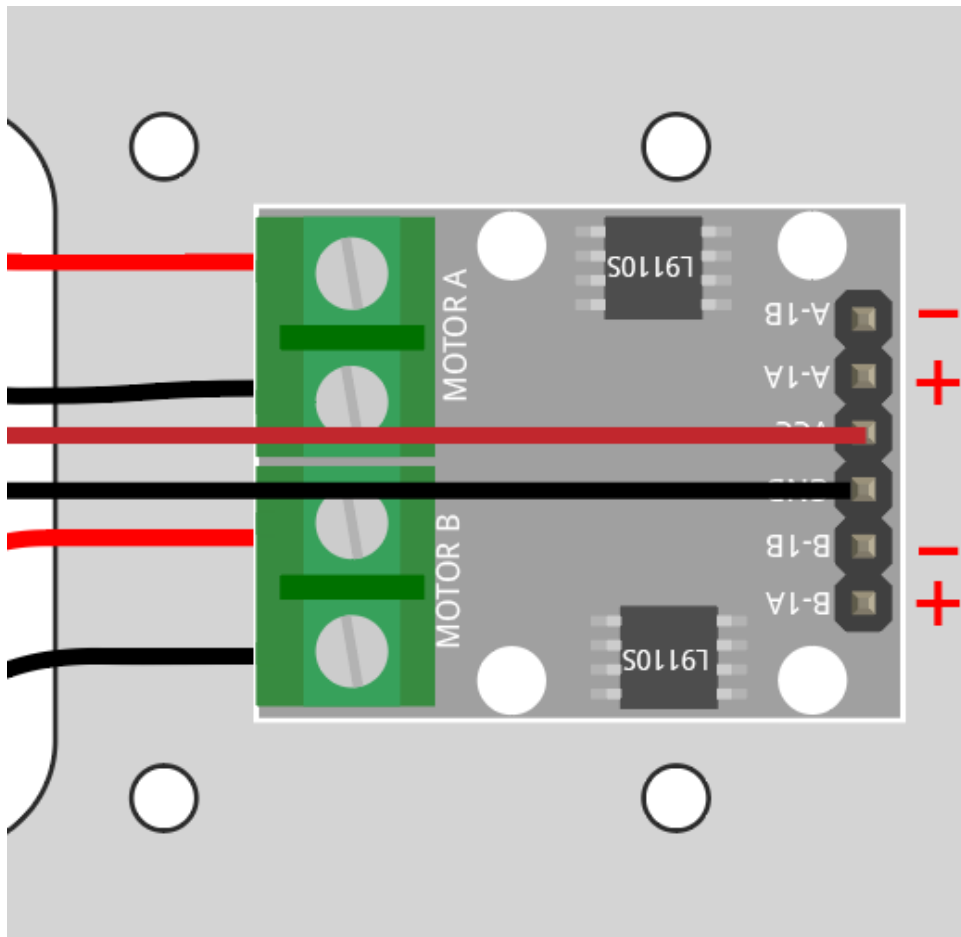
Links abbiegen

Wenn Sie das Auto nach links abbiegen lassen möchten, also beide Motoren im Uhrzeigersinn drehen möchten, müssen Sie B-1A und A-1A mit GND und B-1B(B-2A) sowie A-1B mit VCC verbinden.



Rechts abbiegen

Umgekehrt, wenn Sie das Auto nach rechts abbiegen lassen möchten, also beide Motoren gegen den Uhrzeigersinn drehen möchten, müssen Sie B-1A und A-1A mit VCC und B-1B(B-2A) sowie A-1B mit GND verbinden.



Stopp

Um den Motor zu stoppen, verbinden Sie die Eingänge auf der gleichen Seite gleichzeitig mit 12V oder GND, z. B. verbinden Sie B-1A und B-1B(B-2A) gleichzeitig mit 12V oder 5V und ebenso für A-1A und A-1B.

Dies ist natürlich theoretisch und wird später beim Steuern mit Code benötigt. Hier kann das Entfernen der Stromversorgung vom Auto es stoppen.

6.3 2. Bewegung durch Code

Im vorherigen Projekt haben wir versucht, den Betrieb des Motors durch Verwendung verschiedener Pegelsignale für den Eingang des L9110-Moduls zu steuern.

Wenn wir die Pegelsignale über 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	TEILE IN DIESEM SET	LINK
3-in-1 Starter-Set	380+	

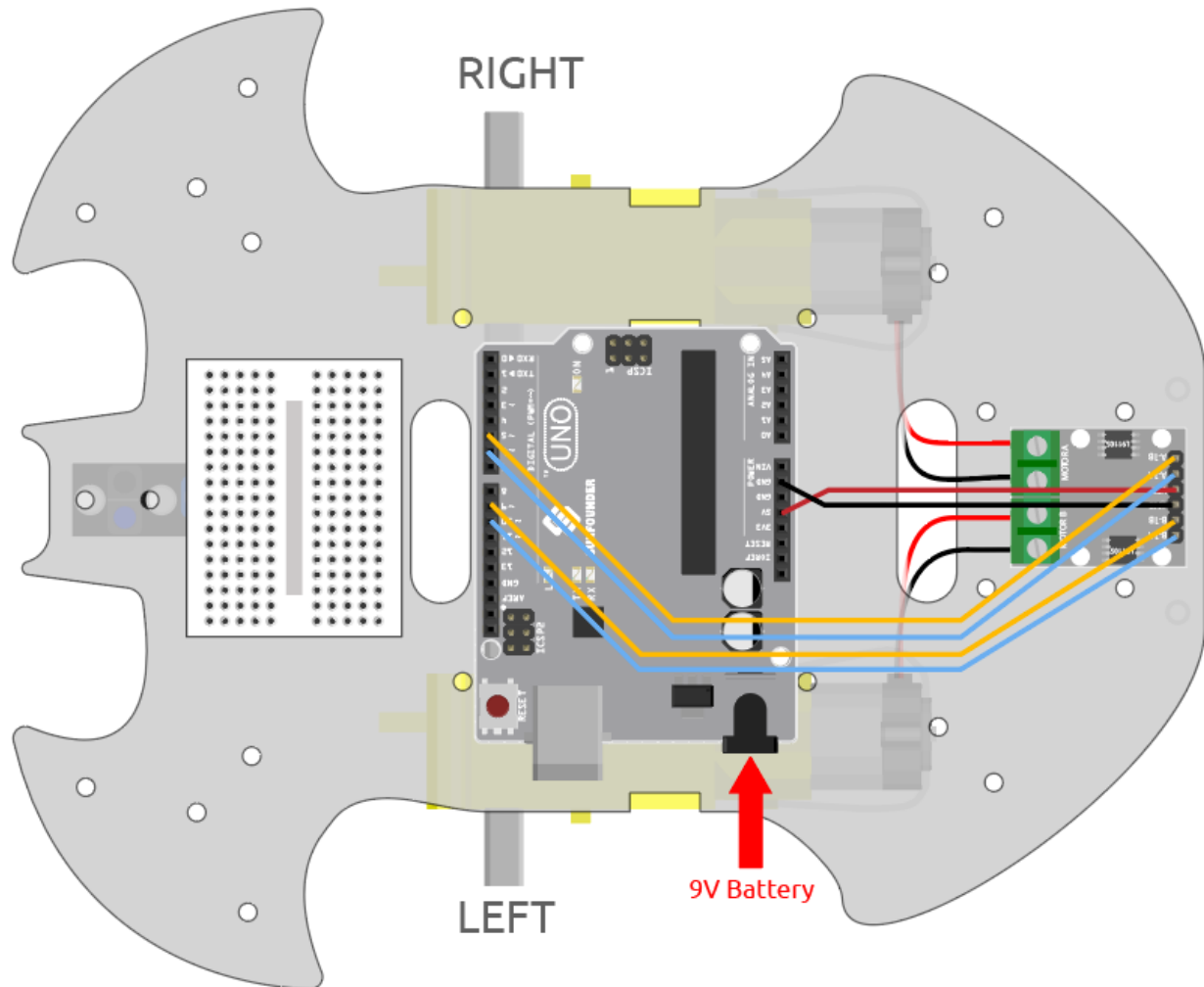
Sie können diese auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-

Verdrahtung

Verbinden Sie die Drähte zwischen dem L9110-Modul und dem R3-Board gemäß dem untenstehenden Diagramm.

L9110 Modul	R3 Board	Motor
A-1B	5	
A-1A	6	
B-1B(B-2A)	9	
B-1A	10	
OB(B)		Schwarzer Draht des rechten Motors
OA(B)		Roter Draht des rechten Motors
OB(A)		Schwarzer Draht des linken Motors
OA(A)		Roter Draht 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 auf das R3-Board richten Sie das Linienverfolgungsmodul unter dem Auto auf die schwarze Linie aus und Sie werden sehen, wie das Auto der Linie folgt.

Wie funktioniert das?

Dieses Projekt ist im Wesentlichen dasselbe wie das vorherige und beinhaltet, dass das Auto vorwärts, rückwärts, links und rechts fährt, sowie durch Bereitstellung verschiedener Signalpegel an den Eingabepins des L9110-Moduls anhält.

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 verschiedene 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 auf einen digitalen Pin. Wenn der Pin mit `pinMode()` als OUTPUT konfiguriert wurde, wird seine Spannung auf den entsprechenden Wert gesetzt: 5V (oder 3,3V auf 3,3V-Platinen) für HIGH, 0V (Masse) für LOW.

- `pinMode(pin, mode)`

- pin: die Arduino-Pin-Nummer, für die der Modus eingestellt werden soll.
- mode: INPUT, OUTPUT oder INPUT_PULLUP.

Konfiguriert den angegebenen Pin, um sich entweder als Eingang oder als Ausgang zu verhalten.

- `delay(ms)`
 - ms: die Anzahl der Millisekunden zum Pausieren. Zulässige Datentypen: unsigned long.
- Pausiert das Programm für die angegebene Zeit (in Millisekunden). (Es gibt 1000 Millisekunden in einer Sekunde.)

6.4 3. Beschleunigung

Neben dem digitalen Signal (HOCH/NIEDRIG) kann der Eingang des L9110-Moduls auch ein PWM-Signal empfangen, um die Geschwindigkeit des Ausgangs zu steuern.

Mit anderen Worten, wir können `AnalogWrite()` verwenden, um die Bewegungsgeschwindigkeit des Autos zu steuern.

In diesem Projekt lassen wir das Auto seine Vorwärtsgeschwindigkeit allmählich ändern, zuerst beschleunigen und dann abbremesen.

Verdrahtung

Dieses Projekt hat die gleiche Verdrahtung wie [2. Bewegung 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 allmählich beschleunigen und dann allmählich abbremesen.

Wie funktioniert das?

Das Ziel dieses Projekts ist es, verschiedene PWM-Werte an die Eingangspins des L9110-Moduls zu schreiben, um die Vorwärtsgeschwindigkeit des Autos zu steuern.

1. Verwenden Sie die `for()` Anweisung, um `speed` in Schritten von 5 zu geben, wobei Werte von 0 bis 255 geschrieben werden, sodass Sie die Änderung 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 Funktion `moveForward()`.

Im Gegensatz zu [2. Bewegung durch Code](#), welcher direkt hohe/niedrige Pegel an die Eingangspins des L9110-Moduls gibt, übergeben wir hier einen Parameter `speed` an die Stellen, an denen 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 Block von Anweisungen, der von geschweiften Klammern eingeschlossen ist, zu wiederholen. Ein Inkrementzähler wird normalerweise verwendet, um die Schleife zu inkrementieren und zu beenden.

```
for (Initialisierung; Bedingung; Inkrement) {
    // Anweisung(en);
}
```

- **Initialisierung:** erfolgt zuerst und genau einmal.
- **Bedingung:** Bei jedem Durchlauf der Schleife wird die Bedingung geprüft; wenn sie wahr ist, wird der Anweisungsblock und das Inkrement ausgeführt, dann wird die Bedingung erneut geprüft. Wenn die Bedingung falsch wird, endet die Schleife.
- **Inkrement:** wird bei jedem Schleifendurchlauf ausgeführt, wenn die Bedingung wahr ist.

6.5 4. Die Linie verfolgen

Das Auto ist mit einem Linienverfolgungsmodul ausgestattet, mit dem es einer schwarzen Linie folgen kann.

Wenn das Linienverfolgungsmodul die schwarze Linie erkennt, dreht sich der rechte Motor, während der linke Motor stehen bleibt. Dadurch bewegt sich das Auto einen Schritt nach vorne links. Während sich das Auto bewegt und das Modul von der Linie wegbewegt, dreht sich der linke Motor und der rechte Motor bleibt stehen. Das Auto bewegt sich einen Schritt nach rechts, um zur Linie zurückzukehren. Die beiden oben genannten Schritte wiederholen sich, sodass das Auto der schwarzen Linie folgen kann.

Bevor Sie mit dem Projekt beginnen, sollten Sie eine Kurvenkarte mit schwarzem Klebeband erstellen. Die empfohlene Linienbreite liegt zwischen 0,8-1,5 cm, und der Drehwinkel 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.

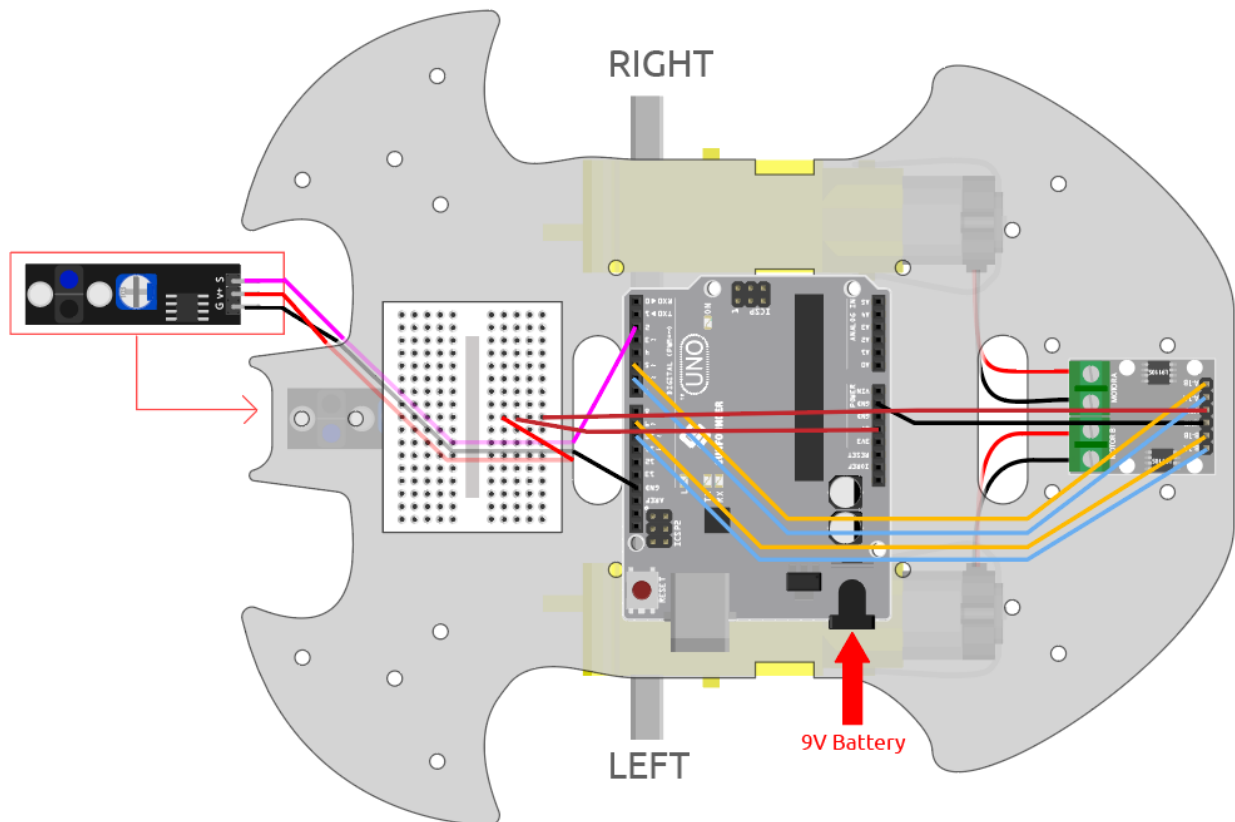
KOMPONENTENÜBERSICHT	KAUF LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Linienverfolgungsmodul</i>	

Verdrahtung

Dies ist ein digitales Linienverfolgungsmodul. Bei Erkennung einer schwarzen Linie gibt es den Wert 1 aus; bei Erkennung einer weißen Linie den Wert 0. Zusätzlich können Sie über das Potentiometer auf dem Modul seinen Erfassungsabstand einstellen.

Bauen Sie den Schaltkreis gemäß dem folgenden Diagramm auf.

Linienverfolgungsmodul	R3 Board
S	2
V+	5V
G	GND



Das Modul anpassen

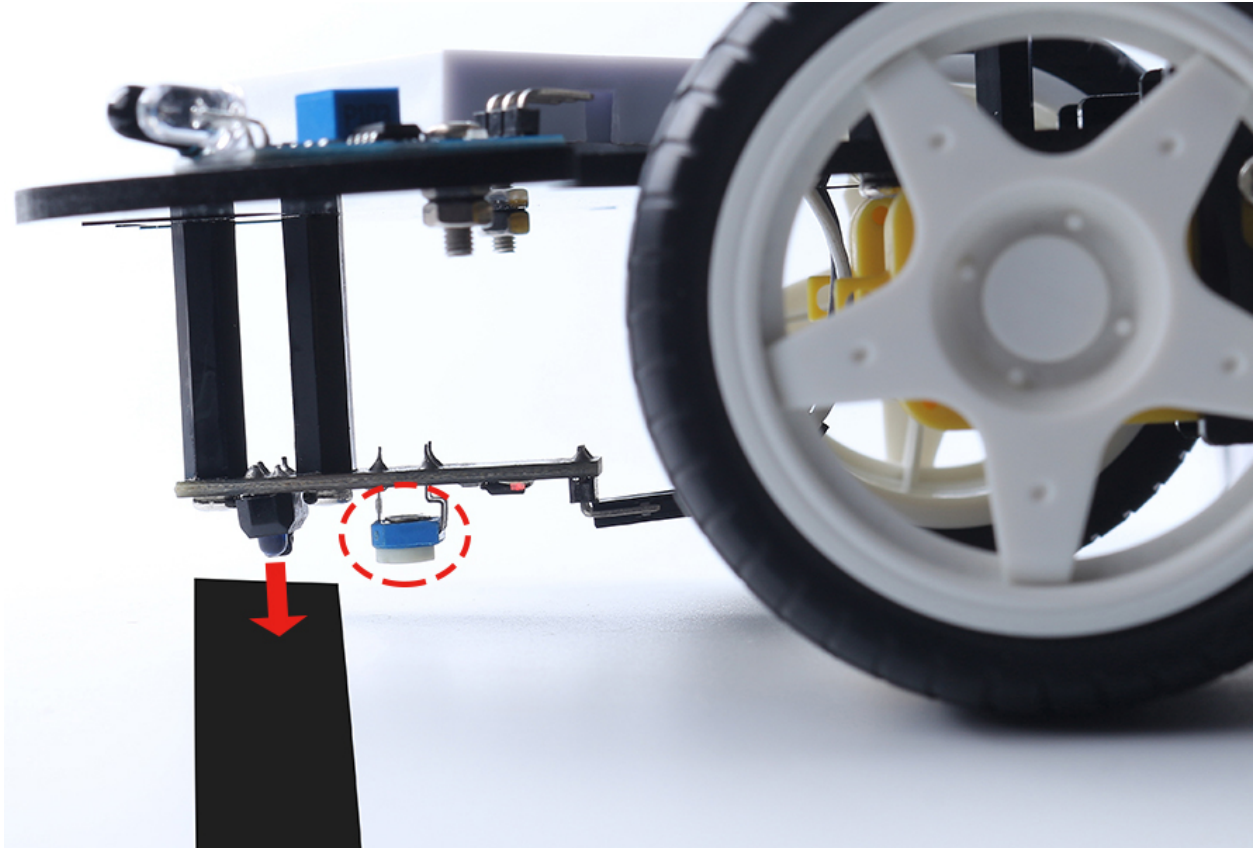
Vor Beginn des Projekts müssen Sie die Empfindlichkeit des Moduls anpassen.

Verkabeln Sie entsprechend dem obigen Diagramm und schalten Sie das R3-Board ein (entweder direkt über das USB-Kabel oder das 9V Batteriekнопfkabel), ohne den Code hochzuladen.

Kleben Sie ein schwarzes Elektroband 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 dies nicht der Fall ist, müssen Sie das Potentiometer am Modul anpassen, damit es den oben beschriebenen Effekt erzielt.



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 der Code auf das R3-Board hochgeladen wurde, richten Sie das Linienverfolgungsmodul unter dem Auto an der schwarzen Linie aus, und Sie werden sehen, wie das Auto der Linie folgt.

Wie funktioniert das?

In diesem Code werden die beiden Motoren entsprechend dem Wert des Linienverfolgungsmoduls mikrorotiert, sodass 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. Wenn es 1 ist, lassen Sie das Auto nach links vorfahren; andernfalls fahren Sie nach rechts vor. Sie können auch den seriellen Monitor öffnen, indem Sie auf das Lupensymbol in der oberen rechten Ecke klicken, um die Änderung des Wertes des Linienverfolgungsmoduls auf der schwarzen und weißen Linie vor dem Ausstecken des USB-Kabels zu sehen.

```

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()`.

Im Gegensatz zur Links-Rechts-Drehfunktion im Projekt 2. *Bewegung durch Code* sind hier nur kleine Links-Rechts-Drehungen erforderlich. Sie müssen daher nur den Wert von A_1A oder B_1B jedes Mal anpassen. Wenn Sie zum Beispiel nach vorne links fahren (`moveLeft()`), müssen Sie die Geschwindigkeit nur auf A_1A setzen und alle anderen auf 0, damit der rechte Motor im Uhrzeigersinn dreht und der linke Motor 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 am seriellen Port als menschenlesbaren ASCII-Text aus, gefolgt von einem Wagenrücklaufzeichen (ASCII 13 oder ,r‘) und einem Zeilenumbruchszeichen (ASCII 10 oder ,n‘).

- `if else`

Mit `if else` lässt sich der Codefluss besser steuern als mit der einfachen `if`-Anweisung, da mehrere Tests gruppiert werden können.

6.6 5. Mit dem Hindernisvermeidungs-Modul spielen

Zwei Infrarot-Hindernisvermeidungsmodule sind vorne am Auto montiert. Sie dienen dazu, nahegelegene Hindernisse zu erkennen.

In diesem Projekt darf das Auto frei nach vorne fahren. Bei einem Hindernis kann es dieses umgehen und in anderen 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 SET	LINK
3 in 1 Starter-Set	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Hindernisvermeidungsmodul</i>	

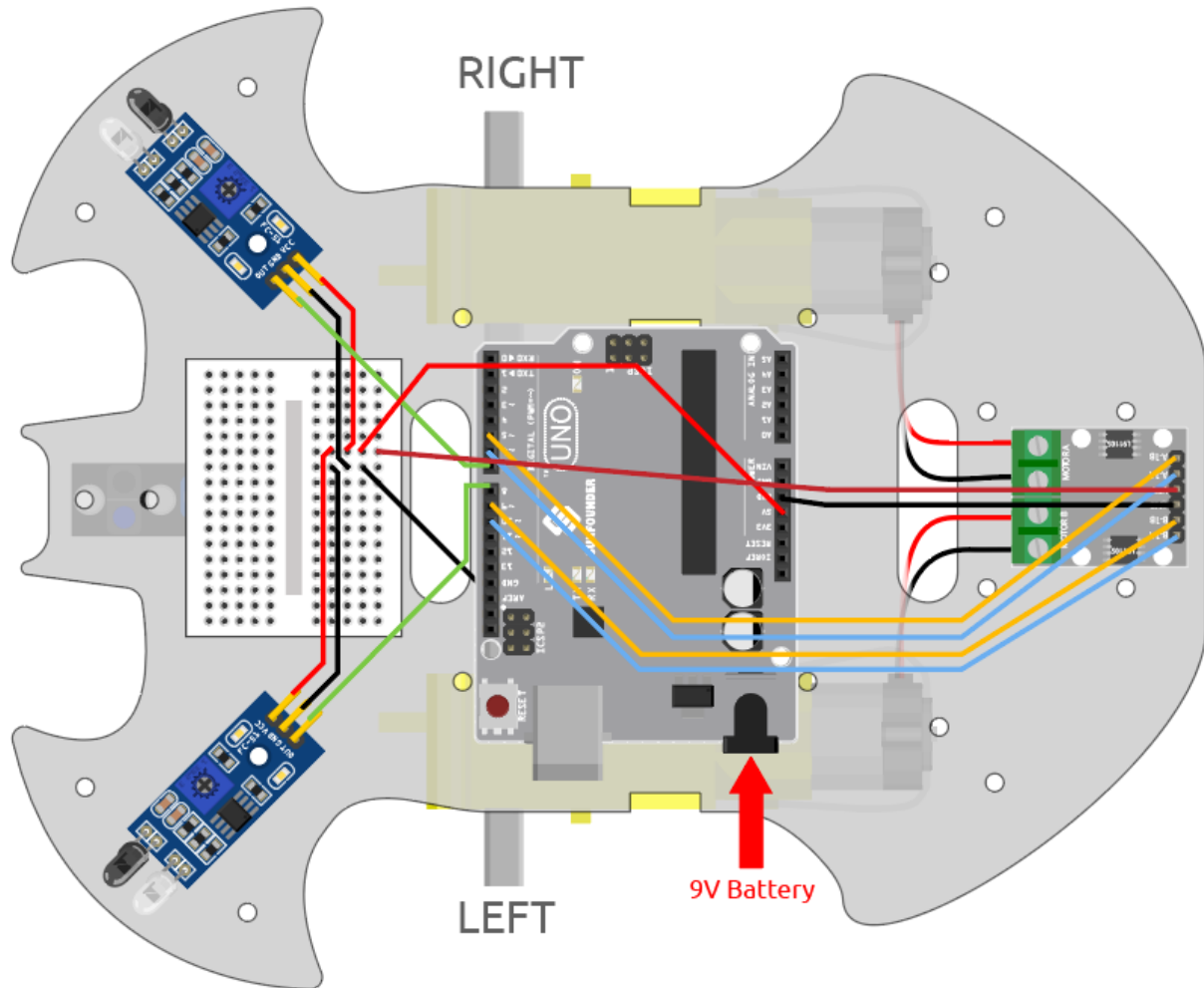
Verdrahtung

Das Hindernisvermeidungs-Modul ist ein einstellbarer Infrarot-Näherungssensor. Sein Ausgang ist normalerweise hoch und wird niedrig, wenn ein Hindernis erkannt wird.

Bauen Sie die Schaltung gemäß dem untenstehenden Diagramm.

Linkes IR-Modul	R3-Platine
OUT	8
GND	GND
VCC	5V

Rechtes IR-Modul	R3-Platine
OUT	7
GND	GND
VCC	5V



Modul justieren

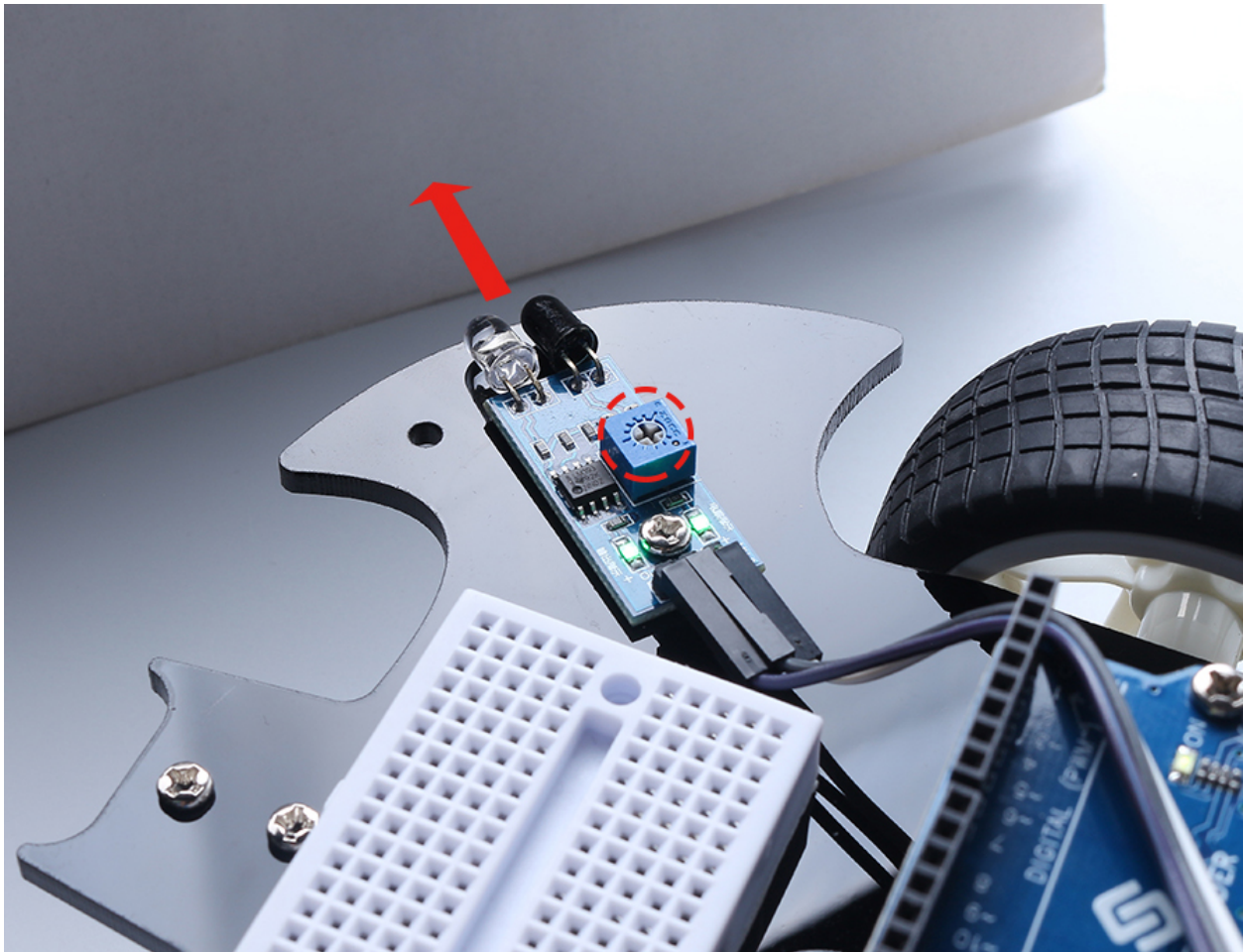
Bevor Sie mit dem Projekt beginnen, müssen Sie den Erkennungsabstand des Moduls einstellen.

Nach der oben gezeigten Verdrahtung, schalten Sie die R3-Platine ein (entweder durch direktes Anschließen des USB-Kabels oder durch Anschließen des 9V-Batteriekabels), ohne den Code hochzuladen.

Platzieren Sie ein Notizbuch oder ein anderes flaches Objekt etwa 5 cm vor der IR-Hindernisvermeidung.

Verwenden Sie dann einen Schraubendreher, um das Potentiometer auf dem Modul zu drehen, bis die Signalanzeige auf dem Modul gerade aufleuchtet, um den maximalen Erkennungsabstand von 5 cm einzustellen.

Verfahren Sie auf die gleiche Weise, um ein weiteres Infrarot-Modul einzustellen.



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 fährt vorwärts, 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 dem Auto die entsprechende Aktion ausführen zu lassen.

1. Fügen Sie die Pin-Definition für die 2 Hindernisvermeidungsmodule hinzu, hier sind sie auf INPUT gesetzt.

```
...
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.
- Andernfalls fährt das Auto weiter vorwärts.

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 true, wenn beide Operanden wahr sind.
- !: Logisches NICHT ergibt true, wenn der Operand falsch ist und umgekehrt.

6.7 6. Spielen mit dem Ultraschallmodul

Im Projekt 5. *Mit dem Hindernisvermeidungs-Modul spielen* werden die 2 Infrarot-Hindernisvermeidungsmodule zur Hindernisvermeidung verwendet, aber die Erkennungsdistanz des IR-Hindernisvermeidungsmoduls ist kurz, wodurch das Auto möglicherweise zu spät kommt, um den Hindernissen auszuweichen.

In diesem Projekt verwenden wir ein Ultraschallmodul für eine Langstreckenerkennung, damit das Auto Hindernisse aus einer größeren Entfernung erkennen und eine Entscheidung treffen 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 sie auch einzeln über die untenstehenden Links kaufen.

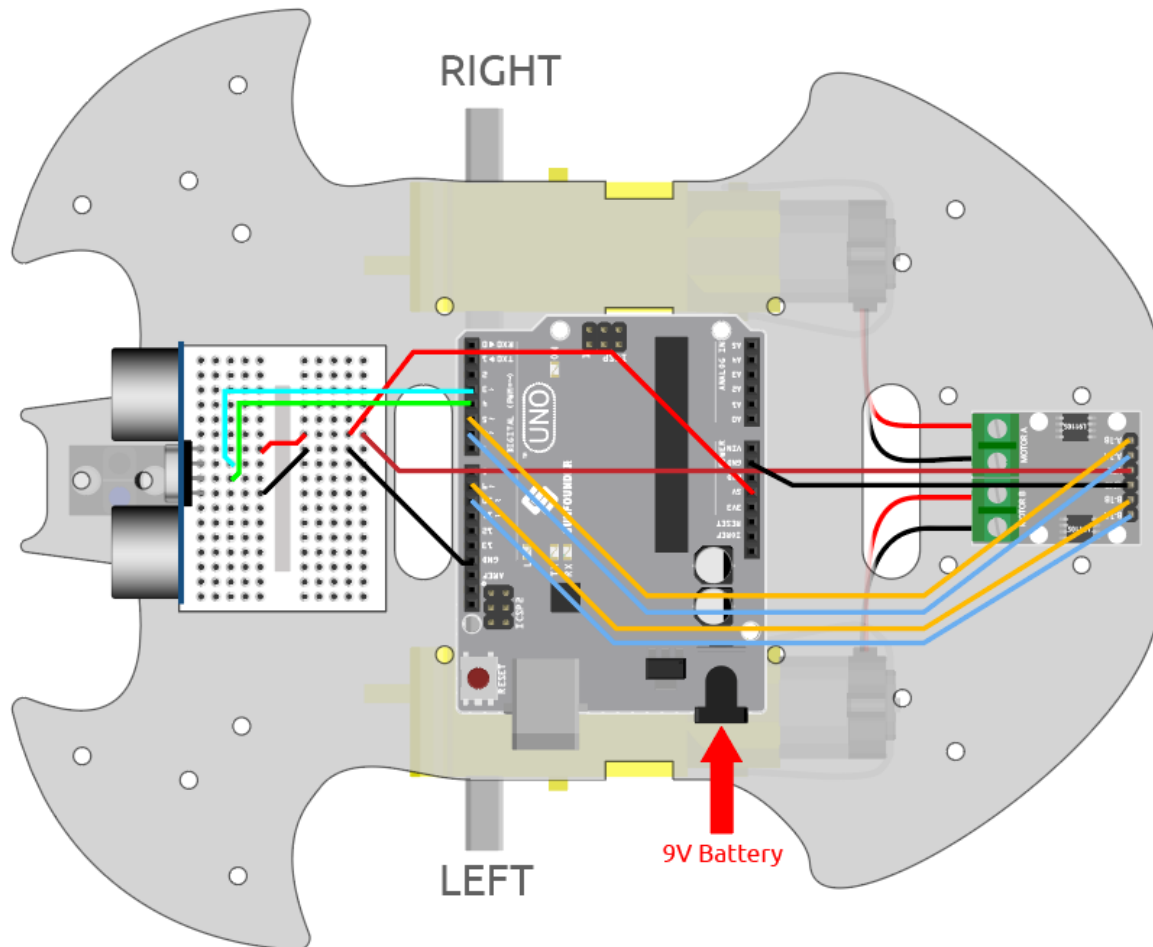
KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Ultraschall-Modul</i>	

Verdrahtung

Ein Ultraschallsensormodul ist ein Instrument, das die Entfernung zu einem Objekt mit Ultraschallwellen misst. Es hat zwei Sonden. Eine sendet Ultraschallwellen und die andere empfängt die Wellen und wandelt die Sende- und Empfangszeit in eine Entfernung um, wodurch die Entfernung zwischen dem Gerät und einem Hindernis erfasst wird.

Bauen Sie die Schaltung entsprechend dem folgenden Diagramm auf.

Ultraschallmodul	R3 Board
Vcc	5V
Trig	3
Echo	4
Gnd	GND



Code

Bemerkung:

- Öffnen Sie die Datei 6.ultrasonic_module.ino im 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 die Wand. Wenn der Abstand zu groß ist, fährt es vorwärts; wenn es zu nah ist, fährt es rückwärts; wenn es sich in sicherer Entfernung befindet, hält es an.

Wie funktioniert das?

Dieses Projekt basiert auf der vom Ultraschallmodul gelesenen Entfernung, um das Auto entsprechend zu bewegen.

1. Fügen Sie die Pin-Definition für das Ultraschallmodul hinzu, trigPin wird verwendet, um Ultraschallwellen zu senden, setzen Sie es also auf OUTPUT; echoPin wird auf INPUT gesetzt, um Ultraschallwellen zu empfangen.

```
...
const int trigPin = 3;
const int echoPin = 4;
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

void setup() {
  ...

  //ultrasonic
  pinMode(echoPin, INPUT);
  pinMode(trigPin, OUTPUT);
}

```

2. Zuerst lesen Sie den Abstandswert, der vom Ultraschallmodul erhalten wurde. Wenn der Abstand größer als 25 ist, lassen Sie das Auto vorwärts fahren; wenn der Abstand 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 Rechteckwellensignal, und der Empfänger empfängt ein High-Level-Signal, wenn sich ein Hindernis im Bereich befindet. Verwenden Sie die Funktion pulseIn(), um die Zeit vom Senden bis zum Empfangen aufzuzeichnen, 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; //Entspricht (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: Typ des zu lesenden Pulses: entweder HIGH oder LOW. Zulässige Datentypen: int.

Liest einen Impuls (entweder HIGH oder LOW) an einem Pin. Wenn zum Beispiel der Wert HIGH ist, wartet pulseIn() darauf, dass der Pin von LOW auf HIGH wechselt, beginnt mit der Zeitmessung und wartet dann darauf, dass der Pin LOW wird und stoppt die Zeitmessung.

6.8 7. Folge Deiner Hand

Betrachten Sie dieses Auto als Ihr Haustier hier. Wenn Sie ihm zuwinken, kommt es zu Ihnen gelaufen.

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 IM SET	LINK
3-in-1 Starter Kit	380+	

Sie können diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Ultraschall-Modul</i>	
<i>Hindernisvermeidungsmodul</i>	

Verkabelung

Schließen Sie das Ultraschallmodul und die beiden IR-Hinderniserkennungs-Module gleichzeitig an.

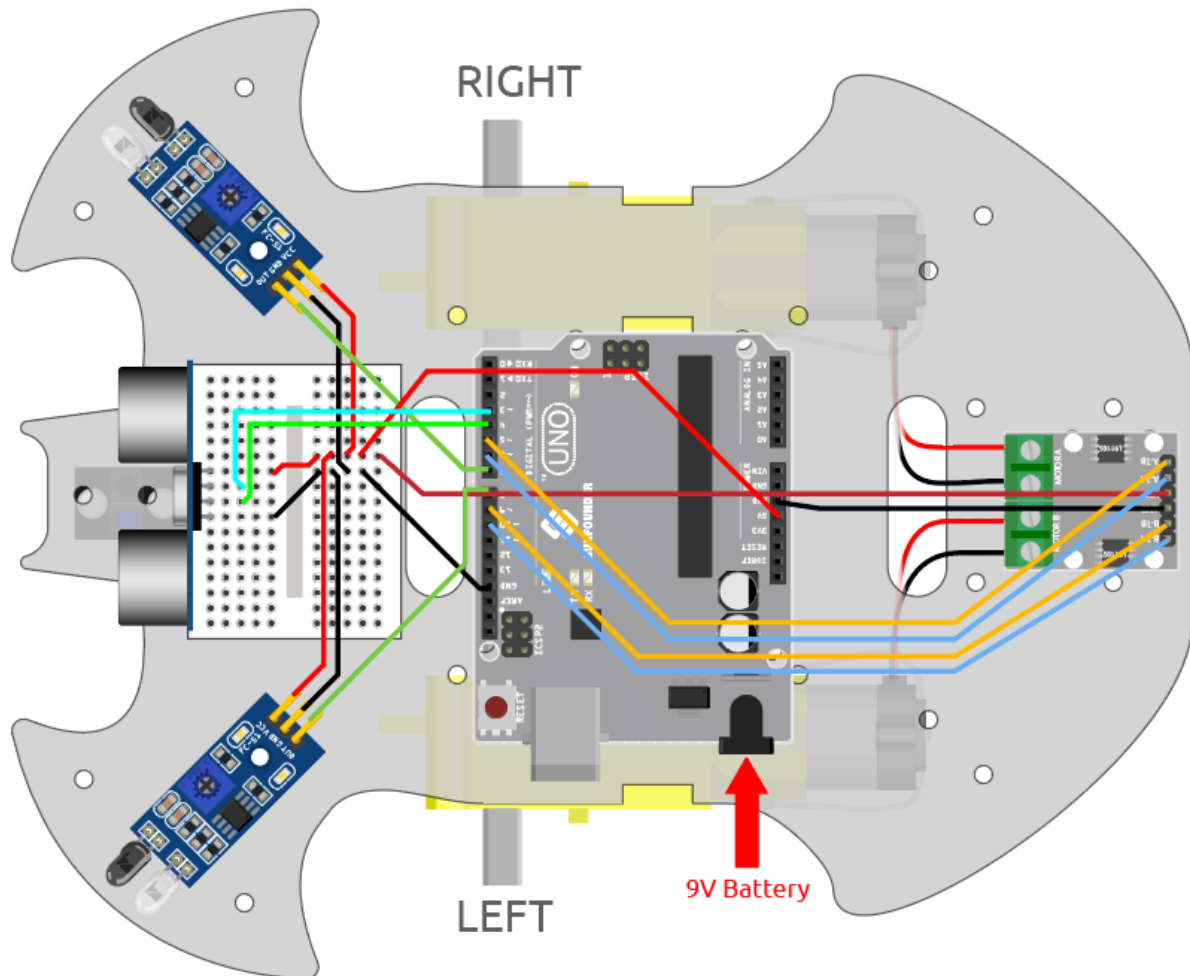
Verbinden Sie das Ultraschallmodul wie folgt mit dem R3-Board.

Ultraschallmodul	R3 Board
Vcc	5V
Trig	3
Echo	4
Gnd	GND

Die Verkabelung der beiden IR-Hinderniserkennungs-Module zum R3-Board ist wie folgt:

Linkes IR-Modul	R3 Board
OUT	8
GND	GND
VCC	5V

Rechtes IR-Modul	R3 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 nah an das IR-Hindernismodul auf beiden Seiten halten, wird es auch in die entsprechende Richtung drehen.

Wie funktioniert es?

Dieses Projekt ist eine Kombination aus den vorherigen zwei Projekten *6. Spielen mit dem Ultraschallmodul* und *5. Mit dem Hindernisvermeidungs-Modul spielen*, jedoch mit einem anderen implementierten Effekt. Bei den vorherigen 2 Projekten wurde ein Hindernis rückwärts erkannt, während hier erkannt wird, dass Ihre Hand der Vorwärts- oder Drehrichtung folgt. Der Ablauf dieses Projekts ist wie folgt:

- Lesen Sie die vom Ultraschallmodul erfasste Entfernung und den Wert beider Infrarotmodule.
- Wenn die Entfernung 5~10cm beträgt, lassen Sie das Auto mit Ihrer Hand fahren.

- Wenn das linke IR-Modul Ihre Hand erkennt, nach links abbiegen.
- Wenn das rechte IR-Modul Ihre Hand erkennt, nach rechts abbiegen.
- Wenn weder das Infrarotmodul noch das Ultraschallmodul Ihre Hand erkennen, lassen Sie 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();  
    }  
}
```

6.9 8. Autonomes Fahren

Dieses Projekt kombiniert zwei Projekte: *6. Spielen mit dem Ultraschallmodul* und *5. Mit dem Hindernisvermeidungs-Modul spielen*. Die 2 Infrarot-Hindernisvermeidungsmodule dienen zur kurzfristigen oder Rand-Erkennung, während die Ultraschallmodule zur Fernerkennung verwendet werden, um sicherzustellen, dass das Auto während des freien Fahrens keinem Hindernis begegnet.

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

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

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

Alternativ können Sie diese auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</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.

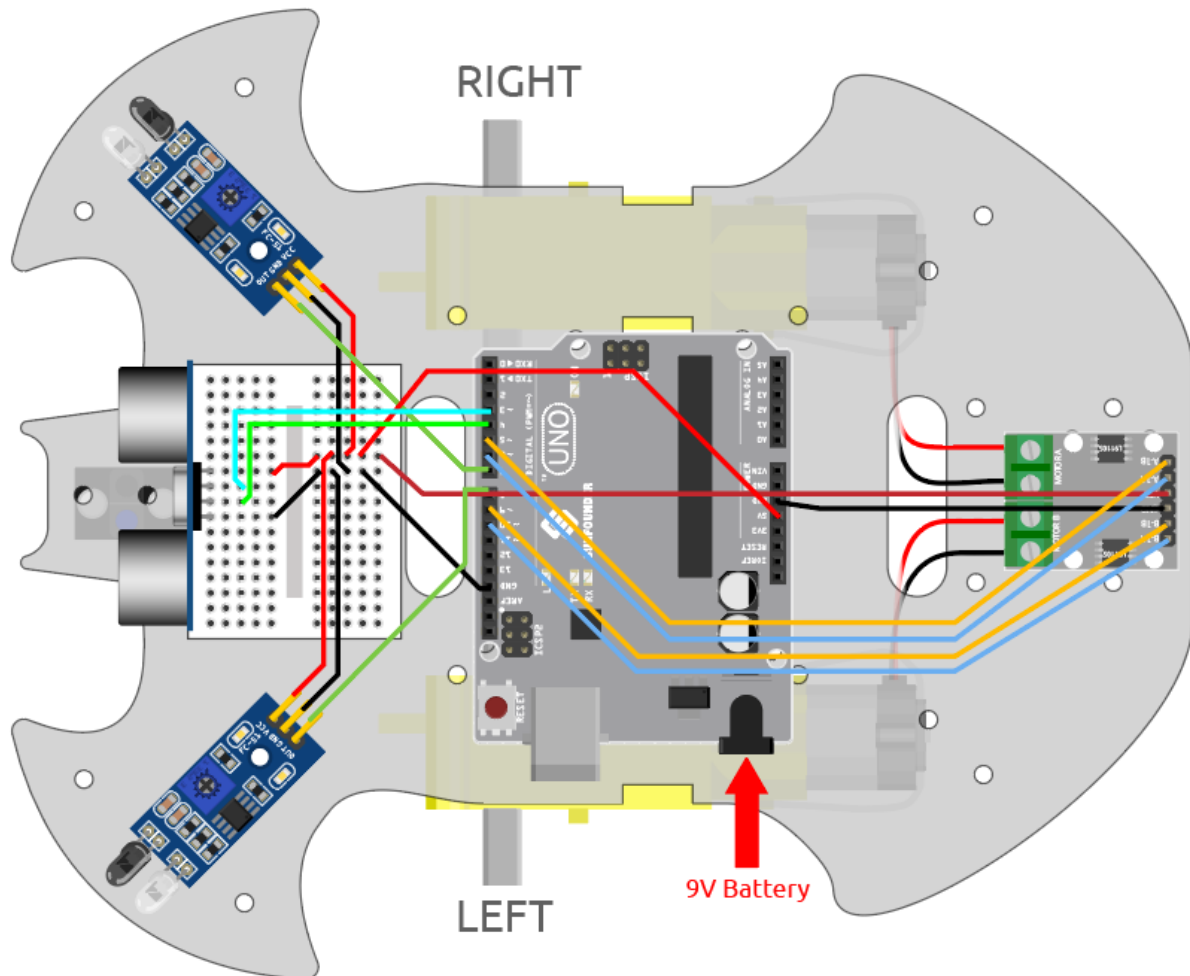
Verbinden Sie das Ultraschallmodul wie folgt mit dem R3-Board.

Ultraschallmodul	R3 Board
Vcc	5V
Trig	3
Echo	4
Gnd	GND

Die Verdrahtung der beiden IR-Hindernisvermeidungsmodule zum R3-Board ist wie folgt:

Linkes IR-Modul	R3 Board
OUT	8
GND	GND
VCC	5V

Rechtes IR-Modul	R3 Board
OUT	7
GND	GND
VCC	5V



Programmierung

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**.
- Alternativ können Sie den Code über den [Arduino Web Editor](#) hochladen.

Sobald der Code erfolgreich hochgeladen wurde, beginnt das Auto mit dem freien Fahren. Wenn das IR-Hindernisvermeidungsmodul auf beiden Seiten ein Hindernis erkennt, wird es in die entgegengesetzte Richtung für eine Notausweichbewegung fahren; wenn direkt vor dem Auto ein Hindernis in einem Abstand von 2~10cm ist, wird es rückwärts nach links fahren, seine Richtung korrigieren und dann vorwärts fahren.

Wie funktioniert es?

Der Arbeitsablauf dieses Projekts ist wie folgt.

- Lese zuerst den Wert des linken und rechten IR-Hindernisvermeidungsmoduls.
- Wenn das linke IR-Modul 0 ist (Hindernis erkannt) und das rechte IR-Modul 1 ist, lasse das Auto nach links zurückfahren.

- Wenn das rechte IR-Modul 0 ist (Hindernis erkannt), lasse das Auto nach rechts zurückfahren.
- Wenn beide IR-Module gleichzeitig ein Hindernis erkennen, wird das Auto rückwärts fahren.
- Ansonsten lies den vom Ultraschallmodul erkannten Abstand.
- Wenn der Abstand größer als 50 cm ist, lasse das Auto vorwärts fahren.
- Wenn der Abstand zwischen 2-10cm liegt, lasse das Auto zurückfahren, bevor es sich dreht.
- Wenn der Abstand zwischen 10-50cm liegt, lasse 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);
        }
    }
}
```

6.10 9. Fernbedienung

Dieses Kit wird mit einem IR-Empfänger geliefert, mit dem Sie ein Auto mithilfe einer IR-Fernbedienung steuern können.

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 separat über die untenstehenden Links kaufen.

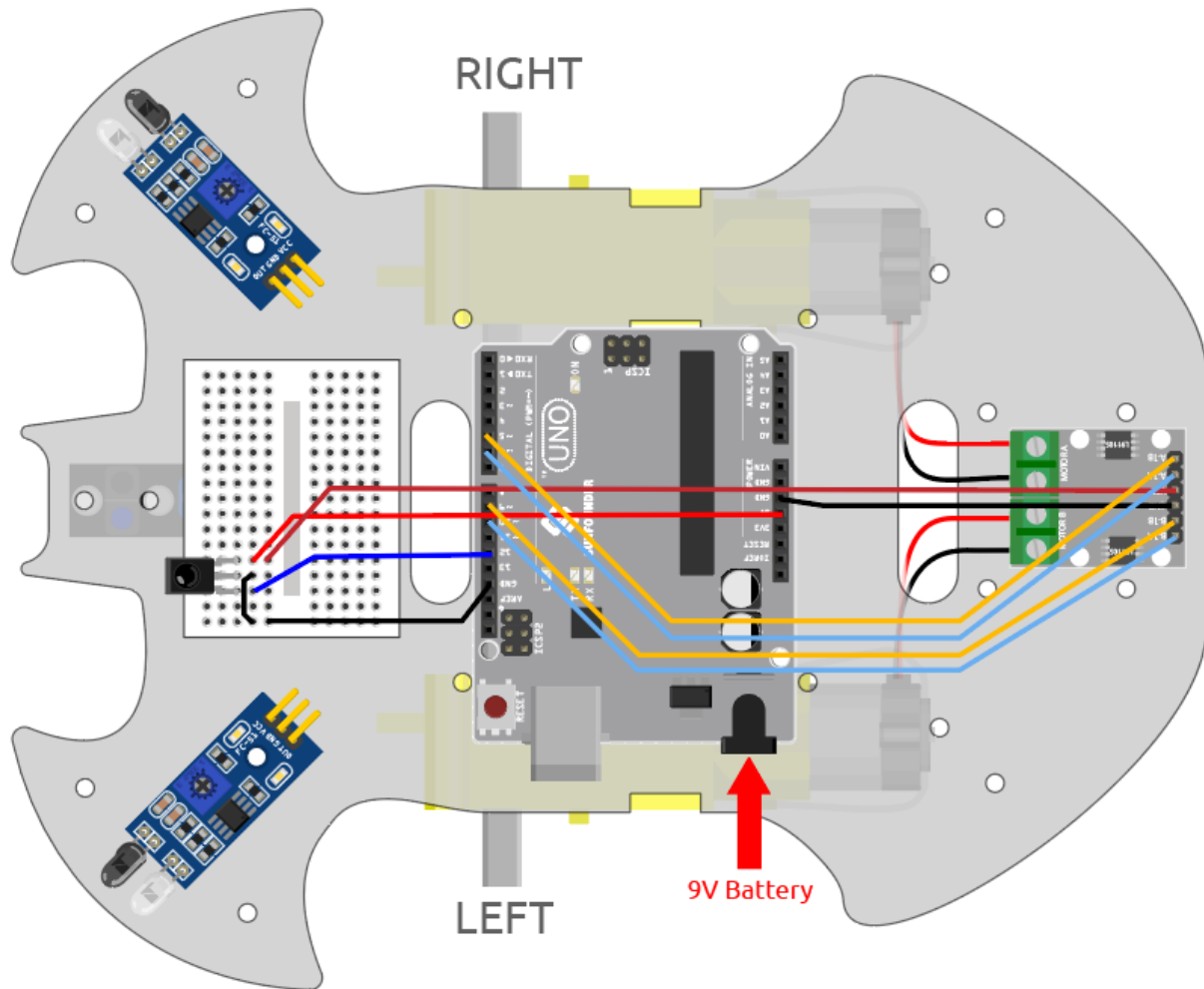
KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>LED</i>	
<i>IR-Empfänger</i>	-

Verdrahtung

Bauen Sie jetzt den Schaltkreis gemäß dem untenstehenden Diagramm.

IR-Empfänger	R3 Platine
OUT	12
GND	GND
VCC	5V

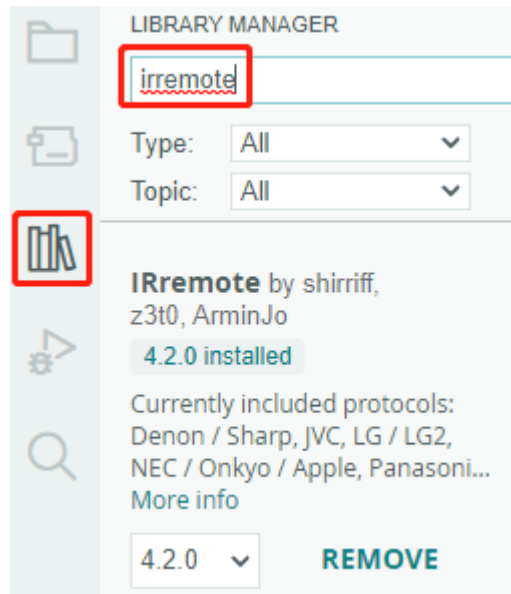
LED	R3 Platine
Anode (Der längere Pin)	13
Kathode	GND



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. Sie können sie über den **Library Manager** installieren.



Nachdem der Code erfolgreich hochgeladen wurde, drücken Sie die Taste auf der Fernbedienung. Die LED blinkt einmal, um anzuzeigen, dass das Signal empfangen wurde, und das Auto bewegt sich entsprechend der gedrückten Taste. Sie können die folgenden Tasten verwenden, um das Auto zu steuern.

- +: Beschleunigen
- -: Verlangsamen
- 1: Vorwärts nach links
- 2: Vorwärts
- 3: Vorwärts nach rechts
- 4: Links abbiegen
- 6: Rechts abbiegen
- 7: Rückwärts nach links
- 8: Rückwärts
- 9: Rückwärts nach rechts

Wie funktioniert es?

Das Ziel dieses Projekts ist es, das Auto durch Auslesen des Tastenwerts der IR-Fernbedienung zu bewegen. Zusätzlich wird die LED an Pin 13 blinken, um den erfolgreichen Empfang des IR-Signals anzuzeigen.

1. Importieren Sie die Bibliothek IRremote. Sie können diese aus dem **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. Initialisiert 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
    }
}

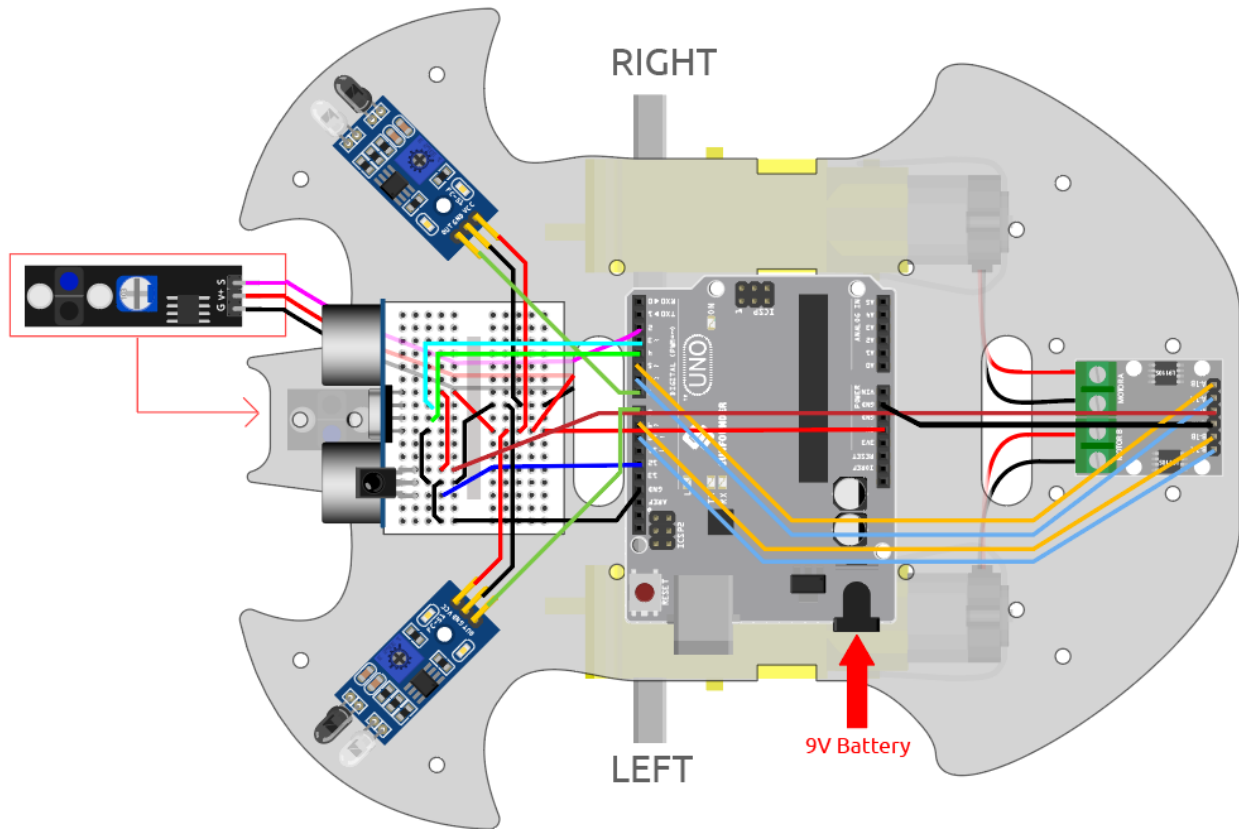
```

- Überprüft, ob ein IR-Signal empfangen und erfolgreich decodiert wurde.
- Decodiert den IR-Befehl und speichert ihn in `key` mit einer benutzerdefinierten Funktion `decodeKeyValue()`.
- Überprüft, ob der decodierte Wert kein Fehler ist.
- Gibt den decodierten IR-Wert auf dem seriellen Monitor aus.
- Setzt den IR-Signalempfang für das nächste Signal fort.

6.11 10. Start mit einem Tastendruck

In diesem Projekt haben wir die vorherigen Projekte - Linienverfolgung, Nachfahren, Hindernisvermeidung, autonomes Fahren usw. zusammengeführt. Sie können über Tasten der Fernbedienung umgeschaltet werden, sodass Sie das Auto starten und alle Funktionen gleichzeitig erleben können.

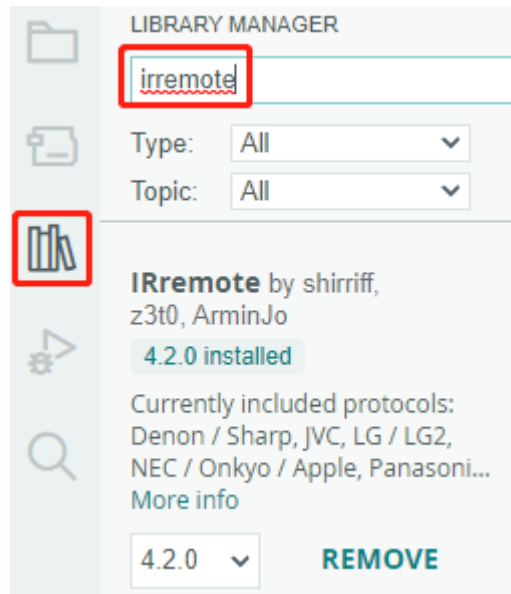
Verdrahtung



Code

Bemerkung:

- Öffnen Sie die Datei `10.one_touch_start.ino` im Pfad `3in1-kit\car_project\10.one_touch_start`.
- Oder kopieren Sie diesen Code in die **Arduino IDE**.
- Hier wird die Bibliothek `IRremote` verwendet. Sie können sie über den **Library Manager** installieren.



Nachdem der Code erfolgreich hochgeladen wurde, blinkt die LED 3 Mal schnell für jedes vom IR-Empfänger von der Fernbedienung empfangene Signal. Sie können die folgenden Tasten drücken, um den Wagen zu bedienen.

- **+**: Beschleunigen
- **-**: Verlangsamen
- **1**: Nach links vorne bewegen
- **2**: Vorwärts
- **3**: Nach rechts bewegen
- **4**: Links abbiegen
- **6**: Rechts abbiegen
- **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**: Ihrer Hand folgen
- **0**: Stoppen

6.12 11. Geschwindigkeitskalibrierung

Beim Vorwärtsbewegen des Autos stellen Sie möglicherweise fest, dass das Auto nicht geradeaus fährt. Das liegt daran, dass die beiden Motoren möglicherweise ab Werk nicht die gleiche Geschwindigkeit haben. Wir können jedoch einen Ausgleich für die beiden Motoren schreiben, damit sich ihre Drehgeschwindigkeiten angleichen.

In diesem Projekt werden wir lernen, den Ausgleich in **EEPROM** zu speichern. Der Sinn dahinter ist, dass nach jeder Kalibrierung alle Projekte den Ausgleichswert direkt aus dem EEPROM abrufen können, damit das Auto reibungslos geradeaus fährt.

Verdrahtung

Dieses Projekt hat die gleiche Verdrahtung wie *2. Bewegung 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 lassen Sie es vorwärts fahren, um zu sehen, zu welcher Seite es verschoben ist.
 - Wenn das Auto nach links vorne fährt, bedeutet das, dass die Geschwindigkeit des rechten Motors zu hoch ist und reduziert werden muss.

```
EEPROM.write(1, 100) // 1 bedeutet der rechte Motor, 100 bedeutet 100%
↪ Geschwindigkeit, kann je nach tatsächlicher Situation auf 90, 95 usw.
↪ gesetzt werden.
```

- Wenn das Auto nach rechts fährt, bedeutet das, dass die Geschwindigkeit des linken Motors zu schnell ist und reduziert werden muss.

```
EEPROM.write(0, 100) // 0 bedeutet der linke Motor, 100 bedeutet die
↪ Geschwindigkeit beträgt 100%, kann je nach tatsächlicher Situation auf 90,
↪ 95 usw. gesetzt werden.
```

3. Nach der Code-Änderung laden Sie den Code auf das R3-Board hoch, um die Wirkung zu sehen. Wiederholen Sie die obigen Schritte, bis das Auto fast gerade fährt.
4. Dieser Ausgleich wird in **EEPROM** aufgezeichnet. Sie müssen diesen Ausgleich nur abrufen, wenn Sie ihn in anderen Projekten verwenden, nehmen Sie *5. Mit dem Hindernisvermeidungs-Modul spielen* als Beispiel.

```
#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);
```

(Fortsetzung auf der nächsten Seite)

(Fortsetzung der vorherigen Seite)

```

//motor
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);
}

```

(Fortsetzung auf der nächsten Seite)

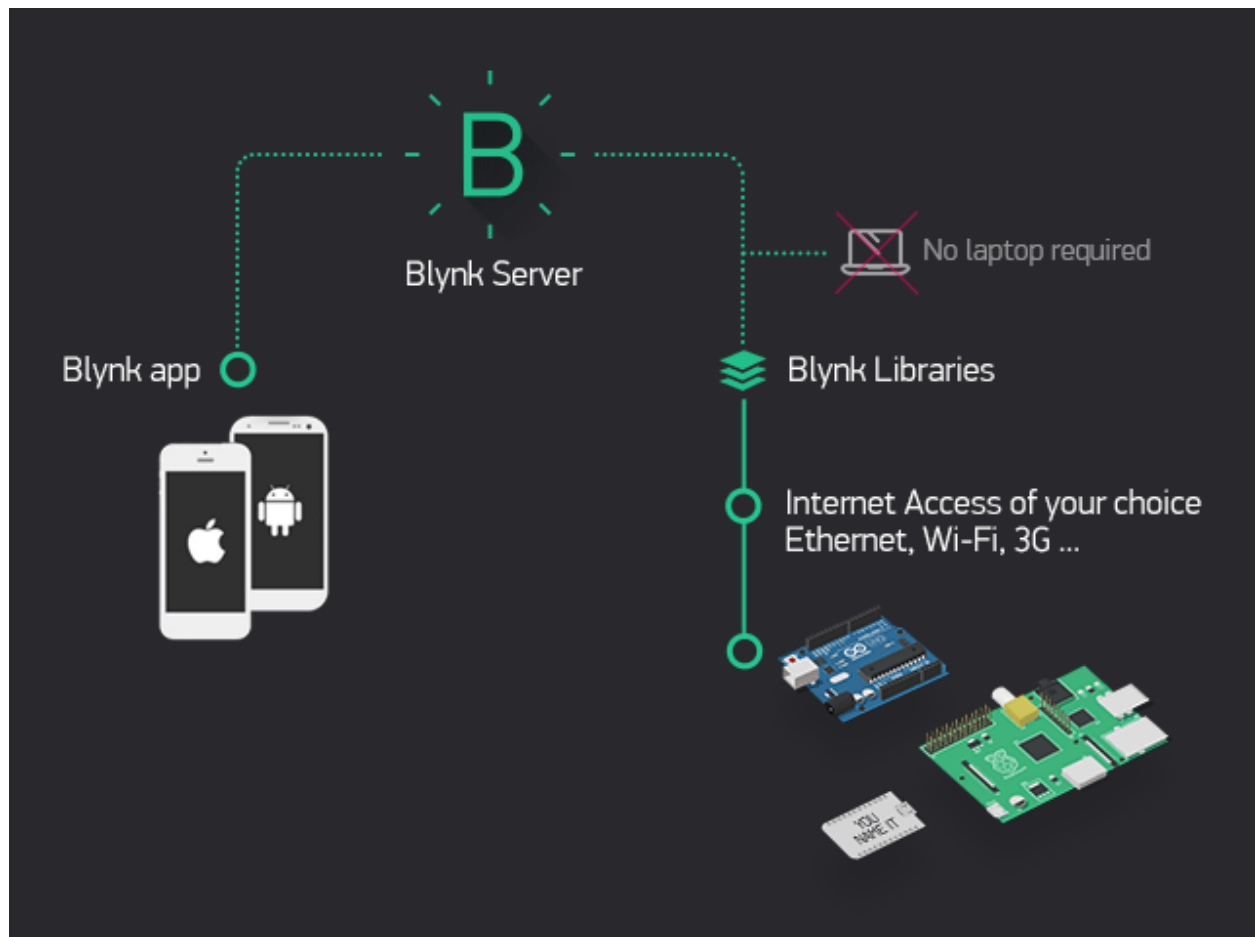
(Fortsetzung der vorherigen Seite)

```
void backRight(int speed) {  
    analogWrite(A_1B, 0);  
    analogWrite(A_1A, 0);  
    analogWrite(B_1B, 0);  
    analogWrite(B_1A, speed);  
}
```

Dieses Set enthält das ESP8266 Wifi-Modul, mit dem Arduino eine Verbindung zum Internet für IoT-Experimente herstellen kann.

Hier zeigen wir Ihnen, wie Sie Arduino mithilfe des ESP8266 Wifi-Moduls eine Verbindung zur [BLYNK](#)-Plattform herstellen können, um spannende IoT-Projekte zu realisieren. Außerdem können Sie die Blynk-App auf Ihrem Handy verwenden, um das Smart Car zu steuern.

Blynk ist eine umfassende Softwarelösung, die erforderlich ist, um verbundene elektronische Geräte in jeder Größenordnung zu prototypisieren, zu implementieren und aus der Ferne zu verwalten: von persönlichen IoT-Projekten bis hin zu Millionen von kommerziellen verbundenen Produkten. Mit Blynk kann jeder seine Hardware mit der Cloud verbinden und ohne Code iOS-, Android- und Web-Anwendungen erstellen, um Echtzeit- und historische Daten von Geräten zu analysieren, sie von überall auf der Welt aus fernzusteuern, wichtige Benachrichtigungen zu erhalten und vieles mehr...



7.1 1. Erste Schritte mit Blynk

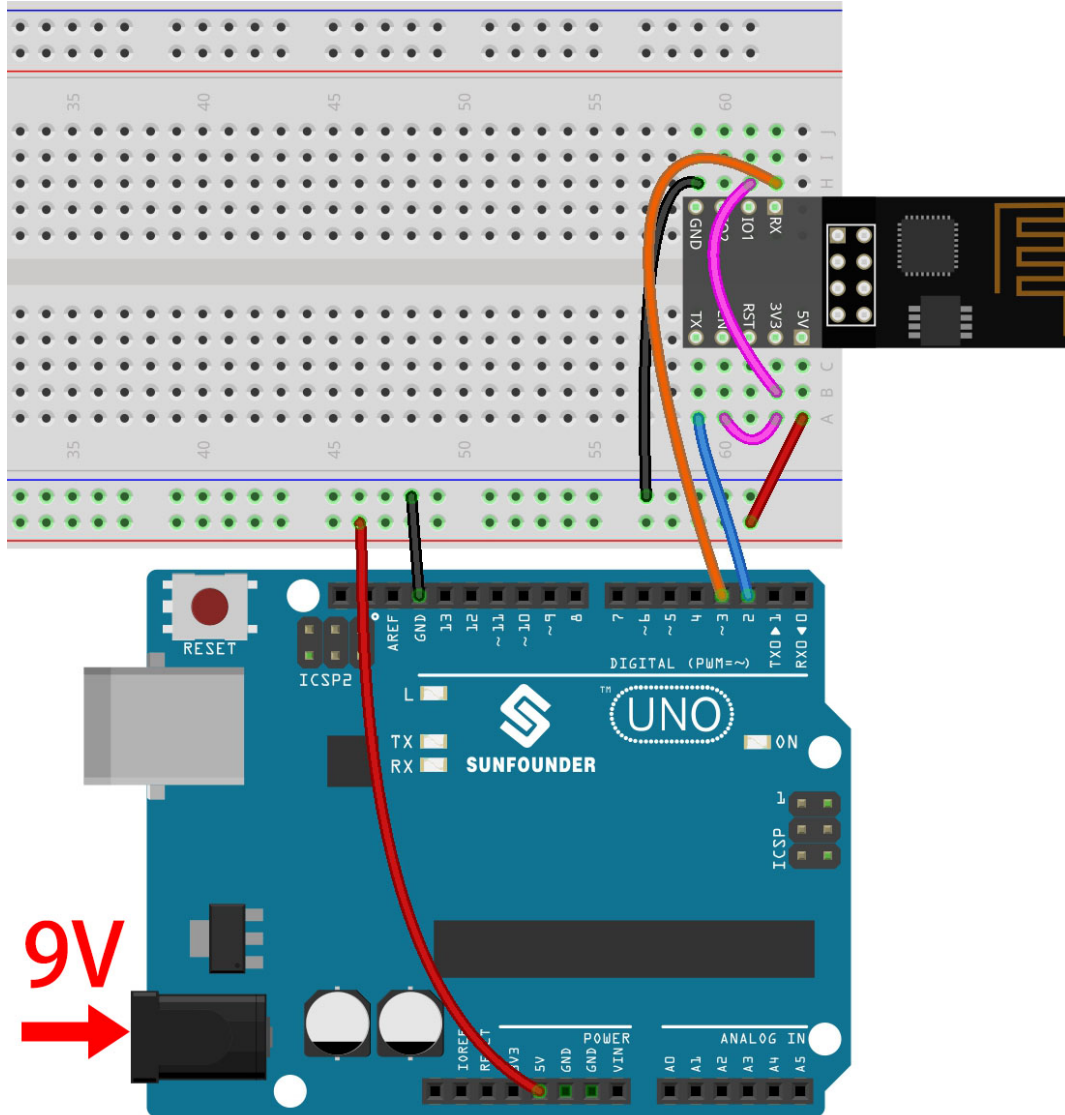
Um das R3-Board mit Blynk kommunizieren zu lassen, sind einige Konfigurationen erforderlich, wenn Sie Blynk zum ersten Mal verwenden.

Befolgen Sie die unten aufgeführten Schritte und beachten Sie, dass Sie sie in der angegebenen Reihenfolge durchführen und keine Kapitel überspringen dürfen.

7.1.1 1.1 Konfiguration des ESP8266

Das ESP8266-Modul, das im Kit enthalten ist, ist bereits mit der AT-Firmware vorgebrannt. Sie müssen jedoch seine Konfiguration ändern, indem Sie den untenstehenden Schritten folgen.

1. Bauen Sie den Schaltkreis.



2. Öffnen Sie die Datei 1.set_software_serial.ino im 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

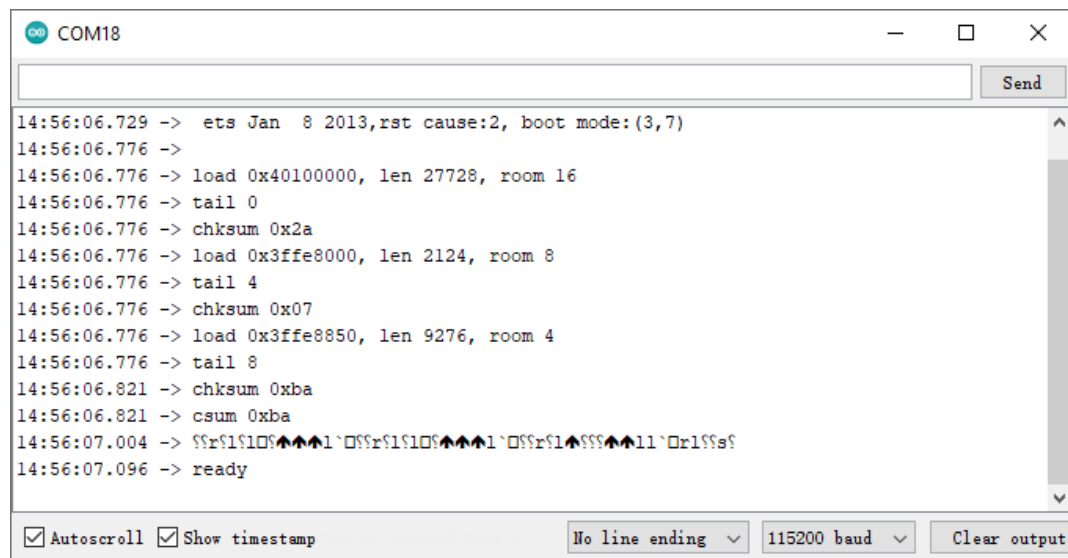
void setup() {
  // Setzen Sie hier Ihren Setup-Code ein, der einmal ausgeführt wird:
  Serial.begin(115200);
  espSerial.begin(115200);
}
```

(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 Lupen-Symbol (Serial Monitor) in der oberen rechten Ecke und setzen Sie die Baudrate auf **115200**. (Es kann sein, dass Sie einige Informationen wie ich erhalten, 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 lautet, müssen Sie möglicherweise die Firmware erneut brennen. Bitte schauen Sie unter [Wie kann die Firmware für das ESP8266-Modul neu geflasht werden?](#) nach Details. Wenn Sie das Problem immer noch nicht lösen können, machen Sie bitte einen Screenshot vom Serial Monitor und senden Sie ihn an sevice@sunfounder.com. Wir werden Ihnen so schnell wie möglich helfen.

4. Klicken Sie auf **NEWLINE DROPDOWN BOX**, wählen Sie im Dropdown-Menü **both NL & CR** aus und geben Sie **AT** ein. Wenn OK zurückgegeben wird, bedeutet dies, dass der ESP8266 erfolgreich eine Verbindung zum R3-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 auf **Station und AP Koexistenz** geändert.

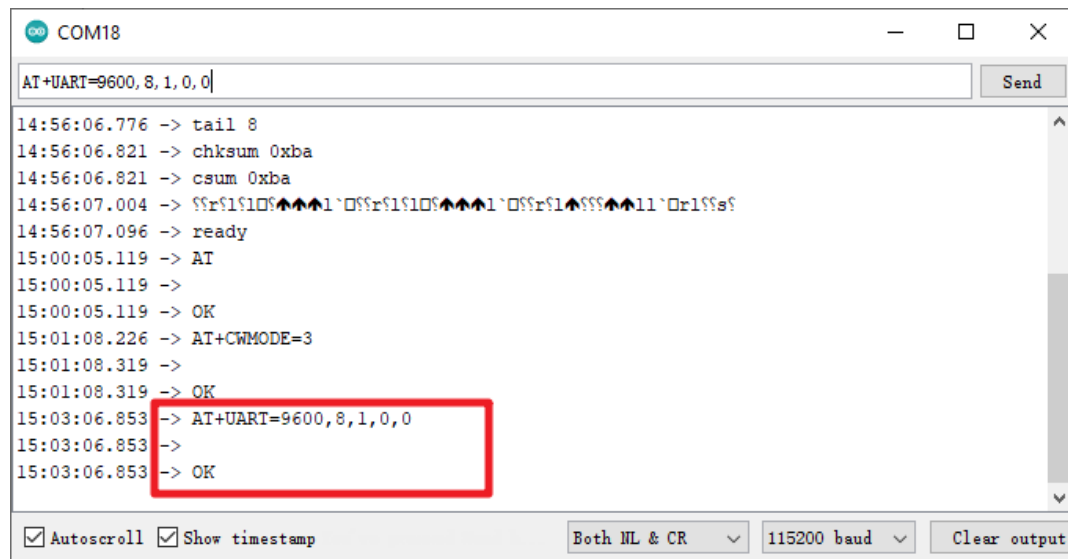
```

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-Seriennummer verwenden zu können, müssen Sie `AT+UART=9600,8,1,0,0` eingeben, um die Baudrate des ESP8266 auf 9600 zu ändern.

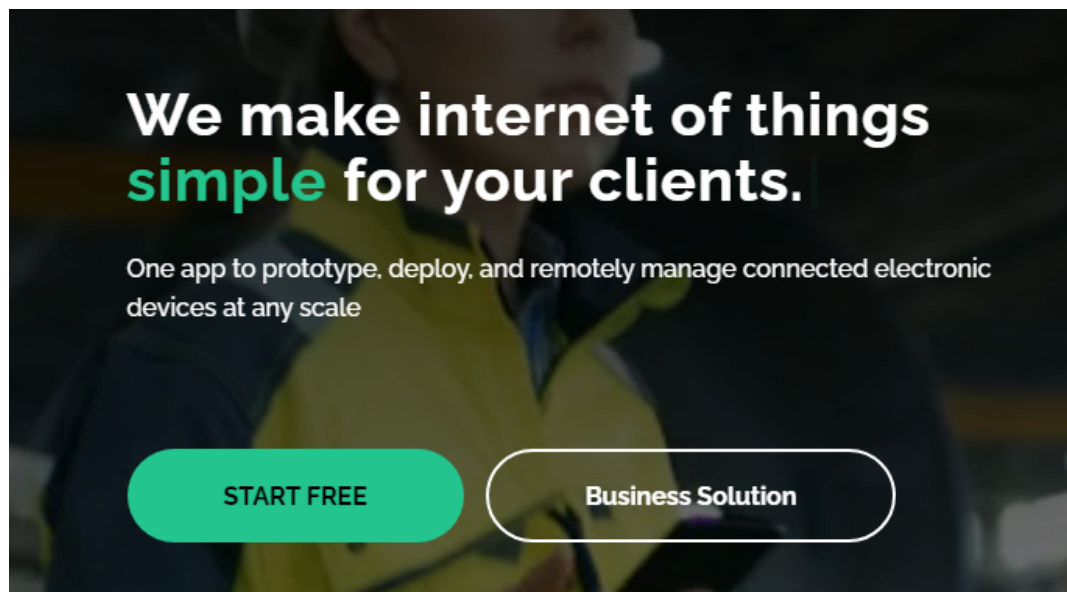


```
COM18
AT+UART=9600,8,1,0,0
14:56:06.776 -> tail 8
14:56:06.821 -> chksum 0xba
14:56:06.821 -> csum 0xba
14:56:07.004 -> $$$1$10$1111`0$$r$1$10$1111`0$$r$111111`0r1$$$
14:56:07.096 -> ready
15:00:05.119 -> AT
15:00:05.119 ->
15:00:05.119 -> OK
15:01:08.226 -> AT+CWMODE=3
15:01:08.319 ->
15:01:08.319 -> 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
```

7.1.2 1.2 Blynk konfigurieren

1. Besuchen Sie die [BLYNK](#)-Seite 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, um die Konto-Registrierung 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. Danach erscheint die **Blynk Tour**. 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**.

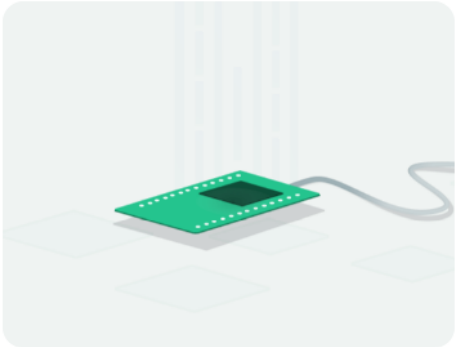
Quickstart

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 aus.

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

▼

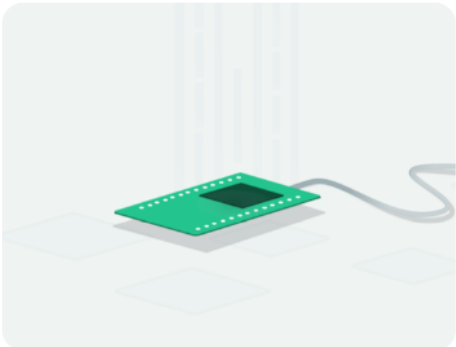
What is your device connectivity type

Blynk supports various connection types (BLE is not supported yet).

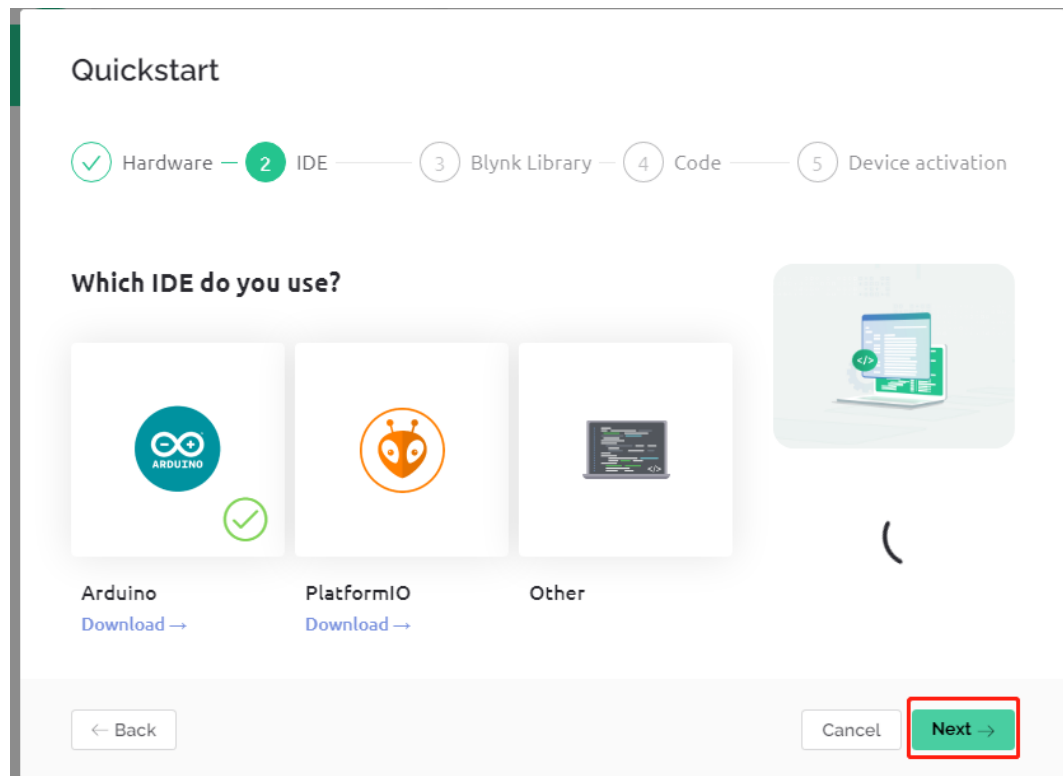
WiFi

▼

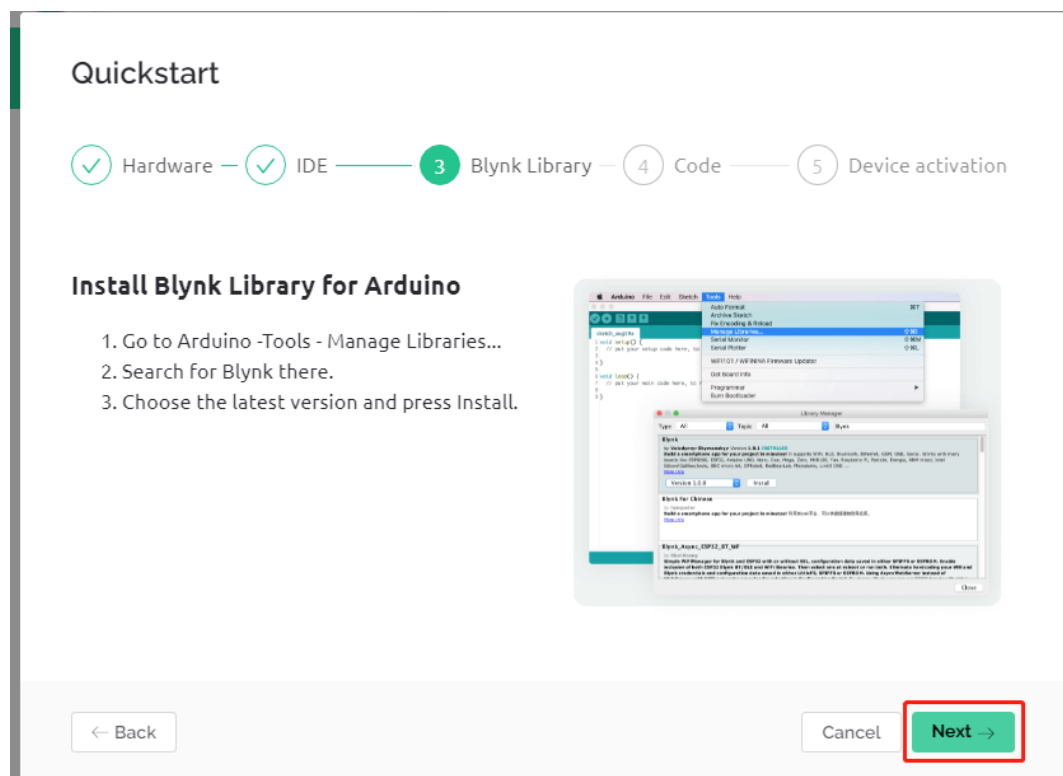
CancelNext →



7. Ihnen wird mitgeteilt, welche IDE Sie benötigen. Wir empfehlen die **Arduino IDE**.



8. Hier ist die benötigte Bibliothek aufgeführt. Die hier empfohlene Bibliothek ist jedoch etwas problematisch. Wir müssen andere Bibliotheken manuell hinzufügen (dazu später mehr). Klicken Sie hier auf **Next** und eine neue Vorlage sowie ein Gerät werden erstellt.



9. Die nächsten Schritte wären, den relevanten Code hochzuladen und Ihr Board mit Blynk zu verbinden. Da es

jedoch ein Problem mit der zuvor bereitgestellten Bibliothek gibt, müssen Sie erneut andere Bibliotheken hinzufügen. Klicken Sie also hier auf **Cancel**, um den **Quick Start** zu beenden.

Quickstart

✓ Hardware — ✓ IDE — ✓ Blynk Library — **4** Code — 5 Device activation

Here is a code for your device

1. Enter your Wi-Fi network SSID (name) and password to connect your device.
 * We never store or send this information anywhere. It's only used to generate the firmware code. You can leave these fields empty and manually add WiFi credentials in your sketch.

Wi-Fi Network Name (SSID)
 Password

← Back **Cancel** Next →

```

/*****
 * This is a simple demo of sending and receiving some data.
 * Be sure to check out other examples!
 *****/

// Template ID, Device Name and Auth Token are provided by the Blynk.Cloud
// See the Device Info tab, or Template settings
#define BLYNK_TEMPLATE_ID      "TMPLDbHgRCnL"
#define BLYNK_DEVICE_NAME     "Quickstart Device"
#define BLYNK_AUTH_TOKEN      "zsWQM55heRPSotJcGmPz0Zhpj9Mr4ivv"

// Comment this out to disable prints and save space
// #define BLYNK_PRINT true
  
```

10. Klicken Sie auf den **Search**-Button und Sie sehen das Gerät, das Sie gerade erstellt haben.

My organization - 3901HS

DEVICES

My devices 1

All 1

LOCATIONS

My locations 0

All 0

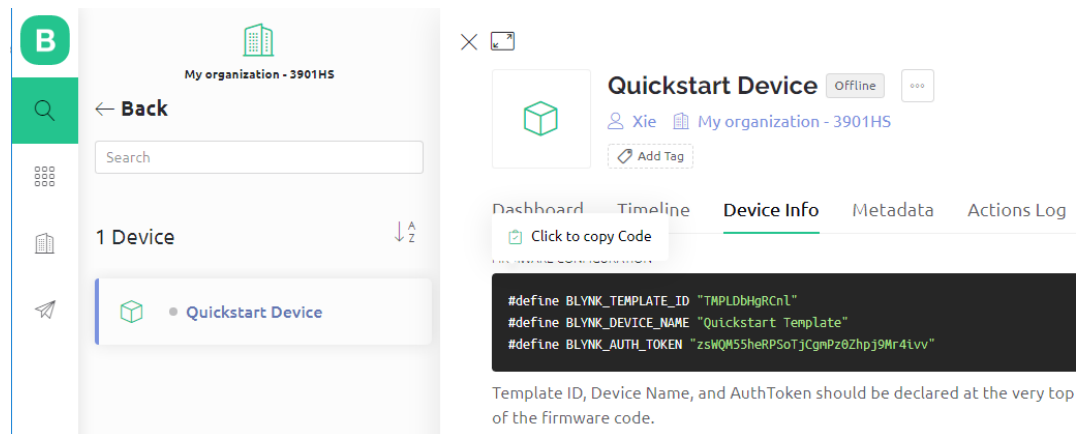
My devices

+ New Device

1 Device

Device name	Device	Actions
Quickstart Device	Xie	

11. Gehen Sie zu diesem **Quickstart Device** und auf der **Device info**-Seite sehen Sie **TEMPLATE_ID**, **DEVICE_NAME** und **AUTH_TOKEN**. Diese Informationen werden Sie später benötigen und müssen sie kopieren.






7.1.3 1.3 Die benötigten Bibliotheken hinzufügen

Um Blynk in 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.

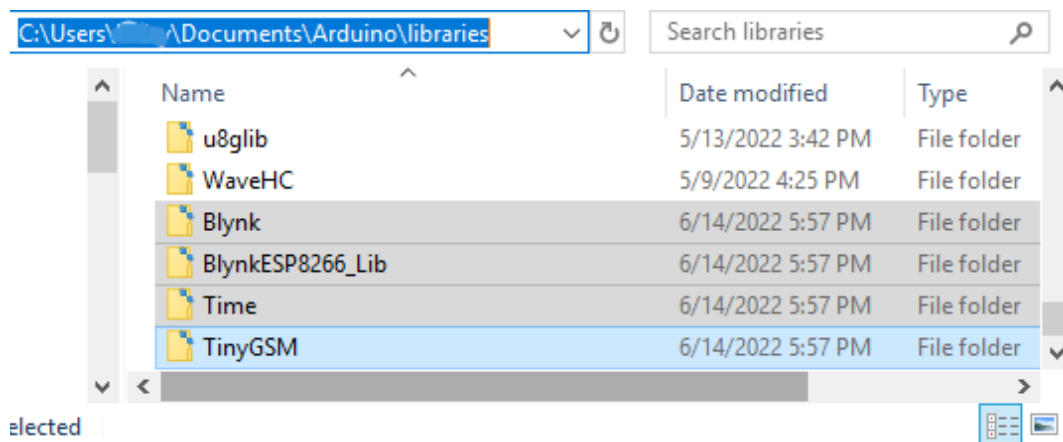
▼ Assets 3

 Blynk_Release_v1.1.0.zip	779 KB	22 days ago
 Source code (zip)		22 days ago
 Source code (tar.gz)		22 days ago

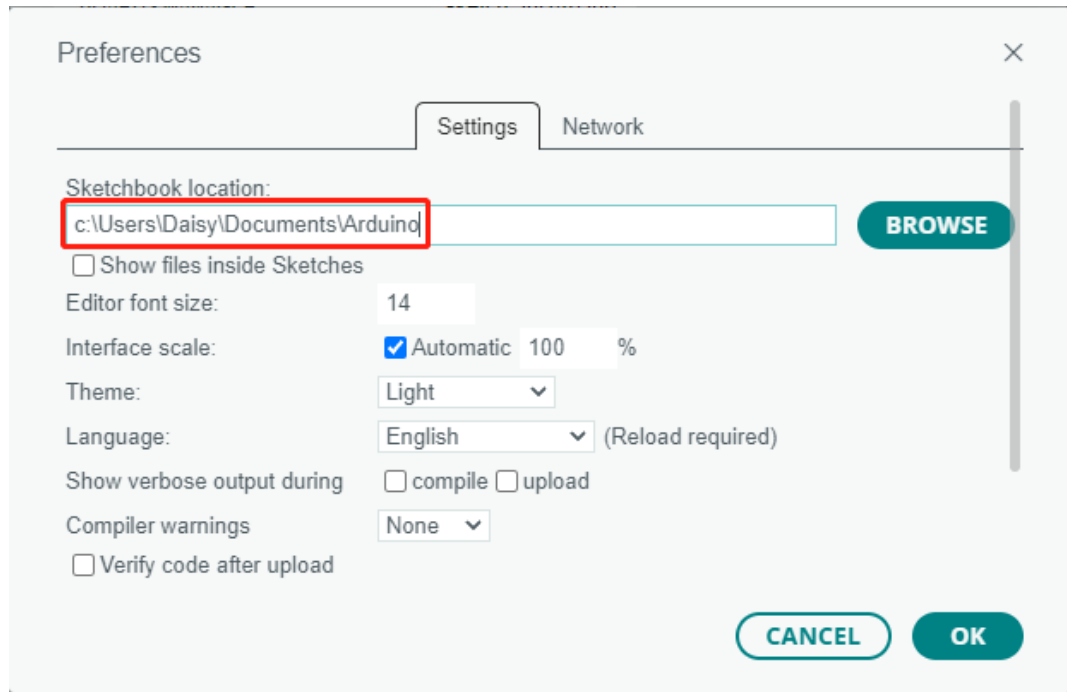
2. Entpacken Sie diese Datei und Sie sehen die folgenden Ordner.

..			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-Bibliotheksverzeichnis der Arduino IDE ein, welches normalerweise unter C:\Users\xxx\Documents\Arduino\libraries zu finden ist.



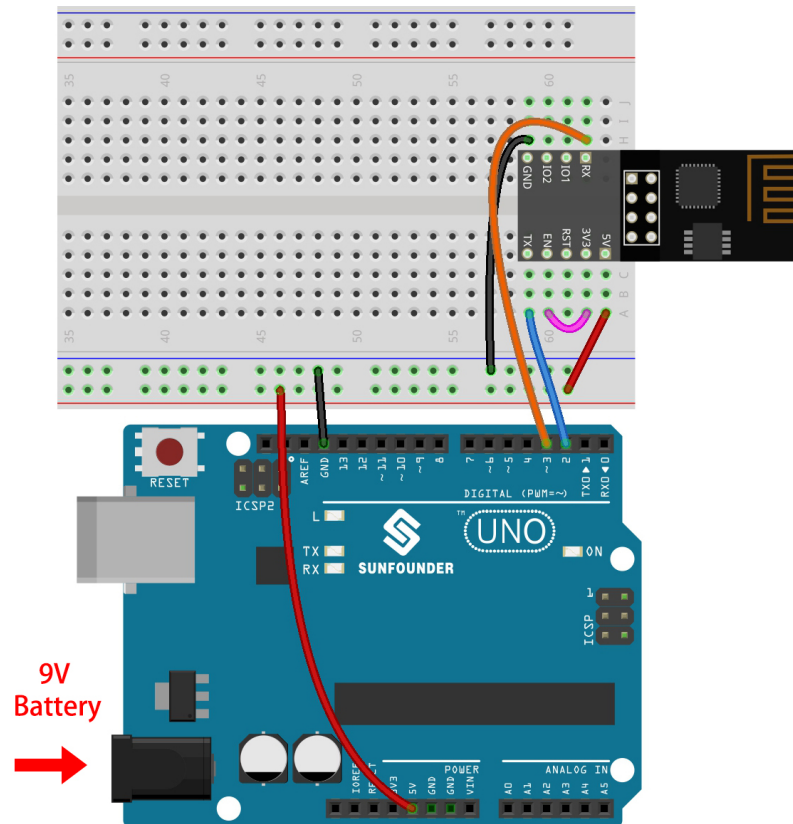
4. Falls Ihr Bibliotheksverzeichnis anders ist, können Sie es überprüfen, indem Sie zu **File -> Preferences** navigieren.



7.1.4 1.4 Verbindung des R3-Boards mit Blynk herstellen

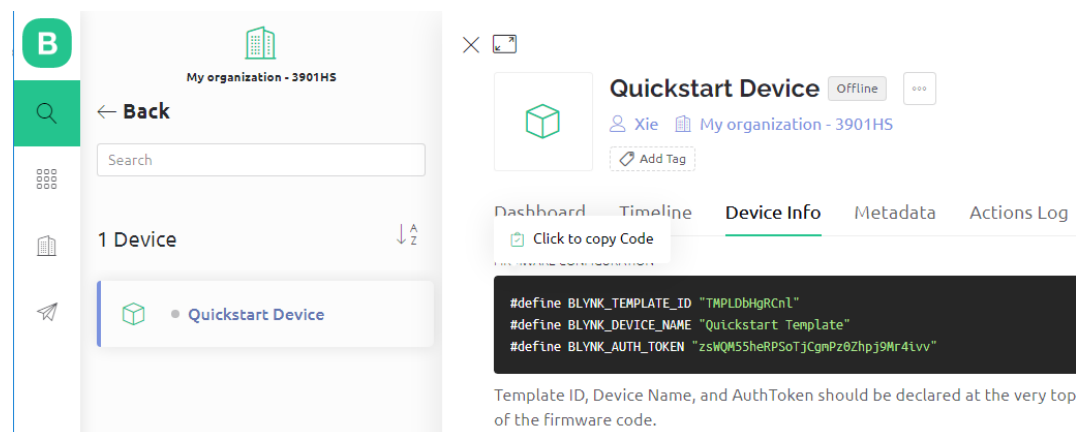
1. Verbinden Sie das ESP8266-Modul erneut mit dem R3-Board. Hier wird die Software-Serienschnittstelle verwendet, sodass TX und RX mit den Pins 2 und 3 des R3-Boards verbunden werden.

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` im 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 Zeilen ermöglichen es Ihrem R3-Board, Ihr Blynk-Konto zu finden.

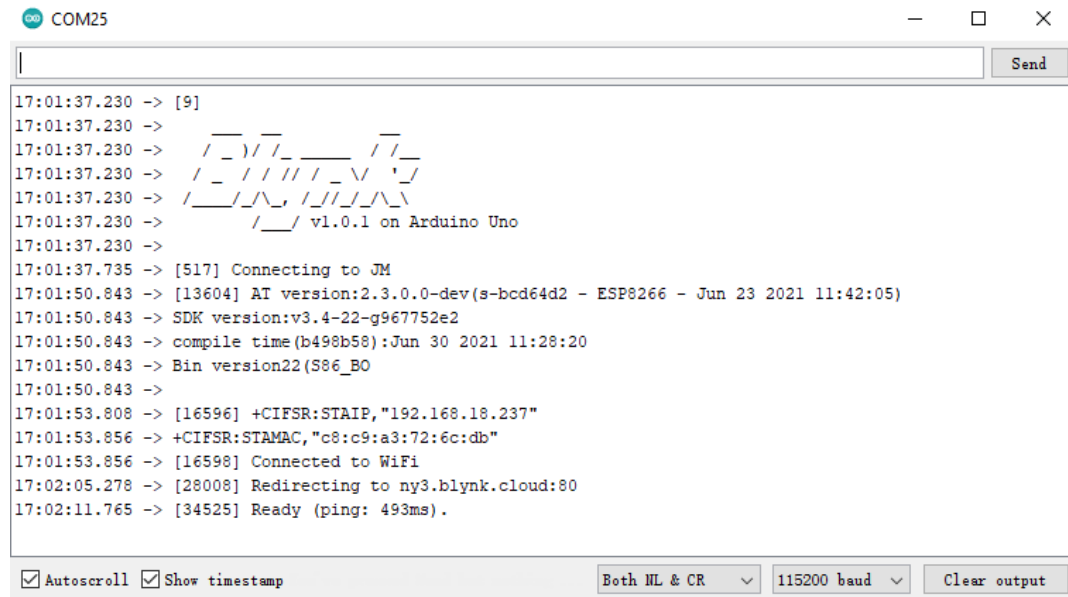
```
#define BLYNK_TEMPLATE_ID "TMPLxxxxxx"
#define BLYNK_DEVICE_NAME "Device"
#define BLYNK_AUTH_TOKEN "YourAuthToken"
```



- Geben Sie die `ssid` und das `password` des von Ihnen verwendeten WLANs ein.

```
char ssid[] = "ssid";
char pass[] = "password";
```

5. Laden Sie den Code auf das R3-Board, öffnen Sie dann den seriellen Monitor und stellen Sie die Baudrate auf 115200 ein. Wenn das R3-Board erfolgreich mit Blynk kommuniziert, zeigt der serielle Monitor das Zeichen **ready** an.



The screenshot shows the Serial Monitor window for COM25. The output text is as follows:

```
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).
```

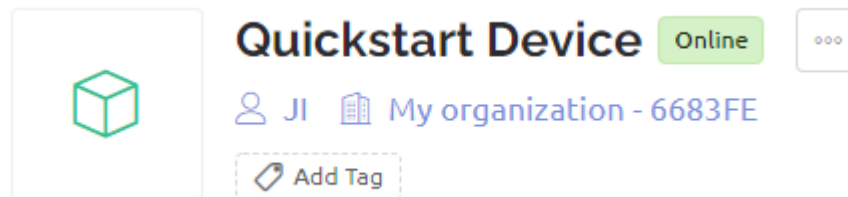
At the bottom of the window, the settings are: ☒ Autoscroll, ☒ Show timestamp, Both NL & CR, 115200 baud, and a Clear output button.

Bemerkung: Wenn die Meldung ESP antwortet 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 Pin RST für 1 Sekunde mit GND verbinden und dann trennen.
- Drücken Sie die Reset-Taste auf dem R3-Board.

Manchmal müssen Sie den obigen Vorgang 3-5 Mal wiederholen. Bitte haben Sie Geduld.

6. Der Status von Blynk ändert sich von **offline** zu **online**.



7.2 2. Daten von Blynk abrufen

In diesem Kapitel erfahren Sie, wie Sie den Schaltkreis mit Blynk steuern können. Lassen Sie uns die LEDs über das Internet leuchten!

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

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

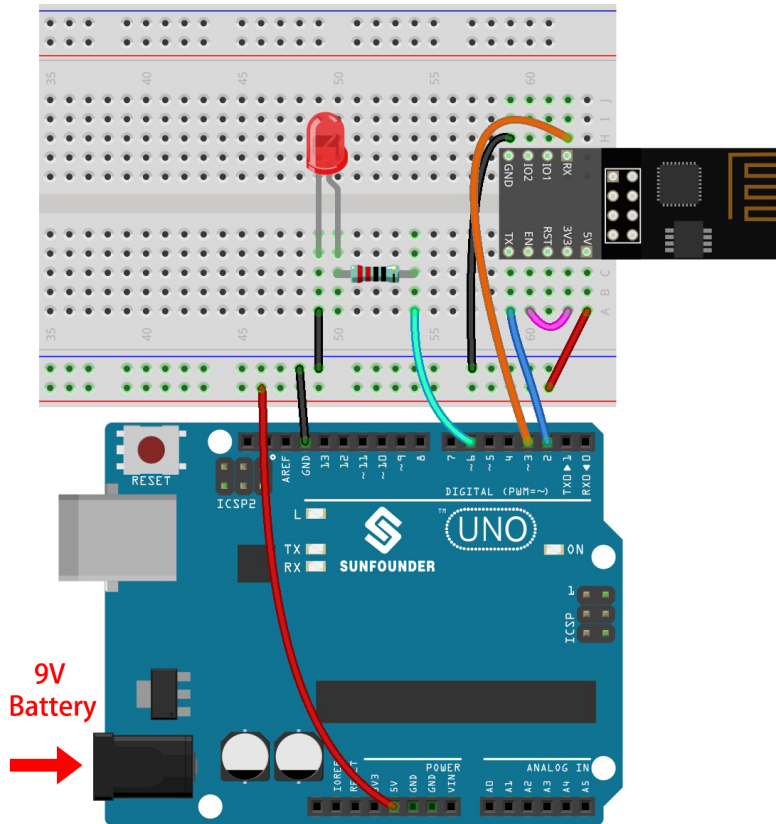
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>ESP8266 Modul</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	

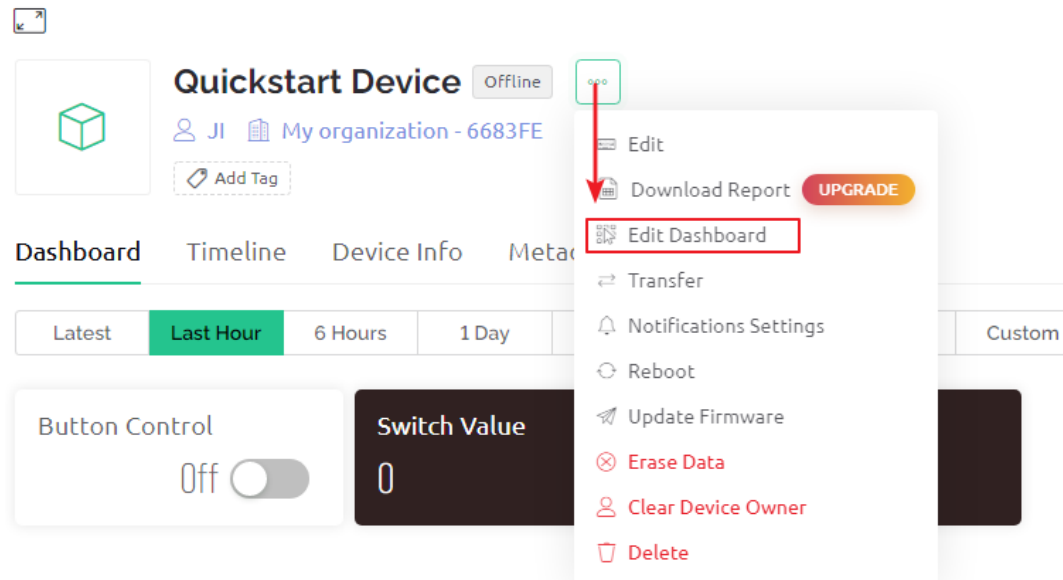
1. Schaltkreis aufbauen

Bemerkung: Das ESP8266-Modul benötigt einen hohen Strom, um eine stabile Betriebsumgebung zu gewährleisten. Stellen Sie daher sicher, dass die 9V-Batterie angeschlossen ist.



2. Dashboard bearbeiten

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 dem R3-Board, sich gegenseitig zu erkennen. Um den vollständigen Konfigurationsprozess zu erleben, entfernen Sie alle Datastreams von der Datastreams-Seite.

Quickstart Template

Cancel Save And Apply

Info Metadata **Datastreams** Events Automations Web Dashboard Mobile Dashboard

Search datastream


+ New Datastream

4 Datastreams selected of 4 Clear selection

<input checked="" type="checkbox"/>	ID	Name	Alias	Color	Pin	Data Type	Units	Is	Actions
<input checked="" type="checkbox"/>	1	Switch Control	Switch Control		V0	Integer		fa	
<input checked="" type="checkbox"/>	2	Switch Value	Switch Value		V1	Integer		fa	
<input checked="" type="checkbox"/>	3	Seconds	Seconds		V2	Integer		fa	
<input checked="" type="checkbox"/>	4	Button Image	Button Image		V3	String		fa	

- Bitte lesen Sie die Warnung sorgfältig durch und bestätigen Sie vor dem Löschen der Datastreams, dass sie korrekt ist.

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.

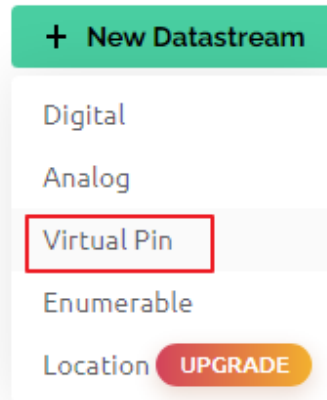
Type DELETE in the field below to proceed with the deletion.

☒ I fully understand that this action is critical and will lead to data loss

4. Erstellen Sie einen Datastream des Typs **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 Knopf und die LED nur EIN und AUS sein müssen, setzen Sie DEN DATENTYP auf **Integer** und MIN und MAX auf **0** und **1**.

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATA TYPE:

UNITS:

MIN: MAX: DEFAULT VALUE:

6. Gehen Sie zur **Web Dashboard**-Seite und löschen Sie die vorhandenen Widgets.

Quickstart Template

Info Metadata Datastreams Events Automations **Web Dashboard** Mobile Dashboard

Widget Box
3 of 30 widgets

CONTROL

Switch ☒

Slider

Number Input

Device name Online

Device Owner Company Name

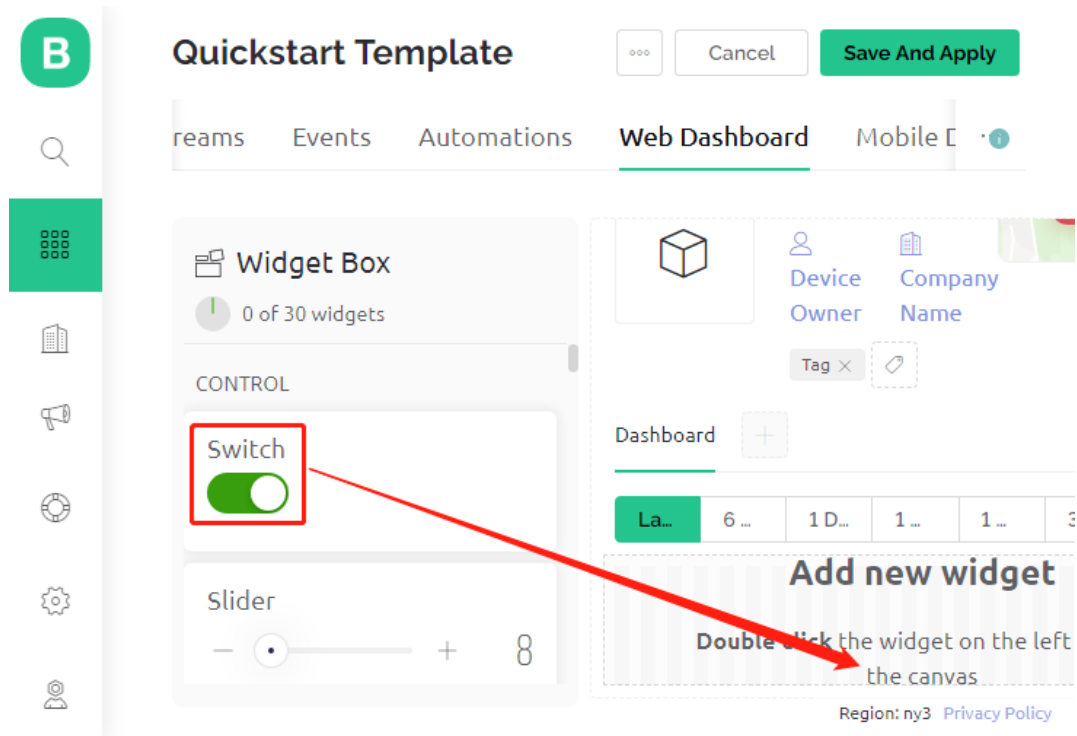
Tag

Dashboard

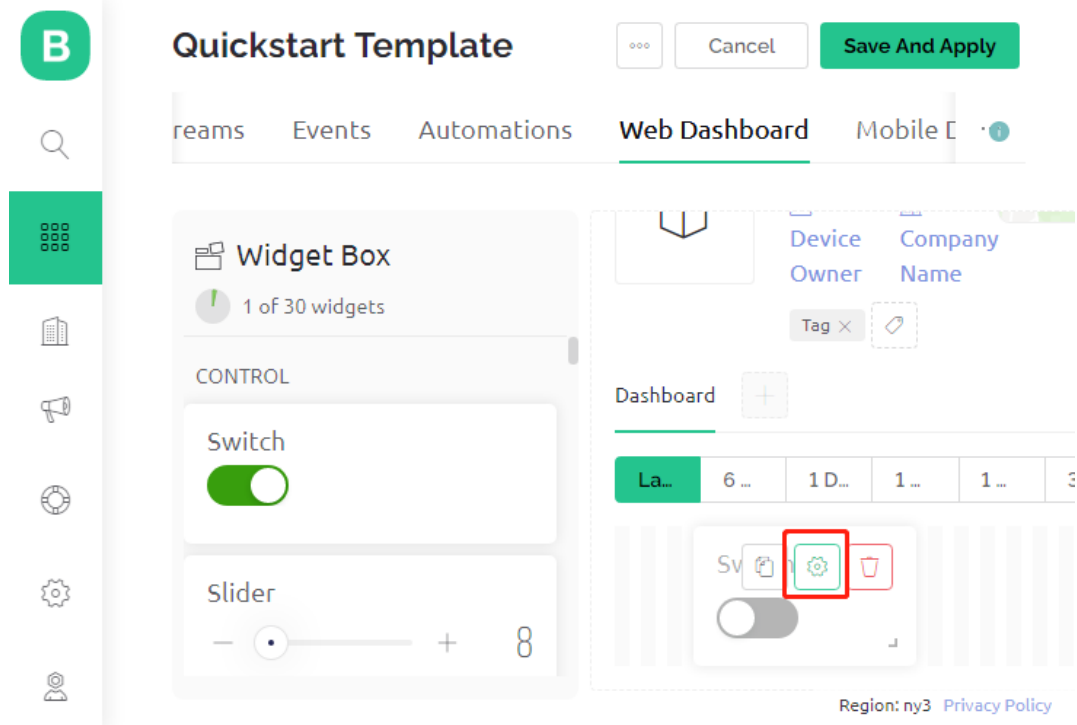
Last H... 6 Hours 1 Day 1 Week 1 Month 3 Mont... Custom

Switch... Uptime

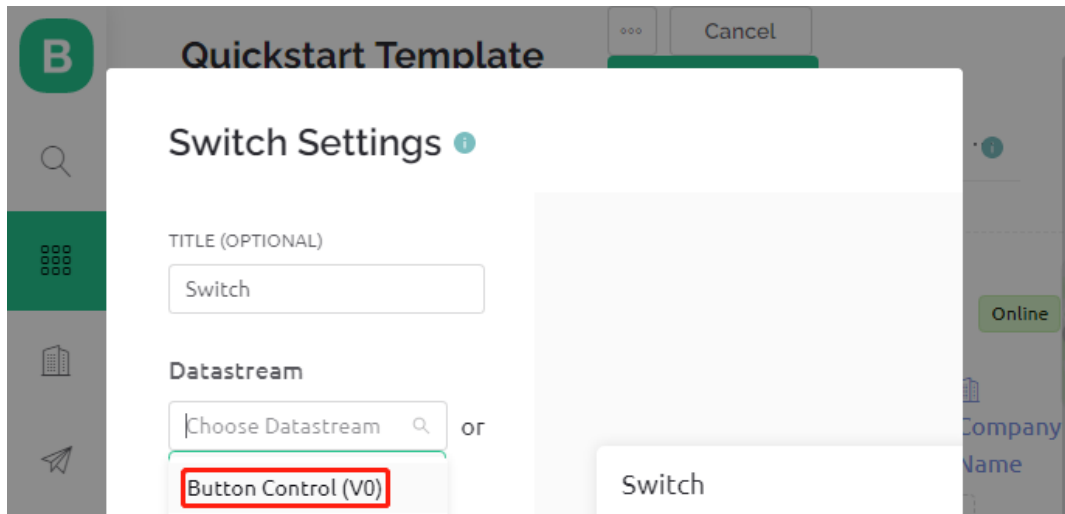
7. Ziehen Sie ein **switch**-Widget aus dem **Widget Box** links in den Arbeitsbereich.



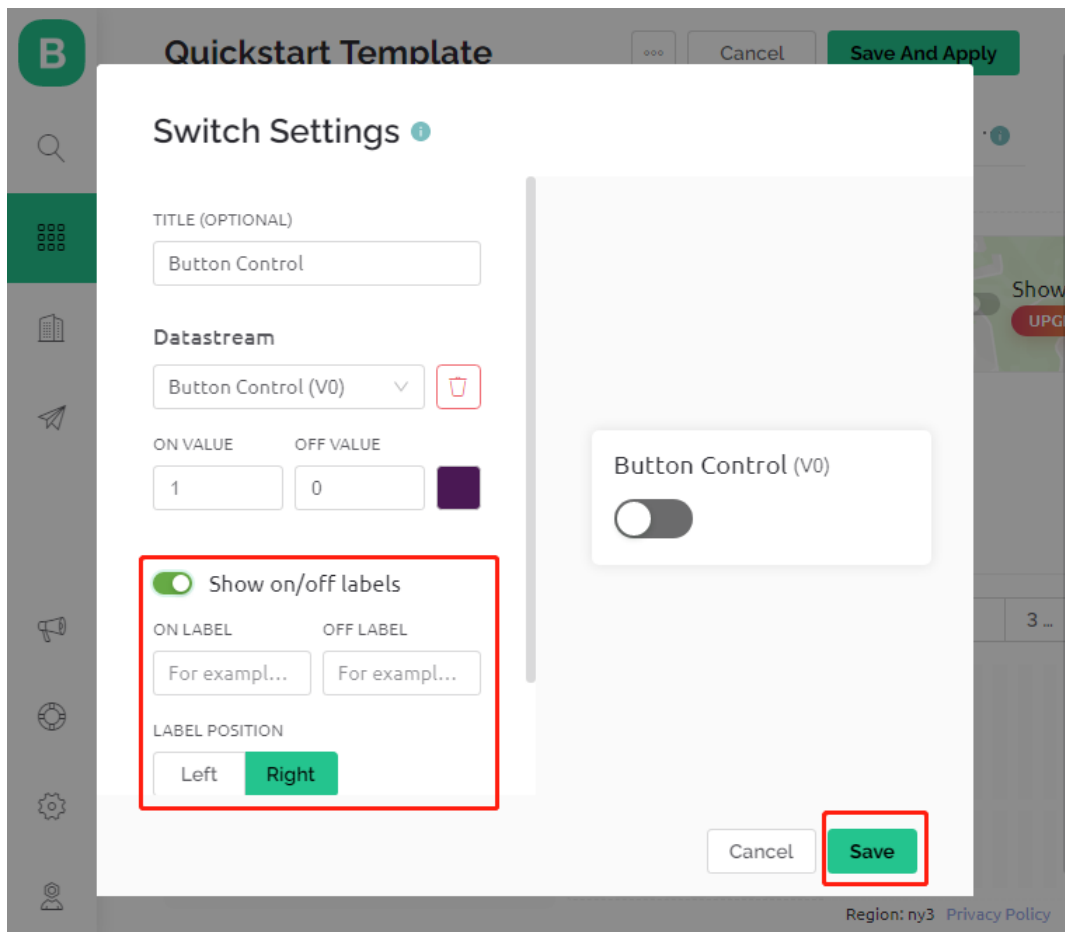
8. Nun geht es an die Einrichtung.



9. Wählen Sie **Datastream** wie zuvor eingestellt.



10. Nach Auswahl von Datastream sehen Sie einige benutzerdefinierte Einstellungen. Drücken Sie dann auf Speichern.

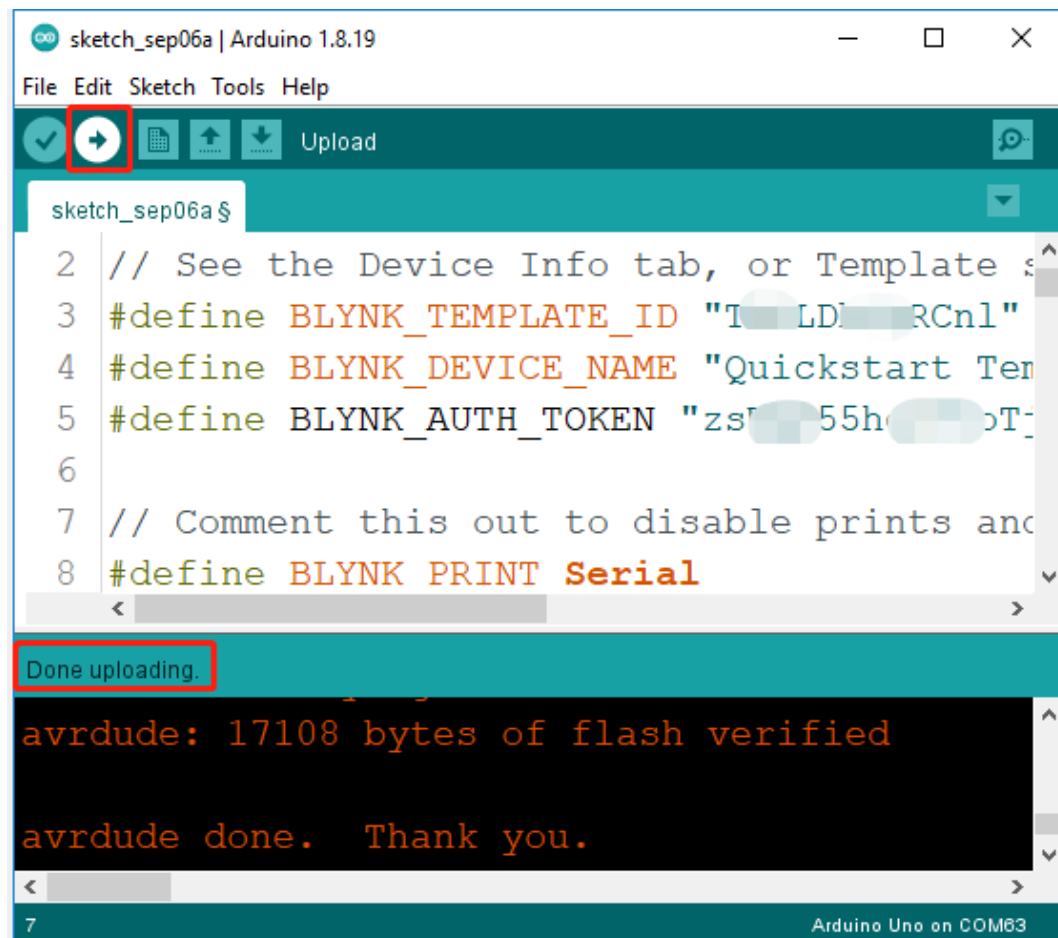


11. Klicken Sie abschließend auf **Save And Apply**.

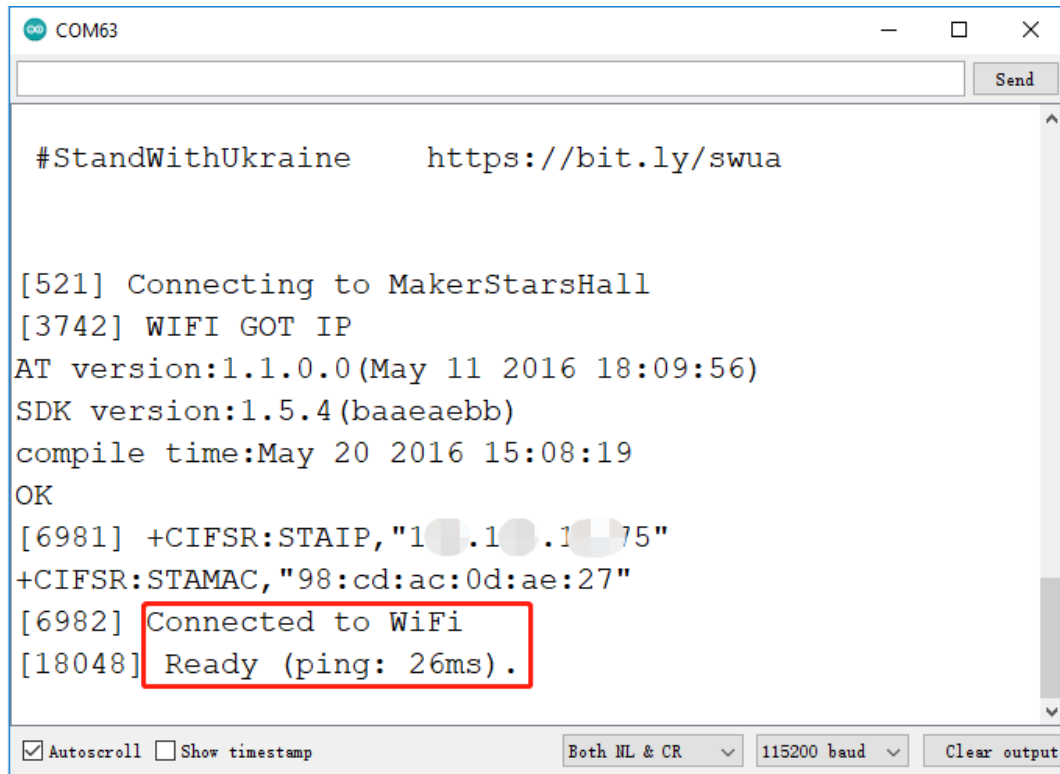


3. Den Code ausführen

1. Öffnen Sie die Datei `2.get_data_from_blynk.ino` unter dem 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, Device Name und Auth Token durch Ihre eigenen. Geben Sie auch die ssid und das password Ihres WiFi ein. Für detaillierte Anleitungen verweisen Sie bitte auf [1.4 Verbindung des R3-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 Aufforderung, z.B. 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,"1.1.1.175"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

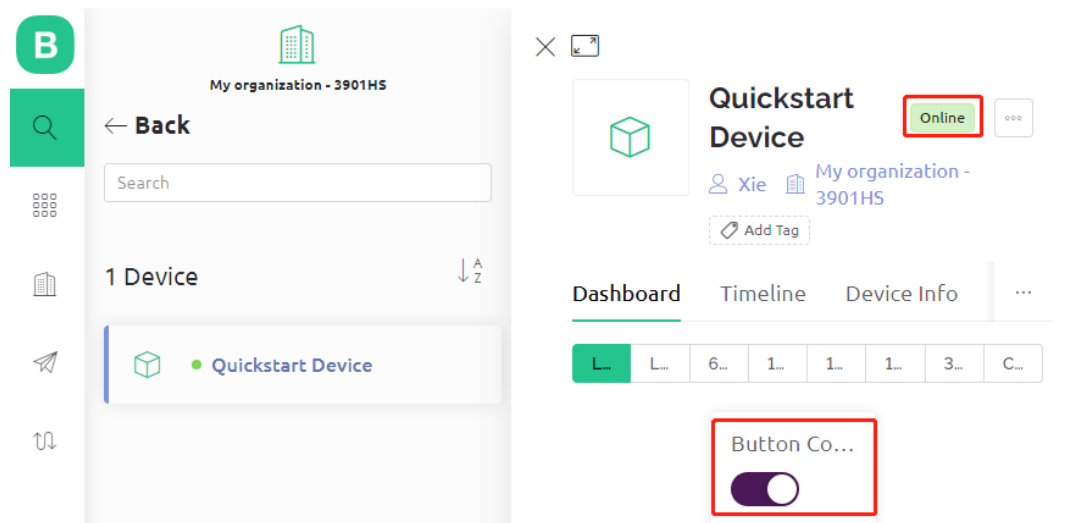
☒ Autoscroll ☐ Show timestamp Both NL & CR 115200 baud Clear output

Bemerkung: Wenn beim Verbinden die Meldung ESP antwortet 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 Pin RST für 1 Sekunde auf GND legen und dann wieder entfernen.
- Drücken Sie den Reset-Button auf dem R3-Board.

Manchmal müssen Sie den obigen Vorgang 3-5 Mal wiederholen, bitte haben Sie Geduld.

5. Zurück bei Blynk sehen Sie, dass der Status auf online geändert wurde und Sie können nun das Schalter-Widget auf Blynk verwenden, um die an das R3-Board angeschlossene LED zu steuern.



6. Wenn Sie Blynk auf mobilen Geräten verwenden möchten, verweisen Sie bitte auf *Wie verwendet man Blynk auf einem mobilen Gerät?*.

Wie funktioniert das?

Der Unterschied zwischen dem Code in diesem Projekt und dem Code im vorherigen Kapitel *1.4 Verbindung des R3-Boards mit Blynk herstellen* sind die folgenden Zeilen.

```
const int ledPin=6;

BLYNK_WRITE(V0)
{
    int pinValue = param.asInt();// Zuweisen des eingehenden Werts von Pin V0 zu einer
    ↪ Variablen
    // Sie können auch Folgendes verwenden:
    // String i = param.asStr();
    // double d = param.asDouble();
    digitalWrite(ledPin,pinValue);
}

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

Bezüglich der `pinMode` und `digitalWrite` des `ledPin` bin ich sicher, dass Sie bereits damit vertraut sind, deshalb werde ich sie nicht noch einmal erklären. Worauf Sie sich konzentrieren müssen, ist die Funktion `BLYNK_WRITE(V0)`.

Was sie macht ist, dass, wenn der Wert von Blynks `V0` sich ändert, Blynk.Cloud Ihrem Gerät mitteilt „Ich schreibe auf den **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 Switch-Widget zugewiesen. Das bedeutet, dass jedes Mal, wenn wir das Switch-Widget bedienen, `BLYNK_WRITE(V0)` ausgelöst wird.

Wir schreiben zwei Anweisungen in dieser 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 erhaltenen Wert von V0 auf den ledPin, sodass das Schalter-Widget auf Blynk die LED steuern kann.

7.3 3. Daten an Blynk senden

In diesem Kapitel zeigen wir Ihnen, wie Sie Daten an Blynk senden können.

Wir erstellen hier ein Tür- und Fenstererkennungsgerät. Die Schaltung mit dem Reed-Schalter wird neben der Tür und dem Fenster platziert und der Magnet wird am Rand der Tür und des Fensters angebracht. Wenn die Tür oder das Fenster geschlossen sind, wird der Reed-Schalter durch die magnetische Kraft eingeschaltet und der entsprechende Pinwert auf dem R3-Board ändert sich. Blynk.cloud erhält diesen Wert, sodass Sie sehen können, ob die Türen und Fenster Ihres Hauses geschlossen sind, auch wenn Sie nicht zu Hause sind.

Jetzt werden wir ein LED-Widget in Blynk verwenden, um anzuzeigen, ob Ihre Fenster und Türen geschlossen sind (d.h. ob der Reed-Schalter ein- oder ausgeschaltet ist).

Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

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

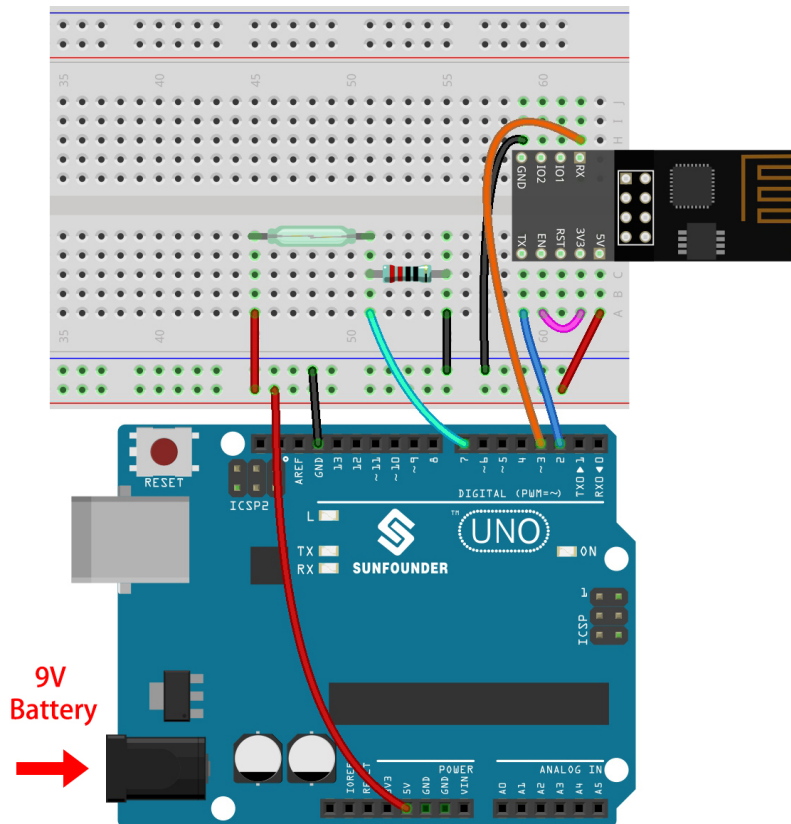
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>ESP8266 Modul</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Reedschalter</i>	-

1. Die Schaltung 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.

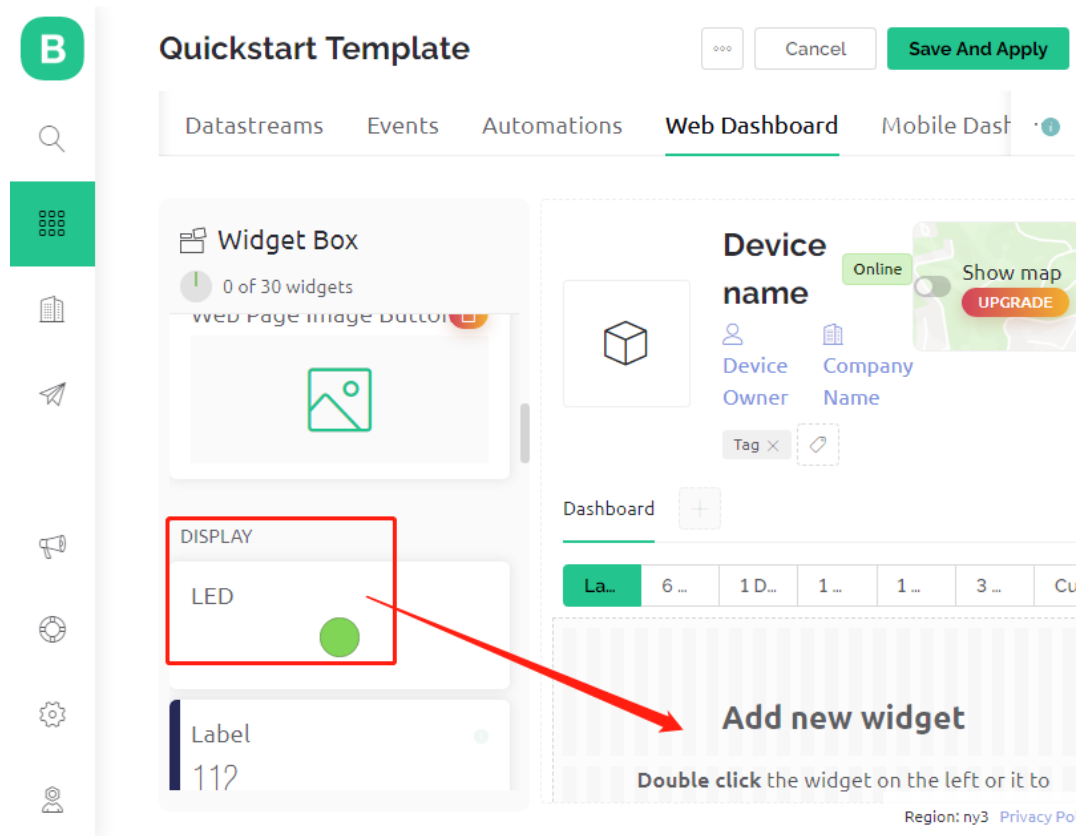


2. Dashboard bearbeiten

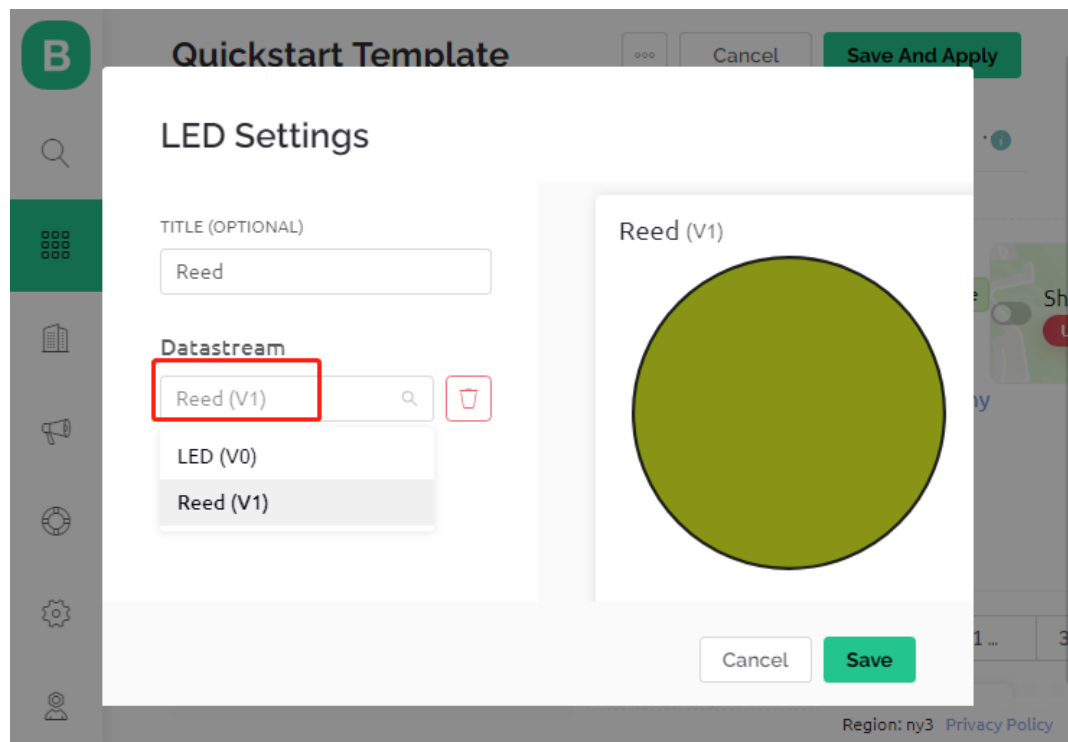
1. Erstellen Sie einen **Datastream** vom Typ **Virtual Pin** auf der **Datastream**-Seite, um den Wert des Reed-Schalters abzurufen. Stellen Sie den DATENTYP auf **Integer** und MIN und MAX auf **0** und **1**.

The screenshot shows the 'Virtual Pin Datastream' configuration window. The 'NAME' field is highlighted with a red box and contains the text 'Reed'. The 'PIN' field is also highlighted with a red box and contains 'V1'. The 'DATA TYPE' dropdown is set to 'Integer'. The 'MIN' field is set to 0 and the 'MAX' field is set to 1. The 'DEFAULT VALUE' field is set to 0. There are 'Cancel' and 'Save' buttons at the bottom right.

2. Ziehen Sie ein **LED widget** auf die **Web Dashboard**-Seite. Bei einem Wert von 1 leuchtet es (farbig), ansonsten ist es weiß.

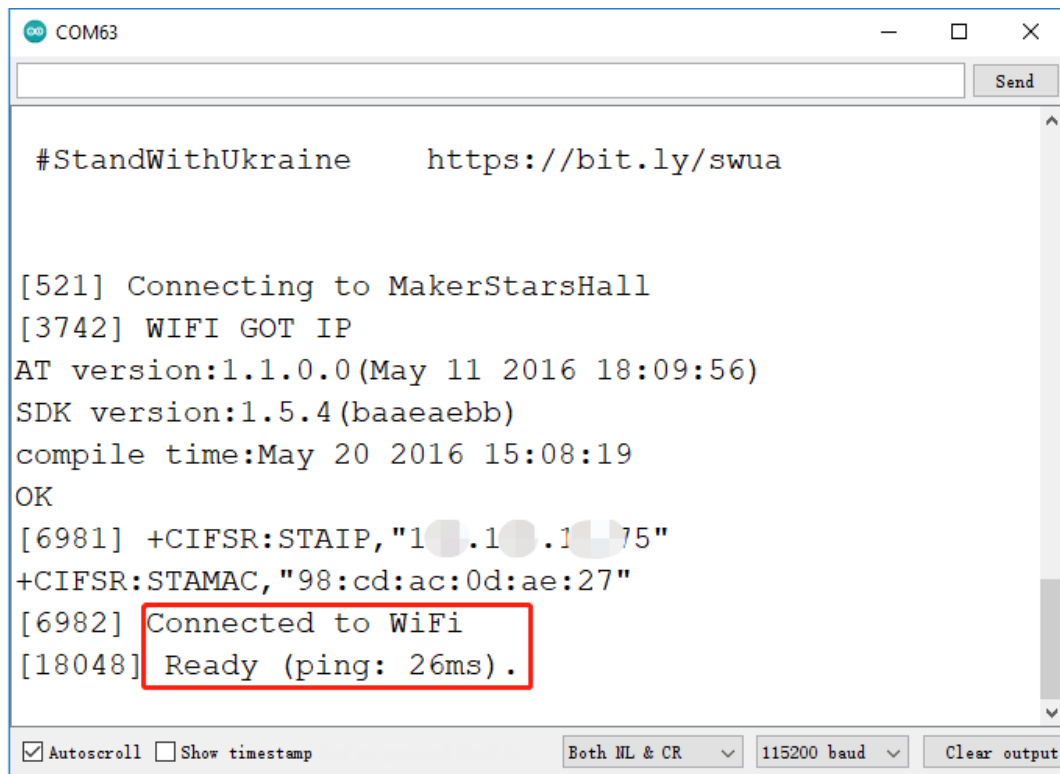


3. Auf der Einstellungsseite des **LED widget** wählen Sie **Datastream** als **Reed(V1)** und speichern es.



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 Werte. Geben Sie außerdem die ssid und das password Ihres WLANs ein. Für detaillierte Anleitungen verweisen Sie bitte auf [1.4 Verbindung des R3-Boards mit Blynk herstellen](#).
3. Nachdem Sie das richtige Board und den Port ausgewählt haben, klicken Sie auf die Schaltfläche **Upload**.
4. Öffnen Sie den Seriellen Monitor (Baudrate auf 115200 einstellen) und warten Sie, bis eine Meldung wie eine erfolgreiche Verbindung angezeigt wird.



```
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,"1.1.1.175"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
```

Bemerkung: Wenn die Meldung ESP antwortet nicht erscheint, wenn Sie sich verbinden, 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 1 Sekunde lang auf GND setzen und dann entfernen.
- Drücken Sie die Reset-Taste auf dem R3-Board.

Manchmal müssen Sie den obigen Vorgang 3-5 Mal wiederholen. Bitte haben Sie Geduld.

5. Jetzt zeigt Blynk den Status Ihrer Türen und Fenster an. Wenn Ihre Türen und Fenster geschlossen sind, wird das LED-Widget grün sein, ansonsten grau.
6. Wenn Sie Blynk auf mobilen Geräten verwenden möchten, verweisen Sie bitte auf [Wie verwendet man Blynk auf einem mobilen Gerät?](#).

Wie funktioniert das?

In diesem Beispiel sollten Sie sich auf die folgenden Zeilen konzentrieren. „Daten jede Sekunde an den Blynk Cloud's 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(); // Initiiert BlynkTimer
}
```

Die Blynk-Bibliothek bietet einen eingebauten Timer; 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)` wird verwendet, um den Datenpin-Wert für V1 zu schreiben.

```
void myTimerEvent()
{
  Blynk.virtualWrite(V1, pinValue);
}
```

7.4 4. Cloud-Musikplayer

Das Ziel dieses Projekts ist es, einen Musikplayer mit Blynk zu erstellen. Die Musik wird genauso abgespielt wie in *5.7 Tone() oder noTone()*, indem man das Lied im Programm schreibt und es mit einem passiven Buzzer abspielt. 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 Set zu kaufen, hier ist der Link:

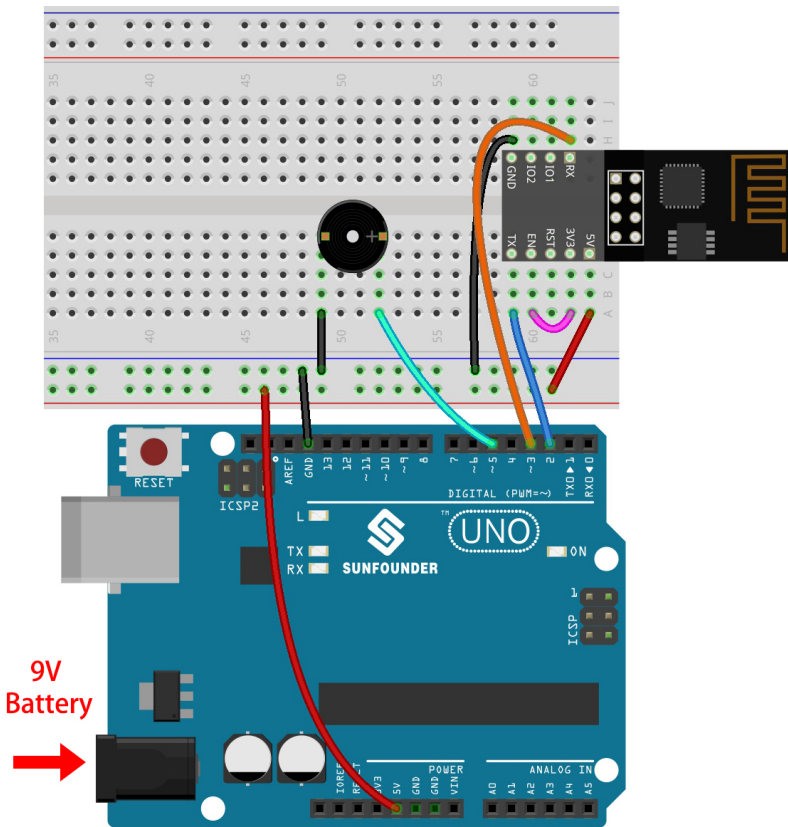
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>ESP8266 Modul</i>	
<i>Jumper-Kabel</i>	
<i>Summer</i>	

1. Schaltung 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.



2. Dashboard bearbeiten

1. Erstellen Sie einen **Datastream** des Typs **Virtual Pin** auf der **Datastream**-Seite als den später vom Slider-Widget oder Code geänderten Wert. Setzen Sie den DATENTYP auf **Integer** und MIN und MAX auf **0** und **30**.

Quickstart Templa...

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

2. Erstellen Sie auch einen weiteren **Datastream** des Typs **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* gesetzt); 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 gelten. Sie müssen jedoch die entsprechende Pinnummer im Code ändern.

3. Den 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, den Device Name und den Auth Token durch Ihre eigenen Daten. Sie müssen auch die ssid und das password Ihres verwendeten WLANs eingeben. Für detaillierte Anleitungen verweisen wir auf [1.4 Verbindung des R3-Boards mit Blynk herstellen](#).
3. Nachdem Sie das richtige Board und den Port ausgewählt haben, 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

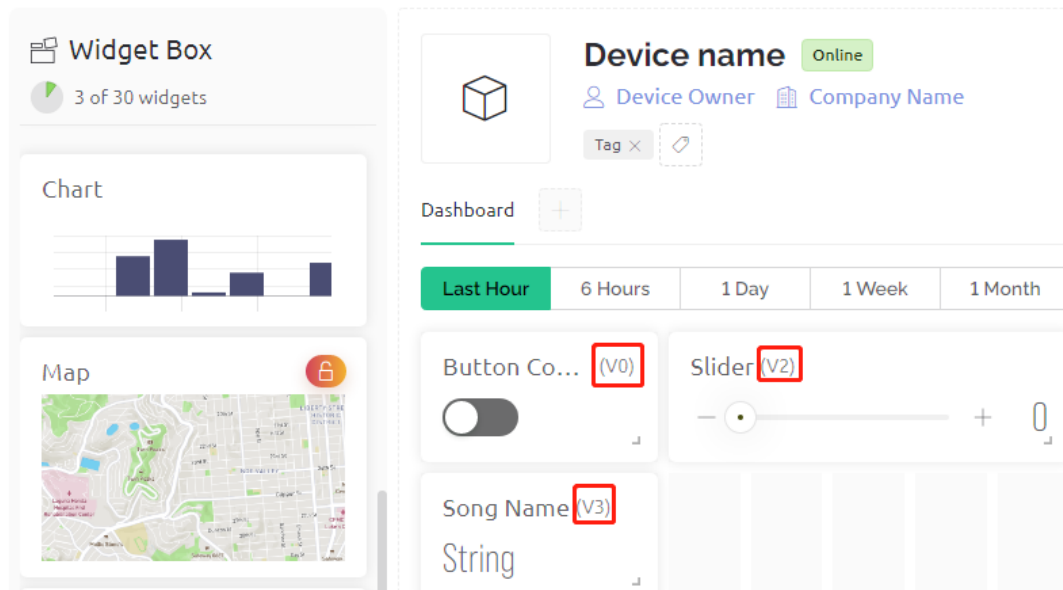
[521] Connecting to MakerStarsHall
[3742] WIFI GOT IP
AT version:1.1.0.0(May 11 2016 18:09:56)
SDK version:1.5.4(baaeaebb)
compile time:May 20 2016 15:08:19
OK
[6981] +CIFSR:STAIP,"192.168.1.75"
+CIFSR:STAMAC,"98:cd:ac:0d:ae:27"
[6982] Connected to WiFi
[18048] Ready (ping: 26ms).
  
```

Bemerkung: Wenn die Nachricht ESP reagiert nicht 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 und dann trennen.
- Drücken Sie den Reset-Button auf dem R3-Board.

Manchmal müssen Sie den obigen Vorgang 3-5 Mal wiederholen, bitte haben Sie Geduld.

5. Nun können Sie mit Blynks Button-Steuerwidget die Musik starten/anhalten und mit dem Schieberegler den Wiedergabefortschritt anpassen, und Sie sehen auch den Namen des Musikstücks.



6. Wenn Sie Blynk auf mobilen Geräten nutzen möchten, verweisen wir auf [Wie verwendet man Blynk auf einem mobilen Gerät?](#).

Wie funktioniert es?

Der Datenstrom **V0** wird verwendet, um den Status des Switch-Widgets zu erhalten und der Variablen **musicPlayFlag** zuzuweisen, die das Pausieren und Abspielen der Musik steuert.

```
int musicPlayFlag=0;

BLYNK_WRITE(V0)
{
    musicPlayFlag = param.asInt(); // MUSIK STARTEN/PAUSIEREN
}
```

Der Datenstrom **V2** wird verwendet, um den Wert des Schieberegler-Widgets zu erhalten und ihn der Variablen **scrubBar** zuzuweisen, wenn der Schieberegler verschoben wird.

```
int scrubBar=0;

BLYNK_WRITE(V2)
{
    scrubBar=param.asInt();
}
```

Wenn das Gerät mit dem **Blynk Cloud** verbunden ist, schreiben Sie den Musiknamen für den **V3** Datenstrom und zeigen Sie ihn dann mit dem **Label**-Widget an.

```
BLYNK_CONNECTED() {
    String songName = "Ode an die Freude";
    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 Fortschrittsbalkenvariable `scrubBar` um 2 erhöht, und der Wert wird dann an 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);
    }
}
```

7.5 5. Überwachung des häuslichen Umfelds

In diesem Kapitel werden wir Blynk verwenden, um einen Monitor für die häusliche Umgebung zu erstellen. Mit dem DHT11 und dem Fotowiderstand können Sie die Temperatur, Luftfeuchtigkeit und Lichtintensität eines Raumes messen. Indem Sie diese Werte an Blynk senden, können Sie die Umgebung Ihres Zuhauses über das Internet überwachen.

Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist definitiv praktischer, ein komplettes Set zu kaufen. Hier ist der Link:

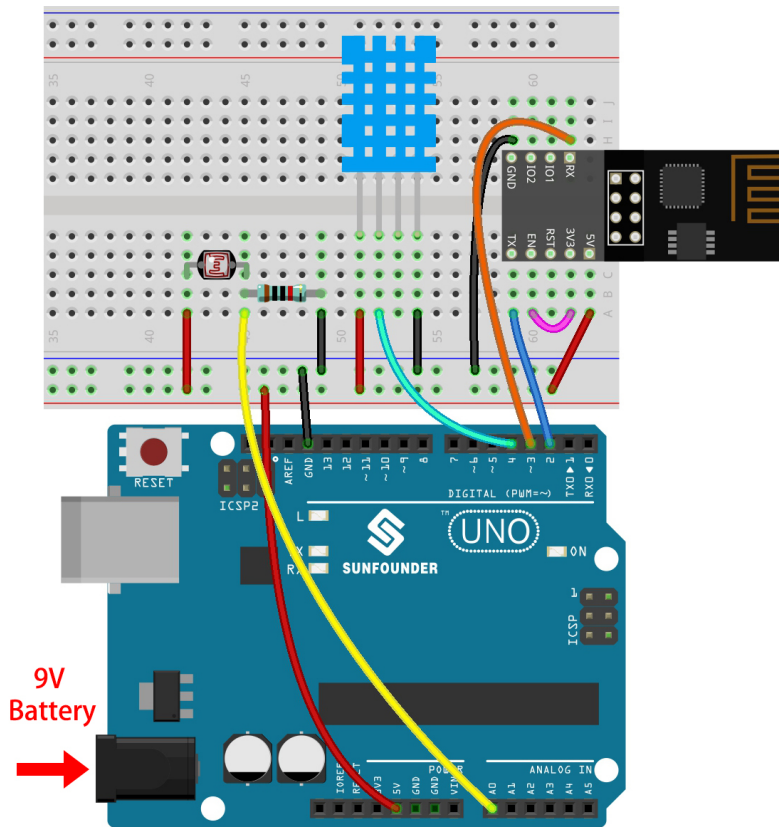
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Alternativ können Sie diese auch über die untenstehenden Links einzeln erwerben.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</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. Den Schaltkreis aufbauen

Bemerkung: Das ESP8266-Modul benötigt einen hohen Strom, um eine stabile Betriebsumgebung zu gewährleisten. Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.



2. Dashboard bearbeiten

1. Für die Erfassung von Luftfeuchtigkeitswerten erstellen Sie einen **Datastream** vom Typ **Virtual Pin** auf der **Datastream**-Seite. Legen Sie den DATENTYP auf **Double** und MIN und MAX auf **0** und **100** fest. Setzen Sie auch die Einheiten auf **Percentage, %**.

Quickstart Template Cancel

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATA TYPE:

UNITS:

MIN: MAX: DECIMALS: DEFAULT VALUE:

+ ADVANCED SETTINGS

Cancel Save

Region: ny3 [Privacy Policy](#)

- Erstellen Sie anschließend einen **Datastream** vom Typ **Virtual Pin** zur Erfassung der Temperatur. Legen Sie den DATENTYP auf Double, MIN und MAX auf -30 und 50 und die Einheiten auf **Celsius, °C** fest.

Quickstart Template

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATA TYPE:

UNITS:

MIN: MAX: DECIMALS: DEFAULT VALUE:

Region: ny3 [Privacy Policy](#)

- Erstellen Sie zudem einen **Datastream** vom Typ **Virtual Pin** zur Erfassung der Lichtintensität. Nutzen Sie den Standarddatentyp - **Integer** mit den Werten MIN und MAX, die auf 0 und 1024 festgelegt sind.

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATA TYPE:

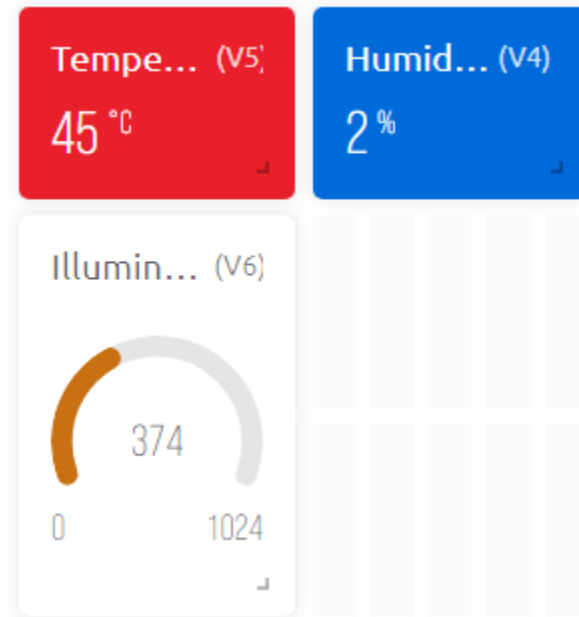
UNITS:

MIN: MAX: DEFAULT VALUE:

[+ ADVANCED SETTINGS](#)

Region: ny3 [Privacy Policy](#)

4. Navigieren Sie zur **Web Dashboard**-Seite, ziehen Sie zwei **Label**-Widgets und weisen Sie ihnen die Datenströme **V4** und **V5** zu. Ziehen Sie ein **Gauge**-Widget und setzen Sie den Datenstrom auf **V6**. In den Widget-Einstellungen können Sie **Farbwechsel basierend auf Wert** aktivieren und die passende Farbe wählen, um das Widget ansprechender und intuitiver zu gestalten.

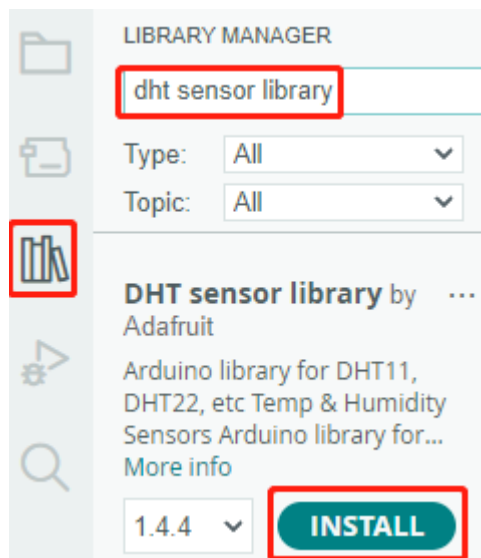


3. Den Code ausführen

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 sie über den **Library Manager** installieren.



2. Ersetzen Sie die `Template ID`, `Device Name` und `Auth Token` durch Ihre eigenen Angaben. Sie müssen auch das `ssid` und das `password` des von Ihnen verwendeten WLANs eingeben. Für detaillierte Anleitungen verweisen Sie bitte auf [1.4 Verbindung des R3-Boards mit Blynk herstellen](#).
3. Nach Auswahl des richtigen Boards und Ports klicken Sie auf die **Upload**-Taste.

4. Öffnen Sie den seriellen Monitor (Baudrate auf 115200 einstellen) und warten Sie, bis eine Meldung 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,"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

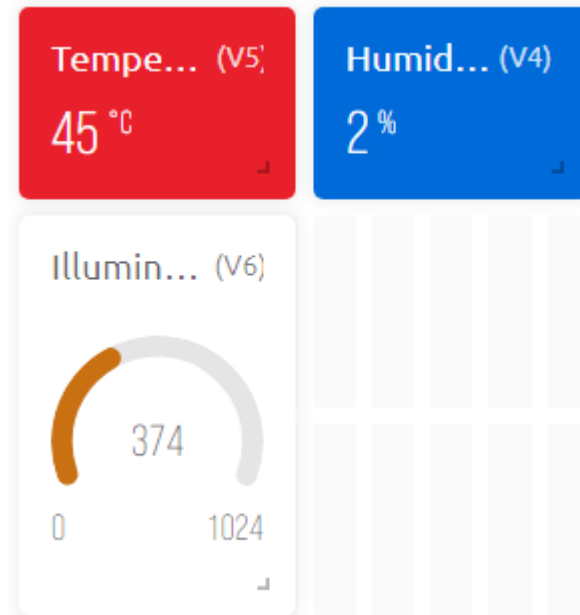
```

Bemerkung: Wenn beim Verbinden die Meldung ESP antwortet 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 auf GND setzen und dann trennen.
- Drücken Sie die Reset-Taste auf dem R3-Board.

Manchmal müssen Sie den oben beschriebenen Vorgang 3-5 Mal wiederholen, bitte haben Sie Geduld.

5. Nun sehen Sie die aktuelle Umgebungstemperatur, Luftfeuchtigkeit und Lichtintensität auf Blynk.



6. Wenn Sie Blynk auf mobilen Geräten verwenden möchten, verweisen Sie bitte auf *Wie verwendet man Blynk auf einem mobilen Gerät?*.



Wie funktioniert das?

Diese beiden Funktionen werden verwendet, um die Temperatur, Luftfeuchtigkeit und Lichtintensität des Raumes zu ermitteln.

```
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 Umgebungstemperatur, Luftfeuchtigkeit und Lichtintensität jede Sekunde abgerufen und an den Datenstrom auf 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);
}

```

7.6 6. Pflanzenüberwachung

Das Ziel dieses Projekts ist es, ein intelligentes Bewässerungssystem zu erstellen, das die aktuelle Temperatur, Luftfeuchtigkeit, Lichtintensität und Bodenfeuchtigkeit erkennt und diese auf Blynk anzeigt.

Sobald Sie den Schalter in der Blynk Cloud umlegen, 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:

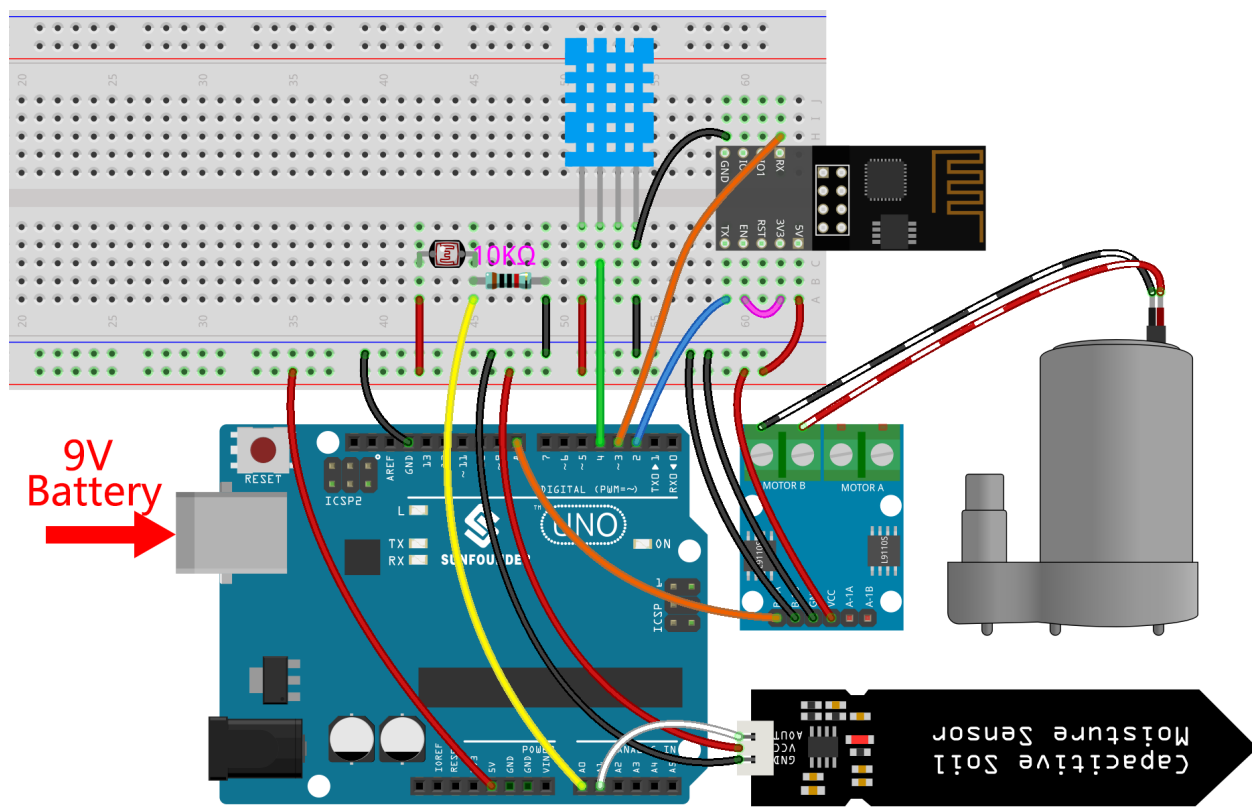
Name	ARTIKEL IN DIESEM SET	LINK
3 in 1 Starter Kit	380+	

Sie können diese auch separat über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>ESP8266 Modul</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Fotowiderstand</i>	
<i>DHT11 Feuchtigkeits- und Temperatursensor</i>	-
<i>Bodenfeuchtigkeitsmodul</i>	
<i>L9110 Motortreibermodul</i>	-
<i>Kreiselpumpe</i>	-

1. Schaltung aufbauen

Bemerkung: Das ESP8266-Modul benötigt einen hohen Strom, um eine stabile Betriebsumgebung zu gewährleisten. Stellen Sie also sicher, dass die 9V-Batterie angeschlossen ist.



2. Dashboard bearbeiten

1. Die in den vorherigen Projekten erstellten Datenströme müssen gespeichert werden und werden auch in diesem Projekt verwendet.
2. Zur Aufzeichnung der Bodenfeuchtigkeit erstellen Sie einen weiteren **Datastream** vom Typ **Virtual Pin** auf der **Datastream**-Seite. Stellen Sie den DATENTYP auf **Integer** und MIN und MAX auf **0** und **1024**.

Virtual Pin Datastream

NAME: ALIAS:

PIN: DATA TYPE:

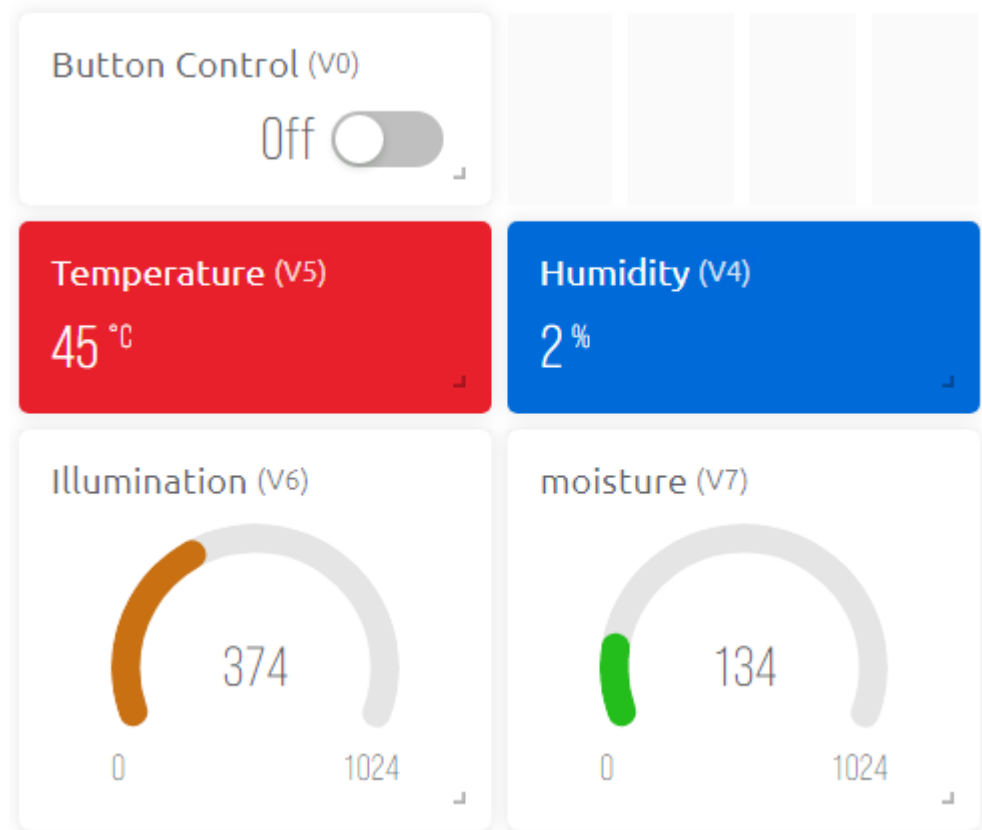
UNITS:

MIN: MAX: DEFAULT VALUE:

[+ ADVANCED SETTINGS](#) [Cancel](#) [Create](#)

Region: ny3 [Privacy Policy](#)

3. Gehen Sie nun zur **Wed Dashboard**-Seite, ziehen Sie 2 **Label**-Widgets und setzen Sie ihre Datenströme auf **V4** bzw. **V5**; ziehen Sie 2 **Gauge**-Widgets und setzen Sie ihre Datenströme auf **V6** und **V7**; ziehen Sie zuletzt ein **Switch**-Widget und setzen Sie dessen Datenstrom auf **V0**.

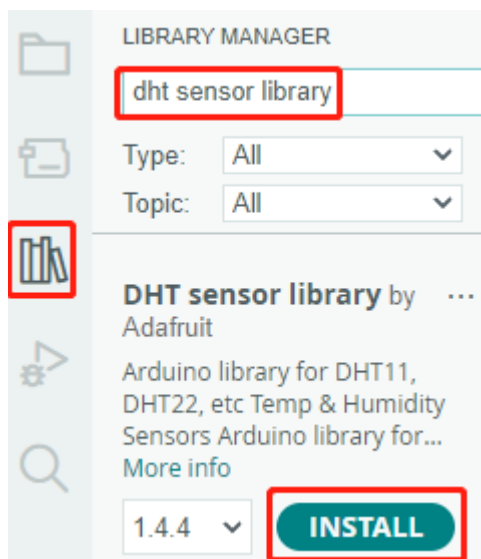


3. Code ausführen

Öffnen Sie die Datei `6.plant_monitoring.ino` unter dem 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 sie über den **Library Manager** installieren.



1. Ersetzen Sie die Template ID, den Gerätenamen und den Auth Token durch Ihre eigenen Angaben. Geben Sie auch die ssid und das Passwort des von Ihnen verwendeten WLANs ein. Für detaillierte Anleitungen verweisen Sie bitte auf [1.4 Verbindung des R3-Boards mit Blynk herstellen](#).
2. Nach Auswahl des richtigen Boards und Ports klicken Sie auf die Schaltfläche **Upload**.
3. Öffnen Sie den Serienmonitor (Baudrate auf 115200 einstellen) und warten Sie auf eine Aufforderung, wie z.B. 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,"1.1.1.1"
+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

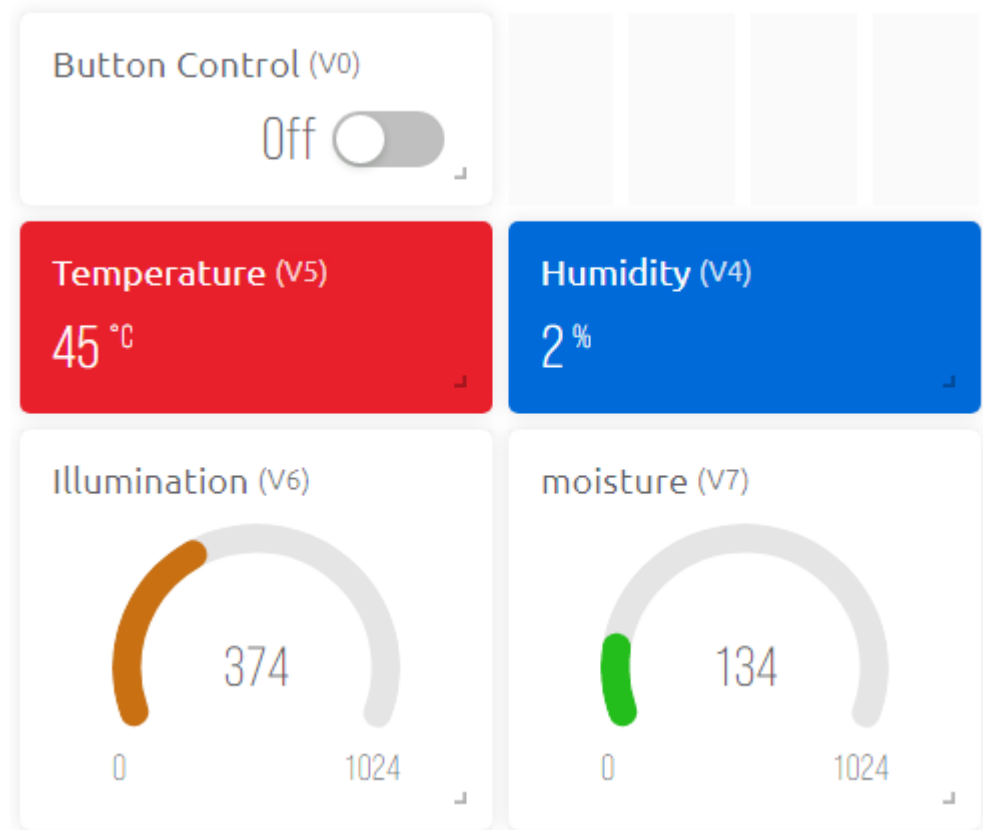
```

Bemerkung: Wenn die Meldung ESP antwortet nicht erscheint, wenn Sie sich verbinden, folgen Sie bitte diesen Schritten.

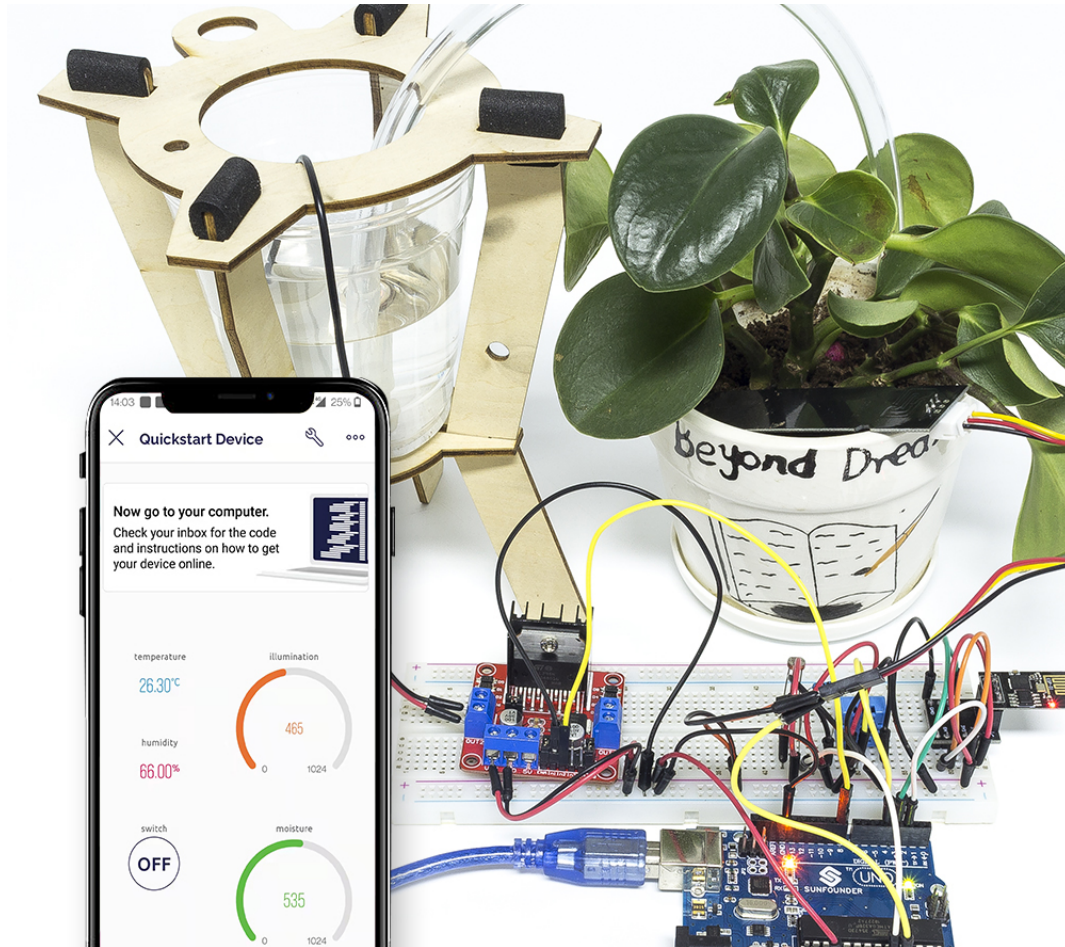
- Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.
- Setzen Sie das ESP8266-Modul zurück, indem Sie den Pin RST für 1 Sekunde mit GND verbinden und dann trennen.
- Drücken Sie den Reset-Knopf auf dem R3-Board.

Manchmal müssen Sie die oben genannte Operation 3-5 Mal wiederholen. Bitte haben Sie Geduld.

4. Zurück zu Blynk, Sie sehen die aktuelle Temperatur, Luftfeuchtigkeit, Lichtintensität und Bodenfeuchtigkeit. Sie können die Pumpe die Pflanzen bewässern lassen, falls notwendig, indem Sie das Button Control-Widget anklicken.



5. Wenn Sie Blynk auf mobilen Geräten verwenden möchten, verweisen Sie bitte auf *Wie verwendet man Blynk auf einem mobilen Gerät?*.



Wie funktioniert das?

Dieses BLYNK_WRITE bewirkt, dass Blynks **Switch**-Widget die Pumpe startet, wenn es EIN ist und sie ausschaltet, wenn es AUS 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 erhalten.

```
int readMoisture(){
  return analogRead(moisturePin);
}

int readLight(){
  return analogRead(lightPin);
}
```

(Fortsetzung auf der nächsten 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 die Umgebungstemperatur, Luftfeuchtigkeit, Lichtintensität und Bodenfeuchtigkeit jede Sekunde abgerufen und an den Datenstrom auf der **Blynk Cloud** gesendet, von dem 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);
}

```

7.7 7. Strombegrenzendes Tor

In manchen Situationen, wie zum Beispiel auf Parkplätzen, ist eine Mengensteuerung erforderlich.

Hier erstellen wir ein intelligentes Tor: Ein Servo dient als Tor und ein IR-Hinderniserkennungsdetektor wird davor platziert; wird ein Objekt (wie ein Auto) erkannt, öffnet sich das Tor und die Anzahl erhöht sich um 1. Die Anzahl wird auf einer 7-Segment-Anzeige dargestellt und auch auf die Blynk Cloud hochgeladen, damit 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 folgende Komponenten.

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

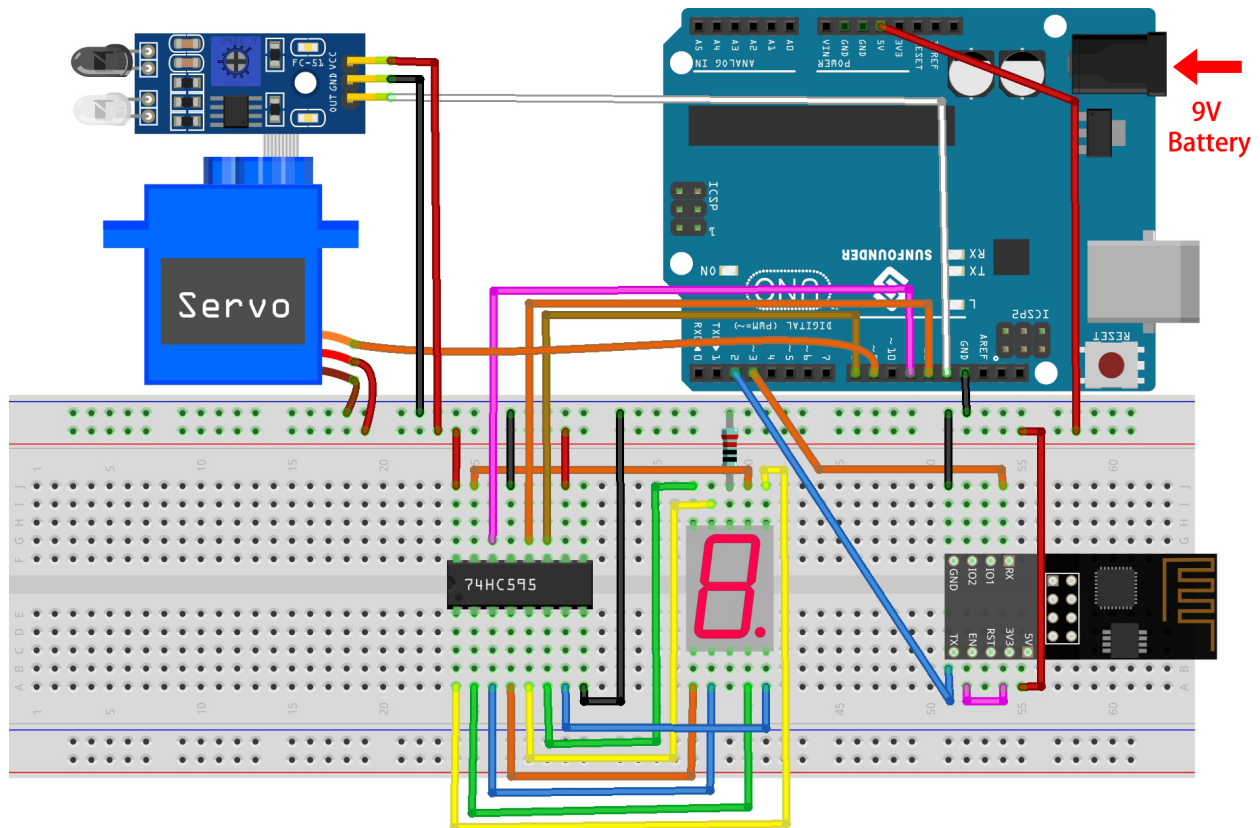
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</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. Bauen Sie den Schaltkreis

Bemerkung: Das ESP8266-Modul benötigt einen hohen Strom, um eine stabile Betriebsumgebung zu gewährleisten, also stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.



2. Dashboard bearbeiten

1. Um die Anzahl aufzuzeichnen, erstellen Sie einen **Datastream** vom Typ **Virtual Pin** auf der **Datastream**-Seite. Setzen Sie den DATENTYP auf Integer und MIN und MAX auf 0 und 10.

Virtual Pin Datastream

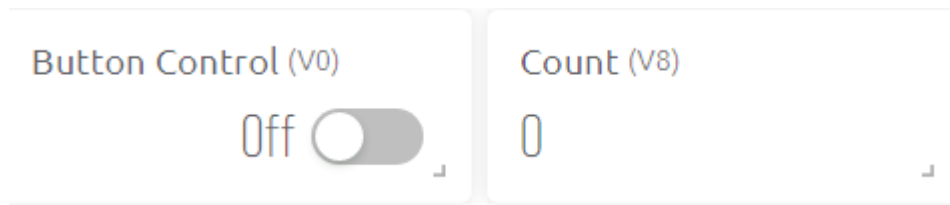
NAME: ALIAS:

PIN: DATA TYPE:

UNITS:

MIN: MAX: DEFAULT VALUE:

- Gehen Sie nun zur **Wed Dashboard**-Seite, ziehen Sie ein **Switch**-Widget, um dessen Datenstrom auf **V0** und ein **Label**-Widget, um dessen Datenstrom auf **V8** zu setzen.



3. Den Code ausführen

- 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**.
- Ersetzen Sie die `Template ID`, den `Device Name` und den `Auth Token` durch Ihre eigenen. Geben Sie auch die `ssid` und das `password` des von Ihnen verwendeten WLANs ein. Für detaillierte Anleitungen verweisen Sie bitte auf [1.4 Verbindung des R3-Boards mit Blynk herstellen](#).
- Nach Auswahl des richtigen Boards und Ports klicken Sie auf die Schaltfläche **Upload**.
- Öffnen Sie den Serienmonitor (Baudrate auf 115200 einstellen) und warten Sie auf eine Aufforderung, wie z.B. 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,"1.1.1.1"
+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

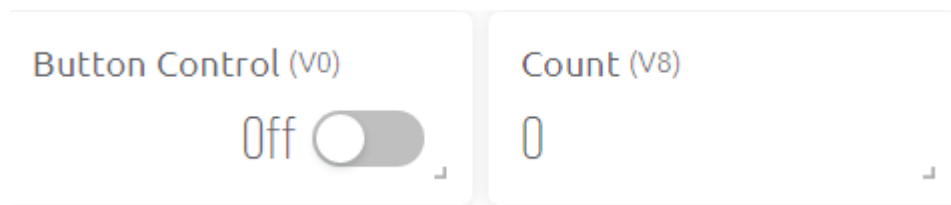
```

Bemerkung: Wenn die Meldung ESP antwortet nicht erscheint, wenn Sie sich verbinden, folgen Sie bitte diesen Schritten.

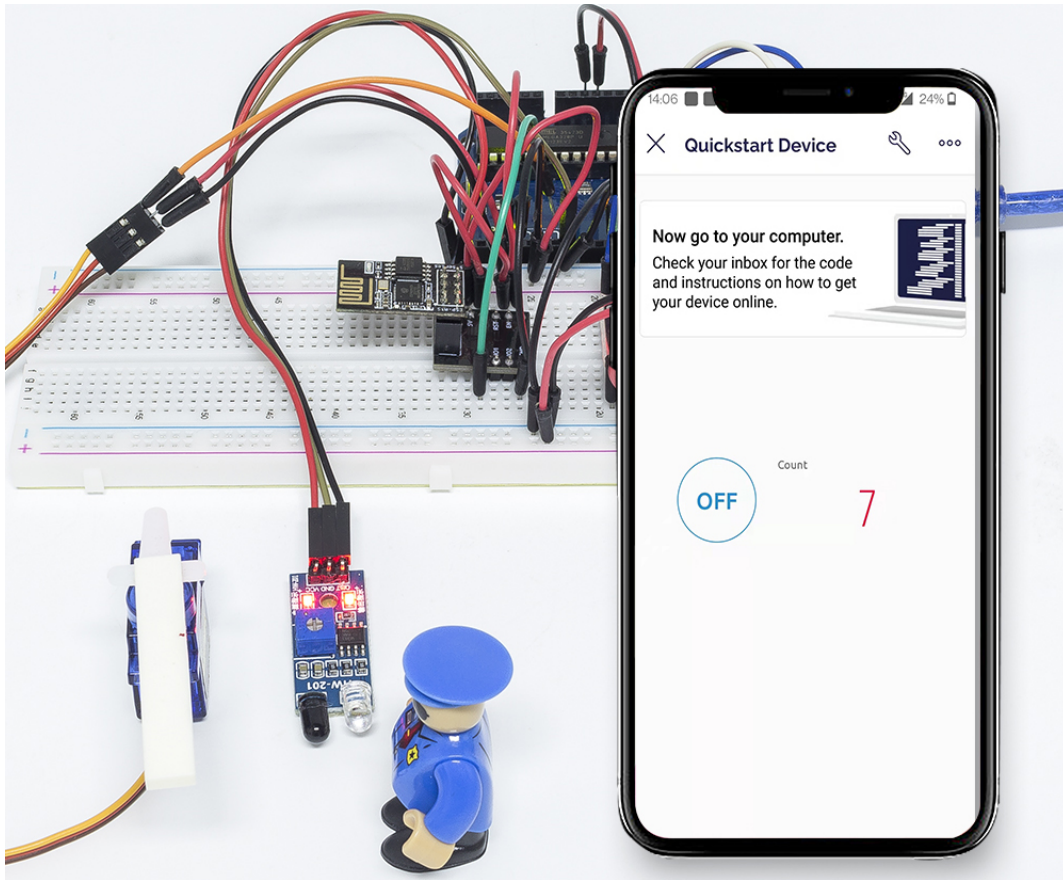
- Stellen Sie sicher, dass die 9V-Batterie angeschlossen ist.
- Setzen Sie das ESP8266-Modul zurück, indem Sie den Pin RST für 1 Sekunde mit GND verbinden und dann trennen.
- Drücken Sie die Reset-Taste auf dem R3-Board.

Manchmal müssen Sie die obige Operation 3-5 Mal wiederholen, bitte haben Sie Geduld.

5. Klicken Sie jetzt auf das Button Control-Widget in Blynk, um das intelligente Türsystem zu aktivieren. Wenn das IR-Hinderniserkennungsmodul ein Hindernis erkennt, öffnet sich das Tor und die 7-Segment-Anzeige sowie das Count-Widget in Blynk werden um 1 erhöht.



6. Wenn Sie Blynk auf mobilen Geräten verwenden möchten, beachten Sie bitte *Wie verwendet man Blynk auf einem mobilen Gerät?*.



Wie funktioniert das?

Die Funktion `BLYNK_WRITE(V0)` erhält den Status des **Switch**-Widgets und weist ihn der Variablen `doorFlag` zu, die verwendet wird, um zu bestimmen, ob das intelligente Torsystem aktiviert ist oder nicht.

```
BLYNK_WRITE(V0)
{
    doorFlag = param.asInt(); // Aktiviere das Tor
}
```

Im Blynk-Timer wird `doorFlag` jede Sekunde geprüft, und wenn es aktiviert ist, wird die Hauptfunktion des Tores ausgeführt.

```
void myTimerEvent()
{
    if (doorFlag)
    {
        channelEntrance();
    }
}
```

Die Hauptfunktion des Tores ist `channelEntrance()`. Wenn ein Objekt sich dem Tor nähert (der Sensor erkennt ein Hindernis), wird `count` um 1 erhöht. Schreiben Sie `count` in den Datastream `V8` der Blynk Cloud und die 7-Segment-Anzeige auf der Schaltung und öffnen Sie das Tor. Wenn das Objekt von vorhanden zu nicht vorhanden wechselt, d.h. das Objekt hat das Tor betreten, schließen Sie das Tor.

```

void channelEntrance()
{
    int currentState = digitalRead(irPin); // 0:Hindernis 1:kein Hindernis
    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) dient dazu, die 7-Segment-Anzeige den Wert anzeigen zu lassen.

```

void showNumber(int num)
{
    digitalWrite(STcp, LOW); //ST_CP erden und so lange auf LOW halten, wie Daten_
    ↳ übertragen werden
    shiftOut(DS, SHcp, MSBFIRST, dataArray[num]);
    digitalWrite(STcp, HIGH); //ST_CP hochziehen, um die Daten zu speichern
}

```

Die Funktion operateGate(bool openGate) öffnet das Tor langsam, wenn die Referenz True ist, und schließt das Tor langsam, wenn die Referenz False ist.

```

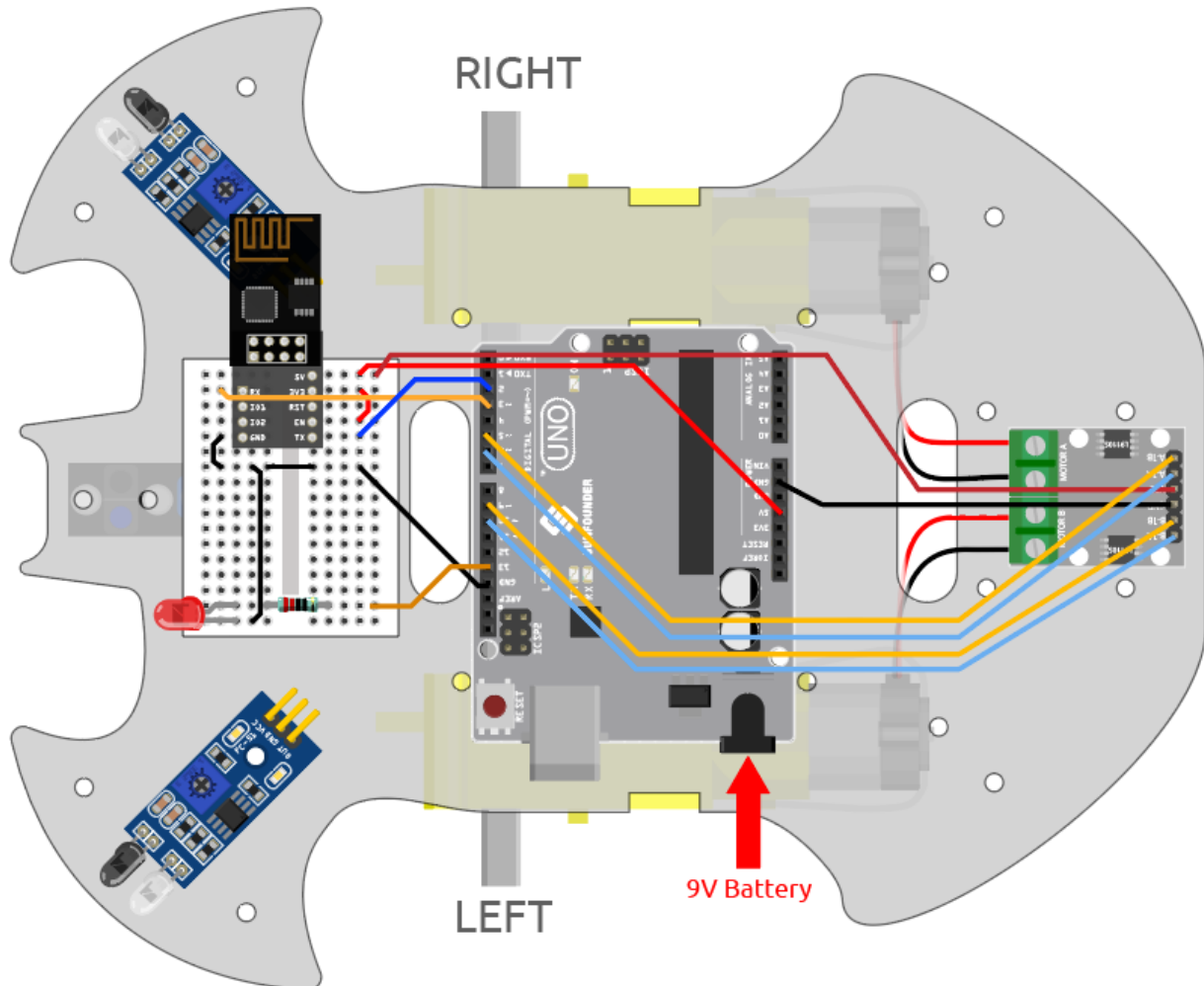
void operateGate(bool openGate) {
    if (openGate == true)
    {
        // open gate
        while (angle <= 90) {
            angle++;
            myservo.write(angle);
            delay(5);
        }
    } else {
        // close gate
        while (angle >= 0){
            angle--;
            myservo.write(angle);
            delay(5);
        }
    }
}

```


7.8 8. IoT Auto

Für dieses Projekt haben wir die Blynk-APP auf dem Handy verwendet, um das Auto zu steuern. Sie müssen jedoch *Auto Projekte* konsultieren, um das Auto zusammenzubauen und ein grundlegendes Verständnis dafür zu erhalten. Im Zeitalter der Verbreitung des 5G-Netzwerks könnte dieser Modus zu einer der Hauptproduktionsmethoden in vielen Branchen werden. Lassen Sie uns also dieses Spiel im Voraus erleben.

1. Schaltkreis aufbauen



2. Dashboard bearbeiten

In Blynk auf dem Handy können keine Datastreams bearbeitet werden. Daher müssen wir diese Schritte immer noch auf der Web-Seite durchführen.

1. Erstellen Sie auf der **Datastream**-Seite einen **Datastream** vom Typ **Virtual Pin**, um den X-Achsen-Wert des Joysticks zu erfassen. Setzen Sie NAME auf Xvalue, DATENTYP auf Integer und MIN und MAX auf -10 und 10.

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 **Datastream** vom Typ **Virtual Pin**, um den Y-Achsen-Wert des Joysticks zu erfassen. Setzen Sie NAME auf Yvalue, DATENTYP auf Integer, MIN und MAX auf -10 und 10.

Virtual Pin Datastream

NAME: Yvalue ALIAS: Yvalue

PIN: V10 DATA TYPE: Integer

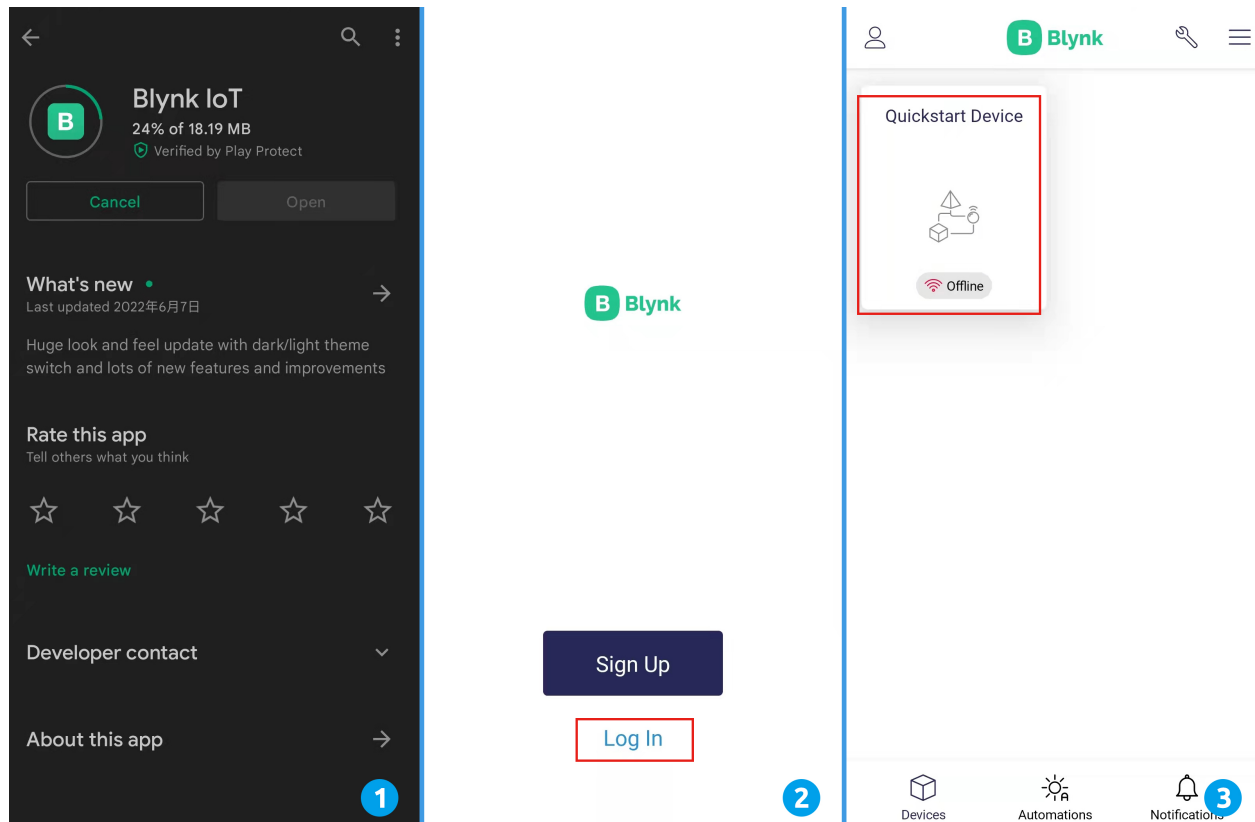
UNITS: None

MIN: -10 MAX: 10 DEFAULT VALUE: 0

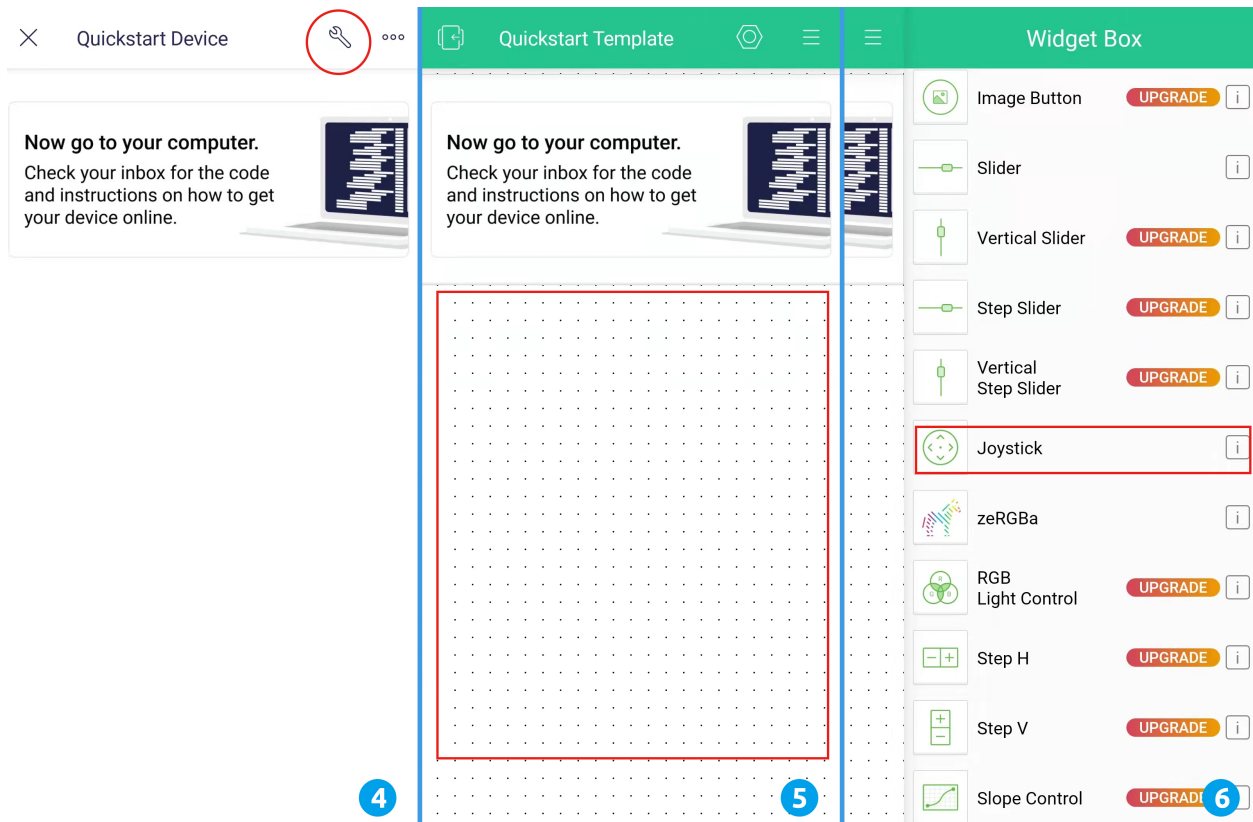
+ ADVANCED SETTINGS Cancel Create

Führen Sie dann die folgenden Schritte auf Ihrem Handy durch.

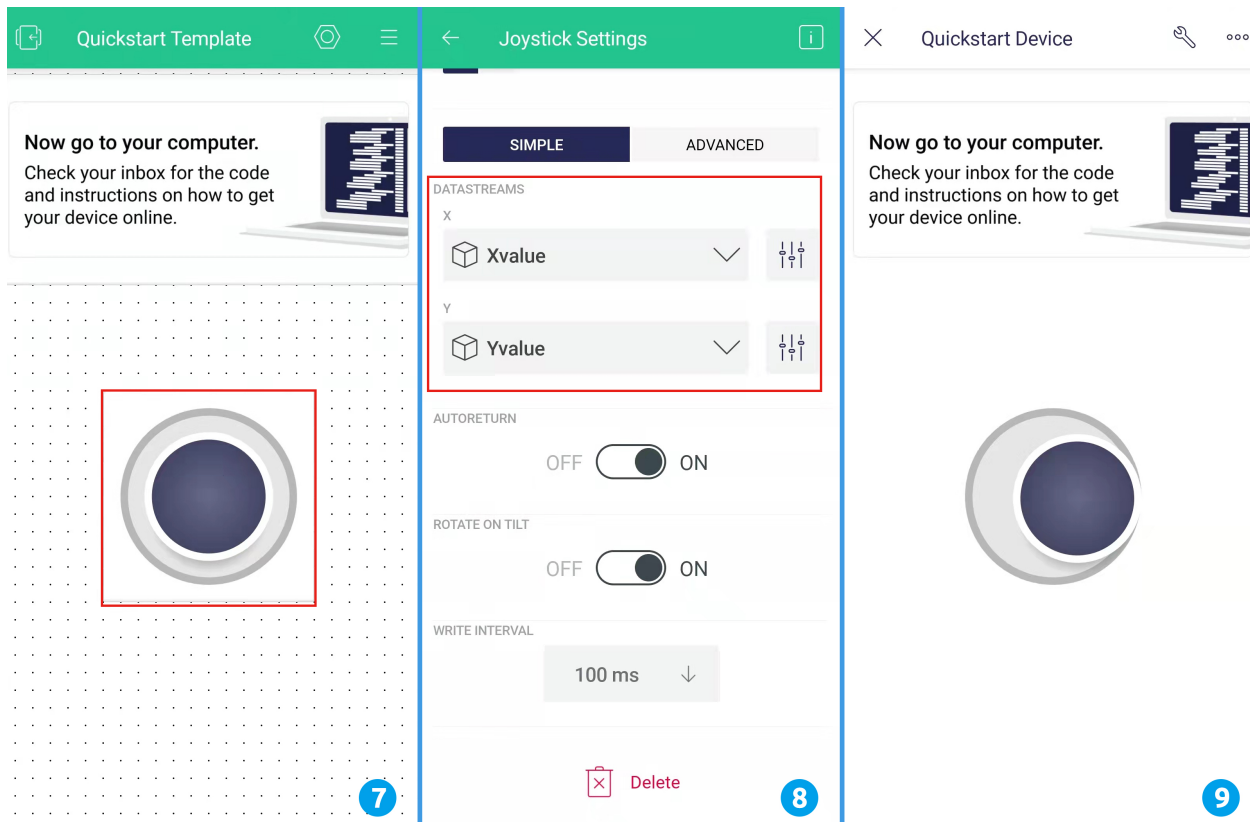
1. Suchen Sie in GOOGLE Play oder im APP Store nach „Blynk IoT“ (nicht Blynk(legacy)) und laden Sie es herunter.
2. Nach dem Öffnen der APP melden Sie sich an. Dieses Konto sollte mit dem Konto auf dem Web-Client übereinstimmen.
3. Gehen Sie dann zum Dashboard (wenn Sie keines haben, erstellen Sie eines), und Sie werden sehen, dass das Dashboard für Mobilgeräte und Web voneinander unabhängig sind.



4. Klicken Sie auf das Bearbeitungssymbol.
5. Klicken Sie auf den leeren Bereich.
6. Wählen Sie ein Joystick-Widget aus.



7. Jetzt sehen Sie ein Joystick-Widget im leeren Bereich, klicken Sie darauf.
8. Die Joystick-Einstellungen erscheinen, wählen Sie die von Ihnen in den Datastreams eingestellten Xvalue und Yvalue.
9. Kehren Sie zur Dashboard-Seite zurück, und Sie können 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`, `Device Name` und `Auth Token` durch Ihre eigenen Angaben. Geben Sie auch die `ssid` und das `password` des von Ihnen verwendeten WLANs ein. Für detaillierte Anleitungen konsultieren Sie bitte [1.4 Verbindung des R3-Boards mit Blynk herstellen](#).
3. Wählen Sie das richtige 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, bis eine Meldung 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,"1.1.1.175"
+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

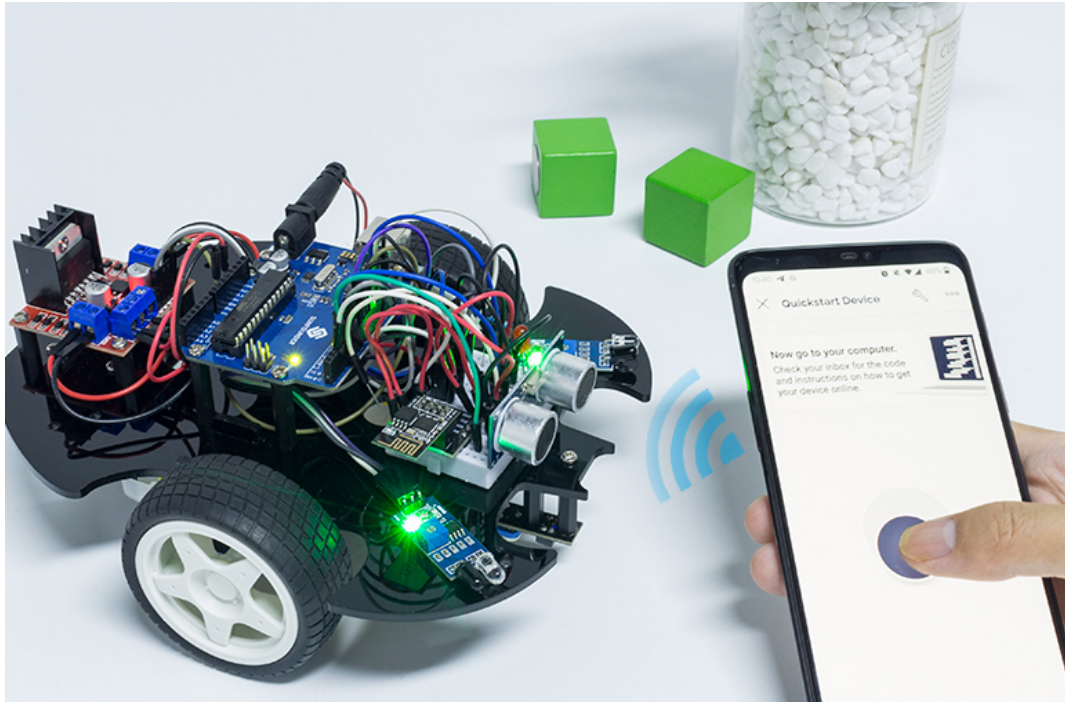
```

Bemerkung: Wenn die Meldung ESP reagiert nicht 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 1 Sekunde lang auf GND verbinden und dann abziehen.
- Drücken Sie die Reset-Taste auf dem R3-Board.

Manchmal müssen Sie die obige Operation 3-5 Mal wiederholen, bitte haben Sie Geduld.

5. Ziehen Sie jetzt das USB-Kabel ab und versorgen Sie den Wagen nur mit einer 9V-Batterie. Warten Sie, bis die LED aufleuchtet, was bedeutet, dass das Auto mit Blynk verbunden ist.
6. Öffnen Sie Blynk auf Ihrem Handy, und Sie können das Joystick-Widget verwenden, um die Bewegung des Autos zu steuern.



Wie funktioniert das?

Diese Funktionen dienen zur Steuerung der Bewegung des Autos.

```
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();
}
```

In loop() lässt man das Auto verschiedene Aktionen basierend auf Xvalue und Yvalue ausführen.

```
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();  
}
```

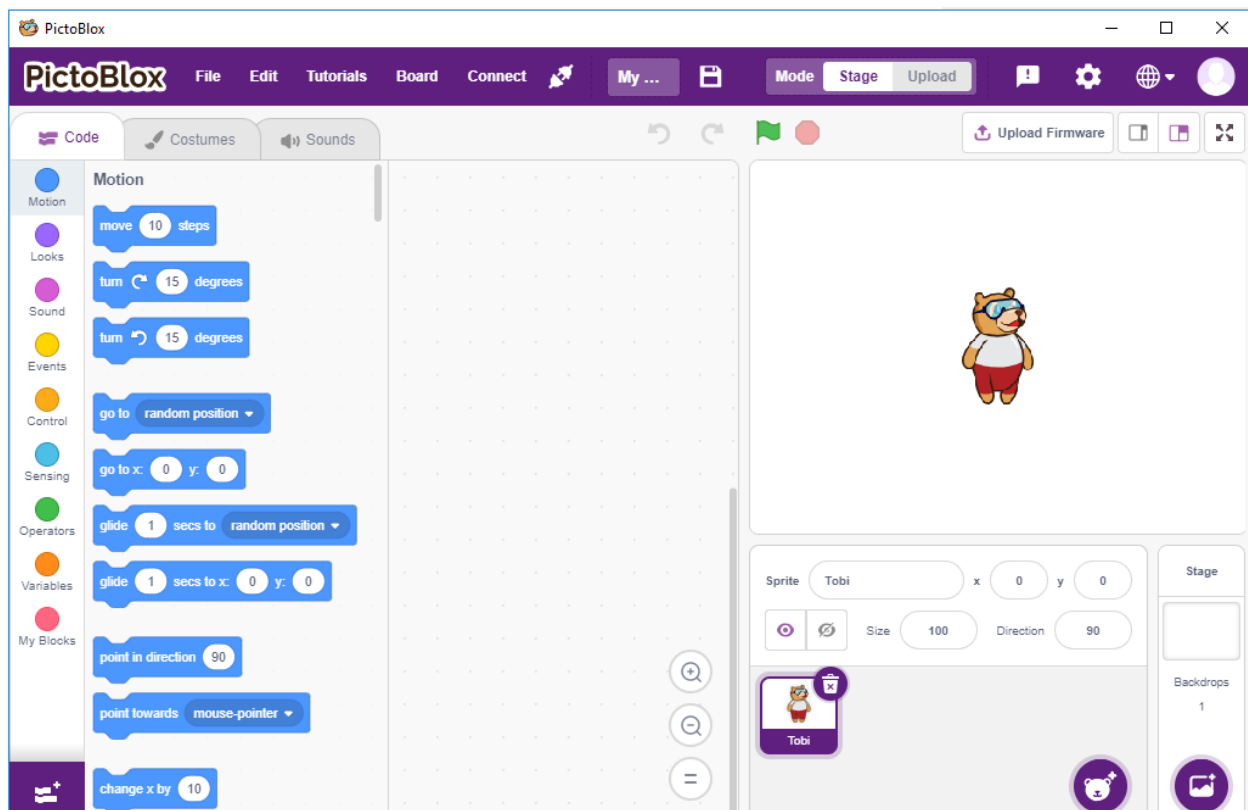
Ebenso fügen Sie eine Netzwerkstatus-Bestimmung zu `loop()` hinzu, um eine LED aufzuleuchten, wenn sie mit Blynk Cloud verbunden ist.

```
if (!Blynk.connected()) {  
    digitalWrite(ledPin, LOW);  
    Serial.print("offline!");  
    bool result = Blynk.connect();  
    Serial.println(result);  
} else {  
    digitalWrite(ledPin, HIGH);  
}
```


Mit Scratch spielen

Neben der Programmierung in der Arduino IDE können wir auch grafische Programmierung nutzen.

Wir empfehlen hier die Programmierung mit Scratch. Allerdings ist das offizielle Scratch derzeit nur mit Raspberry Pi kompatibel. Deshalb haben wir uns mit dem Unternehmen STEMPedia zusammengeschlossen, das eine auf Scratch 3 basierende grafische Programmiersoftware für Arduino-Platinen (Uno, Mega2560 und Nano) entwickelt hat: [PictoBlox](#).



Es behält die Grundfunktionen von Scratch 3 bei, fügt aber auch Steuerplatinen hinzu, wie Arduino Uno, Mega, Nano,

ESP32, Microbit und STEAMPedia-Eigenbau-Hauptplatinen. Damit können externe Sensoren und Roboter verwendet werden, um die Sprites auf der Bühne zu steuern, mit starken Hardware-Interaktionsfähigkeiten.

Darüber hinaus verfügt es über KI und maschinelles Lernen. Selbst wenn Sie nur wenig Programmiererfahrung haben, können Sie diese populären und hochmodernen Technologien erlernen und nutzen.

Sie müssen lediglich die Scratch-Programmierblöcke per Drag & Drop verschieben und coole Spiele, Animationen, interaktive Projekte erstellen und sogar Roboter nach Ihren Wünschen steuern!

Jetzt beginnen wir die Entdeckungsreise!

1. Einführung

8.1 1.1 PictoBlox installieren

Klicken Sie auf diesen Link: <https://thetempedia.com/product/pictoblox/download-pictoblox/>, wählen Sie das passende Betriebssystem (Windows, macOS, Linux) und folgen Sie den Anweisungen zur Installation.

Download PictoBlox
CHOOSE YOUR DEVICE

Click here to choose difference systems

Windows macOS Linux Android

Windows Installer (.exe)

STEP 1: Download the Pictoblox Installer (.exe) for Windows 7 and above ([Release Notes](#)).

WINDOWS INSTALLER 64-BIT V4.1.0

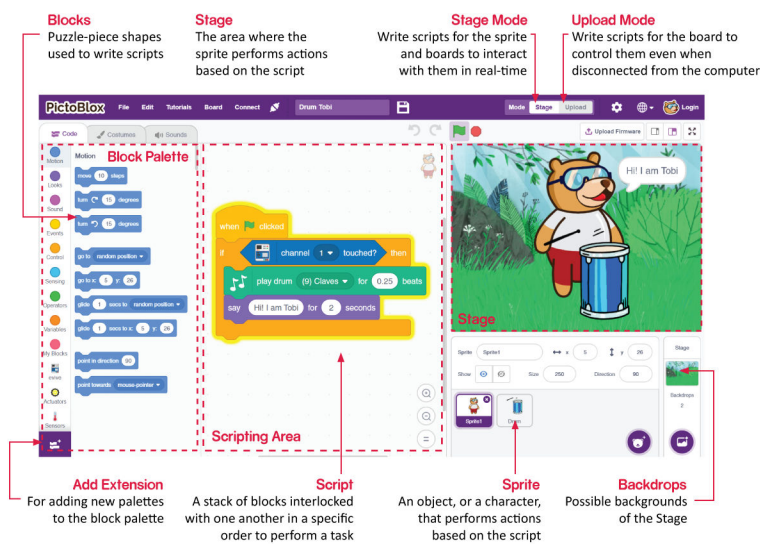
WINDOWS INSTALLER 32-BIT V4.1.0

STEP 2: Run the .exe file.

Some of the device gives the following popup. You don't have to worry, this software is harmless. Click on **More info** and then click on **Run anyway**.

STEP 3: Rest of the installation is straight forward, you can follow the popup and check on the option appropriate for your need.

8.2 1.2 Einführung in die Benutzeroberfläche



Sprites

Ein Sprite ist ein Objekt oder Charakter, der verschiedene Aktionen in einem Projekt ausführt. Es versteht und befolgt die ihm gegebenen Befehle. Jeder Sprite verfügt über spezifische Kostüme und Töne, die Sie ebenfalls anpassen können.

Bühne

Die Bühne ist der Bereich, in dem der Sprite Aktionen in Hintergründen gemäß Ihrem Programm ausführt.

Hintergründe

Hintergründe dienen zur Dekoration der Bühne. Sie können einen Hintergrund aus PictoBlox auswählen, selbst einen zeichnen oder ein Bild von Ihrem Computer hochladen.

Skriptbereich

Ein Skript ist ein Programm oder ein Code im PictoBlox/Scratch-Jargon. Es handelt sich um einen Satz von „Blöcken“, die in einer bestimmten Reihenfolge angeordnet sind, um eine Aufgabe oder eine Reihe von Aufgaben auszuführen. Sie können mehrere Skripte schreiben, die alle gleichzeitig laufen können. Skripte können Sie nur im zentralen Skriptbereich des Bildschirms schreiben.

Blöcke

Blöcke sind wie Teile eines Puzzles, die verwendet werden, um Programme zu schreiben, indem sie einfach im Skriptbereich zusammengestapelt werden. Die Verwendung von Blöcken zum Schreiben von Code kann die Programmierung erleichtern und die Wahrscheinlichkeit von Fehlern verringern.

Blockpaletten

Die Blockpaletten befinden sich im linken Bereich und sind nach ihren Funktionen benannt, wie Bewegung, Ton und Steuerung. Jede Palette enthält verschiedene Blöcke. Zum Beispiel steuern die Blöcke in der Bewegungspalette die Bewegung der Sprites, und die Blöcke in der Steuerungspalette steuern die Arbeit des Skripts unter bestimmten Bedingungen.

Es gibt andere Arten von Blockpaletten, die über die Schaltfläche **Erweiterung hinzufügen** unten links geladen werden können.

Modi

Im Gegensatz zu Scratch verfügt PictoBlox über zwei Modi:

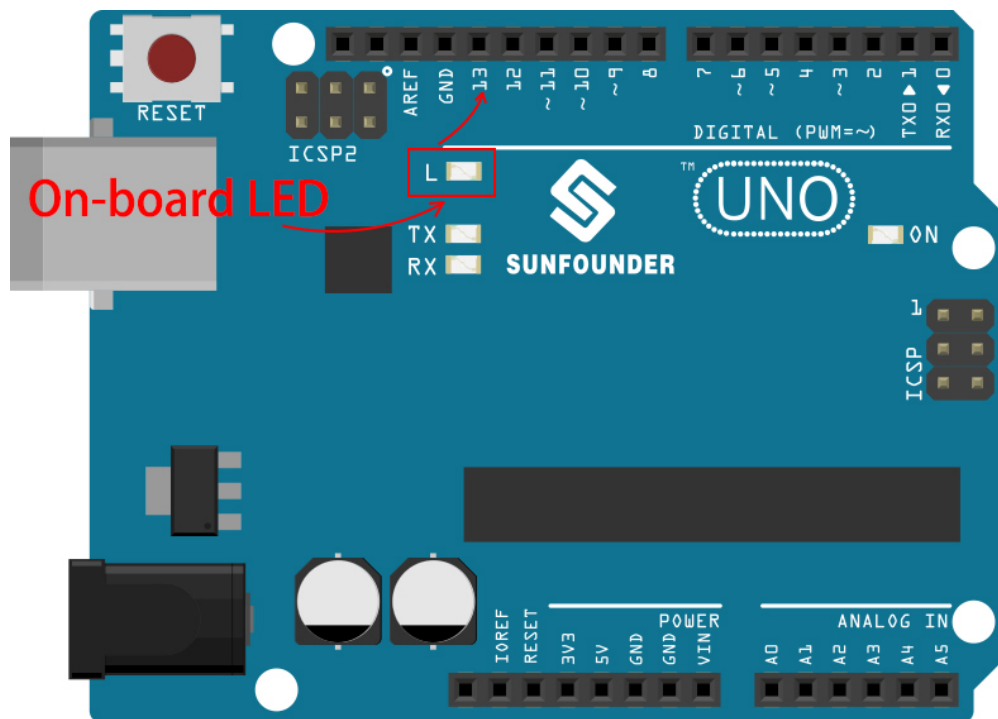
- *Bühnenmodus*: In diesem Modus können Sie Skripte für den Sprite und Platinen schreiben, um in Echtzeit mit Sprites zu interagieren. Wenn Sie die Platine von PictoBlox trennen, können Sie nicht mehr interagieren.
- *Hochlade-Modus*: Dieser Modus ermöglicht es Ihnen, Skripte zu schreiben und auf die Platine hochzuladen, sodass Sie sie auch verwenden können, wenn sie nicht mit Ihrem Computer verbunden ist, beispielsweise, um ein Skript für sich bewegende Roboter hochzuladen.

Für weitere Informationen besuchen Sie bitte: <https://thestempedia.com/tutorials/getting-started-pictoblox>

8.3 1.3 Schnellanleitung zu PictoBlox

Lernen wir nun, wie man PictoBlox in zwei Modi verwendet.

Es gibt auch eine On-Board-LED, die an Pin 13 auf dem R3-Board angeschlossen ist. Wir werden lernen, wie man diese LED in 2 verschiedenen Modi blinken lässt.

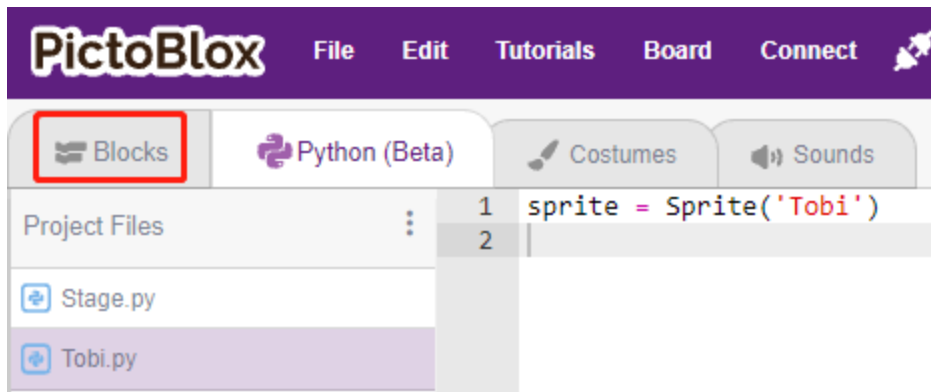


8.3.1 Bühnenmodus

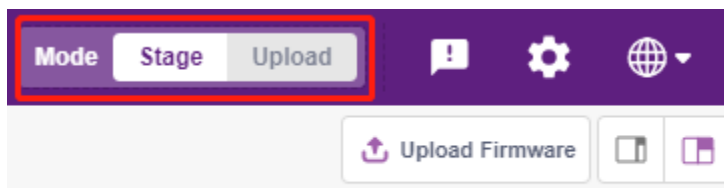
1. Verbindung mit dem Arduino-Board

Schließen Sie Ihr Arduino-Board mit einem USB-Kabel an den Computer an. In der Regel wird der Computer Ihr Board automatisch erkennen und schließlich einen COM-Port zuweisen.

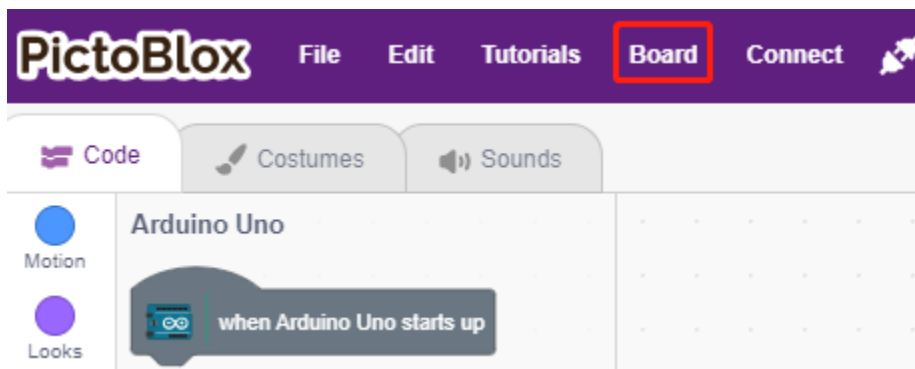
Öffnen Sie PictoBlox. Die Python-Programmierschnittstelle wird standardmäßig geöffnet. Wir müssen zur Blocks-Schnittstelle wechseln.



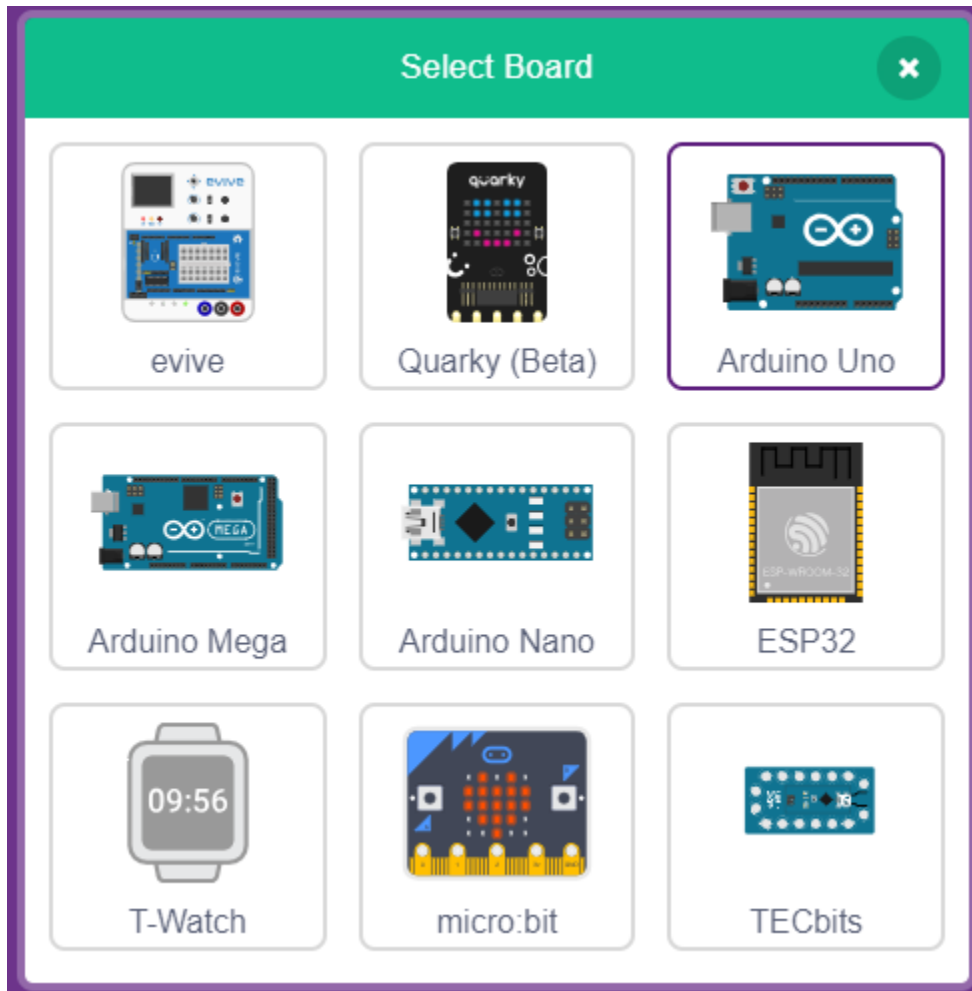
Dann sehen Sie in der oberen rechten Ecke die Option zum Wechseln des Modus. Standardmäßig ist der Bühnenmodus aktiviert, auf dem Tobi auf der Bühne steht.



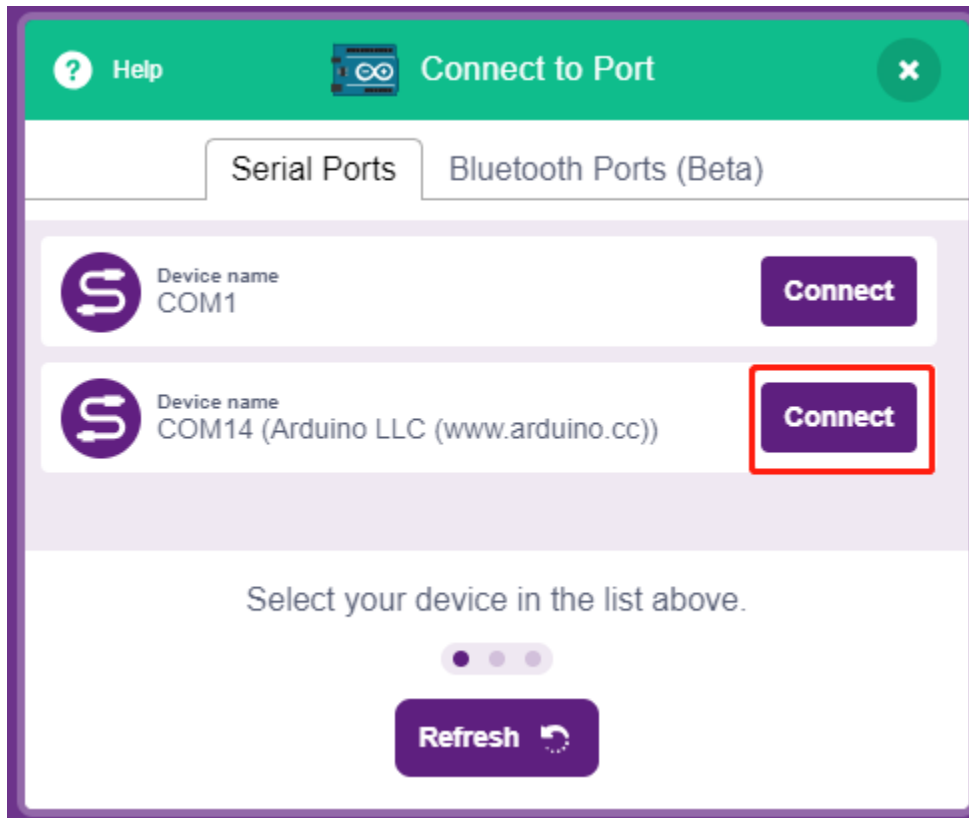
Klicken Sie in der oberen rechten Navigationsleiste auf **Board**, um das Board auszuwählen.



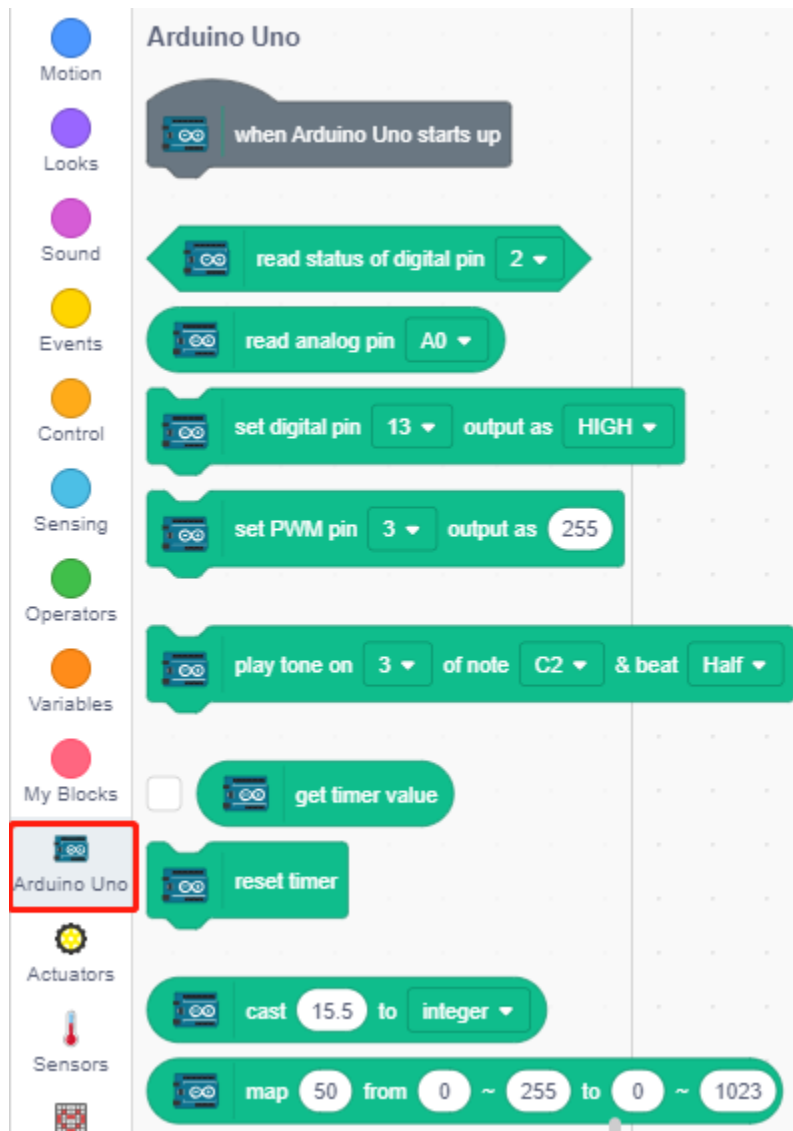
Wählen Sie zum Beispiel **Arduino Uno**.



Ein Verbindungsfenster wird geöffnet, in dem Sie den zu verbindenden Port auswählen können. Kehren Sie zur Startseite zurück, wenn die Verbindung hergestellt ist. Wenn Sie die Verbindung während des Gebrauchs unterbrechen, können Sie auch auf **Connect** klicken, um erneut eine Verbindung herzustellen.



Gleichzeitig erscheinen im **Block Palette** palettenbezogene Elemente, wie Arduino Uno, Aktuatoren usw.

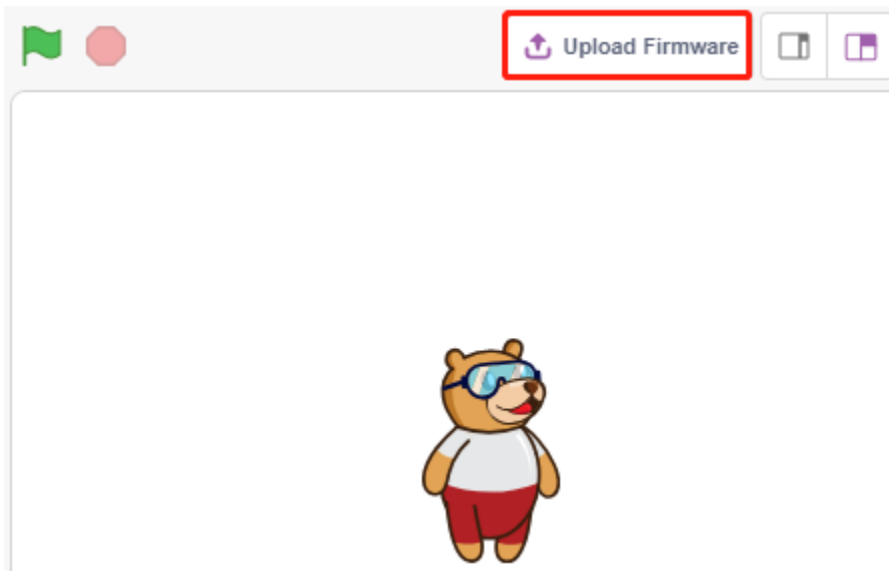


2. Firmware hochladen

Da wir im Bühnenmodus arbeiten möchten, müssen wir die Firmware auf das Board hochladen. Dies gewährleistet die Echtzeitkommunikation zwischen dem Board und dem Computer. Das Hochladen der Firmware ist ein einmaliger Vorgang. Klicken Sie dazu auf die Schaltfläche Firmware hochladen.

Nach einer kurzen Wartezeit wird die Erfolgsmeldung zum Hochladen angezeigt.

Bemerkung: Wenn Sie dieses Arduino-Board zum ersten Mal in PictoBlox verwenden oder wenn dieses Arduino zuvor mit der Arduino IDE hochgeladen wurde, müssen Sie auf **Upload Firmware** tippen, bevor Sie es verwenden können.

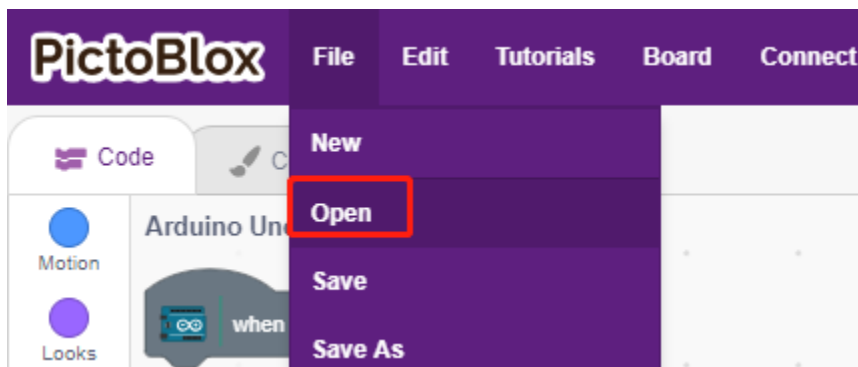


3. Programmierung

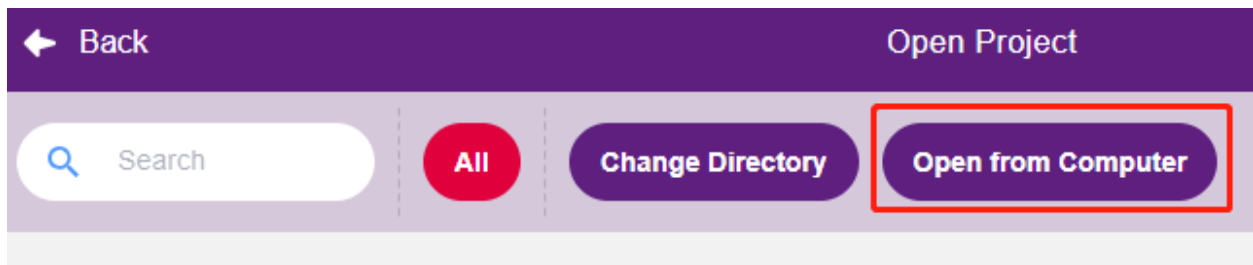
- Skript direkt öffnen und ausführen

Natürlich können Sie die Skripte direkt öffnen und ausführen, aber laden Sie sie bitte zuerst von [github](#) herunter.

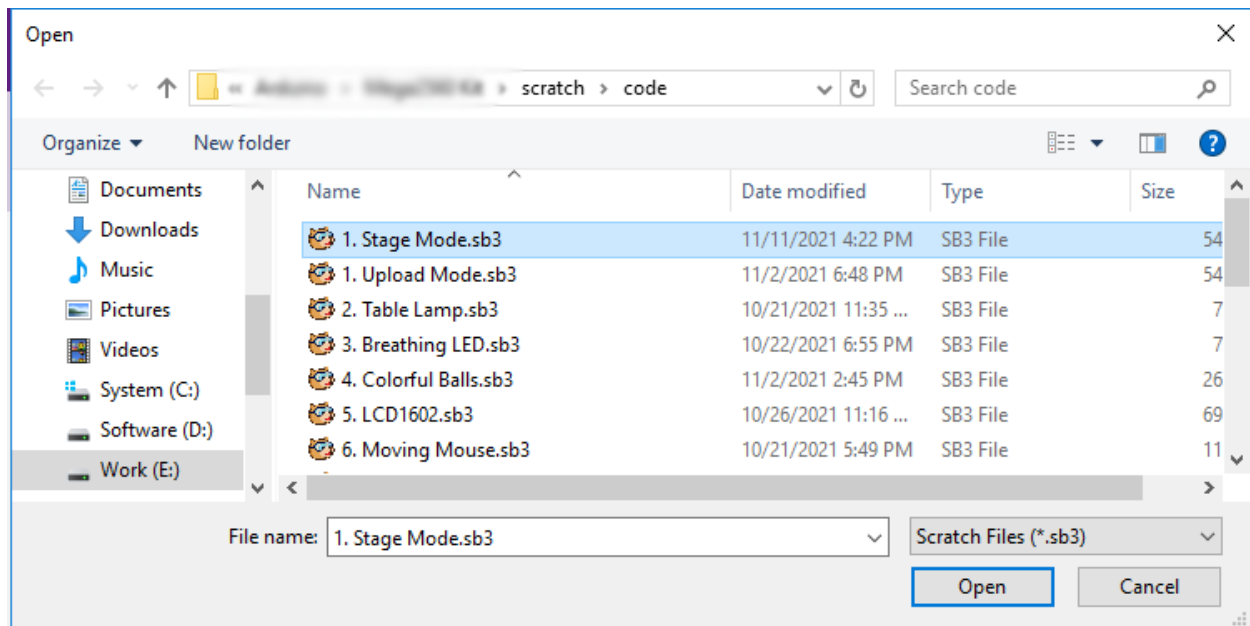
Sie können auf **File** in der oberen rechten Ecke klicken und dann **Open** auswählen.



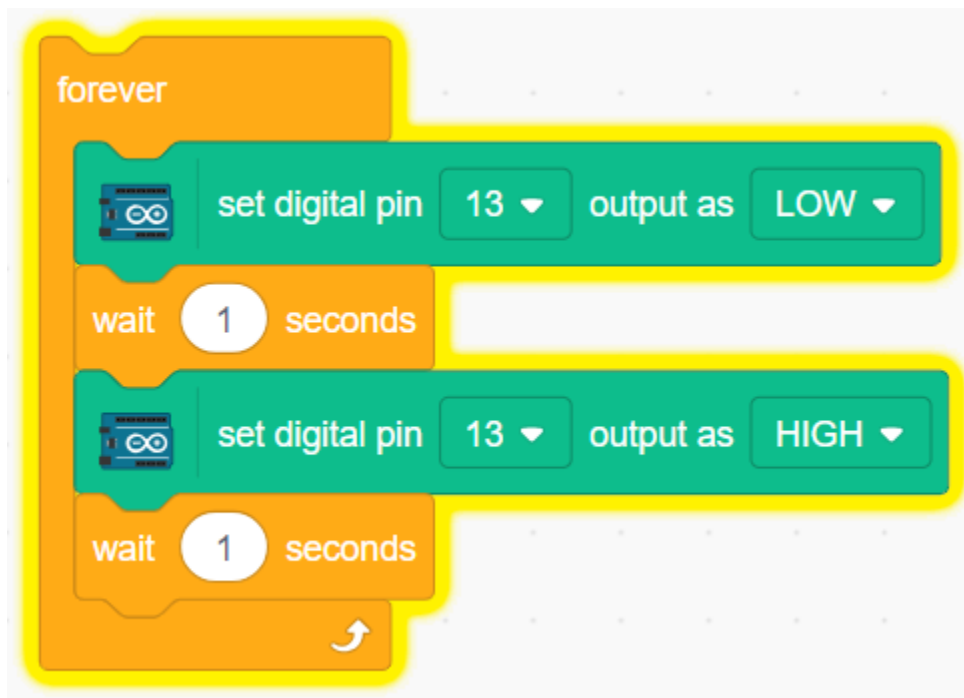
Wählen Sie **Open from Computer**.



Navigieren Sie dann zu `3in1-kit\scratch_project\code` und öffnen Sie **1. Stage Mode.sb3**. Stellen Sie sicher, dass Sie den benötigten Code von [github](#) heruntergeladen haben.



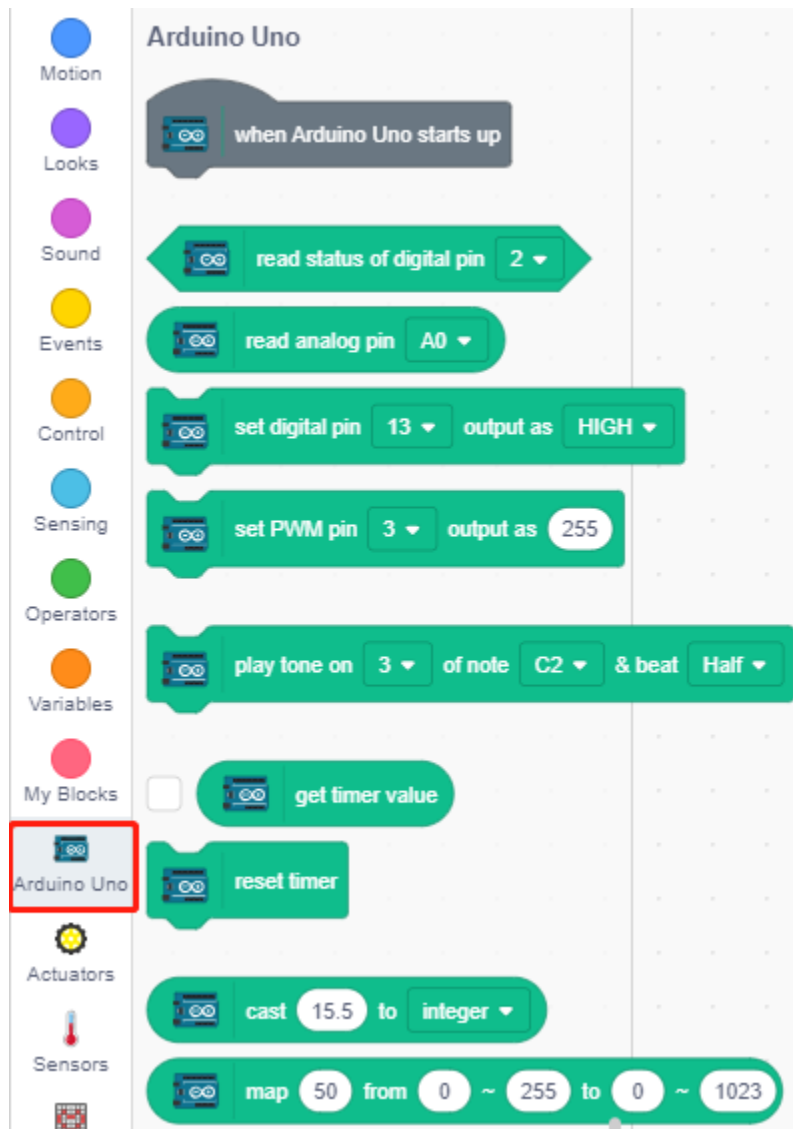
Klicken Sie direkt auf das Skript, um es auszuführen. Bei manchen Projekten klicken Sie auf die grüne Flagge oder auf das Sprite.



- Schritt-für-Schritt-Programmierung

Sie können das Skript auch Schritt für Schritt schreiben, indem Sie diesen Anweisungen folgen.

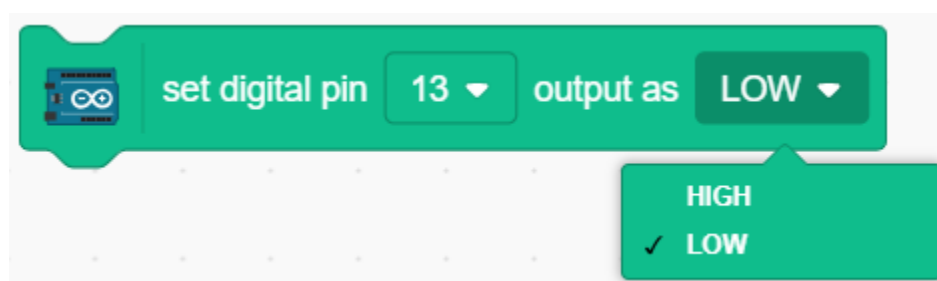
Klicken Sie auf die **Arduino Uno**-Palette.



Die LED auf dem Arduino-Board wird vom digitalen Pin 13 gesteuert (nur 2 Zustände, HIGH oder LOW). Ziehen Sie den Block [set digital pin out as] in den Skriptbereich.

Da der Standardzustand der LED leuchtet, setzen Sie jetzt Pin 13 auf LOW und klicken Sie auf diesen Block. Sie werden sehen, dass die LED erlischt.

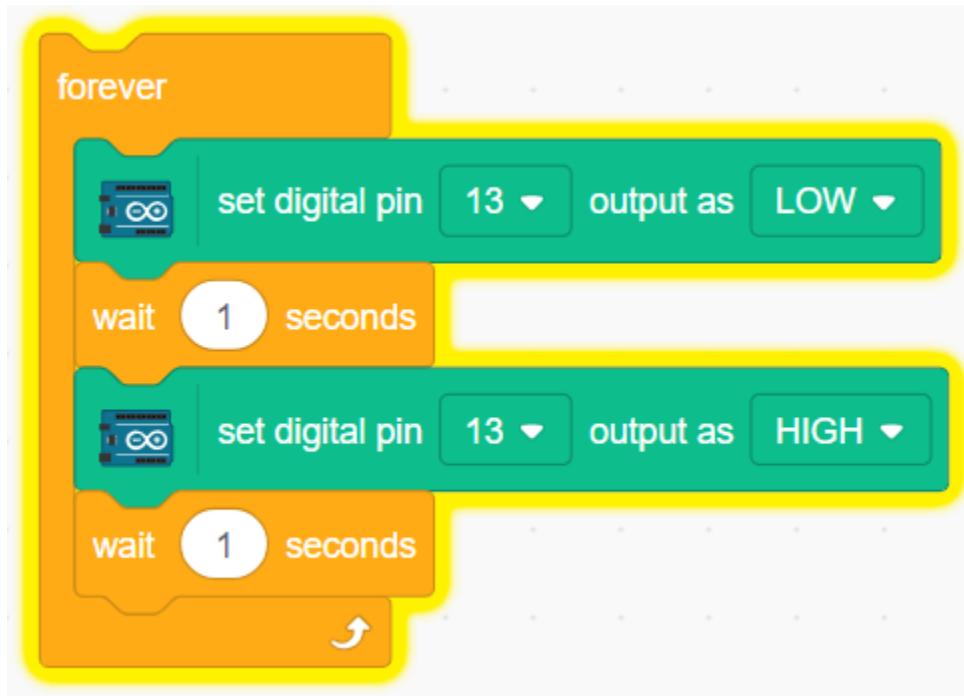
- [set digital pin out as]: Setzt die digitalen Pins (2~13) auf (HIGH/LOW) Level.



Um den Effekt der kontinuierlich blinkenden LED zu sehen, müssen Sie die Blöcke [Wait 1 seconds] und [forever] aus der **Control**-Palette verwenden. Wenn Sie auf diese Blöcke nach dem Schreiben klicken, zeigt ein gelber Halo, dass

sie ausgeführt werden.

- [Wait 1 seconds]: aus der **Control**-Palette, dient zum Festlegen des Zeitintervalls zwischen 2 Blöcken.
- [forever]: aus der **Control**-Palette, ermöglicht es dem Skript, kontinuierlich zu laufen, es sei denn, es wird manuell pausiert.

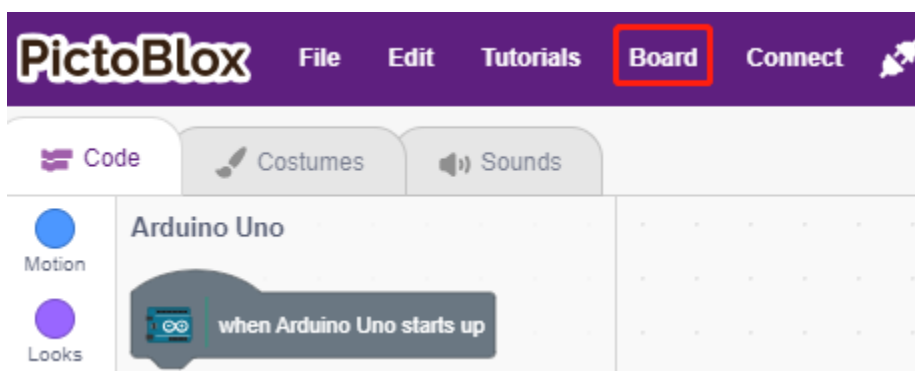


8.3.2 Hochlade-Modus

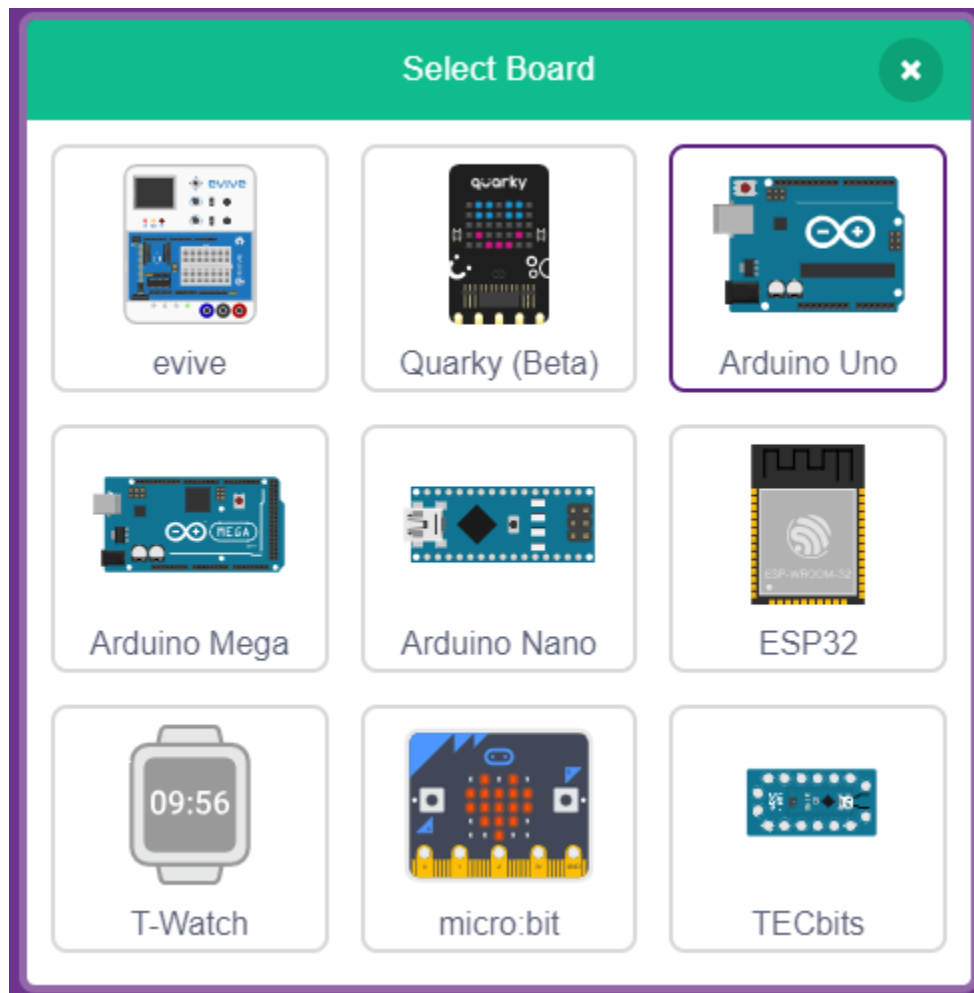
1. Verbindung zum Arduino-Board herstellen

Verbinden Sie Ihr Arduino-Board über ein USB-Kabel mit dem Computer. In der Regel wird der Computer Ihr Board automatisch erkennen und schließlich einen COM-Port zuweisen.

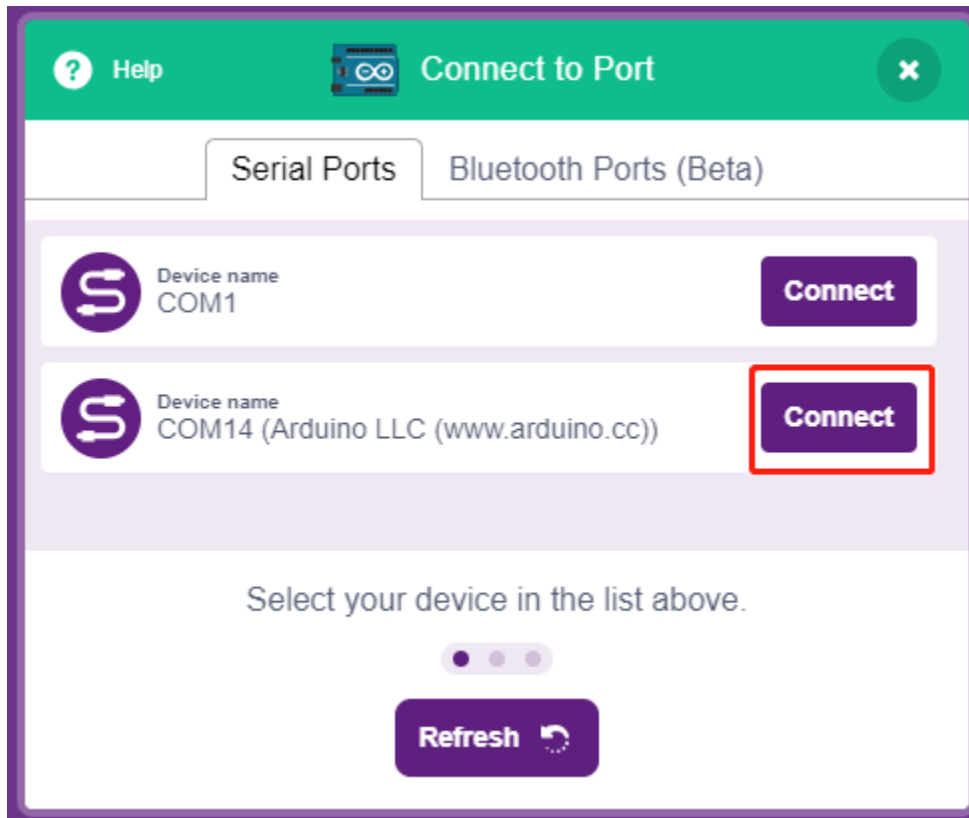
Öffnen Sie PictoBlox und klicken Sie in der oberen rechten Navigationsleiste auf **Board**, um das Board auszuwählen.



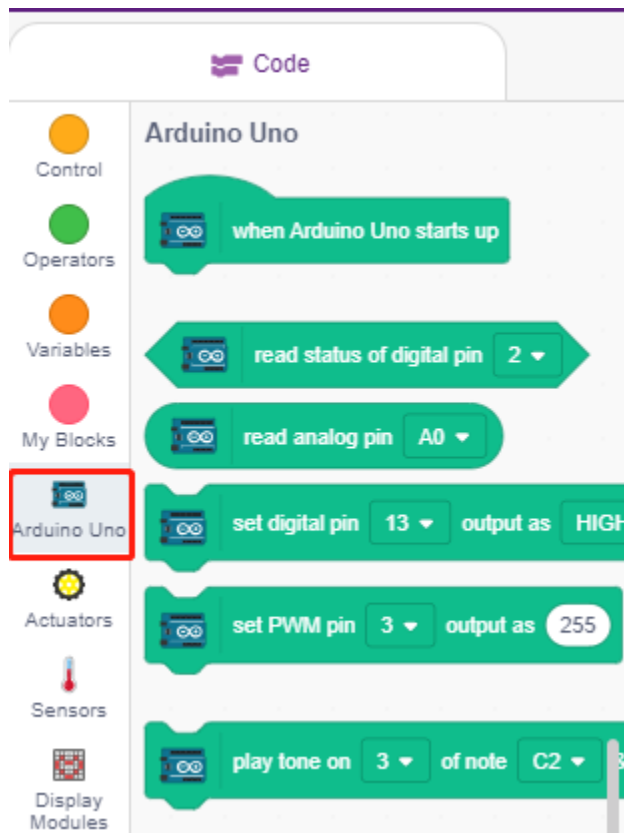
Wählen Sie beispielsweise **Arduino Uno**.



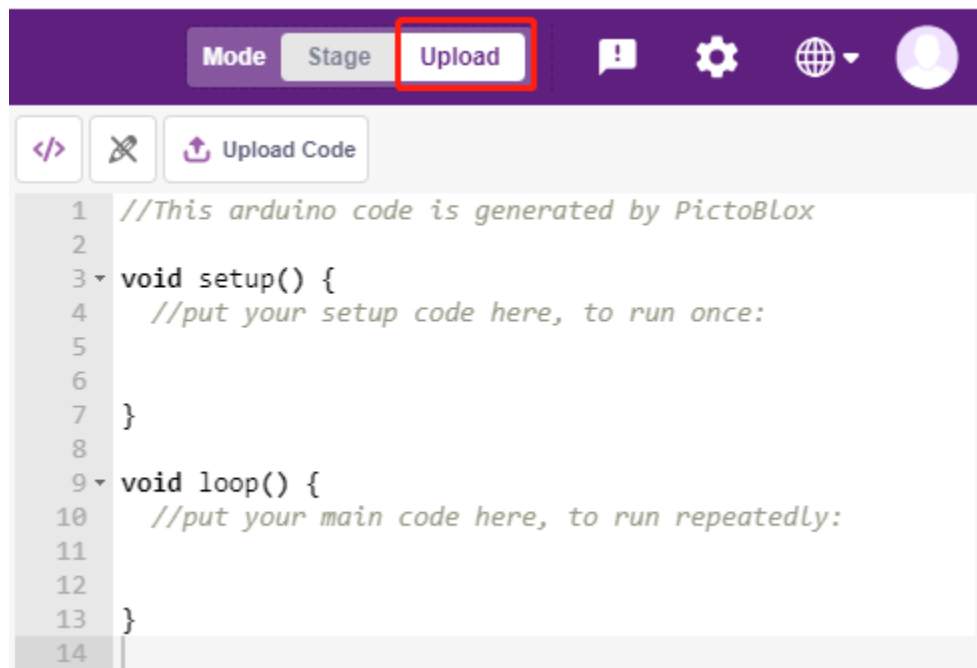
Ein Verbindungsfenster wird geöffnet, in dem Sie den Port zum Verbinden auswählen können. Nach Abschluss der Verbindung kehren Sie zur Startseite zurück. Wenn die Verbindung während der Nutzung unterbrochen wird, können Sie auch auf **Connect** klicken, um sie wiederherzustellen.



Gleichzeitig werden im **Block Palette** Paletten in Bezug auf Arduino Uno, wie z.B. Arduino Uno, Aktuatoren usw., angezeigt.



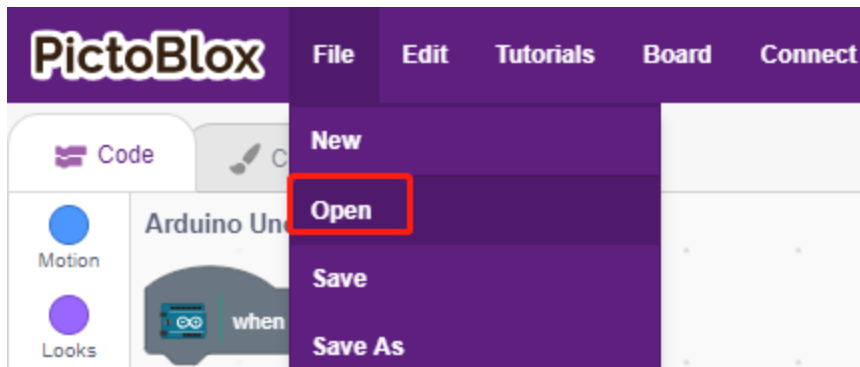
Nach der Auswahl des Hochlade-Modus wird die Bühne zum ursprünglichen Arduino-Codebereich wechseln.



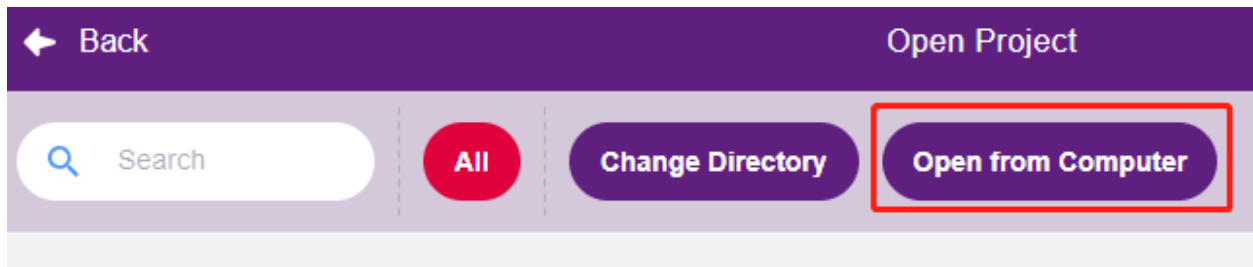
2. Programmierung

- Skript direkt öffnen und ausführen

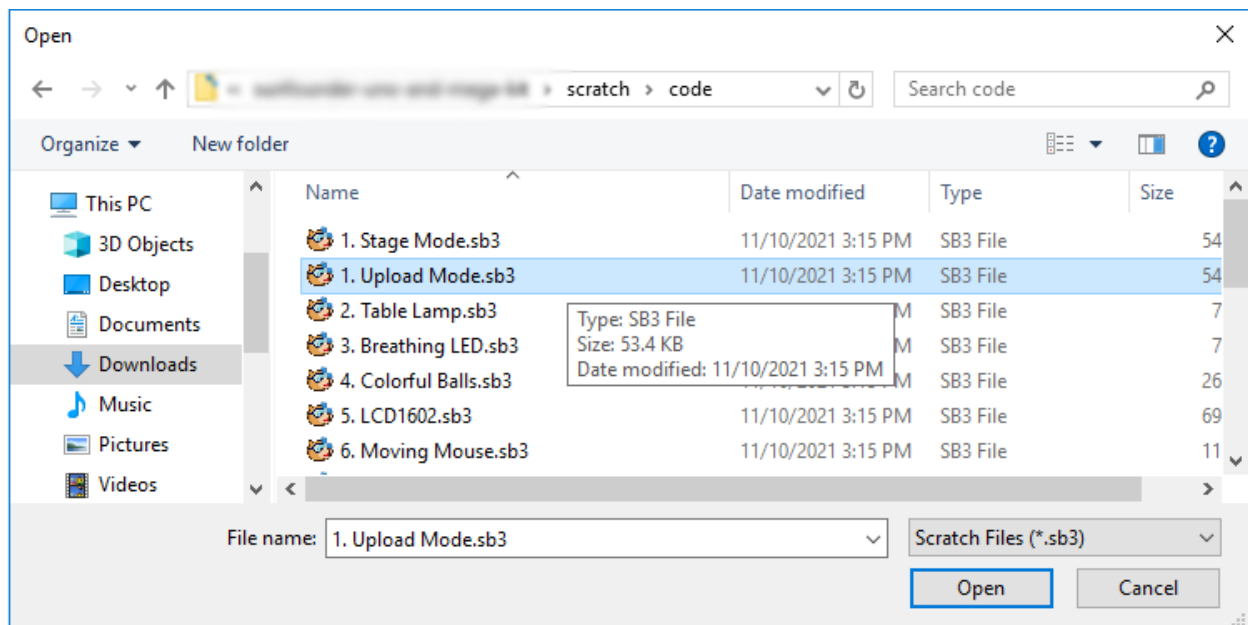
Sie können in der oberen rechten Ecke auf **File** klicken.



Wählen Sie **Open from Computer**.



Navigieren Sie dann zum Pfad `3in1-kit\scratch_project\code` und öffnen Sie **1. Upload Mode.sb3**. Stellen Sie sicher, dass Sie den erforderlichen Code von [github](#) heruntergeladen haben.



Klicken Sie abschließend auf die Schaltfläche **Upload Code**.

The screenshot shows the PictoBlox IDE interface. At the top, there is a purple header bar with buttons for 'Connect', '1. Upload Mode', 'Mode', 'Stage', and 'Upload'. A red box highlights the 'Upload Code' button in the top right corner. The main workspace is divided into two panels. The left panel shows a block-based script: a green 'when Arduino Uno starts up' block followed by an orange 'forever' loop. Inside the loop, there are two green 'set digital pin 13 output as LOW' blocks, each followed by an orange 'wait 1 seconds' block, and then two green 'set digital pin 13 output as HIGH' blocks, each followed by an orange 'wait 1 seconds' block. The right panel shows the corresponding C++ code:

```

1 //This arduino code is generated by PictoBlox
2
3 void setup() {
4   //put your setup code here, to initialize your variables
5   pinMode(13, OUTPUT);
6
7 }
8
9
10 void loop() {
11   //put your main code here, to run repeatedly
12
13
14   digitalWrite(13, false);
15   delay(1 * 1000);
16   digitalWrite(13, true);
17   delay(1 * 1000);
18 }
19

```

At the bottom right, there are buttons for 'Log' and 'Serial Monitor'.

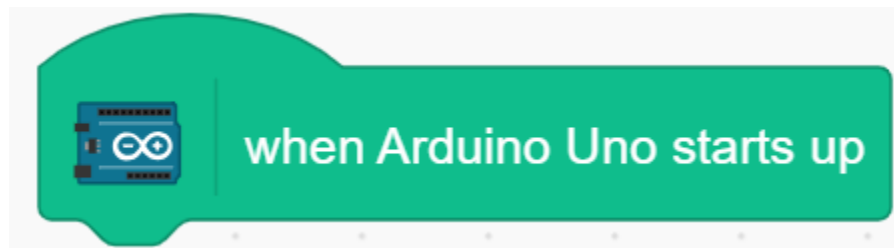
- Programmierung Schritt für Schritt

Sie können das Skript auch Schritt für Schritt schreiben, indem Sie diesen Anweisungen folgen.

Klicken Sie auf die **Arduino Uno**-Palette.



Ziehen Sie [when Arduino Uno starts up] in den Skriptbereich. Dies ist für jedes Skript erforderlich.



Die LED auf dem Arduino-Board wird durch den digitalen Pin13 gesteuert (nur 2 Zustände HIGH oder LOW). Ziehen Sie also den Block [set digital pin out as] in den Skriptbereich.

Da der Standardzustand der LED beleuchtet ist, stellen Sie Pin 13 jetzt auf LOW und klicken Sie auf diesen Block. Sie werden sehen, dass die LED erlischt.

- [set digital pin out as]: Setzt den digitalen Pin (2~13) auf den (HIGH/LOW) Zustand.

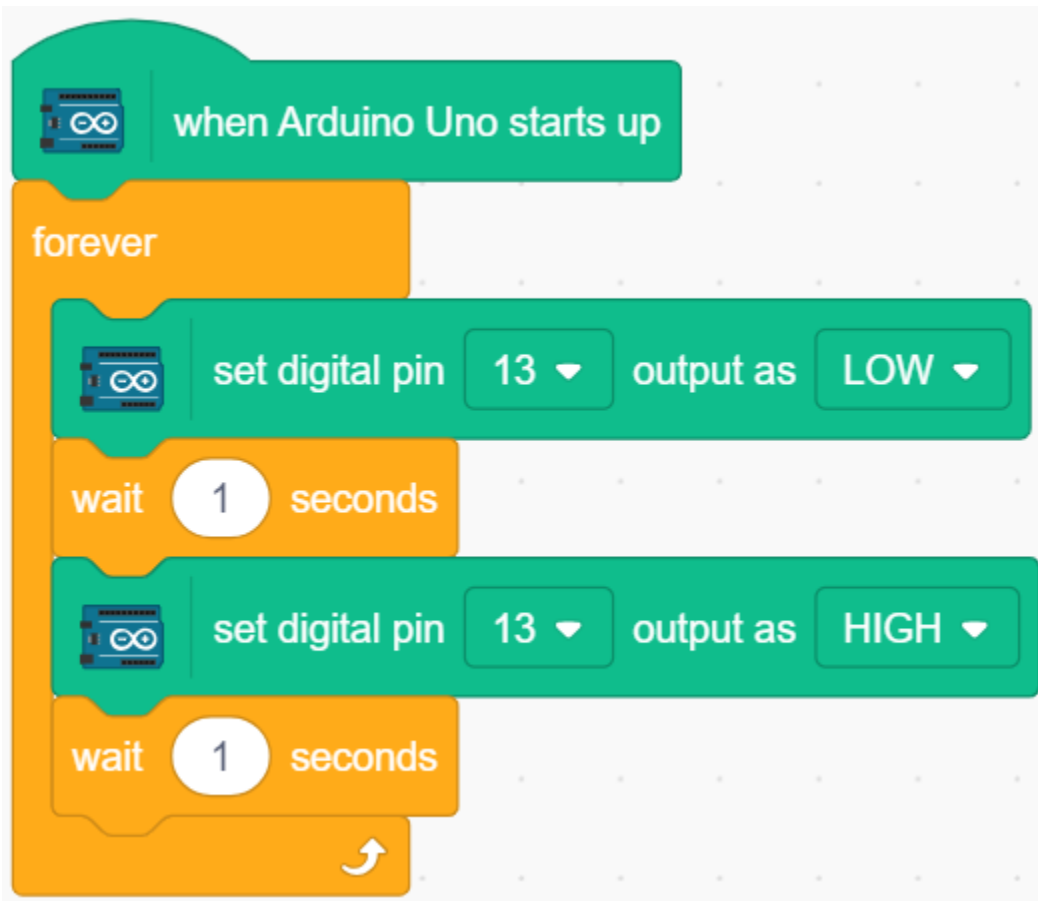


An diesem Punkt wird der Arduino-Code auf der rechten Seite angezeigt. Wenn Sie diesen Code bearbeiten möchten, können Sie den Bearbeitungsmodus aktivieren.

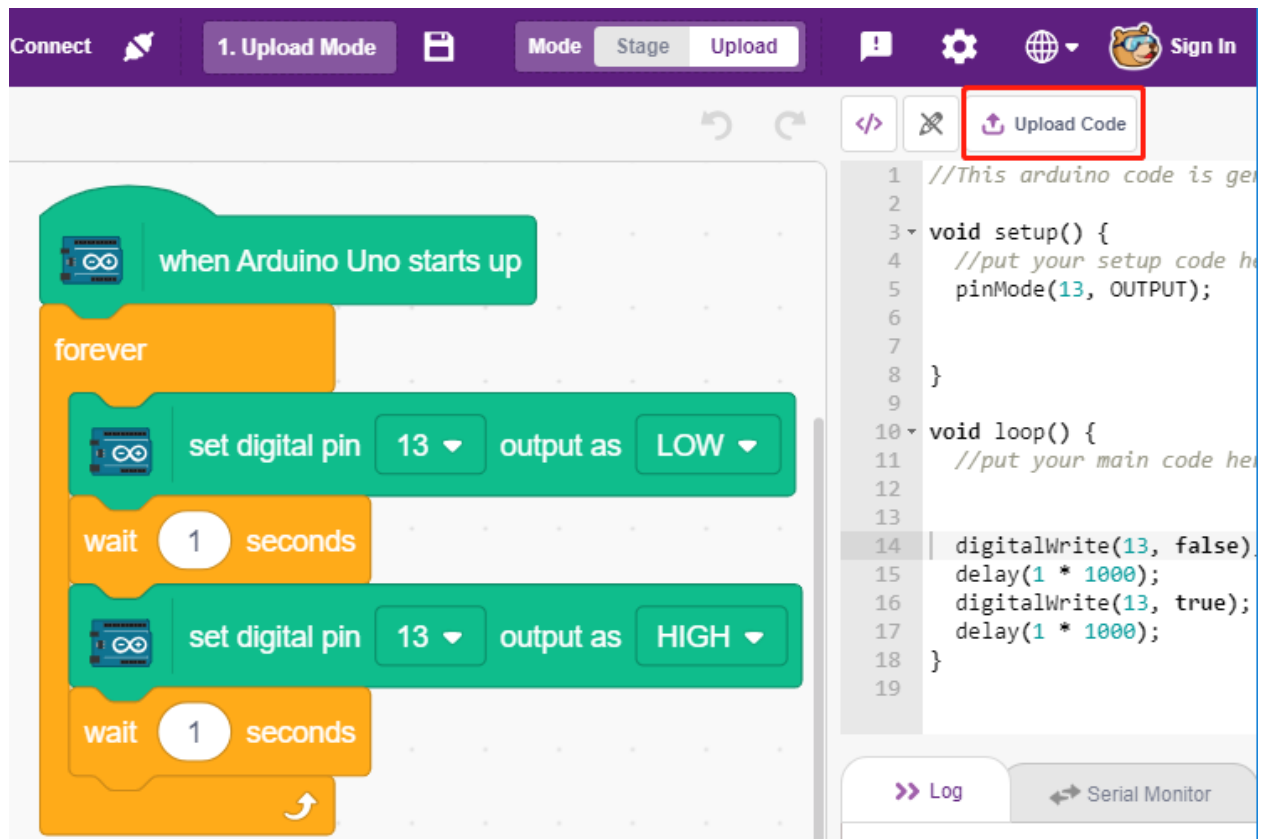


Um den Effekt einer kontinuierlich blinkenden LED zu sehen, müssen Sie die [Wait 1 seconds] und [forever] Blöcke aus der **Control**-Palette verwenden. Klicken Sie nach dem Schreiben auf diese Blöcke. Ein gelber Halo bedeutet, dass sie ausgeführt werden.

- [Wait 1 seconds]: aus der **Control**-Palette, um das Zeitintervall zwischen 2 Blöcken festzulegen.
- [forever]: aus der **Control**-Palette, lässt das Skript laufen, es sei denn, die Stromversorgung wird unterbrochen.



Klicken Sie abschließend auf die Schaltfläche **Upload Code**.



2. Projekte

Die folgenden Projekte sind nach ihrer Programmierschwierigkeit geordnet. Es wird empfohlen, diese Bücher in der gegebenen Reihenfolge durchzugehen.

In jedem Projekt gibt es sehr detaillierte Schritte, die Ihnen beibringen, wie Sie den Schaltkreis aufbauen und wie Sie ihn Schritt für Schritt programmieren können, um das gewünschte Ergebnis zu erzielen.

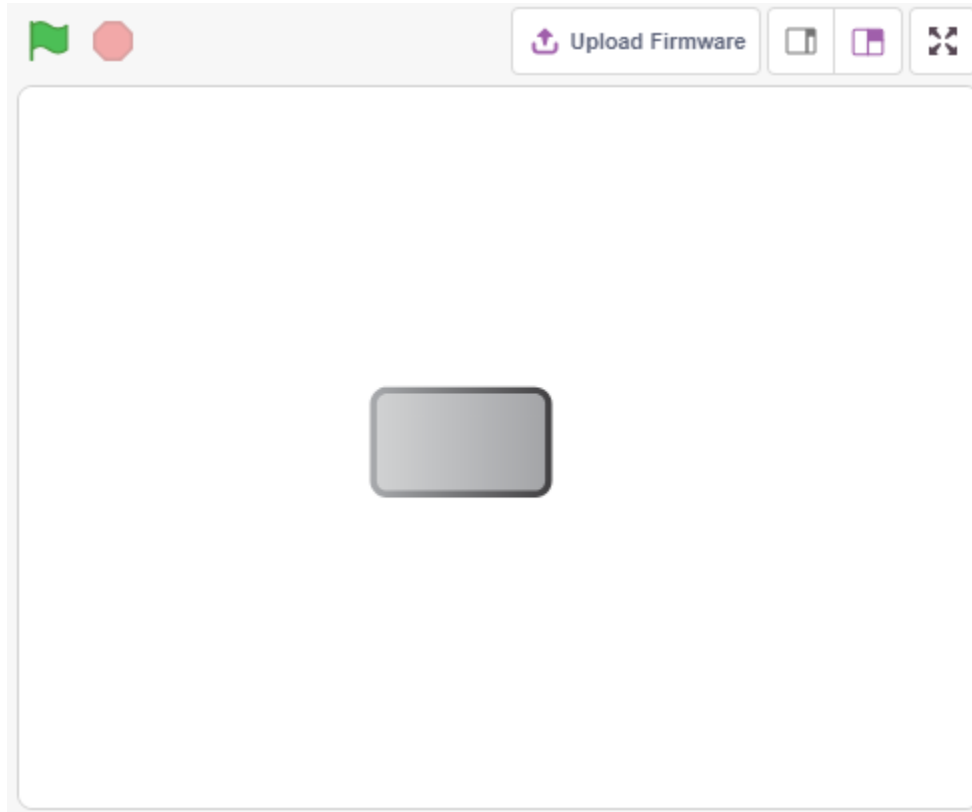
Natürlich können Sie das Skript auch direkt ausführen. Stellen Sie jedoch sicher, dass Sie das relevante Material von [github](#) heruntergeladen haben.

Nach dem Download entpacken Sie es. Beachten Sie *Bühnenmodus*, um einzelne Skripte direkt auszuführen.

Das 2.9 *Temperatur und Feuchtigkeit auslesen* verwendet jedoch den *Hochlade-Modus*.

8.4 2.1 Tischlampe

Hier verbinden wir eine LED auf dem Steckbrett und lassen die Spielfigur (Sprite) das Blinken dieser LED steuern. Wenn der Button-Sprite auf der Bühne angeklickt wird, blinkt die LED 5-mal und stoppt dann.



8.4.1 Was Sie lernen werden

- Steckbrett, LEDs und Widerstände
- Aufbau einer Schaltung auf dem Steckbrett
- Sprites löschen und auswählen
- Kostüme wechseln
- Eine begrenzte Anzahl von Wiederholungsschleifen festlegen

8.4.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden 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+	

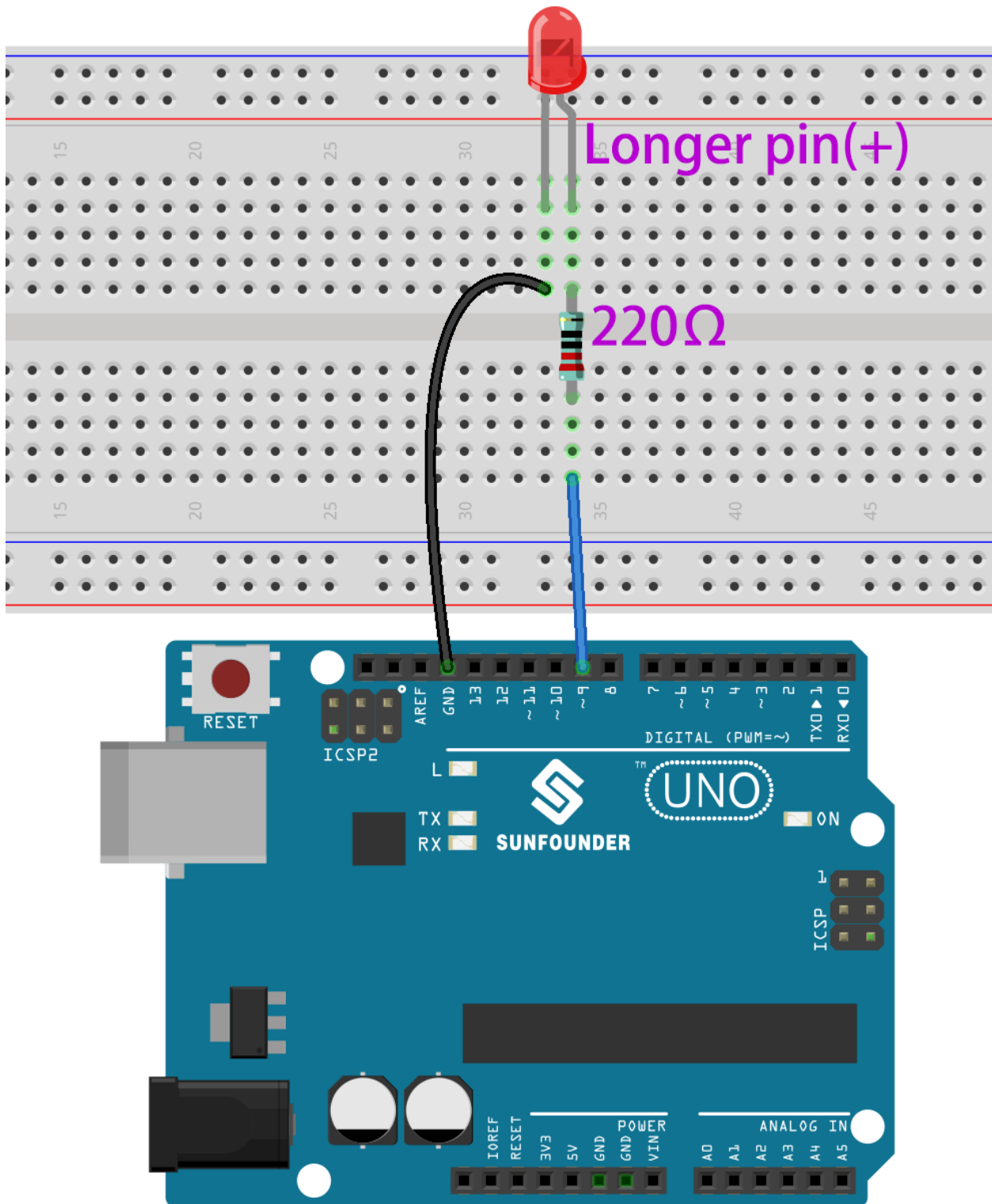
Sie können diese auch separat über die untenstehenden Links kaufen.

KOMPONENTENEINFÜHRUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	

8.4.3 Schaltung aufbauen

Folgen Sie dem untenstehenden Diagramm, um die Schaltung auf dem Steckbrett aufzubauen.

Da die Anode der LED (der längere Pin) über einen 220-Widerstand mit Pin 9 verbunden ist und die Kathode der LED mit GND verbunden ist, können Sie die LED zum Leuchten bringen, indem Sie Pin 9 auf ein hohes Niveau setzen.

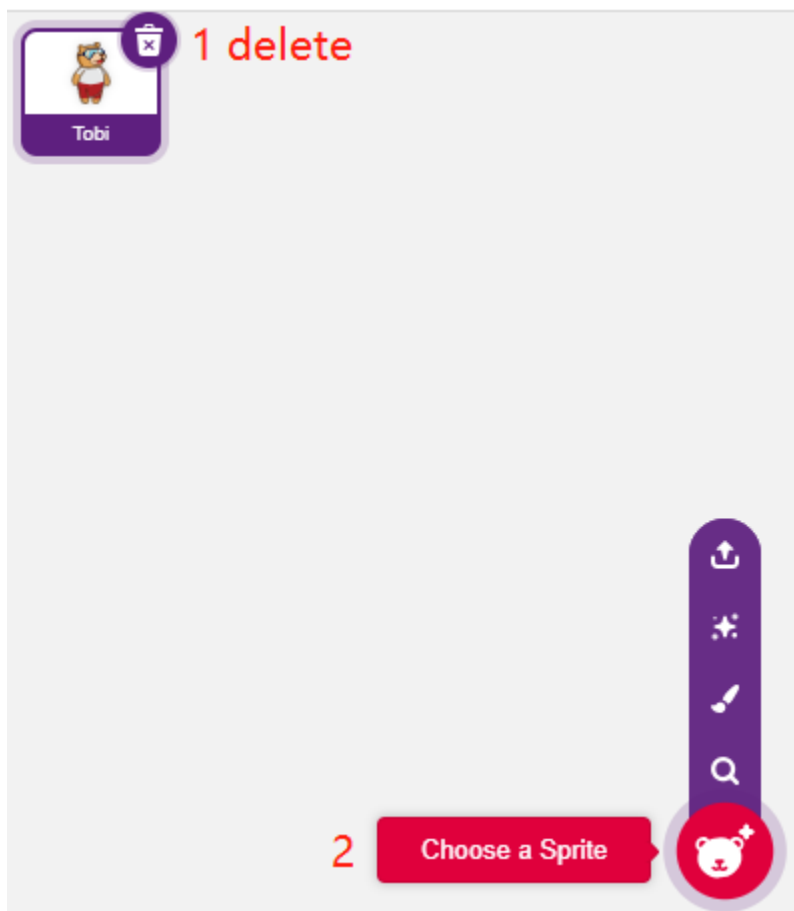


8.4.4 Programmierung

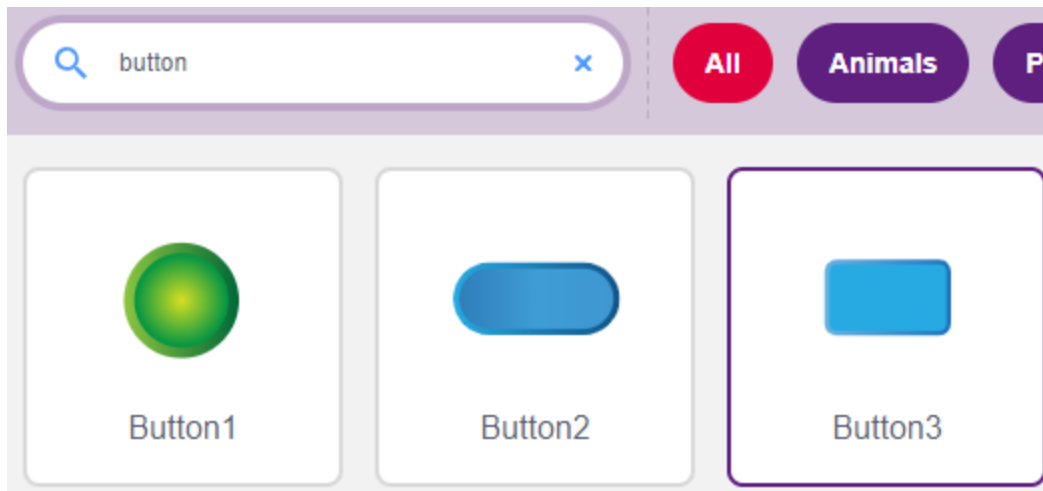
Die gesamte Programmierung ist in 3 Teile unterteilt: Im ersten Teil wird der gewünschte Sprite ausgewählt, im zweiten Teil wird das Kostüm des Sprites gewechselt, um ihn anklickbar erscheinen zu lassen, und im dritten Teil wird die LED zum Blinken gebracht.

1. Button3-Sprite auswählen

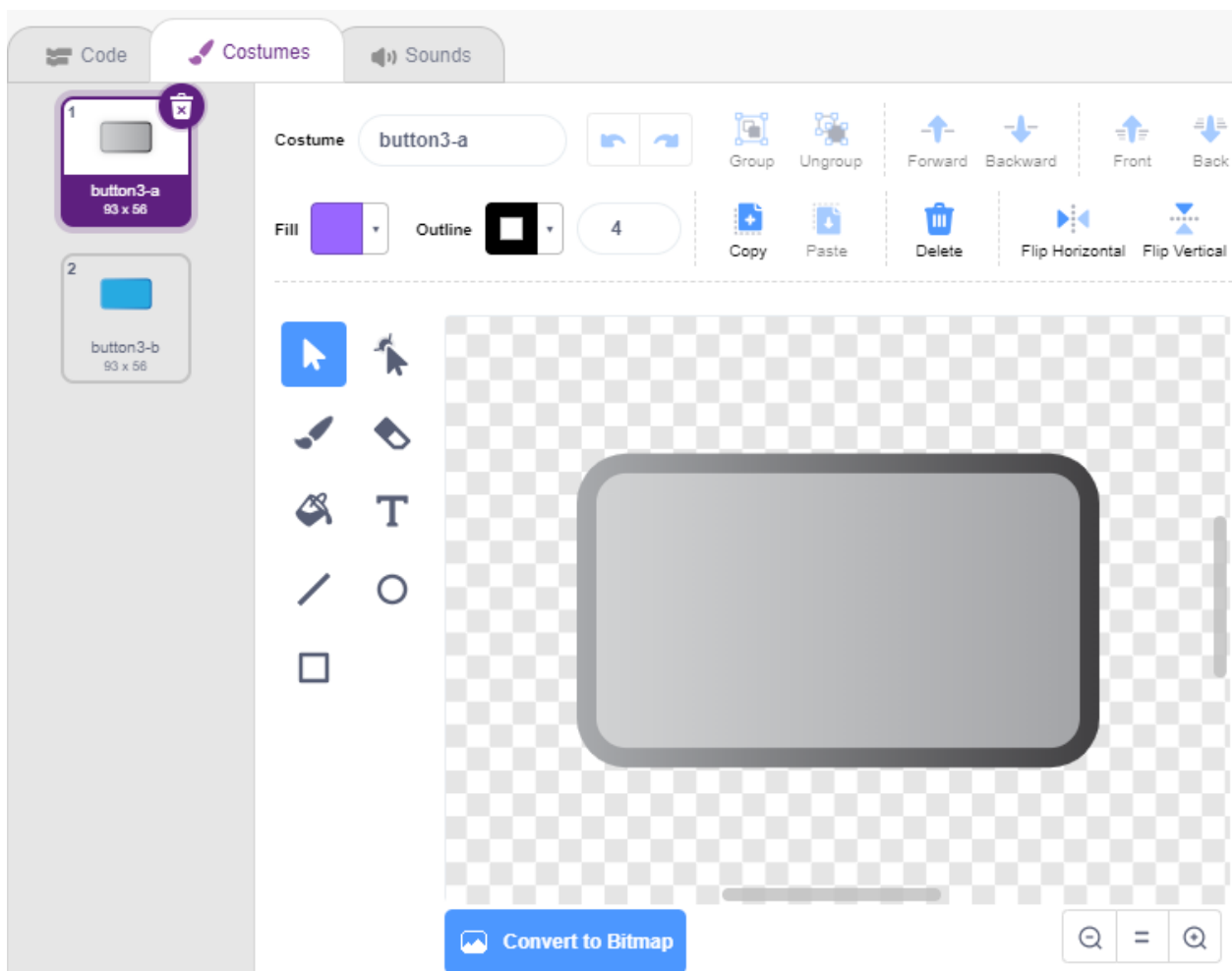
Löschen Sie den vorhandenen Tobi-Sprite mit der Löschen-Taste in der oberen rechten Ecke und wählen Sie erneut einen Sprite aus.



Hier wählen wir den **Button3**-Sprite aus.

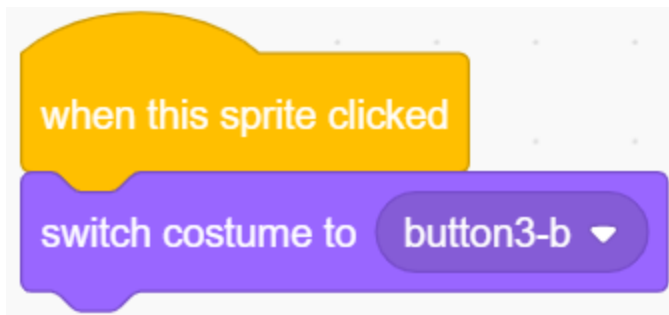


Klicken Sie in der oberen rechten Ecke auf Kostüme und Sie werden sehen, dass der Button3-Sprite 2 Kostüme hat. Wir setzen **button3-a** auf „freigegeben“ und **button3-b** auf „gedrückt“.



2. Kostümwechsel.

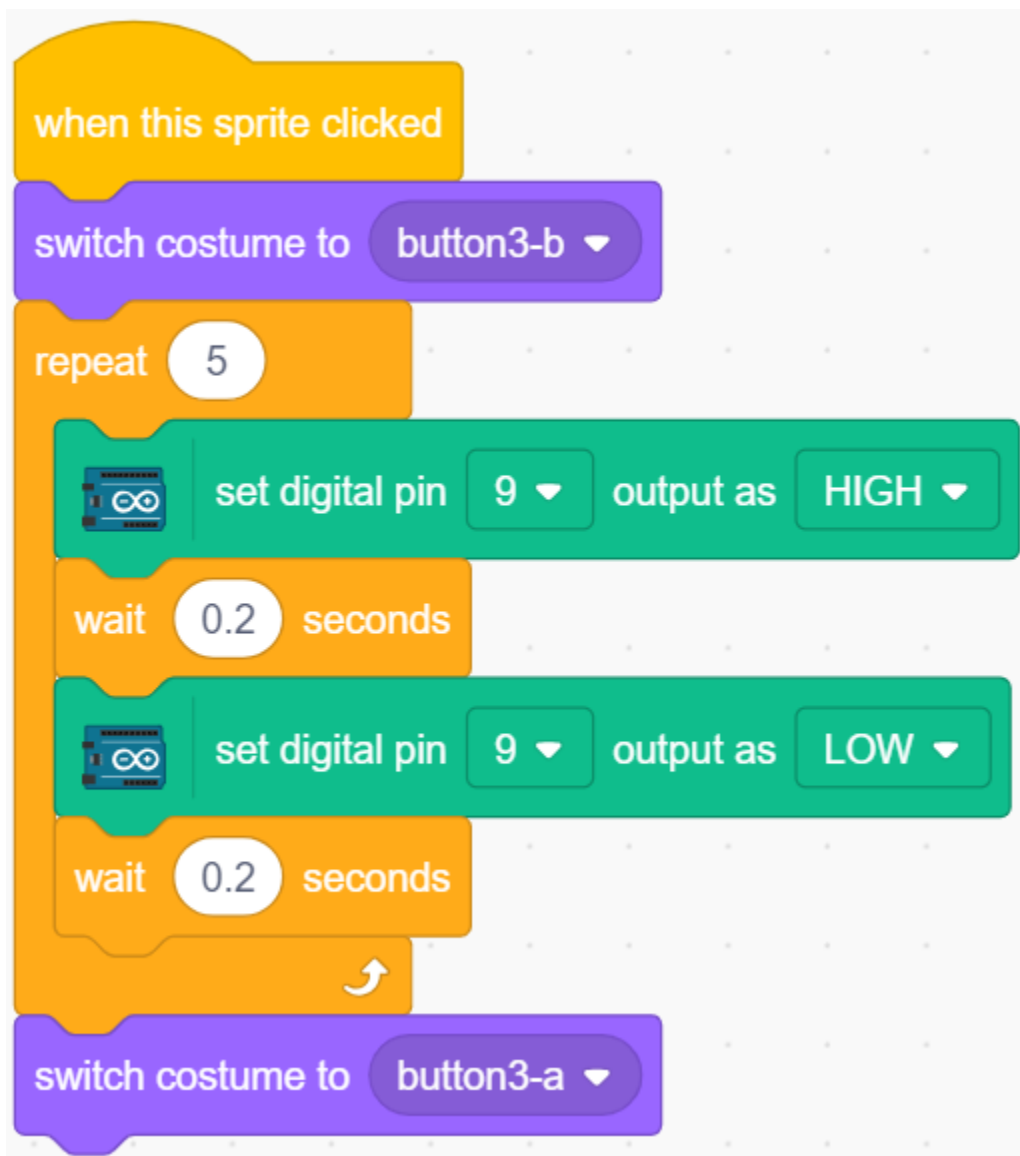
Wenn der Sprite angeklickt wird (**Events**-Palette), wechselt er zum Kostüm für **button3-b** (**looks**-Palette).



3. LED 5-mal blinken lassen

Verwenden Sie den [Repeat]-Block, um die LED 5-mal blinken zu lassen (High-> LOW-Zyklus). Vergessen Sie nicht, Pin 13 auf Pin 9 zu ändern und schließlich das Kostüm zurück auf **button3-a** zu wechseln.

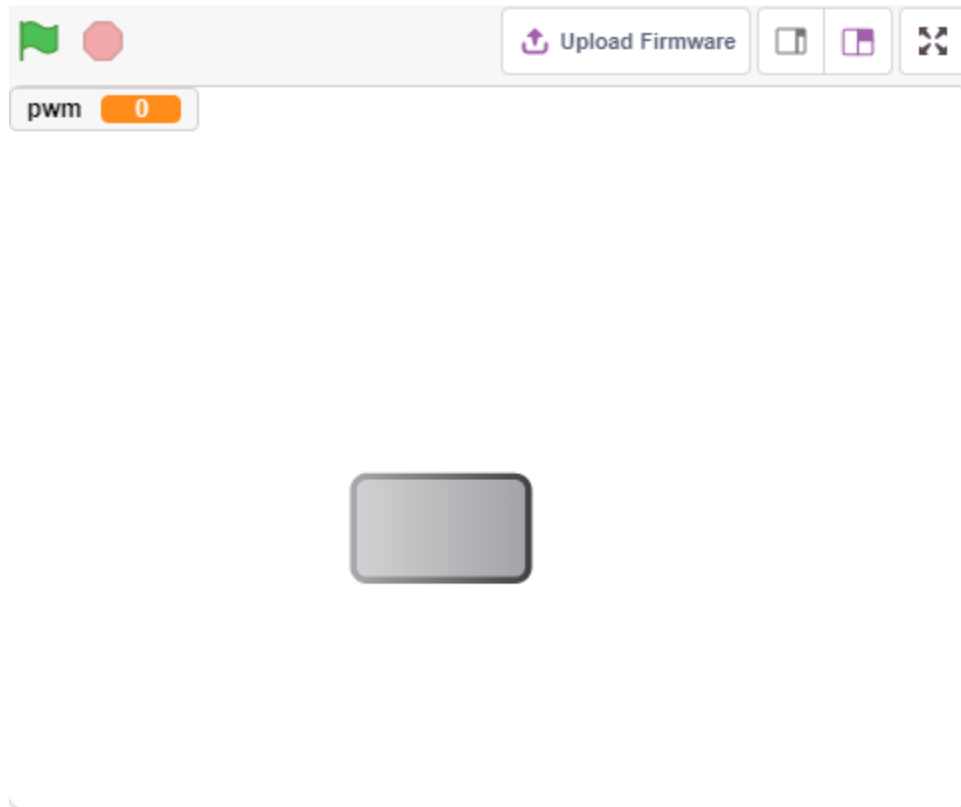
- [Repeat 10]: Begrenzte Anzahl von Wiederholungsschleifen. Sie können die Anzahl der Wiederholungen selbst festlegen, aus der **Control**-Palette.



8.5 2.2 Atmende LED

Verwenden Sie nun eine andere Methode, um die Helligkeit der LED zu steuern. Anders als im vorherigen Projekt wird hier die Helligkeit der LED langsam reduziert, bis sie vollständig erlischt.

Wenn der Sprite auf der Bühne angeklickt wird, erhöht sich die Helligkeit der LED langsam und erlischt dann sofort.



8.5.1 Was Sie lernen werden

- Den Ausgabewert des PWM-Pins einstellen
- Variablen erstellen
- Ändern der Helligkeit des Sprites

8.5.2 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+	

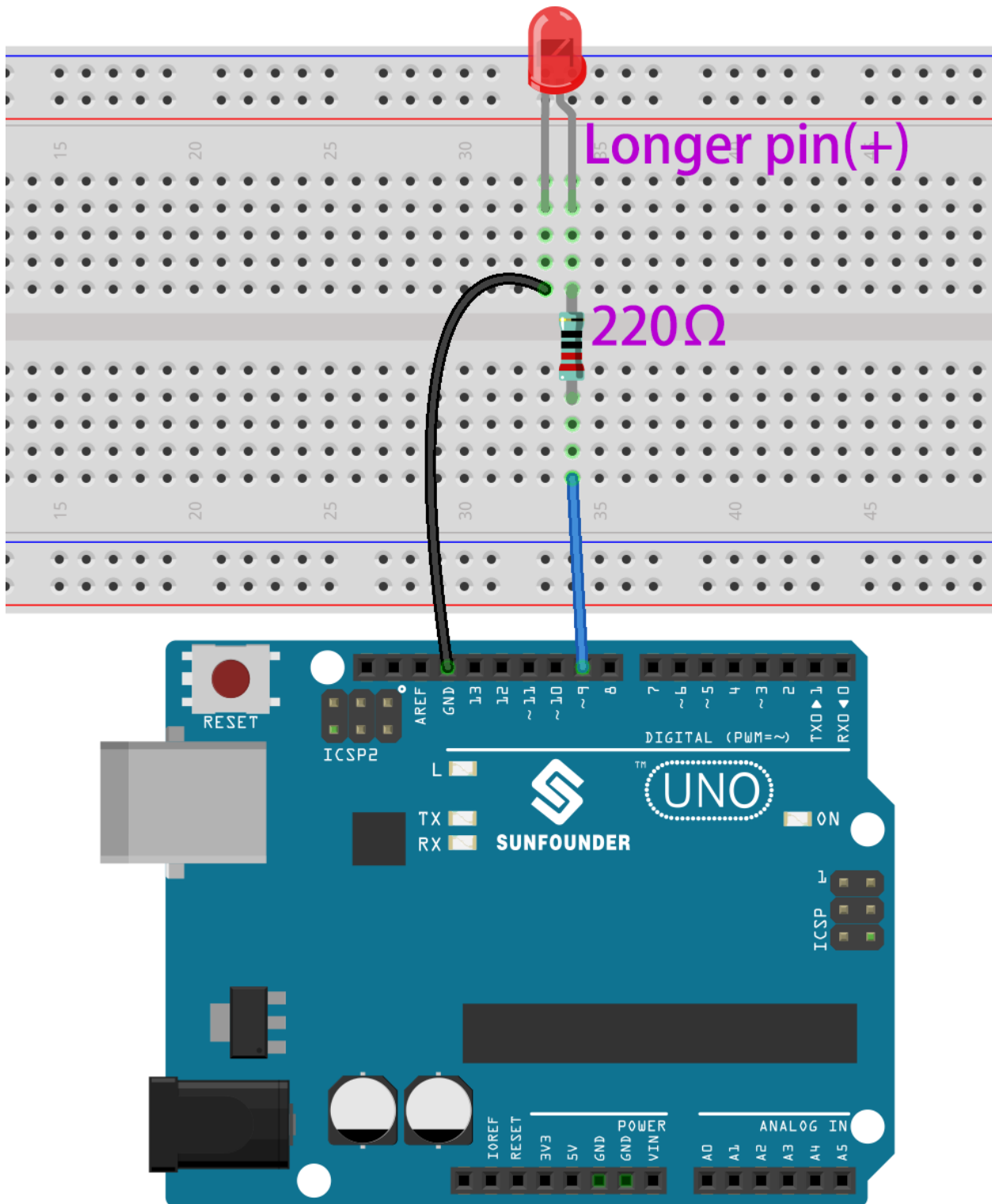
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENÜBERSICHT	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>LED</i>	

8.5.3 Schaltung aufbauen

Dieses Projekt verwendet denselben Schaltkreis wie das vorherige Projekt [2.1 Tischlampe](#). Statt jedoch HIGH/LOW zu verwenden, um die LEDs ein- oder auszuschalten, wird in diesem Projekt das [PWM - Wikipedia](#) Signal verwendet, um die LED langsam heller zu machen oder zu dimmen.

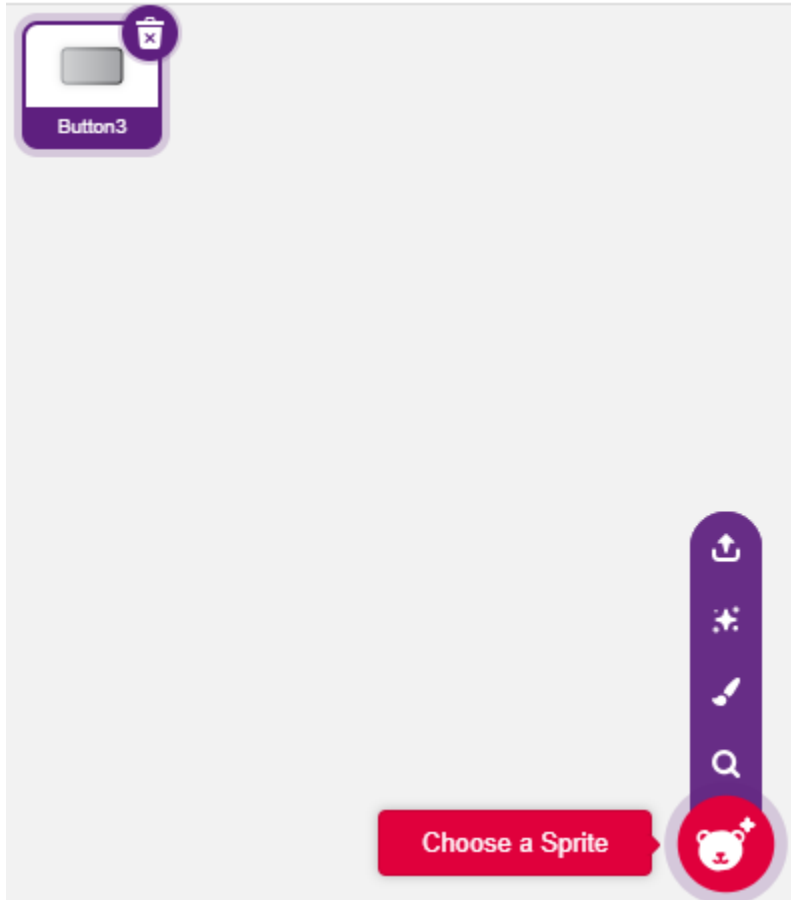
Der PWM-Signalebereich liegt zwischen 0-255. Auf dem Arduino Uno-Board können die Pins 3, 5, 6, 9, 10, 11 ein PWM-Signal ausgeben; auf dem Mega2560 sind es die Pins 2 - 13, 44 - 46.



8.5.4 Programmierung

1. Einen Sprite auswählen

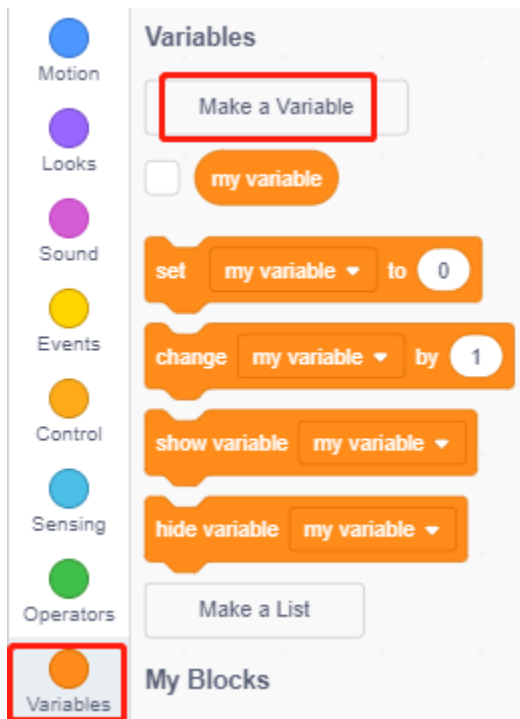
Löschen Sie den Standard-Sprite, klicken Sie auf die Schaltfläche **Choose a Sprite** in der unteren rechten Ecke des Sprite-Bereichs, geben Sie **button3** in das Suchfeld ein und klicken Sie dann darauf, um es hinzuzufügen.



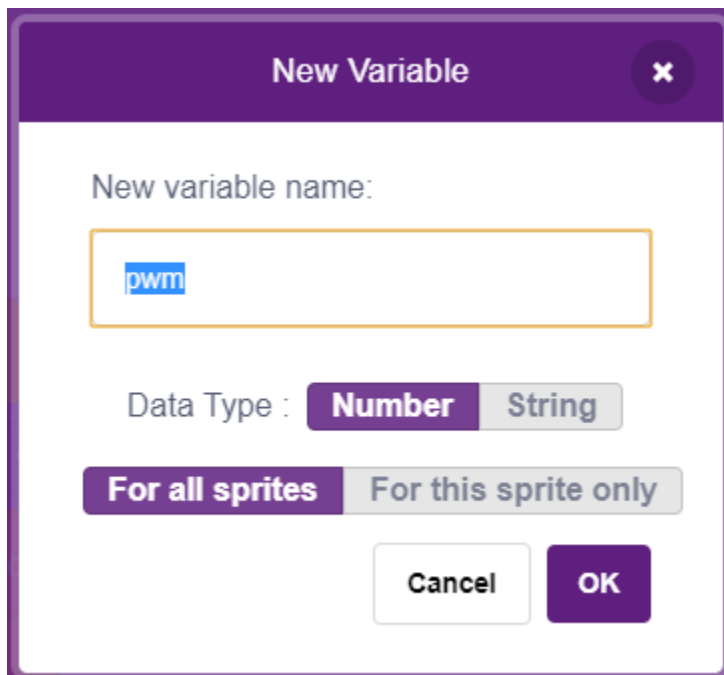
2. Eine Variable erstellen.

Erstellen Sie eine Variable namens **pwm**, um den Wert der PWM-Änderung zu speichern.

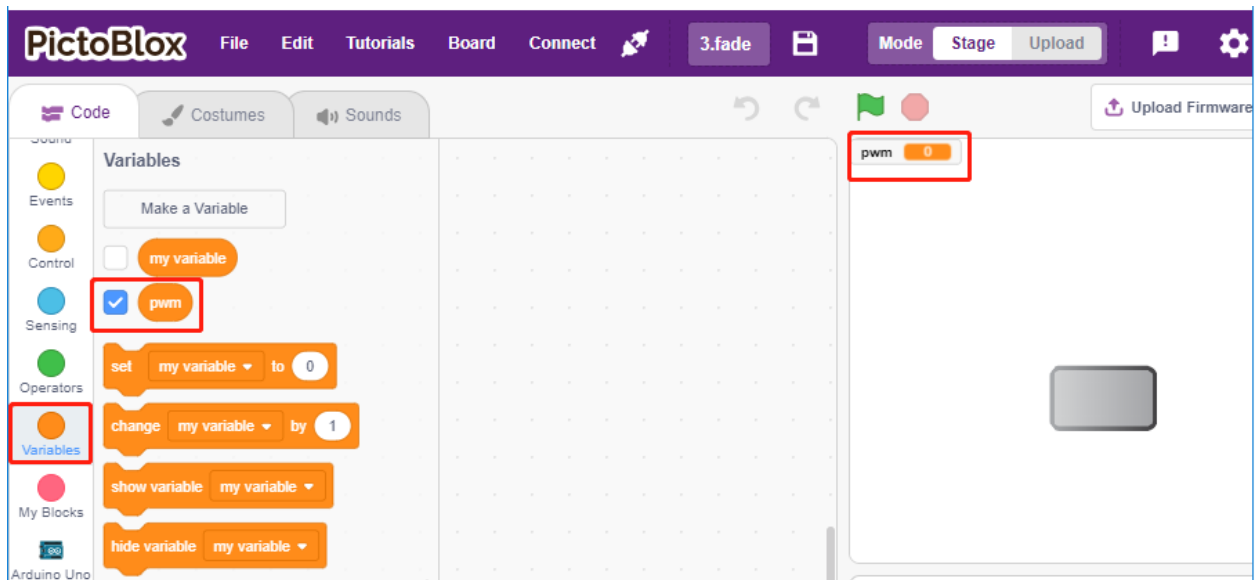
Klicken Sie auf die **Variables**-Palette und wählen Sie **Make a Variable**.



Geben Sie den Namen der Variable ein; es kann jeder Name sein, aber es wird empfohlen, seine Funktion zu beschreiben. Der Datentyp ist Zahl und für alle Sprites.



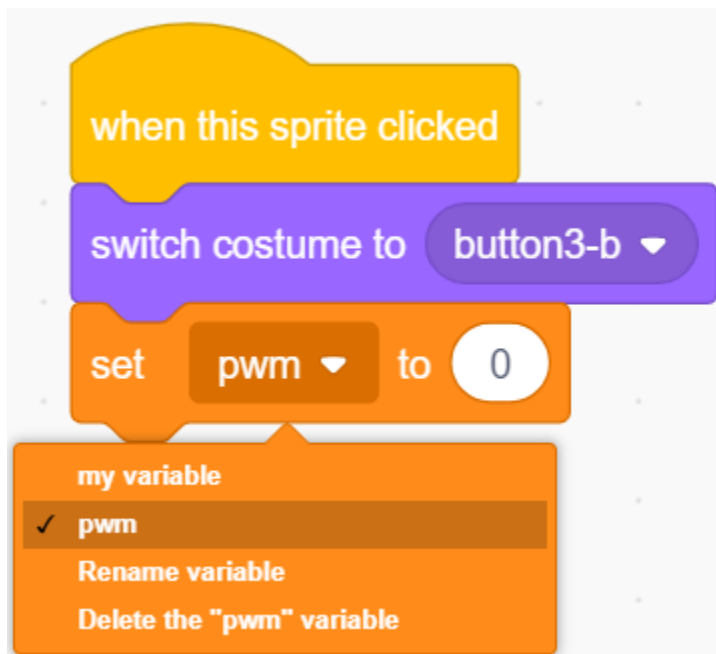
Nach dem Erstellen sehen Sie **pwm** innerhalb der **Variables**-Palette und im aktivierten Zustand. Das bedeutet, dass diese Variable auf der Bühne erscheinen wird. Sie können versuchen, sie zu deaktivieren und zu sehen, ob pwm noch auf der Bühne präsent ist.



3. Den Anfangszustand festlegen

Wenn der **button3**-Sprite angeklickt wird, wechseln Sie das Kostüm zu **button-b** (angeklickter Zustand) und setzen Sie den Anfangswert der Variable **pwm** auf 0.

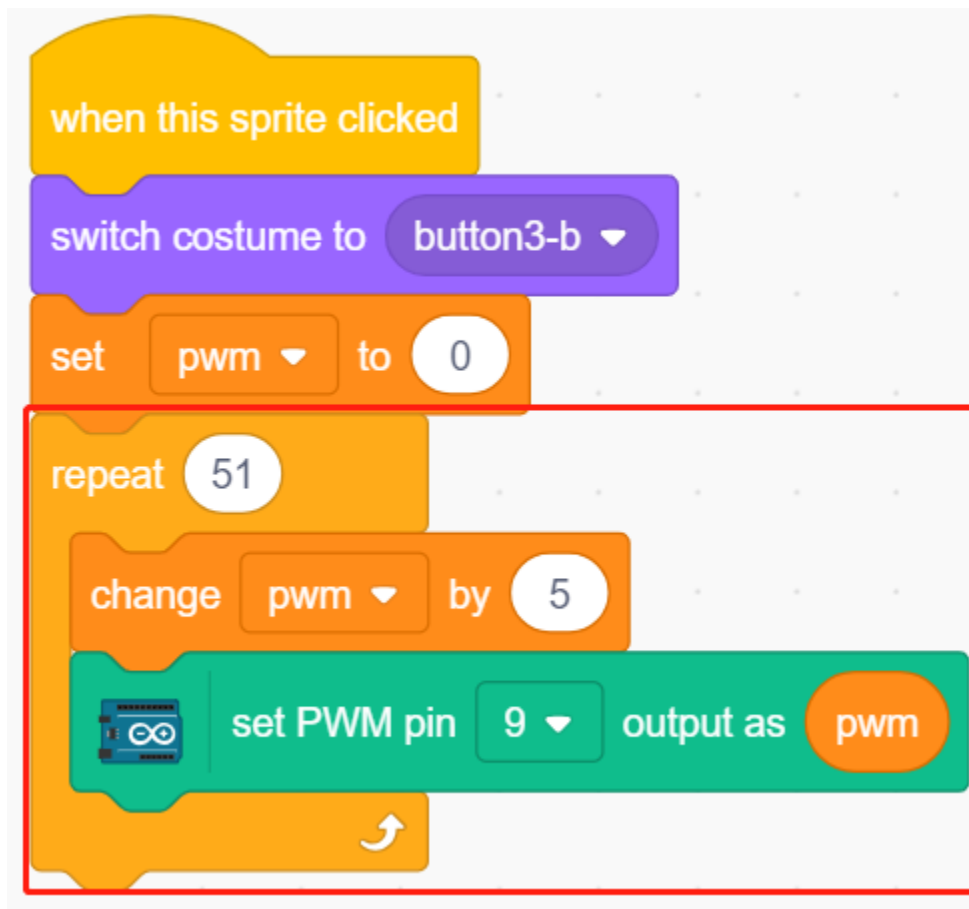
- [set pwm to 0]: aus der **Variables**-Palette, um den Wert der Variable festzulegen.



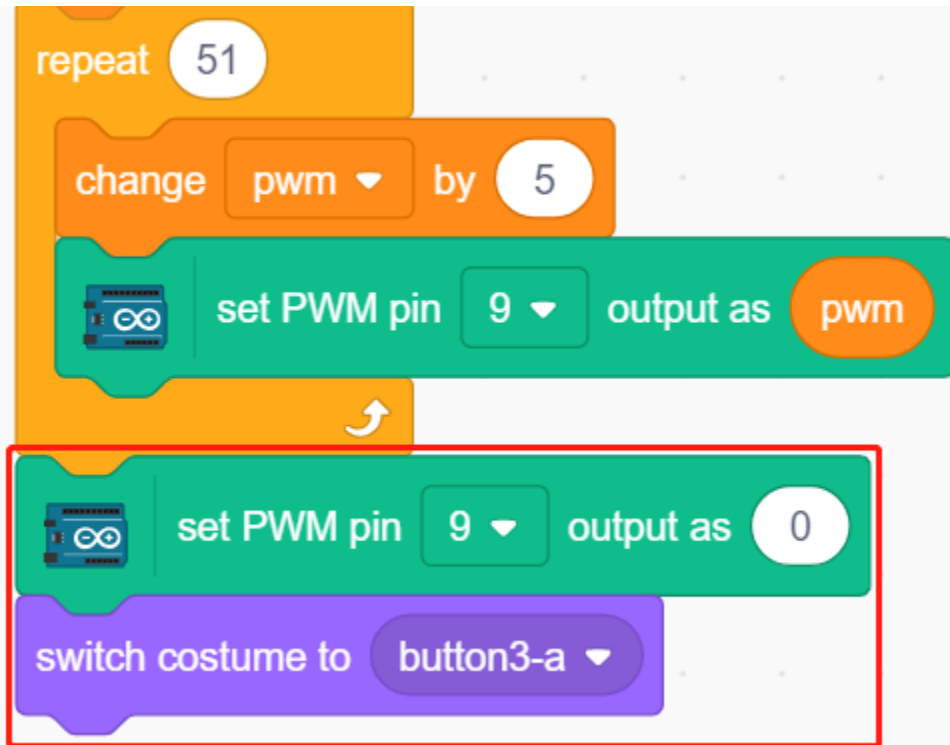
4. Die LED immer heller machen

Da der Bereich von pwm 255 ist, wird durch den [repeat]-Block die Variable **pwm** um 5 auf 255 akkumuliert und dann in den [set PWM pin]-Block eingegeben, sodass Sie sehen können, wie die LED langsam aufleuchtet.

- [change pwm by 5]: aus der **Variablen**-Palette. Lässt die Variable jedes Mal eine bestimmte Zahl ändern. Es kann eine positive oder negative Zahl sein. Positiv bedeutet jedes Mal eine Erhöhung, negativ bedeutet jedes Mal eine Verringerung. Hier wird die Variable pwm jedes Mal um 5 erhöht.
- [set PWM pin]: aus der **Arduino Uno**-Palette, um den Ausgabewert des PWM-Pins festzulegen.



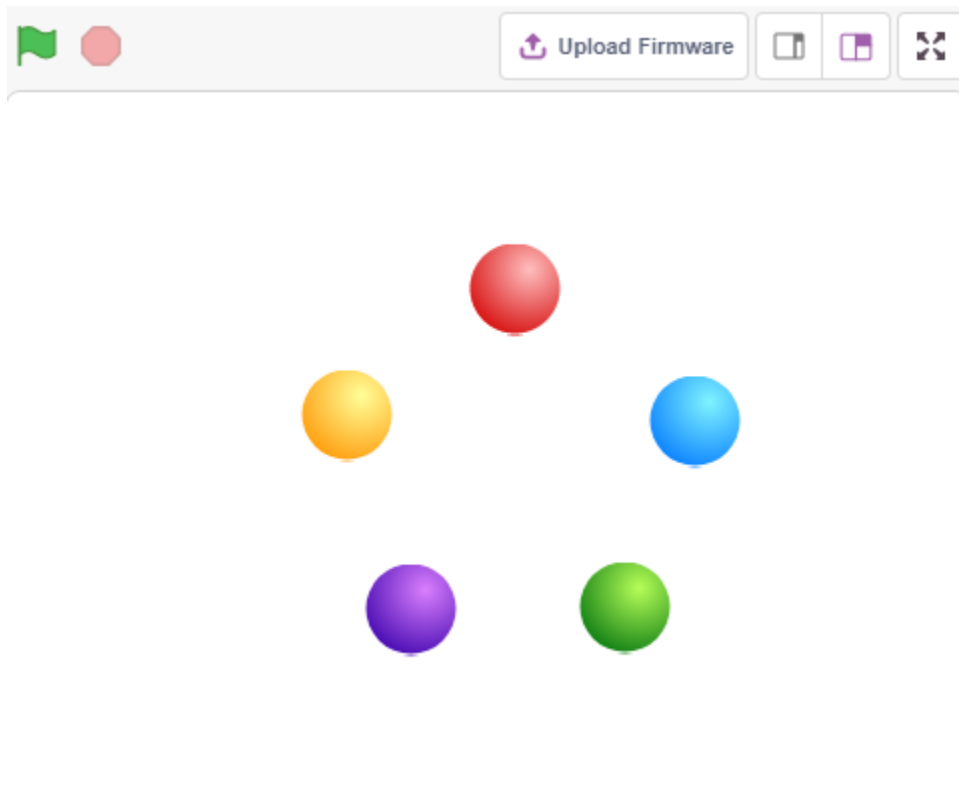
Schließlich wechseln Sie das Kostüm von button3 zurück zu **button-a** und setzen den Wert des PWM-Pins auf 0, sodass die LED langsam aufleuchtet und dann wieder erlischt.



8.6 2.3 Bunte Kugeln

In diesem Projekt werden wir die RGB-LEDs verschiedene Farben anzeigen lassen.

Wenn Sie auf verschiedene farbige Kugeln im Bühnenbereich klicken, leuchtet die RGB-LED in unterschiedlichen Farben auf.



8.6.1 Was Sie lernen werden

- Das Prinzip der RGB-LED
- Sprites kopieren und verschiedene Kostüme auswählen
- Überlagerung der drei Grundfarben

8.6.2 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+	

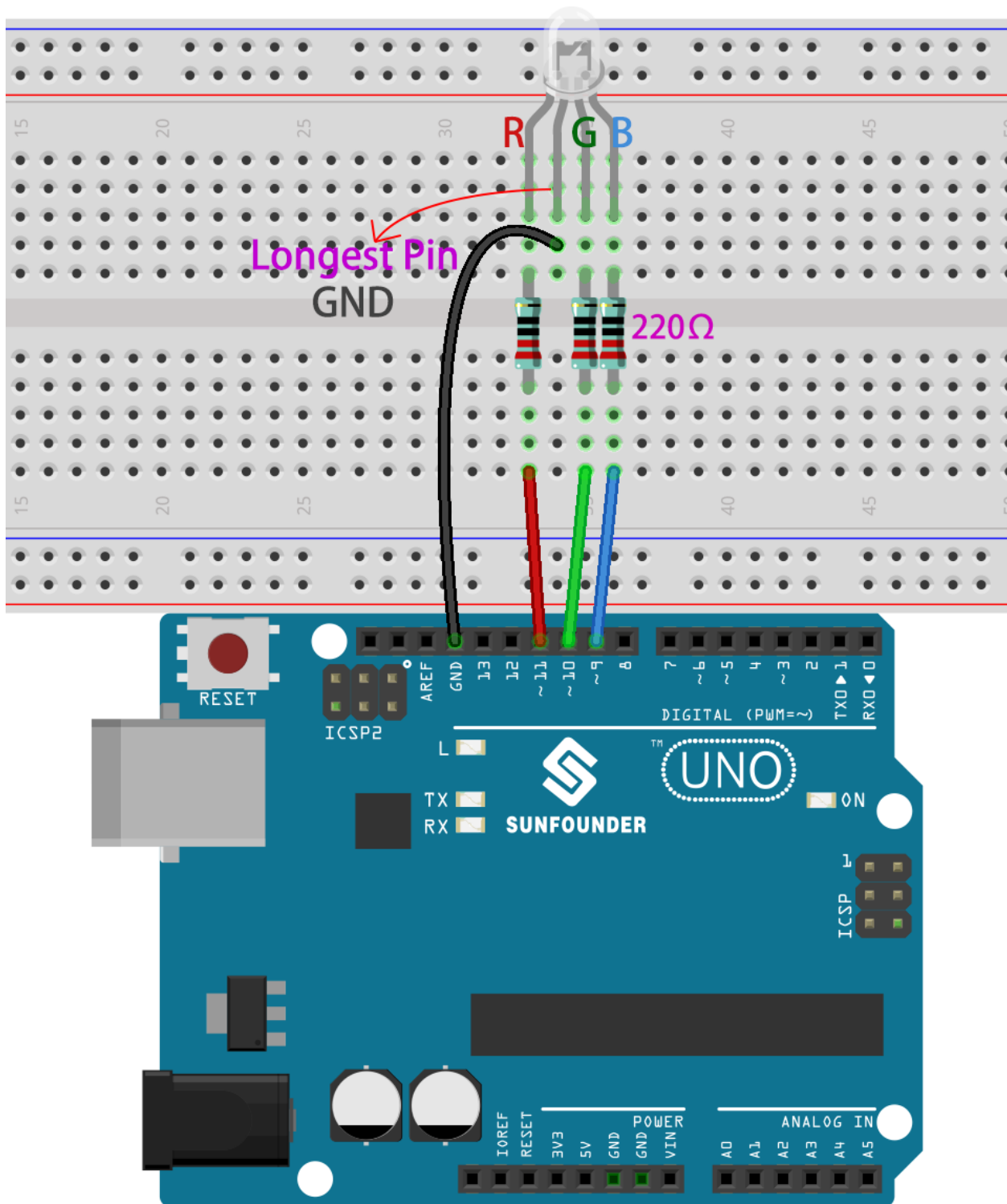
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>RGB LED</i>	

8.6.3 Schaltung aufbauen

Eine RGB-LED verpackt drei LEDs in Rot, Grün und Blau in eine transparente oder halbtransparente Plastikhülle. Durch Ändern der Eingangsspannung der drei Pins und Überlagern von ihnen können, statistisch gesehen, 16.777.216 verschiedene Farben erzeugt werden.

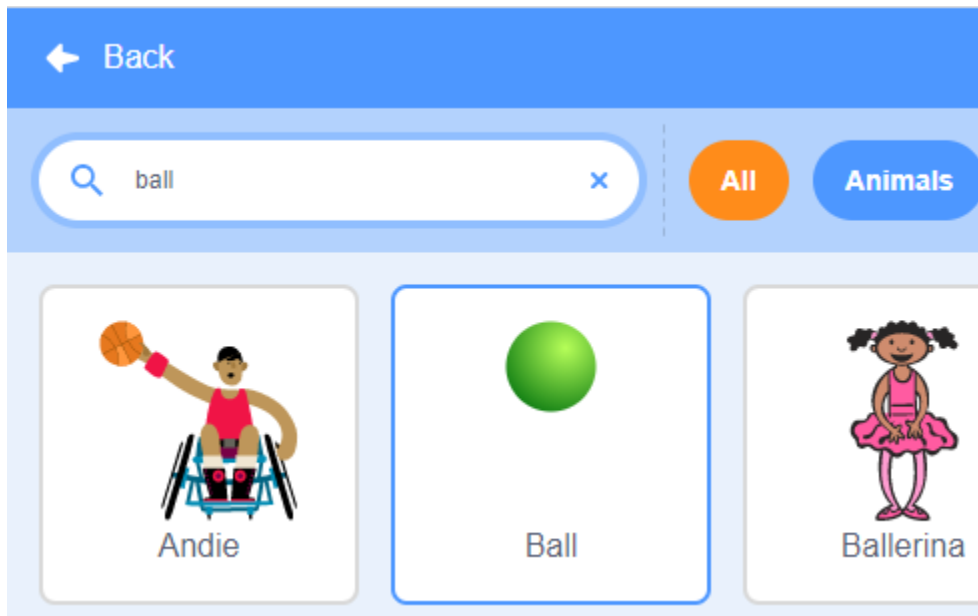




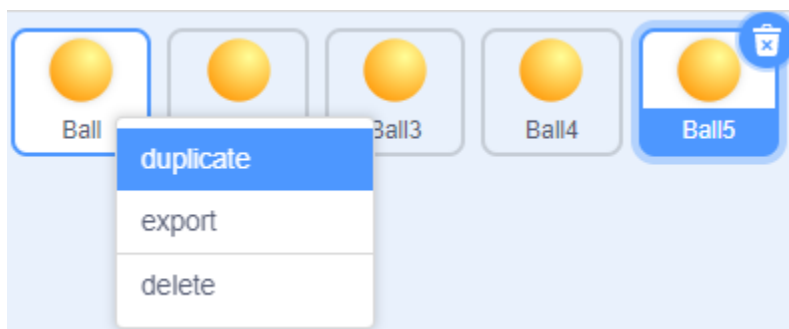
8.6.4 Programmierung

1. Sprite auswählen

Löschen Sie das Standard-Sprite und wählen Sie das **Ball**-Sprite aus.

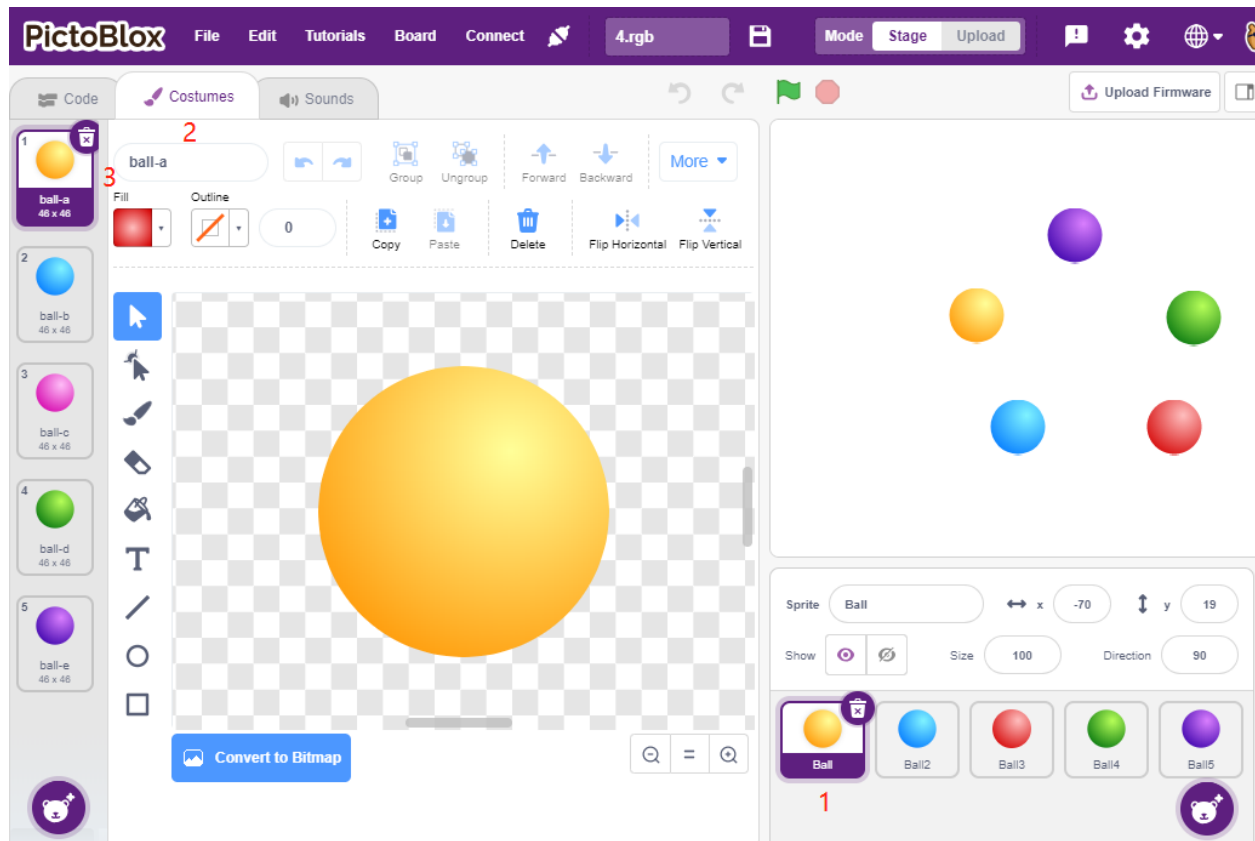


Und duplizieren Sie es 5 Mal.



Wählen Sie verschiedene Kostüme für diese 5 **Ball**-Sprites aus und verschieben Sie sie an die entsprechenden Positionen.

Bemerkung: Die Kostümfarbe des **Ball3**-Sprites muss manuell auf Rot geändert werden.

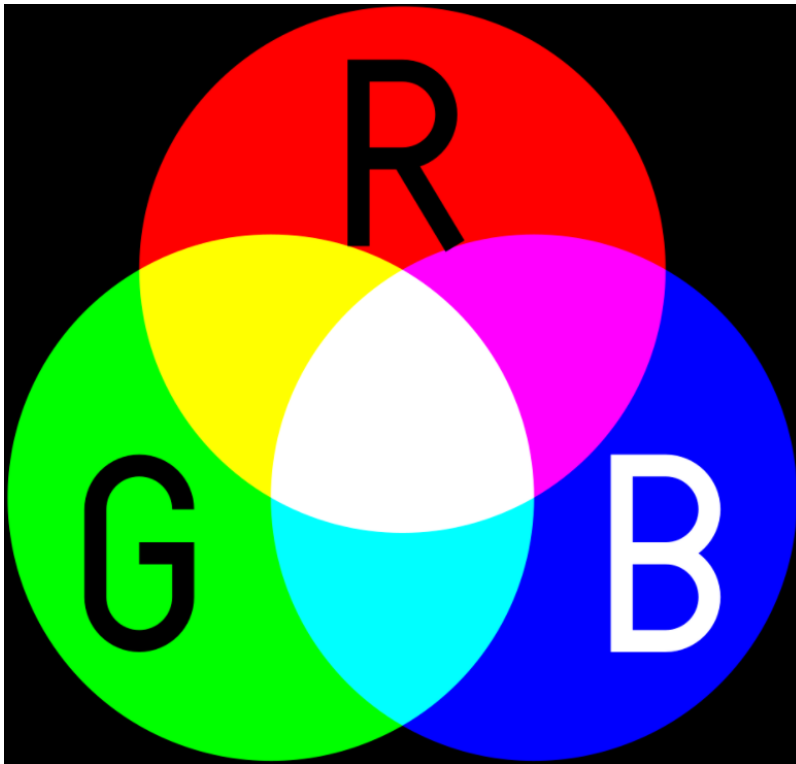


2. RGB-LEDs in der entsprechenden Farbe leuchten lassen

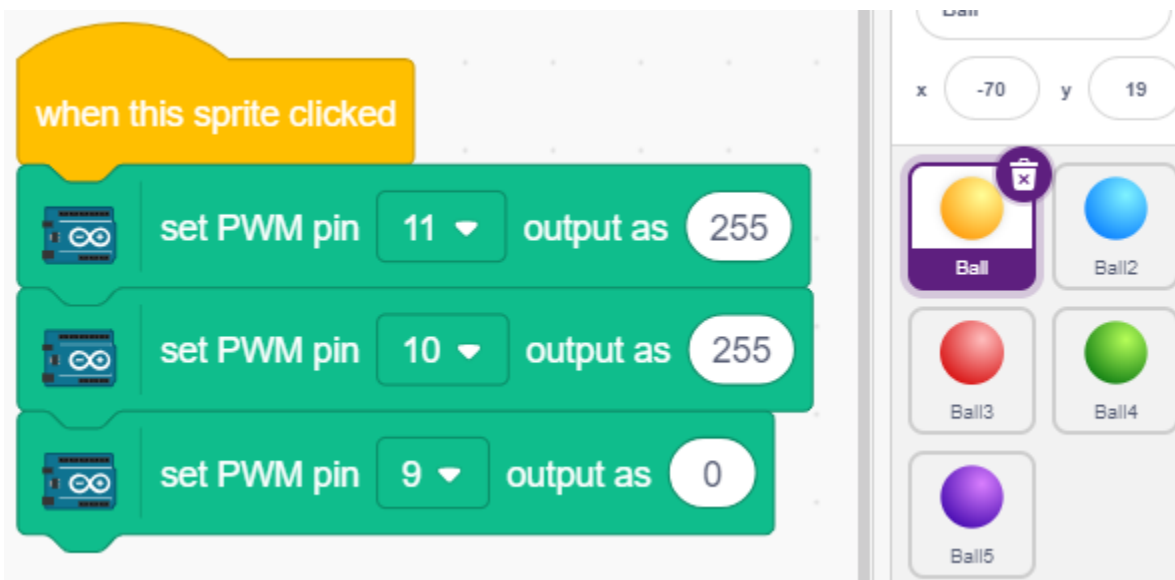
Bevor wir den Code verstehen, müssen wir das **RGB-Farbmodell** verstehen.

Das RGB-Farbmodell ist ein additives Farbmodell, bei dem Rot, Grün und Blau in verschiedenen Kombinationen hinzugefügt werden, um eine Vielzahl von Farben wiederzugeben.

Additive Farbmischung: Rot zu Grün ergibt Gelb; Grün zu Blau ergibt Cyan; Blau zu Rot ergibt Magenta; alle drei Grundfarben zusammen ergeben Weiß.



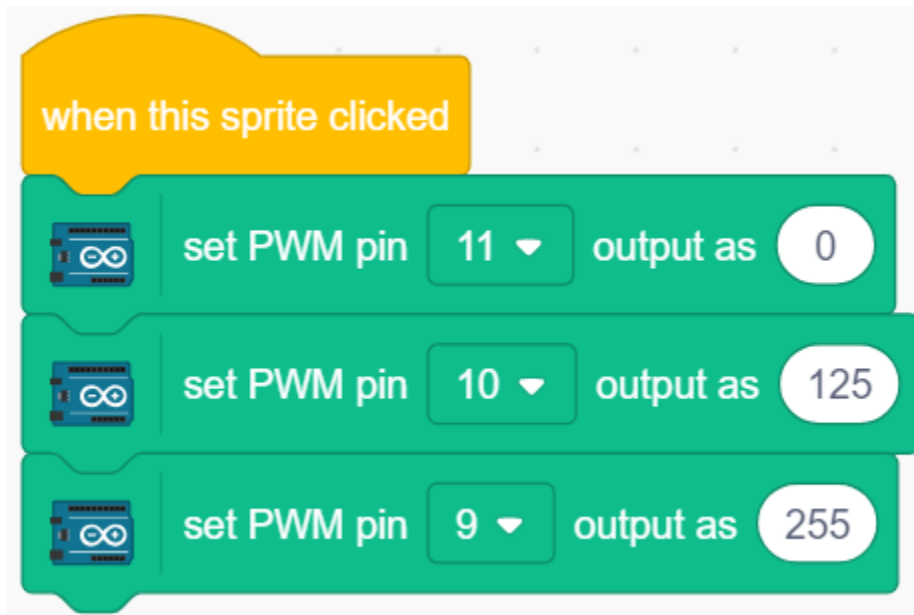
Der Code, um die RGB-LED gelb leuchten zu lassen, lautet also wie folgt.



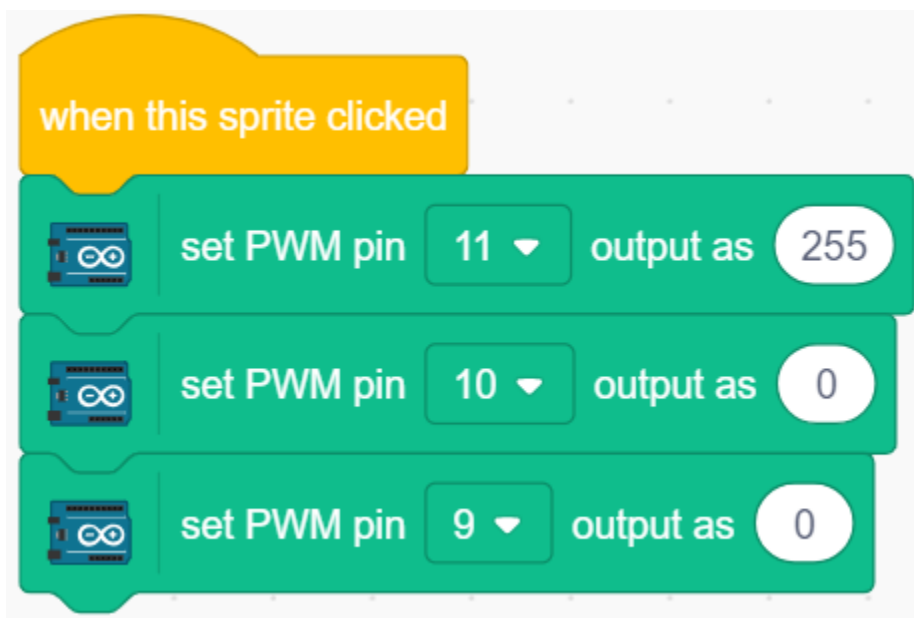
Wenn auf das Ball-Sprite (gelber Ball) geklickt wird, setzen wir Pin 11 hoch (rote LED an), Pin 10 hoch (grüne LED an) und Pin 9 niedrig (blaue LED aus), damit die RGB-LED gelb leuchtet.

Sie können auf die gleiche Weise Codes für andere Sprites schreiben, damit die RGB-LEDs in den entsprechenden Farben leuchten.

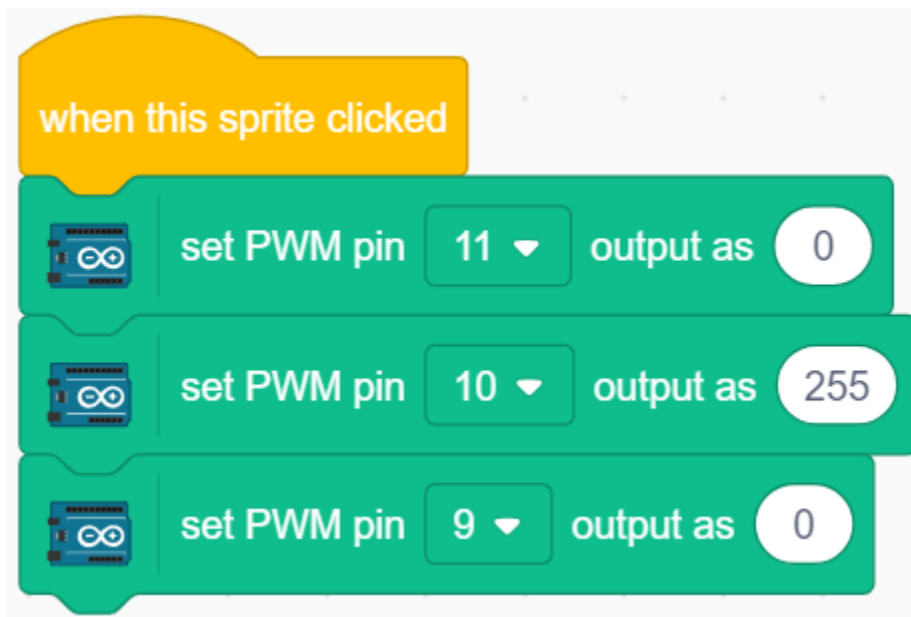
3. Ball2-Sprite (hellblau)



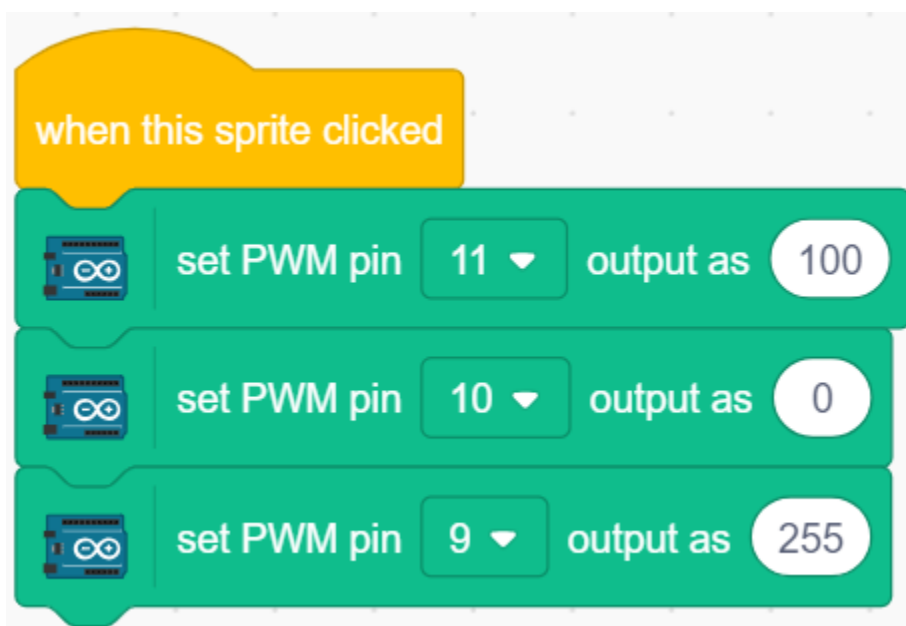
4. Ball3-Sprite (rot)



5. Ball4-Sprite (grün)



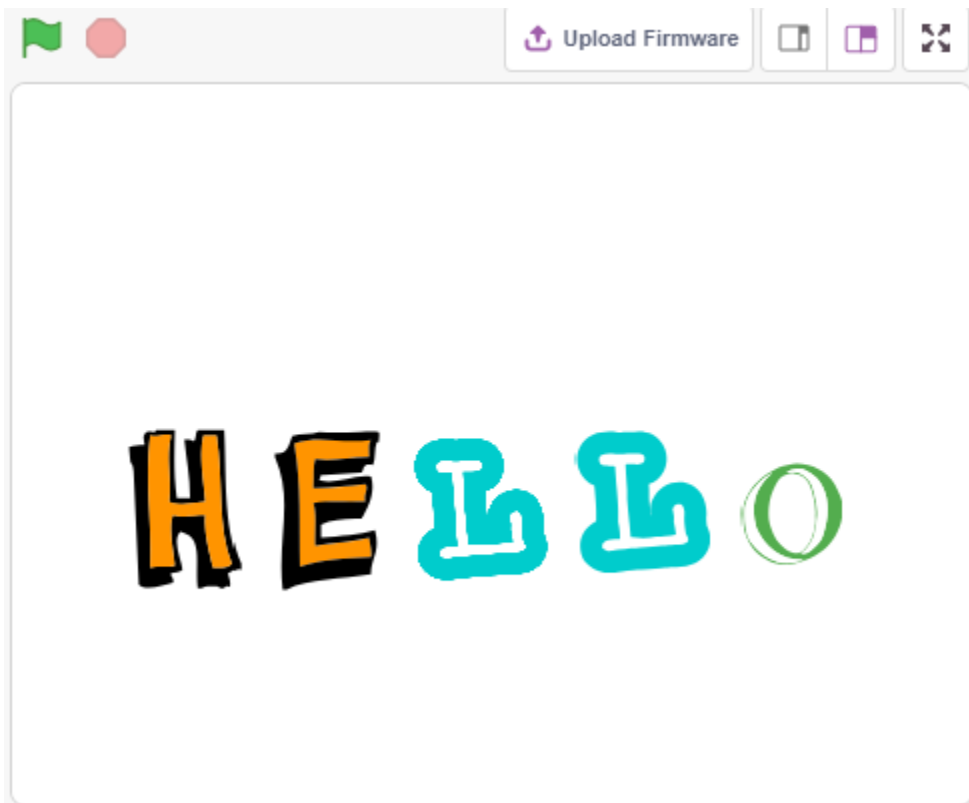
6. Ball5-Sprite (lila)



8.7 2.4 LCD1602

LCD1602 kann 2x16 Zeichen anzeigen. Jetzt lassen wir es die entsprechenden Zeichen anzeigen, die den Zeichen-Sprites auf der Bühne entsprechen.

Wenn Sie nacheinander auf „Hello“ auf der Bühne klicken, haben diese unterschiedliche Animationseffekte und die Zeichen werden gleichzeitig auf dem LCD1602 angezeigt.



8.7.1 Was Sie lernen werden

- Wie man das LCD1602 verwendet
- Mehrere unterschiedliche Sprites auswählen
- Größe, Drehwinkel und Farbe des Sprites ändern sowie Anzeigen oder Verbergen.

8.7.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

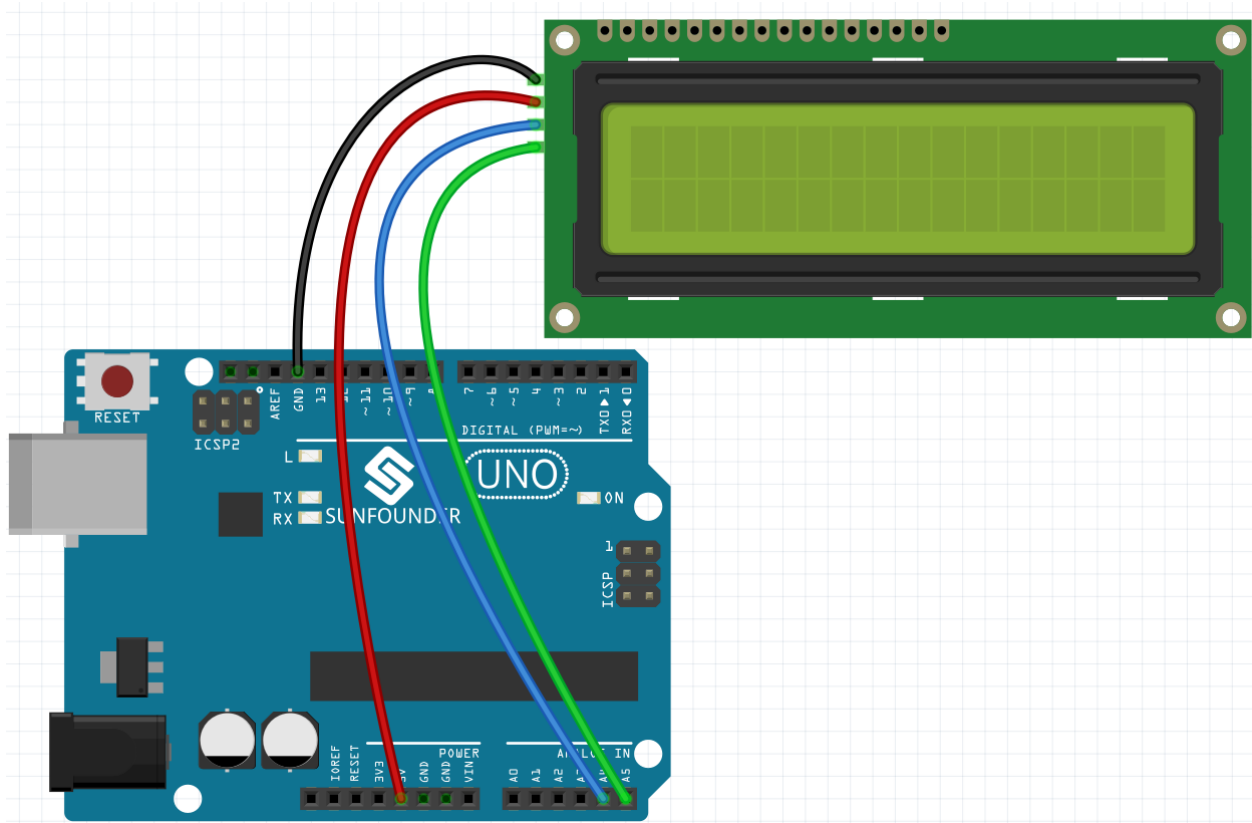
Es ist definitiv praktisch, ein komplettes Set zu kaufen, hier ist der Link:

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>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>I2C LCD1602</i>	

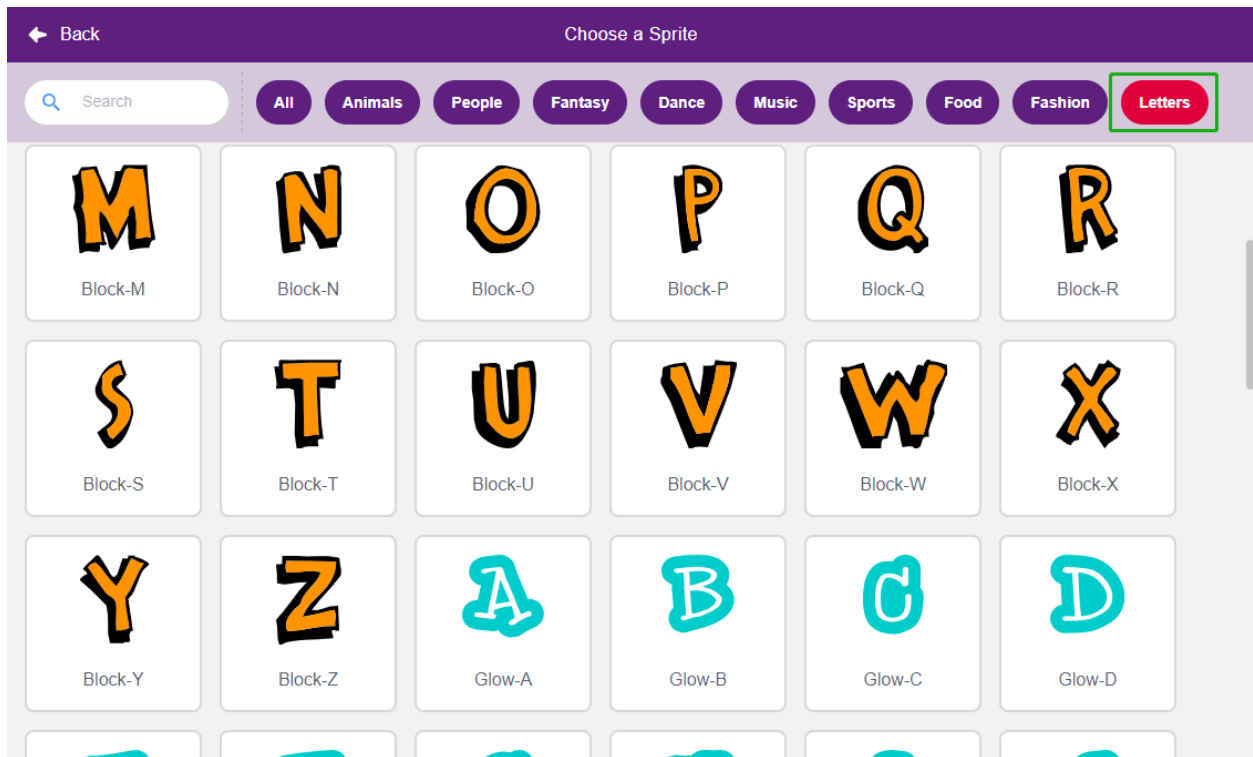
8.7.3 Schaltung aufbauen



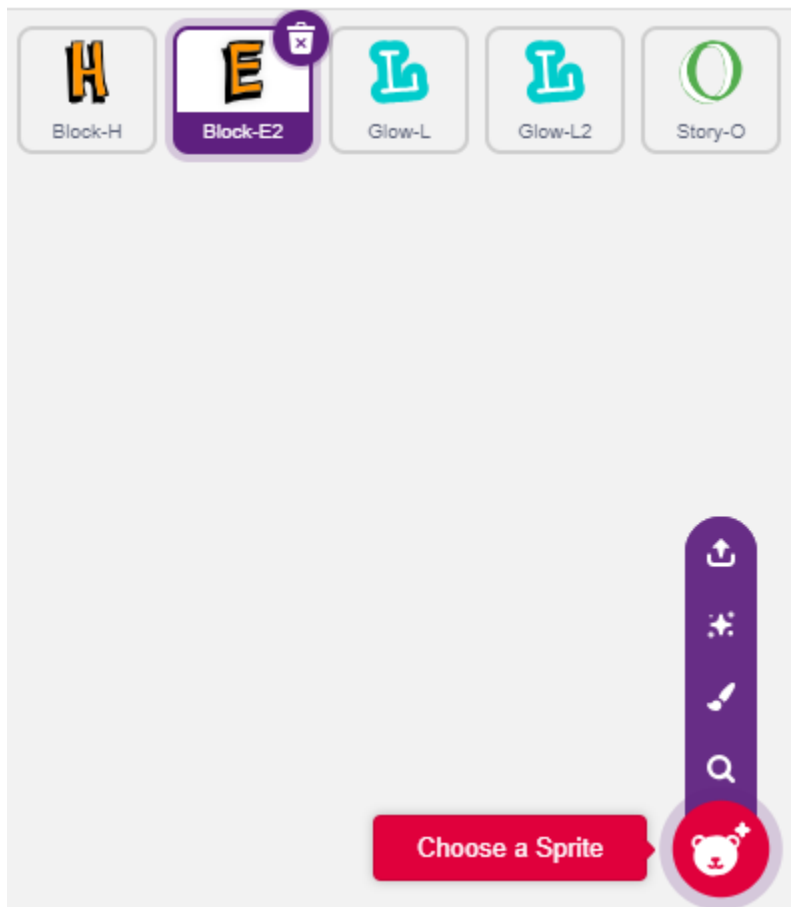
8.7.4 Programmierung

1. Sprite auswählen

Löschen Sie das Standard-Sprite, klicken Sie auf **Choose a Sprite**, dann auf **letters** und wählen Sie das gewünschte Sprite aus.



Zum Beispiel habe ich „Hello“ gewählt, wie unten gezeigt.



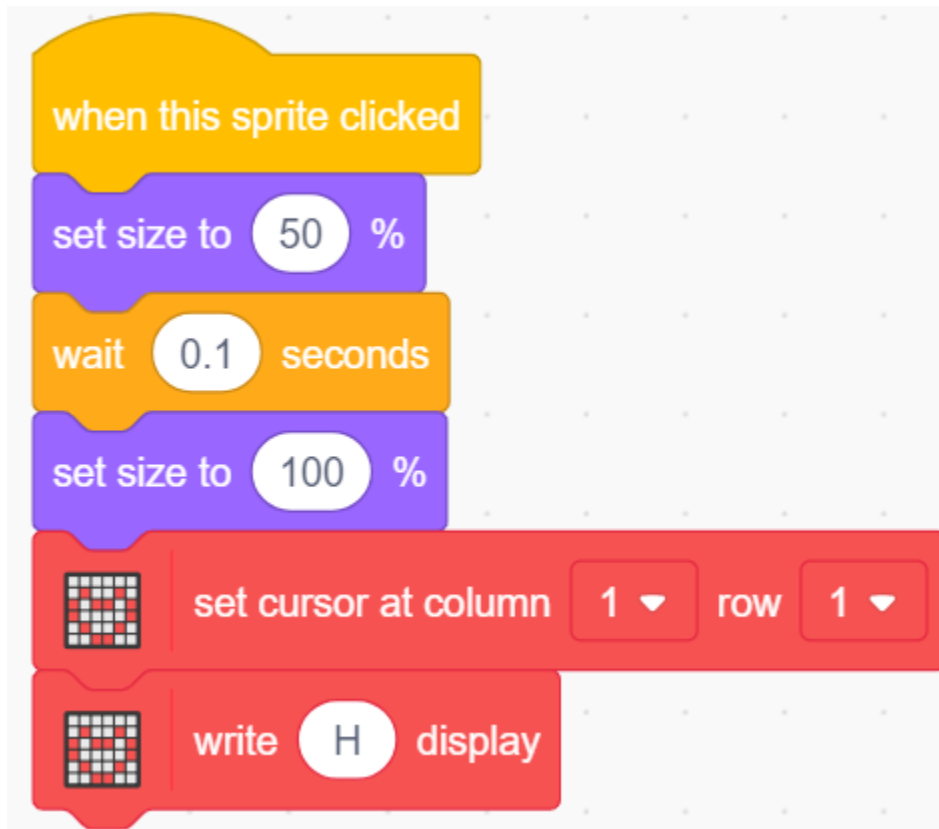
Legen Sie jetzt verschiedene Effekte für diese Sprites fest und zeigen Sie sie beim Klicken auf dem LCD1602 an.

2. H wird vergrößert und verkleinert

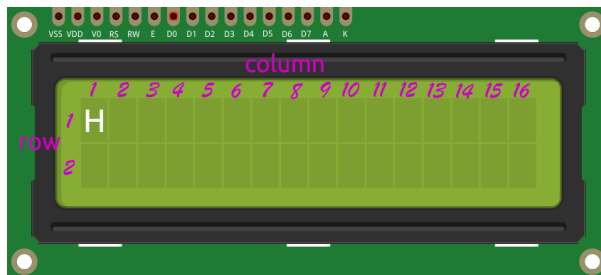
Klicken Sie auf das Sprite **H** und schreiben Sie jetzt ein Skript dafür.

Wenn das Sprite **H** angeklickt wird, setzen Sie seine Größe auf 50%, dann stellen Sie es wieder her; während H in der ersten Zeile und Spalte des LCD1602 angezeigt wird.

- [set size to]: Aus der **Aussehen**-Palette, zum Einstellen der Größe des Sprites von 0% bis 100%.
- [set cursor at column row]: Aus der **Display Module**-Palette, um den Cursor in einer bestimmten Zeile des LCD1602 zu positionieren und Zeichen anzuzeigen.
- [write display]: Aus der **Display Module**-Palette, zum Anzeigen von Zeichen oder Zeichenketten auf dem LCD1602.



Die Verteilung von Zeilen und Spalten auf dem LCD1602 wird in der Abbildung gezeigt.

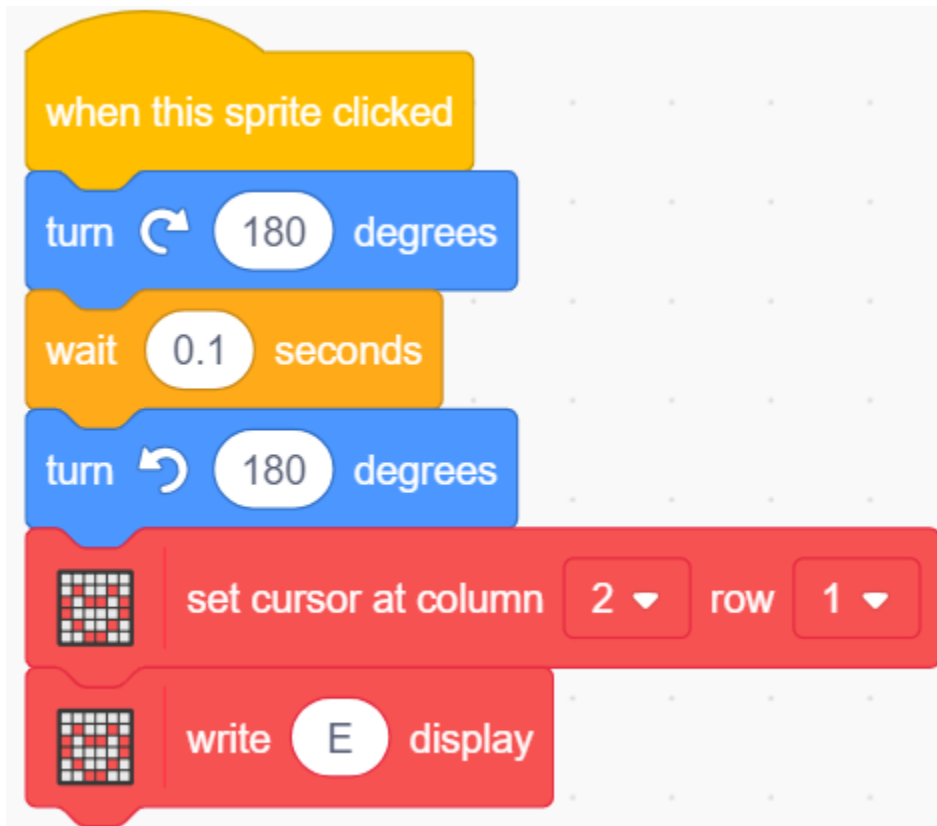


3. E kippt nach links und rechts

Klicken Sie auf das Sprite **E** und schreiben Sie jetzt ein Skript dafür.

Wenn das Sprite **E** angeklickt wird, lassen Sie es 180 Grad im Uhrzeigersinn drehen, dann 180 Grad gegen den Uhrzeigersinn, so dass Sie es nach links und rechts kippen sehen; und zeigen Sie H in der ersten Zeile und Spalte 2 des LCD1602 an.

- [turn degrees]: Aus der **Bewegung**-Palette, zum Drehen des Sprites im oder gegen den Uhrzeigersinn im Bereich von 0-360 Grad.

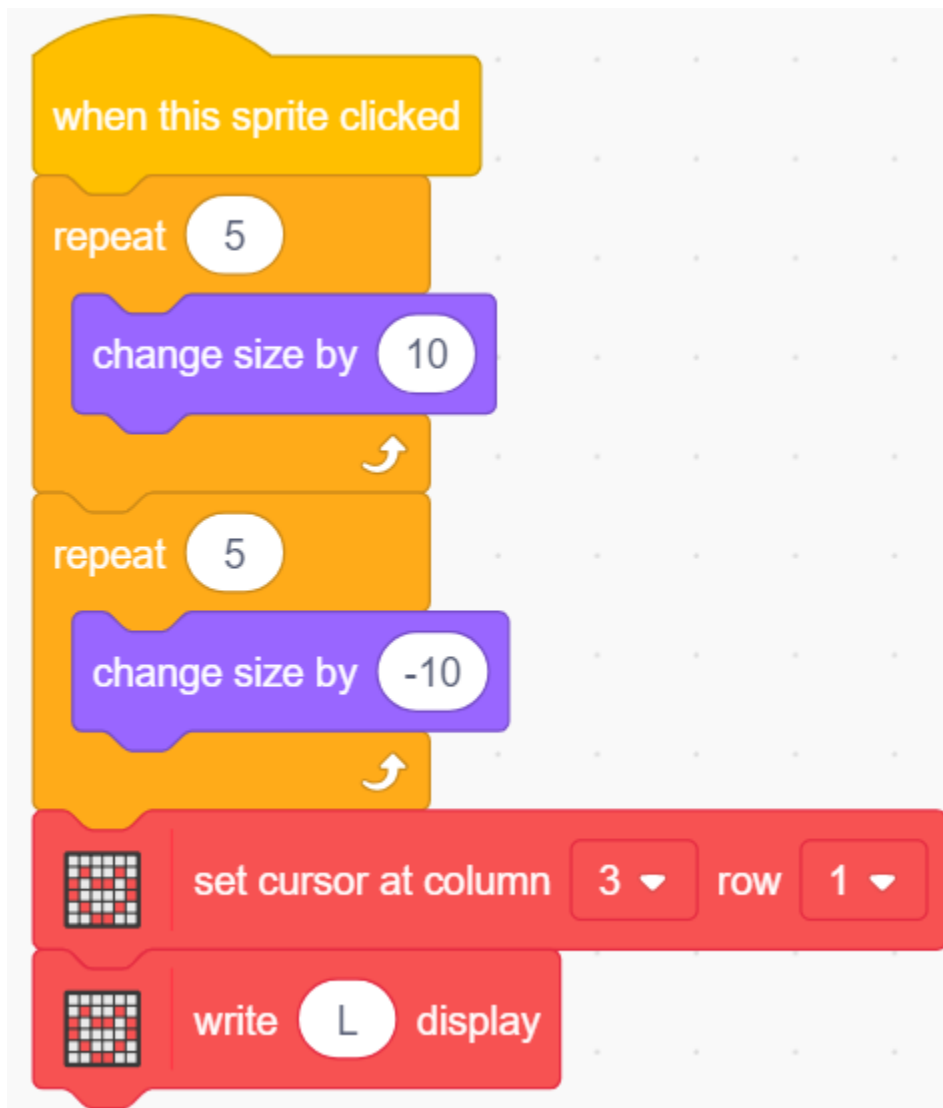


4. L verkleinert sich langsam und zoomt herein

Klicken Sie auf das **erste L** Sprite und schreiben Sie jetzt ein Skript dafür.

Wenn das Sprite **L** angeklickt wird, verwenden Sie den [repeat]-Block, um seine Größe um 50% zu erhöhen (5 Mal, jedes Mal um 10), dann verkleinern Sie es wieder auf seine ursprüngliche Größe auf die gleiche Weise, während L in der ersten Zeile und Spalte 3 des LCD1602 angezeigt wird.

- [change size by]: Aus der Bewegungspalette, um die Größe des Sprites zu ändern.

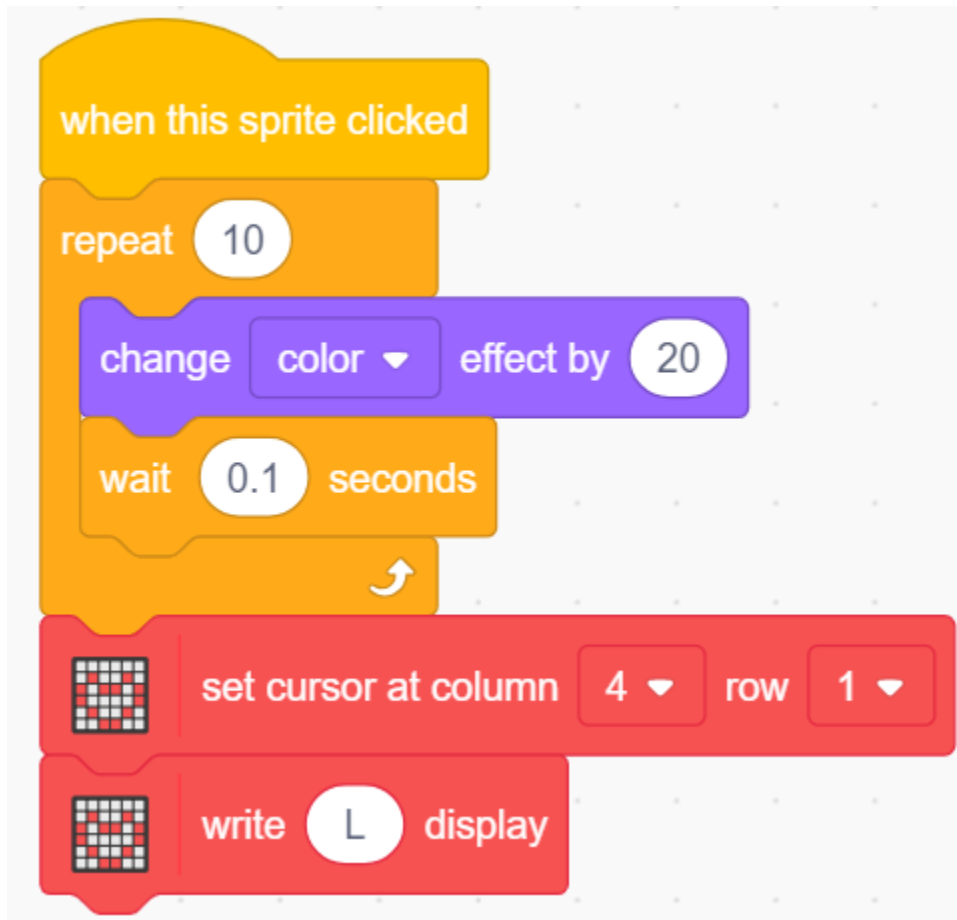


5. Das zweite L ändert die Farbe

Klicken Sie auf das **zweite L** Sprite und schreiben Sie jetzt ein Skript dafür.

Wenn das Sprite **L** angeklickt wird, verwenden Sie den [repeat]-Block, um 10 Mal mit einer Rate von 20 Schritten die Farben zu wechseln und zur ursprünglichen Farbe zurückzukehren. Zeigen Sie auch L in der ersten Zeile und Spalte 4 des LCD1602 an.

- [change color effect by]: Zum Ändern des Farbeffekts, ein Kostüm kann mit dem Farbeffekt 200 verschiedene Farbschemata annehmen, 0 und 200 sind dieselbe Farbe.

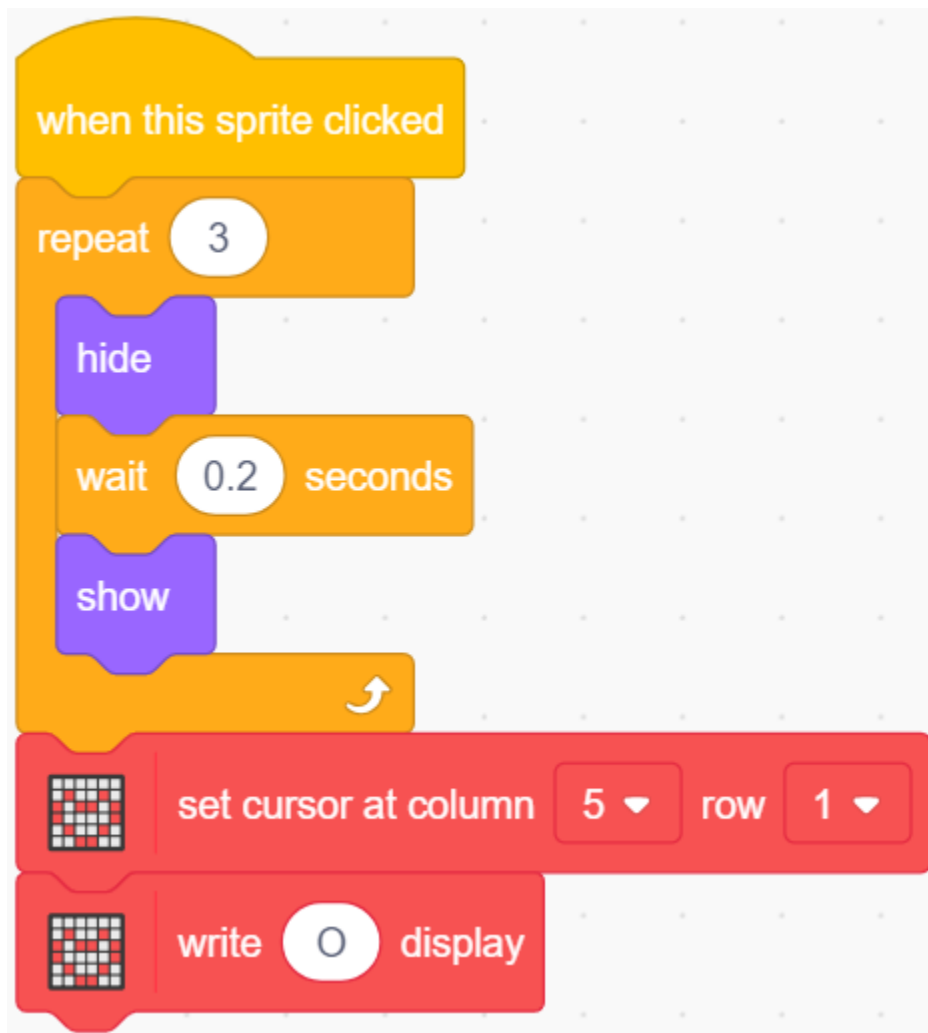


6. O versteckt und zeigt

Klicken Sie auf das Sprite **O** und schreiben Sie jetzt ein Skript dafür.

Wenn das Sprite **O** angeklickt wird, wiederholt es den Vorgang Verbergen und Anzeigen 3 Mal, während O in der ersten Zeile und Spalte 5 des LCD1602 angezeigt wird.

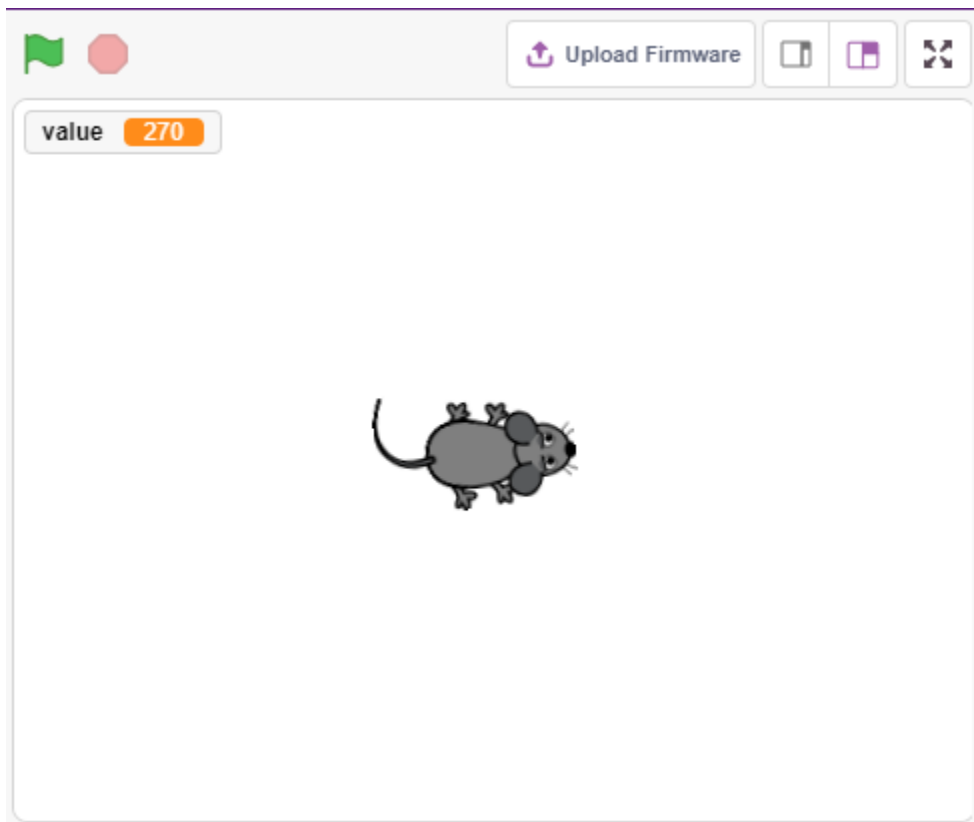
- [Hide] & [Show]: lässt das Sprite sich verbergen und anzeigen.



8.8 2.5 Bewegende Maus

Heute werden wir ein Mäusespielzeug erstellen, das mit einem Potentiometer gesteuert wird.

Wenn man auf die grüne Flagge klickt, bewegt sich die Maus auf der Bühne vorwärts. Dreht man am Potentiometer, ändert die Maus die Bewegungsrichtung.



8.8.1 Was Sie lernen werden

- Prinzip des Potentiometers
- Analogen Pin und dessen Bereich lesen
- Einen Bereich auf einen anderen abbilden
- Bewegen und Ändern der Richtung des Sprites

8.8.2 Benötigte Komponenten

Für dieses Projekt benötigen wir folgende Komponenten.

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

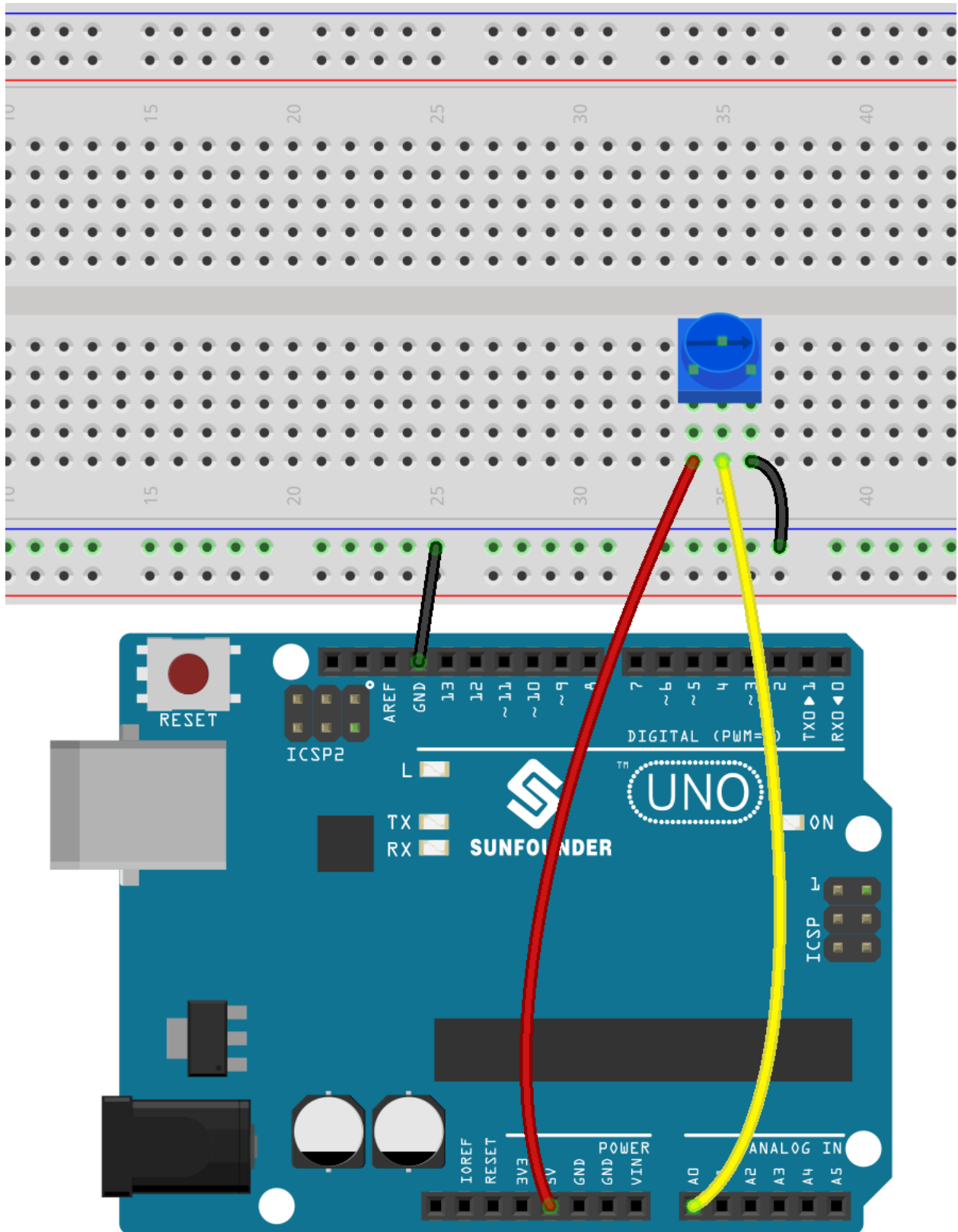
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Potentiometer</i>	

8.8.3 Schaltung aufbauen

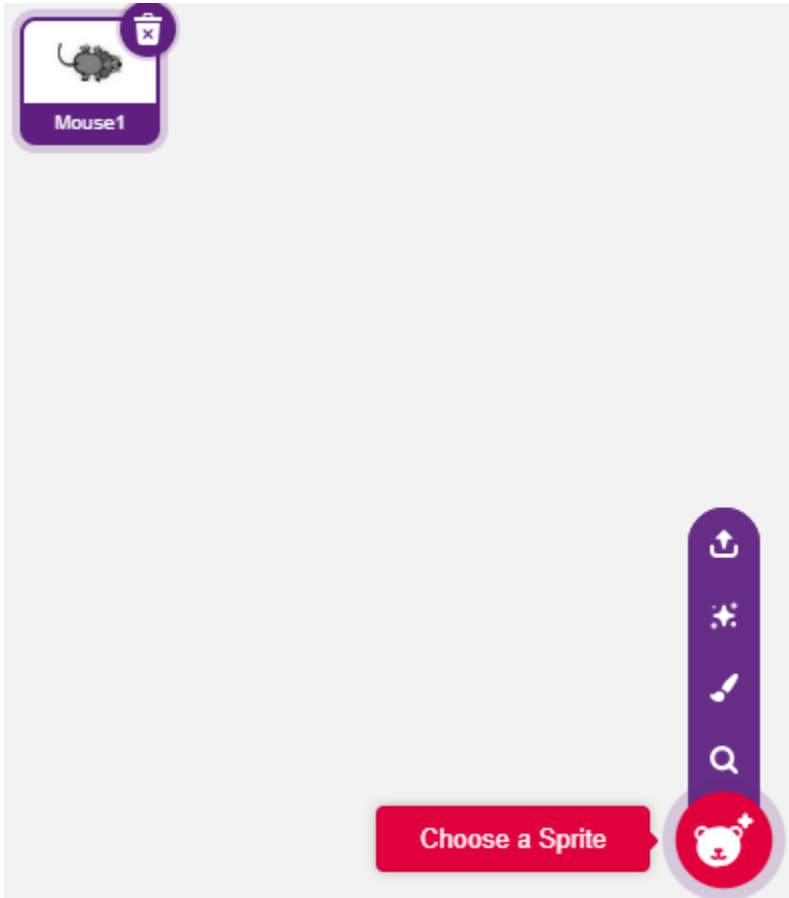
Das Potentiometer ist ein resistives Element mit 3 Anschlüssen. Die 2 seitlichen Pins werden an 5V und GND angeschlossen, und der mittlere Pin wird an A0 angeschlossen. Nach der Umwandlung durch den ADC-Konverter des Arduino-Boards liegt der Wertebereich zwischen 0-1023.



8.8.4 Programmierung

1. Wählen Sie einen Sprite

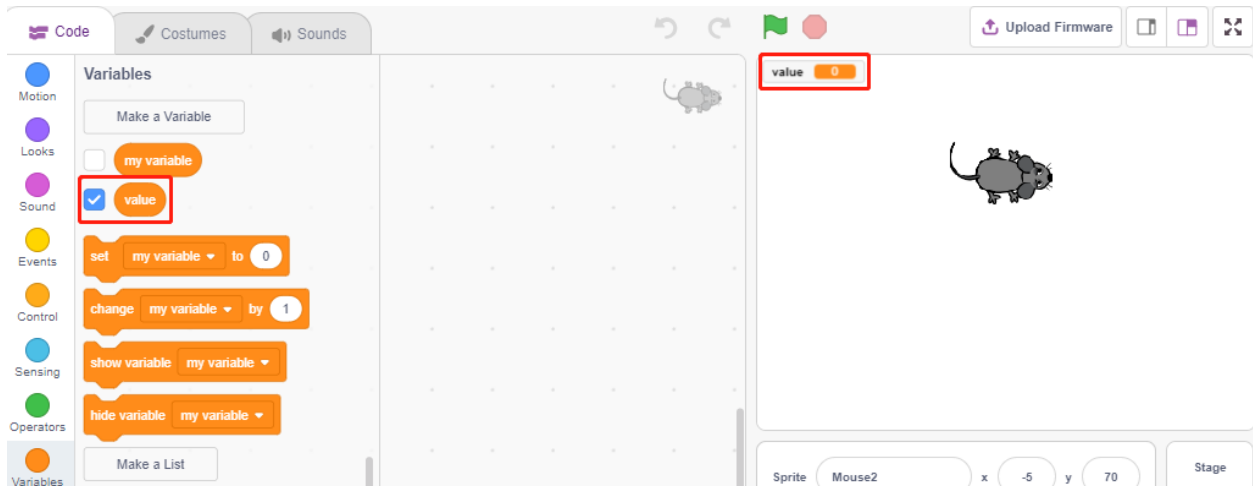
Löschen Sie den Standard-Sprite, klicken Sie unten rechts im Sprite-Bereich auf die Schaltfläche **Choose a Sprite**, geben Sie **mouse** in das Suchfeld ein und klicken Sie dann, um ihn hinzuzufügen.



2. Eine Variable erstellen

Erstellen Sie eine Variable namens **value**, um den gelesenen Wert des Potentiometers zu speichern.

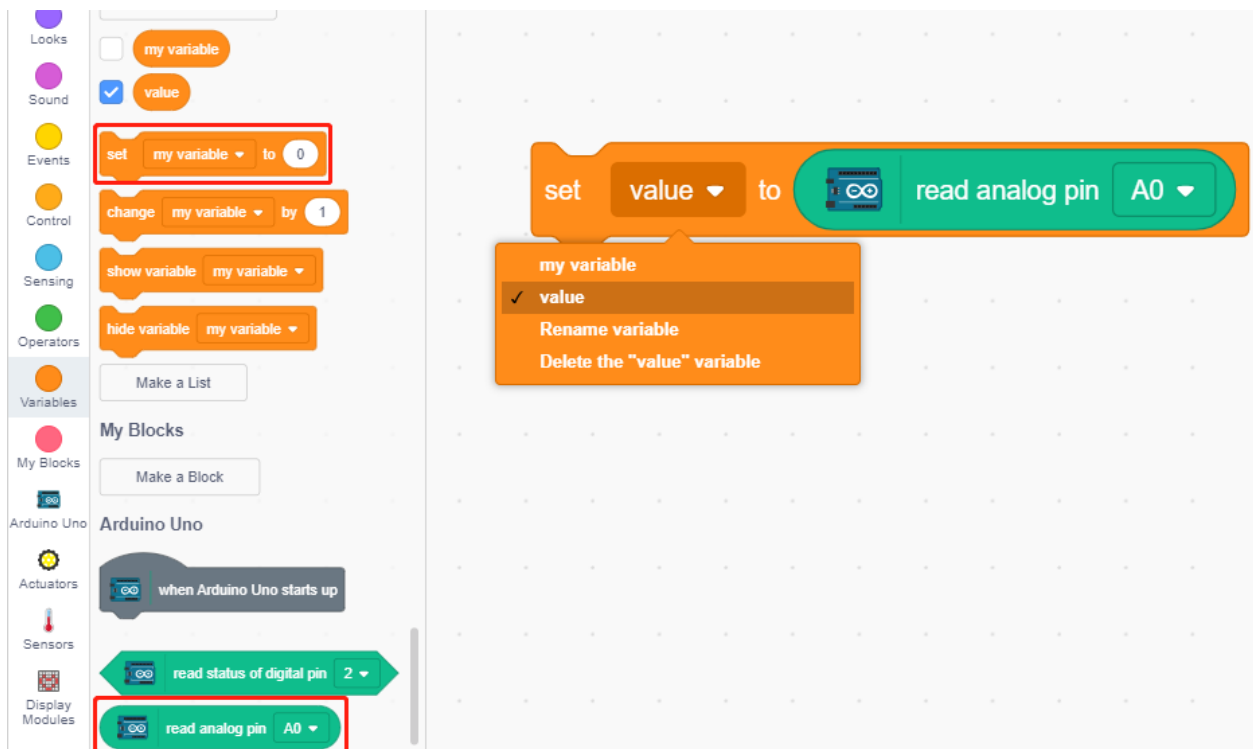
Nach dem Erstellen sehen Sie **value** im **Variables**-Bereich und im markierten Zustand, was bedeutet, dass diese Variable auf der Bühne angezeigt wird.



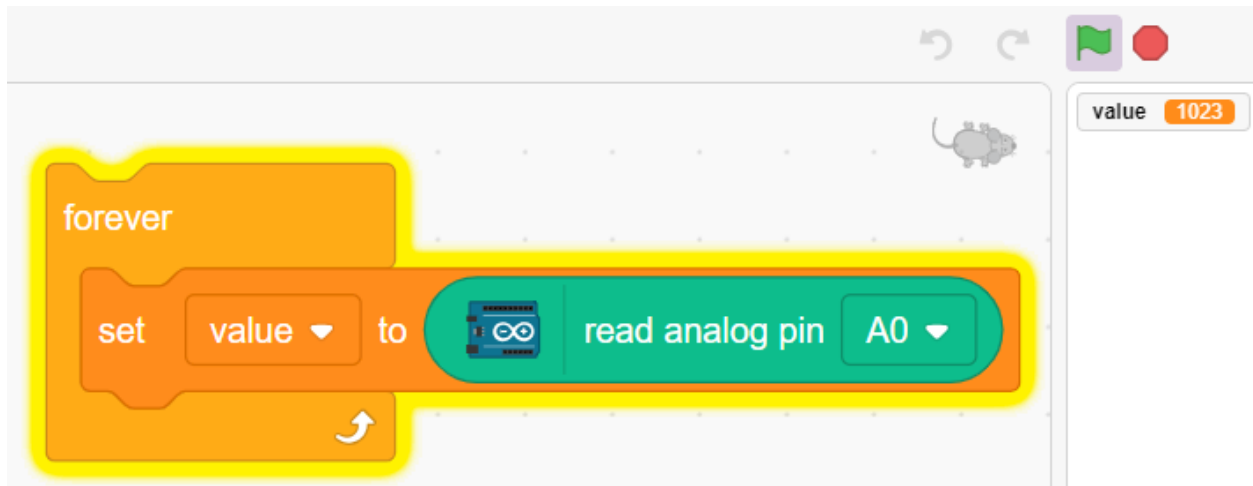
3. Den Wert von A0 lesen

Speichern Sie den gelesenen Wert von A0 in der Variable **value**.

- [set my variable to 0]: Den Wert der Variable festlegen.
- [read analog pin A0]: Den Wert von A0~A5 im Bereich von 0-1023 lesen.

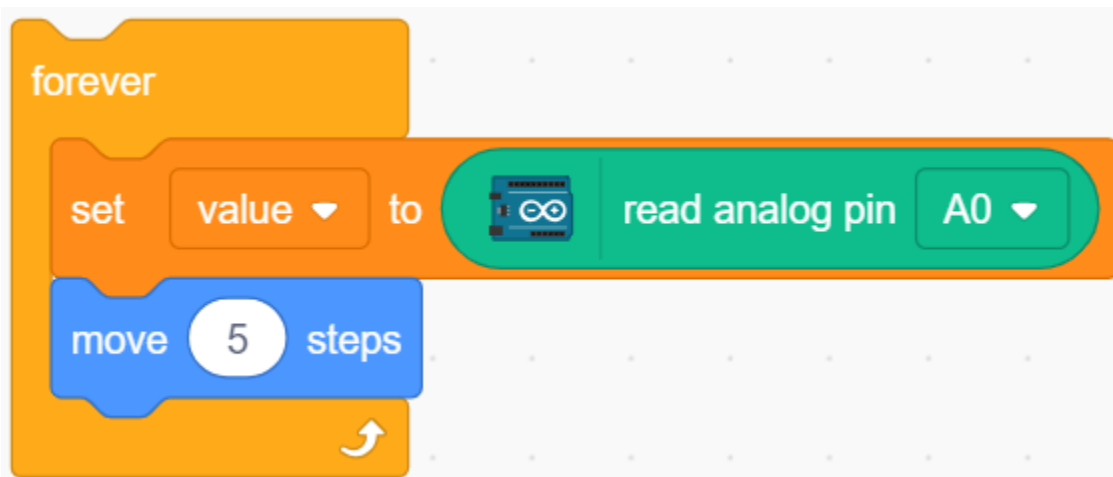


Um durchgehend lesen zu können, müssen Sie den [forever]-Block verwenden. Klicken Sie auf dieses Skript, um es auszuführen, drehen Sie das Potentiometer in beide Richtungen, und Sie werden sehen, dass der Wertebereich 0-1023 beträgt.



4. Den Sprite bewegen

Verwenden Sie den [move steps]-Block, um den Sprite zu bewegen. Führen Sie das Skript aus, und Sie werden sehen, dass sich der Sprite von der Mitte nach rechts bewegt.

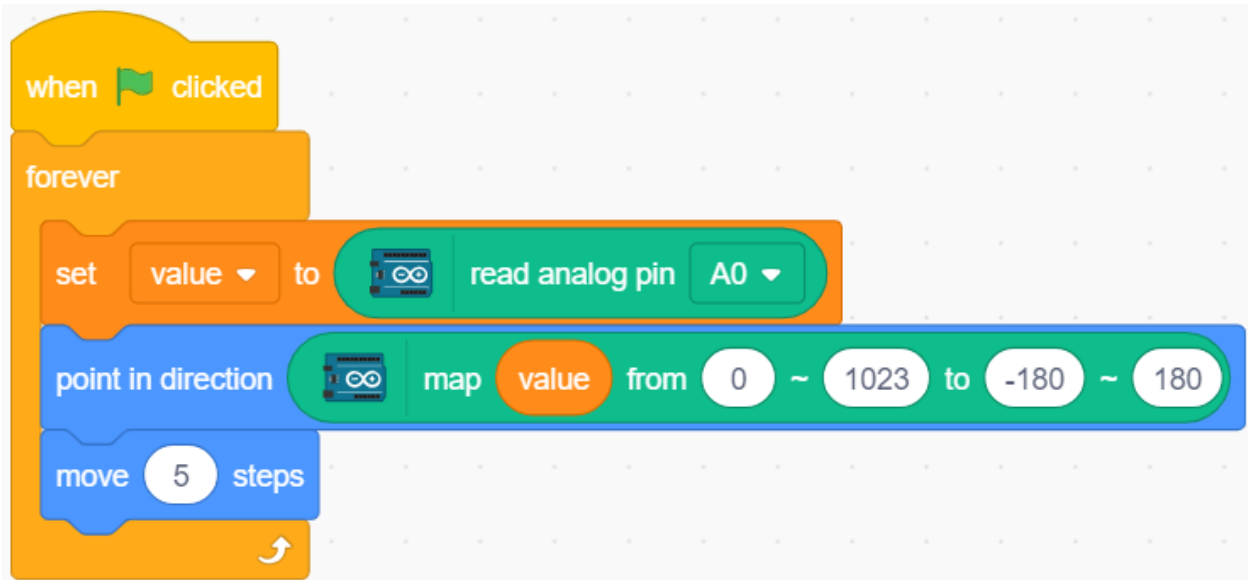


5. Die Richtung des Sprites ändern

Ändern Sie nun die Bewegungsrichtung des Sprites entsprechend dem Wert von A0. Da der Wert von A0 von 0-1023 reicht, aber die Drehrichtung des Sprites von -180~180 ist, muss ein [map]-Block verwendet werden.

Fügen Sie auch [when green flag clicked] am Anfang hinzu, um das Skript zu starten.

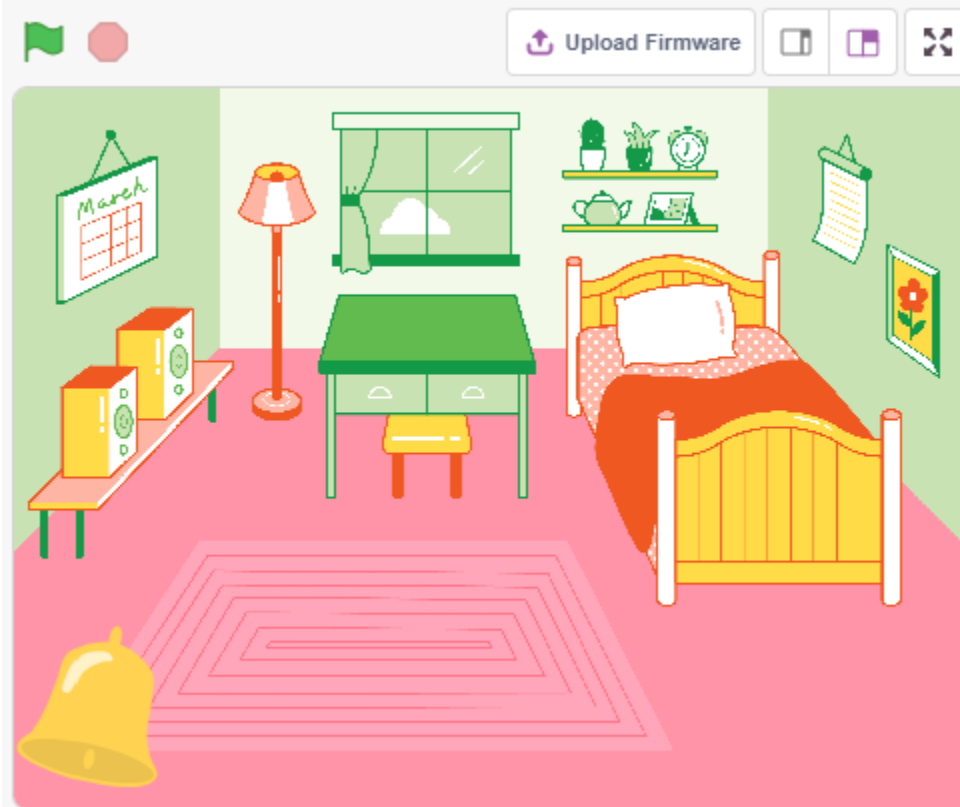
- [point in direction]: Den Lenkwinkel des Sprites festlegen, aus der **Bewegung**-Palette.
- [map from to]: Einen Bereich auf einen anderen abbilden.



8.9 2.6 Türklingel

Hier werden wir den Knopf und die Klingel auf der Bühne verwenden, um eine Türklingel zu erstellen.

Nachdem Sie auf die grüne Flagge geklickt haben, können Sie den Knopf drücken und die Klingel auf der Bühne wird einen Ton von sich geben.



8.9.1 Was Sie lernen werden

- Wie der Knopf funktioniert
- Digitalen Pin und Bereiche lesen
- Eine bedingte Schleife erstellen
- Einen Hintergrund hinzufügen
- Ton abspielen

8.9.2 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.

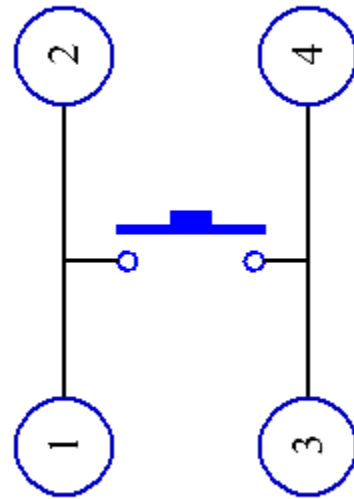
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Knopf</i>	
<i>Kondensator</i>	

8.9.3 Schaltung aufbauen

Der Knopf ist ein 4-Pin-Gerät, da der Pin 1 mit Pin 2 und Pin 3 mit Pin 4 verbunden ist. Wenn der Knopf gedrückt wird, werden die 4 Pins verbunden und der Schaltkreis geschlossen.



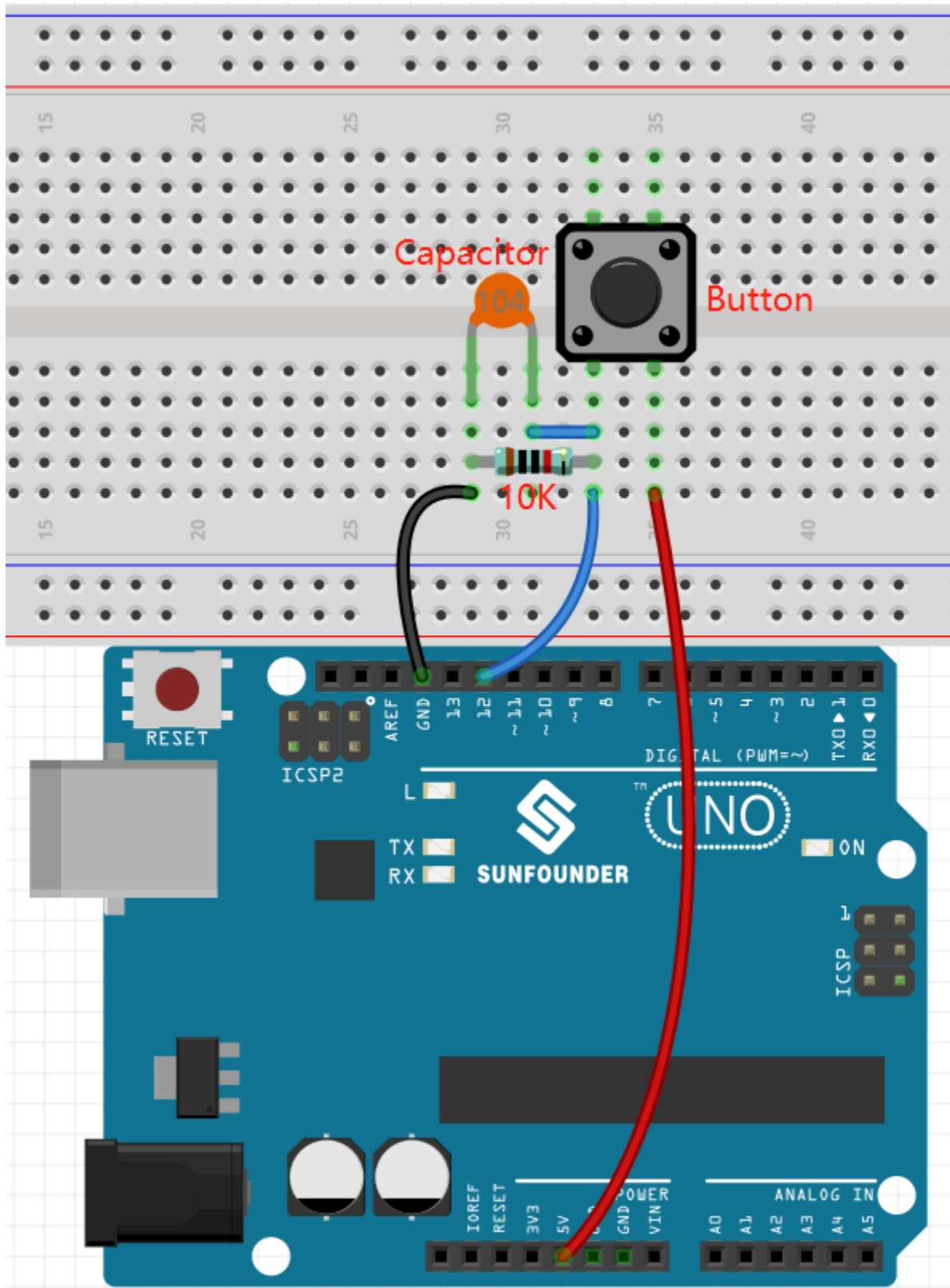
Button



Internal Structure

Bauen Sie den Schaltkreis entsprechend dem folgenden Diagramm auf.

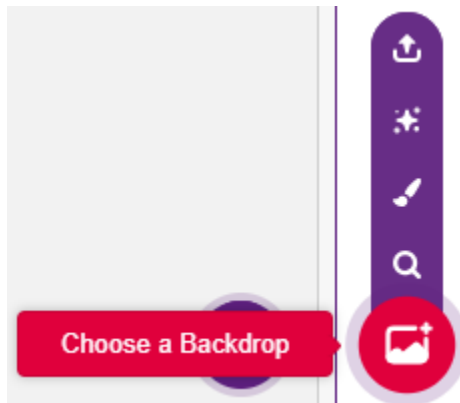
- Verbinden Sie einen der Pins auf der linken Seite des Knopfes mit Pin 12, der mit einem Pull-down-Widerstand und einem 0,1 μ F (104) Kondensator verbunden ist (um Schwankungen zu eliminieren und ein stabiles Signal auszugeben, wenn der Knopf in Betrieb ist).
- Verbinden Sie das andere Ende des Widerstands und des Kondensators mit GND und einen der Pins auf der rechten Seite des Knopfes mit 5V.



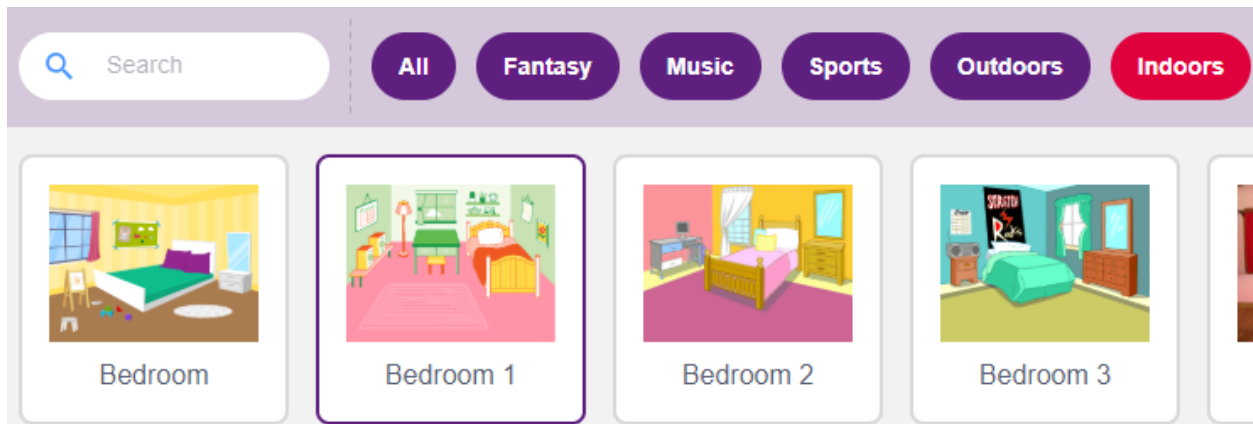
8.9.4 Programmierung

1. Einen Hintergrund hinzufügen

Klicken Sie auf den **Choose a Backdrop**-Button in der unteren rechten Ecke.

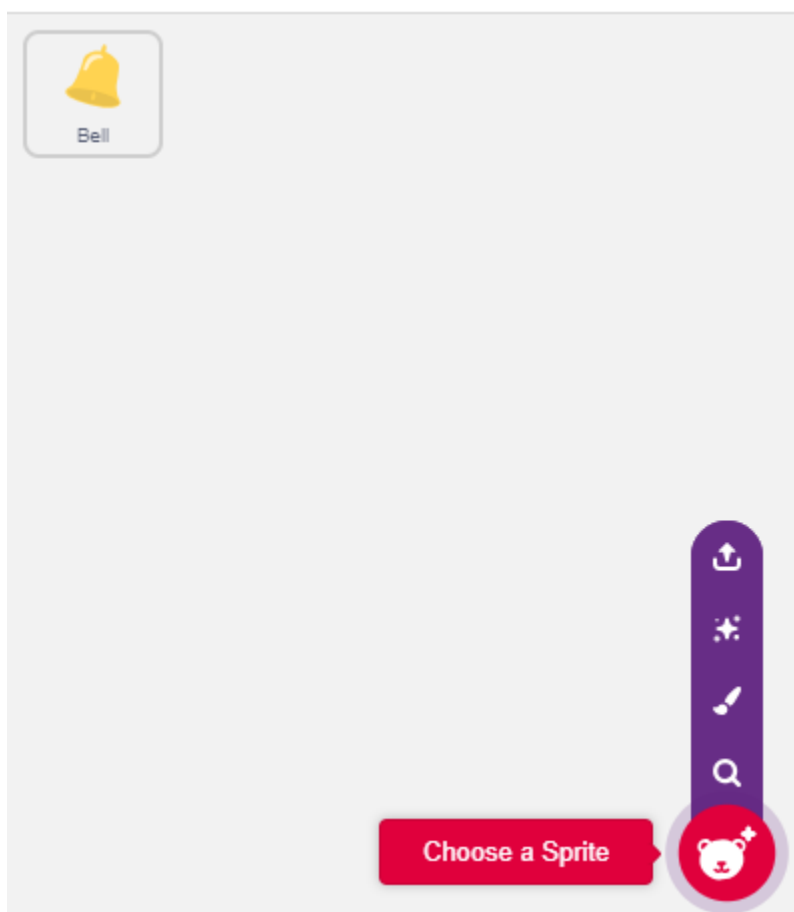


Wählen Sie **Bedroom 1** aus.

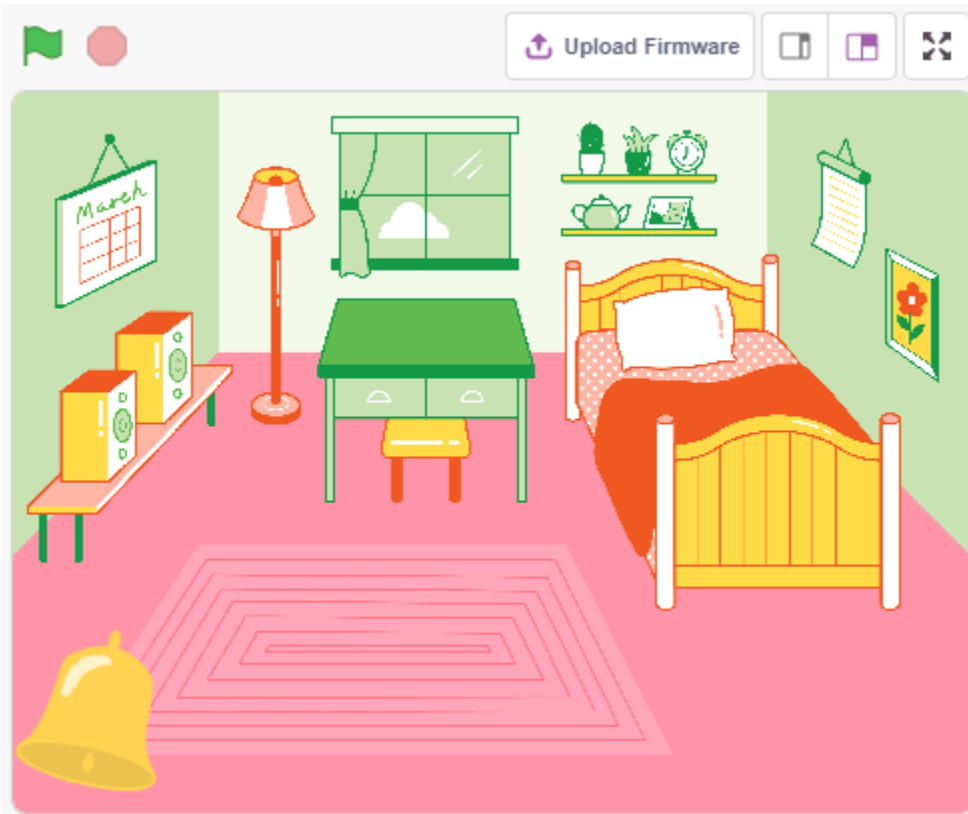


2. Das Sprite auswählen

Löschen Sie das Standard-Sprite, klicken Sie auf den **Choose a Sprite**-Button in der unteren rechten Ecke des Sprite-Bereichs, geben Sie **bell** in das Suchfeld ein und fügen Sie es dann hinzu.



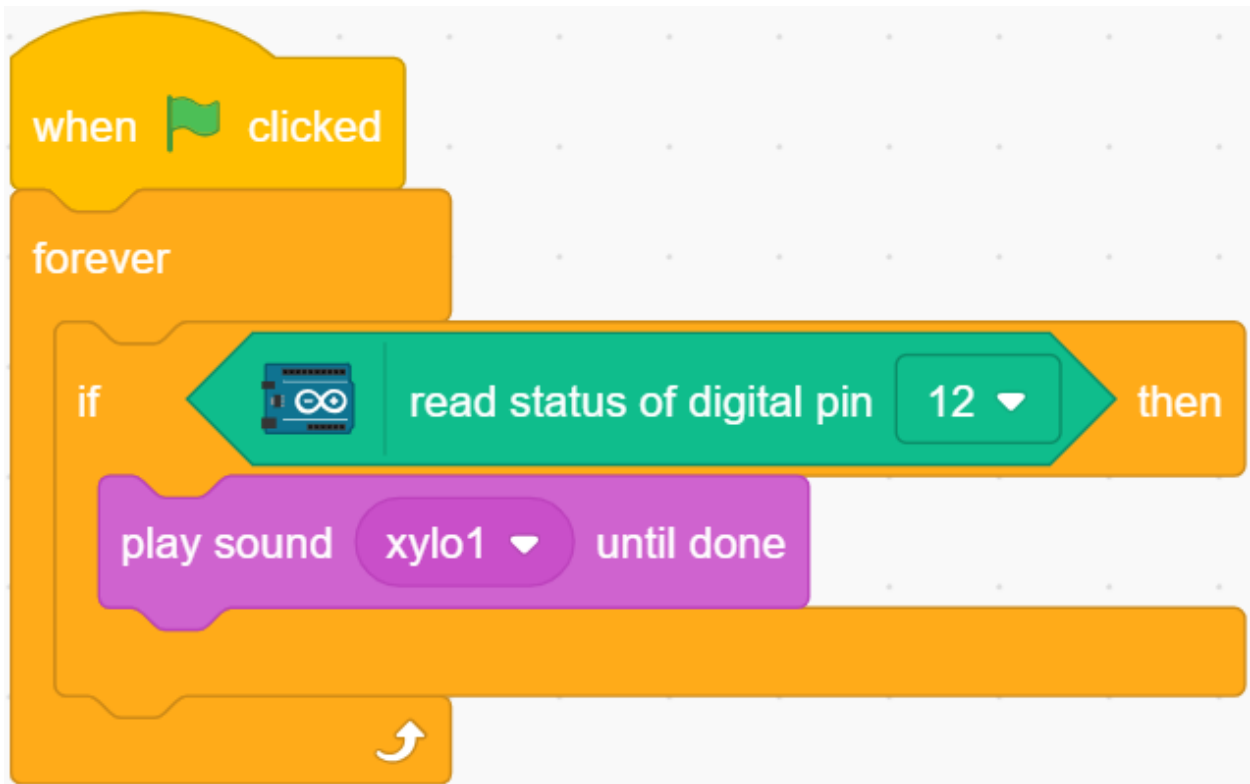
Wählen Sie dann das **bell**-Sprite auf der Bühne und verschieben Sie es an die richtige Position.



3. Drücken Sie den Knopf und die Glocke gibt einen Ton von sich

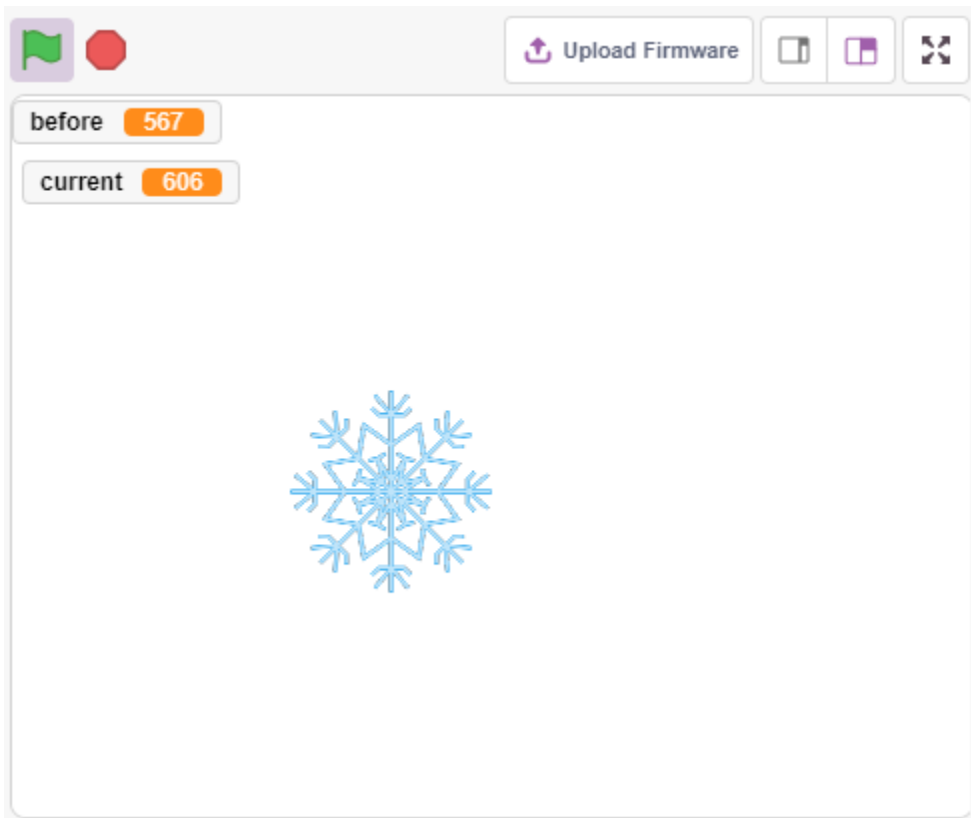
Verwenden Sie [wenn dann], um eine bedingte Anweisung zu erstellen, dass wenn der Wert des gelesenen Pin12 gleich 1 ist (der Schlüssel gedrückt wird), der Ton **xylo1** abgespielt wird.

- [read status of digital pin]: Dieser Block stammt aus der **Arduino Uno**-Palette und wird verwendet, um den Wert eines digitalen Pins zu lesen, das Ergebnis ist 0 oder 1.
- [if then]: Dieser Block ist ein Steuerblock und stammt aus der **Steuerung**-Palette. Wenn seine boolsche Bedingung wahr ist, werden die darin enthaltenen Blöcke ausgeführt und dann wird das beteiligte Skript fortgesetzt. Wenn die Bedingung falsch ist, werden die Skripte im Block ignoriert. Die Bedingung wird nur einmal überprüft; wenn die Bedingung während der Ausführung des Skripts im Block falsch wird, wird sie weiterhin ausgeführt, bis sie abgeschlossen ist.
- [play sound until done]: aus der Sound-Palette, verwendet um bestimmte Töne abzuspielen.



8.10 2.7 Niedrigtemperaturalarm

In diesem Projekt werden wir ein Alarmsystem für niedrige Temperaturen erstellen. Wenn die Temperatur unter einen bestimmten Schwellenwert fällt, erscheint das **Schneeflocke**-Symbol auf der Bühne.



8.10.1 Was Sie lernen werden

- Funktionsweise des Thermistors
- Multivariable und subtraktive Vorgänge

8.10.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

Es ist sicherlich praktisch, ein komplettes Set zu kaufen, hier 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>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Thermistor</i>	

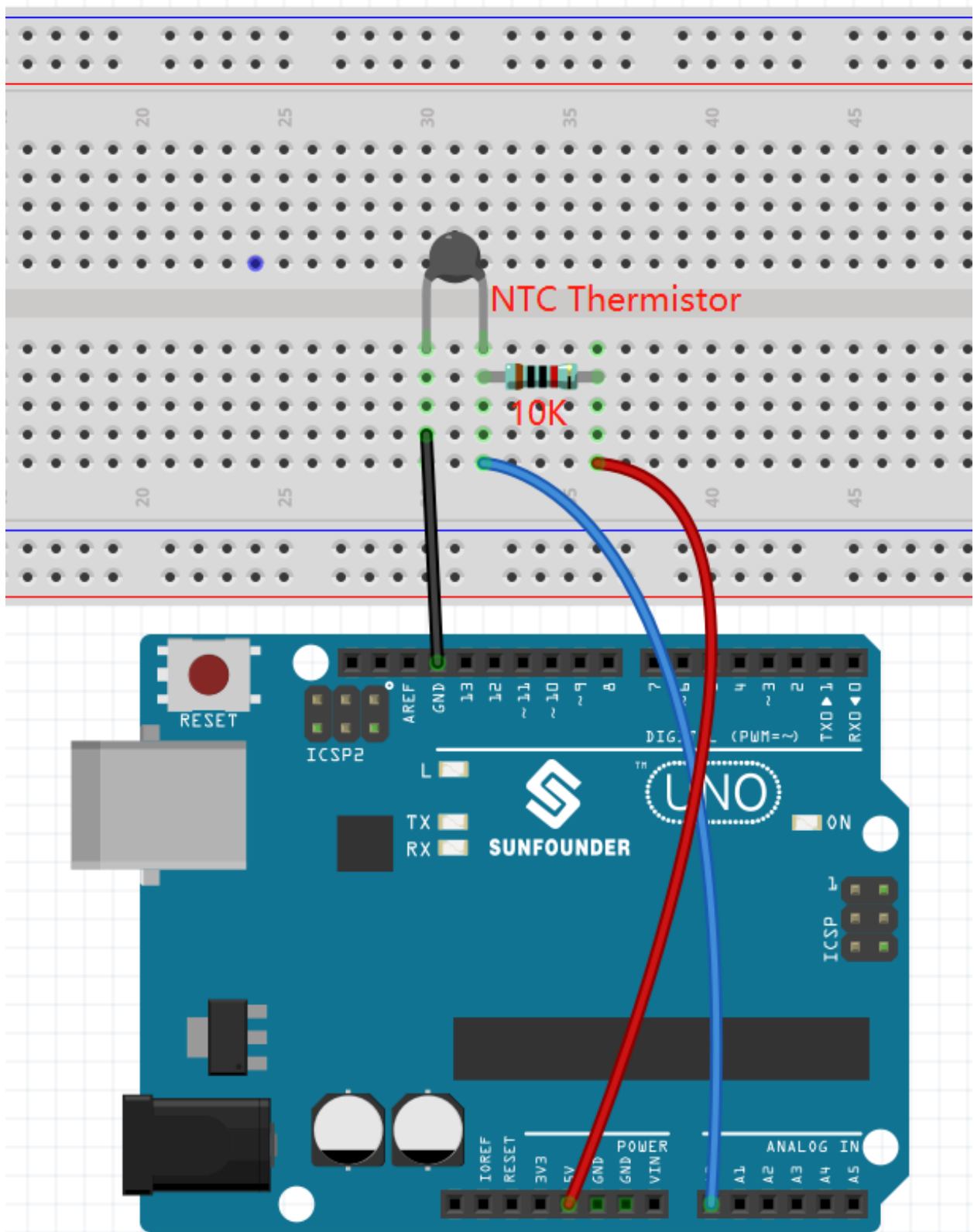
8.10.3 Schaltung aufbauen

Ein Thermistor ist ein Widerstand, dessen Widerstandswert stark temperaturabhängig ist. Es gibt zwei Typen von Widerständen, PTC (Widerstand steigt mit steigender Temperatur) und NTC (Widerstand nimmt mit steigender Temperatur ab).

Bauen Sie den Schaltkreis gemäß dem folgenden Diagramm.

Ein Ende des Thermistors ist mit GND verbunden, das andere Ende mit A0, und ein 10K Widerstand ist in Reihe mit 5V geschaltet.

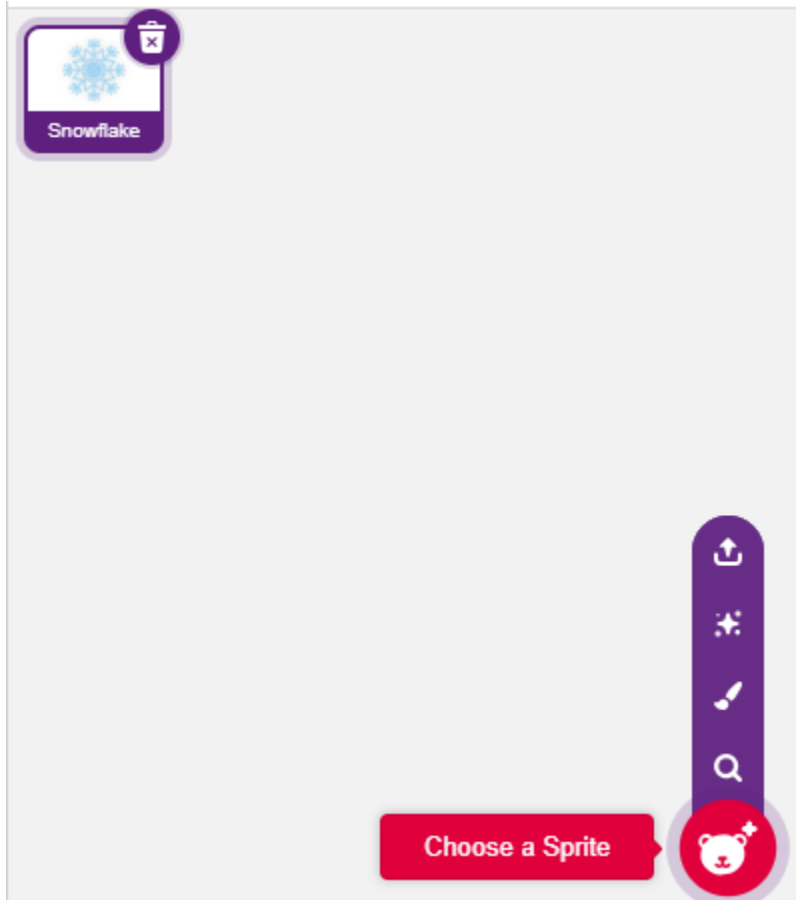
Hier wird der NTC-Thermistor verwendet. Wenn die Temperatur steigt, verringert sich der Widerstand des Thermistors, die Spannungsteilung von A0 nimmt ab, und der von A0 erhaltene Wert nimmt ab, und umgekehrt steigt er.



8.10.4 Programmierung

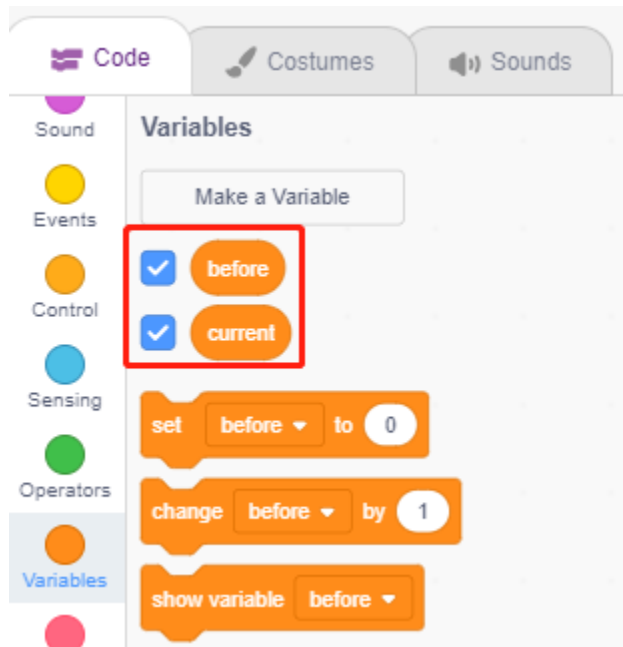
1. Wählen Sie eine Figur

Löschen Sie die Standardfigur, klicken Sie auf den Button **Choose a Sprite** in der unteren rechten Ecke des Figurbereichs, geben Sie **Snowflake** in das Suchfeld ein und fügen Sie diese hinzu.



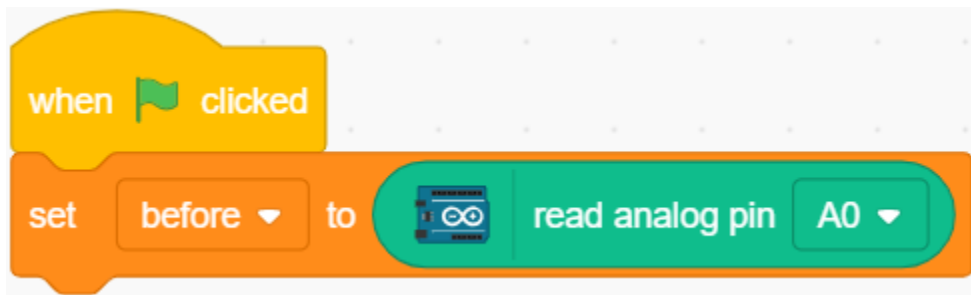
2. Erstellen Sie 2 Variablen

Erstellen Sie zwei Variablen, **before** und **current**, um den Wert von A0 in verschiedenen Zuständen zu speichern.



3. Lesen Sie den Wert von A0

Wenn das grüne Flaggen-Symbol angeklickt wird, lesen Sie den Wert von A0 und speichern Sie ihn in der Variablen **before**.



4. Lesen Sie den Wert von A0 erneut

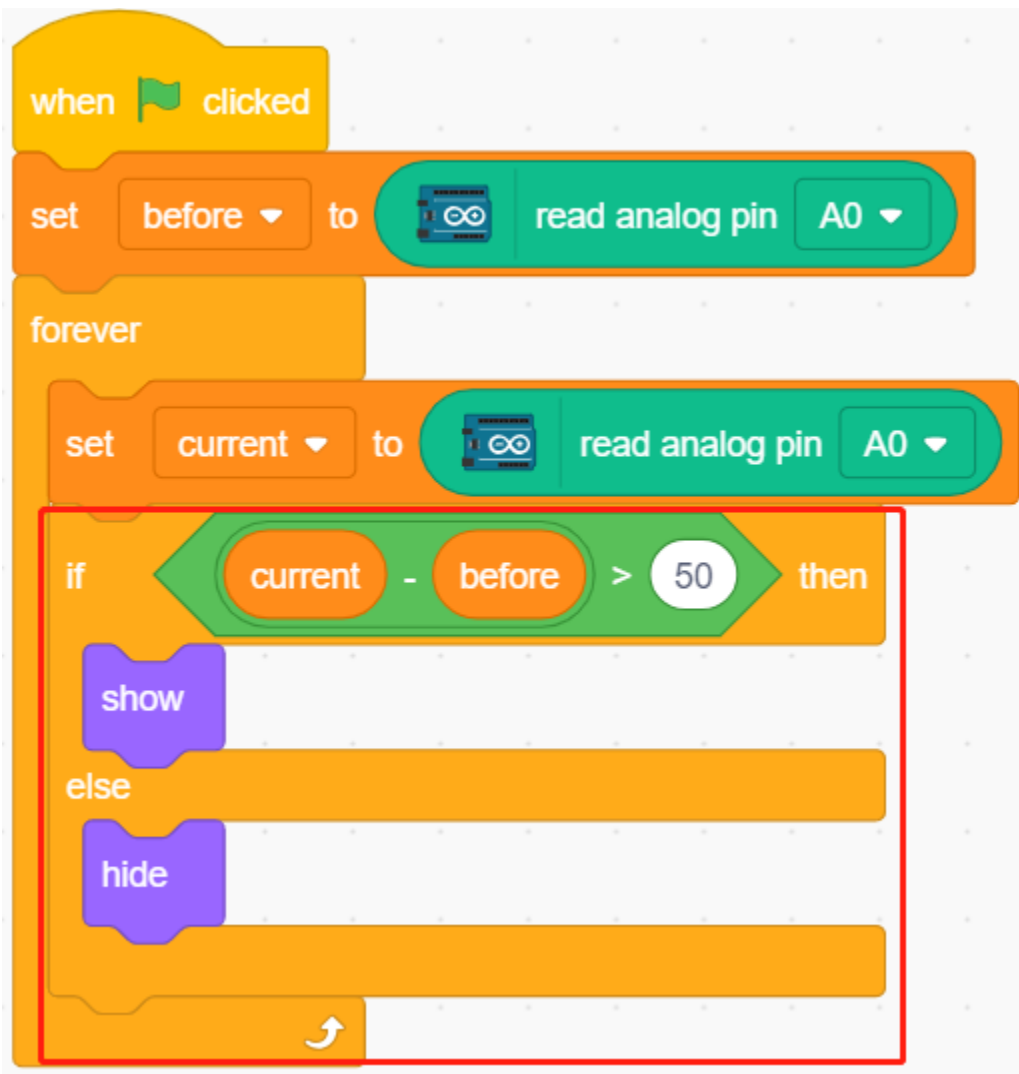
In [forever], lesen Sie den Wert von A0 erneut und speichern Sie ihn in der Variablen **current**.



5. Bestimmung von Temperaturänderungen

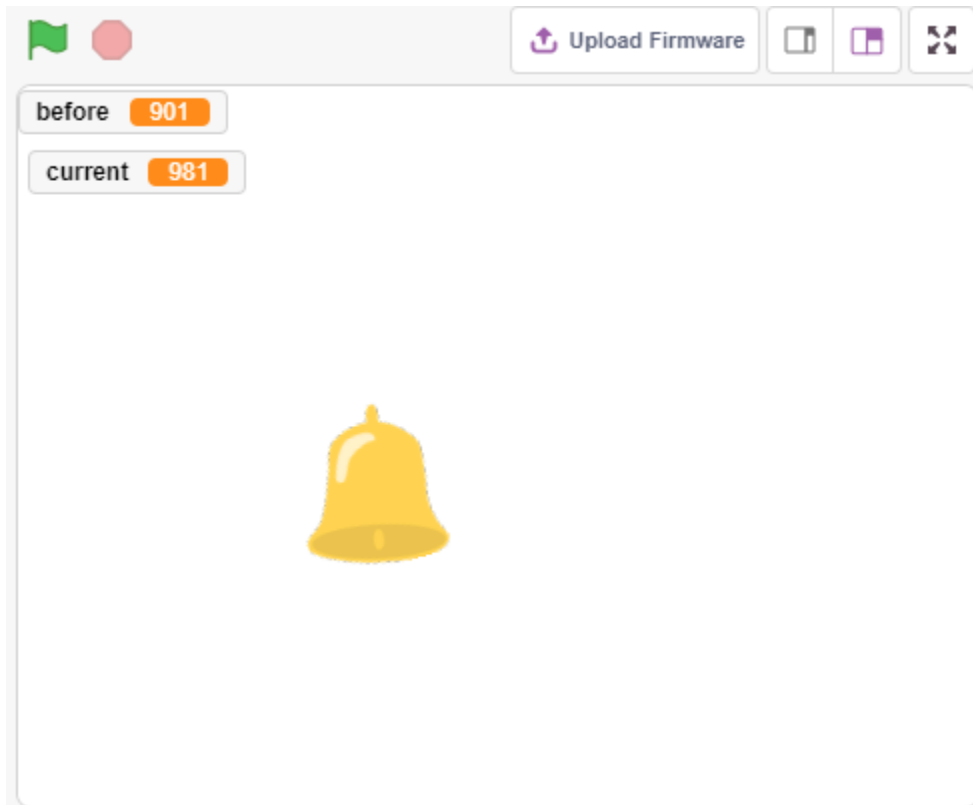
Verwenden Sie den [if else] Block, um festzustellen, ob der aktuelle Wert von A0 50 größer ist als zuvor, was eine Abnahme der Temperatur bedeutet. Lassen Sie in diesem Fall das **Schneeflocke**-Symbol anzeigen, sonst verbergen.

- [-] & [>]: Subtraktions- und Vergleichsoperatoren aus der **Operatoren**-Palette.



8.11 2.8 Lichtwecker

Im Leben gibt es viele verschiedene Weckerarten. Lassen Sie uns jetzt einen lichtgesteuerten Wecker erstellen. Wenn der Morgen anbricht und die Helligkeit zunimmt, wird dieser Lichtwecker Sie daran erinnern, dass es Zeit ist aufzustehen.



8.11.1 Was Sie lernen werden

- Funktionsprinzip des Fotowiderstands
- Stoppen der Soundwiedergabe und Beenden laufender Skripte

8.11.2 Benötigte Komponenten

Für dieses Projekt benötigen Sie folgende Bauteile.

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

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

Sie können die Teile auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Fotowiderstand</i>	

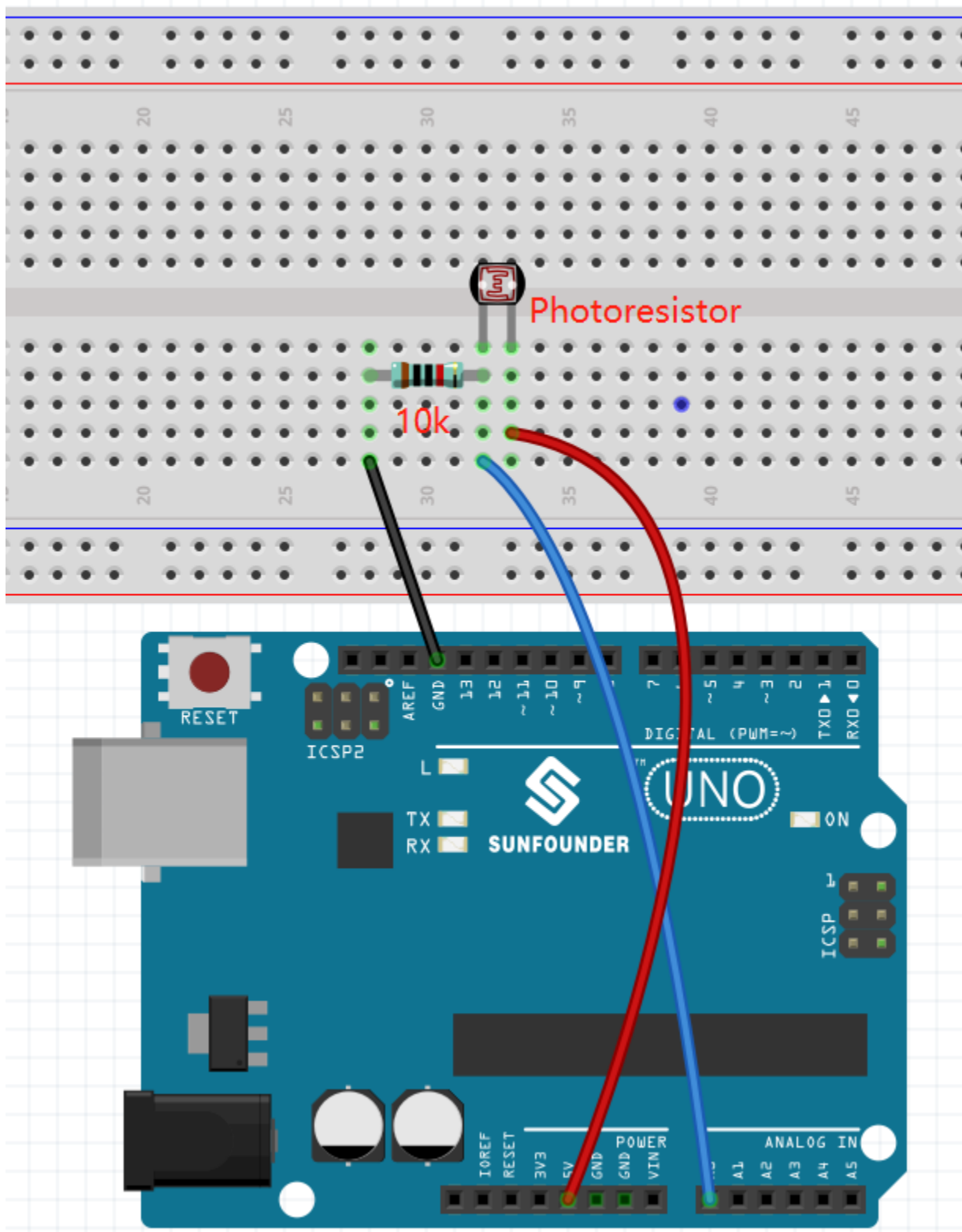
8.11.3 Schaltung aufbauen

Ein Fotowiderstand oder Fotodiode ist ein lichtgesteuerter variabler Widerstand. Der Widerstand eines Fotowiderstands verringert sich mit zunehmender Lichteinstrahlung.

Bauen Sie die Schaltung gemäß dem folgenden Schaltplan auf.

Verbinden Sie ein Ende des Fotowiderstands mit 5V, das andere Ende mit A0 und schließen Sie einen 10K-Widerstand in Reihe mit GND an diesem Ende an.

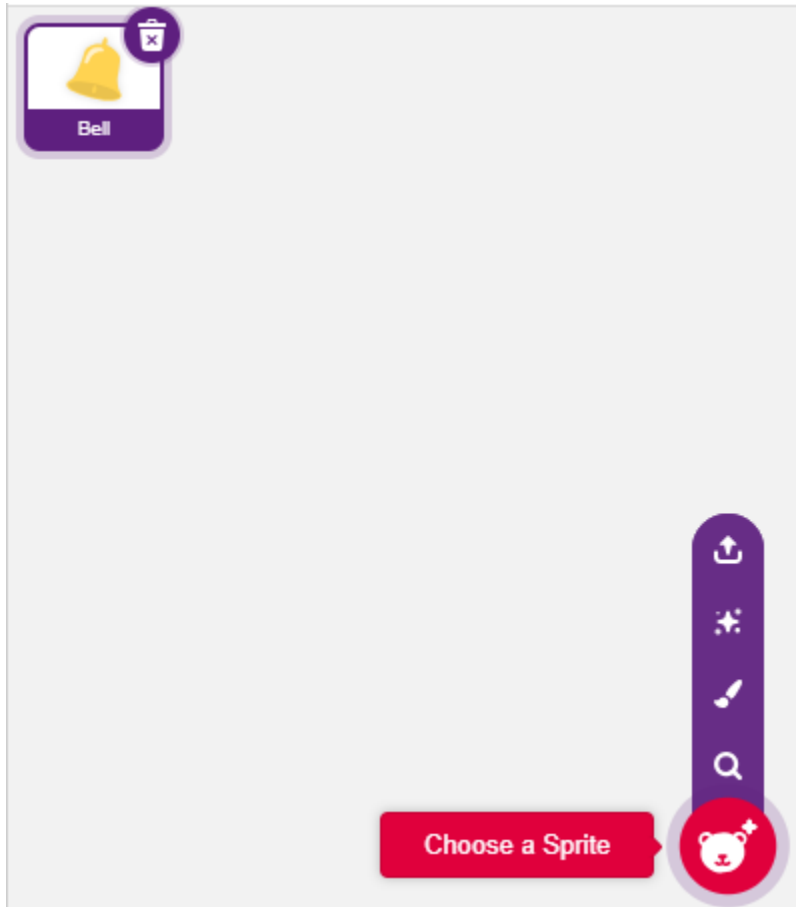
Wenn also die Lichtintensität zunimmt, verringert sich der Widerstand des Fotowiderstands, die Spannungsteilung des 10K-Widerstands steigt und der von A0 ermittelte Wert wird größer.



8.11.4 Programmierung

1. Wählen Sie eine Figur

Löschen Sie die Standardfigur, klicken Sie auf den **Choose a Sprite**-Button in der unteren rechten Ecke des Figurenbereichs, geben Sie **bell** in das Suchfeld ein und fügen Sie sie dann hinzu.



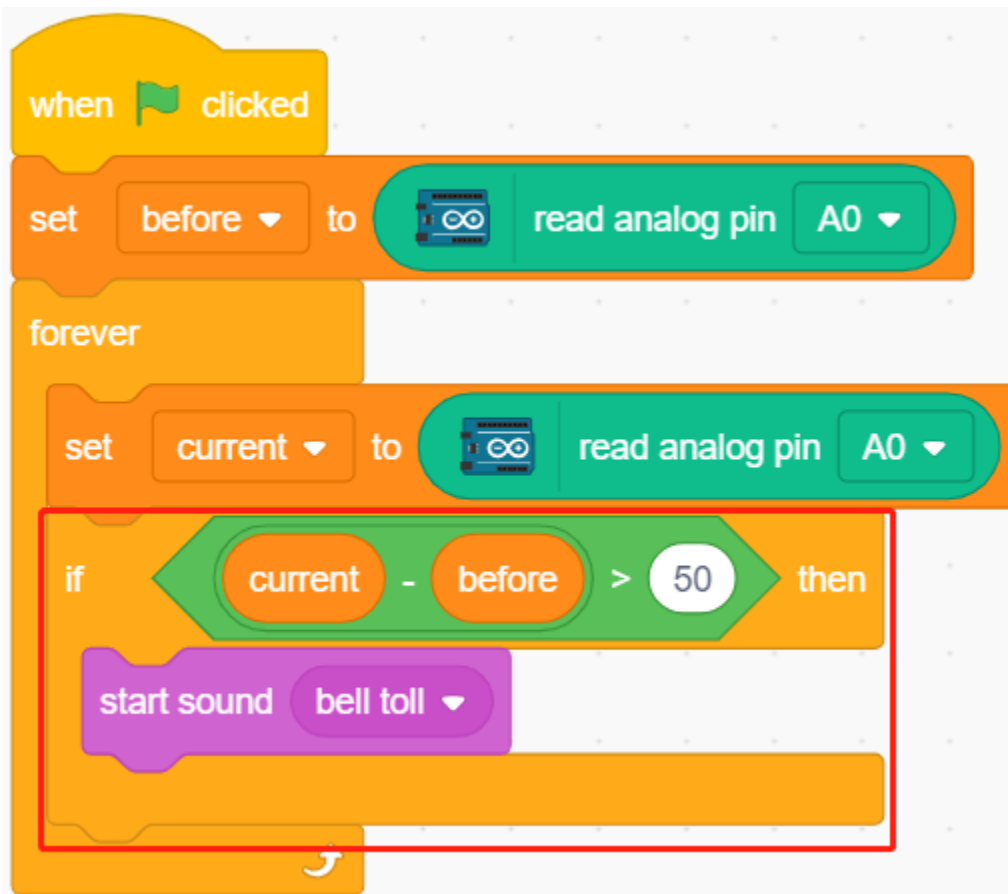
2. Wert von A0 auslesen

Erstellen Sie zwei Variablen **before** und **current**. Wenn Sie auf die grüne Flagge klicken, lesen Sie den Wert von A0 aus und speichern ihn in der Variable **before** als Referenzwert. In [forever], lesen Sie den Wert von A0 erneut und speichern ihn in der Variable **current**.



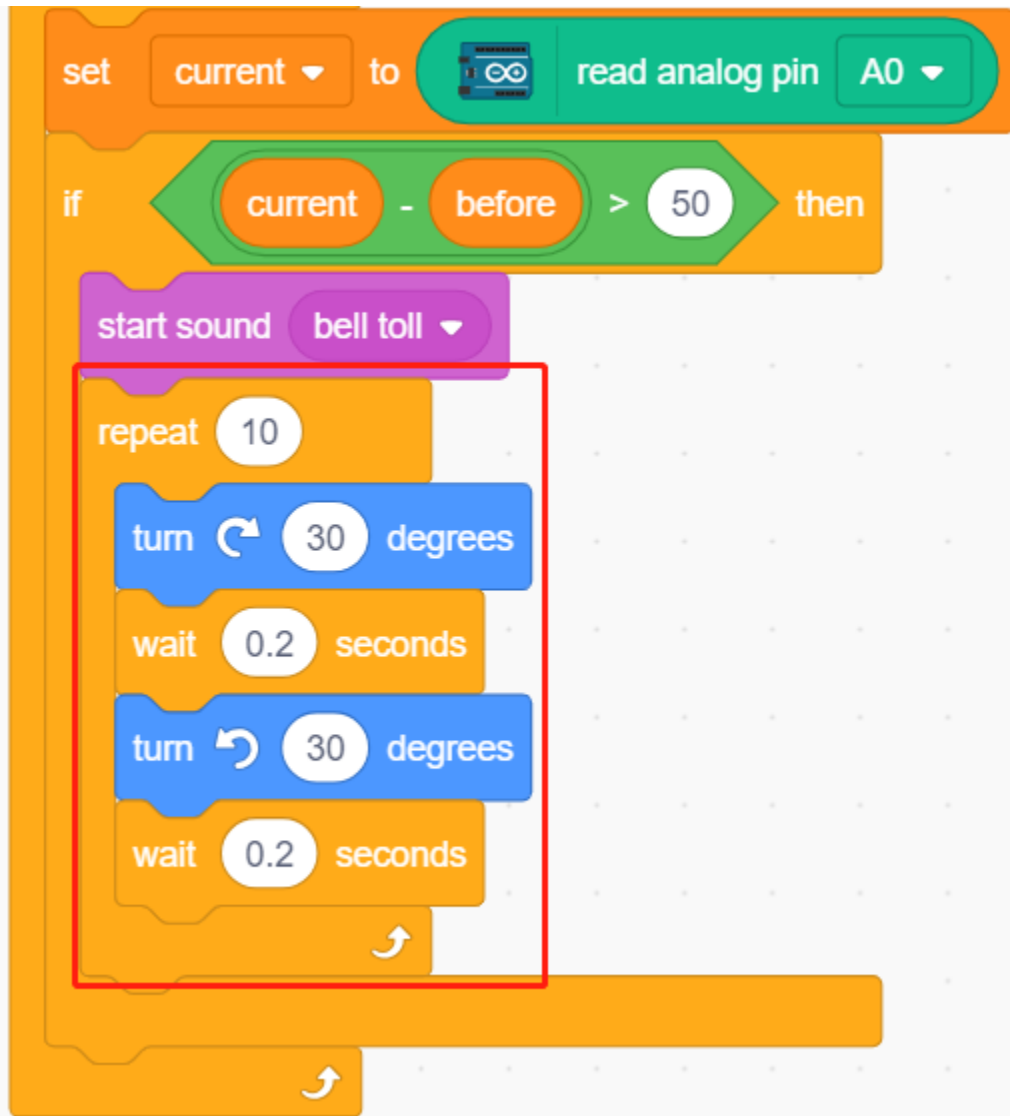
3. Einen Ton erzeugen

Wenn der aktuelle Wert von A0 um 50 höher ist als der vorherige, was bedeutet, dass die aktuelle Lichtintensität über dem Schwellenwert liegt, lassen Sie die Figur einen Ton erzeugen.



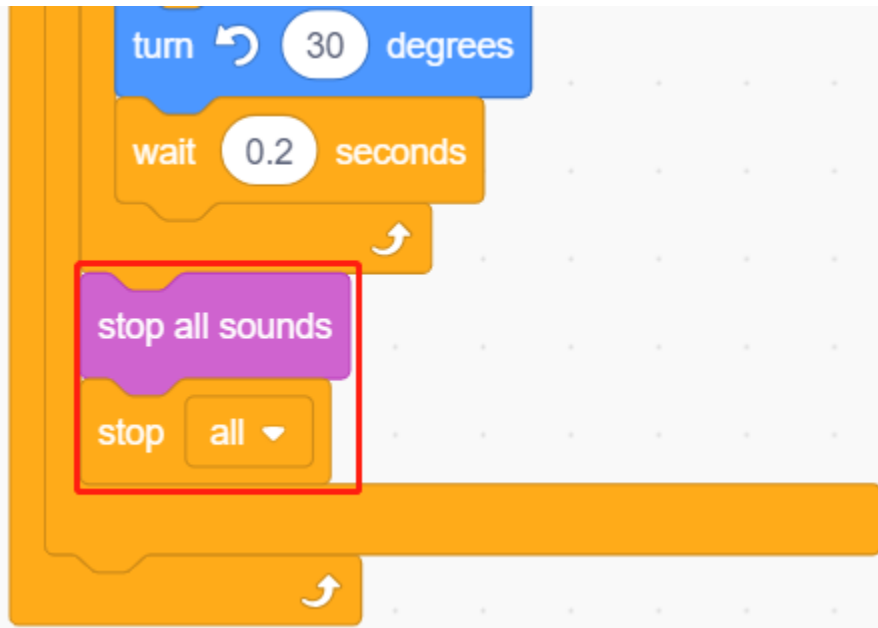
4. Die Figur drehen

Verwenden Sie den [turn block], um die **bell**-Figur links und rechts drehen zu lassen, um den Weckereffekt zu erzielen.



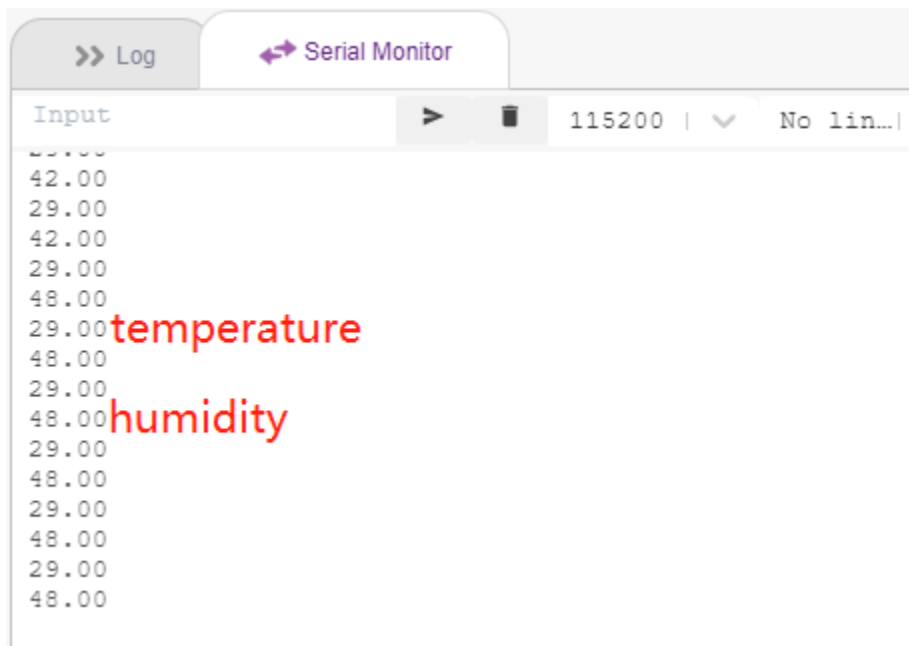
5. Alles stoppen

Stoppen Sie den Alarm, nachdem er eine Weile geklingelt hat.



8.12 2.9 Temperatur und Feuchtigkeit auslesen

Bisherige Projekte nutzten den Bühnenmodus, aber einige Funktionen sind nur im Hochlademodus verfügbar, wie z.B. die serielle Kommunikation. In diesem Projekt werden wir die Temperatur und die Feuchtigkeit des DHT11 über den Seriellen Monitor im *Hochlade-Modus* ausgeben.



8.12.1 Was Sie lernen werden

- Temperatur und Feuchtigkeit vom DHT11-Modul auslesen
- Seriellen Monitor für den *Hochlade-Modus* nutzen
- Erweiterungen hinzufügen

8.12.2 Benötigte Komponenten

Für dieses Projekt benötigen Sie folgende Bauteile.

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

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

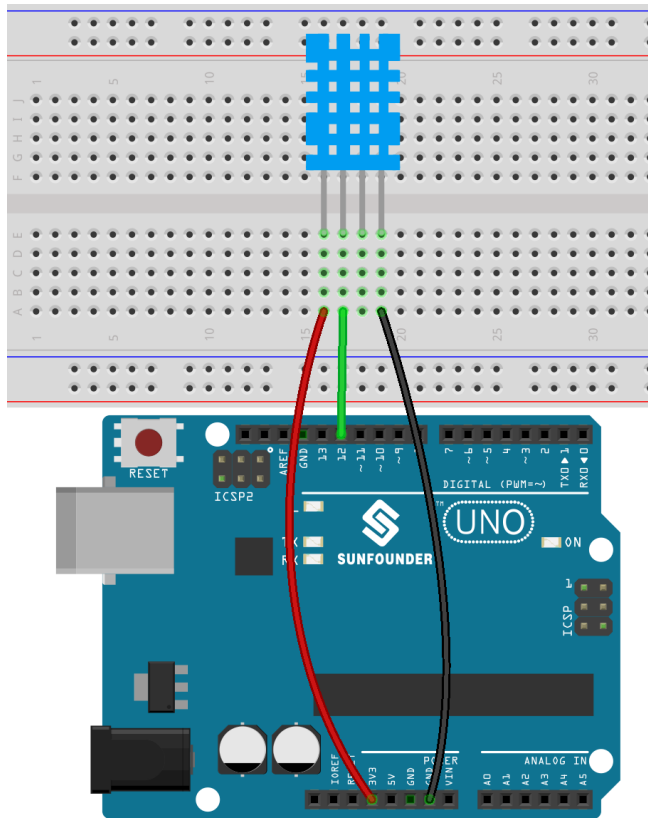
Sie können die Teile auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>DHT11 Feuchtigkeits- und Temperatursensor</i>	-

8.12.3 Schaltung aufbauen

Der digitale Temperatur- und Feuchtigkeitssensor DHT11 ist ein Kombisensor, der einen kalibrierten digitalen Signalausgang für Temperatur und Feuchtigkeit bietet.

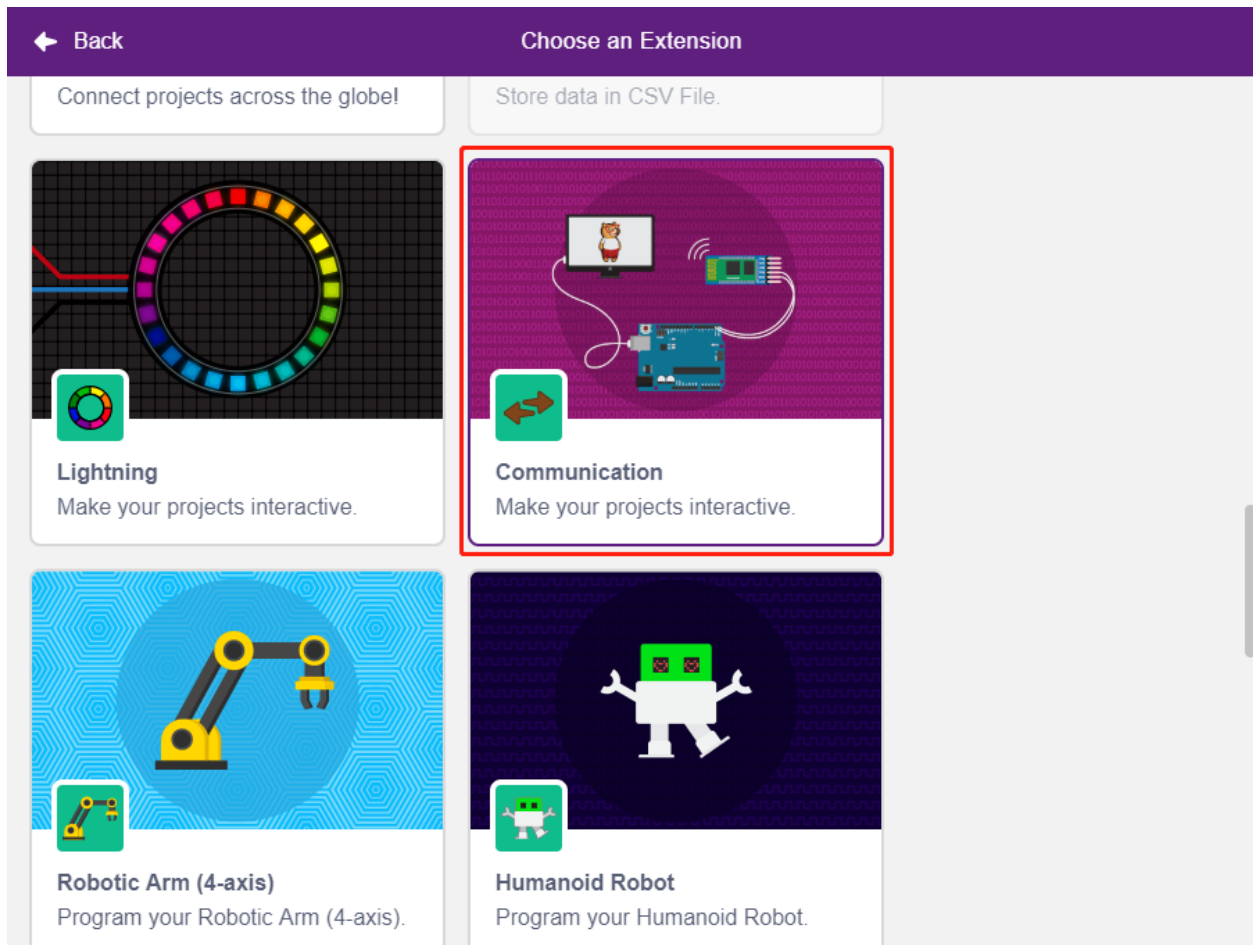
Bauen Sie die Schaltung gemäß dem folgenden Diagramm auf.



8.12.4 Programmierung

1. Erweiterungen hinzufügen

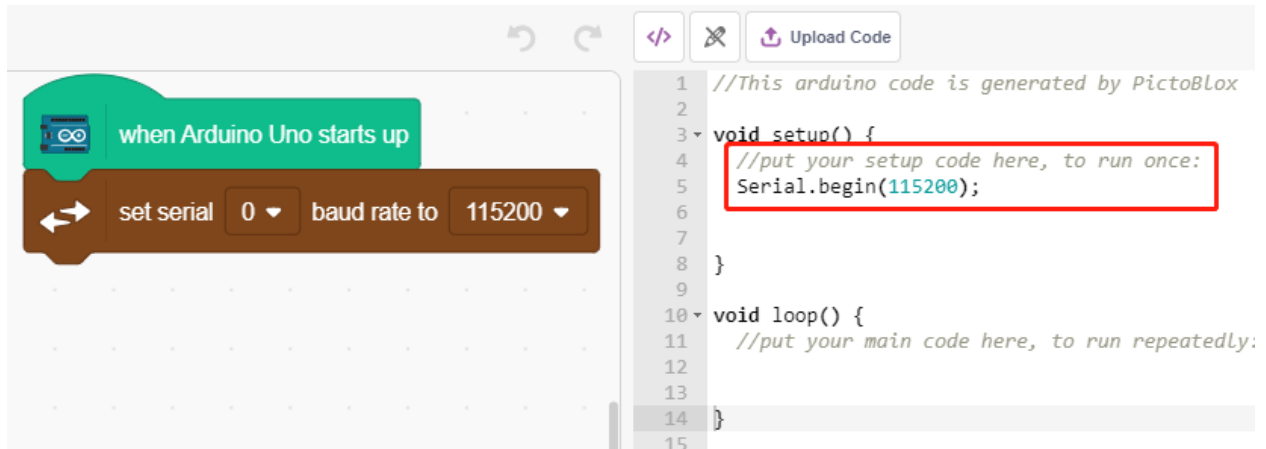
Wechseln Sie in den **Upload**-Modus, klicken Sie auf den **Add Extension**-Button in der unteren linken Ecke und wählen Sie dann **Communication** aus. Dies wird am Ende des Palettenbereichs angezeigt.



2. Arduino Uno und Seriellen Monitor initialisieren

Im **Upload**-Modus starten Sie Arduino Uno und setzen dann die Baudrate des seriellen Ports.

- [when Arduino Starts up]: Im **Upload**-Modus starten Sie Arduino Uno.
- [set serial baud rate to]: Aus der **Communications**-Palette, zum Setzen der Baudrate des seriellen Ports 0, Standard ist 115200. Wenn Sie einen Mega2560 verwenden, können Sie die Baudrate in den seriellen Ports 0~3 einstellen.



3. Temperatur und Feuchtigkeit auslesen

Erstellen Sie 2 Variablen **tem** und **humi** zum Speichern von Temperatur und Feuchtigkeit. Der Code erscheint auf der rechten Seite, während Sie den Block ziehen und ablegen.

```

1 //This arduino code is generated by PictoBlox
2
3 //Included Libraries
4 #include <DHT.h>
5
6 //MACROS are defined here
7 #define DHTPIN_12 12
8 #define DHTTYPE DHT11
9 DHT dht_12(DHTPIN_12, DHTTYPE);
10
11 //Global Variables are declared here
12 float tem;
13 float humi;
14
15 void setup() {
16   //put your setup code here, to run once:
17   Serial.begin(115200);
18   dht_12.begin();
19 }
20
21
22
23 void loop() {
24   //put your main code here, to run repeatedly:
25
26
27   tem = dht_12.readTemperature();
28   humi = dht_12.readHumidity();
29 }

```

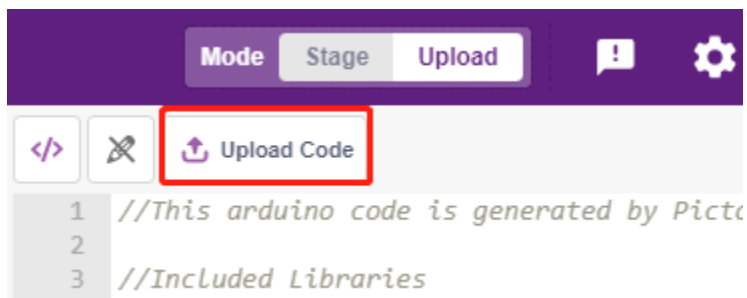
4. Auf dem Seriellen Monitor ausgeben

Schreiben Sie die ausgelesene Temperatur und Feuchtigkeit auf den Seriellen Monitor. Um ein zu schnelles Übertragen und ein Blockieren von PictoBlox zu vermeiden, verwenden Sie den [wait seconds]-Block, um ein Zeitintervall für die nächste Ausgabe hinzuzufügen.



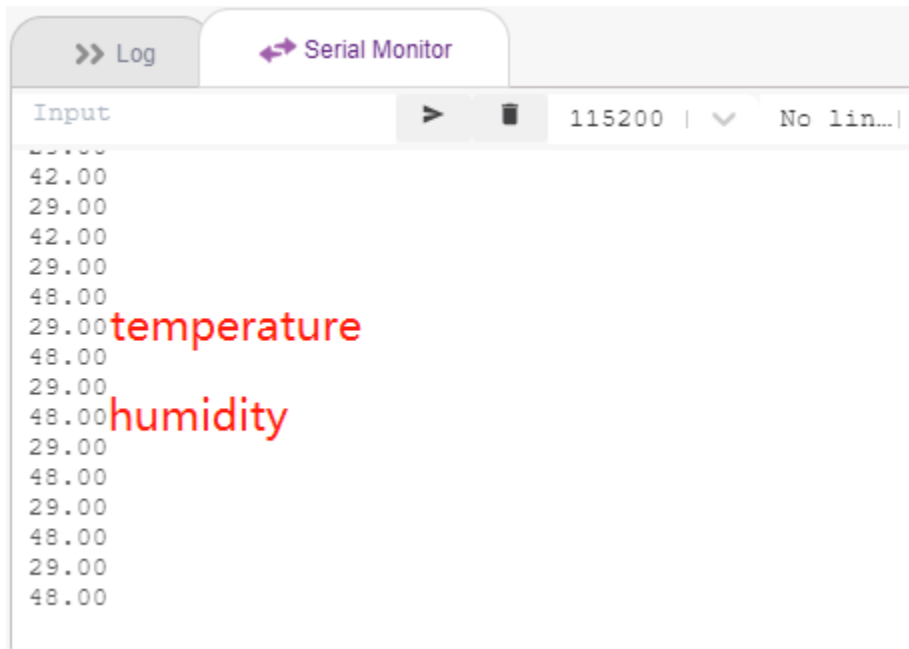
5. Code hochladen

Im Gegensatz zum **Stage**-Modus muss der Code im **Upload**-Modus mit dem **Upload Code**-Button auf das Arduino-Board geladen werden, um den Effekt zu sehen. So können Sie das USB-Kabel abziehen und das Programm läuft weiter.



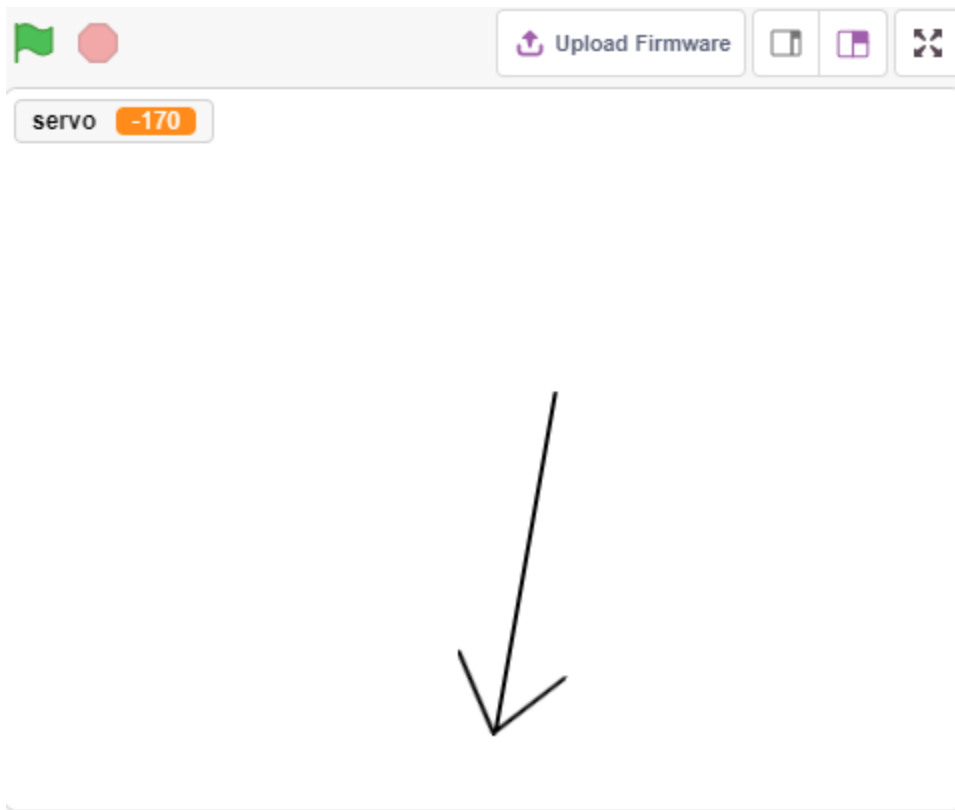
6. Seriellen Monitor starten

Öffnen Sie jetzt den **Serial Monitor**, um die Temperatur und die Feuchtigkeit zu sehen.



8.13 2.10 Pendel

In diesem Projekt erstellen wir ein Pfeilpendel, während der Servo der Drehung folgt.



8.13.1 Was Sie lernen werden

- Wie der Servo funktioniert und welchen Drehwinkelbereich er hat
- Ein Sprite zeichnen und den Mittelpunkt am Ende setzen.

8.13.2 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 sie auch einzeln über die untenstehenden Links kaufen.

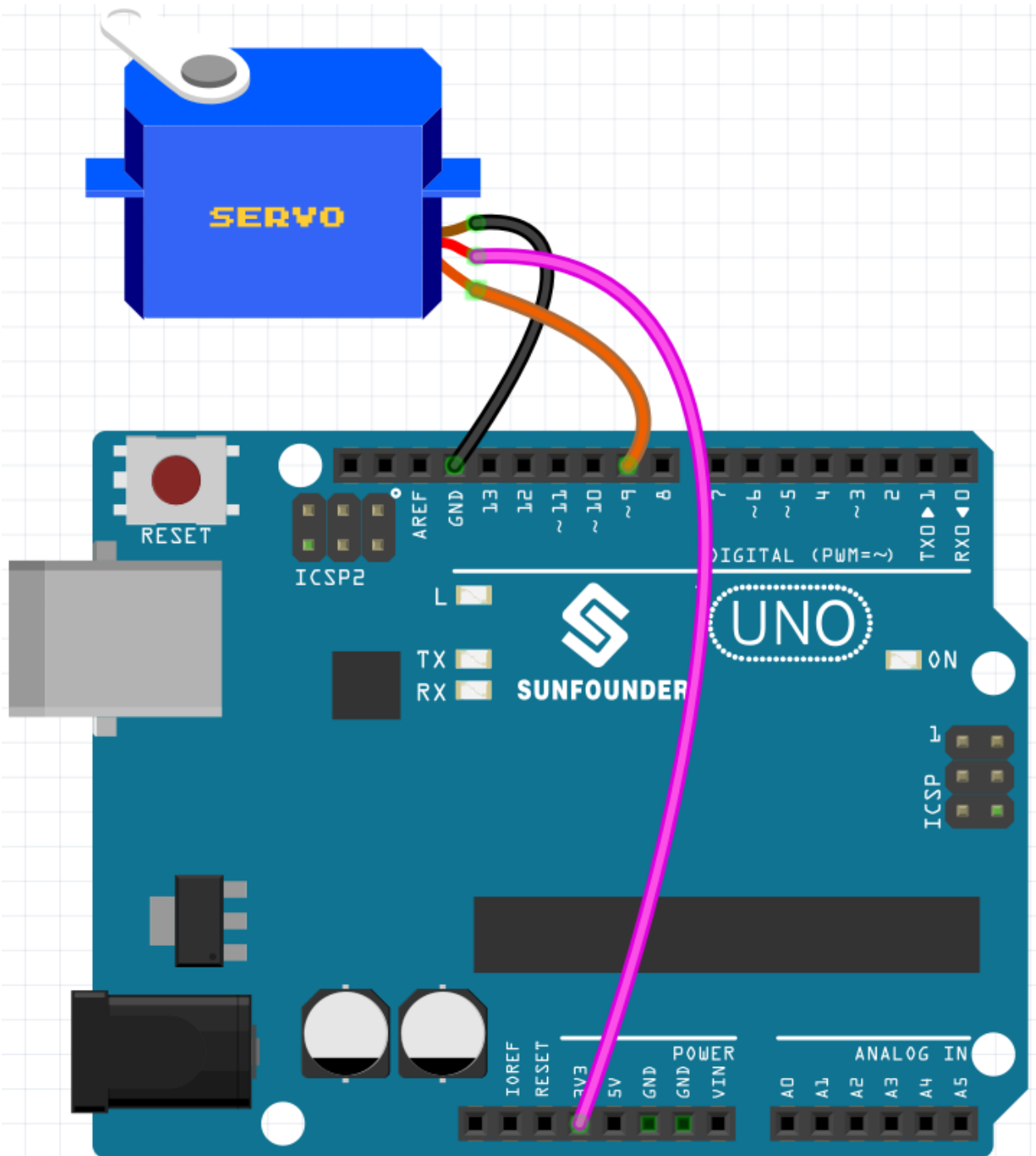
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>Servo</i>	

8.13.3 Schaltung aufbauen

Ein Servo ist ein Zahnradmotor, der sich nur um 180 Grad drehen kann. Er wird durch das Senden von elektrischen Pulsen von Ihrer Schaltung gesteuert. Diese Pulse geben dem Servo die Position vor, in die er sich bewegen soll.

Der Servo hat drei Kabel: Das braune Kabel ist GND, das rote ist VCC (an 3.3V anschließen) und das orange ist das Signalkabel. Der Drehwinkelbereich liegt bei 0-180 Grad.

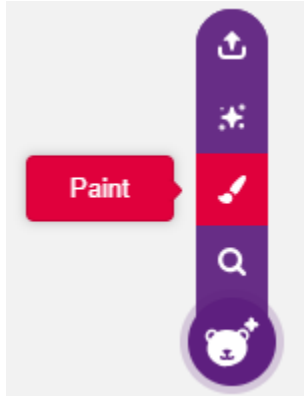
Bauen Sie den Schaltkreis gemäß dem folgenden Diagramm.



8.13.4 Programmierung

1. Ein Sprite malen

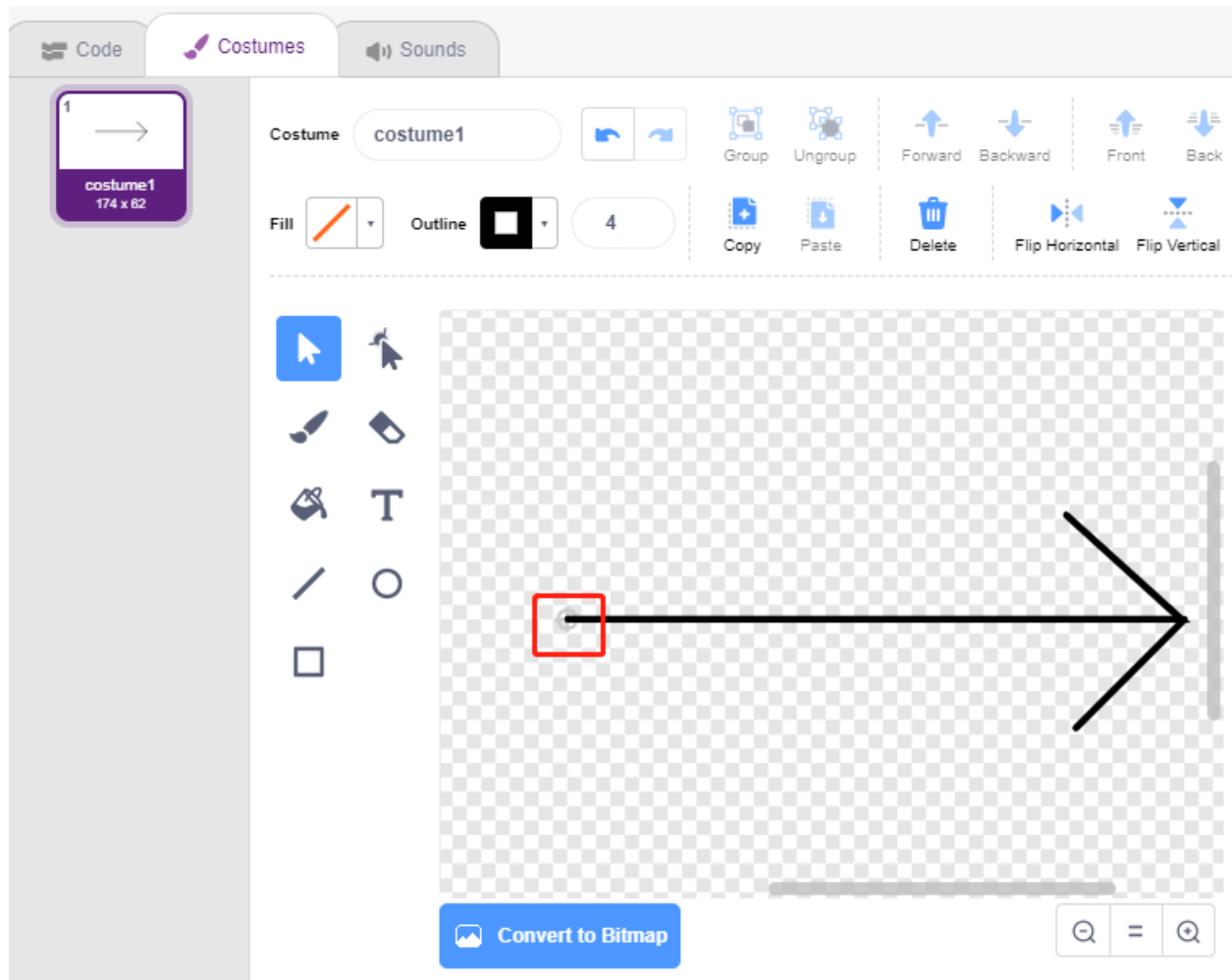
Löschen Sie das Standard-Sprite, wählen Sie den Sprite-Button und klicken Sie auf **Paint**. Ein leeres Sprite namens **Sprite1** wird angezeigt.



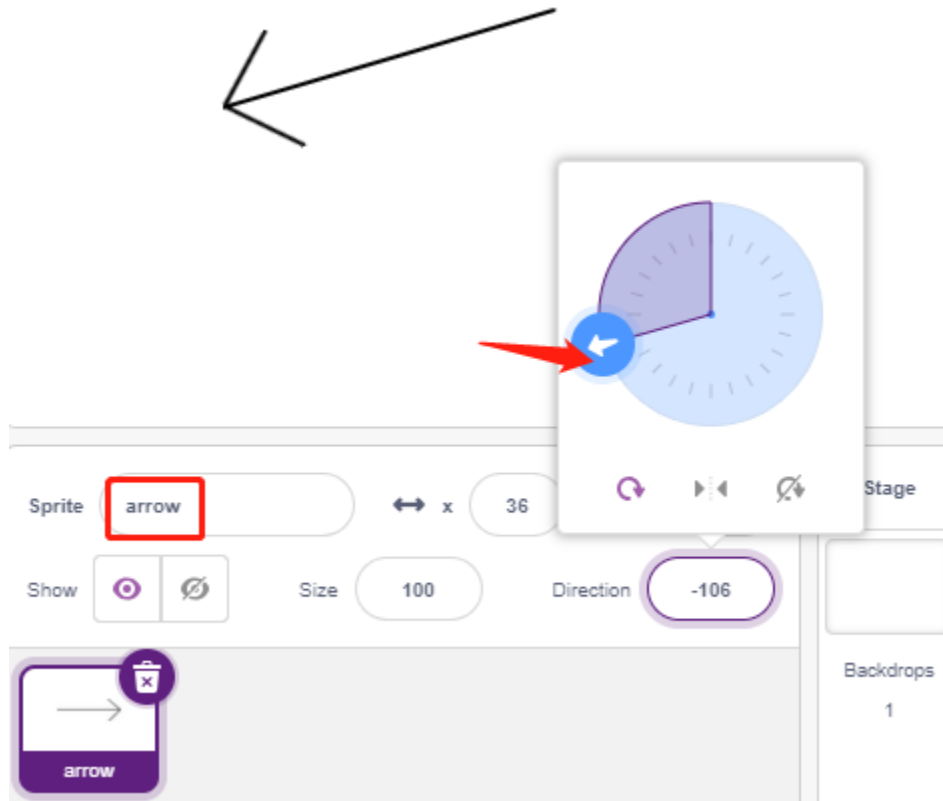
Auf der geöffneten **Costumes**-Seite verwenden Sie das **Line tool**, um einen Pfeil zu zeichnen.

Bemerkung:

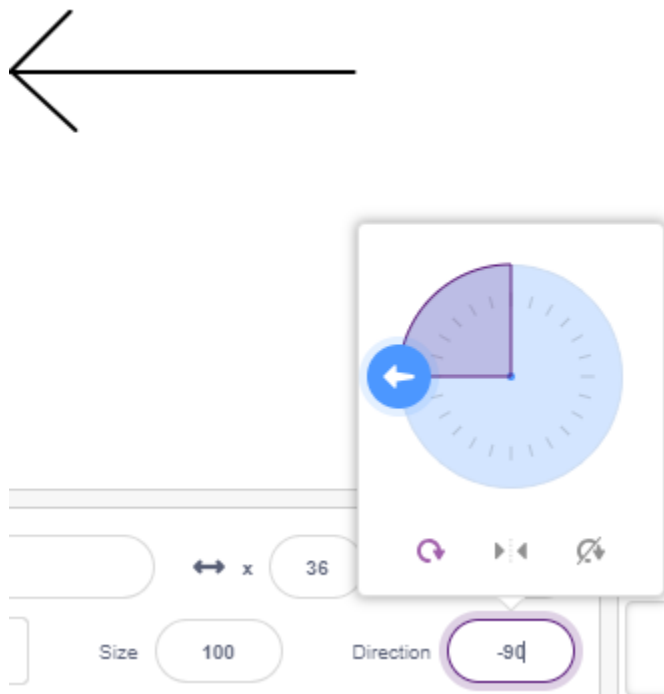
- Beginnen Sie, den Pfeil von der Mitte der Leinwand aus zu zeichnen, damit sich der Pfeil mit dem Mittelpunkt als Ursprung im Kreis dreht.
 - Halten Sie die Umschalttaste gedrückt, um den Linienwinkel gerade oder 45 Grad zu machen.
-

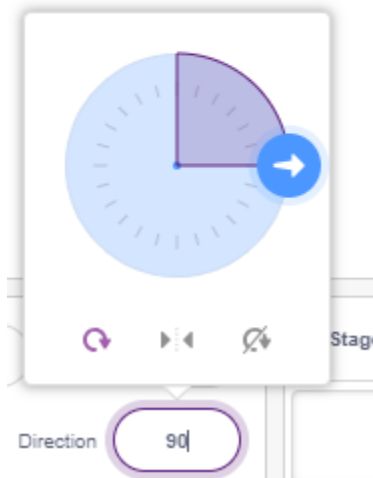
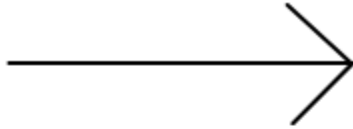


Nach dem Zeichnen wird das **arrow**-Sprite auf der Bühne angezeigt. Nennen Sie es **arrow**. Klicken Sie dann auf die Nummer nach **Direction**. Ein kreisförmiges Zifferblatt erscheint. Ziehen Sie jetzt diesen Pfeil und prüfen Sie, ob das **arrow**-Sprite auf der Bühne sich mit dem Ende als Ursprung dreht.



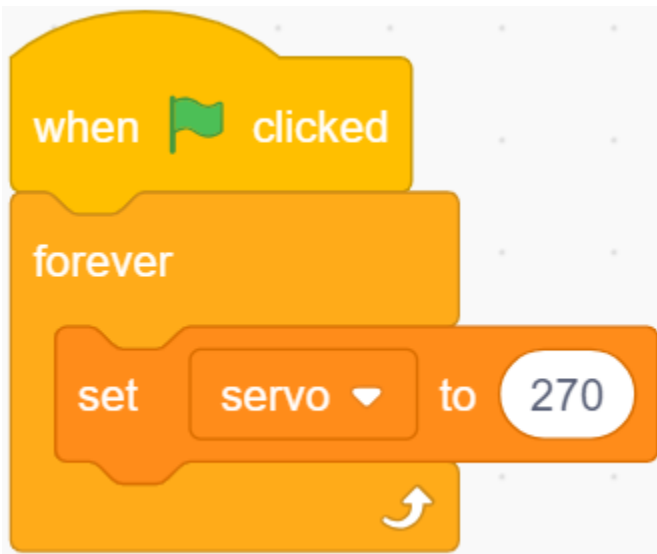
Damit das **arrow**-Sprite von links nach rechts schwingt, liegt der Winkelbereich bei -90 bis -180, 180 bis 90.





2. Eine Variable erstellen.

Erstellen Sie eine Variable namens **servo**, die den Winkelwert speichert und setzen Sie den Anfangswert auf 270.



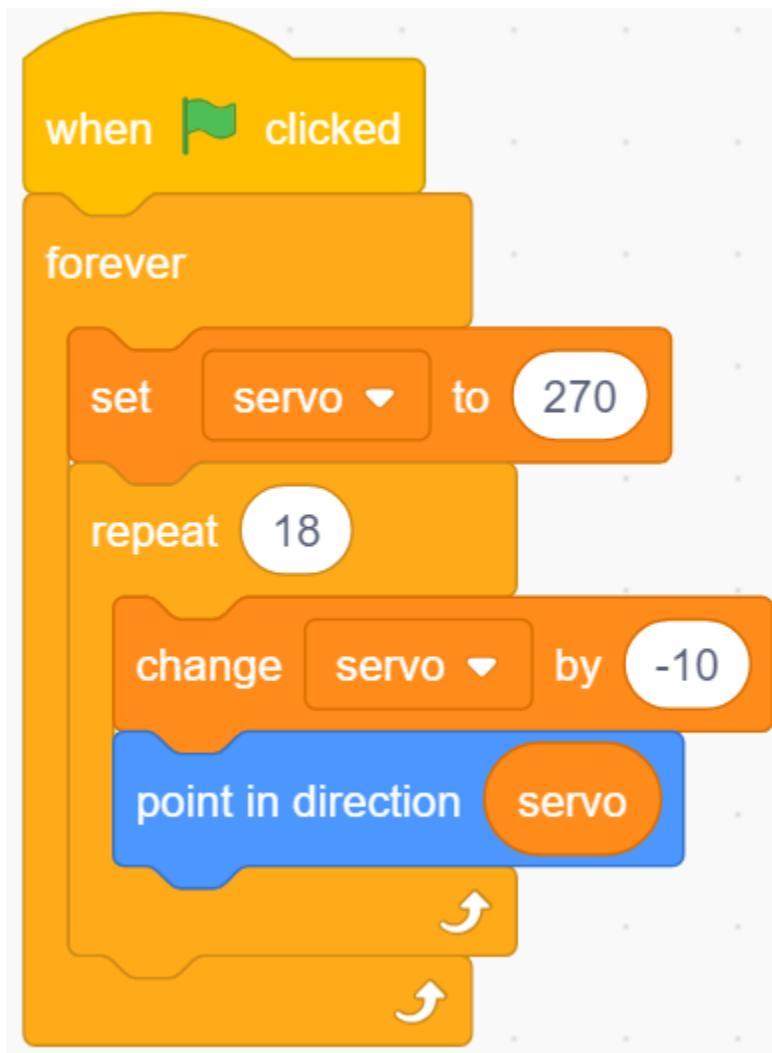
3. Von links nach rechts schwingen

Lassen Sie das **arrow**-Sprite nun von der linken -90-Grad-Position zur rechten 90-Grad-Position schwingen.

Mit dem [repeat]-Block fügen Sie bei jedem Durchlauf -10 zum Variablenwert hinzu und erreichen nach 18 Durchläufen 90 Grad. Verwenden Sie dann den [point in block]-Block, damit sich das Pfeil-Sprite zu diesen Winkeln dreht.

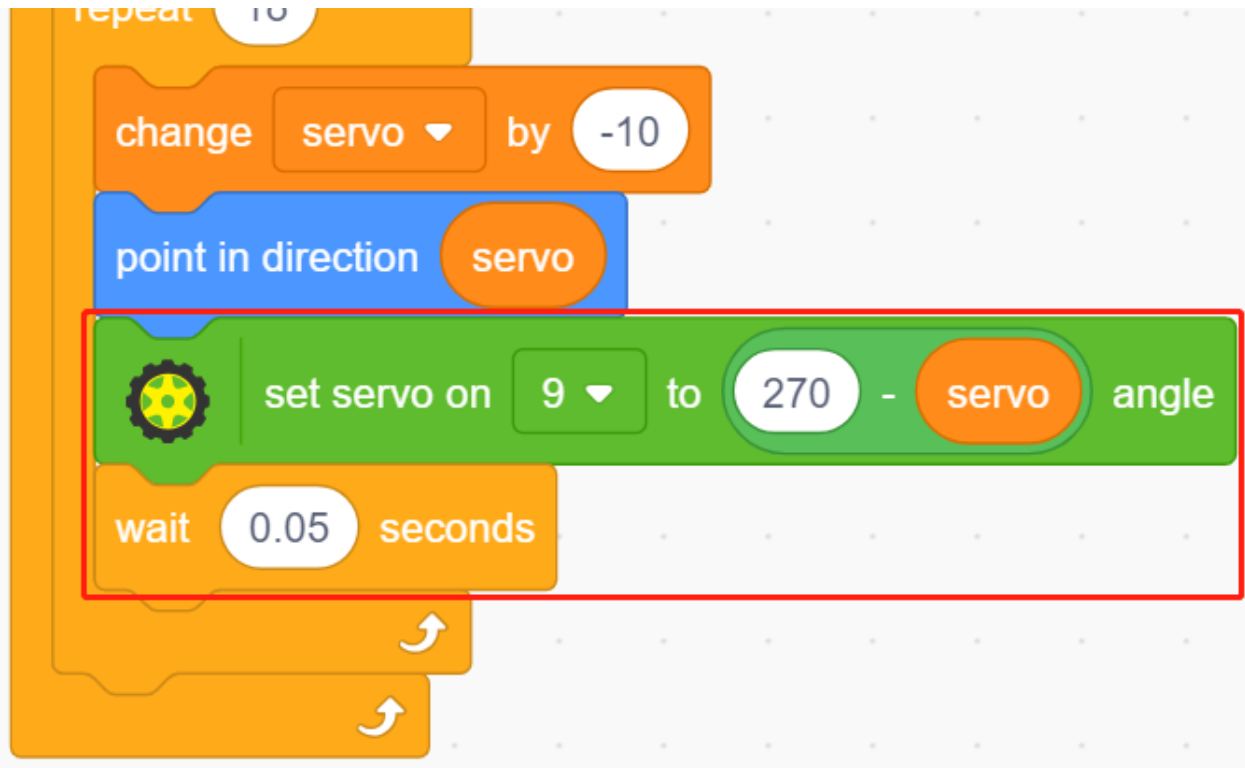
Da der Drehwinkel des Sprites -180 ~ 180 beträgt, werden Winkel außerhalb dieses Bereichs durch die folgenden Bedingungen konvertiert.

- Wenn Winkel > 180, dann Winkel - 360.



4. Den Servo drehen

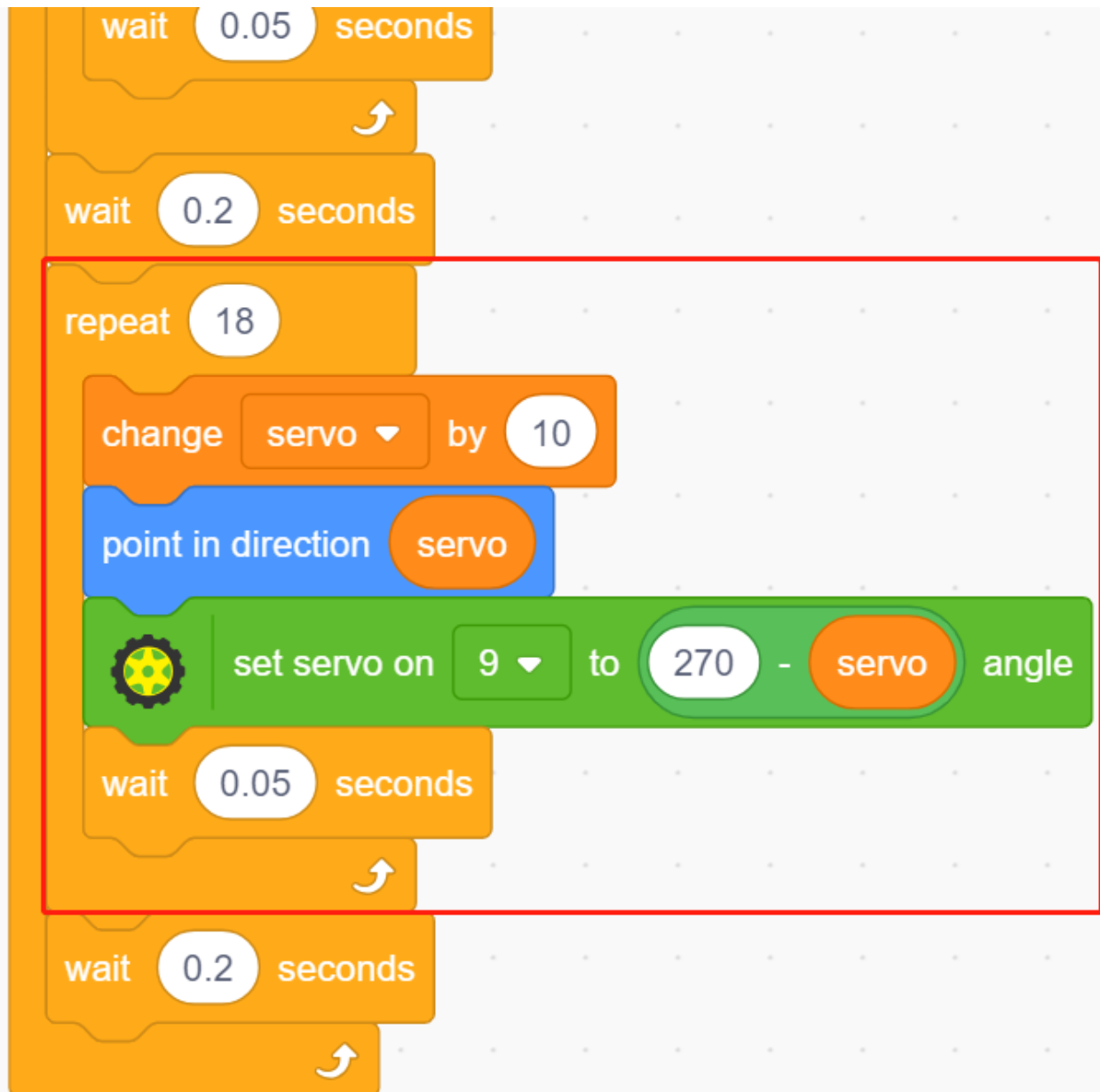
Wenn Sie auf die grüne Flagge klicken, sehen Sie, dass der Pfeil schnell nach rechts und dann wieder nach links dreht. Verwenden Sie hier einen [wait seconds]-Block, um die Drehung zu verlangsamen. Verwenden Sie auch den [set servo on to angle]-Block, um den mit dem Arduino-Board verbundenen Servo auf einen bestimmten Winkel zu drehen.



5. Von rechts nach links schwingen

Auf die gleiche Weise lassen Sie den Servo und das **arrow**-Sprite langsam von rechts nach links drehen.

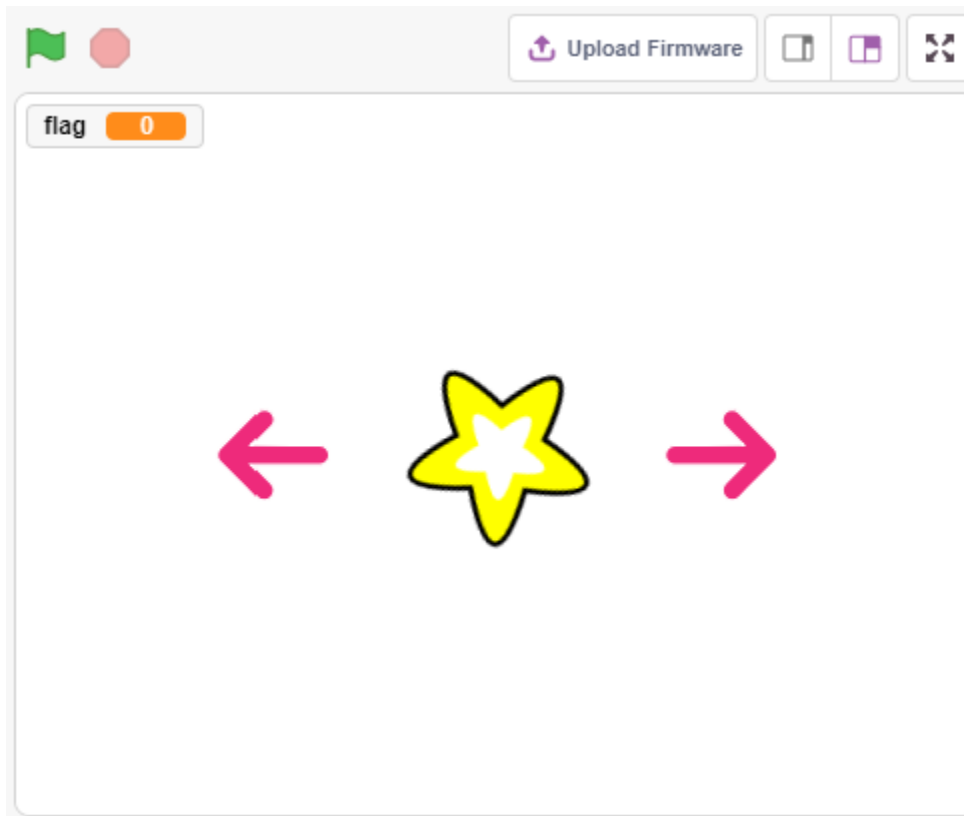
- Wenn Winkel > 180, dann Winkel - 360.



8.14 2.11 Drehender Ventilator

In diesem Projekt erstellen wir einen sich drehenden Sternsprite und Ventilator.

Durch Klicken auf die linken und rechten Pfeilsprites auf der Bühne wird die Uhrzeigersinn- und die Gegen-Uhrzeigersinn-Drehung des Motors und des Sternsprites gesteuert. Klicken Sie auf den Sternsprite, um die Drehung zu stoppen.



8.14.1 Was Sie lernen werden

- Arbeitsprinzip des Motors
- Broadcast-Funktion
- Block zum Stoppen anderer Skripte im Sprite

8.14.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

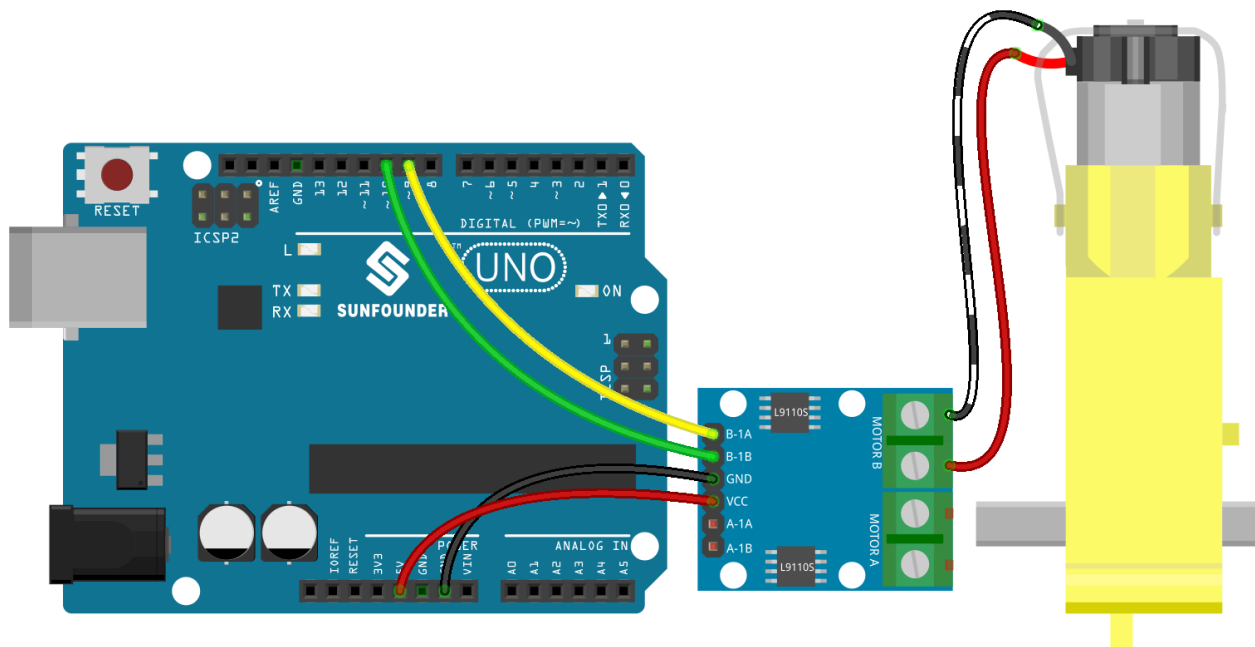
Es ist definitiv praktisch, ein ganzes Set zu kaufen, hier ist der Link:

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

Sie können sie auch einzeln über die folgenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>TT Motor</i>	-
<i>L9110 Motortreibermodul</i>	-

8.14.3 Schaltung aufbauen

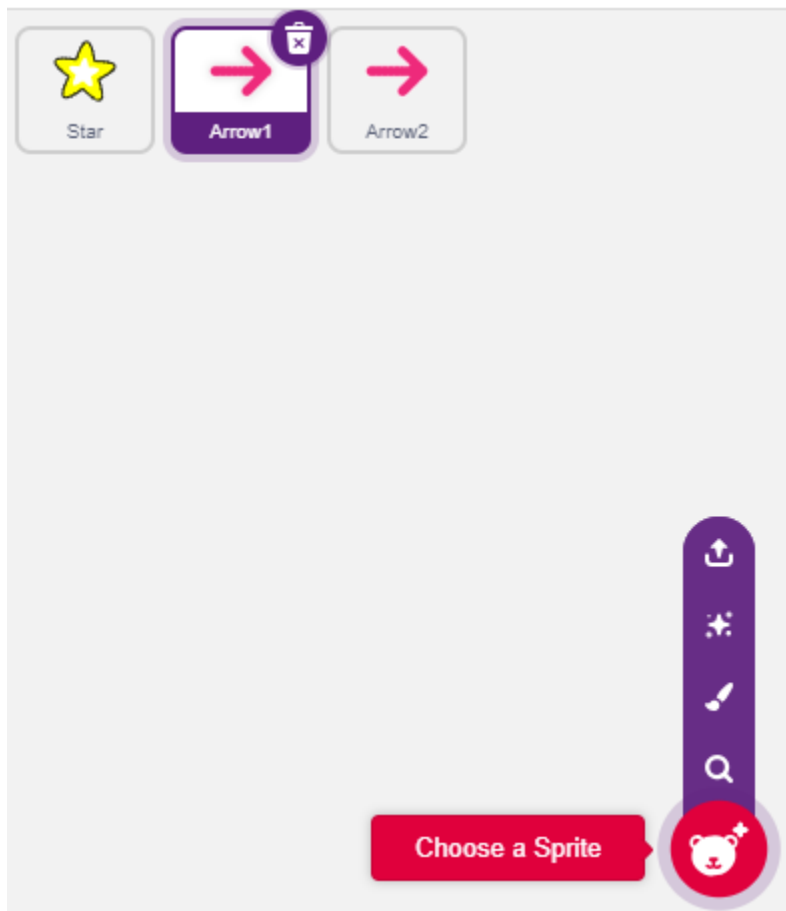


8.14.4 Programmierung

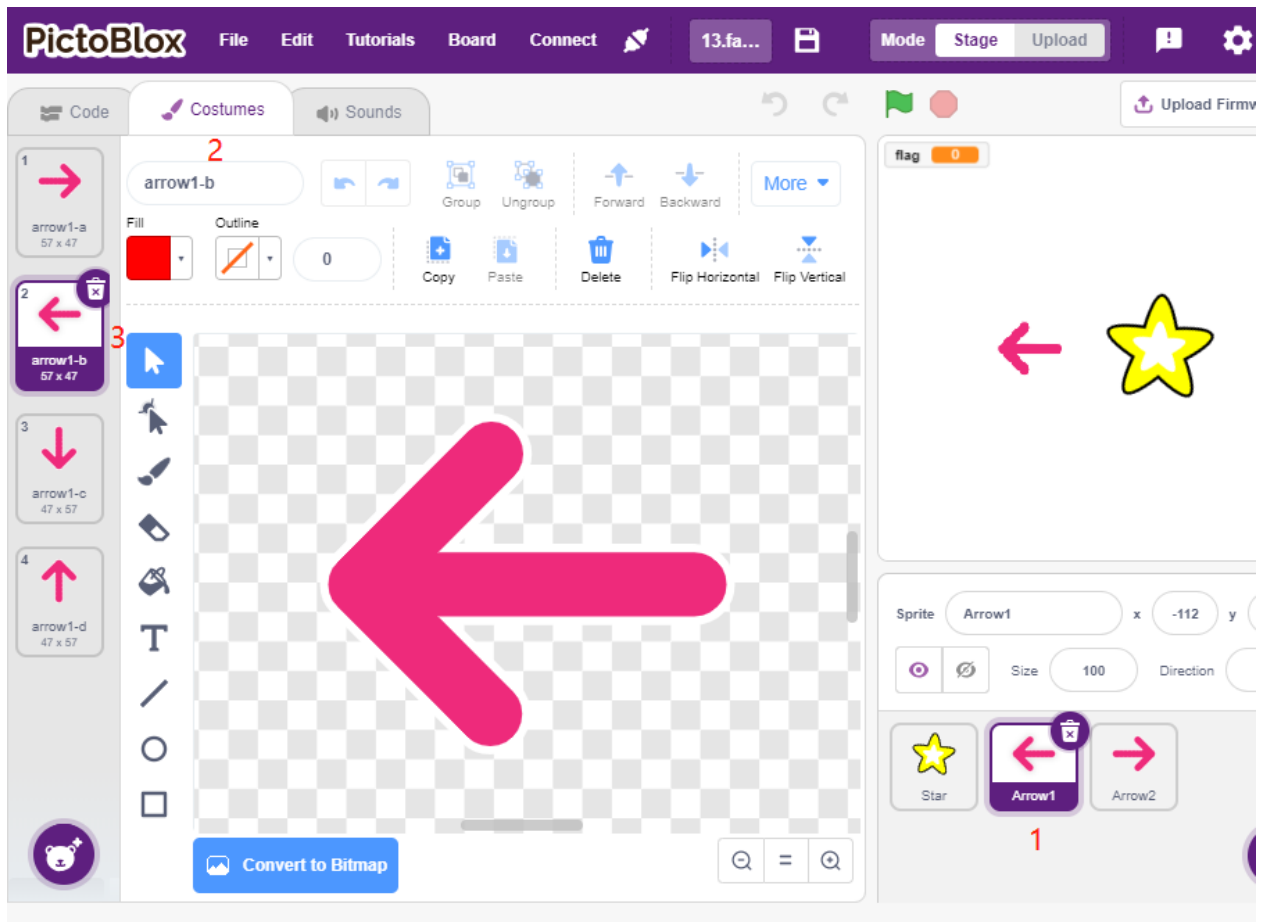
Das gewünschte Ergebnis ist, mit 2 Pfeilsprites die Uhrzeigersinn- und die Gegen-Uhrzeigersinn-Drehung des Motors und des Sternsprites zu steuern. Ein Klick auf den Sternsprite stoppt die Drehung des Motors.

1. Sprites hinzufügen

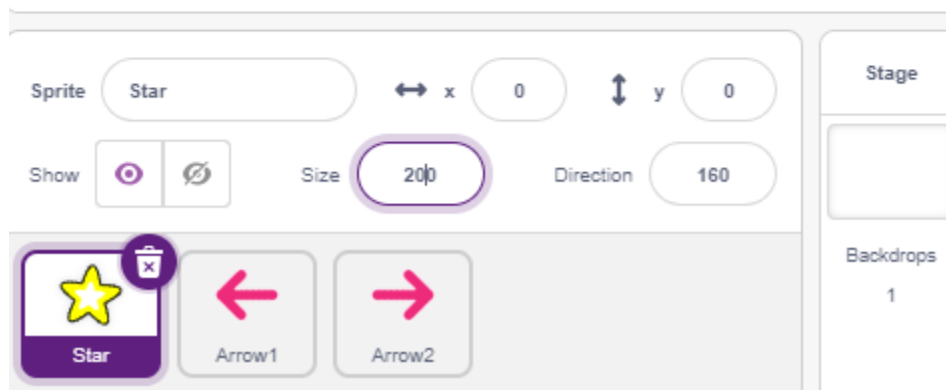
Löschen Sie den Standardsprite, wählen Sie dann den **Star**-Sprite und den **Arrow1**-Sprite und kopieren Sie **Arrow1** einmal.



Unter der Option **Costumes** ändern Sie den **Arrow1**-Sprite zu einem anderen Richtungskostüm.



Passen Sie die Größe und Position des Sprites entsprechend an.

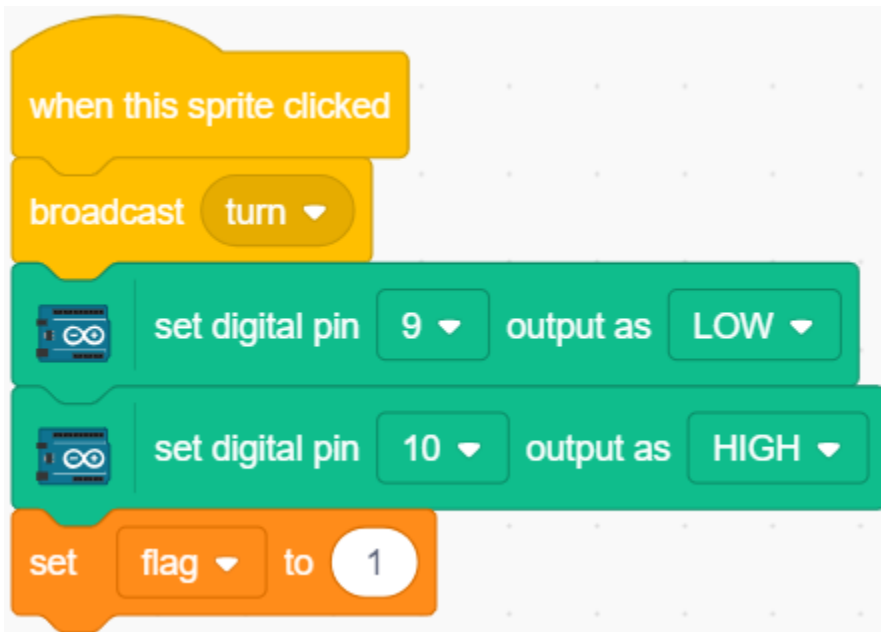


2. Linker Pfeil-Sprite

Wenn man auf diesen Sprite klickt, sendet er eine Nachricht - drehen, setzt dann den digitalen Pin 9 auf niedrig und Pin 10 auf hoch und setzt die Variable **flag** auf 1. Wenn Sie auf den linken Pfeil klicken, dreht sich der Motor gegen den Uhrzeigersinn. Wenn Ihre Drehung im Uhrzeigersinn ist, tauschen Sie die Positionen von Pin 9 und Pin 10.

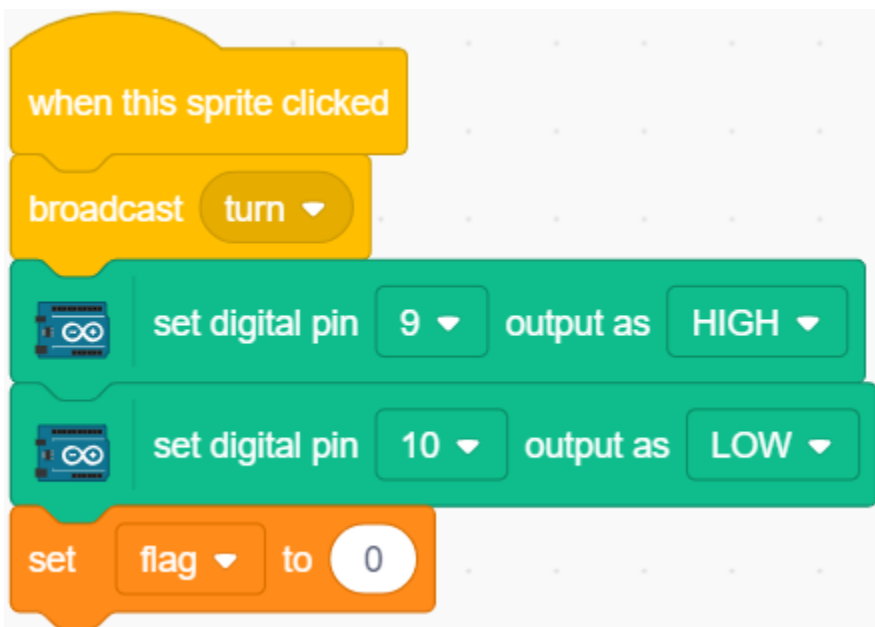
Zwei Dinge sind hier zu beachten:

- **[broadcast]**: aus der **Events**-Palette, wird verwendet, um eine Nachricht an die anderen Sprites zu senden. Wenn die anderen Sprites diese Nachricht erhalten, führen sie ein bestimmtes Ereignis aus. Hier ist es **turn**. Wenn der **star**-Sprite diese Nachricht erhält, führt er das Dreh-Skript aus.
- Variable **flag**: Die Drehrichtung des Sternsprites wird durch den Wert von **flag** bestimmt. Wenn Sie die Variable **flag** erstellen, müssen Sie sicherstellen, dass sie für alle Sprites gilt.



3. Rechter Pfeil-Sprite

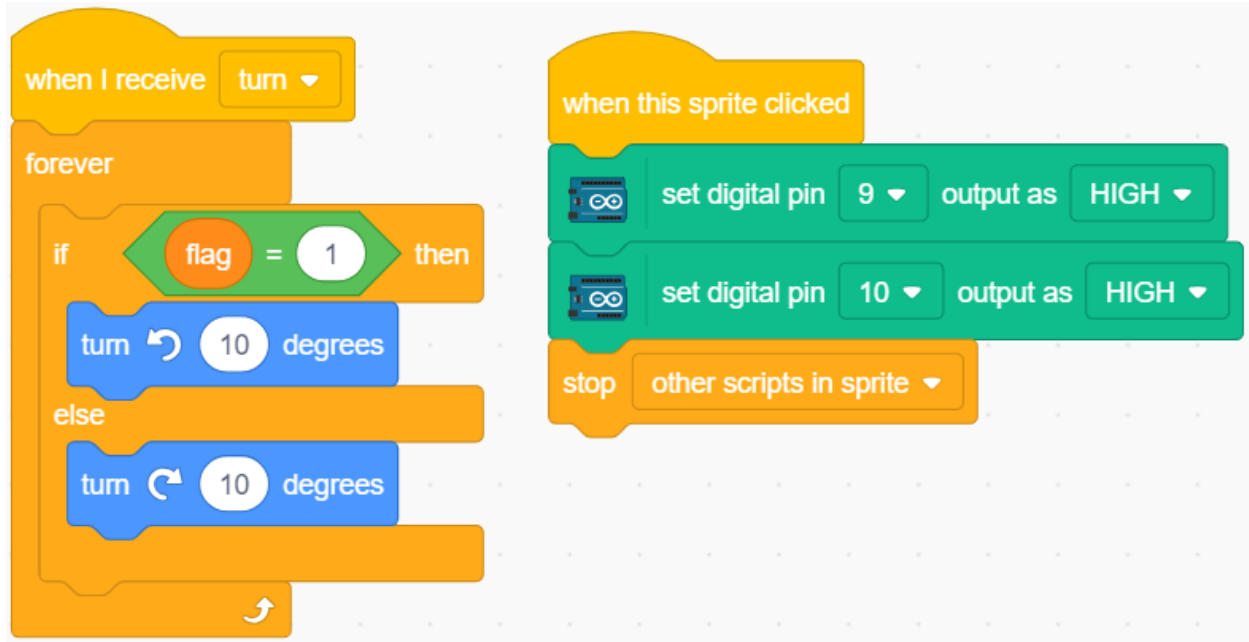
Wenn Sie auf diesen Sprite klicken, senden Sie eine Nachricht zum Drehen, setzen Sie dann den digitalen Pin 9 hoch und Pin 10 niedrig, damit sich der Motor im Uhrzeigersinn dreht und setzen Sie die Variable **flag** auf 0.



4. Stern-Sprite

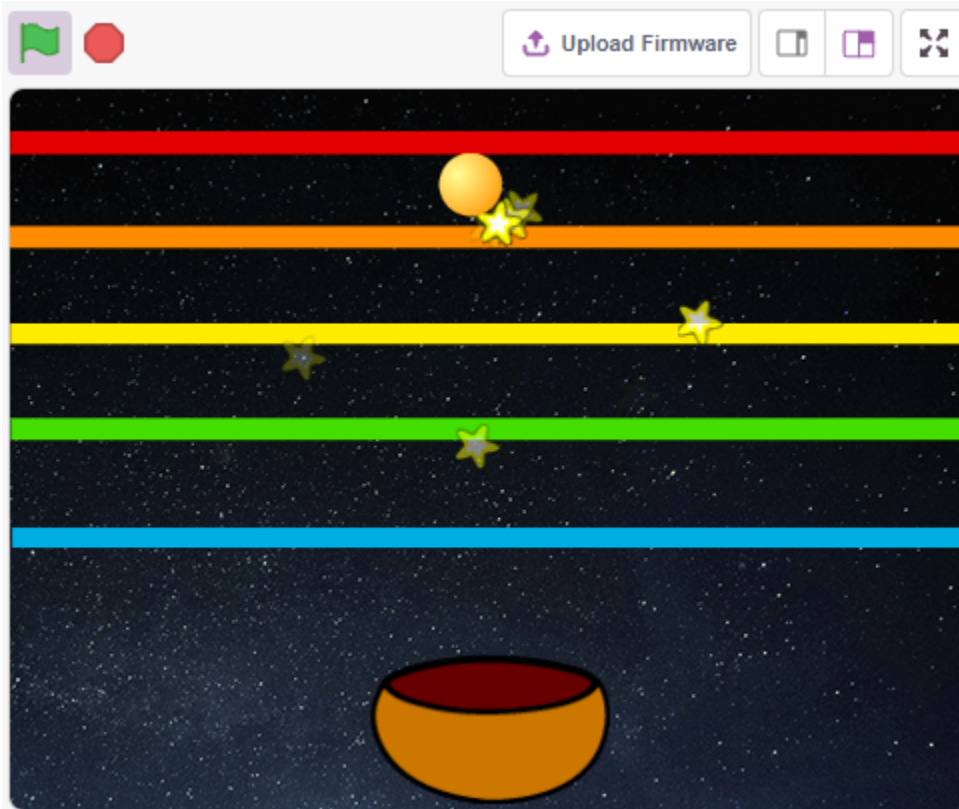
Hier sind 2 Ereignisse enthalten:

- Wenn der **star**-Sprite die gesendete Nachricht zum Drehen empfängt, bestimmt er den Wert von flag; ist flag 1, dreht er sich 10 Grad nach links, sonst umgekehrt. Da es sich im [FOREVER]-Block befindet, wird es weiter drehen.
- Wenn Sie auf diesen Sprite klicken, setzen Sie beide Pins des Motors auf hoch, um ihn zu stoppen und stoppen die anderen Skripte in diesem Sprite.



8.15 2.12 Lichtempfindlicher Ball

In diesem Projekt nutzen wir einen Fotowiderstand, um den Ball auf der Bühne nach oben fliegen zu lassen. Platzieren Sie Ihre Hand über dem Fotowiderstand, um die Lichtintensität zu steuern, die er erhält. Je näher Ihre Hand am Fotowiderstand ist, desto kleiner wird sein Wert und desto höher fliegt der Ball auf der Bühne, ansonsten fällt er. Wenn der Ball die Schnur berührt, ertönt ein angenehmes Geräusch und ein funkelndes Sternenlicht erscheint.



8.15.1 Was Sie lernen werden

- Ein Sprite mit Farben zu füllen
- Berührung zwischen den Sprites

8.15.2 Benötigte Komponenten

Für dieses Projekt benötigen wir folgende 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 einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Fotowiderstand</i>	

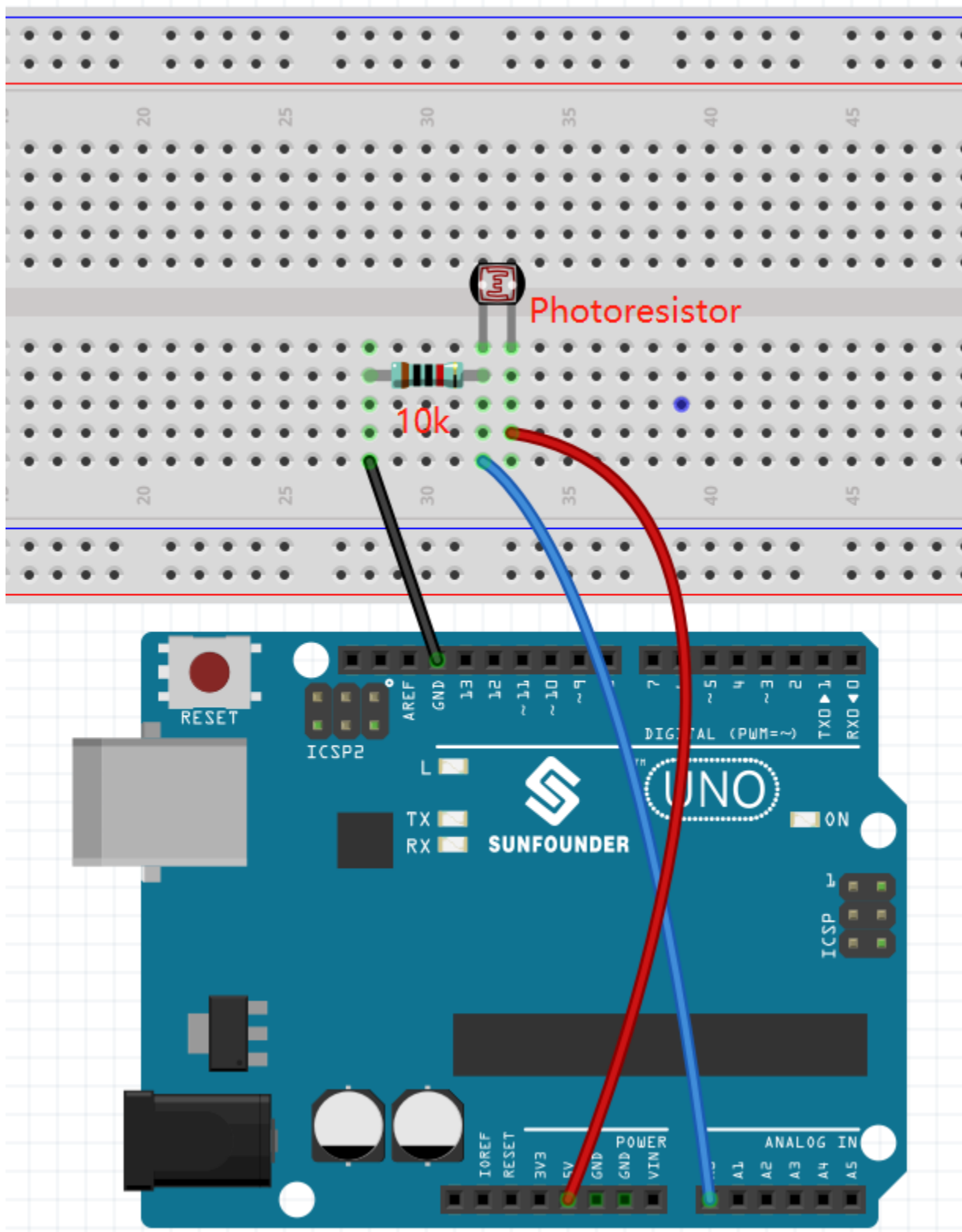
8.15.3 Schaltkreis aufbauen

Ein Fotowiderstand oder Fotodiode ist ein lichtgesteuerter variabler Widerstand. Der Widerstand eines Fotowiderstands verringert sich mit zunehmender Lichteinstrahlung.

Bauen Sie den Schaltkreis gemäß dem folgenden Diagramm auf.

Verbinden Sie ein Ende des Fotowiderstands mit 5V, das andere Ende mit A0 und schließen Sie einen 10K Widerstand in Reihe mit GND an diesem Ende an.

Wenn also die Lichtintensität zunimmt, verringert sich der Widerstand des Fotowiderstands, die Spannungsteilung des 10K Widerstands erhöht sich und der von A0 erhaltene Wert wird größer.

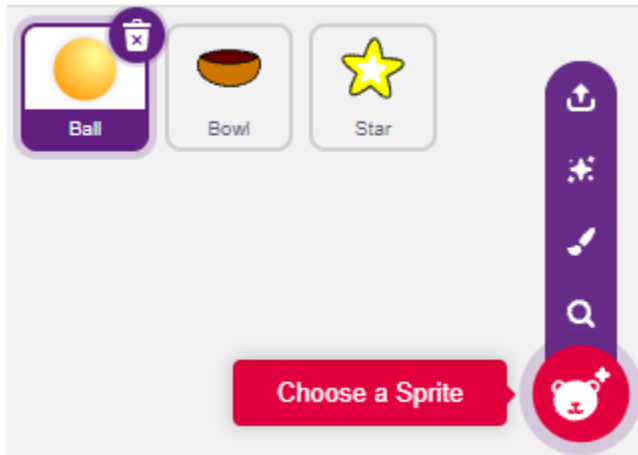


8.15.4 Programmierung

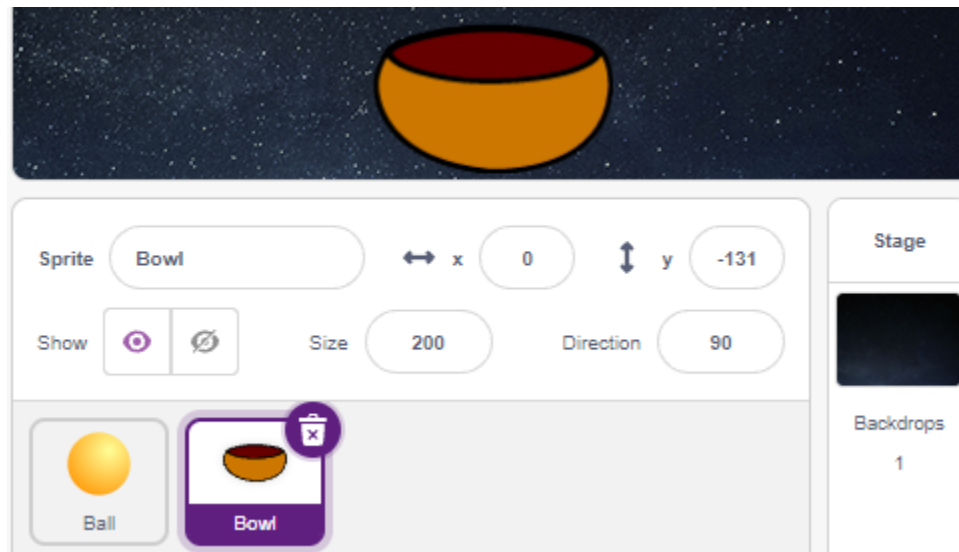
Das gewünschte Ergebnis ist, dass, je näher Ihre Hand am Fotowiderstand ist, der Ball-Sprite auf der Bühne kontinuierlich nach oben geht, sonst fällt er auf den Bowl-Sprite. Wenn er beim Aufsteigen oder Fallen den Line-Sprite berührt, gibt er einen musikalischen Klang von sich und sendet Stern-Sprites in alle Richtungen.

1. Sprite und Hintergrund auswählen

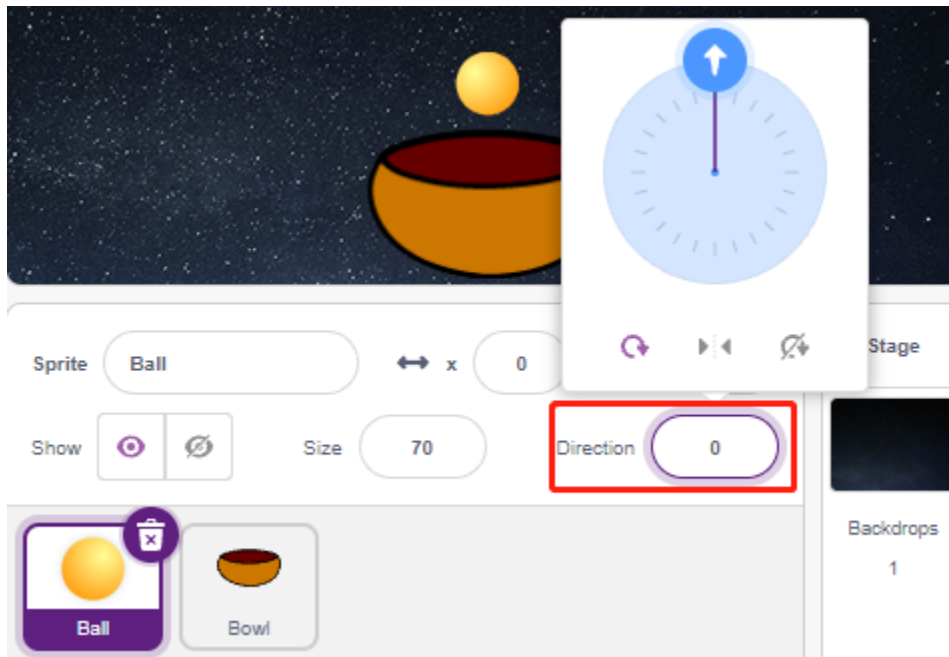
Löschen Sie den Standard-Sprite und wählen Sie den **Ball**, **Bowl** und **Star** Sprite aus.



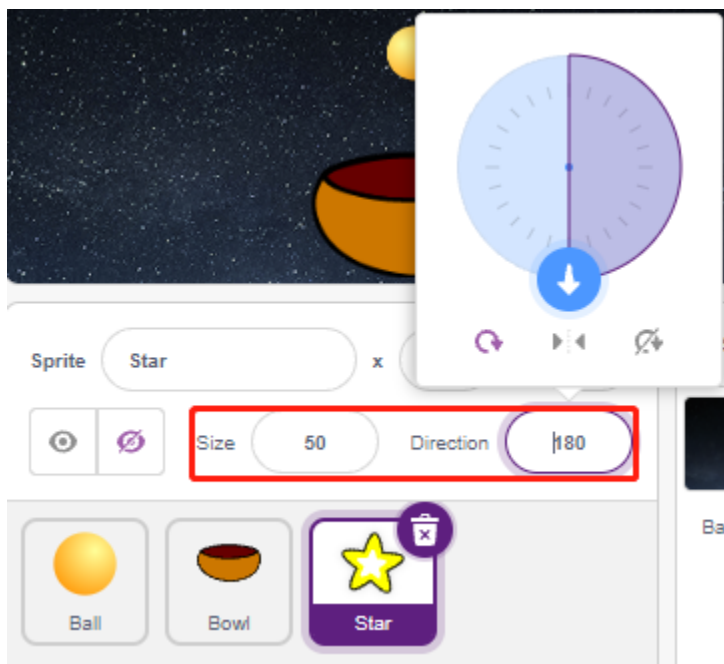
Verschieben Sie den **Bowl** Sprite in die untere Mitte der Bühne und vergrößern Sie seine Größe.



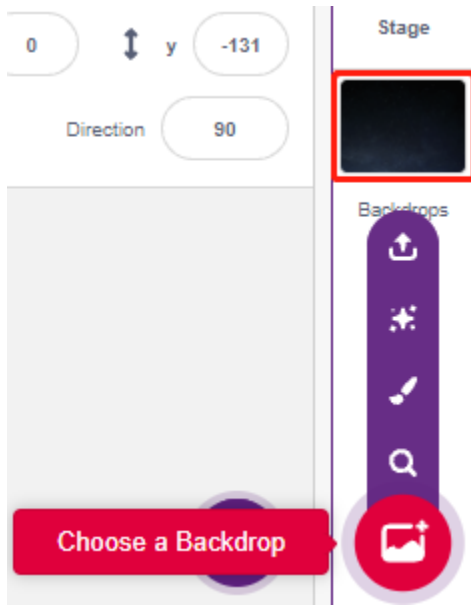
Da wir ihn nach oben bewegen müssen, setzen Sie die Richtung des **Ball** Sprites auf 0.



Stellen Sie die Größe und Richtung des **Star** Sprites auf 180 ein, da er nach unten fallen muss, oder Sie können ihn in einen anderen Winkel ändern.

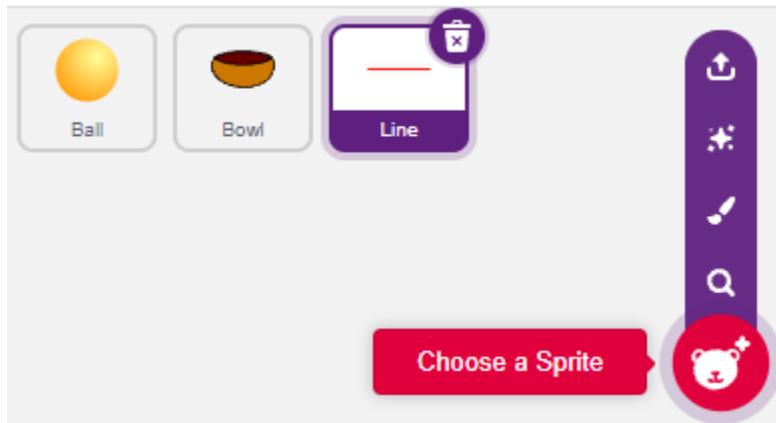


Jetzt fügen Sie den **Stars** Hintergrund hinzu.

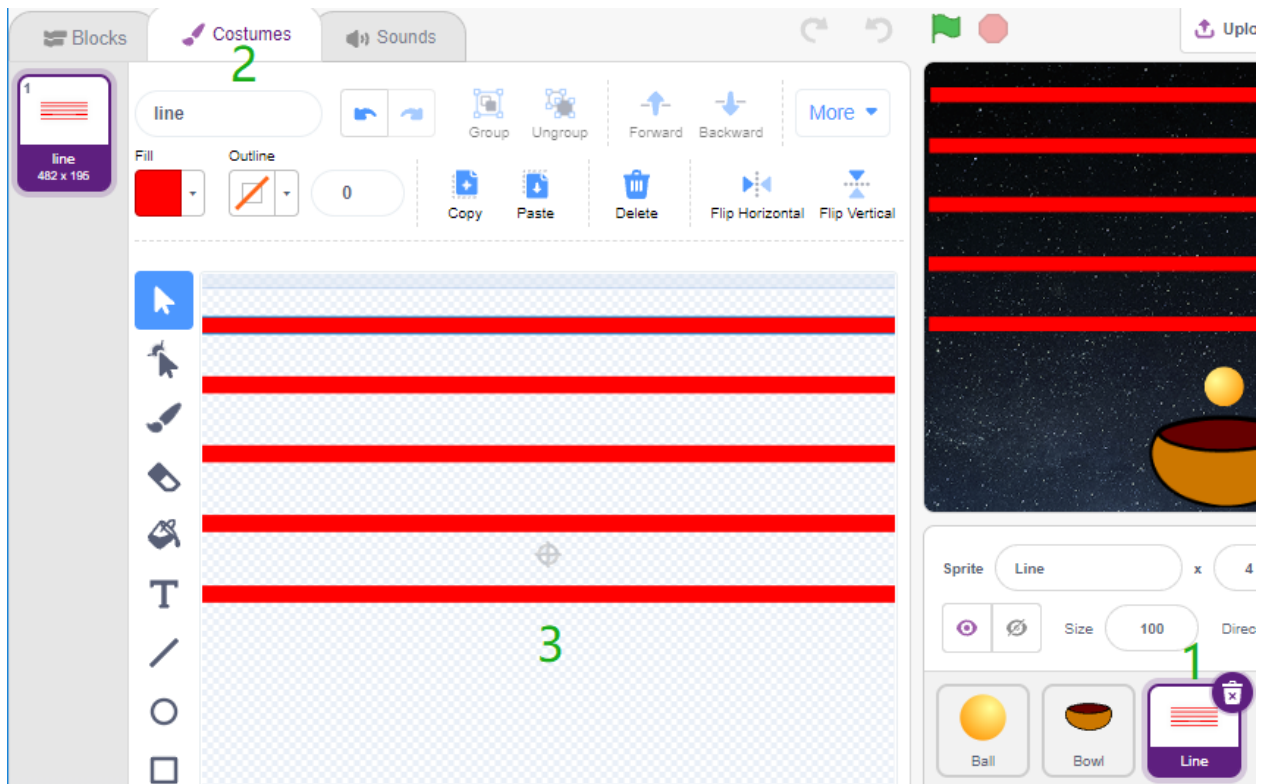


2. Einen Linien-Sprite zeichnen

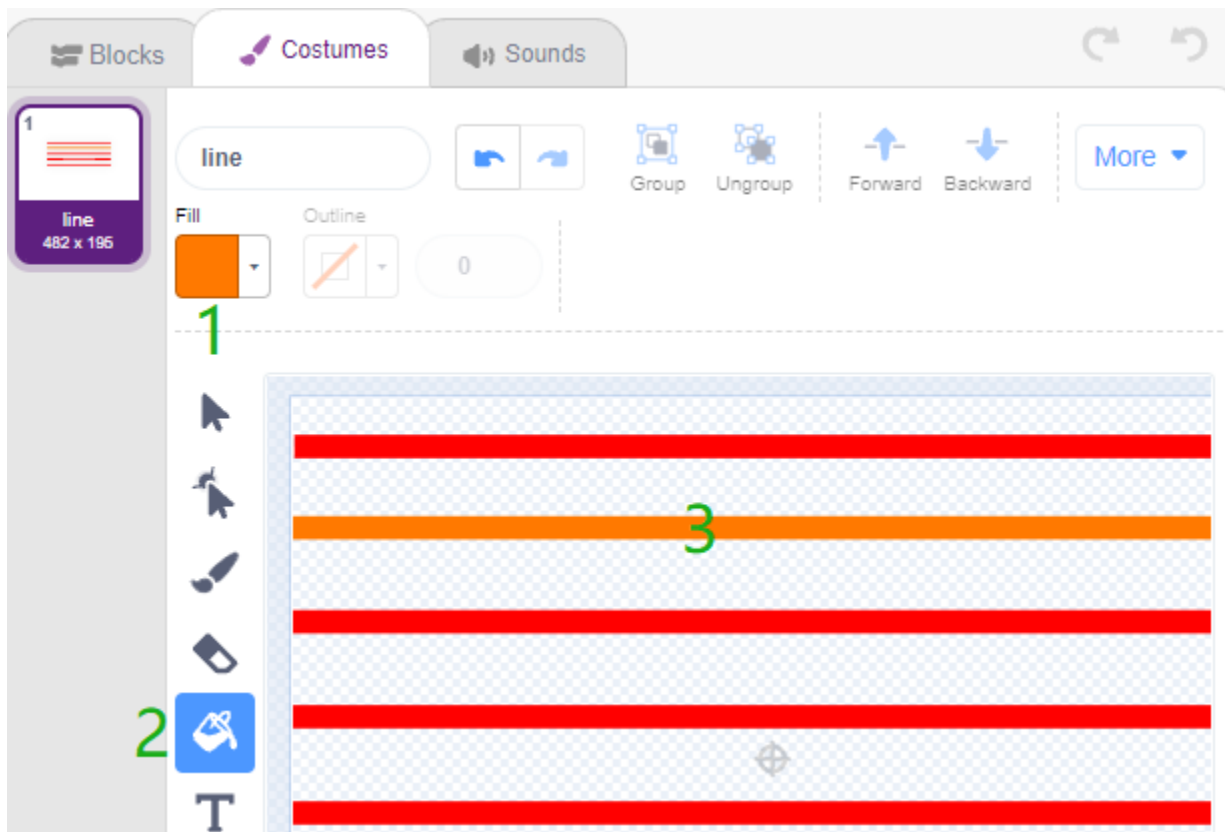
Fügen Sie einen Linien-Sprite hinzu.



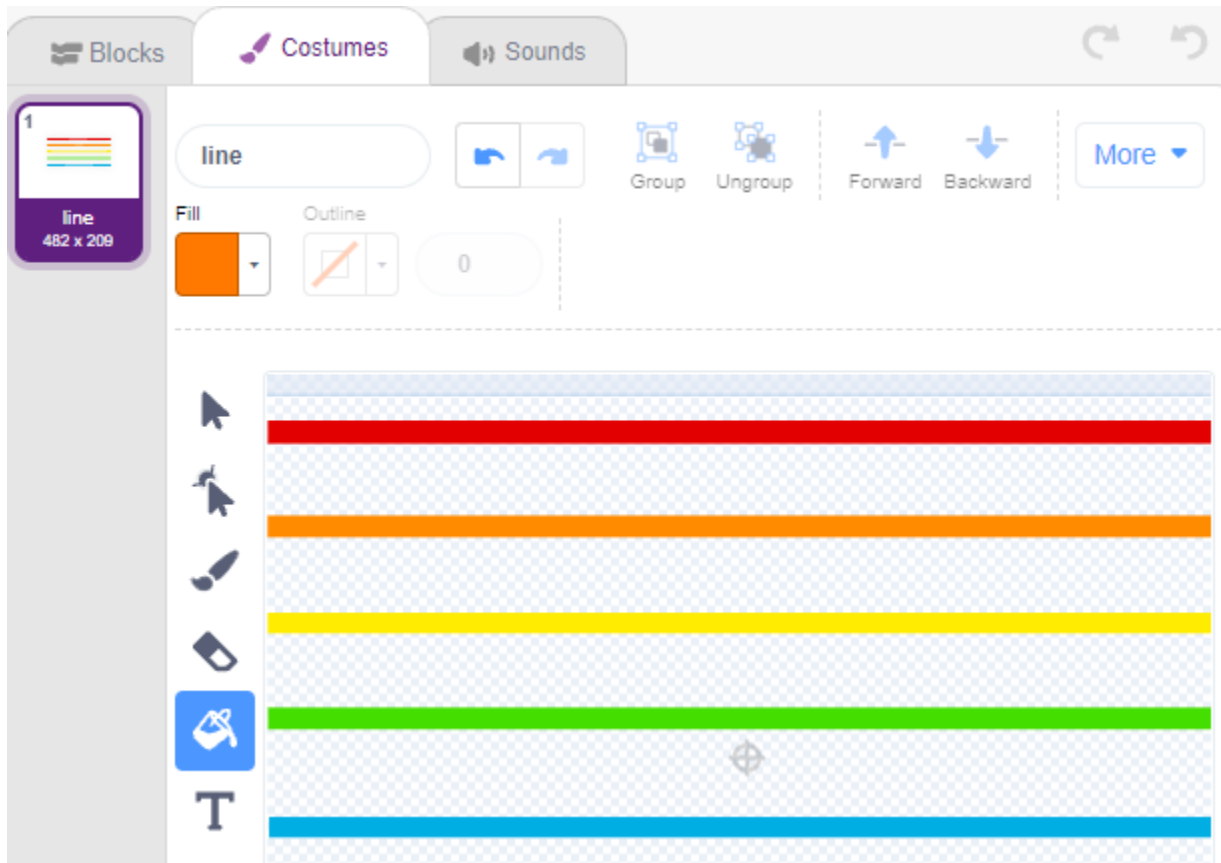
Gehen Sie zur **Costumes**-Seite des **Line** Sprites, verringern Sie die Breite der roten Linie auf der Leinwand etwas, kopieren Sie sie dann 5-mal und richten Sie die Linien aus.



Jetzt füllen Sie die Linien mit verschiedenen Farben. Wählen Sie zuerst eine Farbe, die Ihnen gefällt, klicken Sie dann auf das **Fill**-Werkzeug und fahren Sie mit der Maus über die Linie, um sie zu füllen.



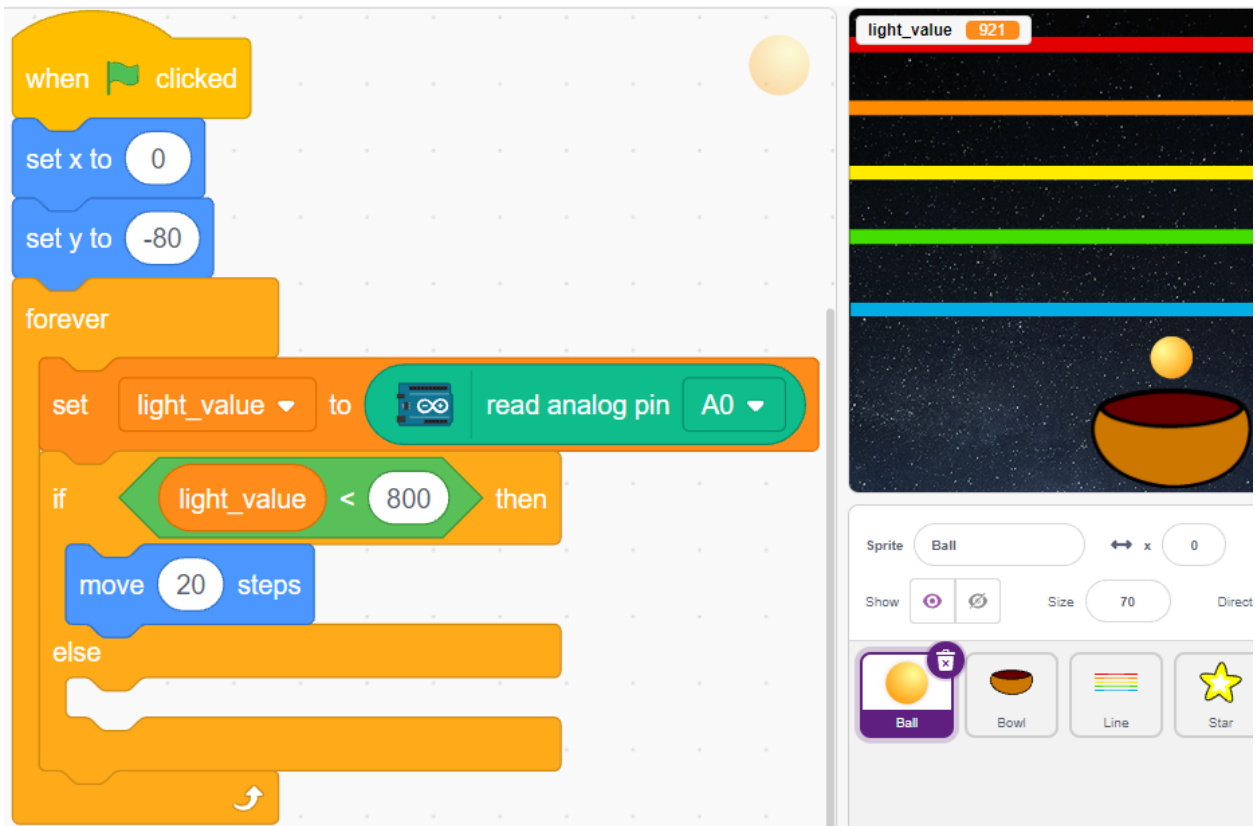
Verfahren Sie auf die gleiche Weise, um die Farbe der anderen Linien zu ändern.



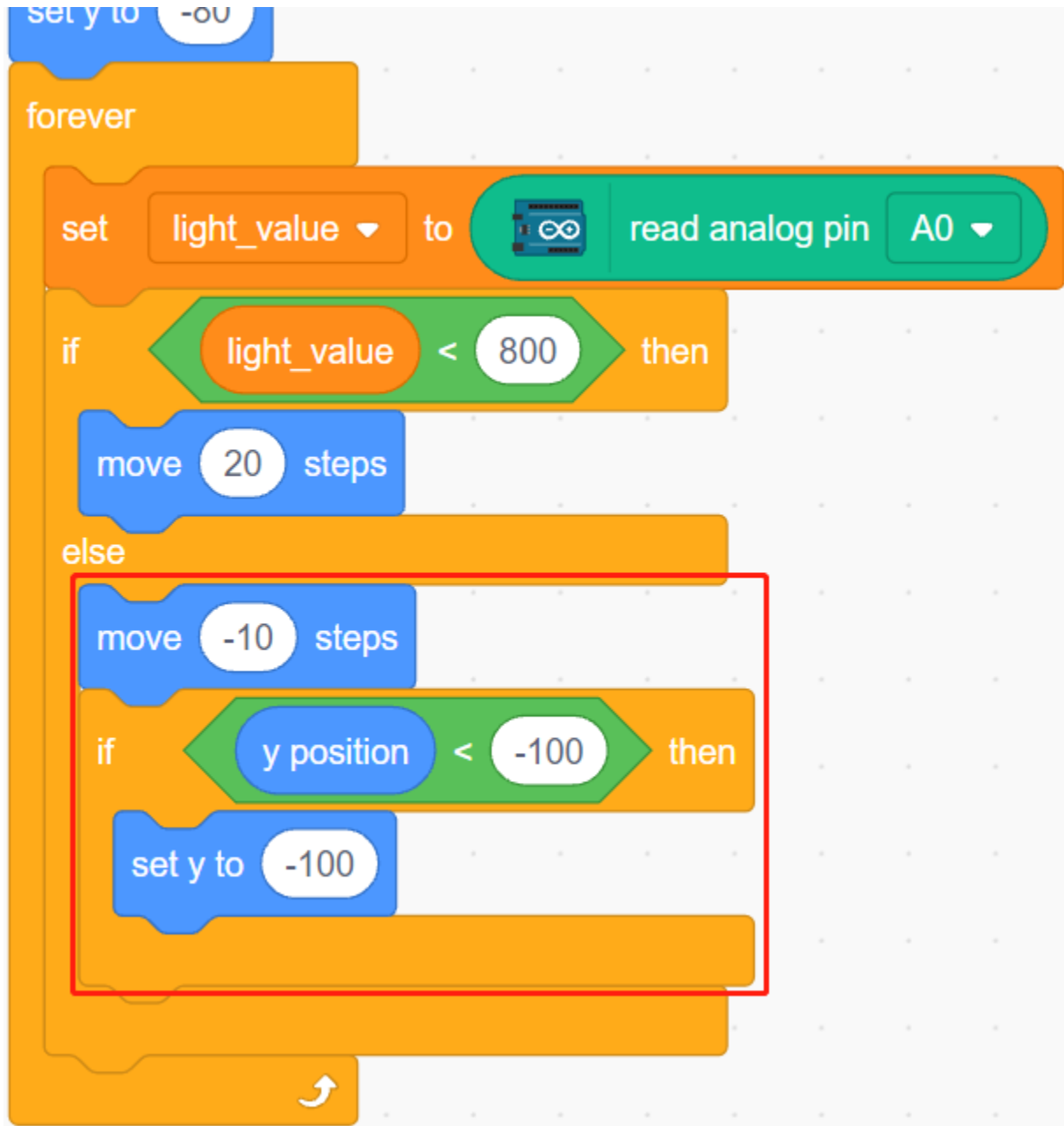
3. Skript für den Ball-Sprite

Setzen Sie die Anfangsposition des **Ball** Sprites. Wenn der Lichtwert kleiner als 800 ist (es kann ein anderer Wert sein, abhängig von Ihrer aktuellen Umgebung), soll der Ball nach oben bewegen.

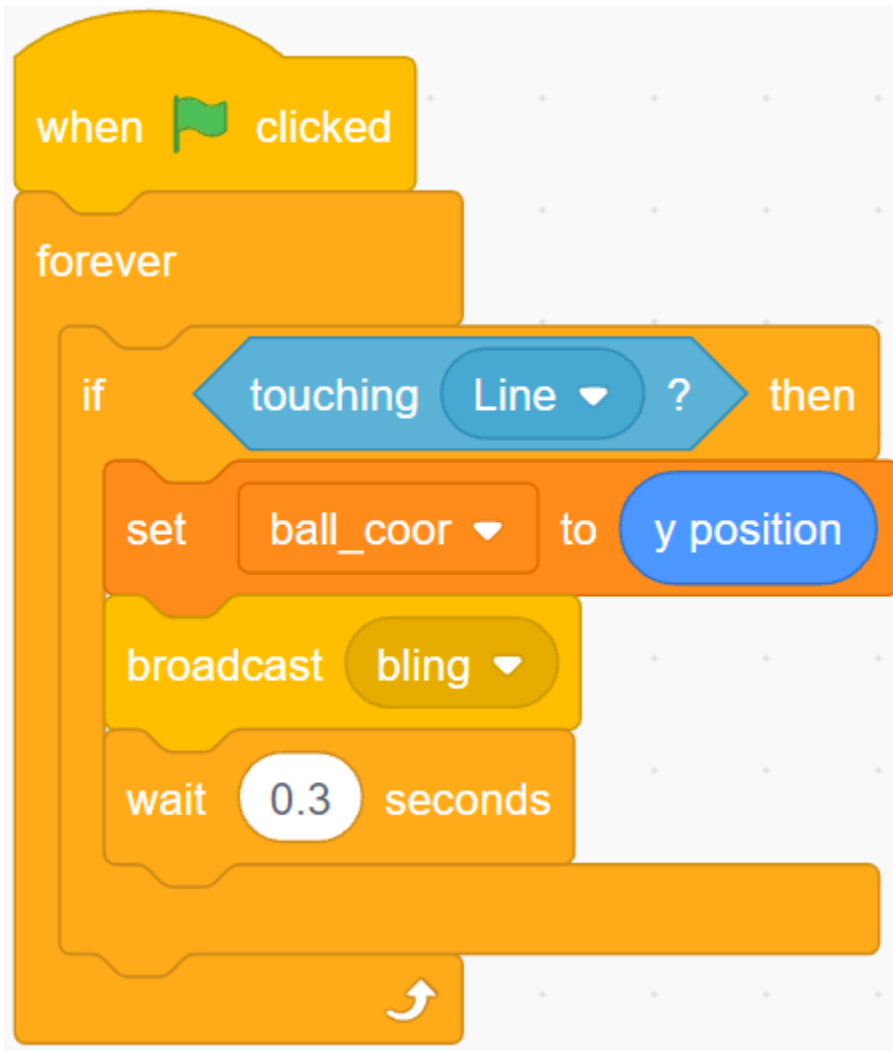
Sie können die Variable Lichtwert auf der Bühne anzeigen lassen, um die Lichtintensitätsänderung jederzeit zu beobachten.



Andernfalls wird der **Ball** Sprite fallen und seine Y-Koordinate auf ein Minimum von -100 begrenzen. Dies kann angepasst werden, damit es so aussieht, als ob er auf den **Bowl** Sprite fällt.

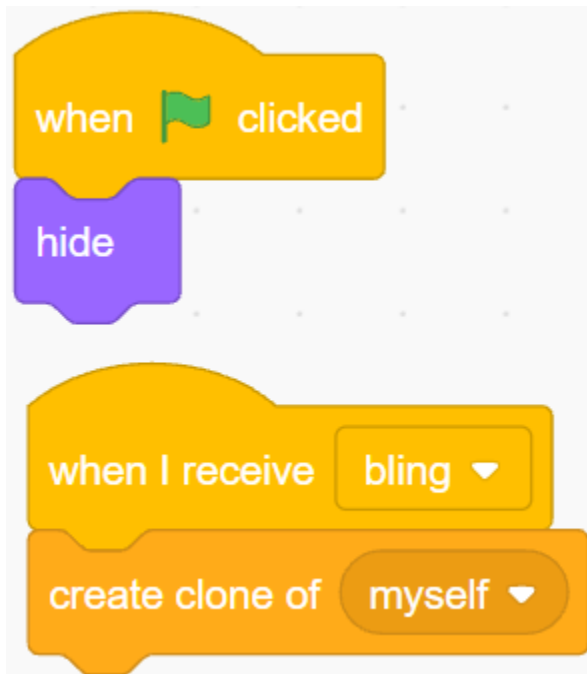


Wenn der **Line** Sprite getroffen wird, wird die aktuelle Y-Koordinate in die Variable **ball_coor** gespeichert und eine **Bling** Nachricht gesendet.

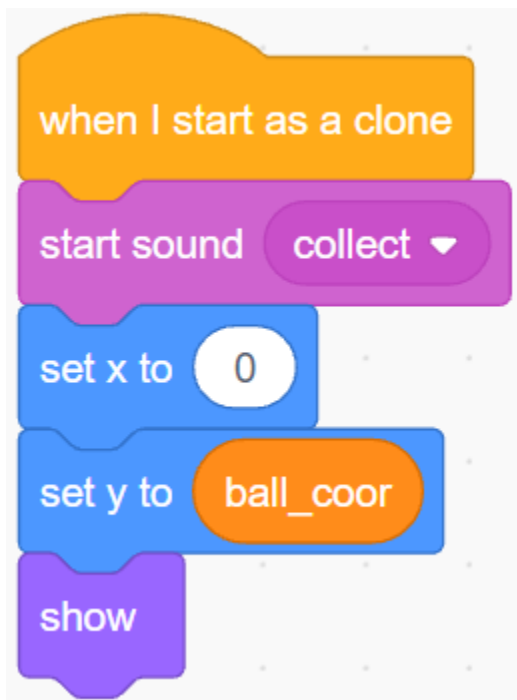


4. Skript für den Stern-Sprite

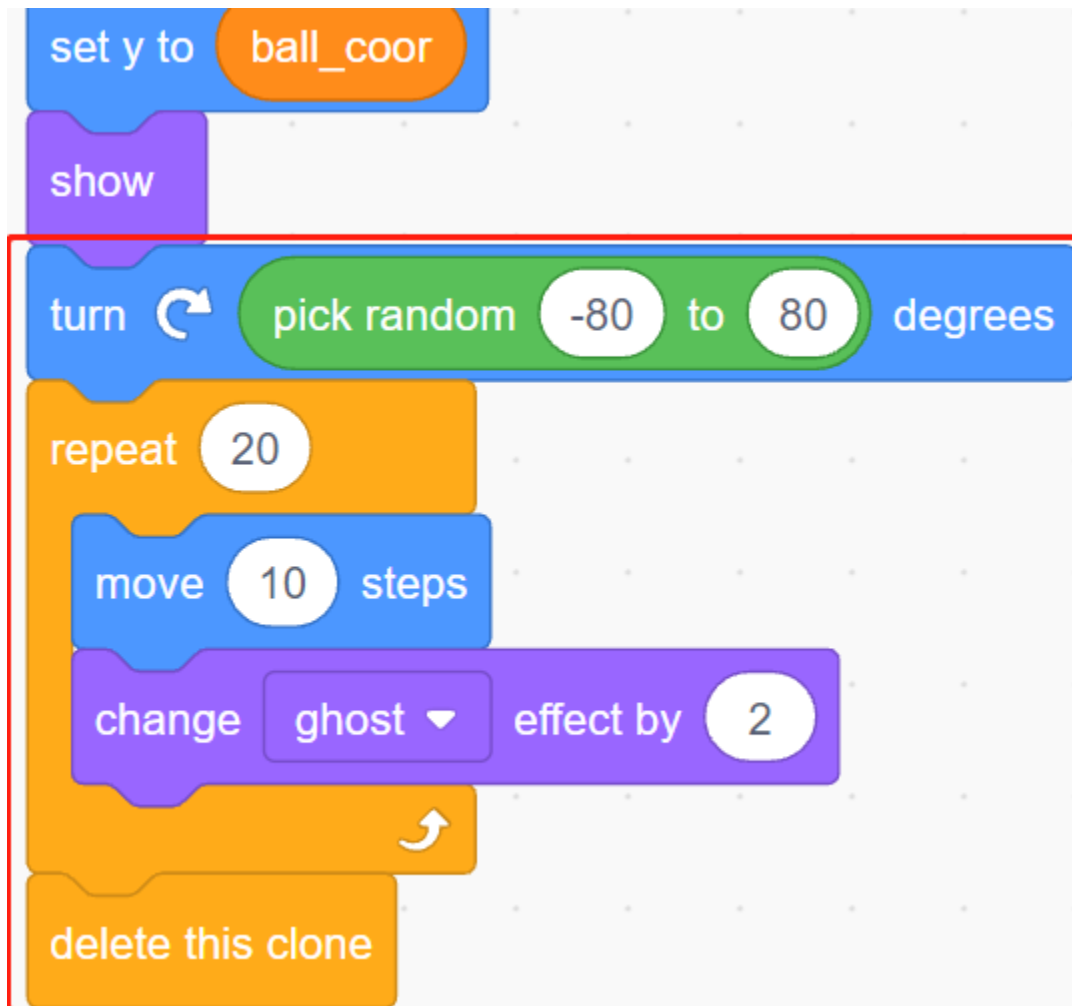
Wenn das Skript startet, verstecken Sie zuerst den **Star** Sprite. Wenn die **Bling** Nachricht empfangen wird, klonen den **Star** Sprite.



Wenn der **Star** Sprite als Klon erscheint, spiele den Soundeffekt ab und setze seine Koordinaten synchron zum **Ball** Sprite.



Erstelle den Effekt des erscheinenden **Star** Sprites und passe ihn bei Bedarf an.

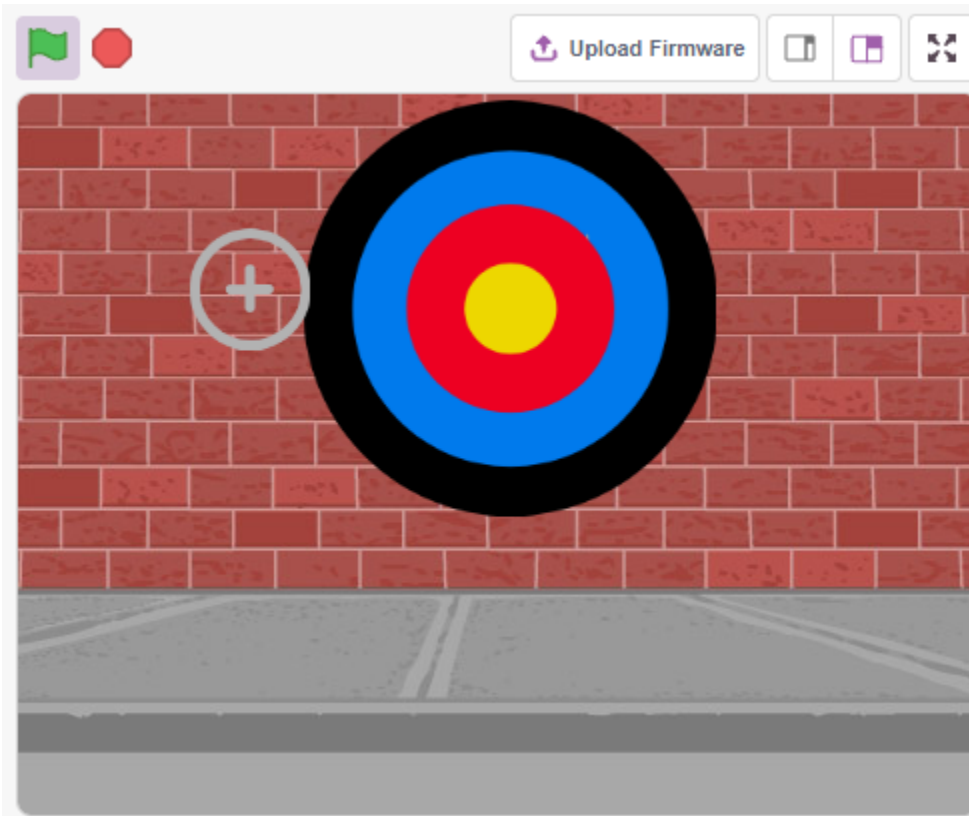


8.16 2.13 SPIEL - Schießen

Haben Sie schon einmal diese Schießspiele im Fernsehen gesehen? Je näher ein Teilnehmer mit einer Kugel das Ziel am Bullseye trifft, desto höher ist seine Punktzahl.

Heute machen wir ebenfalls ein Schießspiel in Scratch. In diesem Spiel soll das Fadenkreuz so nah wie möglich am Bullseye schießen, um eine höhere Punktzahl zu erzielen.

Klicken Sie auf die grüne Flagge, um zu starten. Verwenden Sie das Hindernisvermeidungsmodul, um eine Kugel abzufeuern.



8.16.1 Was Sie lernen werden

- Wie das Hindernisvermeidungsmodul funktioniert und welcher Winkelbereich gegeben ist
- Verschiedene Sprites malen
- Farben berühren

8.16.2 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+	

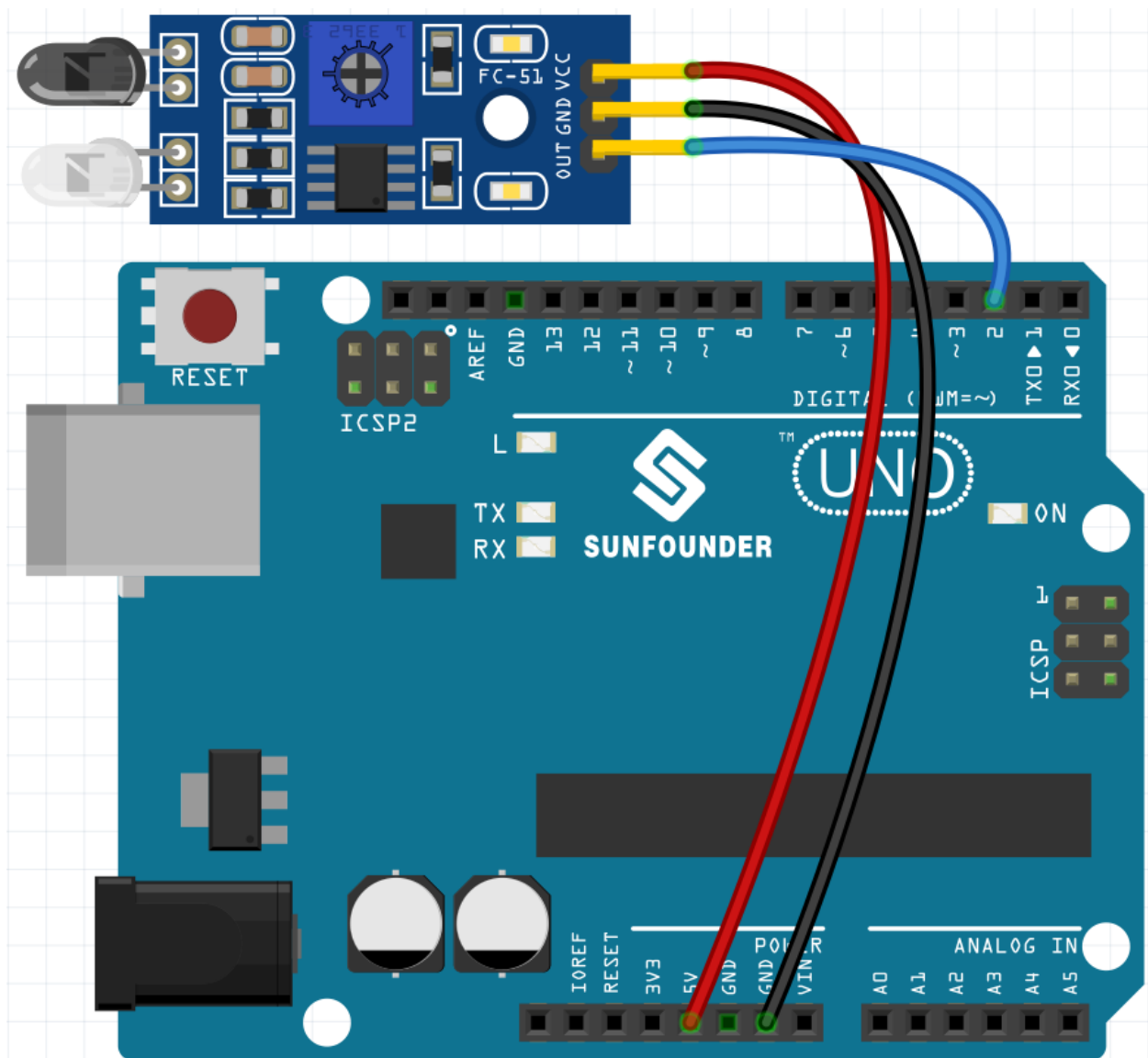
Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>Hindernisvermeidungsmodul</i>	

8.16.3 Schaltung aufbauen

Das Hindernisvermeidungsmodul ist ein infraroter Annäherungssensor mit einstellbarer Entfernung, dessen Ausgabe normalerweise hoch ist und auf niedrig schaltet, wenn ein Hindernis erkannt wird.

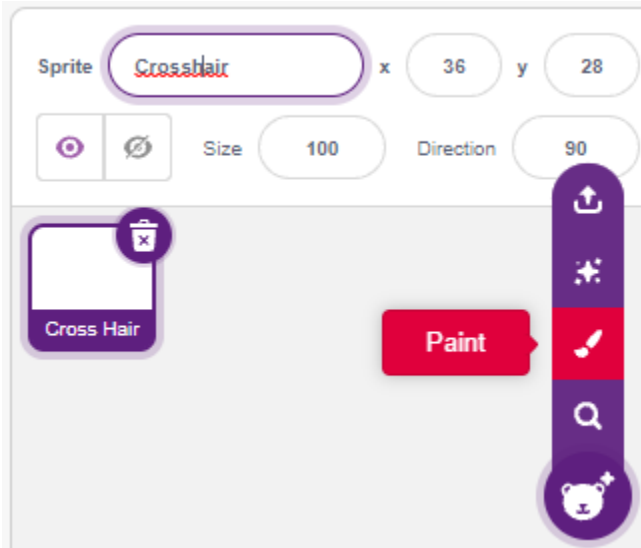
Bauen Sie die Schaltung jetzt gemäß dem untenstehenden Diagramm auf.



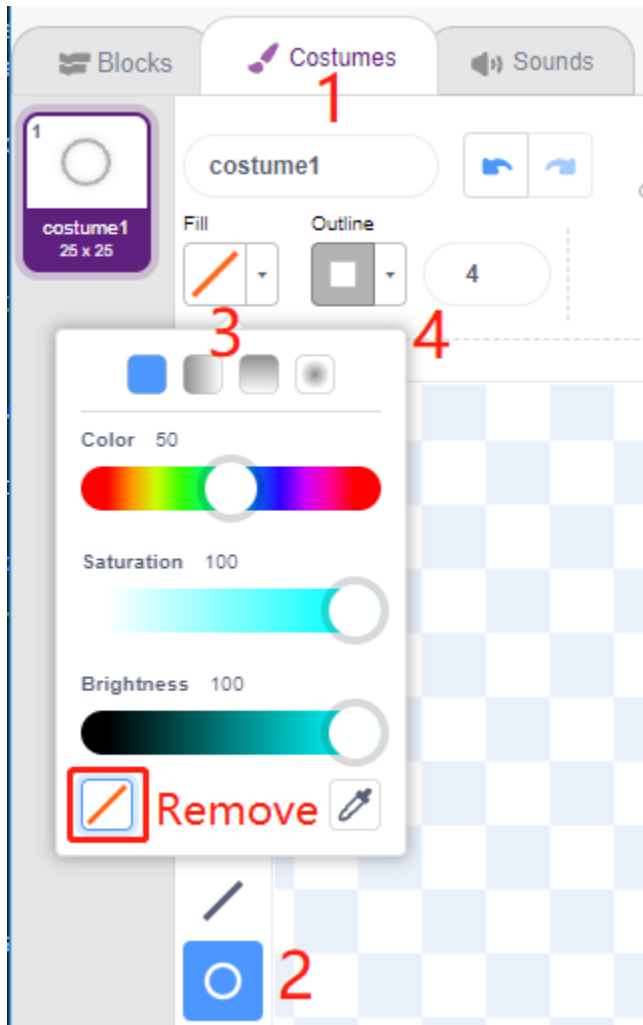
8.16.4 Programmierung

1. Das Fadenkreuz-Sprite malen

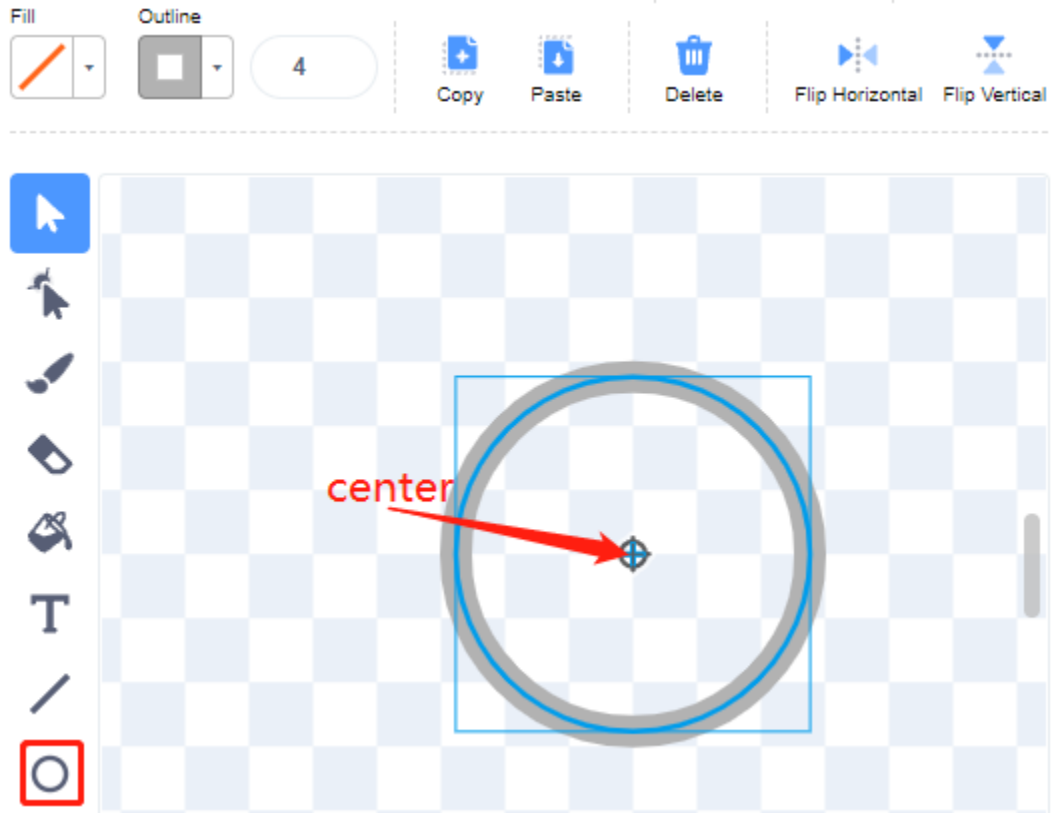
Löschen Sie das Standard-Sprite, wählen Sie den **Sprite**-Button und klicken Sie auf **Paint**. Ein leeres Sprite namens **Sprite1** erscheint. Benennen Sie es in **Crosshair** um.



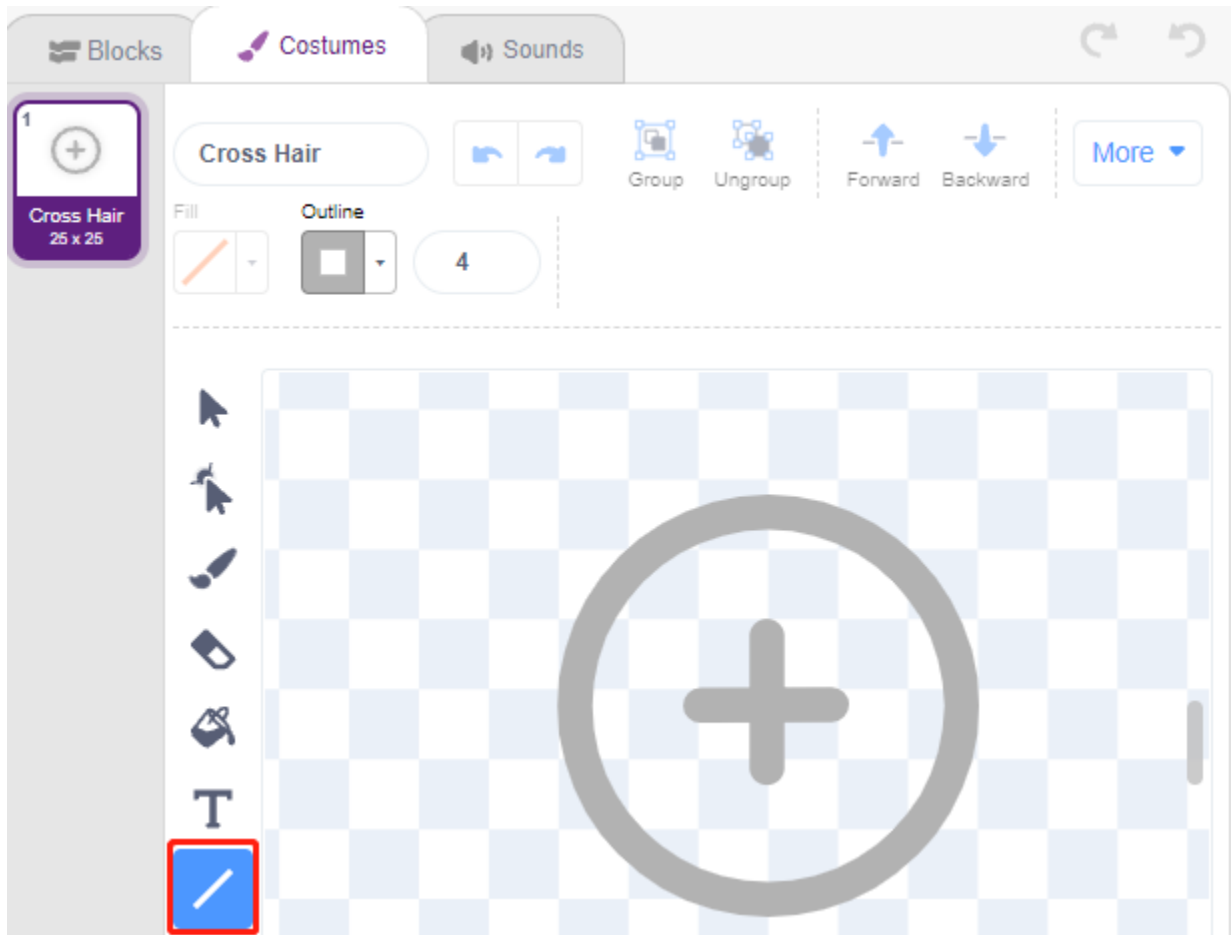
Gehen Sie zur **Costumes**-Seite des **Crosshair**-Sprites. Klicken Sie auf das **Circle**-Werkzeug, entfernen Sie die Füllfarbe und legen Sie die Farbe und Breite der Kontur fest.



Zeichnen Sie nun mit dem **Circle**-Werkzeug einen Kreis. Nach dem Zeichnen können Sie mit dem **Select**-Werkzeug den Kreis verschieben, sodass der Ursprungspunkt mit der Mitte der Leinwand ausgerichtet ist.

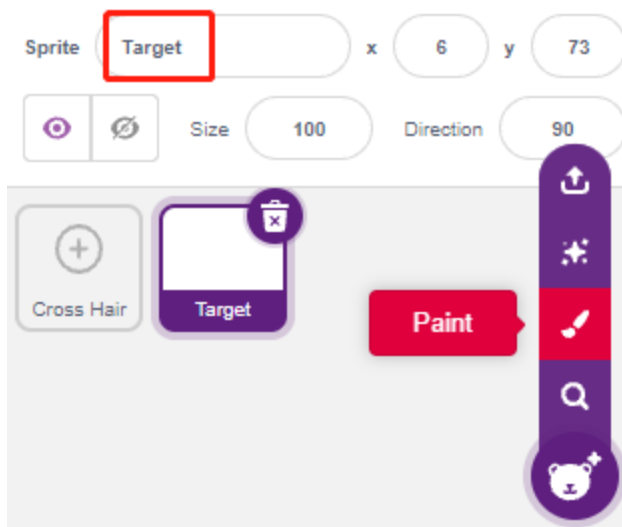


Verwenden Sie das **Line**-Werkzeug, um ein Kreuz innerhalb des Kreises zu zeichnen.

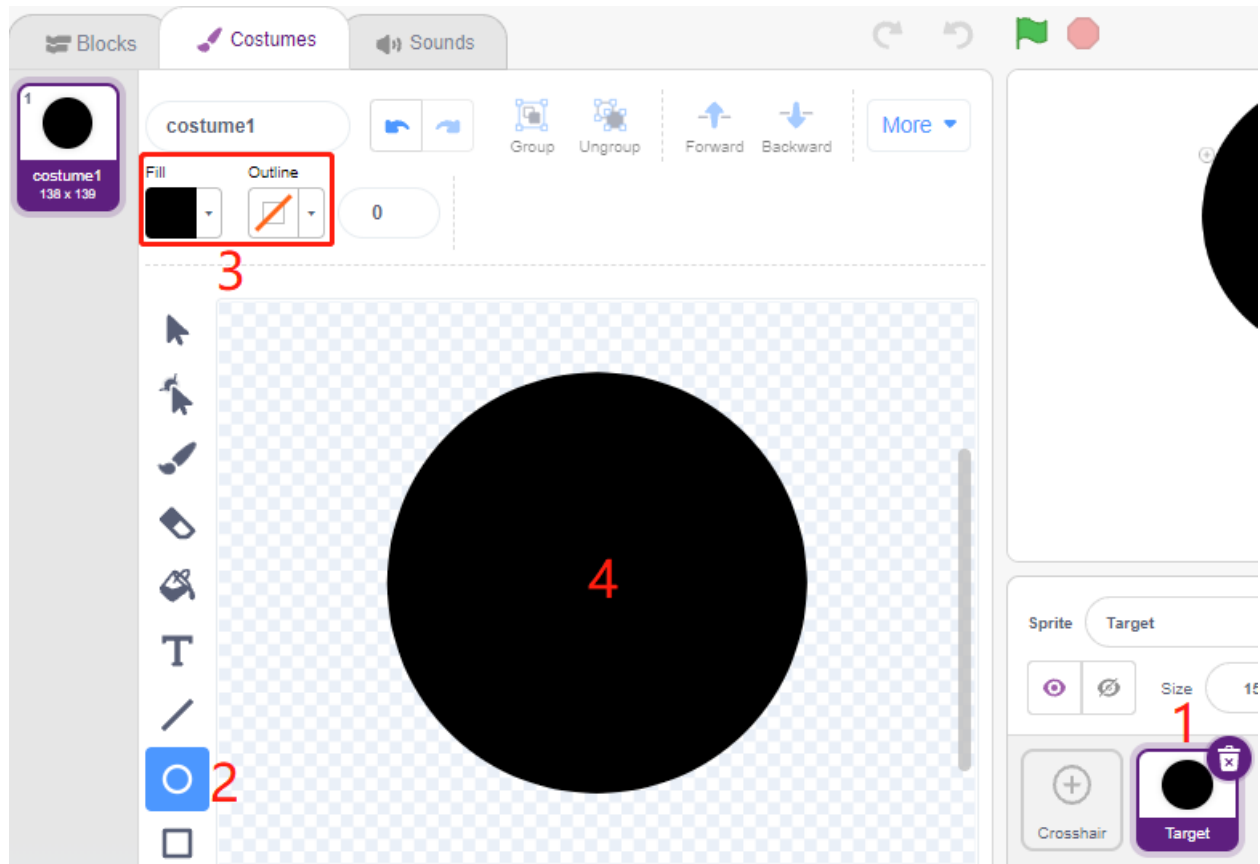


Das Ziel-Sprite malen

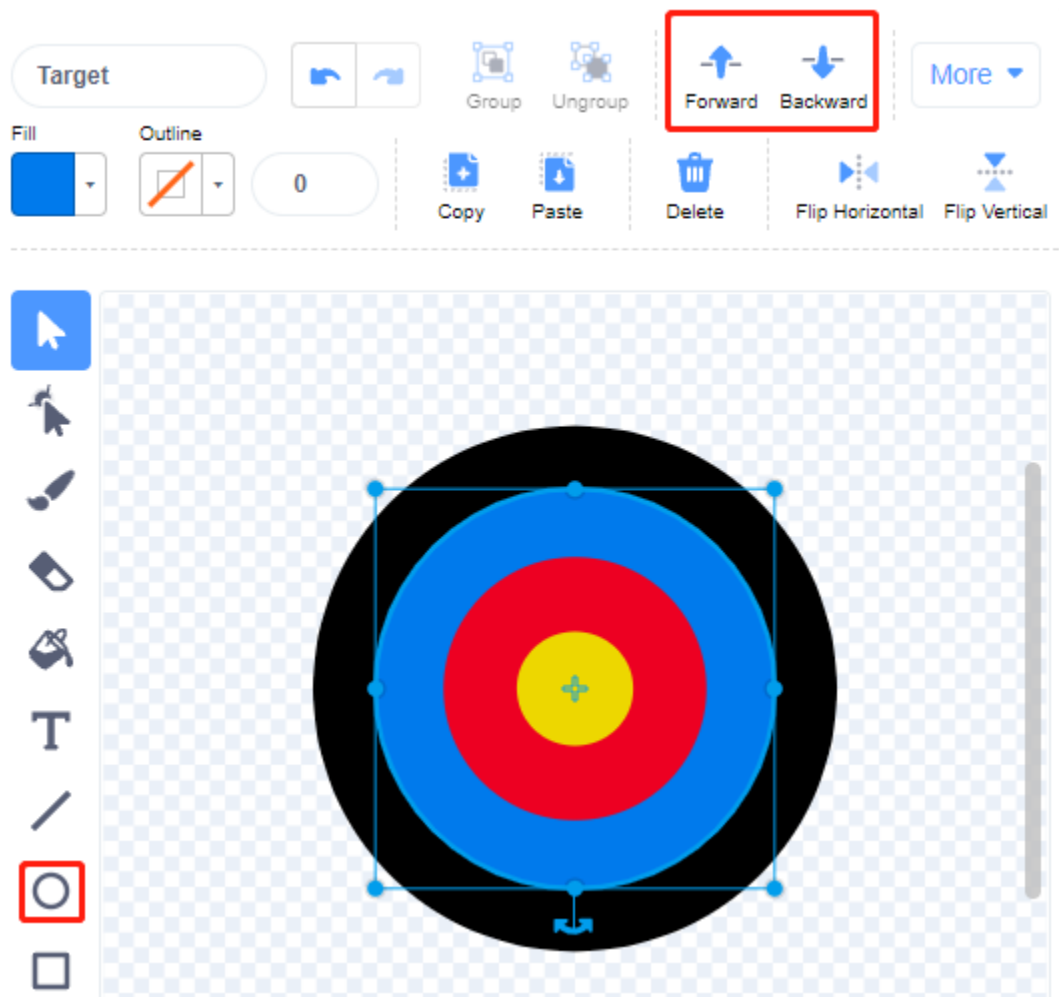
Erstellen Sie ein neues Sprite namens **Target**.



Gehen Sie zur Kostümseite des **Target**-Sprites, klicken Sie auf das **Circle**-Werkzeug, wählen Sie eine Füllfarbe aus, entfernen Sie die Kontur und malen Sie einen großen Kreis.

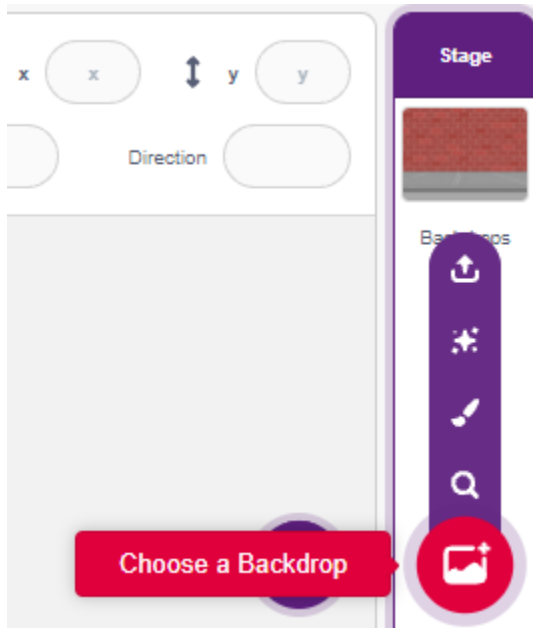


Verwenden Sie dieselbe Methode, um weitere Kreise mit jeweils unterschiedlichen Farben zu zeichnen, und Sie können die Position der überlappenden Kreise mit dem Werkzeug **Forward** oder **Backward** ändern. Beachten Sie, dass Sie auch das Werkzeug zum Verschieben der Kreise auswählen müssen, damit der Ursprung aller Kreise und die Mitte der Leinwand ausgerichtet sind.



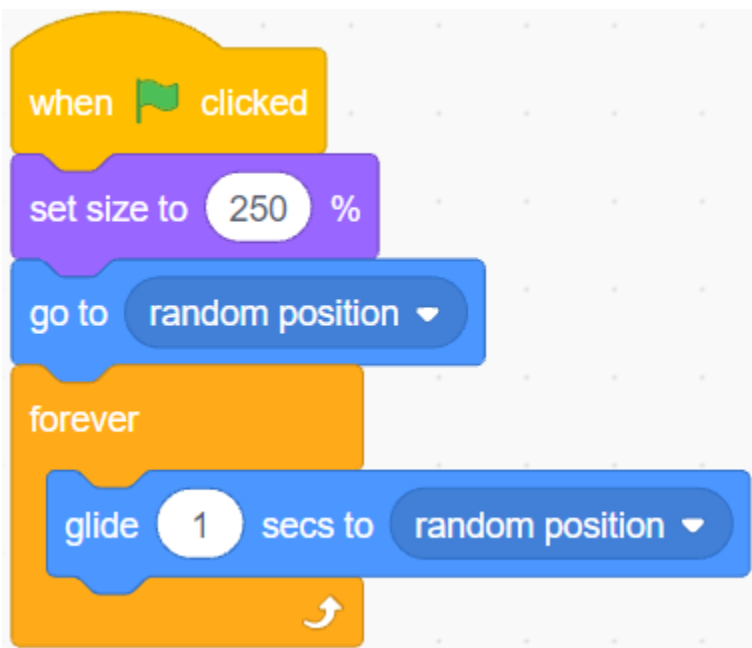
3. Einen Hintergrund hinzufügen

Fügen Sie einen geeigneten Hintergrund hinzu, der vorzugsweise nicht zu viele Farben hat und nicht mit den Farben im **Target**-Sprite übereinstimmt. Hier habe ich den Hintergrund **Wall1** gewählt.

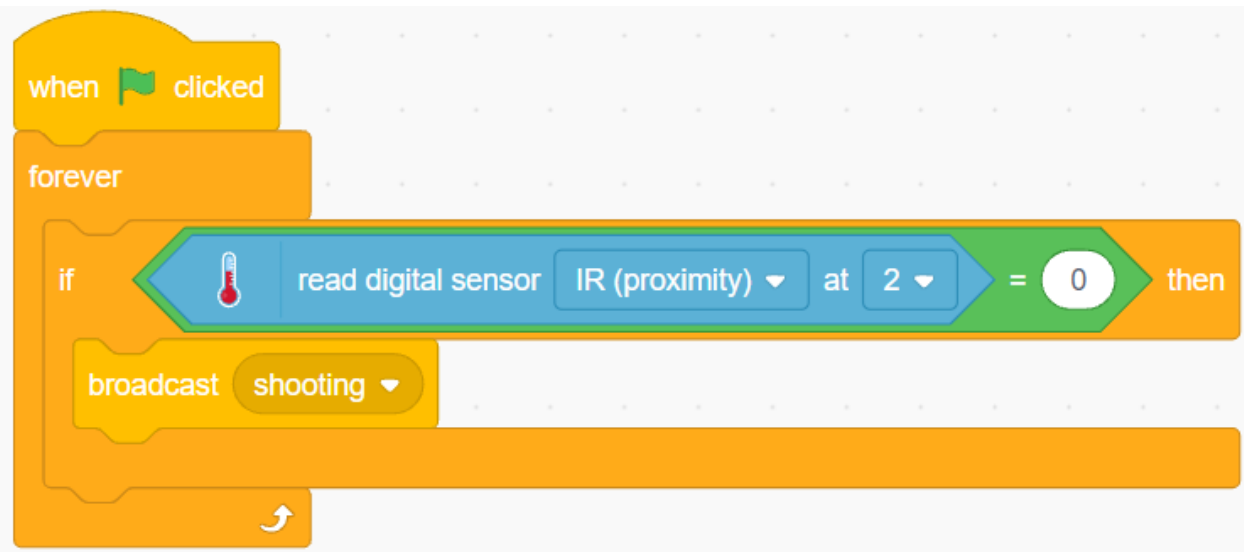


4. Script für das Fadenkreuz-Sprite

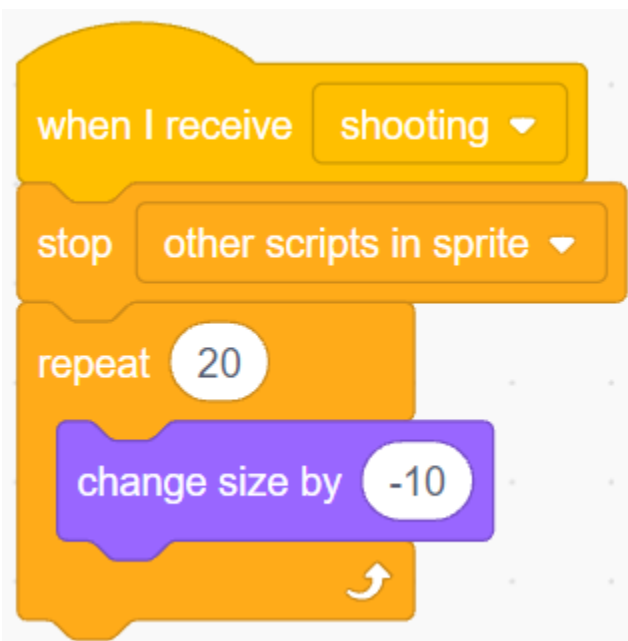
Legen Sie die zufällige Position und Größe des **Crosshair**-Sprites fest und lassen Sie es sich zufällig bewegen.



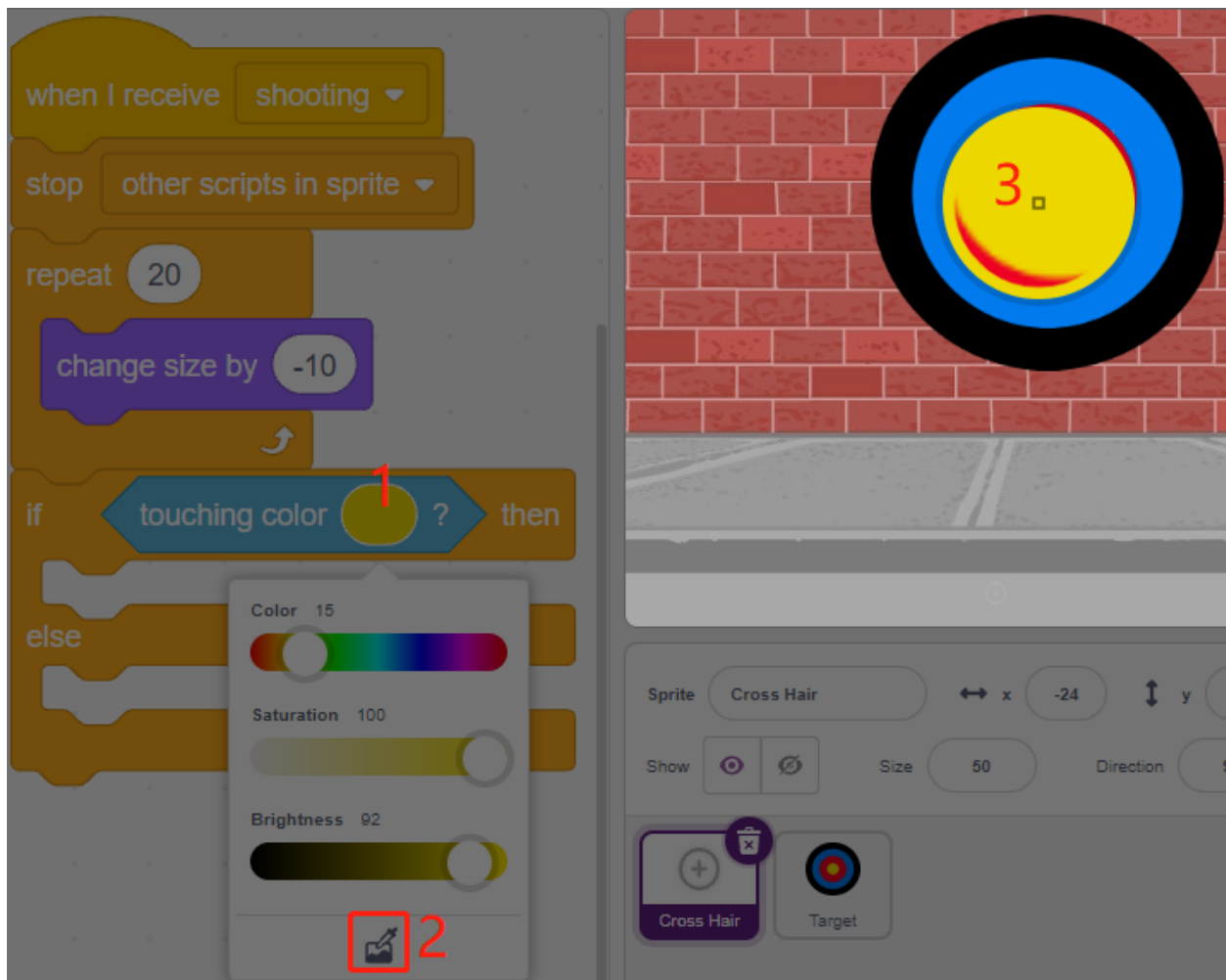
Wenn eine Hand vor das Hindernisvermeidungsmodul gelegt wird, gibt es ein niedriges Signal als Übertragungssignal aus.



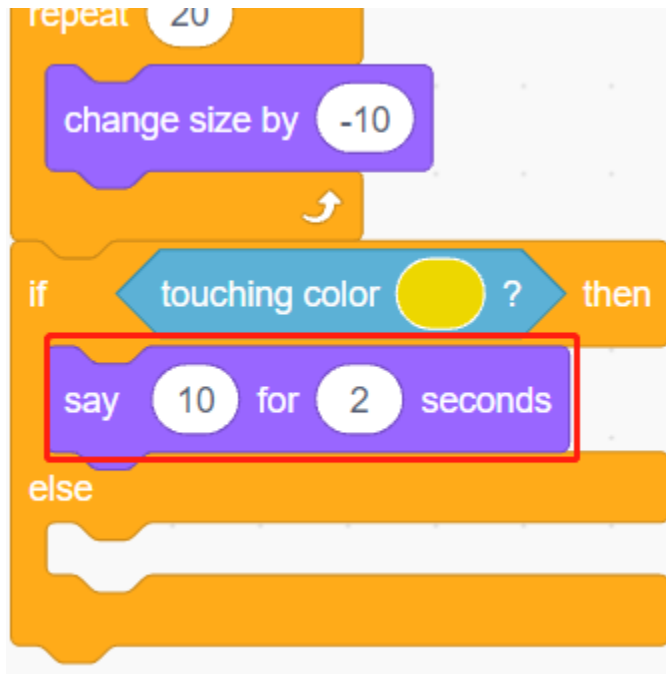
Wenn die **schießen**-Nachricht empfangen wird, stoppt das Sprite seine Bewegung und schrumpft langsam, wodurch der Effekt eines abgefeuerten Geschosses simuliert wird.



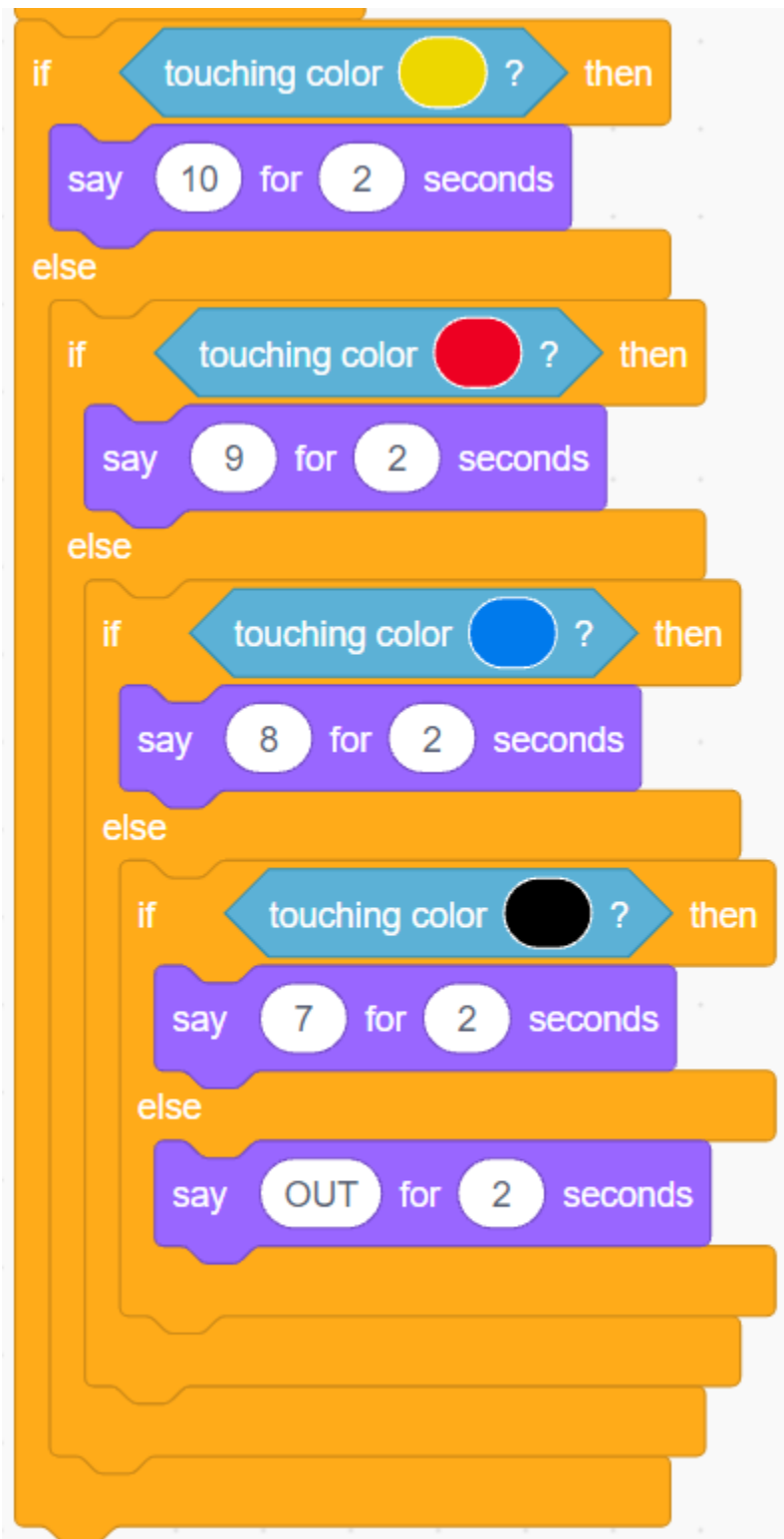
Verwenden Sie den [Touch color ()]-Block, um die Schussposition zu bestimmen.



Wenn der Schuss innerhalb des gelben Kreises liegt, werden 10 Punkte gemeldet.



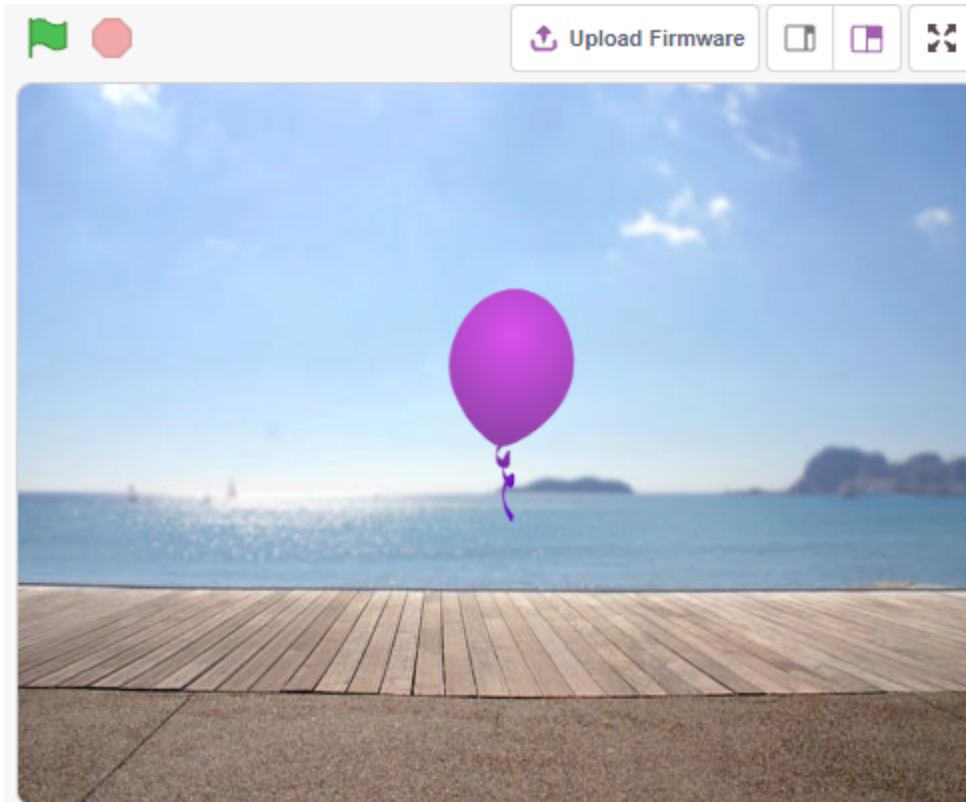
Verwenden Sie dieselbe Methode, um die Position des abgefeuerten Geschosses zu bestimmen. Wenn es nicht auf dem **Target**-Sprite eingestellt ist, bedeutet dies, dass es außerhalb des Kreises liegt.



8.17 2.14 SPIEL - Den Ballon aufblasen

Hier werden wir ein Ballonaufblas-Spiel spielen.

Nachdem Sie auf die grüne Flagge geklickt haben, wird der Ballon immer größer und größer. Wenn der Ballon zu groß wird, wird er platzen; wenn der Ballon zu klein ist, wird er herunterfallen; Sie müssen beurteilen, wann Sie den Knopf drücken, um ihn nach oben fliegen zu lassen.



8.17.1 Was Sie lernen werden

- Ein Kostüm für den Sprite malen

8.17.2 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

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

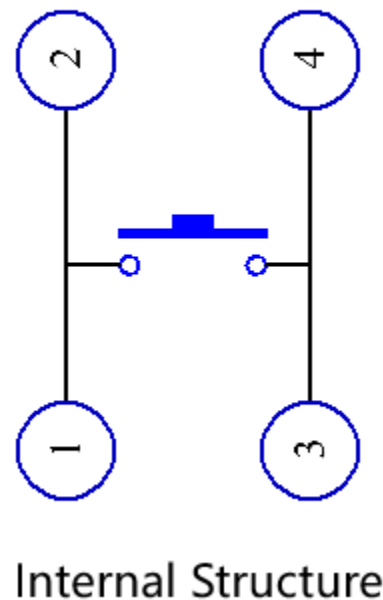
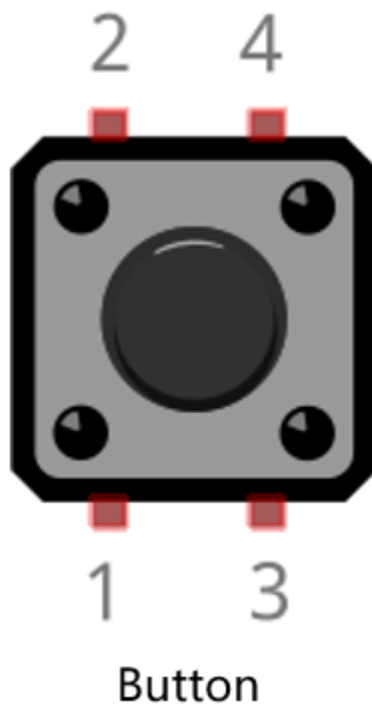
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können diese auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Kondensator</i>	
<i>Knopf</i>	

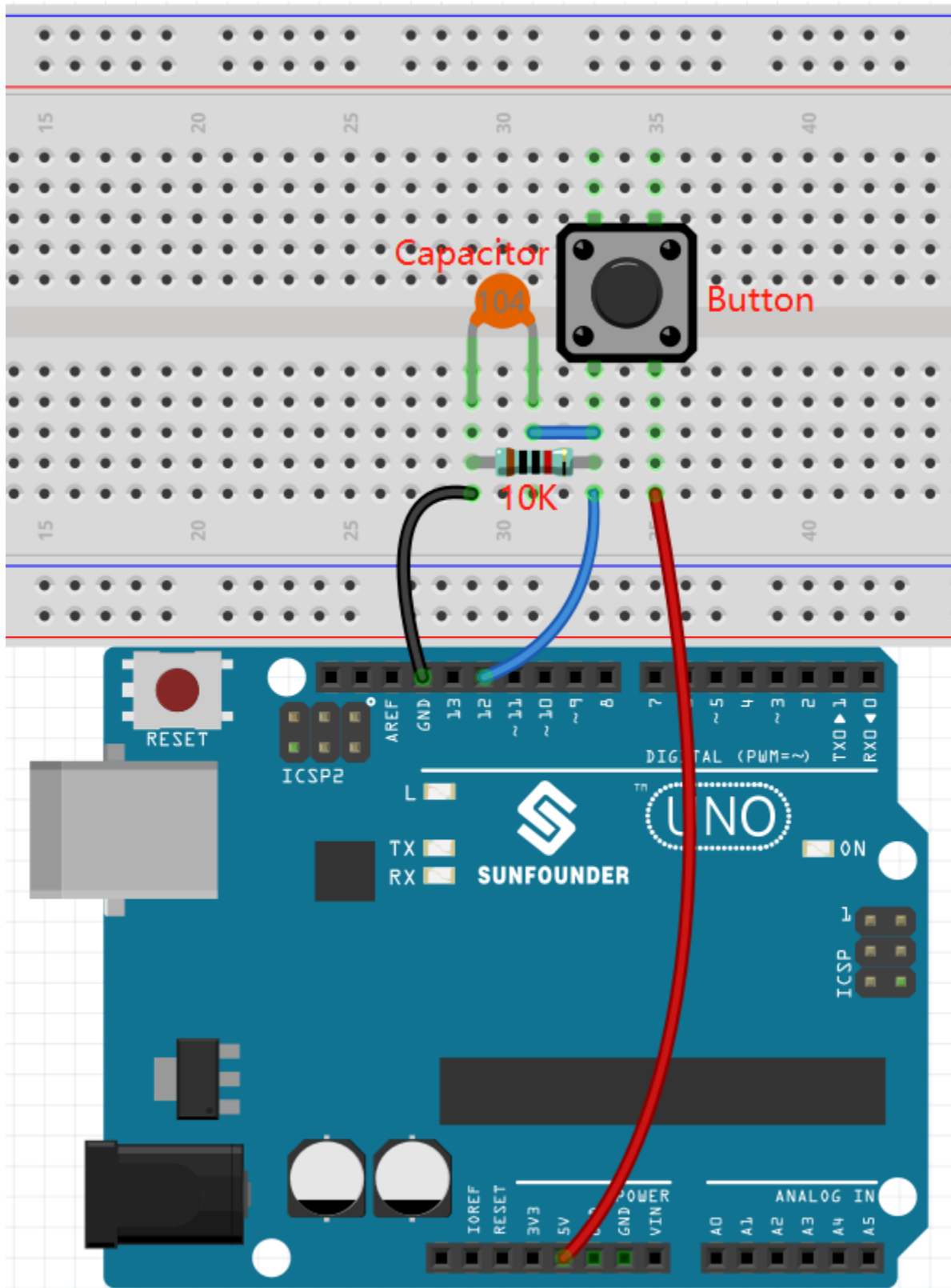
8.17.3 Schaltung aufbauen

Der Knopf ist ein 4-Pin-Gerät, da Pin 1 mit Pin 2 und Pin 3 mit Pin 4 verbunden ist. Wenn der Knopf gedrückt wird, sind alle 4 Pins verbunden und schließen so den Stromkreis.



Bauen Sie den Schaltkreis gemäß dem folgenden Diagramm.

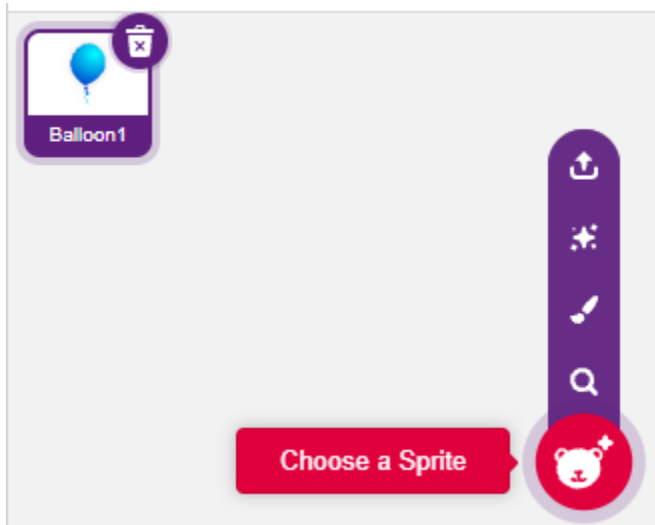
- Verbinden Sie einen der Pins auf der linken Seite des Knopfes mit Pin 12, der mit einem Pull-down-Widerstand und einem 0,1uF (104) Kondensator verbunden ist (um Wackeln zu eliminieren und ein stabiles Level auszugeben, wenn der Knopf arbeitet).
- Verbinden Sie das andere Ende des Widerstands und des Kondensators mit GND und einen der Pins auf der rechten Seite des Knopfes mit 5V.



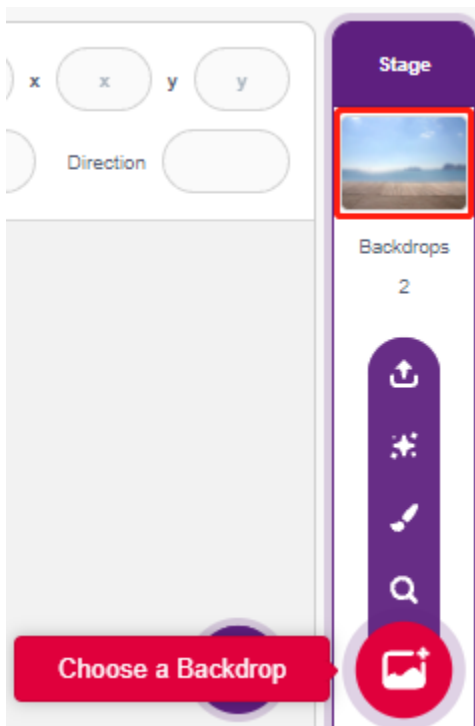
8.17.4 Programmierung

1. Einen Sprite und einen Hintergrund hinzufügen

Löschen Sie den Standard-Sprite, klicken Sie auf die Schaltfläche **Choose a Sprite** in der unteren rechten Ecke des Sprite-Bereichs und wählen Sie dann den **Balloon1** Sprite aus.



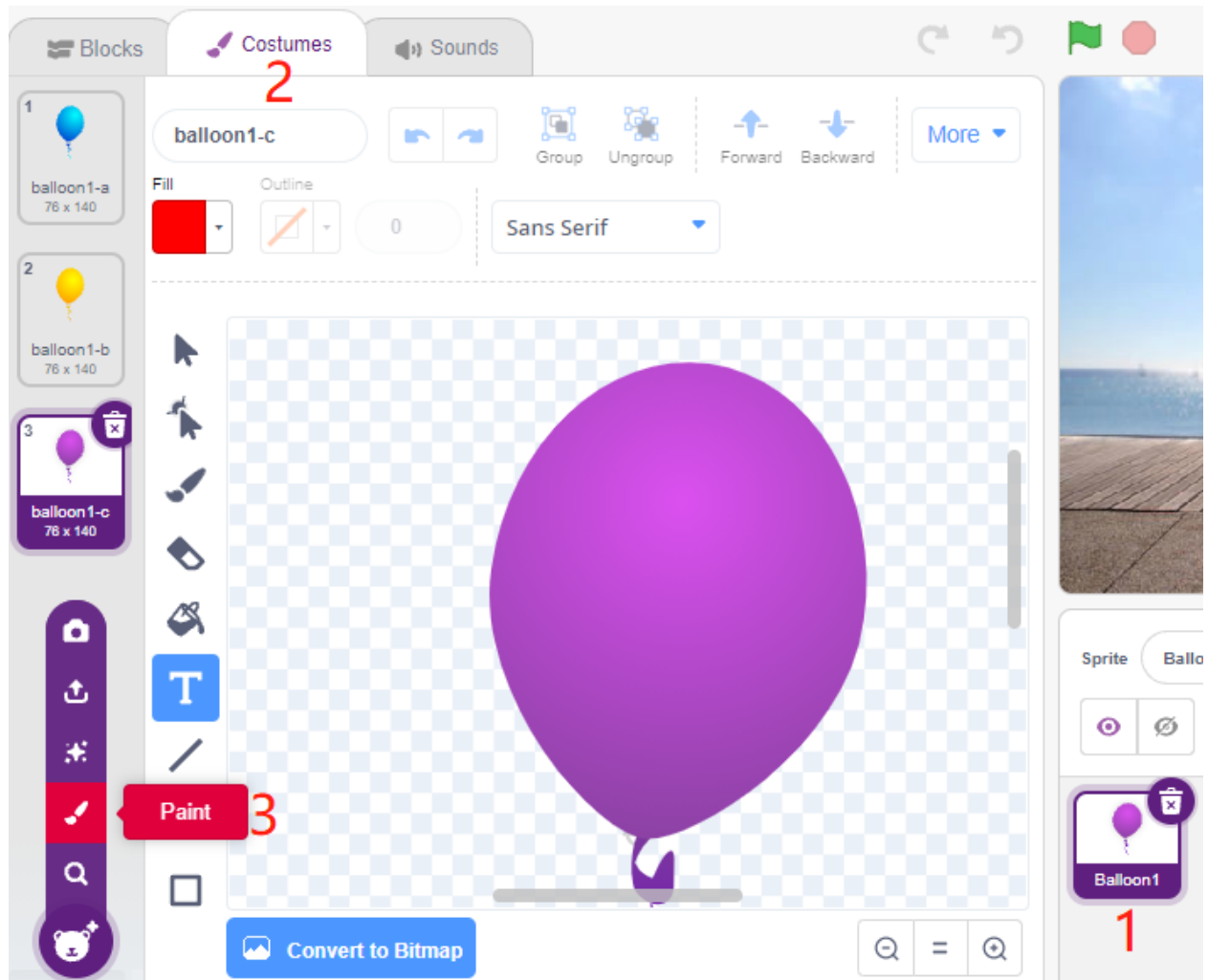
Fügen Sie einen **Boardwalk** Hintergrund über die Schaltfläche **Choose a backdrop** hinzu, oder andere Hintergründe, die Ihnen gefallen.



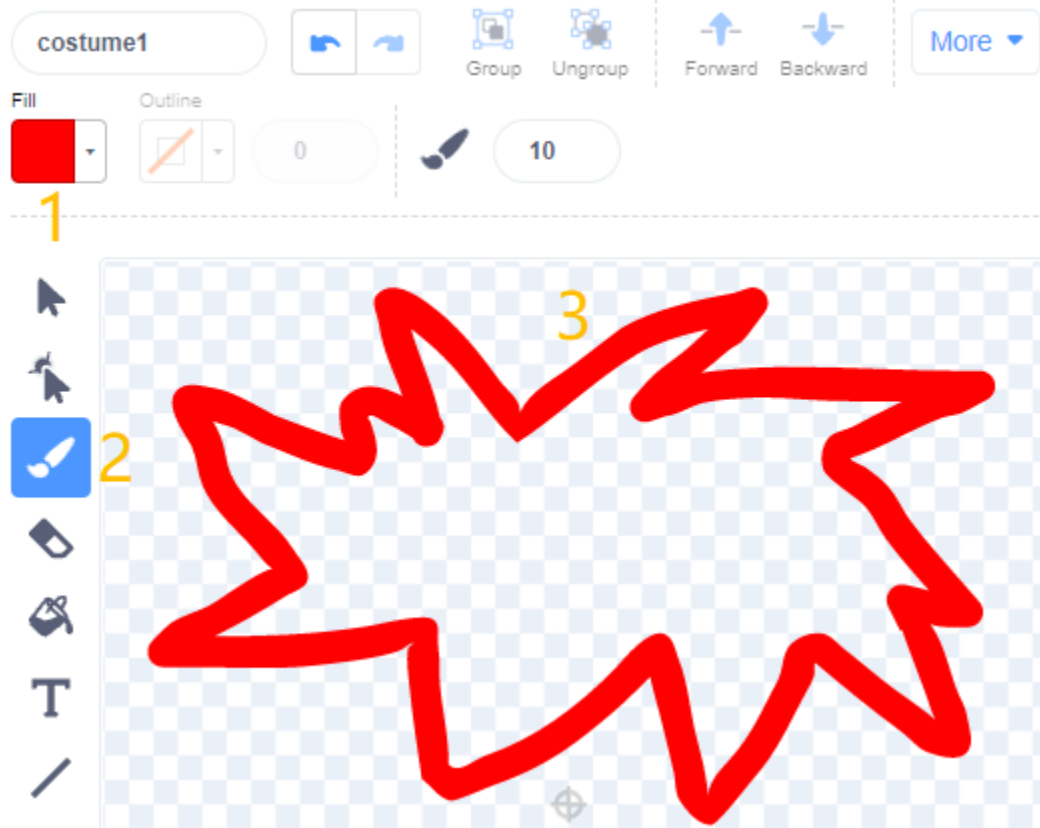
2. Ein Kostüm für den Balloon1 Sprite malen

Jetzt zeichnen wir ein Explosions-Effekt-Kostüm für den Ballon-Sprite.

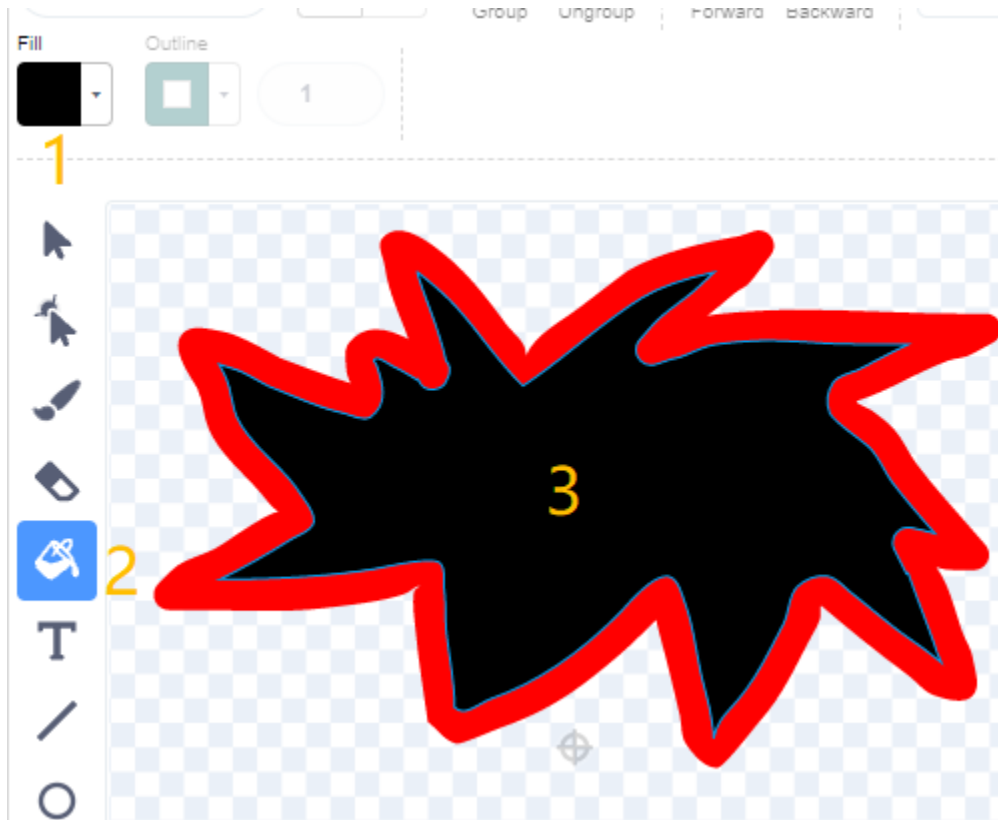
Gehen Sie zur **Costumes** Seite für den **Balloon1** Sprite, klicken Sie auf die Schaltfläche **Choose a Costume** in der unteren linken Ecke und wählen Sie **Paint**, um ein leeres **Costume** aufzurufen.



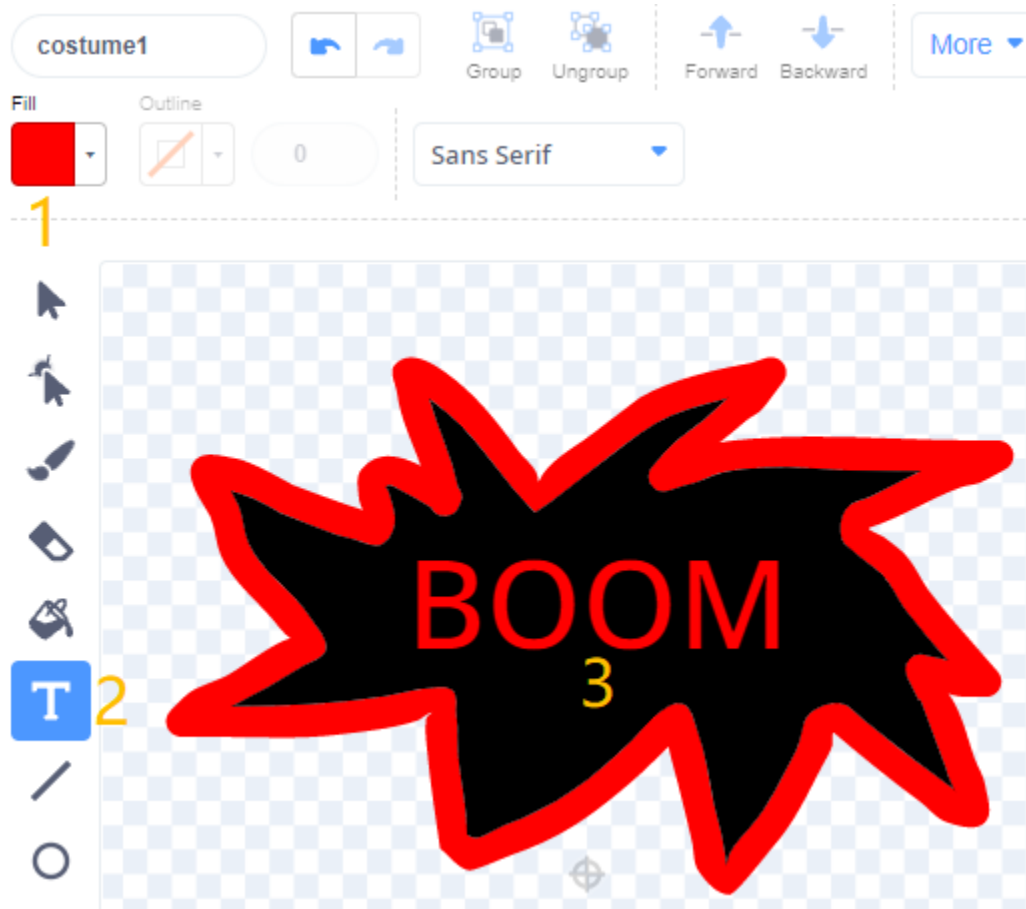
Wählen Sie eine Farbe und verwenden Sie dann das **Brush**-Werkzeug, um ein Muster zu zeichnen.



Wählen Sie erneut eine Farbe, klicken Sie auf das Füllwerkzeug und bewegen Sie die Maus innerhalb des Musters, um es mit einer Farbe zu füllen.

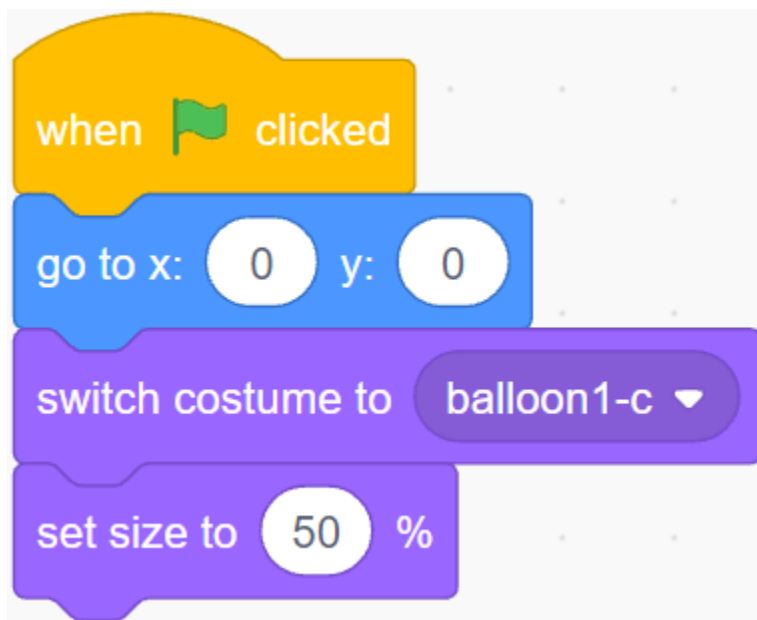


Schreiben Sie abschließend den Text BOOM, sodass ein Explosions-Effekt-Kostüm fertig ist.

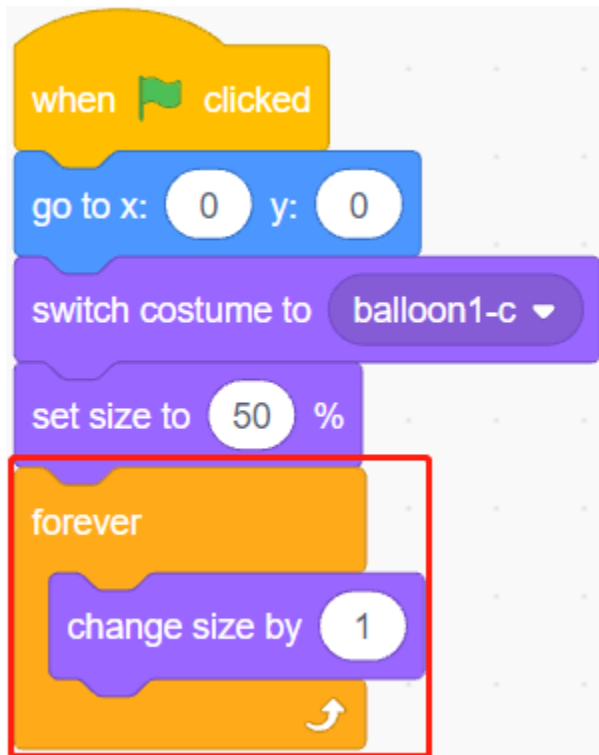


3. Den Balloon Sprite programmieren

Stellen Sie die Anfangsposition und Größe des **Balloon1** Sprites ein.

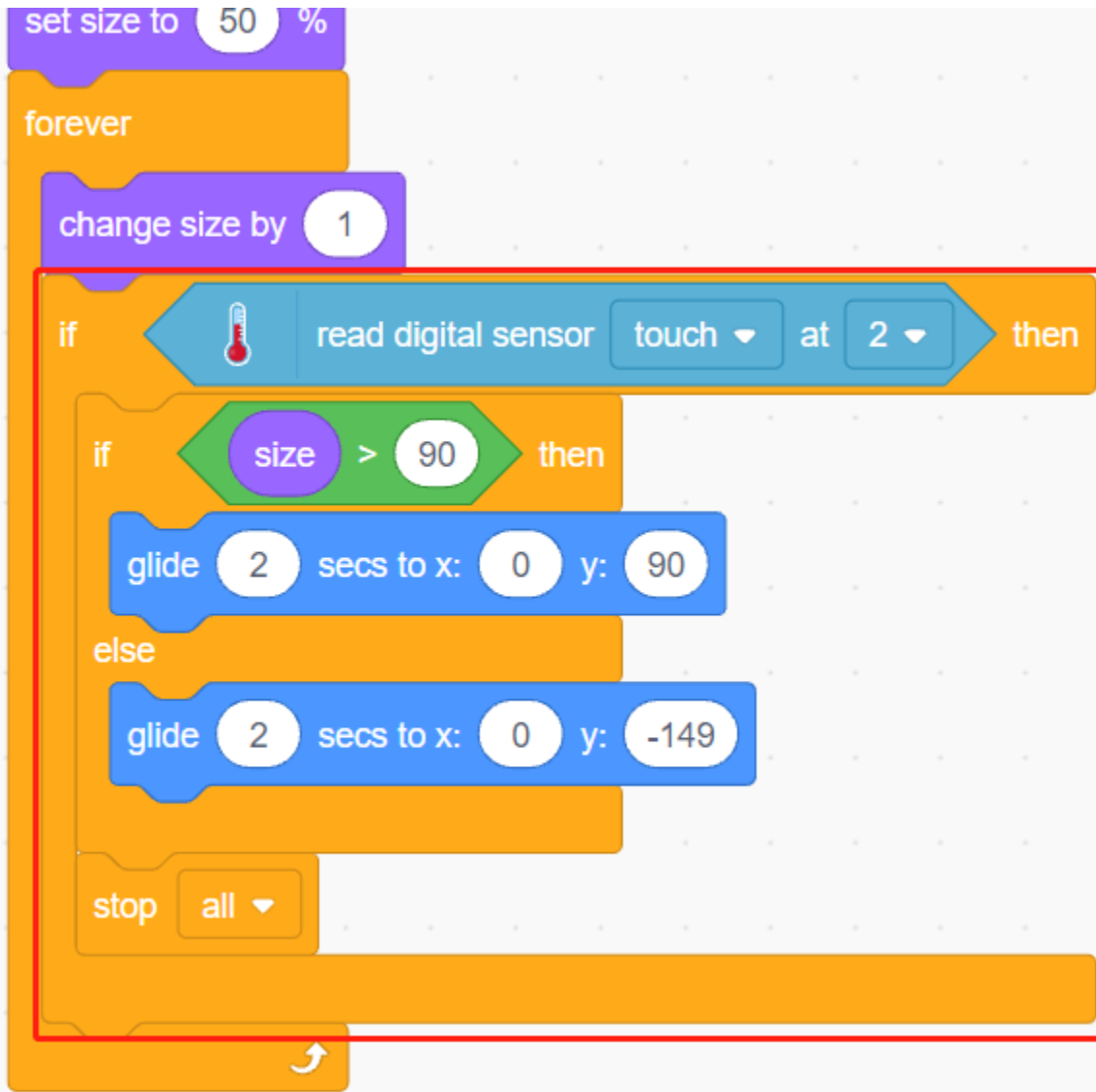


Lassen Sie dann den **Balloon** Sprite langsam größer werden.

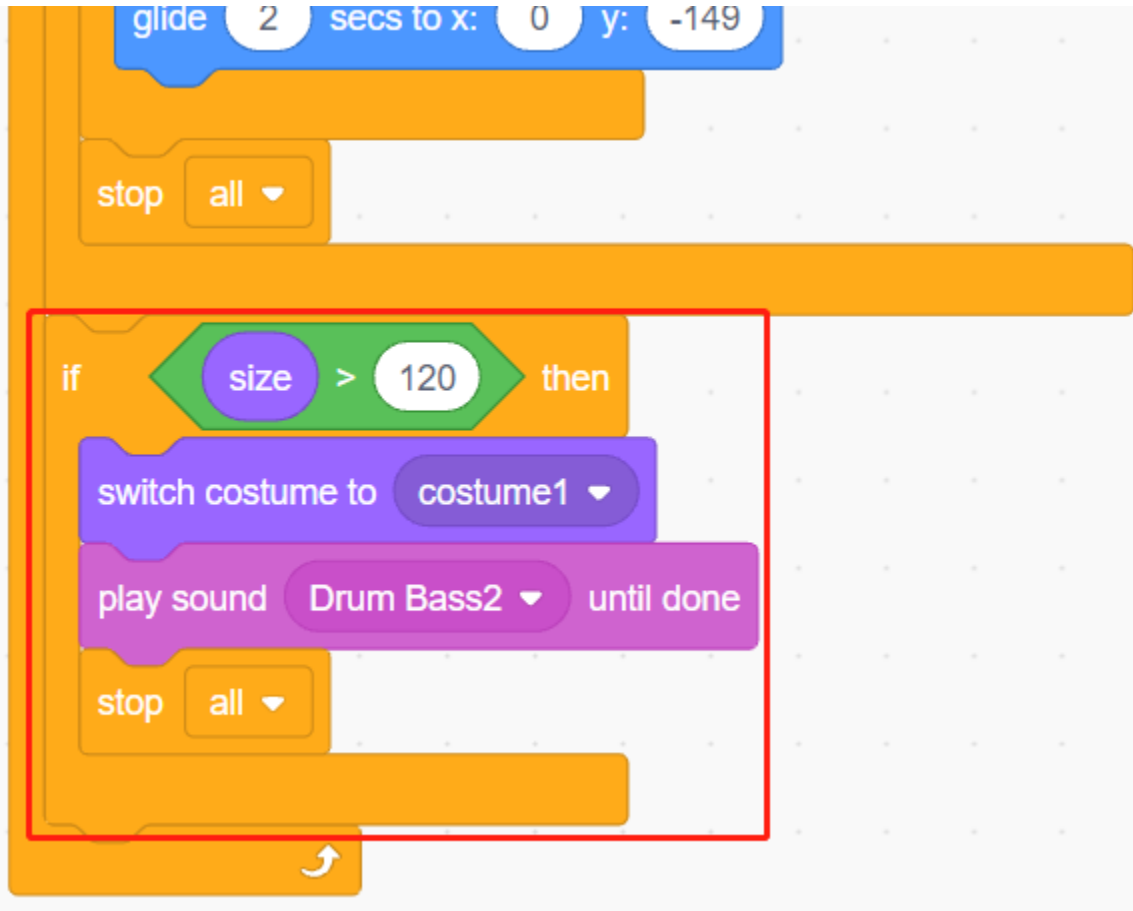


Wenn der Knopf gedrückt wird (Wert ist 1), hört der **Balloon1** Sprite auf, größer zu werden.

- Wenn die Größe kleiner als 90 ist, wird er fallen (y-Koordinate verringert sich).
- Wenn die Größe größer als 90 und kleiner als 120 ist, wird er zum Himmel fliegen (y-Koordinate erhöht sich).



Wenn der Knopf nicht gedrückt wurde, wird der Ballon langsam größer und wenn die Größe größer als 120 ist, wird er explodieren (wechseln Sie zum Explosions-Effekt-Kostüm).

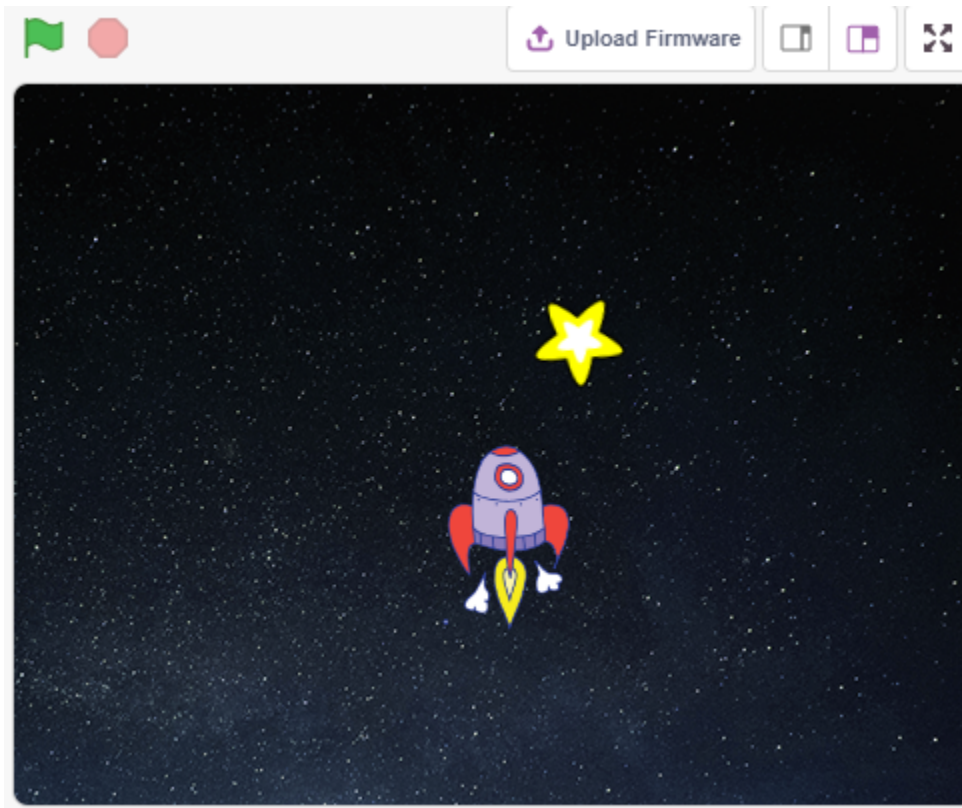


8.18 2.15 SPIEL - Sternengekreuzt

In den nächsten Projekten werden wir einige spaßige Minispiele in PictoBlox spielen.

Hier verwenden wir das Joystick-Modul, um das Spiel „Sternengekreuzt“ zu spielen.

Nachdem das Skript gestartet wurde, erscheinen zufällig Sterne auf der Bühne. Mit dem Joystick steuern Sie das Raumschiff, um den Sternen auszuweichen. Wenn Sie sie berühren, endet das Spiel.



8.18.1 Was Sie lernen werden

- Wie das Joystick-Modul funktioniert
- Das Festlegen der x- und y-Koordinaten des Sprites

8.18.2 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>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>Joystick Modul</i>	-

8.18.3 Schaltung aufbauen

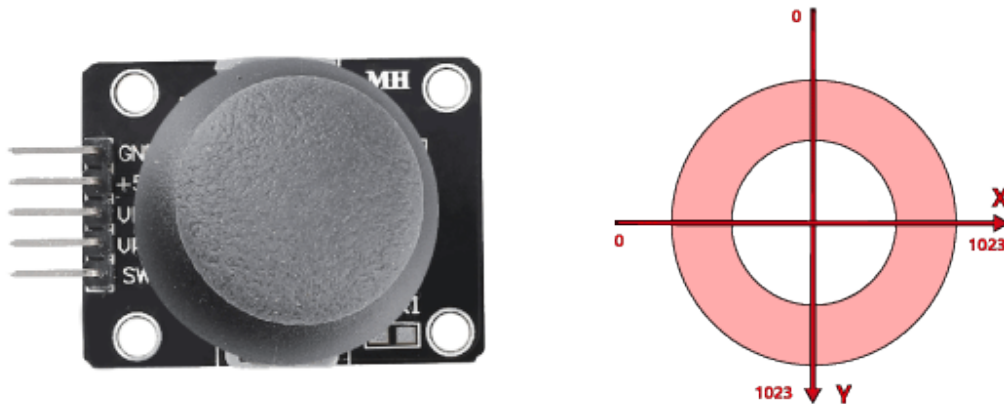
Ein Joystick ist ein Eingabegerät, das aus einem Stick besteht, der auf einer Basis schwenkt und seinen Winkel oder seine Richtung an das zu steuernde Gerät meldet. Joysticks werden oft verwendet, um Videospiele und Roboter zu steuern.

Um dem Computer eine vollständige Bewegungsfreiheit zu melden, muss ein Joystick die Stickposition in zwei Achsen messen – der X-Achse (links nach rechts) und der Y-Achse (oben nach unten).

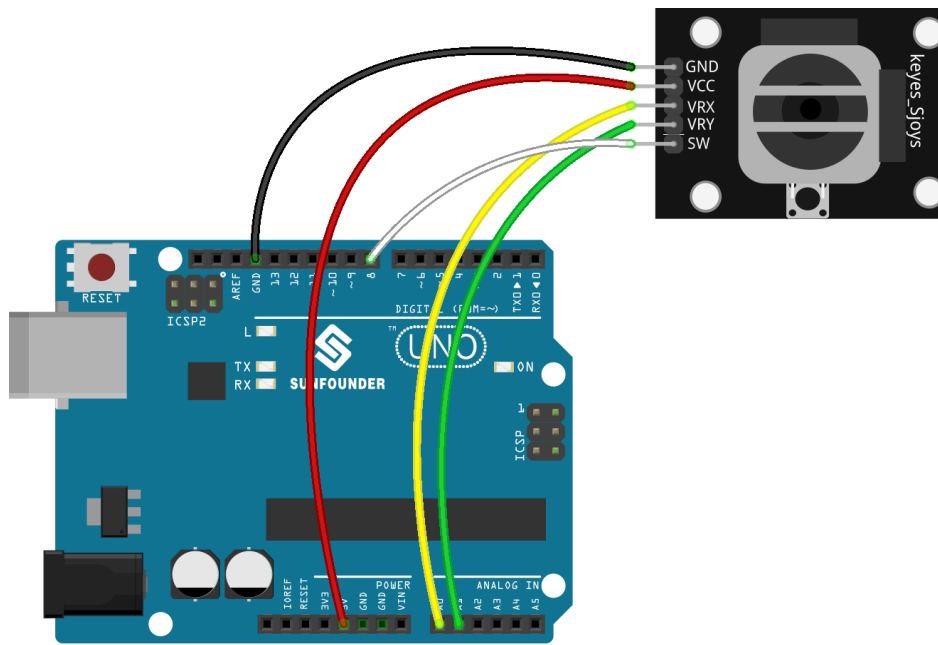
Die Bewegungskordinaten des Joysticks sind in der folgenden Abbildung dargestellt.

Bemerkung:

- Die x-Koordinate geht von links nach rechts, der Bereich liegt zwischen 0-1023.
 - Die y-Koordinate geht von oben nach unten, Bereich 0-1023.
-



Bauen Sie jetzt die Schaltung gemäß dem folgenden Diagramm.

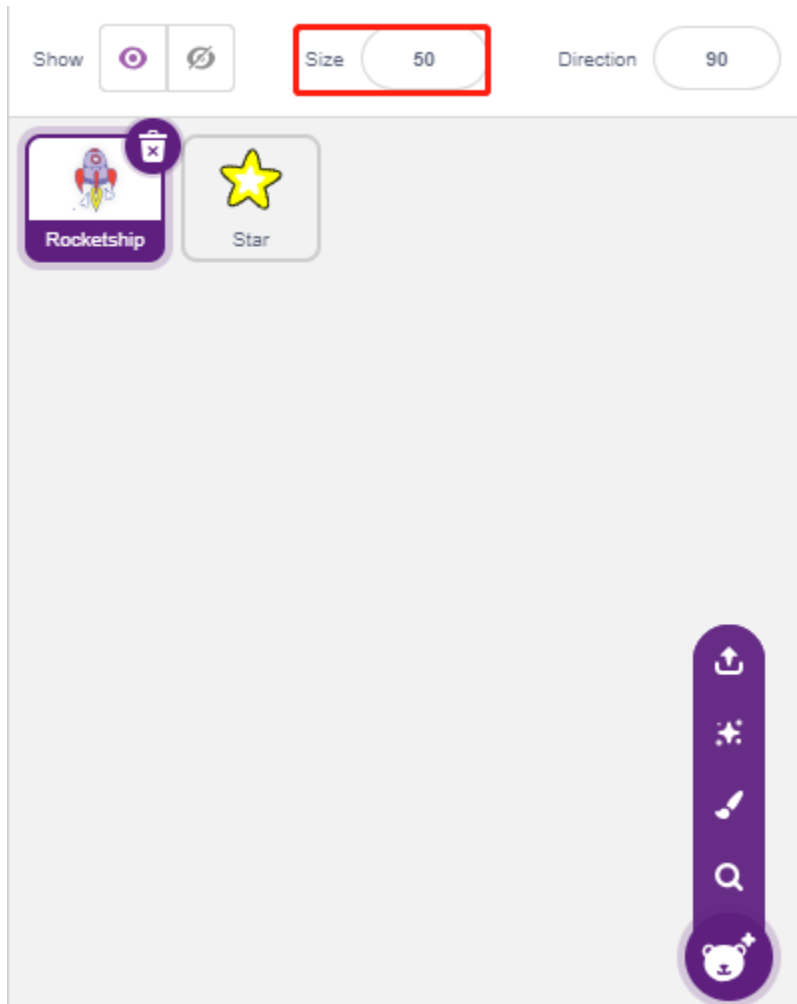


8.18.4 Programmierung

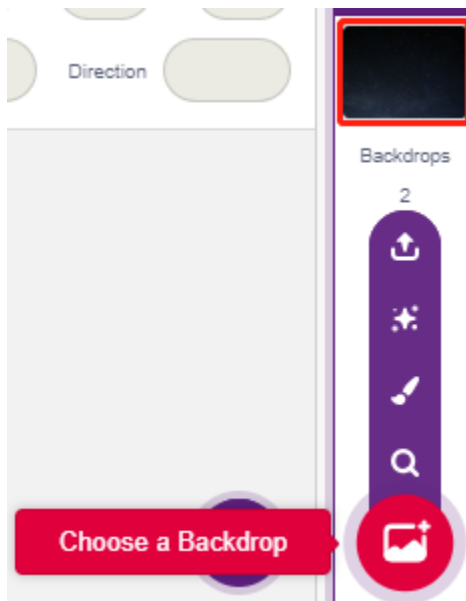
Das gesamte Skript soll den Effekt erzielen, dass bei Klick auf die grüne Flagge der **Stars**-Sprite in einer Kurve auf der Bühne bewegt wird und Sie den Joystick verwenden müssen, um das **Rocketship** zu bewegen, damit es nicht vom **Stars**-Sprite berührt wird.

1. Sprites und Hintergründe hinzufügen

Löschen Sie den Standard-Sprite und verwenden Sie die Schaltfläche **Choose a Sprite** zum Hinzufügen des **Rocketship**-Sprites und des **Star**-Sprites. Beachten Sie, dass die Größe des **Rocket**-Sprites auf 50% eingestellt ist.



Fügen Sie jetzt den **Stars**-Hintergrund durch **Choose a Backdrop** hinzu.

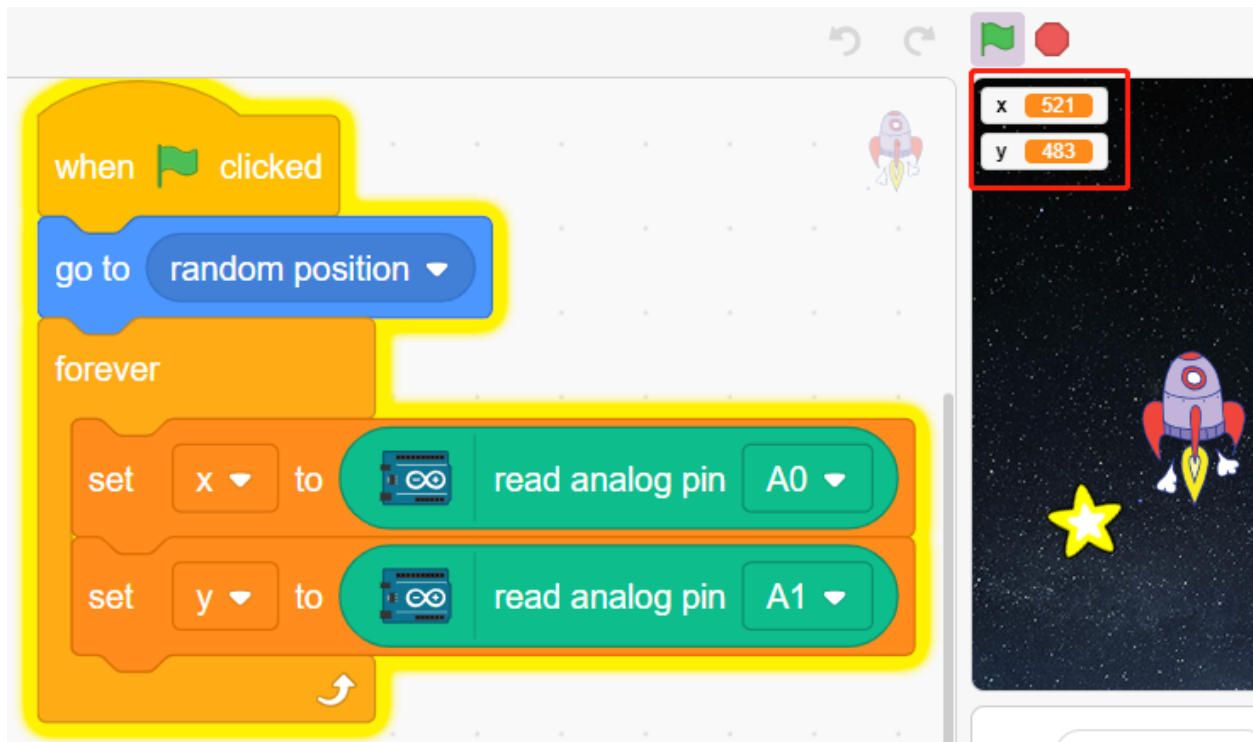


2. Skripting für Raumschiff

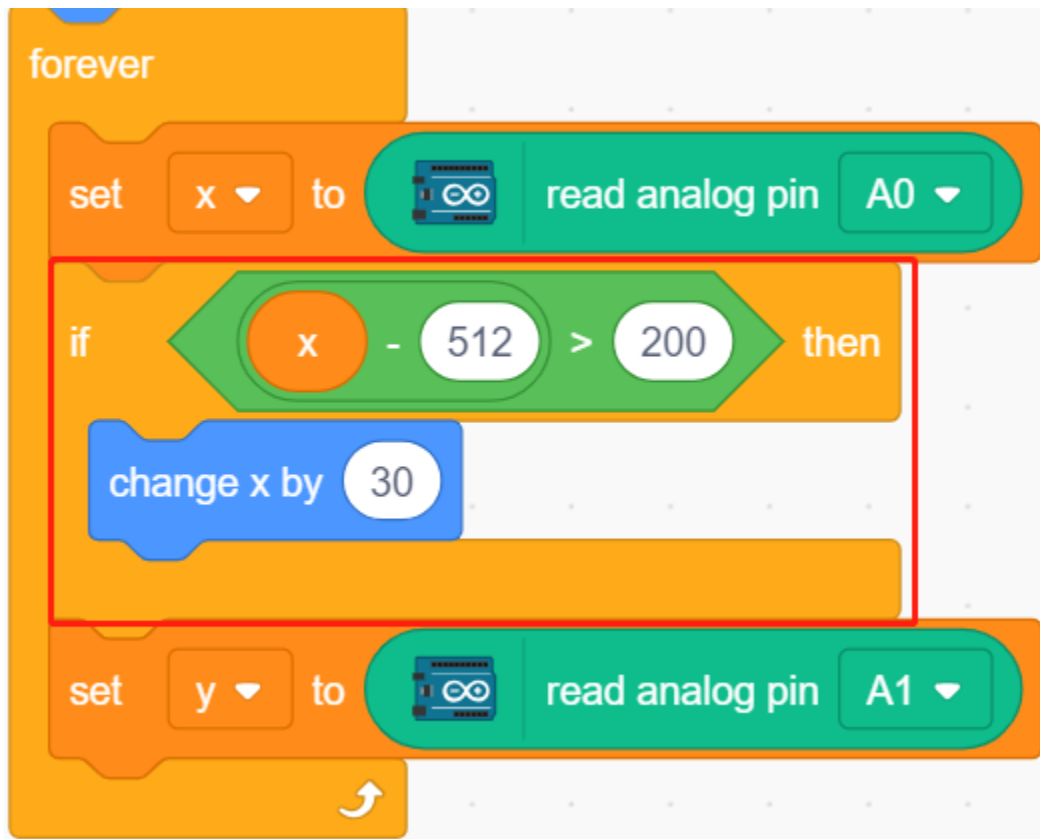
Der **Rocketship**-Sprite soll den Effekt erzielen, dass er an einer zufälligen Position erscheint und dann durch den Joystick gesteuert wird, um ihn nach oben, unten, links und rechts zu bewegen.

Der Ablauf ist wie folgt.

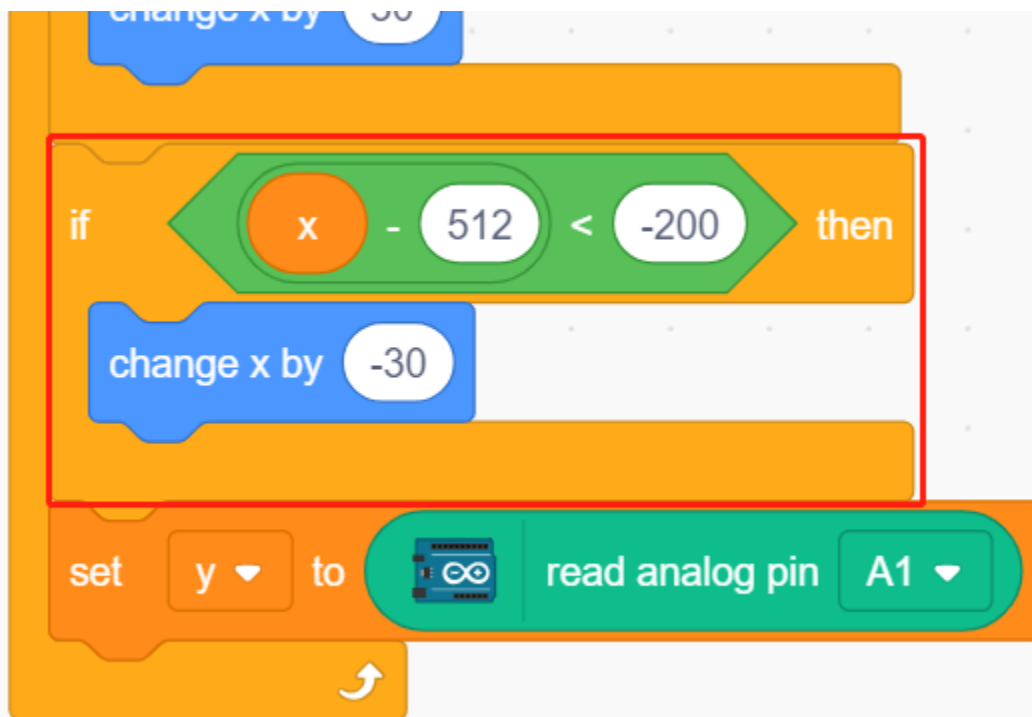
- Wenn auf die grüne Flagge geklickt wird, lassen Sie den Sprite zu einer zufälligen Position gehen und erstellen Sie 2 Variablen **x** und **y**, die die Werte lesen, die von A0 (VRX des Joysticks) bzw. A1 (VRY des Joysticks) kommen. Sie können das Skript ausführen und den Joystick hoch und runter, links und rechts kippen, um den Wertebereich für x und y zu sehen.



- Der Wert von A0 liegt im Bereich 0-1023 (die Mitte liegt bei etwa 512). Verwenden Sie $x - 512 > 200$, um festzustellen, ob der Joystick nach rechts gekippt wird, und wenn ja, lassen Sie die x-Koordinate des Sprites +30 sein (um den Sprite nach rechts zu bewegen).

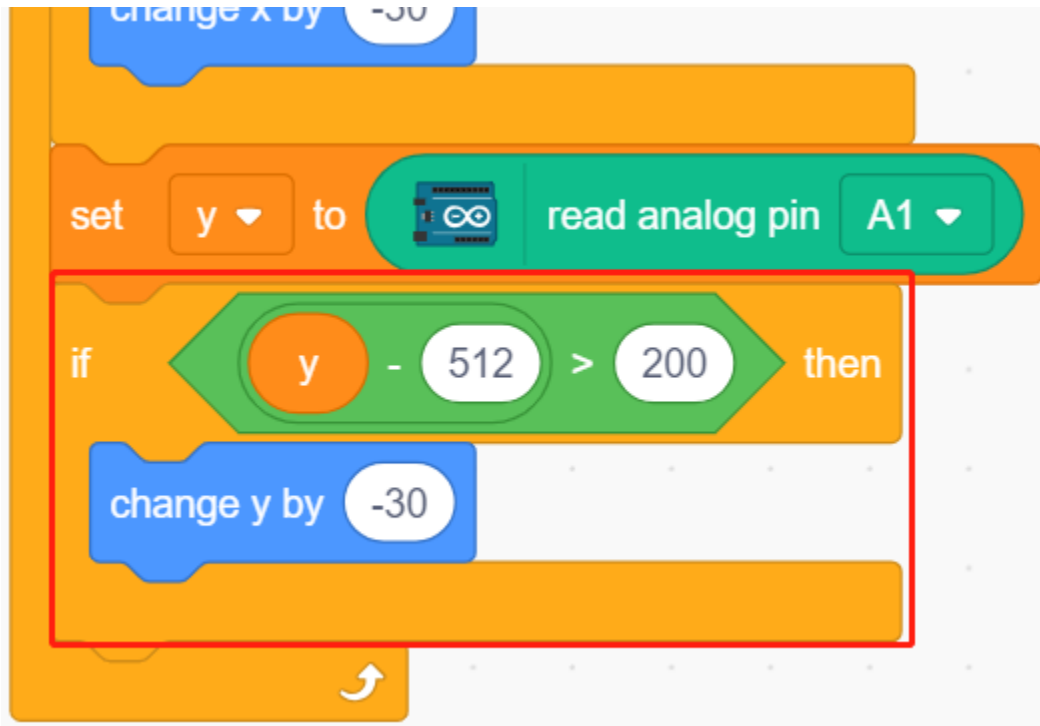


- Wenn der Joystick nach links gekippt wird ($x - 512 < -200$), lassen Sie die x-Koordinate des Sprites -30 sein (um den Sprite nach links zu bewegen).

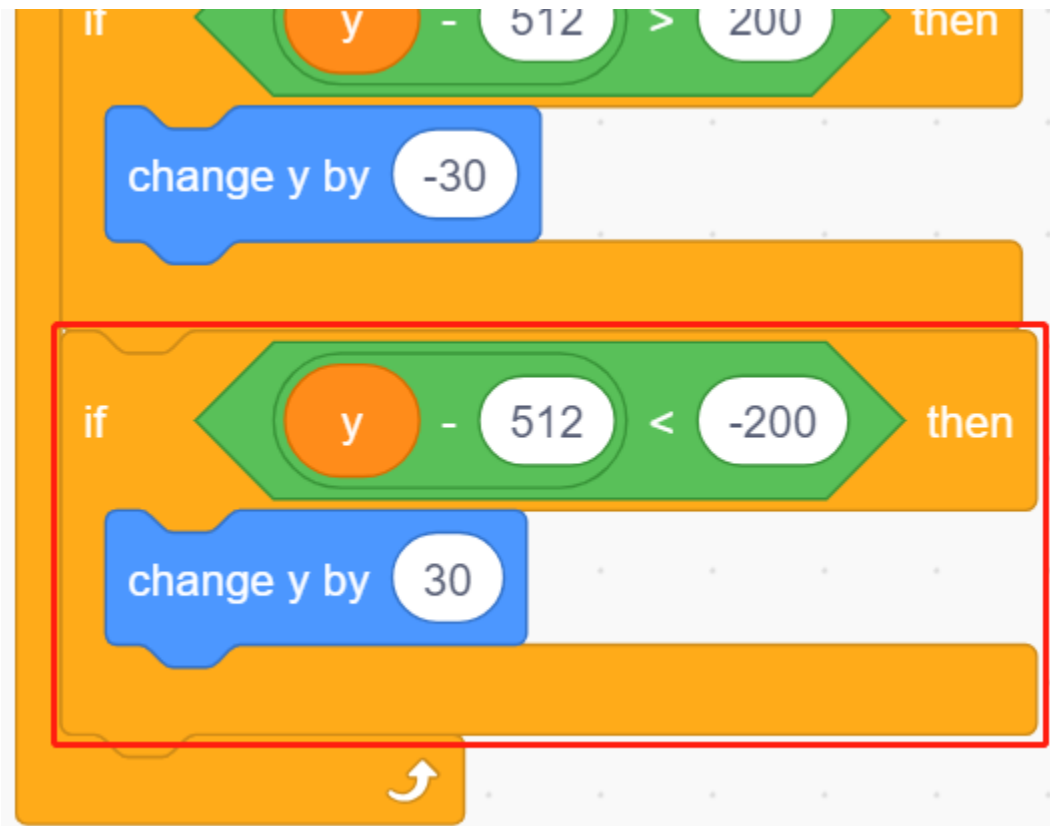


- Da die y-Koordinate des Joysticks von oben (0) nach unten (1023) geht und die y-Koordinate des Sprites von

unten nach oben geht, muss die y-Koordinate im Skript -30 sein, um den Joystick nach oben und den Sprite nach oben zu bewegen.



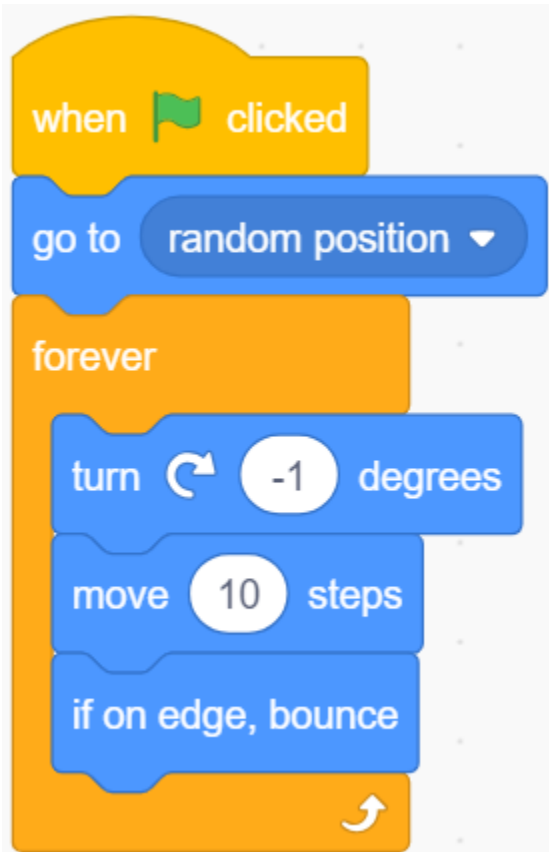
- Wenn der Joystick nach unten gekippt wird, ist die y-Koordinate des Sprites +30.



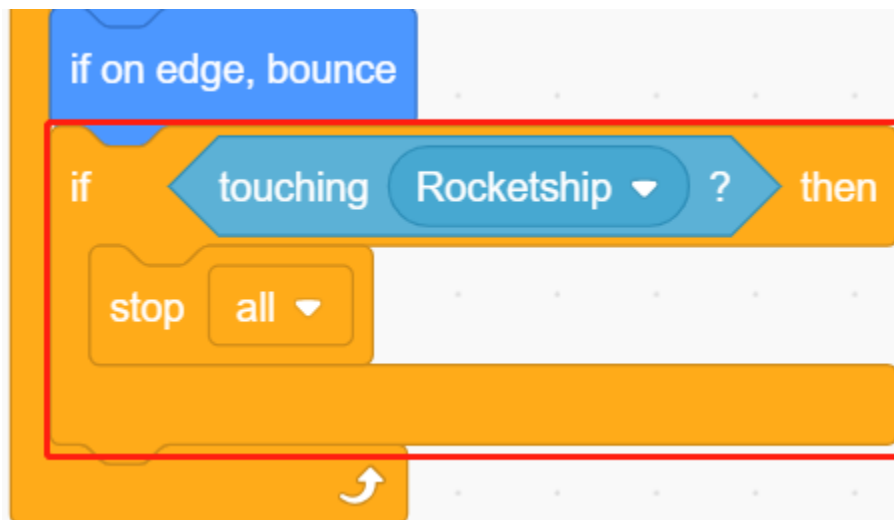
3. Skripting für Stern

Der Effekt, den der **Star**-Sprite erzielen soll, ist, dass er an einer zufälligen Stelle erscheint, und wenn er das **Rocketship** trifft, stoppt das Skript und das Spiel endet.

- Wenn auf die grüne Flagge geklickt wird und der Sprite zu einer zufälligen Position geht, dient der Block [turn degrees] dazu, den **Star**-Sprite mit einer leichten Winkeländerung vorwärts zu bewegen, sodass Sie sehen können, dass er sich in einer Kurve bewegt, und wenn er am Rand ist, prallt er ab.



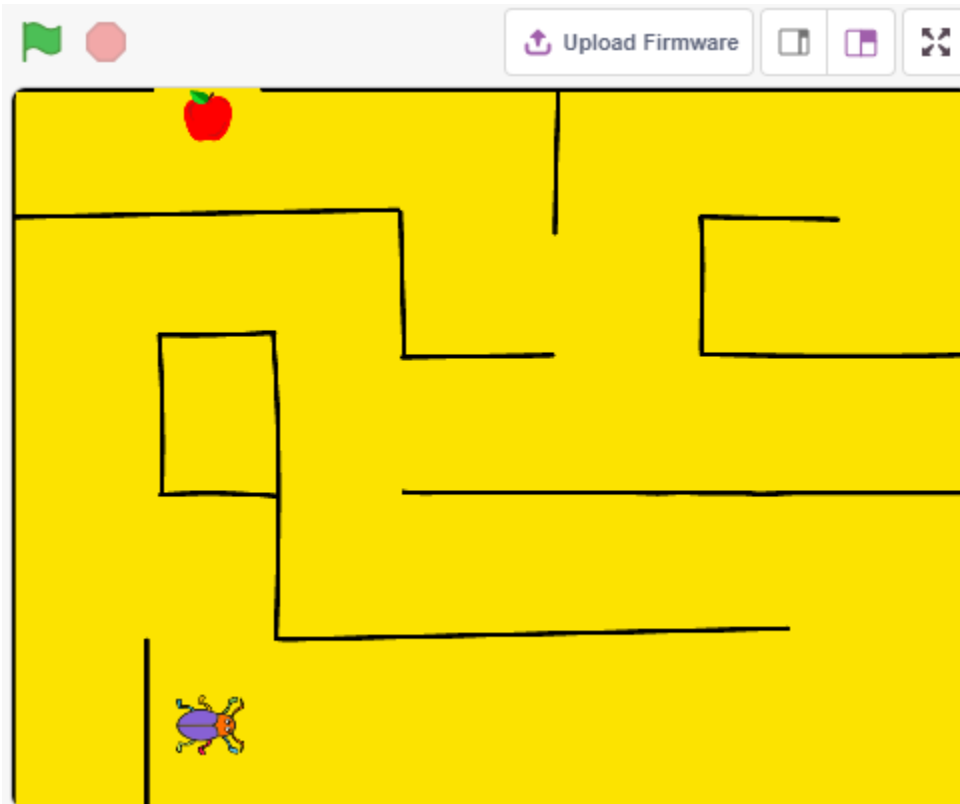
- Wenn der Sprite während seiner Bewegung den **Rocketship**-Sprite berührt, stoppen Sie das Skript.



8.19 2.16 SPIEL - Apfel essen

In diesem Projekt spielen wir ein Spiel, bei dem ein Knopf verwendet wird, um einen Käfer zu steuern, der Äpfel isst.

Wenn die grüne Flagge angeklickt wird, drücken Sie den Knopf und der Käfer dreht sich. Drücken Sie den Knopf erneut und der Käfer hält an und bewegt sich in diesem Winkel vorwärts. Sie müssen den Winkel des Käfers so steuern, dass er sich vorwärts bewegt, ohne die schwarze Linie auf der Karte zu berühren, bis er den Apfel isst. Wenn er die schwarze Linie berührt, ist das Spiel vorbei.



8.19.1 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

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

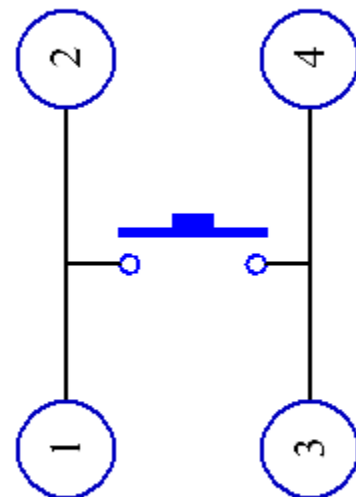
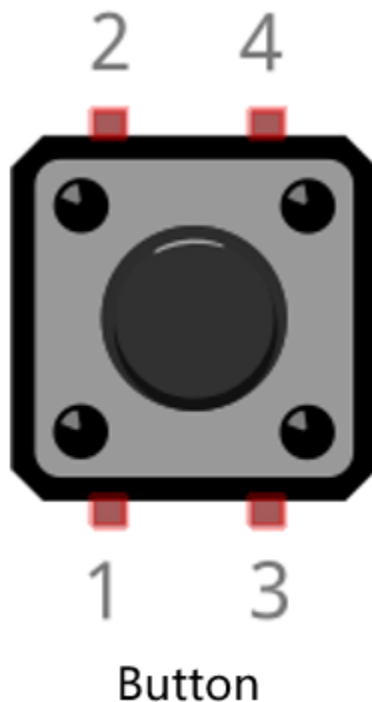
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Kondensator</i>	
<i>Knopf</i>	

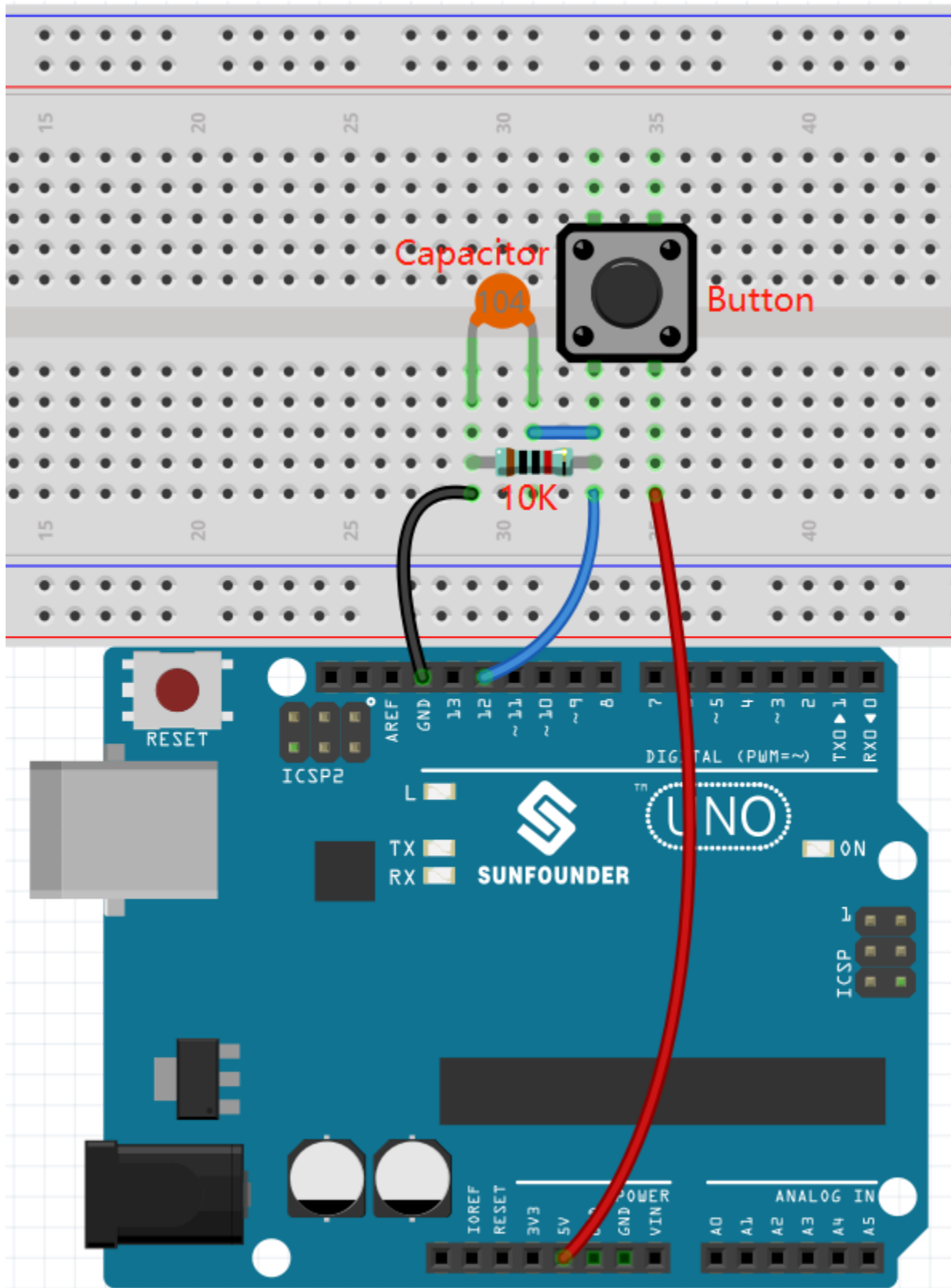
8.19.2 Schaltung aufbauen

Der Knopf ist ein Gerät mit 4 Pins. Da Pin 1 mit Pin 2 und Pin 3 mit Pin 4 verbunden ist, werden beim Drücken des Knopfs die 4 Pins verbunden und der Stromkreis geschlossen.



Bauen Sie den Schaltkreis gemäß dem folgenden Diagramm.

- Verbinden Sie einen der Pins auf der linken Seite des Knopfs mit Pin 12, der mit einem Pull-down-Widerstand und einem 0,1uF (104) Kondensator verbunden ist (um Flackern zu vermeiden und ein stabiles Signal auszugeben, wenn der Knopf betätigt wird).
- Verbinden Sie das andere Ende des Widerstands und des Kondensators mit GND und einen der Pins auf der rechten Seite des Knopfs mit 5V.



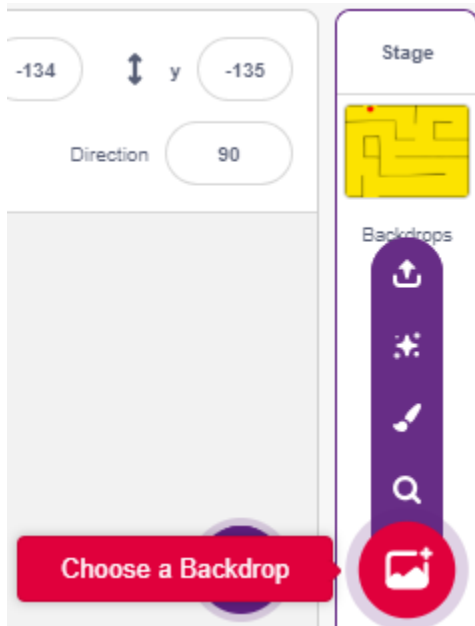
8.19.3 Programmierung

Der Effekt, den wir erreichen möchten, besteht darin, mit der Schaltfläche die Richtung des **Beetle**-Sprites zu steuern, um sich vorwärts zu bewegen und den Apfel zu essen, ohne die schwarze Linie auf dem **Maze**-Hintergrund zu berühren, wodurch der Hintergrund gewechselt wird gegessen.

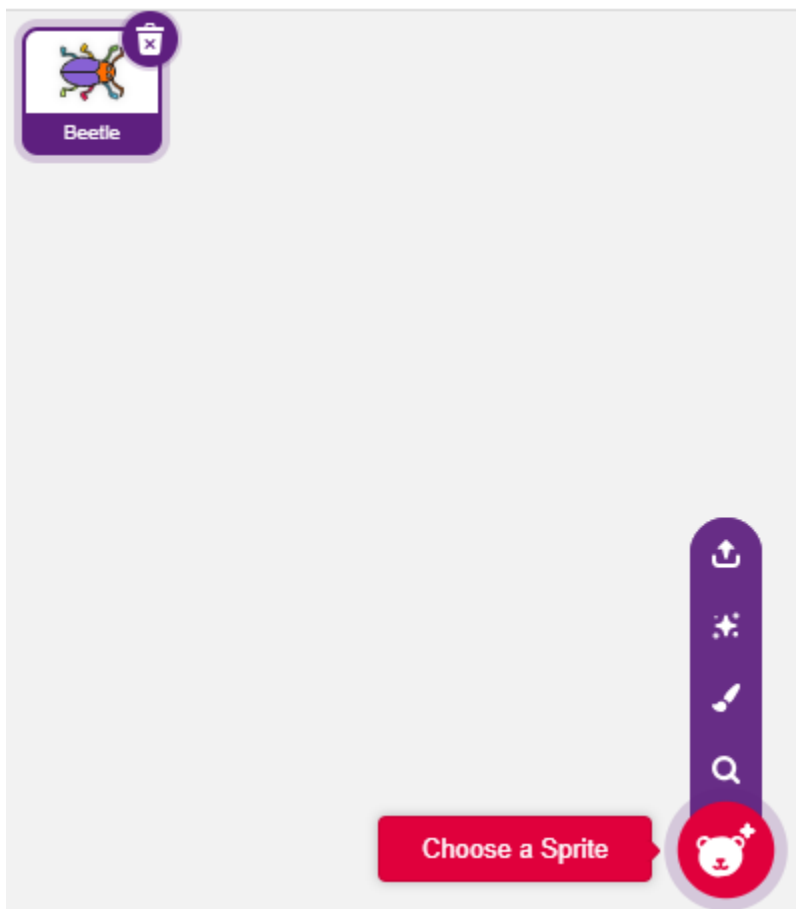
Fügen Sie nun die entsprechenden Hintergründe und Sprites hinzu.

1. Hintergründe und Sprites hinzufügen

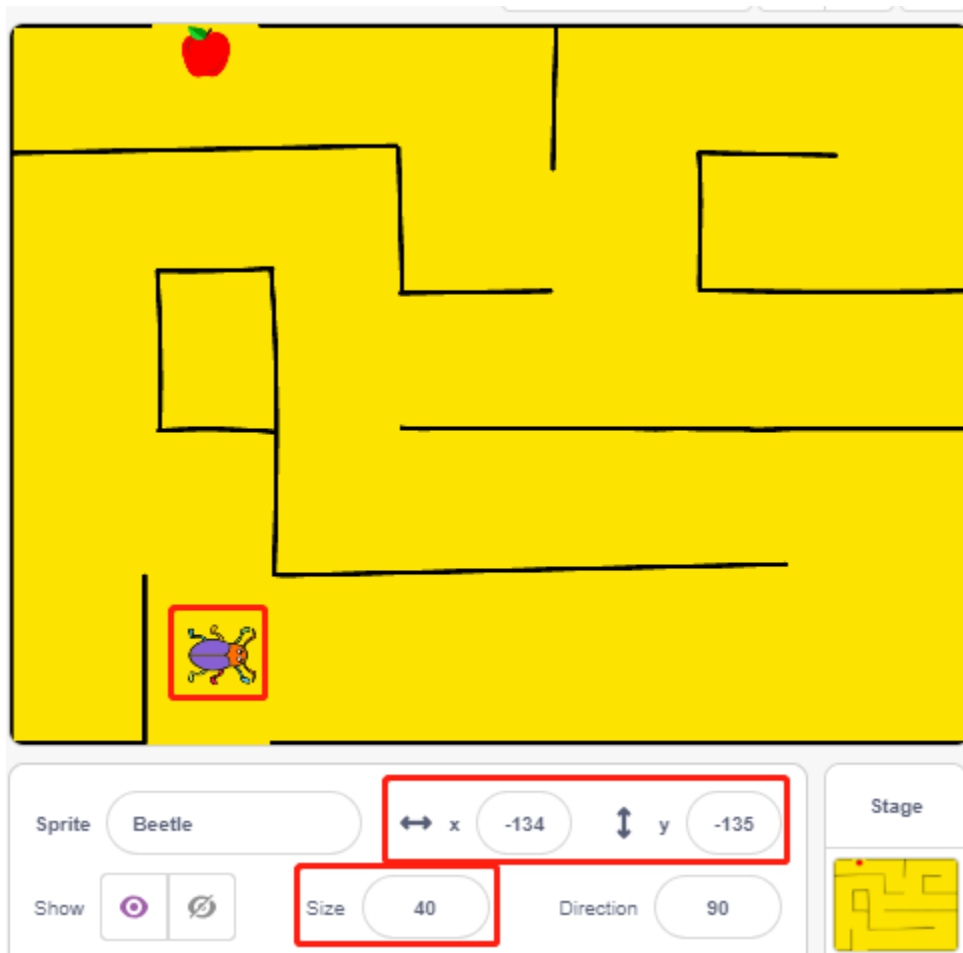
Fügen Sie einen **Maze** Hintergrund über den **Choose a backdrop** Knopf hinzu.



Löschen Sie den Standard-Sprite und wählen Sie dann den **Beetle** Sprite aus.



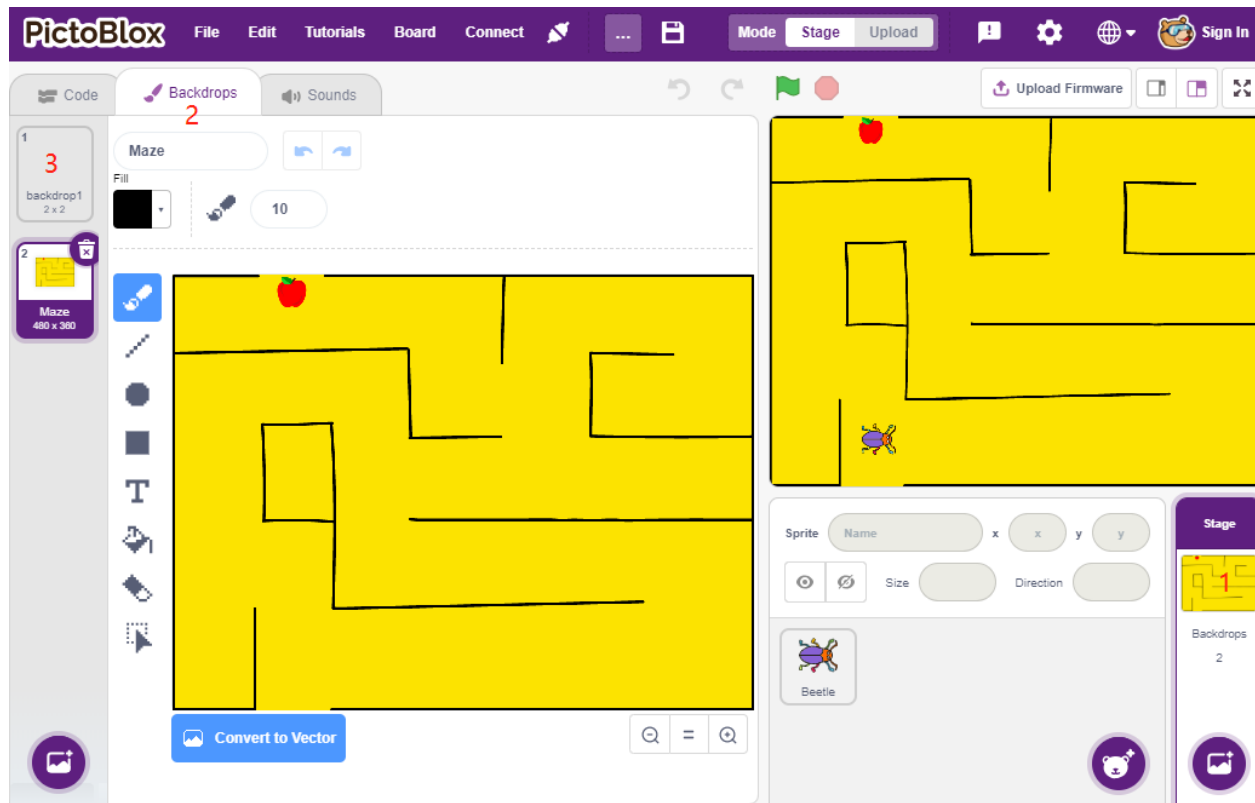
Platzieren Sie den **Beetle** Sprite am Eingang des **Maze** Hintergrunds, merken Sie sich die x,y-Koordinatenwerte an dieser Stelle und ändern Sie die Größe des Sprites auf 40%.



2. Hintergrund zeichnen

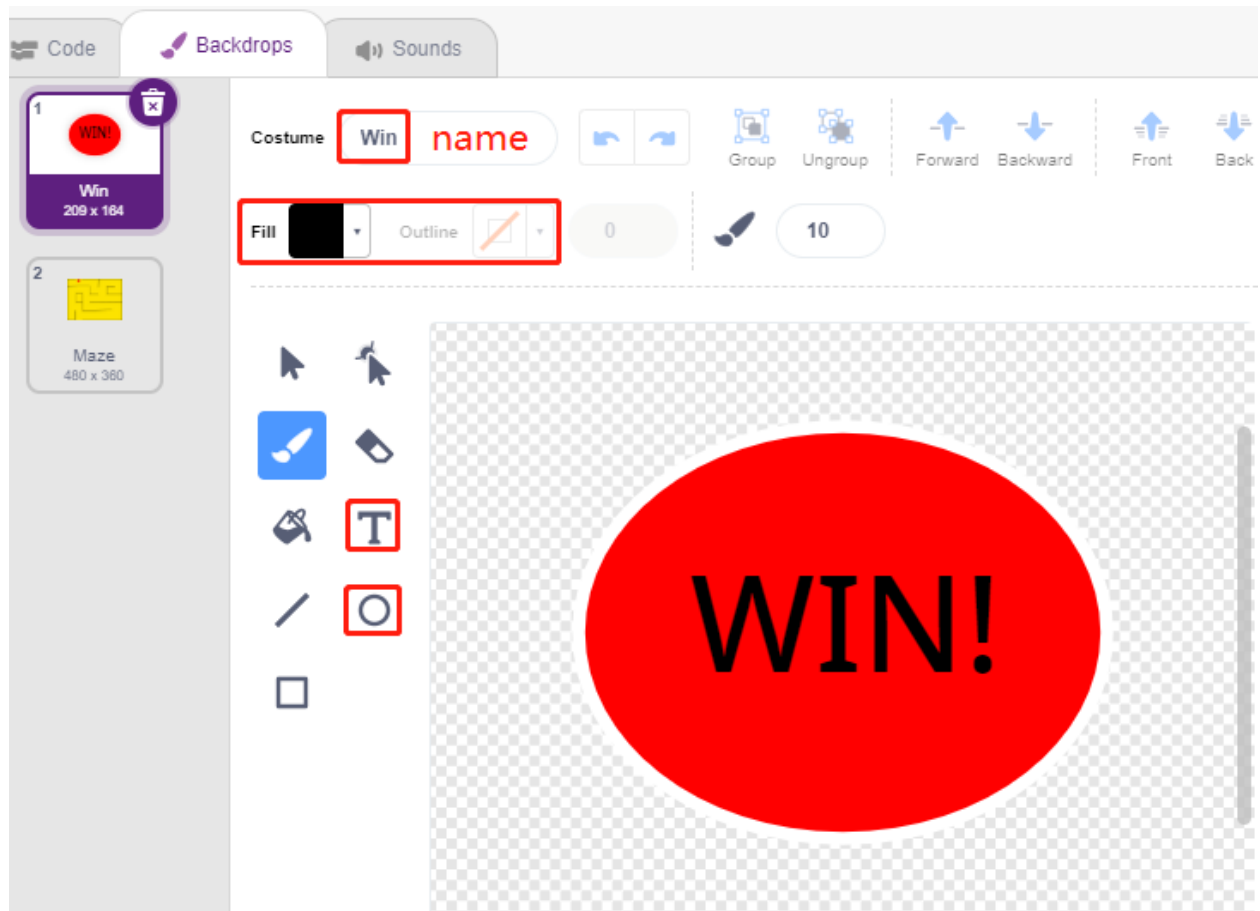
Jetzt wird es Zeit, einen Hintergrund zu zeichnen, auf dem das Zeichen WIN! erscheint.

Klicken Sie zunächst auf die Vorschau grafik des Hintergrunds, um zur Seite **Backdrops** zu gelangen, und klicken Sie dann auf den leeren Hintergrund1.



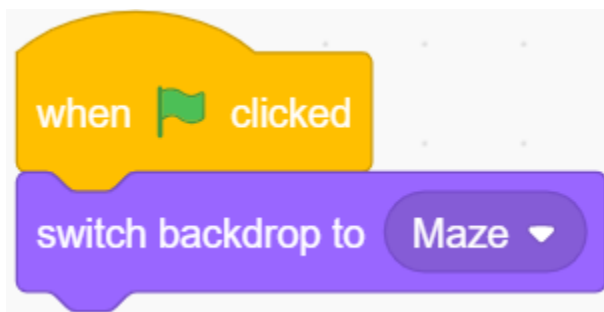
Beginnen Sie jetzt mit dem Zeichnen. Sie können sich an das untenstehende Bild halten oder Ihren eigenen Hintergrund entwerfen, solange die Darstellung gewinnend ist.

- Verwenden Sie das **Circle**-Werkzeug, um eine Ellipse ohne Umriss und in roter Farbe zu zeichnen.
- Verwenden Sie anschließend das **Text**-Werkzeug, um das Wort „WIN!“ zu schreiben. Setzen Sie die Zeichenfarbe auf Schwarz und passen Sie die Größe und Position des Textes an.
- Benennen Sie den Hintergrund **Win**.



3. Skripting für den Hintergrund

Der Hintergrund sollte beim Spielstart immer auf **Maze** wechseln.



4. Skripte für den Sprite Käfer schreiben

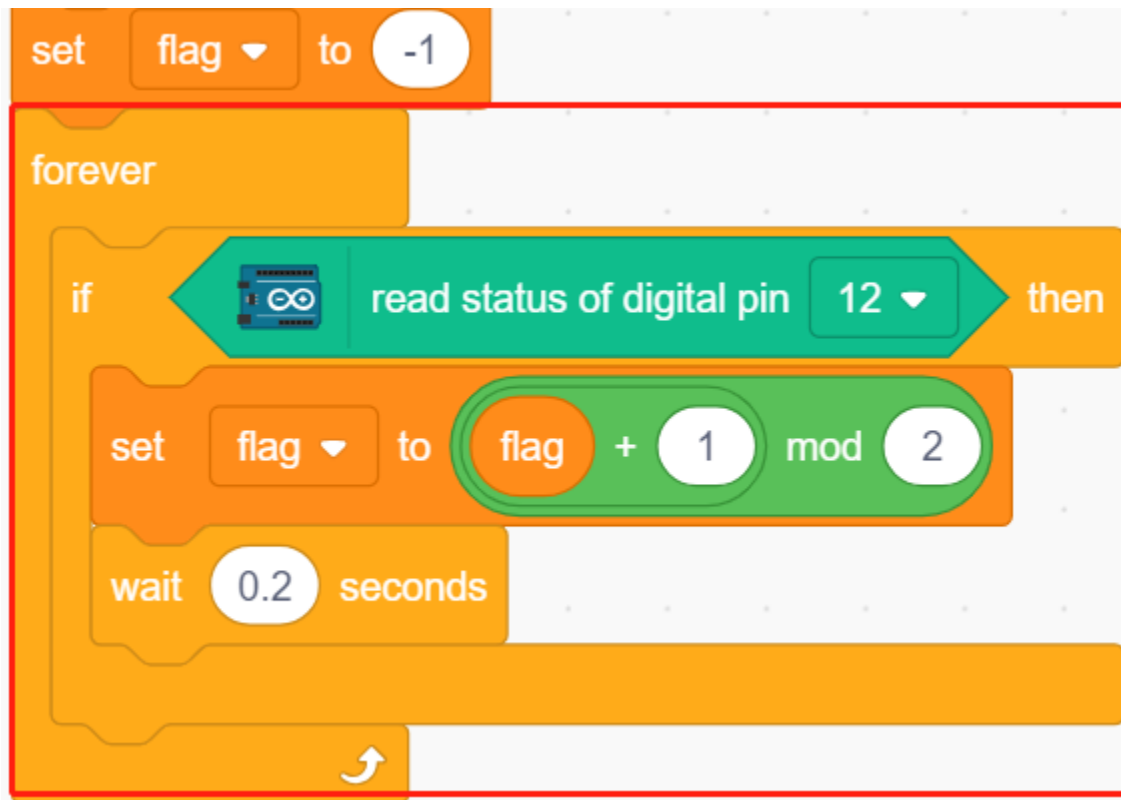
Schreiben Sie jetzt ein Skript, damit der Sprite **Beetle** mit Hilfe eines Knopfes vorwärts bewegt und die Richtung geändert werden kann. Der Ablauf ist wie folgt.

- Wenn die grüne Flagge angeklickt wird, setzen Sie den Winkel des **Beetle** auf 90 und die Position auf (-134, -134) oder ersetzen Sie diese durch den Koordinatenwert Ihrer eigenen Position. Erstellen Sie die Variable **flag** und setzen Sie den Anfangswert auf -1.

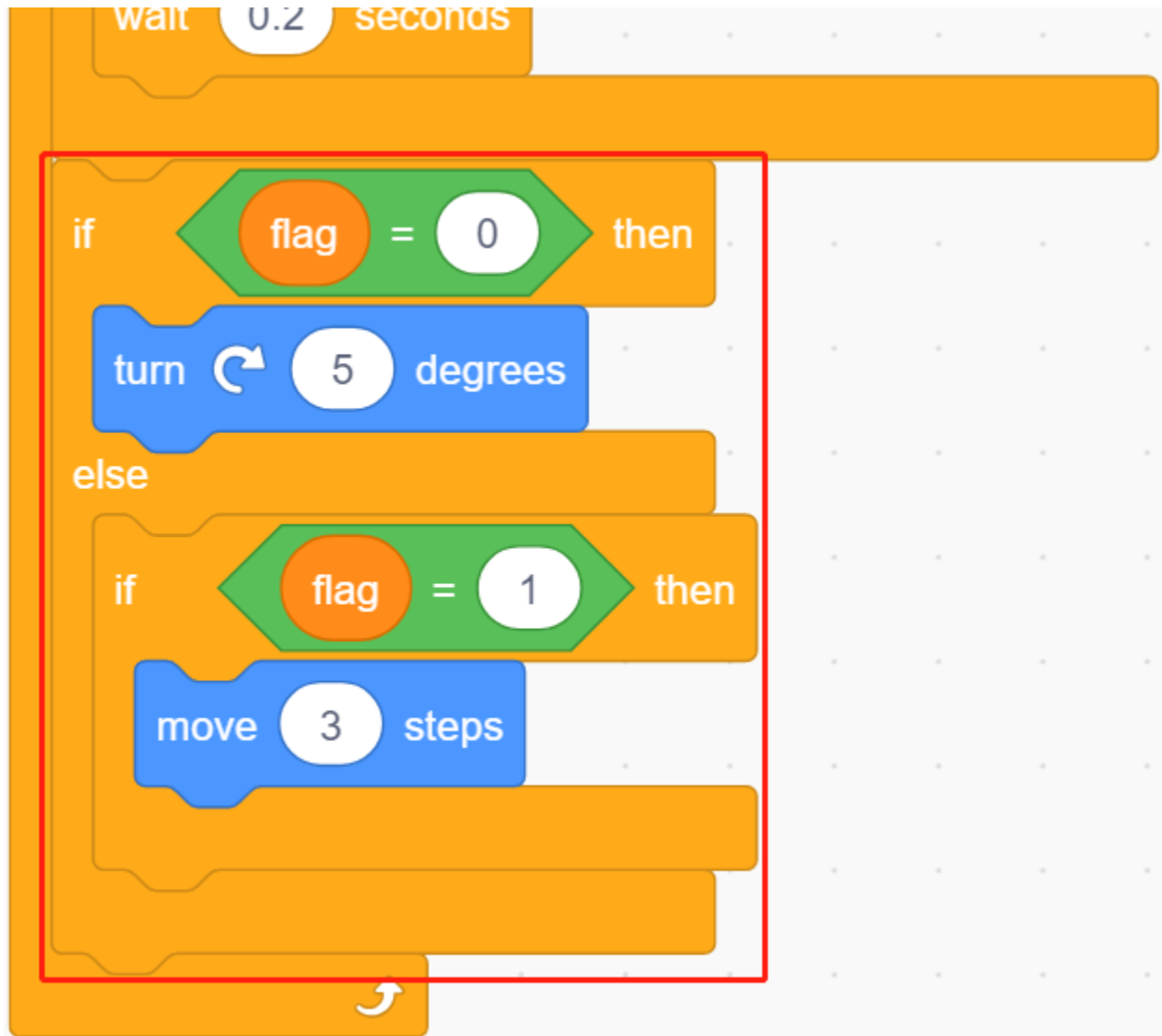


Im [forever]-Block werden vier [if]-Blöcke verwendet, um verschiedene mögliche Szenarien zu bestimmen.

- Wenn die Taste 1 (gedrückt) ist, verwenden Sie den [mod] Block, um den Wert der Variable **flag** zwischen 0 und 1 umzuschalten (abwechselnd 0 für diesen Druck und 1 für den nächsten Druck).

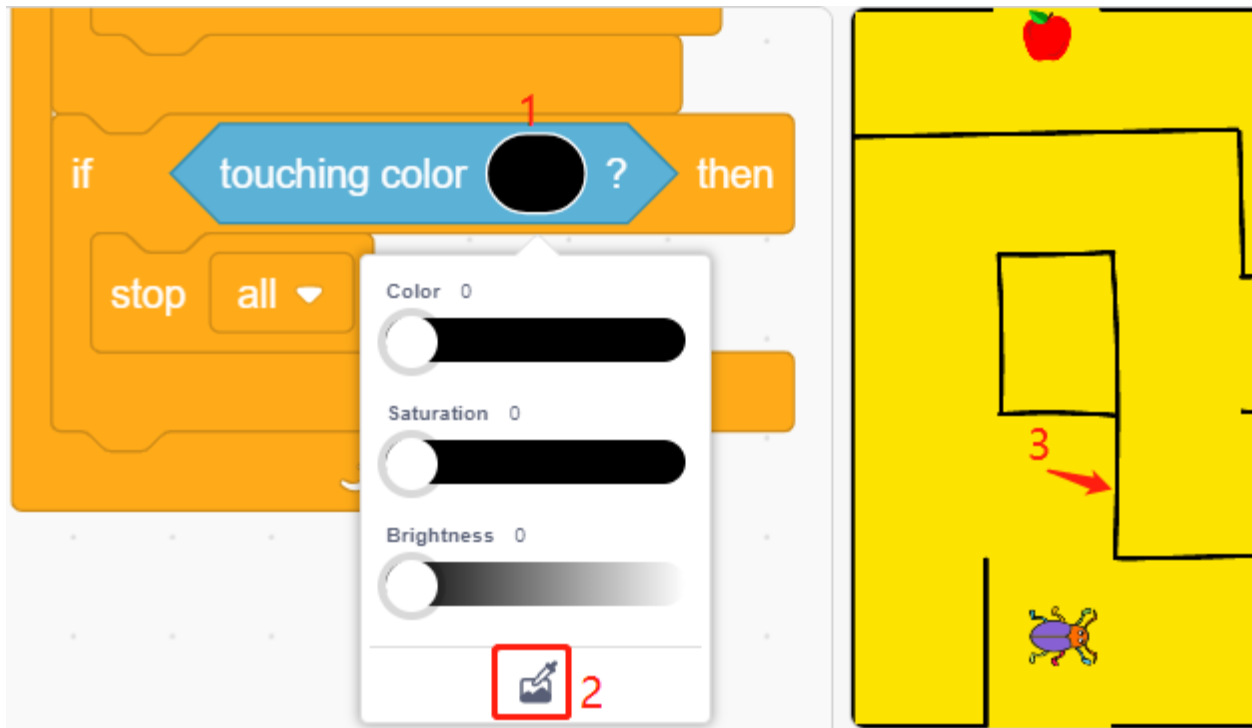


- Wenn Flagge=0 (dieser Tastendruck), soll sich der **Beetle**-Sprite im Uhrzeigersinn drehen. Wird dann festgestellt, dass Flagge gleich 1 ist (Taste erneut gedrückt), bewegt sich der **Beetle**-Sprite vorwärts. Andernfalls dreht er sich weiter im Uhrzeigersinn.

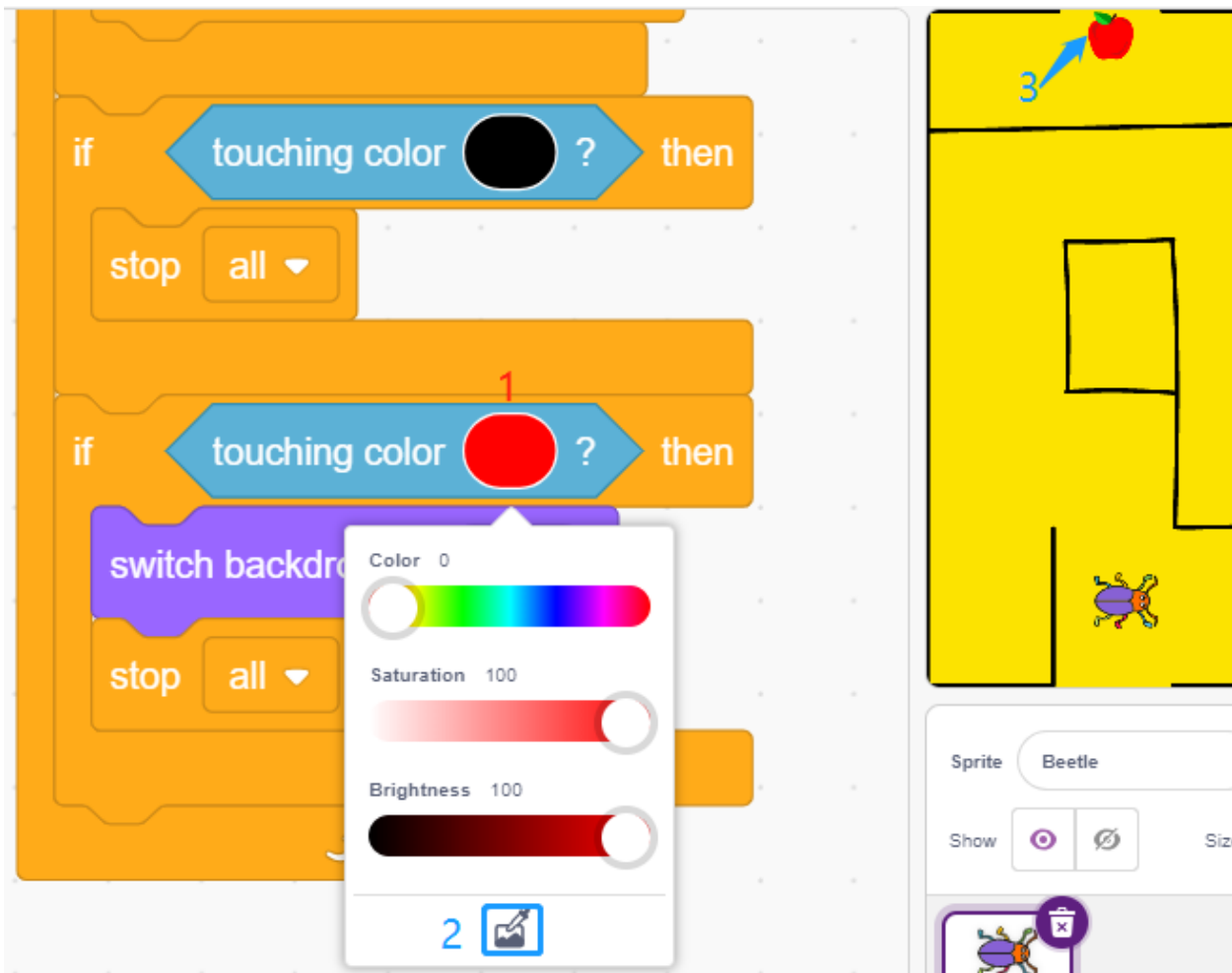


- Wenn der Käfer-Sprite Schwarz berührt (die schwarze Linie im **Maze** Hintergrund), endet das Spiel und das Skript wird gestoppt.

Bemerkung: Sie müssen im [Touch color]-Block auf den Farbbereich klicken und dann das Pipettenwerkzeug verwenden, um die Farbe der schwarzen Linie auf der Bühne aufzunehmen. Wenn Sie einfach Schwarz wählen, funktioniert dieser [Touch color]-Block nicht.



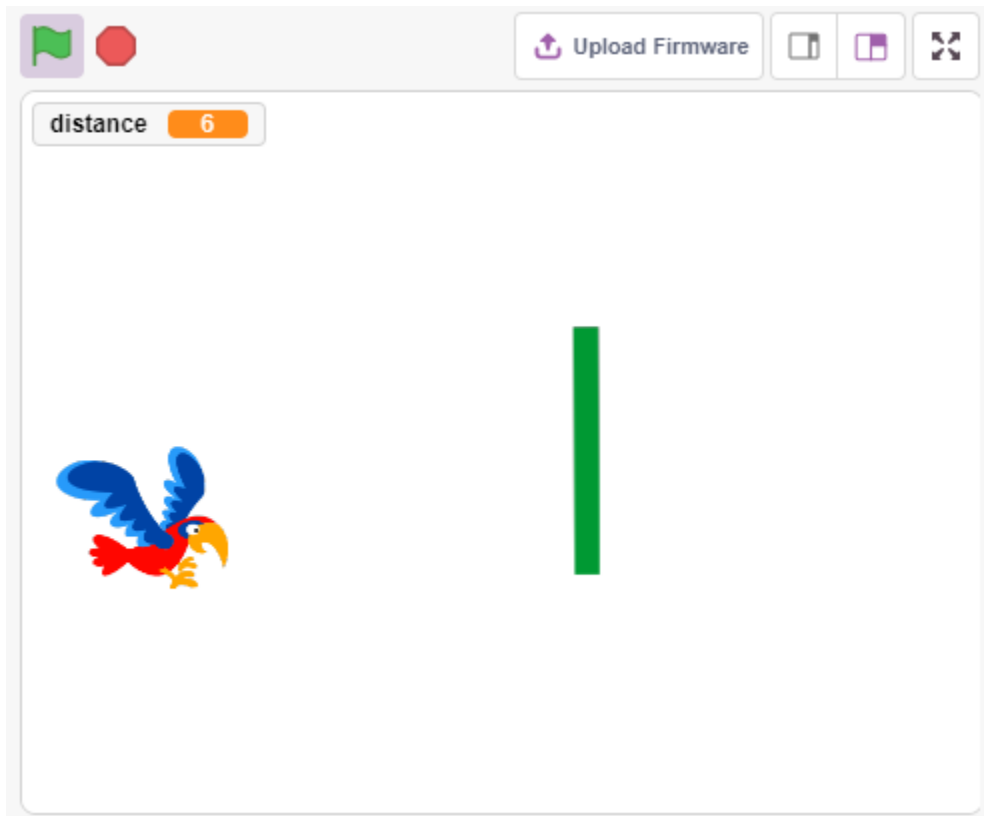
- Wenn der Käfer Rot berührt (Verwenden Sie ebenfalls das Pipettenwerkzeug, um die rote Farbe des Apfels aufzunehmen), wechselt der Hintergrund zu **Win**. Das bedeutet, das Spiel wurde erfolgreich abgeschlossen und das Skript wird gestoppt.



8.20 2.17 SPIEL - Flappy Papagei

Hier verwenden wir das Ultraschallmodul, um ein Flappy-Papagei-Spiel zu spielen.

Nachdem das Skript ausgeführt wird, wird der grüne Bambus langsam von rechts nach links auf zufälliger Höhe bewegt. Platziere nun deine Hand über dem Ultraschallmodul. Wenn der Abstand zwischen deiner Hand und dem Modul weniger als 10 cm beträgt, fliegt der Papagei nach oben, sonst fällt er nach unten. Du musst den Abstand zwischen deiner Hand und dem Ultraschallmodul kontrollieren, sodass der Papagei den grünen Bambus (Schläger) vermeiden kann. Berührt er diesen, ist das Spiel vorbei.



8.20.1 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+	

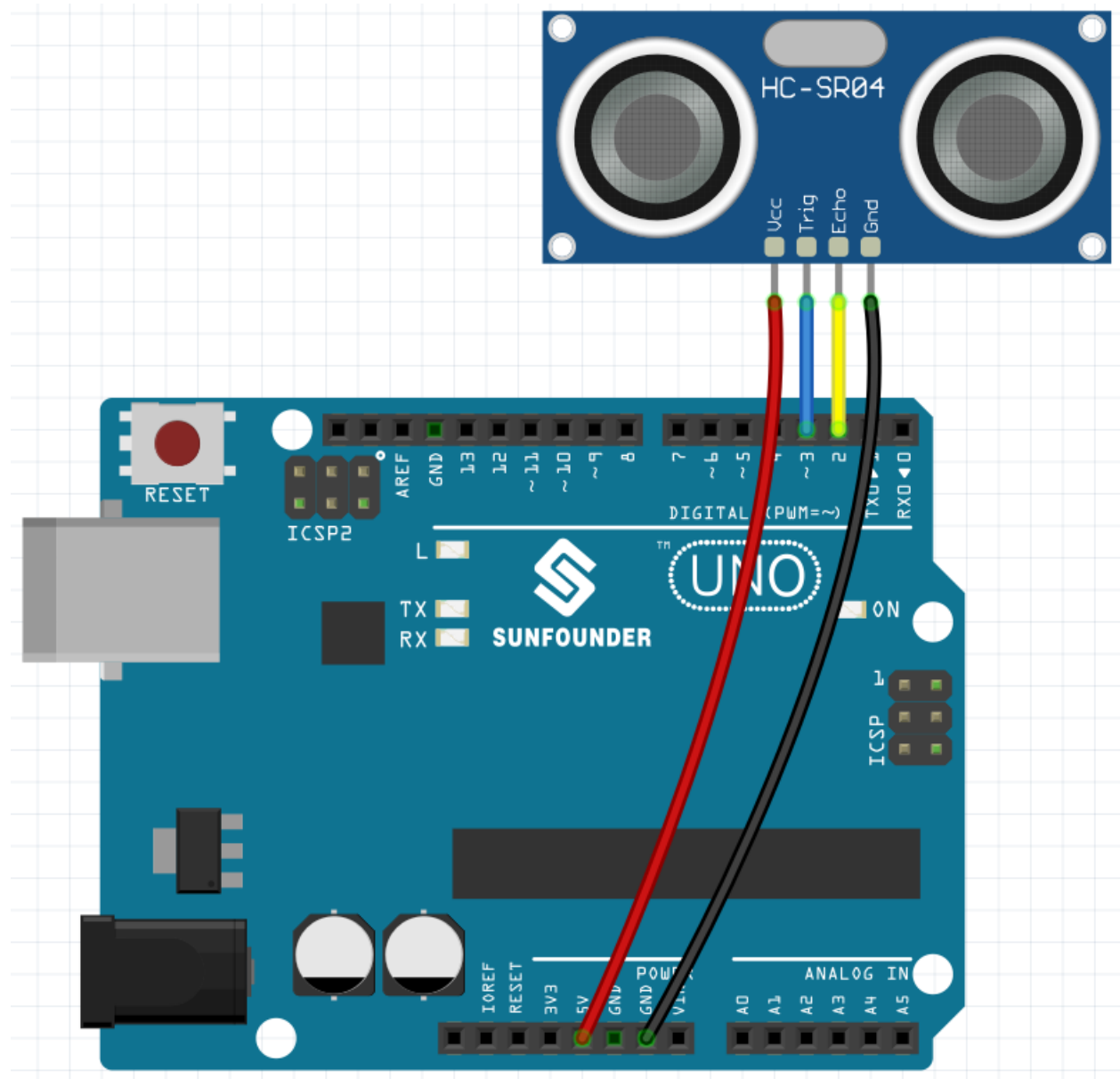
Du kannst sie auch einzeln über die folgenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>Ultraschall-Modul</i>	

8.20.2 Schaltung aufbauen

Ein Ultraschall-Sensormodul ist ein Instrument, das den Abstand zu einem Objekt mithilfe von Ultraschallwellen misst. Es hat zwei Sonden. Eine sendet Ultraschallwellen und die andere empfängt die Wellen und wandelt die Sendungs- und Empfangszeit in eine Entfernung um, um so den Abstand zwischen dem Gerät und einem Hindernis zu bestimmen.

Baue jetzt die Schaltung gemäß dem folgenden Diagramm.

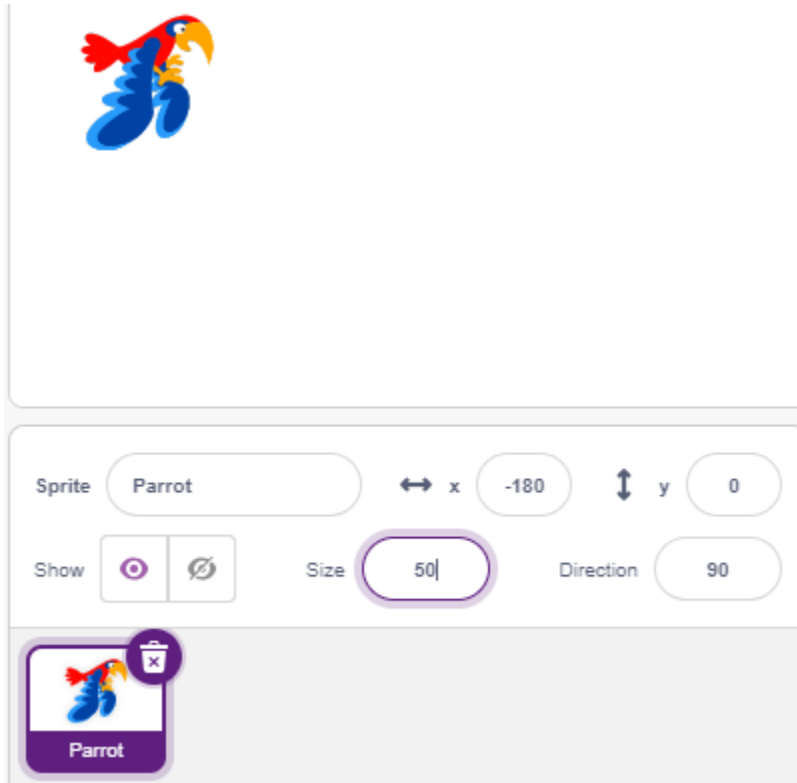


8.20.3 Programmierung

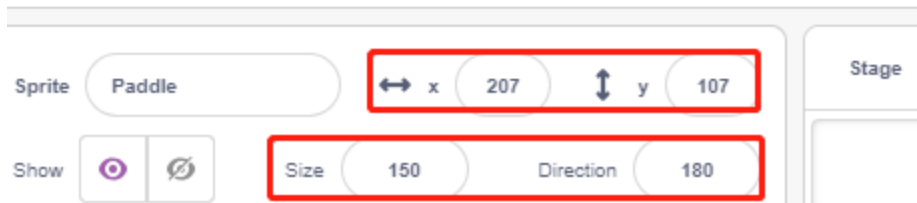
Das gewünschte Ergebnis ist die Steuerung der Flughöhe des Sprites **Parrot** mithilfe des Ultraschallmoduls, während er den Sprite **Paddle** vermeidet.

1. Füge einen Sprite hinzu

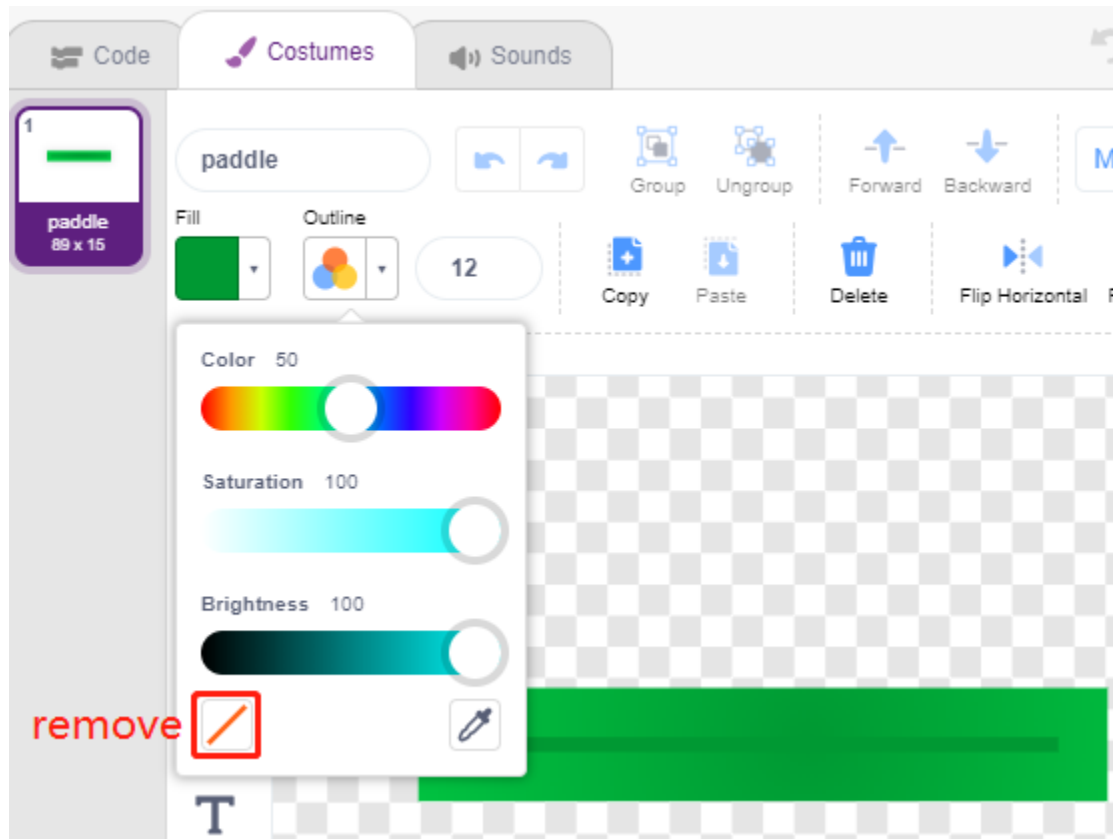
Lösche den Standardsprite und verwende die Schaltfläche **Choose a Sprite**, um den Sprite **Parrot** hinzuzufügen. Setze seine Größe auf 50% und verschiebe seine Position zum linken Zentrum.



Füge jetzt den Sprite **Paddle** hinzu, setze seine Größe auf 150%, setze seinen Winkel auf 180 und verschiebe seine Anfangsposition in die obere rechte Ecke.



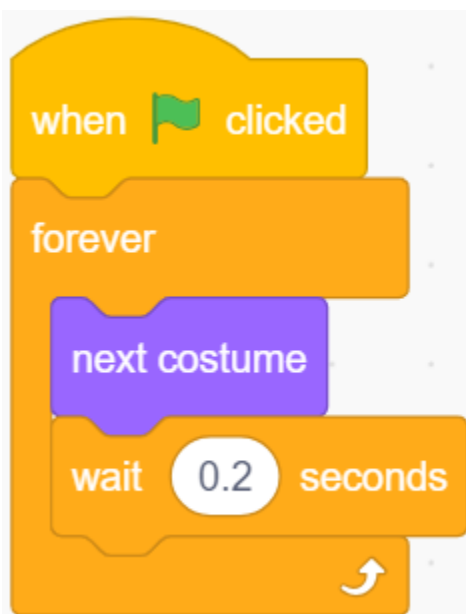
Gehe zur **Costumes**-Seite des Sprites **Paddle** und entferne den Umriss.



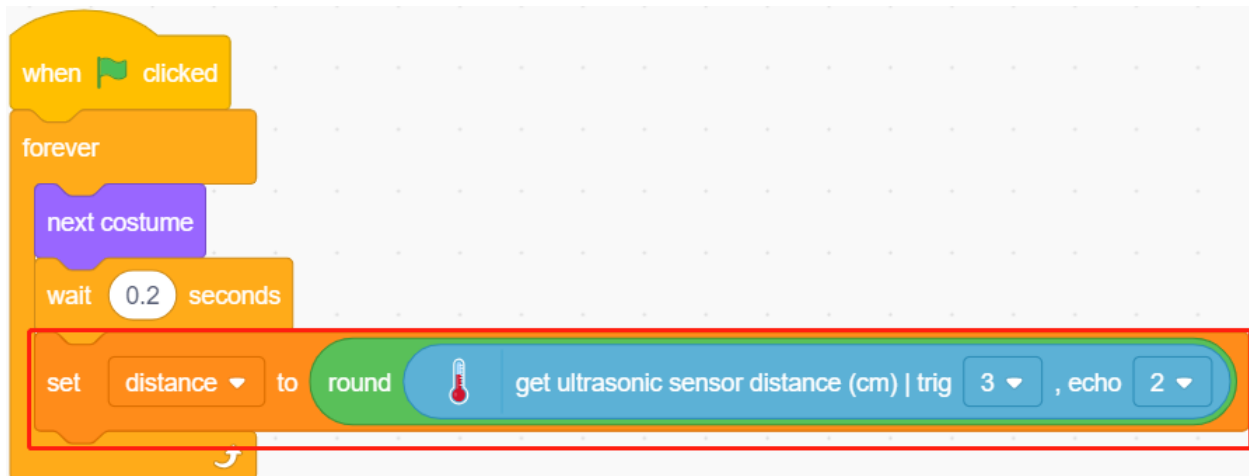
2. Skripting für den Papagei-Sprite

Jetzt das Skript für den **Parrot**-Sprite schreiben, der fliegt und dessen Flughöhe durch die Erkennungsentfernung des Ultraschallmoduls bestimmt wird.

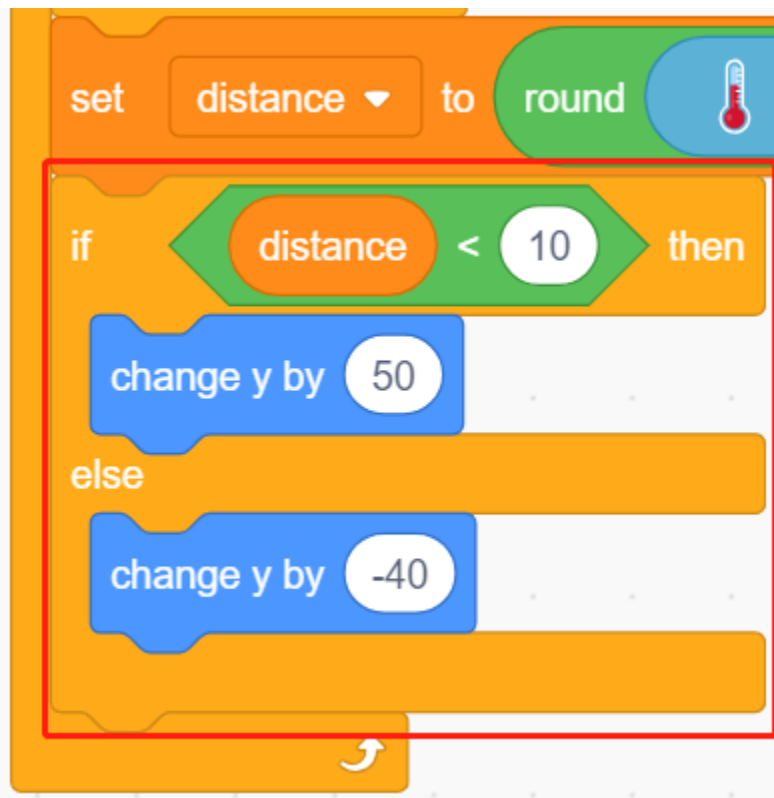
- Wenn auf die grüne Flagge geklickt wird, wechsele das Kostüm alle 0,2s, sodass es ständig fliegt.



- Lies den Wert des Ultraschallmoduls und speichere ihn in der Variable **distance** nachdem er mit dem [round] Block gerundet wurde.



- Wenn die Ultraschall-Erkennungsentfernung weniger als 10 cm beträgt, lass den y-Koordinatenwert um 50 erhöhen, dann wird der **Parrot**-Sprite nach oben fliegen. Andernfalls wird der y-Koordinatenwert um 40 verringert und der **Parrot** wird fallen.



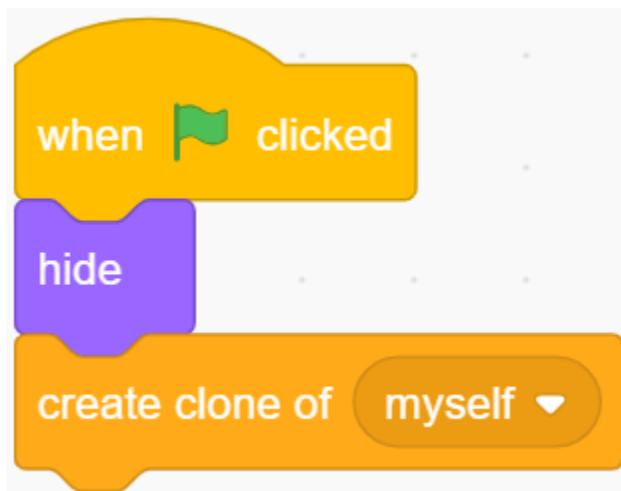
- Wenn der Sprite **Parrot** den Sprite **Paddle** berührt, endet das Spiel und das Skript hört auf zu laufen.



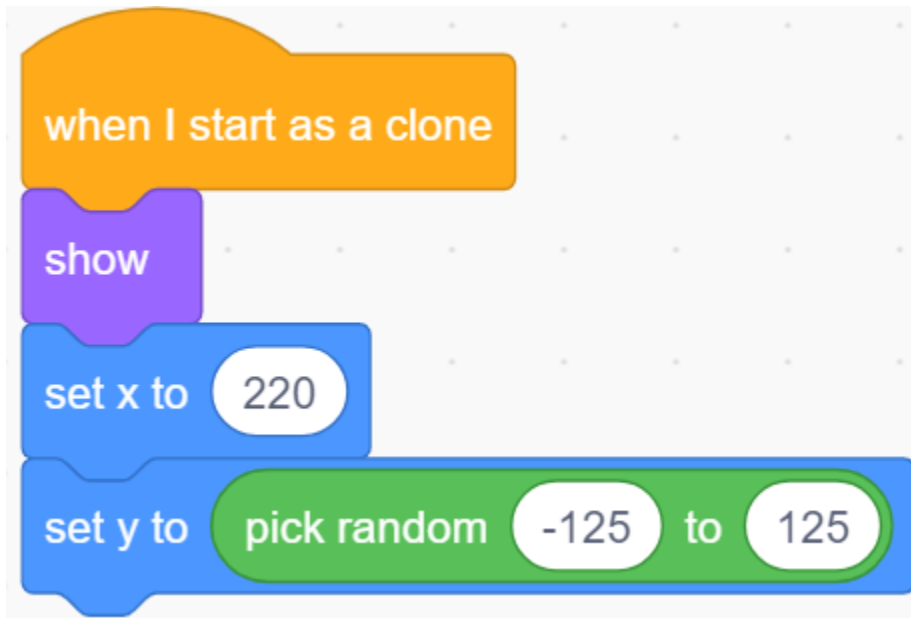
3. Skripting für den Schläger-Sprite

Schreibe jetzt das Skript für den **Paddle**-Sprite, der zufällig auf der Bühne erscheinen muss.

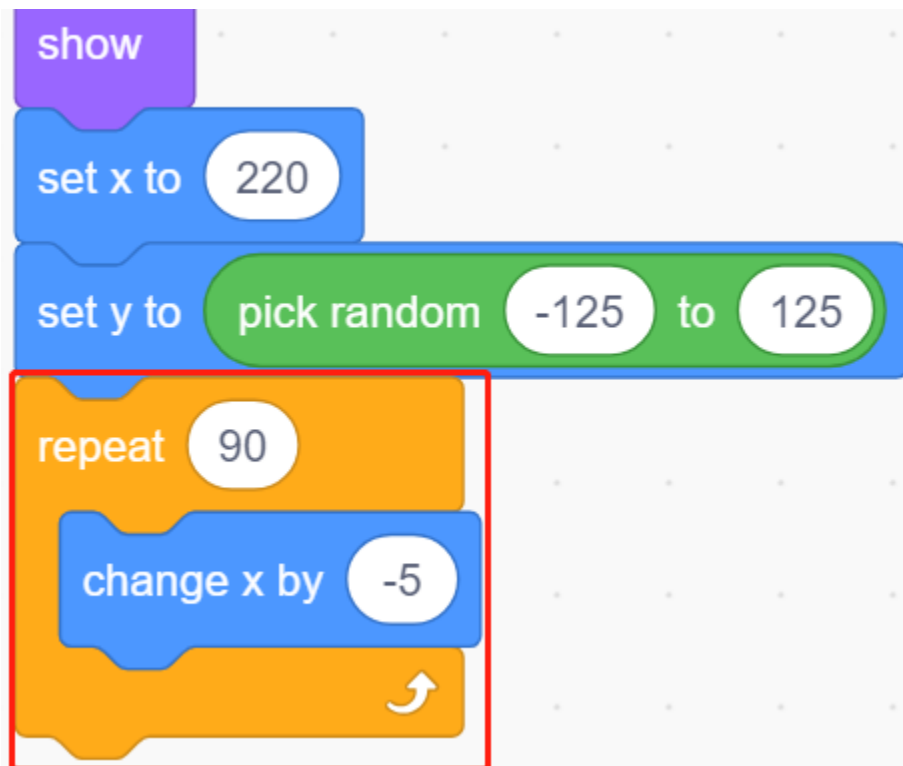
- Verstecke den Sprite **Paddle**, wenn auf die grüne Flagge geklickt wird, und klonen ihn gleichzeitig. Der [create clone of] Block ist ein Steuerblock und ein Stapelblock. Er erstellt einen Klon des Sprites im Argument. Er kann auch den Sprite klonen, in dem er läuft, und Klone von Klonen erstellen, rekursiv.



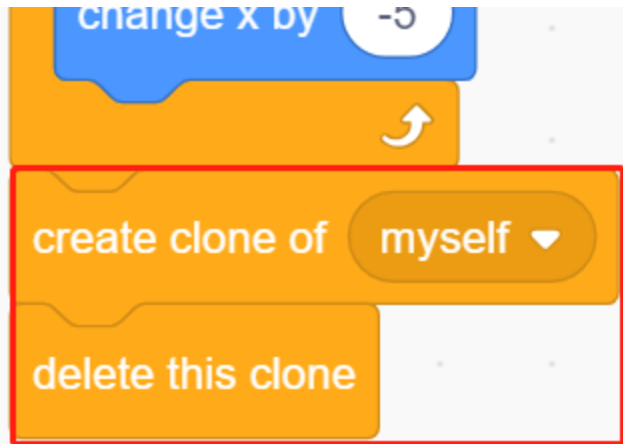
- Wenn **Paddle** als Klon dargestellt wird, beträgt seine Position 220 (ganz rechts) für die x-Koordinate und seine y-Koordinate liegt zufällig zwischen (-125 bis 125) (zufällige Höhe).



- Verwende den [wiederholen] Block, um seinen x-Koordinatenwert langsam zu verringern. So siehst du den Klon des **Paddle**-Sprites, der langsam von rechts nach links bewegt wird, bis er verschwindet.



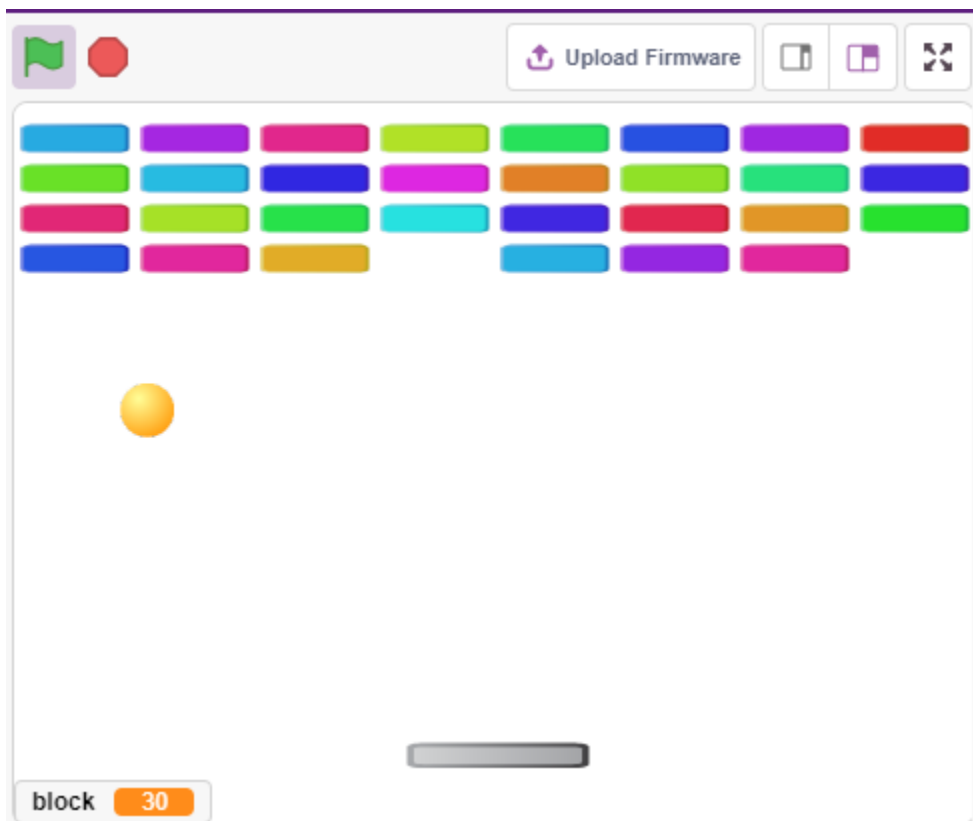
- Klone einen neuen **Paddle**-Sprite und lösche den vorherigen Klon.



8.21 2.18 SPIEL - Breakout-Klon

Hier verwenden wir das Potentiometer, um ein Breakout-Klon-Spiel zu spielen.

Nachdem Sie auf die grüne Flagge geklickt haben, müssen Sie das Potentiometer verwenden, um den Schläger auf der Bühne zu steuern, um den Ball zu fangen. Dadurch kann er nach oben steigen und die Ziegelsteine treffen. Wenn alle Ziegelsteine verschwinden, haben Sie gewonnen. Wenn Sie den Ball nicht fangen, haben Sie verloren.



8.21.1 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

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

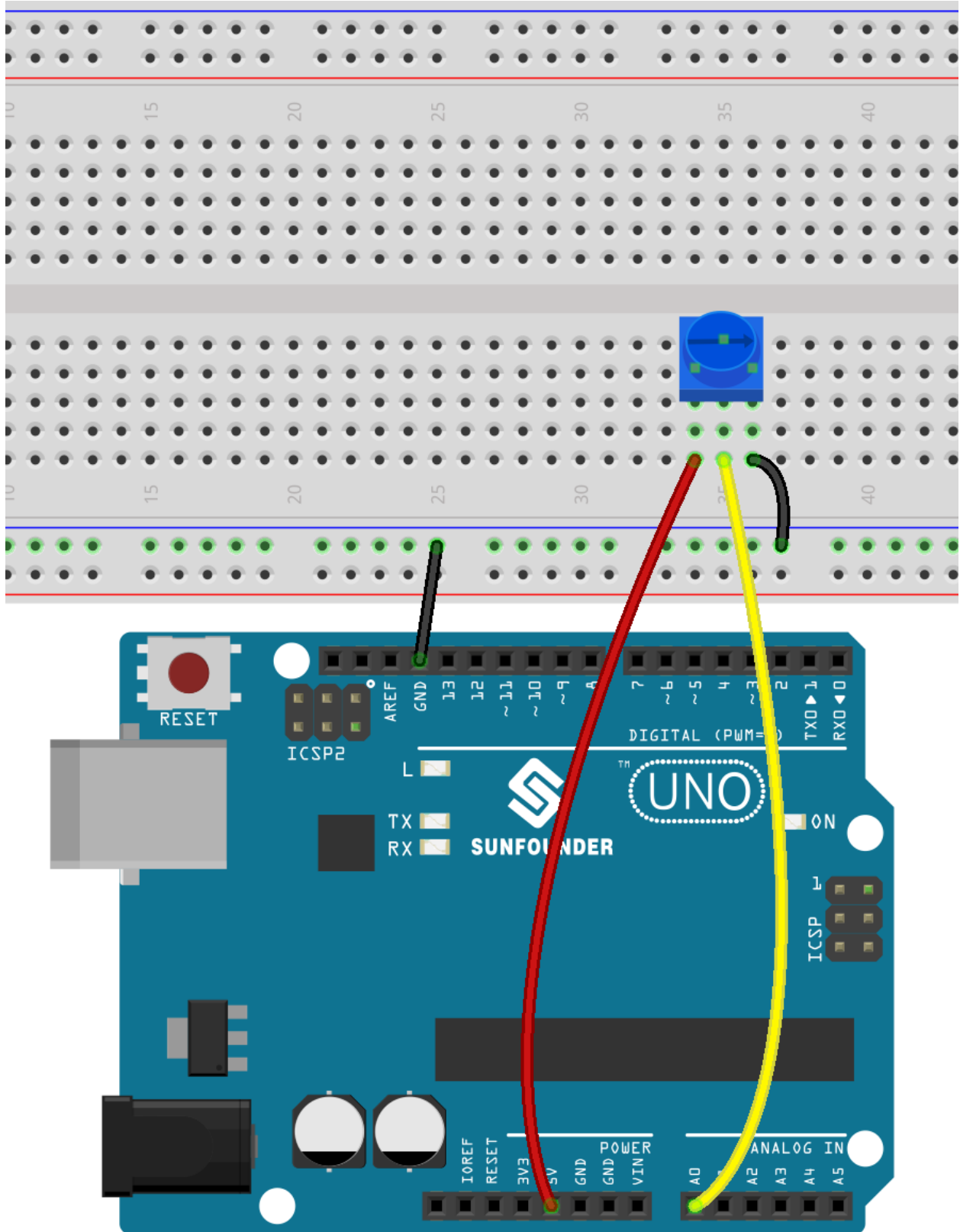
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Sie können sie auch einzeln über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Potentiometer</i>	

8.21.2 Schaltung aufbauen

Das Potentiometer ist ein Widerstandselement mit 3 Anschlüssen, die 2 seitlichen Pins sind mit 5V und GND verbunden, und der mittlere Pin ist mit A0 verbunden. Nach der Umwandlung durch den ADC-Wandler des Arduino-Boards liegt der Wertebereich zwischen 0-1023.



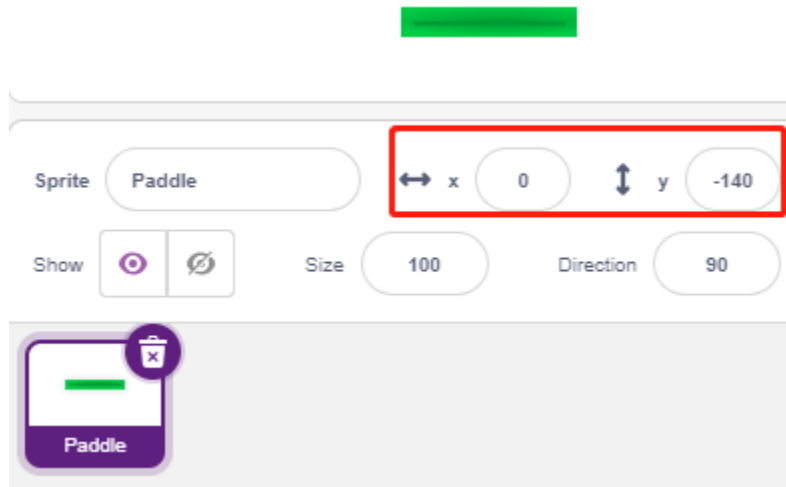
8.21.3 Programmierung

Es gibt 3 Sprites auf der Bühne.

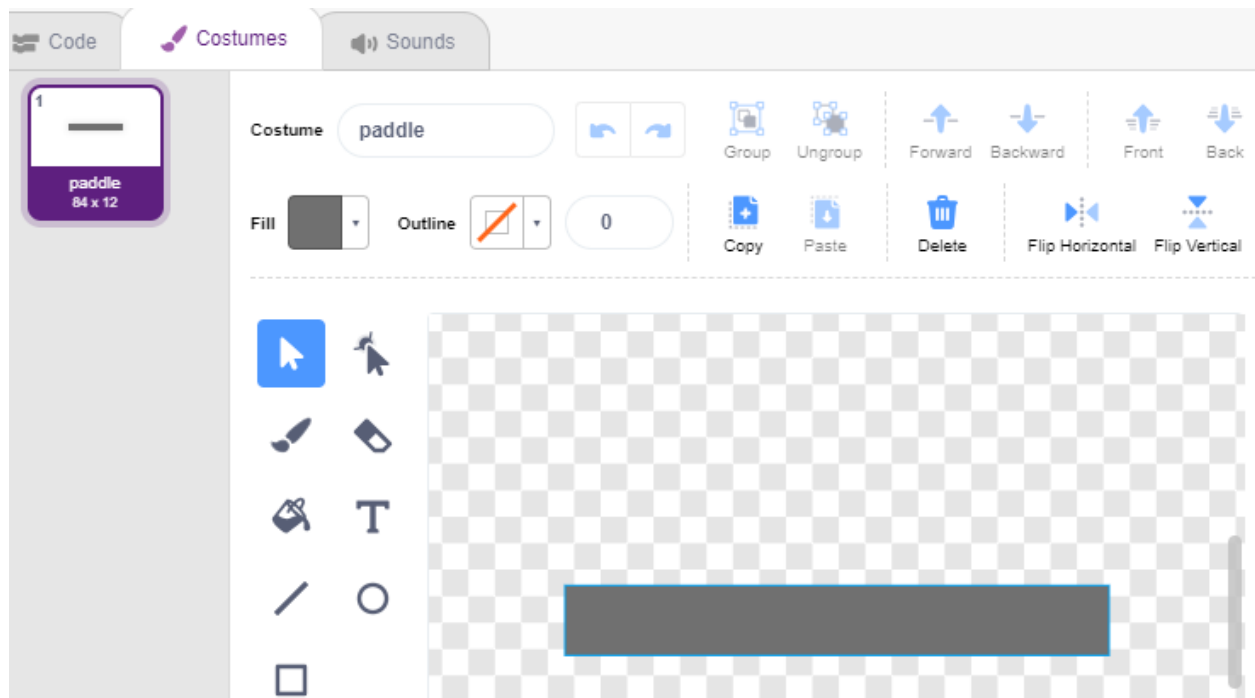
1. Schläger-Sprite (Paddle sprite)

Das Ziel des **Paddle** ist es, dass die Ausgangsposition in der Mitte des unteren Bühnenrandes liegt und durch ein Potentiometer gesteuert wird, um es nach links oder rechts zu bewegen.

- Löschen Sie den Standard-Sprite, verwenden Sie die Schaltfläche **Choose a Sprite**, um den **Paddle**-Sprite hinzuzufügen, und setzen Sie seine x- und y-Koordinaten auf (0, -140).

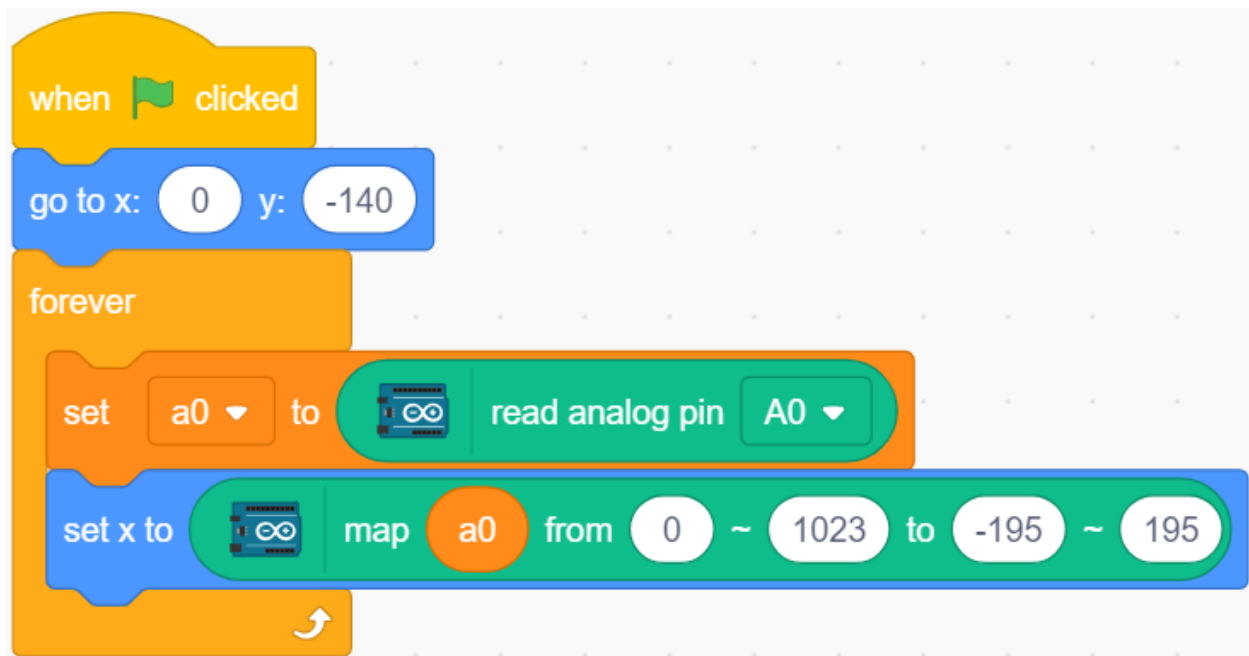


- Gehen Sie zur Seite **Costumes**, entfernen Sie den Umriss und ändern Sie seine Farbe in dunkelgrau.



- Jetzt skripten Sie den **Paddle**-Sprite, um seine Anfangsposition auf (0, -140) zu setzen, wenn die grüne Flagge angeklickt wird, und lesen Sie den Wert von A0 (Potentiometer) in die Variable **a0**. Da der **Paddle**-Sprite sich von links nach rechts auf der Bühne bei x-Koordinaten -195~195 bewegt, müssen Sie den [map] Block verwenden,

um den Bereich der Variable **a0** von 0~1023 auf -195~195 abzubilden.

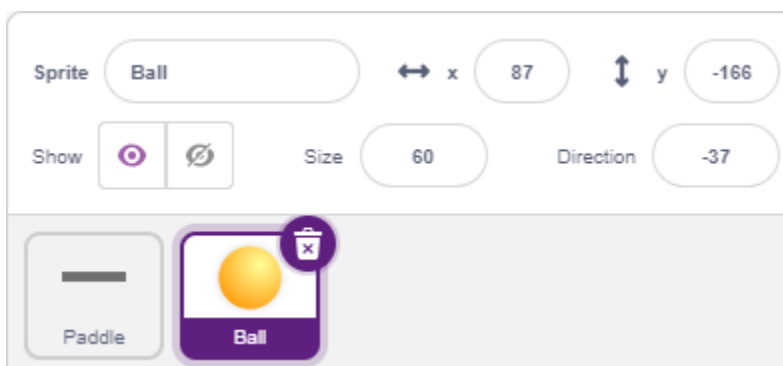


- Jetzt können Sie das Potentiometer drehen und überprüfen, ob der **Paddle** sich links und rechts auf der Bühne bewegen kann.

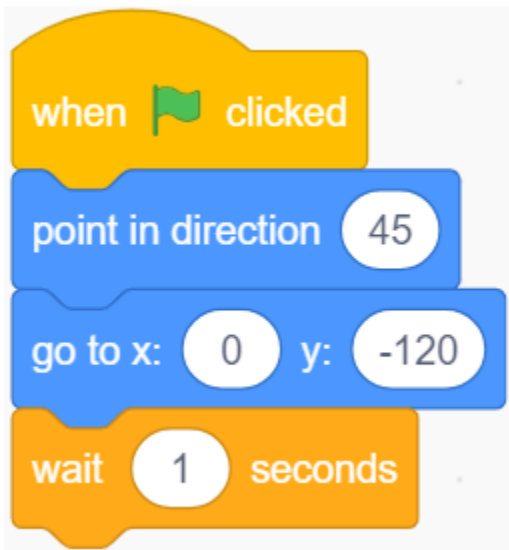
2. Ball-Sprite

Der Ball-Sprite soll sich auf der Bühne bewegen und abprallen, wenn er den Rand berührt. Er prallt nach unten ab, wenn er den Block oberhalb der Bühne berührt. Er prallt nach oben ab, wenn er während des Fallens den Paddle-Sprite berührt. Tut er das nicht, wird das Skript gestoppt und das Spiel endet.

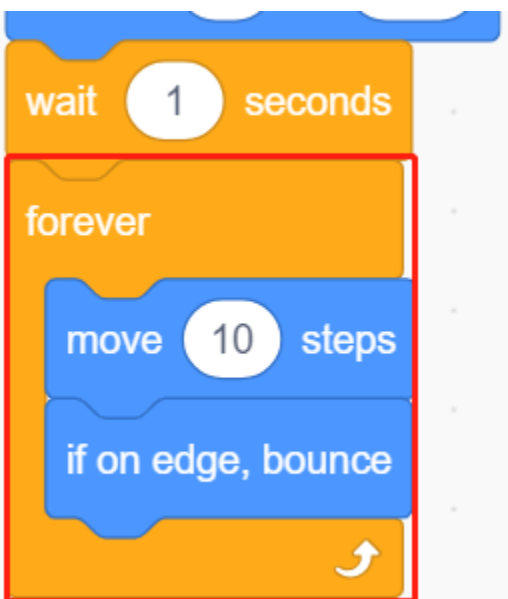
- Fügen Sie den **Ball**-Sprite hinzu.



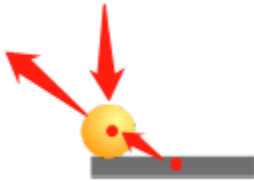
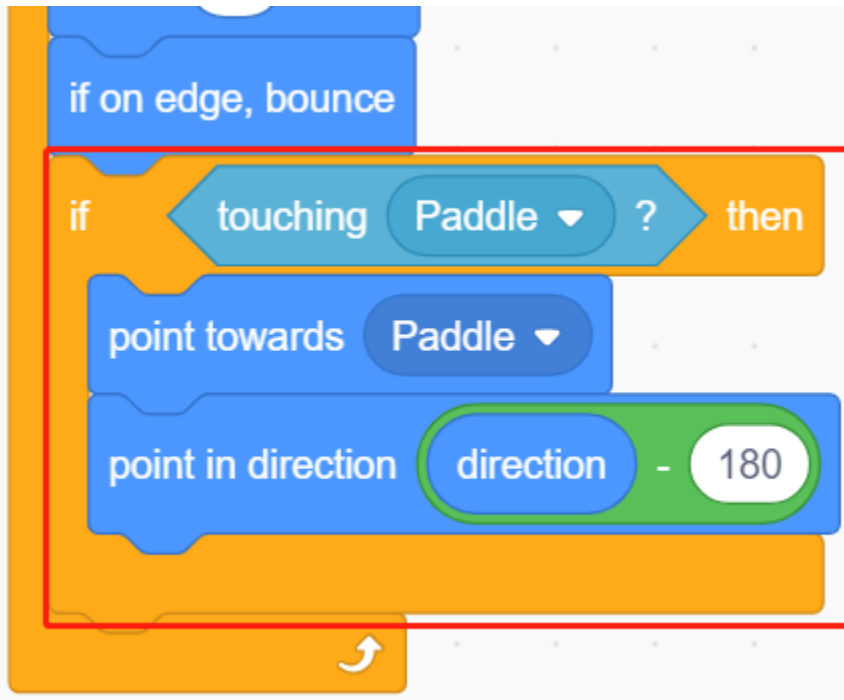
- Wenn auf die grüne Flagge geklickt wird, setzen Sie den Winkel des **Ball**-Sprites auf 45° und die Anfangsposition auf (0, -120).



- Lassen Sie den **Ball**-Sprite sich auf der Bühne bewegen und abprallen, wenn er den Rand berührt. Klicken Sie auf die grüne Flagge, um den Effekt zu sehen.



- Wenn der **Ball**-Sprite den **Paddle**-Sprite berührt, führen Sie eine Reflexion durch. Der einfache Weg, dies zu tun, ist, den Winkel direkt umzukehren. Dann werden Sie jedoch feststellen, dass der Pfad des Balls völlig festgelegt ist, was zu langweilig ist. Daher verwenden wir das Zentrum der beiden Sprites, um den Ball in die entgegengesetzte Richtung des Zentrums der Abschirmung abprallen zu lassen.



- Wenn der **Ball**-Sprite an den Rand der Bühne fällt, wird das Skript gestoppt und das Spiel endet.



3. Block1-Sprite

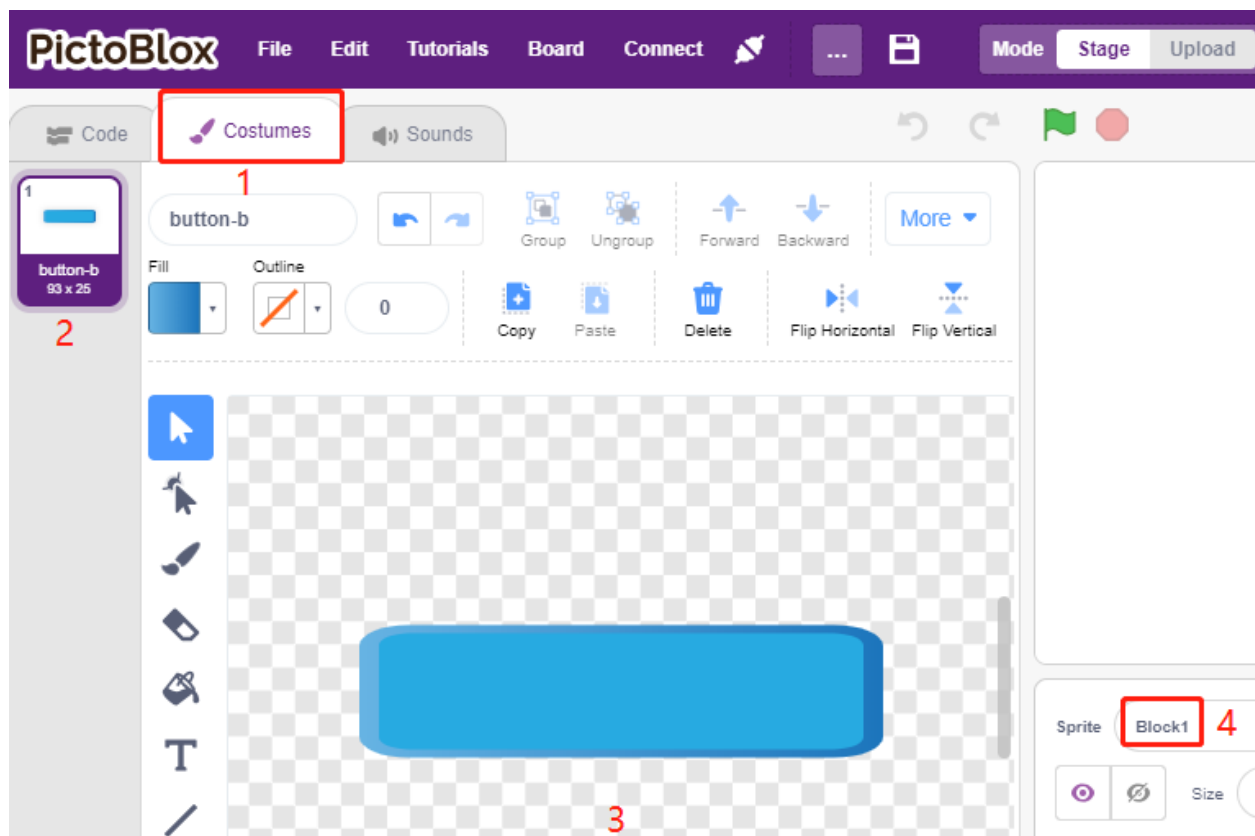
Der **Block1**-Sprite soll mit dem Effekt erscheinen, 4x8 Klone von sich selbst in einer zufälligen Farbe oberhalb der Bühne zu erstellen und einen Klon zu löschen, wenn er vom **Ball**-Sprite berührt wird.

Der **Block1**-Sprite ist nicht in der **PictoBlox**-Bibliothek verfügbar. Sie müssen ihn selbst zeichnen oder mit einem vorhandenen Sprite modifizieren. Hier werden wir ihn mit dem **Button3**-Sprite modifizieren.

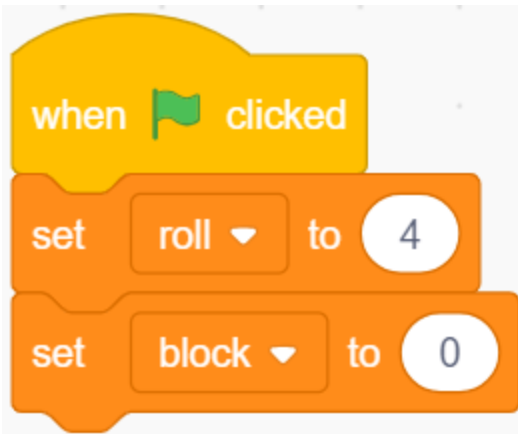
- Nachdem Sie den **Button3**-Sprite hinzugefügt haben, gehen Sie zur Seite **Costumes**. Löschen Sie zuerst **button-a**, verkleinern Sie dann sowohl die Breite als auch die Höhe von **button-b** und ändern Sie den Sprite-Namen in **Block1**, wie im folgenden Bild gezeigt.

Bemerkung:

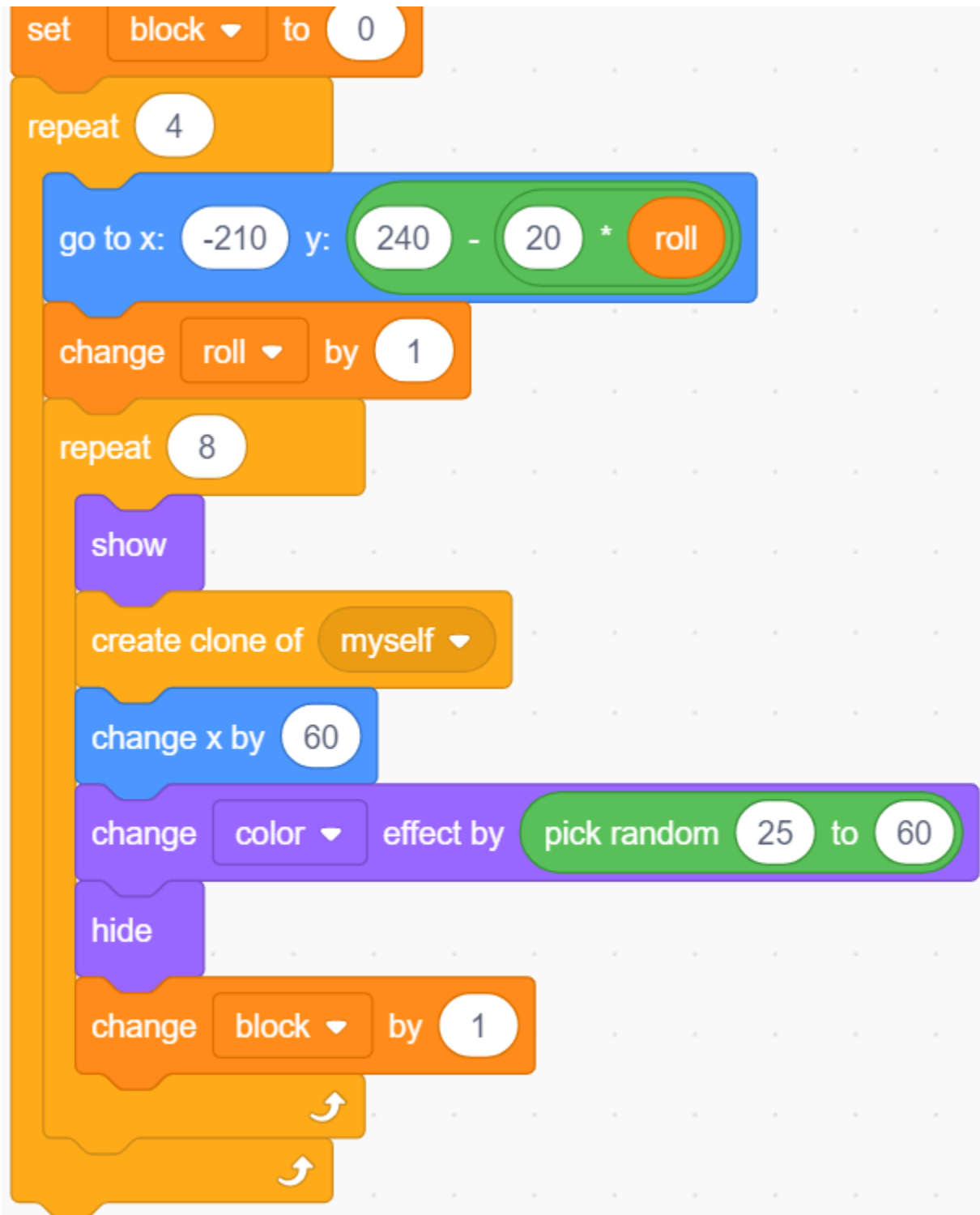
- Für die Breite von **Block1** können Sie wahrscheinlich auf dem Bildschirm simulieren, um zu sehen, ob Sie 8 in einer Reihe platzieren können. Wenn nicht, reduzieren Sie die Breite entsprechend.
- Beim Verkleinern des **Block1**-Sprites müssen Sie den Mittelpunkt in der Mitte des Sprites halten.



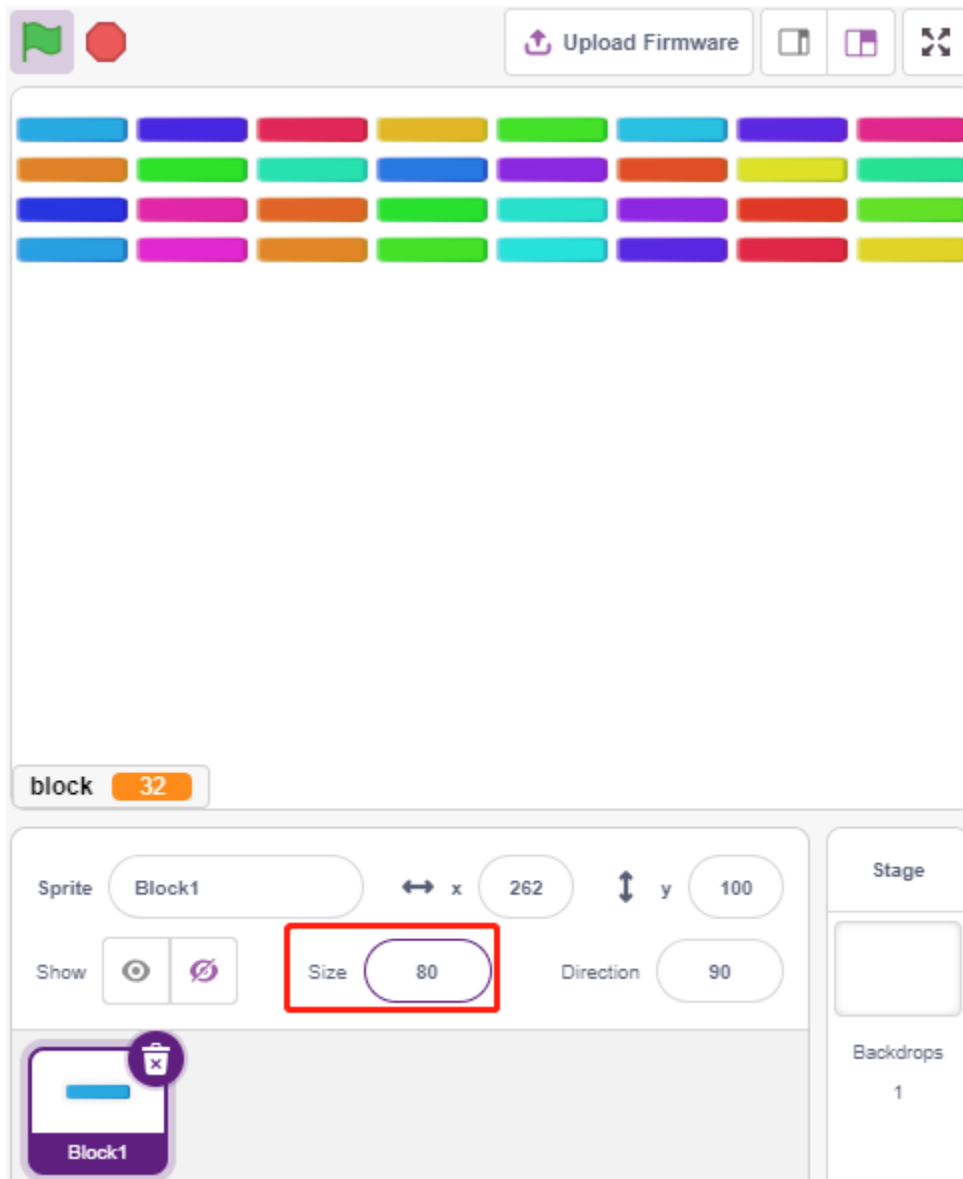
- Erstellen Sie jetzt zuerst 2 Variablen, **block**, um die Anzahl der Blöcke und **roll**, um die Anzahl der Reihen zu speichern.



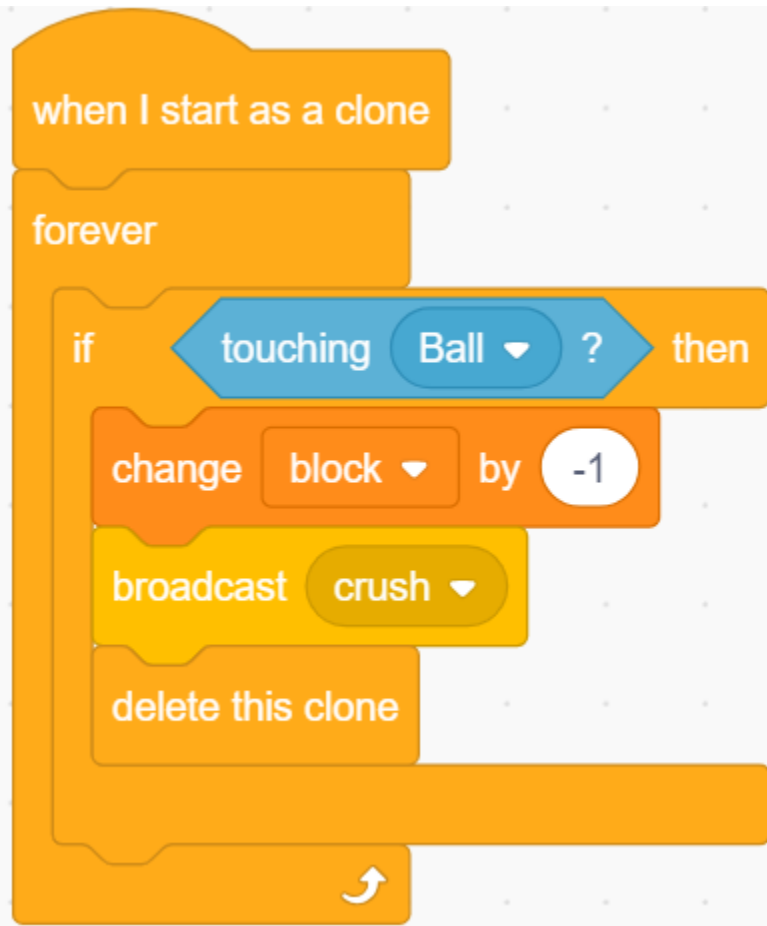
- Wir müssen einen Klon des **Block1**-Sprites erstellen, damit er von links nach rechts, von oben nach unten, eins nach dem anderen, insgesamt 4x8 mit zufälligen Farben angezeigt wird.



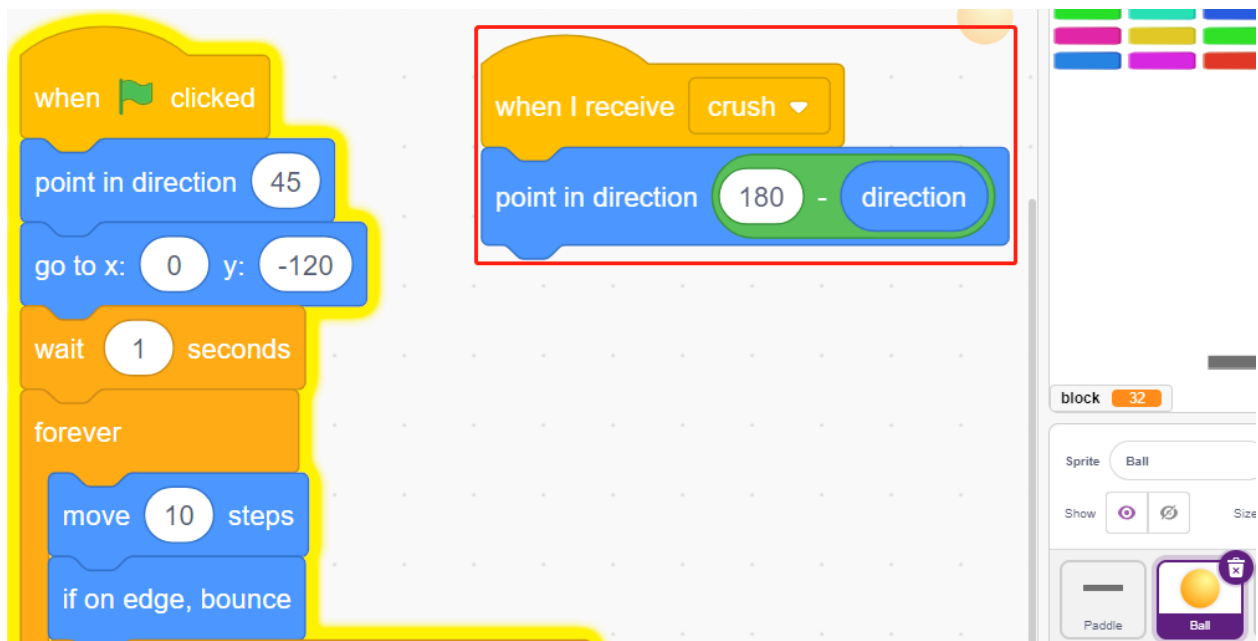
- Nachdem das Skript geschrieben wurde, klicken Sie auf die grüne Flagge und schauen Sie sich die Anzeige auf der Bühne an. Wenn es zu kompakt oder zu klein ist, können Sie die Größe ändern.



- Schreiben Sie nun das Auslöseereignis. Wenn der geklonte **Block1**-Sprite den **Ball**-Sprite berührt, löschen Sie den Klon und senden Sie die Nachricht **crush**.



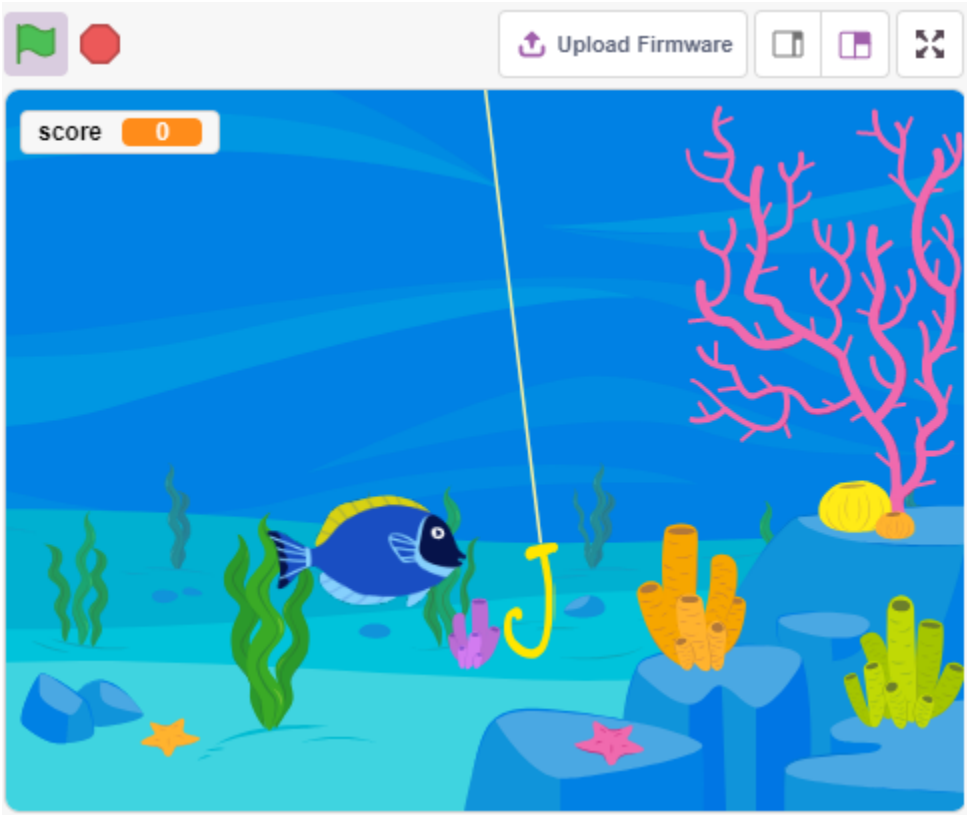
- Gehen Sie zurück zum **Ball**-Sprite. Wenn die Übertragung **crush** empfangen wird (der **Ball**-Sprite berührt den Klon des **Block1**-Sprites), wird der **Ball** aus der entgegengesetzten Richtung abgeschossen.



8.22 2.19 SPIEL - Angeln

Hier spielen wir ein Angelspiel mit einem Knopf.

Wenn das Skript läuft, schwimmen die Fische links und rechts auf der Bühne. Du solltest den Knopf drücken, wenn der Fisch fast nahe am Haken ist (es wird empfohlen, länger zu drücken), um den Fisch zu fangen. Die Anzahl der gefangenen Fische wird automatisch erfasst.



8.22.1 Benötigte Komponenten

Für dieses Projekt benötigen wir die folgenden Komponenten.

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

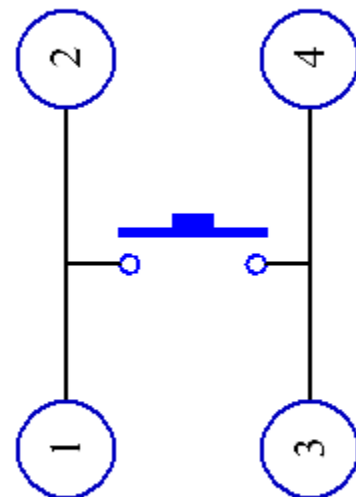
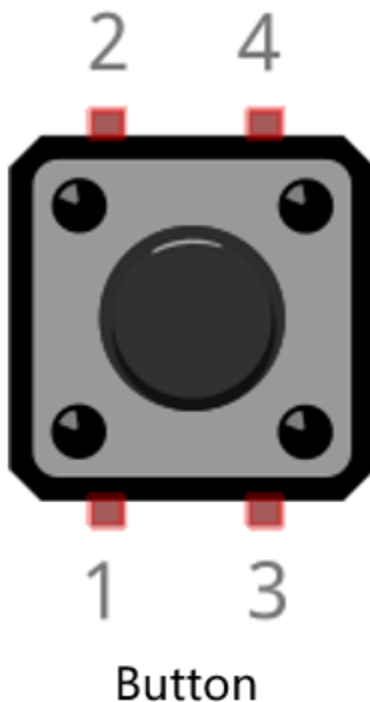
Name	ARTIKEL IN DIESEM KIT	LINK
3 in 1 Starter Kit	380+	

Du kannst sie auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Widerstand</i>	
<i>Kondensator</i>	
<i>Knopf</i>	

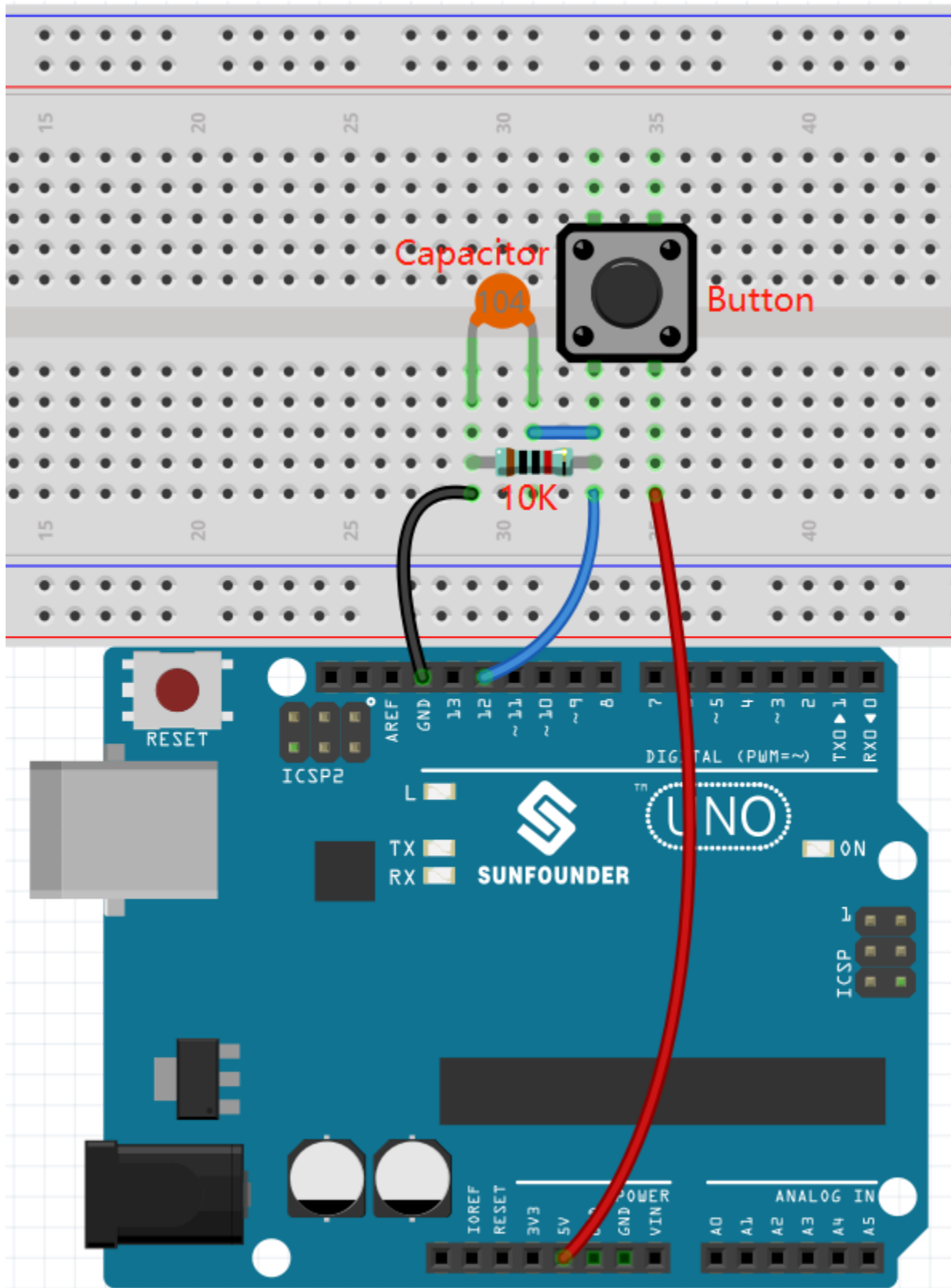
8.22.2 Schaltung aufbauen

Der Knopf ist ein 4-Pin-Gerät. Da Pin 1 mit Pin 2 und Pin 3 mit Pin 4 verbunden ist, werden beim Drücken des Knopfes alle 4 Pins miteinander verbunden, wodurch der Stromkreis geschlossen wird.



Baue die Schaltung gemäß dem folgenden Diagramm auf.

- Verbinde einen der Pins auf der linken Seite des Knopfes mit Pin 12, der mit einem Pull-Down-Widerstand und einem 0,1µF (104) Kondensator verbunden ist (um Zittern zu eliminieren und ein stabiles Signal zu liefern, wenn der Knopf betätigt wird).
- Verbinde das andere Ende des Widerstands und des Kondensators mit GND und einen der Pins auf der rechten Seite des Knopfes mit 5V.

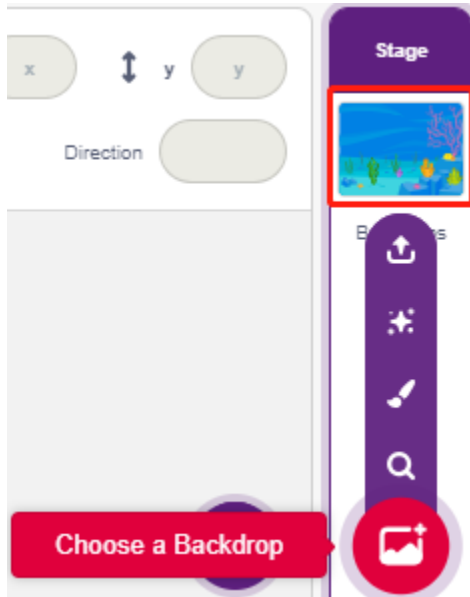


8.22.3 Programmierung

Zuerst müssen wir einen **Underwater**-Hintergrund auswählen, dann ein **Fish**-Sprite hinzufügen und es auf der Bühne hin und her schwimmen lassen. Zeichne dann ein **Fishhook**-Sprite und steuere es mit einem Knopf, um mit dem Angeln zu beginnen. Wenn das **Fisch**-Sprite den **Fishhook**-Sprite im gehakten Zustand (wird rot) berührt, wird es gehakt.

1. Hintergrund hinzufügen

Nutze die **Choose a Backdrop**-Schaltfläche, um einen **Underwater**-Hintergrund hinzuzufügen.

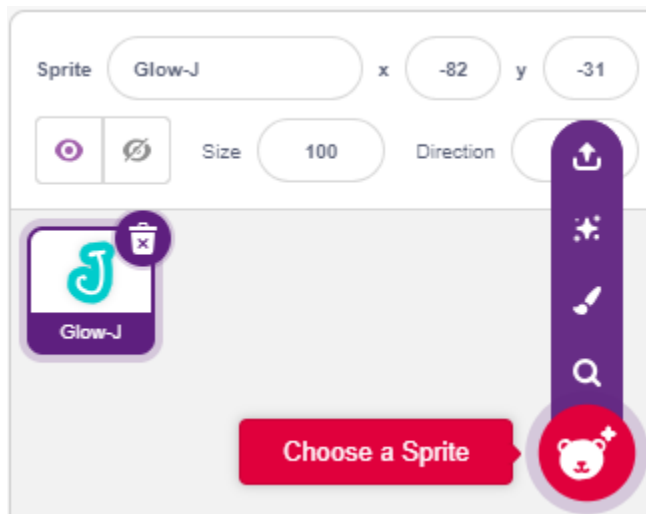


2. Angelhaken-Sprite

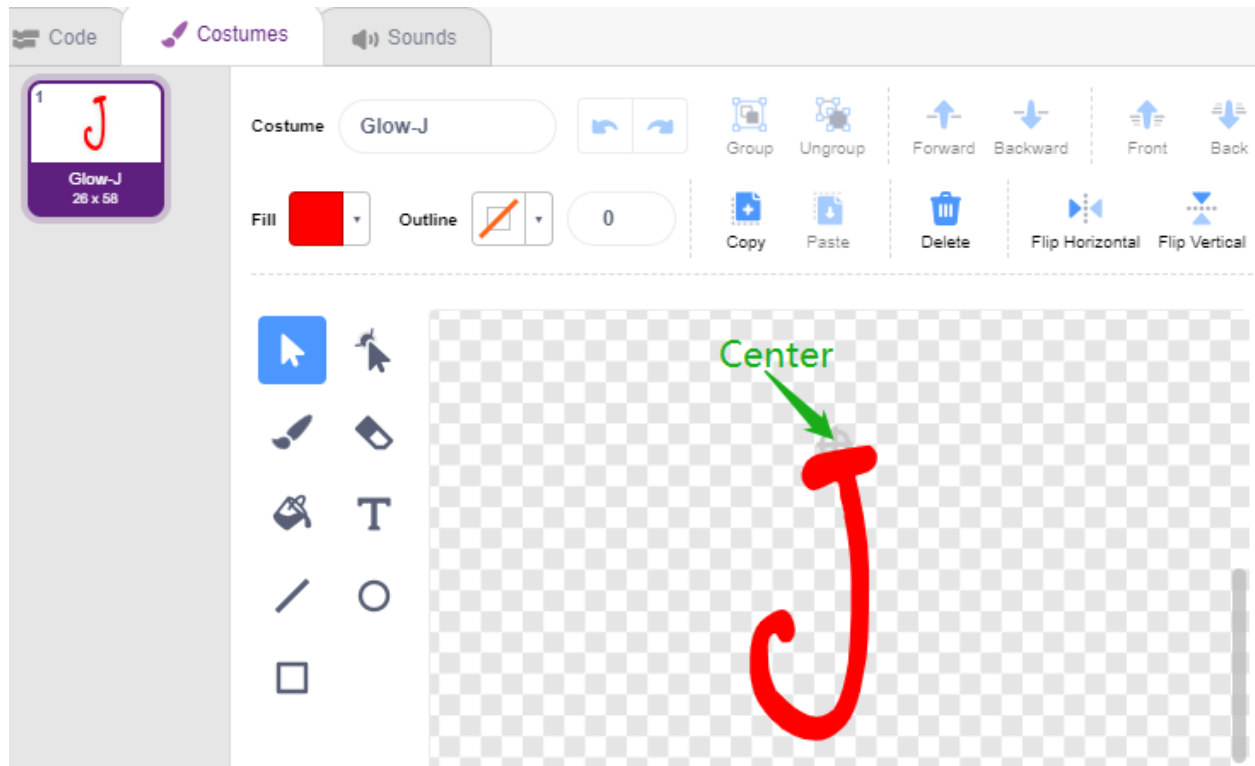
Der **Fishhook**-Sprite bleibt normalerweise im gelben Zustand unter Wasser; wenn der Knopf gedrückt wird, befindet er sich im Angelzustand (rot) und bewegt sich über die Bühne.

In Pictoblox gibt es keinen **Fishhook**-Sprite, wir können den **Glow-J**-Sprite so modifizieren, dass er wie ein Angelhaken aussieht.

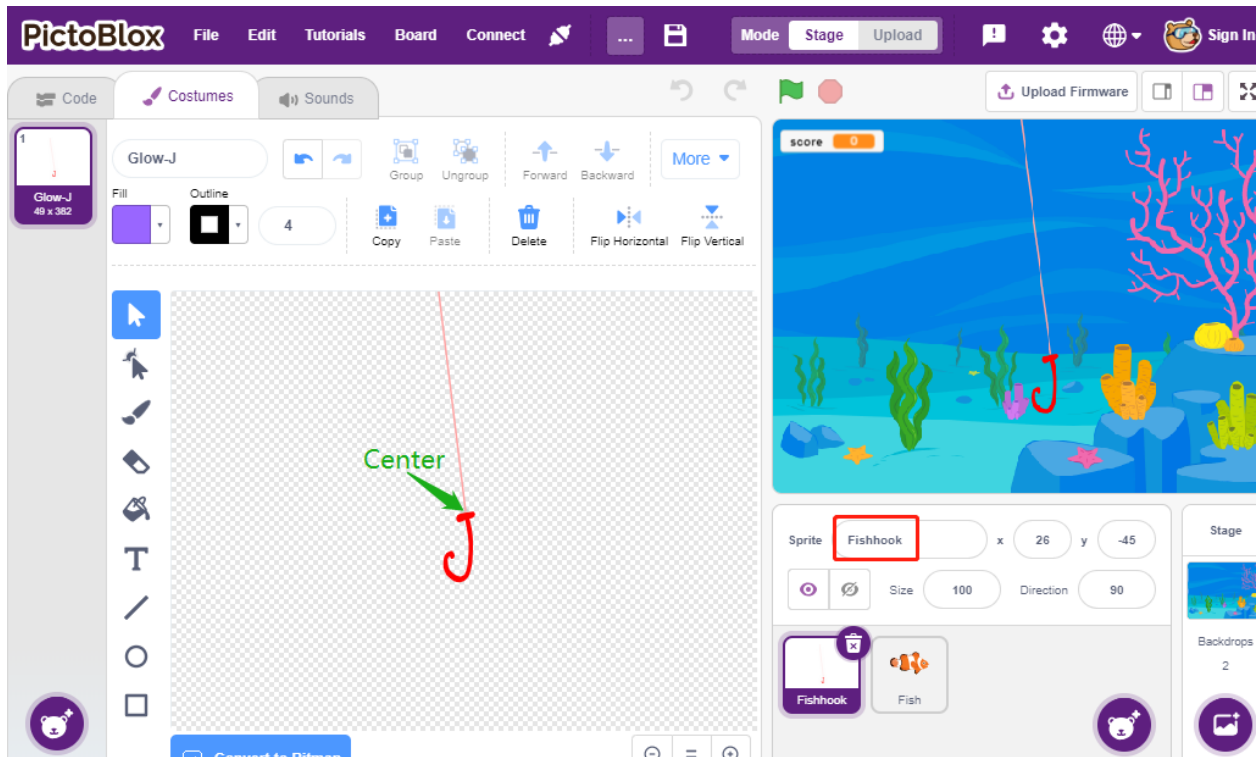
- Füge den **Glow-J**-Sprite über **Choose a Sprite** hinzu.



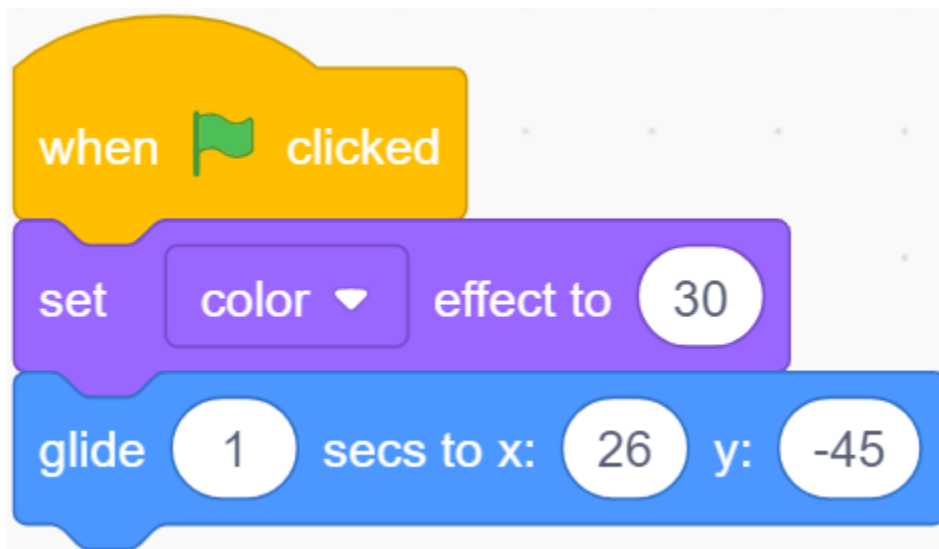
- Gehe jetzt zur **Costumes**-Seite des **Glow-J**-Sprites, wähle die Füllung von Cyan auf dem Bildschirm aus und entferne sie. Ändere dann die Farbe von J in Rot und verringere auch seine Breite. Der wichtigste Punkt ist, dass du die Spitze genau am Mittelpunkt haben musst.



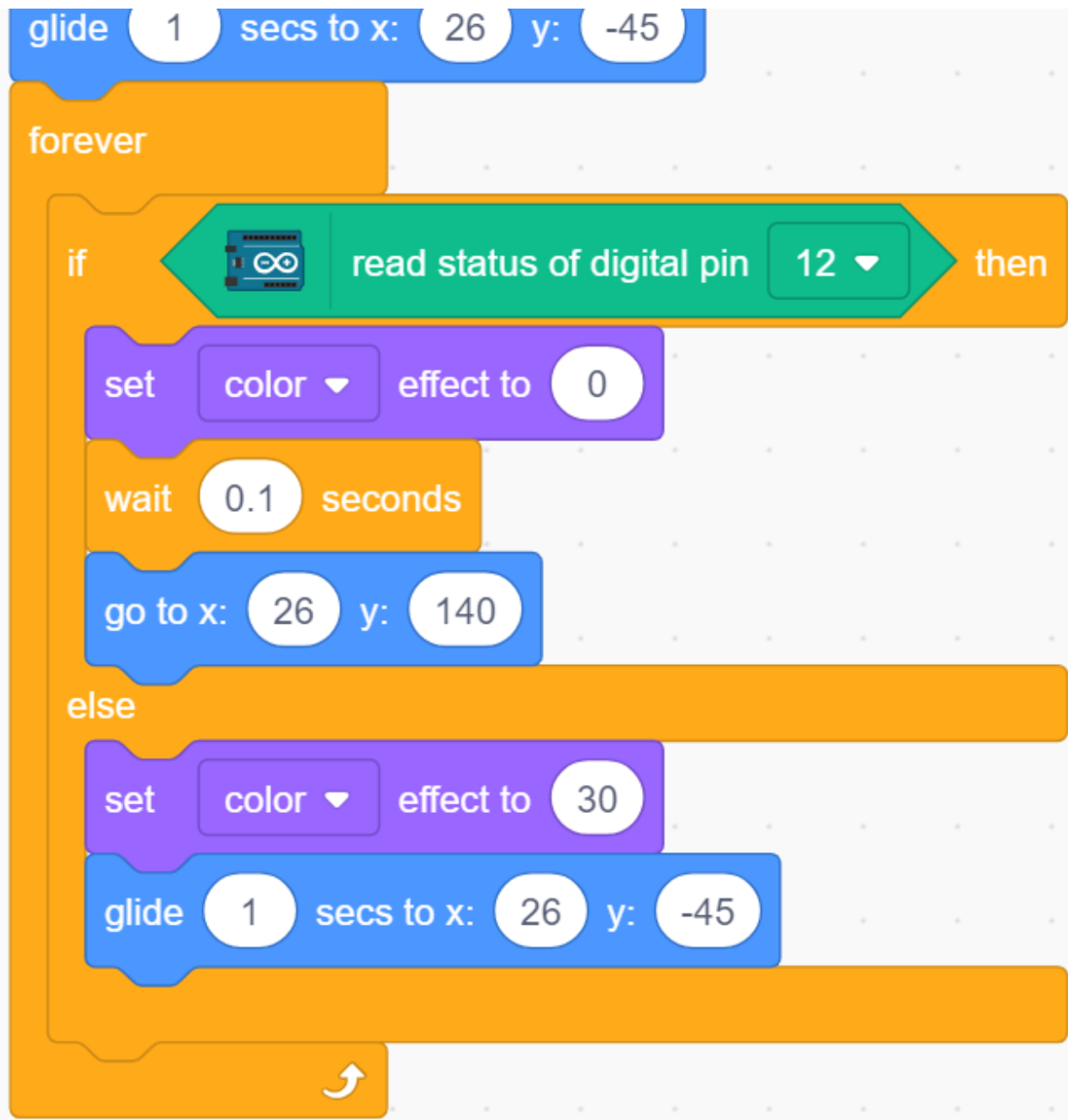
- Verwende das **Line tool**, um eine möglichst lange Linie vom Mittelpunkt nach oben zu zeichnen (Linie außerhalb der Bühne). Nachdem der Sprite gezeichnet wurde, benenne den Sprite in **Fishhook** um und verschiebe ihn an die richtige Position.



- Wenn die grüne Flagge angeklickt wird, setze den Farbeffekt des Sprites auf 30 (gelb) und setze seine Anfangsposition.



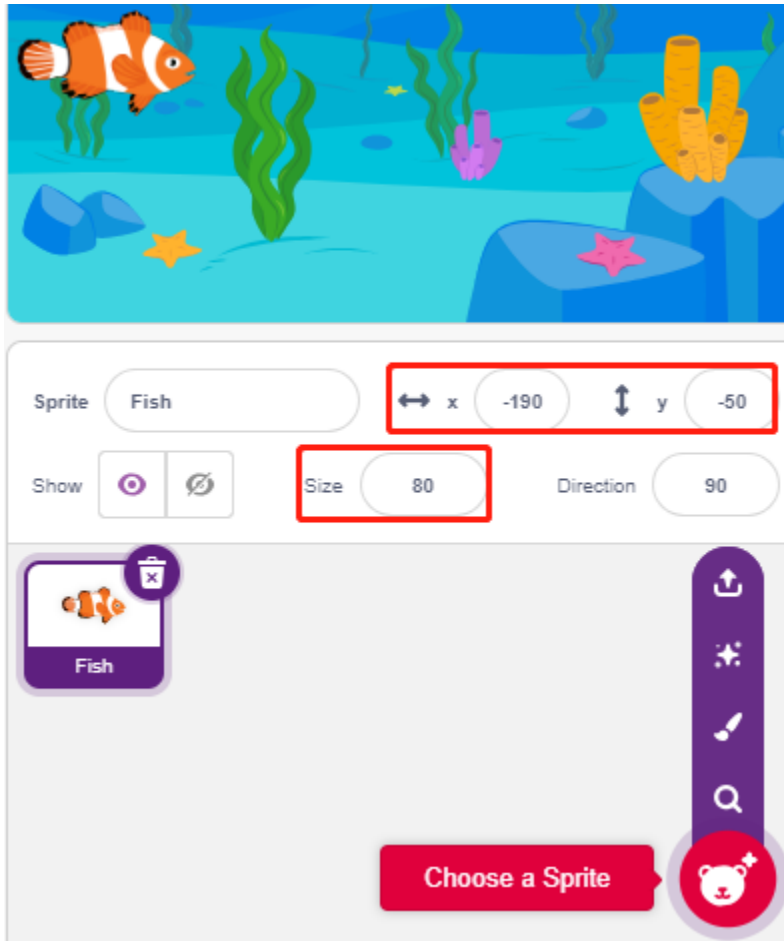
- Wenn der Knopf gedrückt wird, setze den Farbeffekt auf 0 (rot, Angelstartzustand), warte 0,1 Sekunden und bewege den **Fishhoo**-Sprite dann nach oben auf die Bühne. Lasse den Knopf los und lass den **Fishhoo** wieder in seine Anfangsposition zurückkehren.



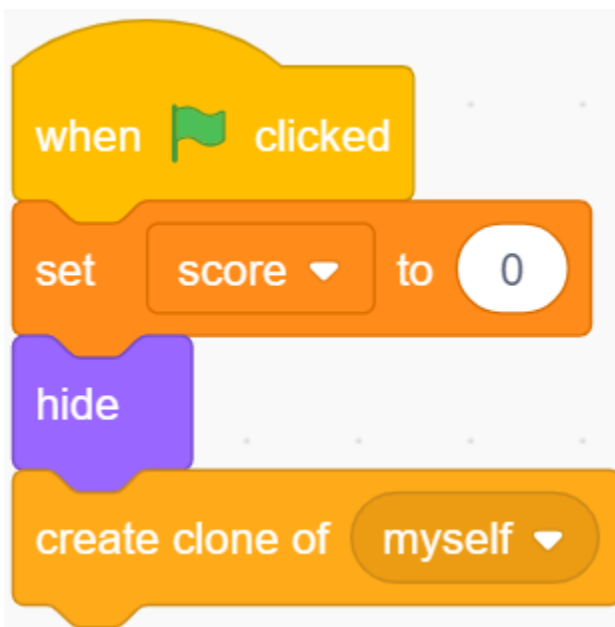
3. Fisch-Sprite

Der Effekt, den der **Fish**-Sprite erzielen soll, ist, sich links und rechts auf der Bühne zu bewegen, und wenn er einen **Fishhook**-Sprite im Angelzustand trifft, schrumpft er, bewegt sich an eine bestimmte Position und verschwindet dann und klonet anschließend einen neuen **fish**-Sprite.

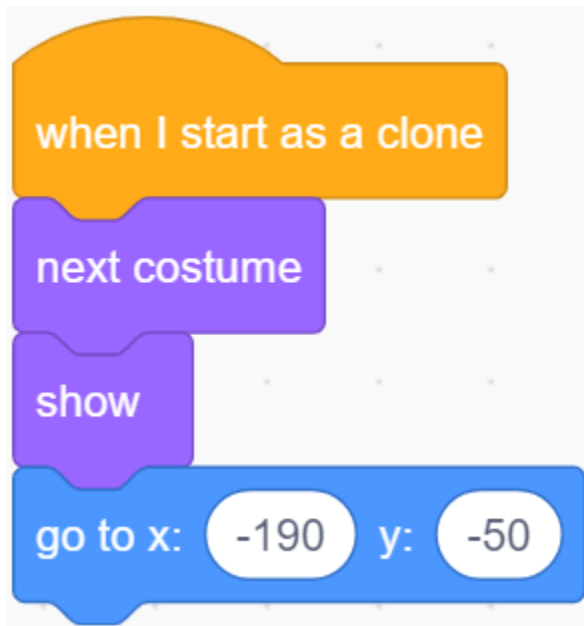
- Füge jetzt den **fish**-Sprite hinzu und passe seine Größe und Position an.



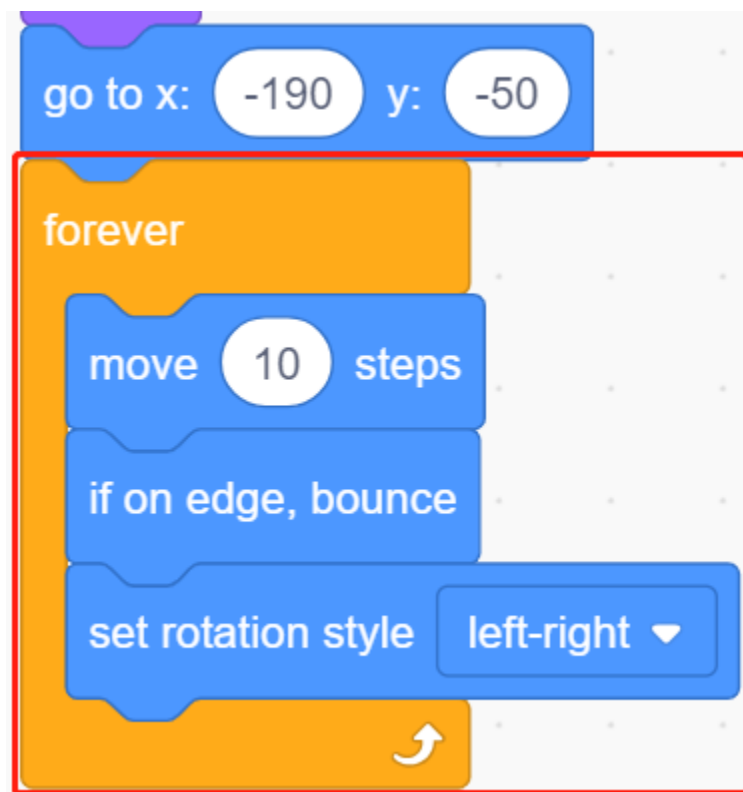
- Erstelle eine Variable **score**, um die Anzahl der gefangenen Fische zu speichern, verberge diesen Sprite und kloniere ihn.



- Zeige den Klon des **fish**-Sprites an, wechsele sein Kostüm und setze schließlich die Anfangsposition.

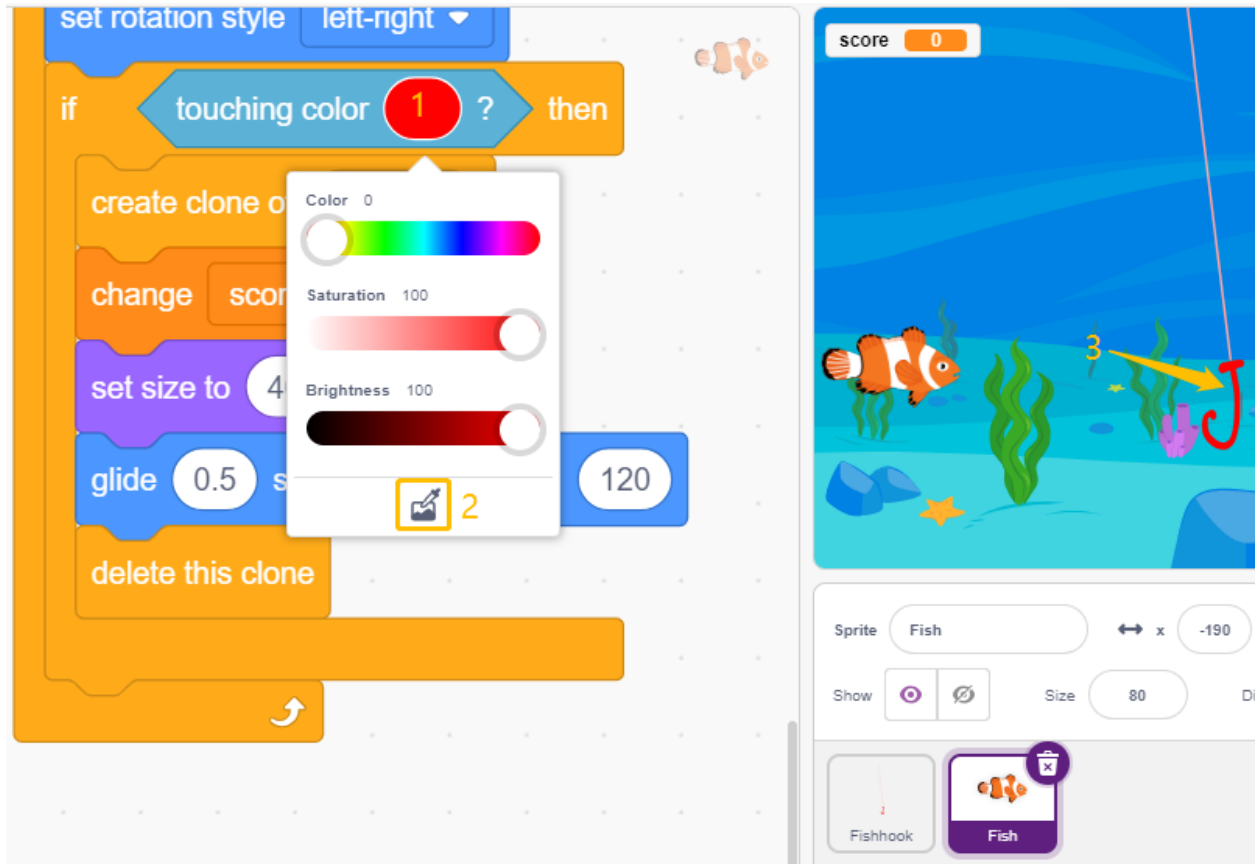


- Lasse den Klon des **fish**-Sprites sich links und rechts bewegen und beim Berühren des Randes zurückprallen.



- Der **fish**-Sprite (vom Klon) wird nicht reagieren, wenn er am **Fishhook**-Sprite vorbeigeht; berührt er jedoch den **Fishhook**-Sprite im Angelzustand (wird rot), wird er gefangen. Zu diesem Zeitpunkt wird der Punktestand (Variable Punktestand) +1 und es wird auch eine Punkteanimation angezeigt (schrumpft um 40%, bewegt sich schnell zur Position der Punkteanzeige und verschwindet). Gleichzeitig wird ein neuer Fisch erstellt (ein neuer Fisch-Sprite-Klon) und das Spiel geht weiter.

Bemerkung: Du musst in den [Touch color]-Block im Farbbereich klicken und dann das Pipettenwerkzeug auswählen, um die rote Farbe des **Fishhook**-Sprites auf der Bühne aufzunehmen. Wenn du eine Farbe willkürlich wählst, wird dieser [Touch color]-Block nicht funktionieren.



8.23 2.20 SPIEL - Klicke nicht auf das weiße Feld

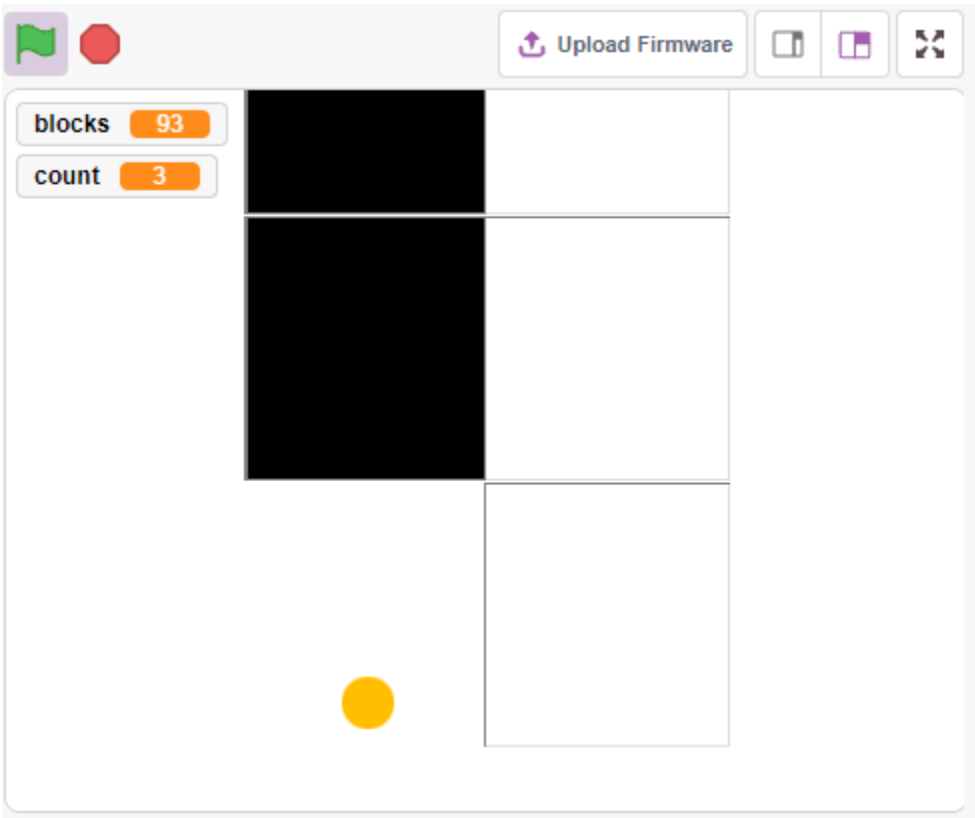
Viele von Ihnen haben dieses Spiel sicherlich schon auf Ihrem Handy gespielt. In diesem Spiel geht es darum, auf zufällig erscheinende schwarze Felder zu klicken, um Punkte zu sammeln. Die Geschwindigkeit erhöht sich stetig. Wenn man auf ein weißes Feld klickt oder ein schwarzes Feld verfehlt, ist das Spiel vorbei.

Jetzt replizieren wir es mit PictoBlox.

Setzen Sie zwei IR-Hindernisvermeidungsmodule vertikal auf das Steckbrett. Wenn Sie Ihre Hand über eines der IR-Module legen, erscheint ein blinkender Punkt auf der Bühne, der einen getätigten Klick darstellt.

Wenn Sie auf das schwarze Feld klicken, wird der Punktestand um 1 erhöht; berühren Sie das weiße Feld, wird der Punktestand um 1 verringert.

Je nach Position des schwarzen Blocks auf der Bühne müssen Sie entscheiden, ob Sie Ihre Hand über das linke oder das rechte IR-Modul legen.



8.23.1 Benötigte Komponenten

Für dieses Projekt benötigen wir folgende 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+	

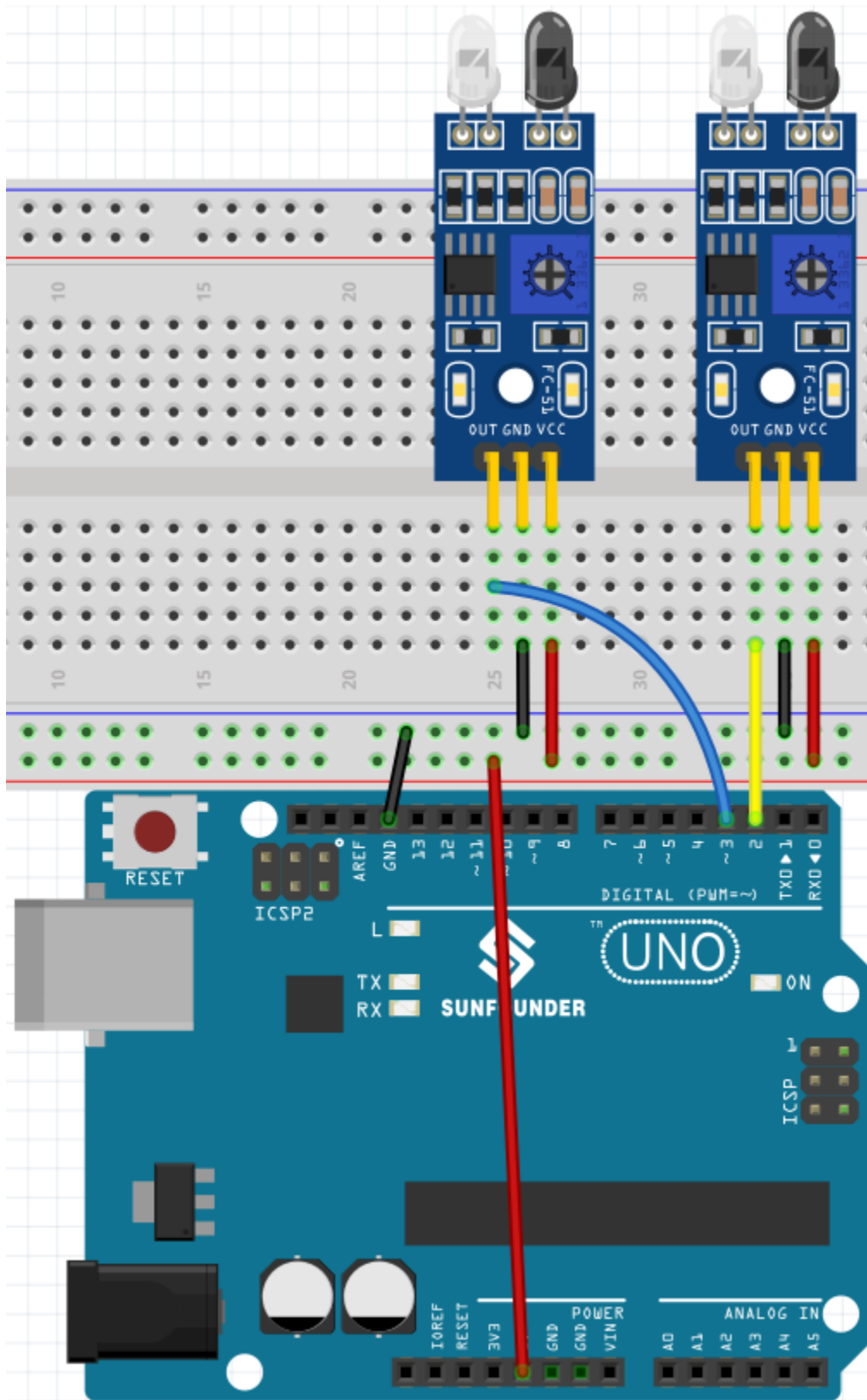
Sie können diese auch separat über die untenstehenden Links kaufen.

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Breadboard</i>	
<i>Jumper-Kabel</i>	
<i>Hindernisvermeidungsmodul</i>	

8.23.2 Schaltung aufbauen

Das Hindernisvermeidungsmodul ist ein infrarotbasierter Näherungssensor mit einstellbarer Entfernung, dessen Ausgang normalerweise hoch ist und niedrig wird, wenn ein Hindernis erkannt wird.

Bauen Sie die Schaltung entsprechend dem untenstehenden Diagramm.



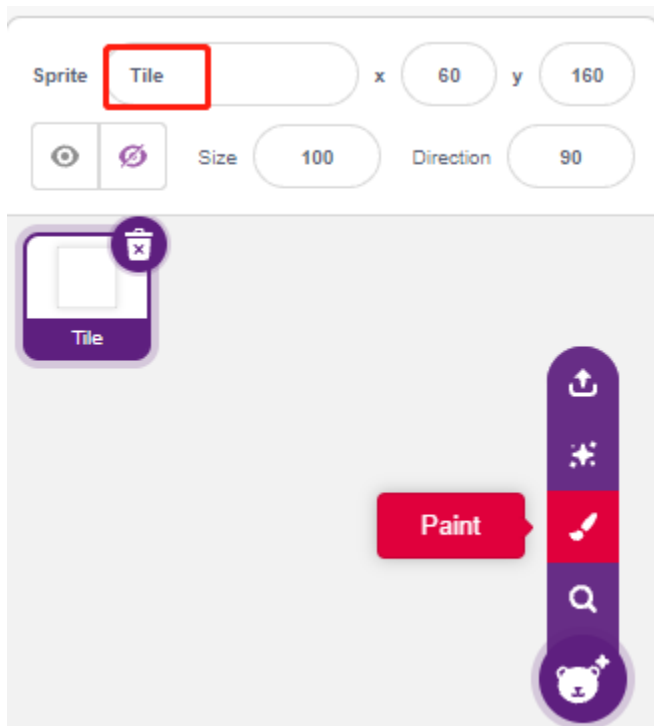
8.23.3 Programmierung

Hier benötigen wir 3 Sprites: **Tile**, **Left IR** und **Right IR**.

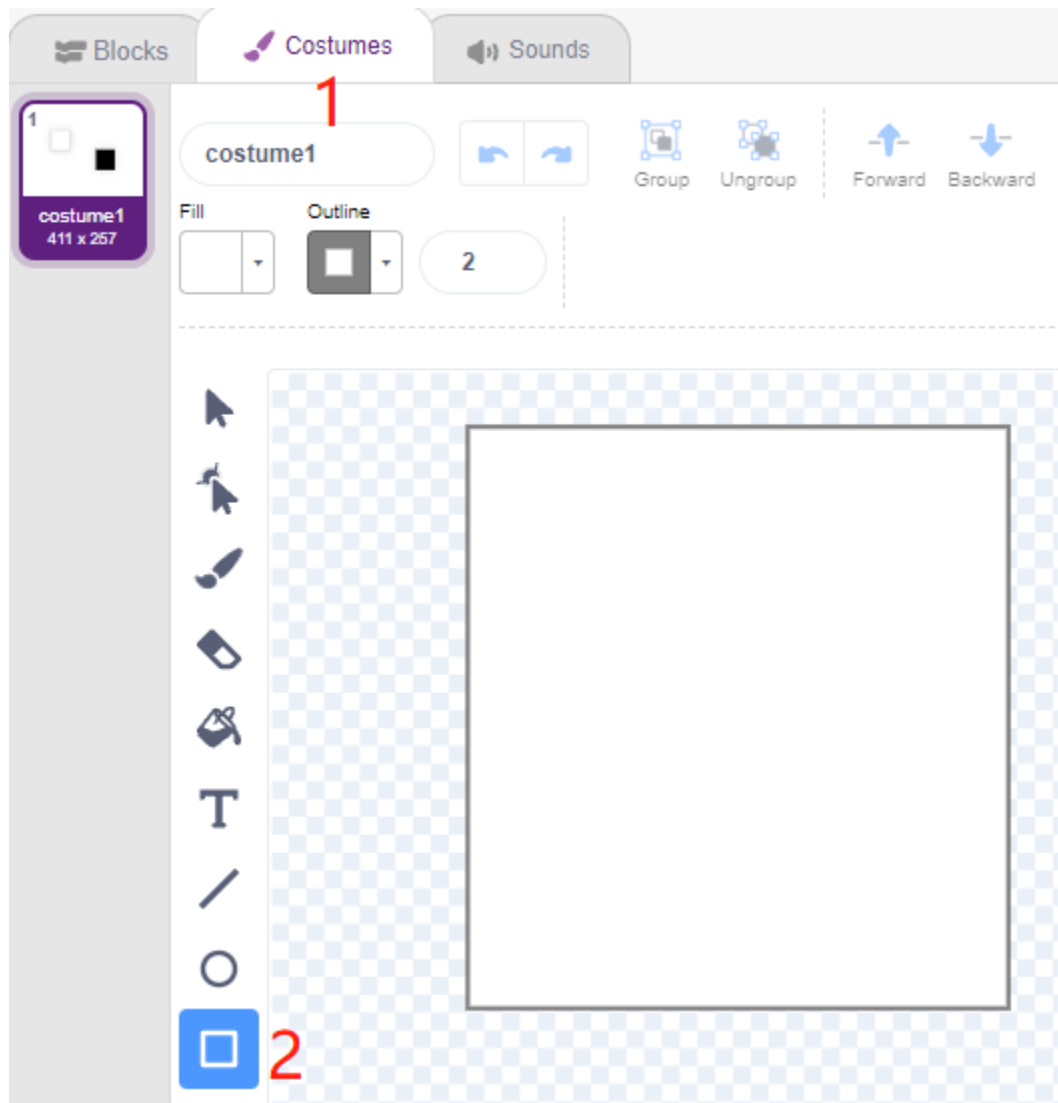
- **Tile-Sprite**: Dient zur Darstellung von abwechselnd schwarzen und weißen Feldern, die nach unten gehen. In der Handy-Version dieses Spiels gibt es normalerweise 4 Spalten, hier machen wir nur zwei.
- **Left IR-Sprite**: Dient zur Darstellung des Klick-Effekts. Wenn das linke IR-Modul Ihre Hand erkennt, sendet es eine Nachricht - **left** an den **Left IR**-Sprite, um ihn zu aktivieren. Berührt er das schwarze Feld auf der Bühne, erhöht sich der Punktestand um 1, andernfalls verringert er sich um 1.
- **Right IR-Sprite**: Die Funktion ist im Wesentlichen die gleiche wie bei **Left IR**, außer dass er die Information **Right** erhält.

1. Erstellen Sie ein Tile-Sprite.

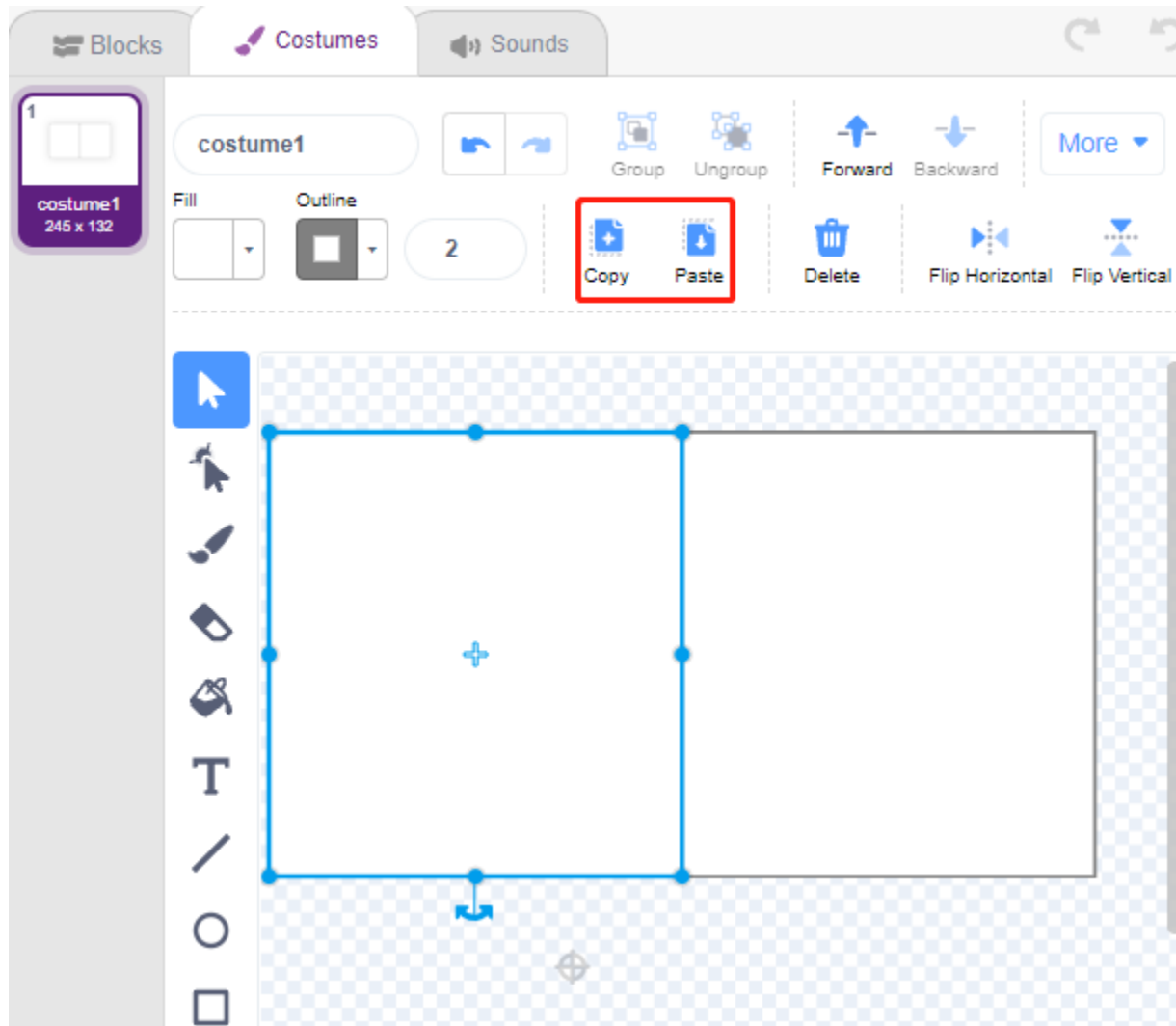
Löschen Sie das Standard-Sprite, bewegen Sie die Maus über das Symbol **Add Sprite**, wählen Sie **Paint** und ein leeres Sprite erscheint, welches Sie **Tile** nennen.



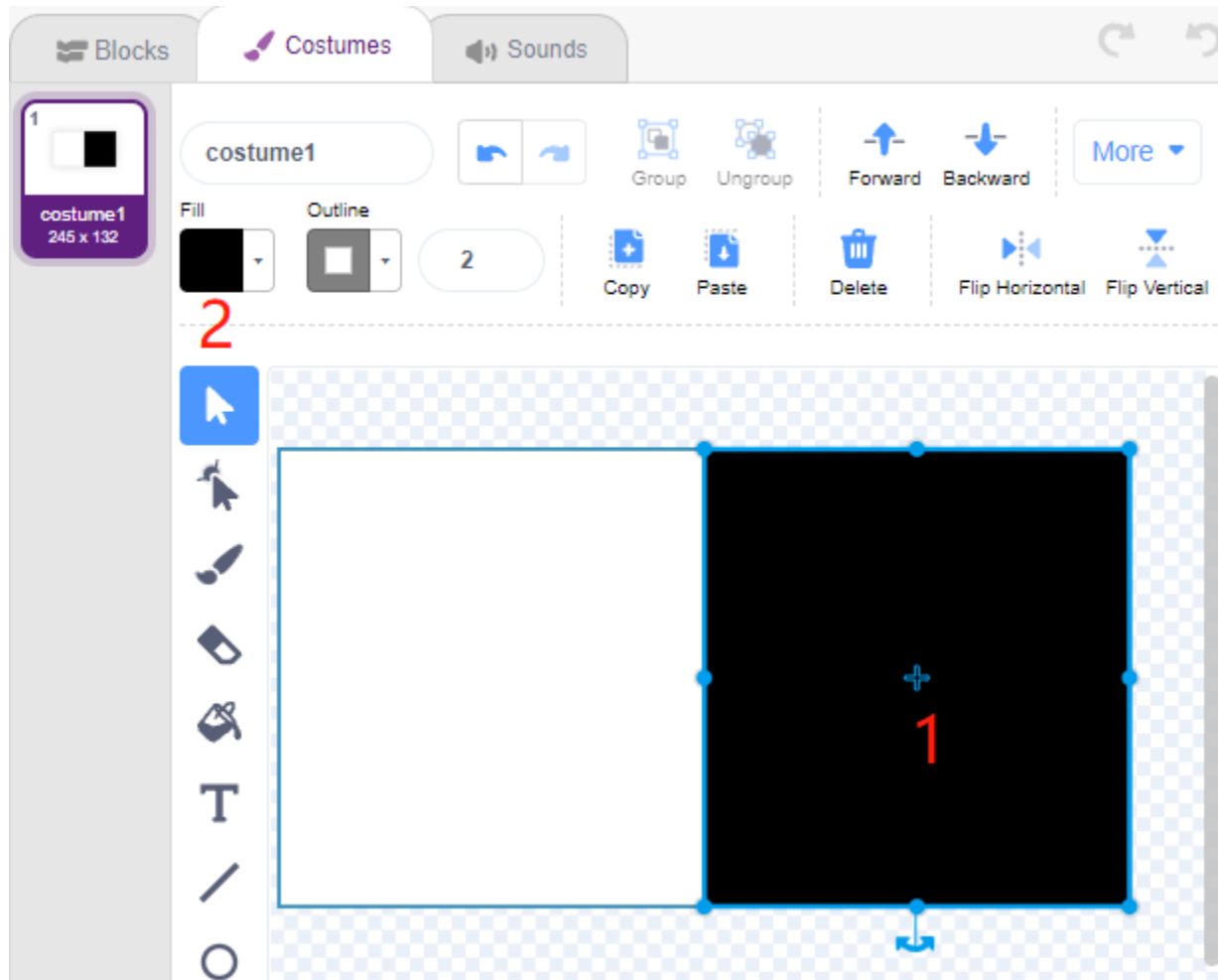
Gehen Sie zur **Costumes**-Seite und verwenden Sie das **Rectangle**-Werkzeug, um ein Rechteck zu zeichnen.



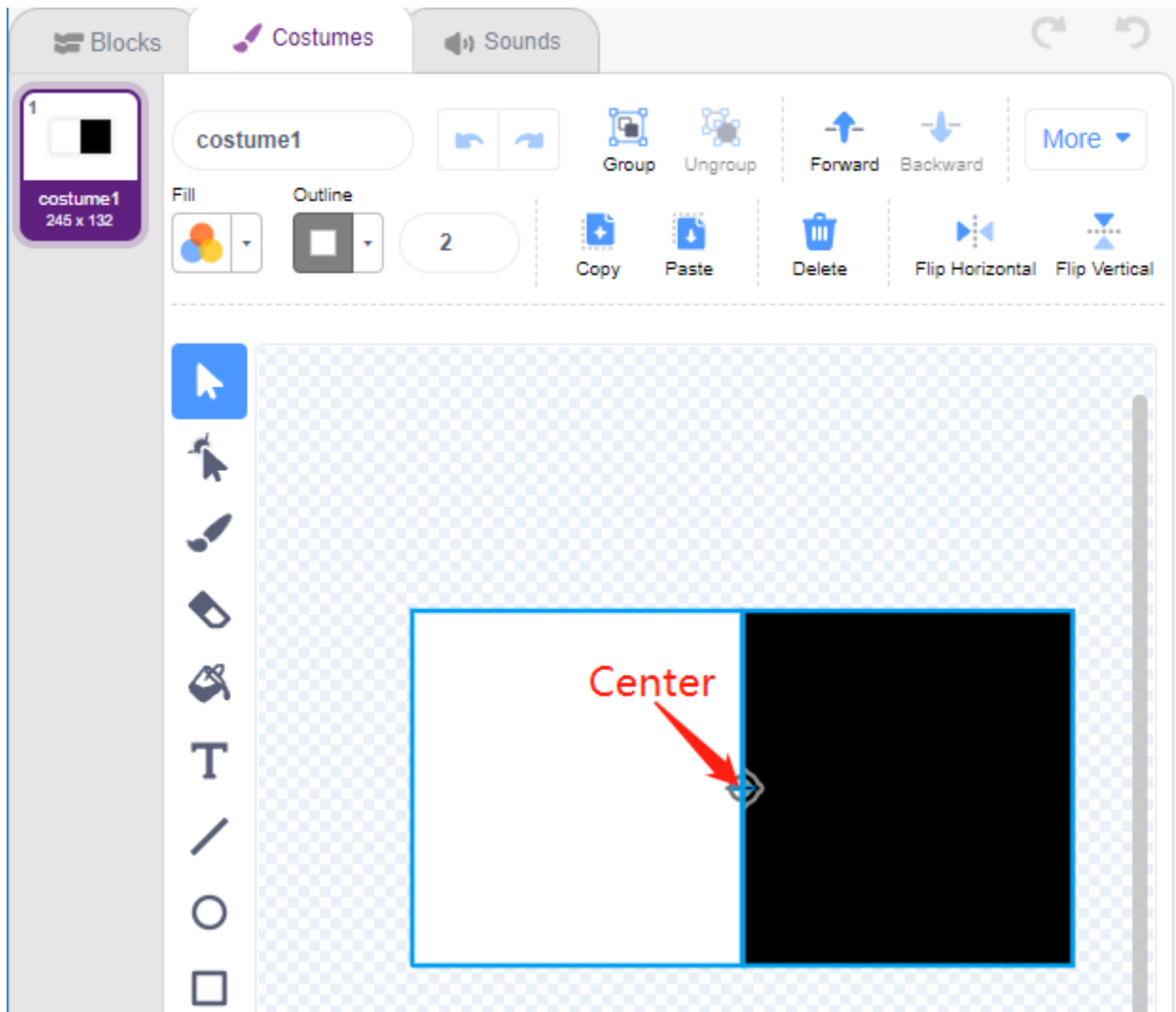
Wählen Sie das Rechteck aus und klicken Sie auf **Copy** -> **Paste**, um ein identisches Rechteck zu erstellen. Anschließend verschieben Sie die beiden Rechtecke so, dass sie bündig zueinander stehen.



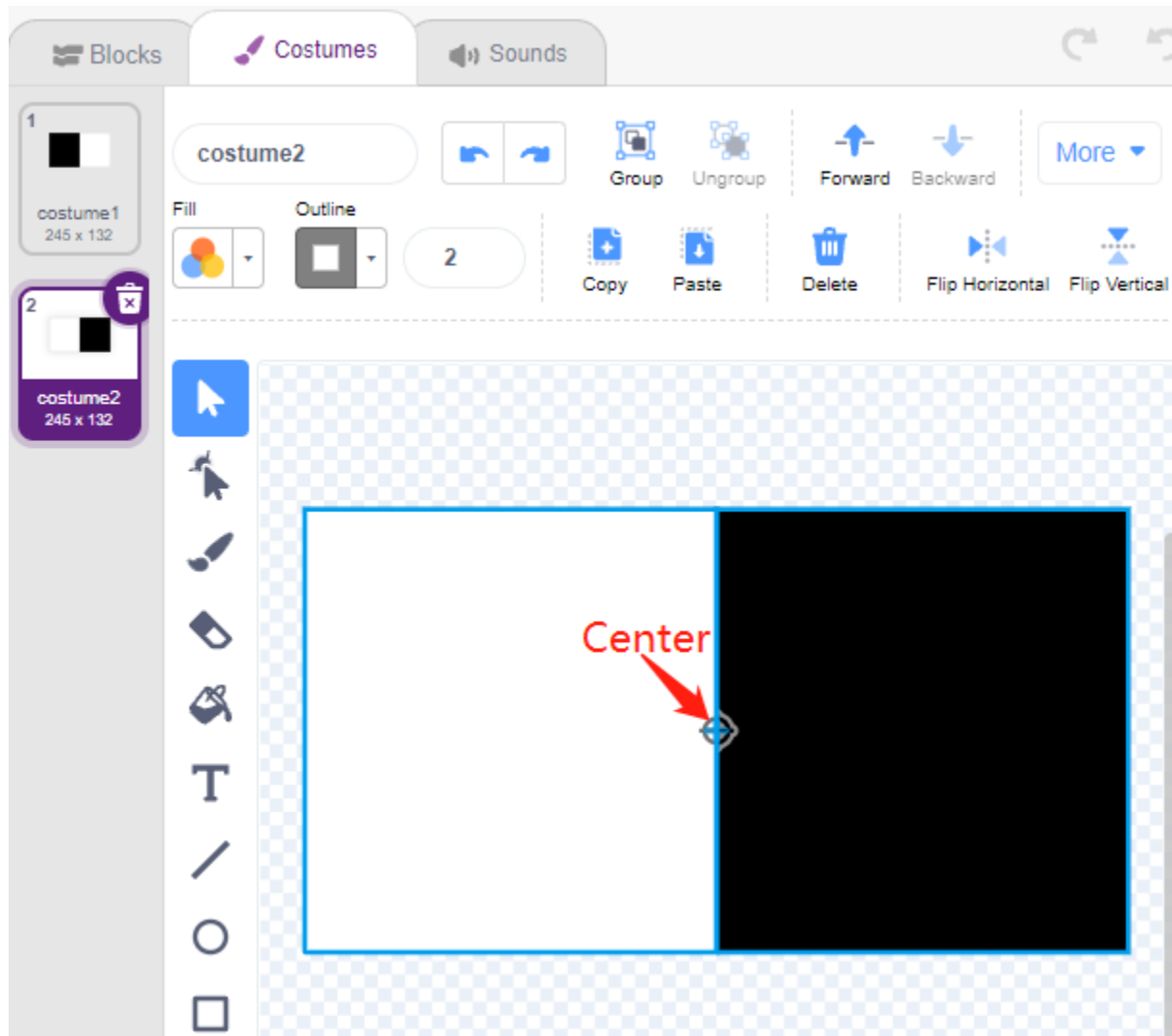
Wählen Sie eines der Rechtecke aus und wählen Sie eine schwarze Füllfarbe.



Jetzt wählen Sie beide Rechtecke aus und verschieben Sie sie so, dass ihre Mittelpunkte mit dem Zentrum der Leinwand übereinstimmen.

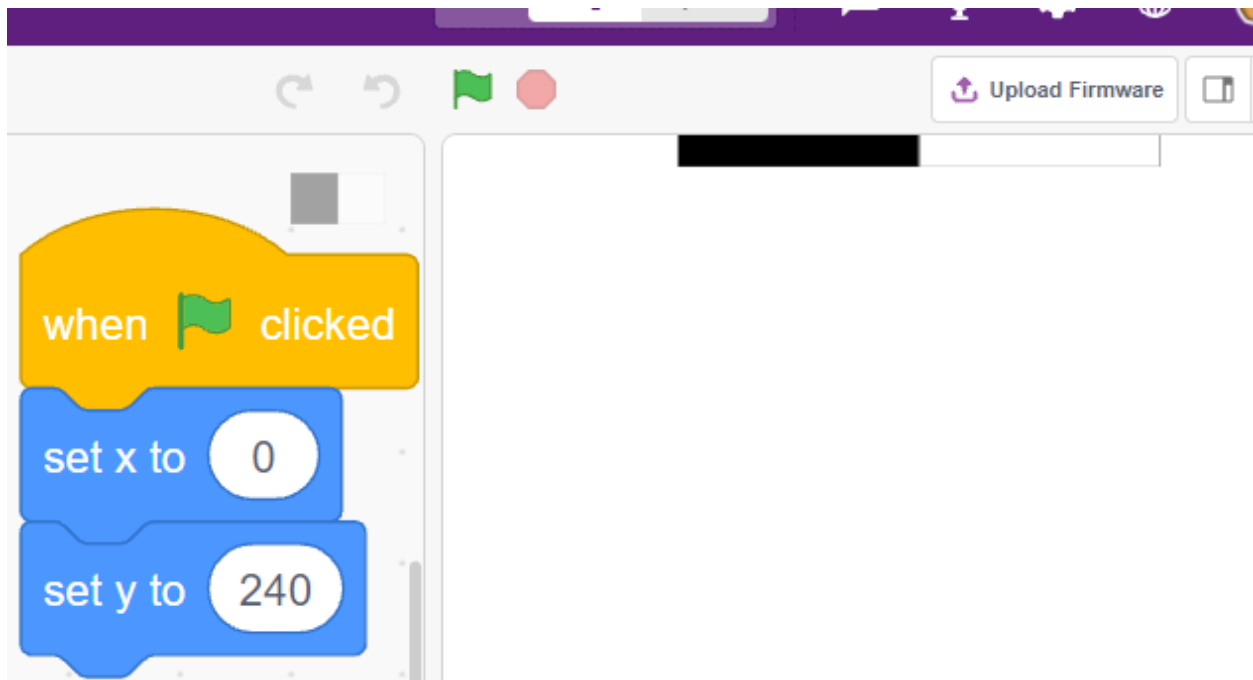


Duplizieren Sie das Kostüm1 und wechseln Sie die Füllfarben der beiden Rechtecke. Zum Beispiel hat Kostüm1 links die Füllfarbe Weiß und rechts die Füllfarbe Schwarz, bei Kostüm2 ist es umgekehrt.



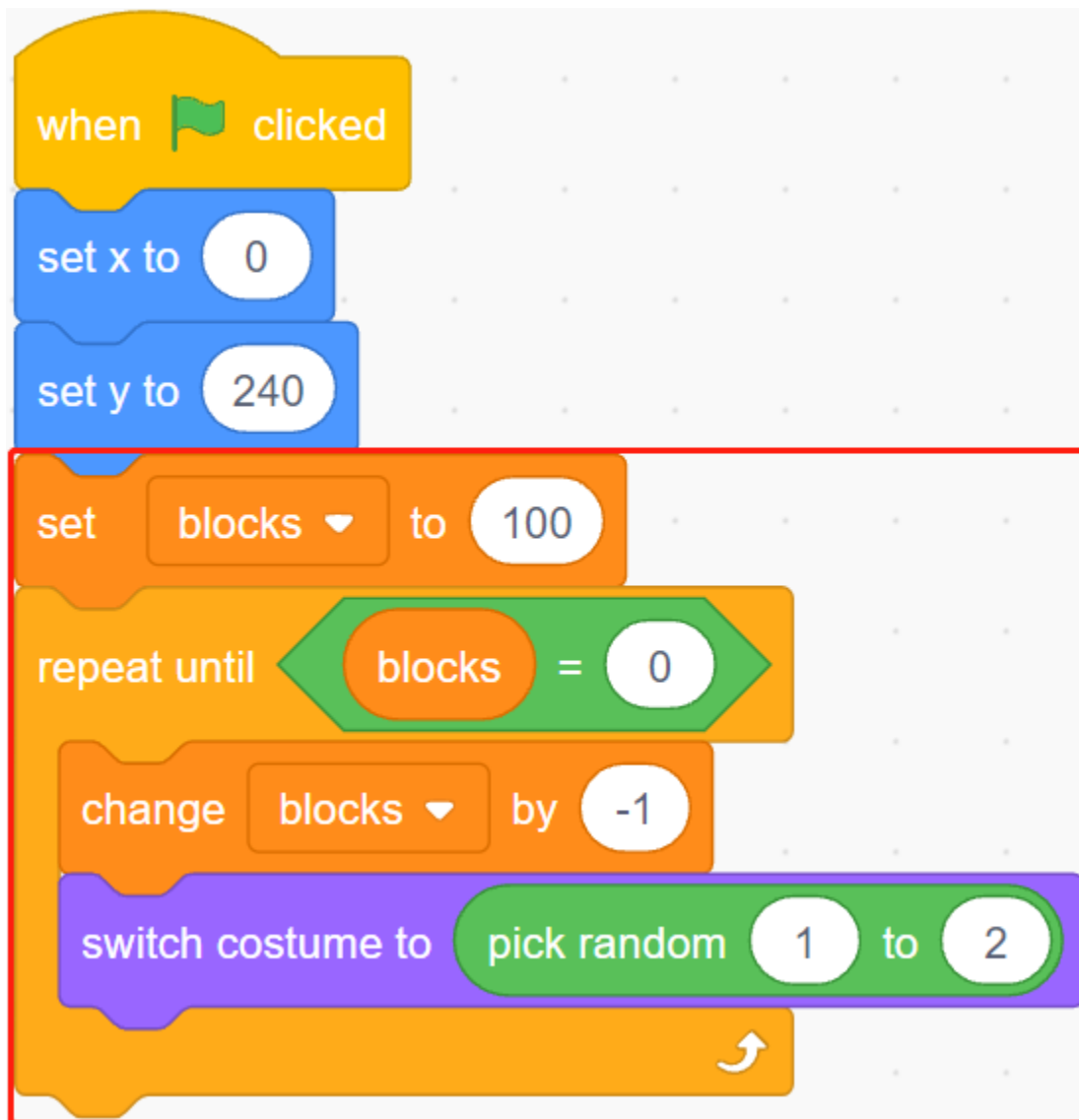
2. Das Script für den Tile-Sprite erstellen

Gehen Sie nun zurück zur **Blocks**-Seite und legen Sie die Anfangsposition des **Tile**-Sprites fest, damit dieser am oberen Bühnenrand positioniert ist.

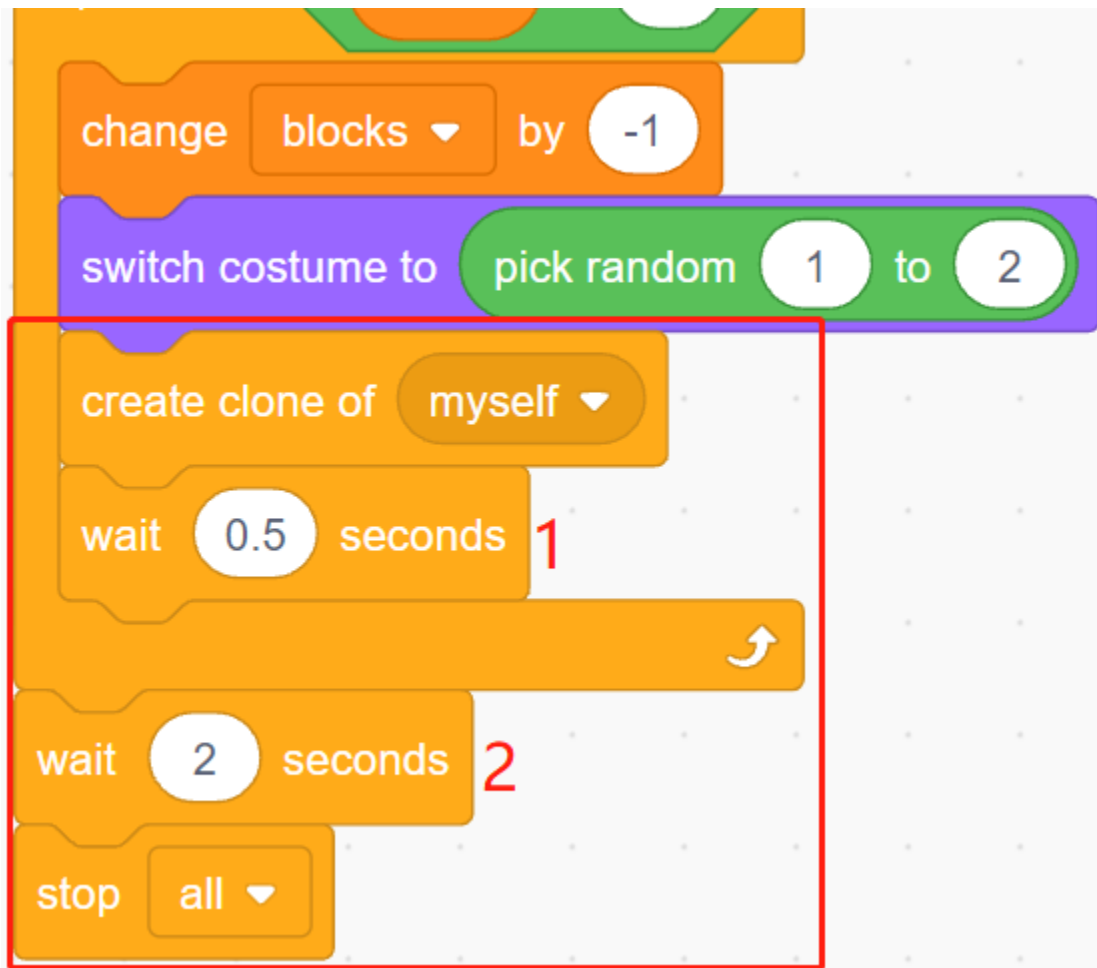


Erstellen Sie eine Variable - **blocks** und geben Sie ihr einen Startwert, um die Anzahl der Erscheinungen des **Tile**-Sprites zu bestimmen. Nutzen Sie den [Wiederhole bis]-Block, damit die Variable **blocks** stetig verringert wird, bis **blocks** 0 erreicht. Währenddessen sollte der Sprite **Tile** zufällig sein Kostüm wechseln.

Nachdem Sie auf die grüne Flagge geklickt haben, wird der **Tile**-Sprite auf der Bühne schnell die Kostüme wechseln.



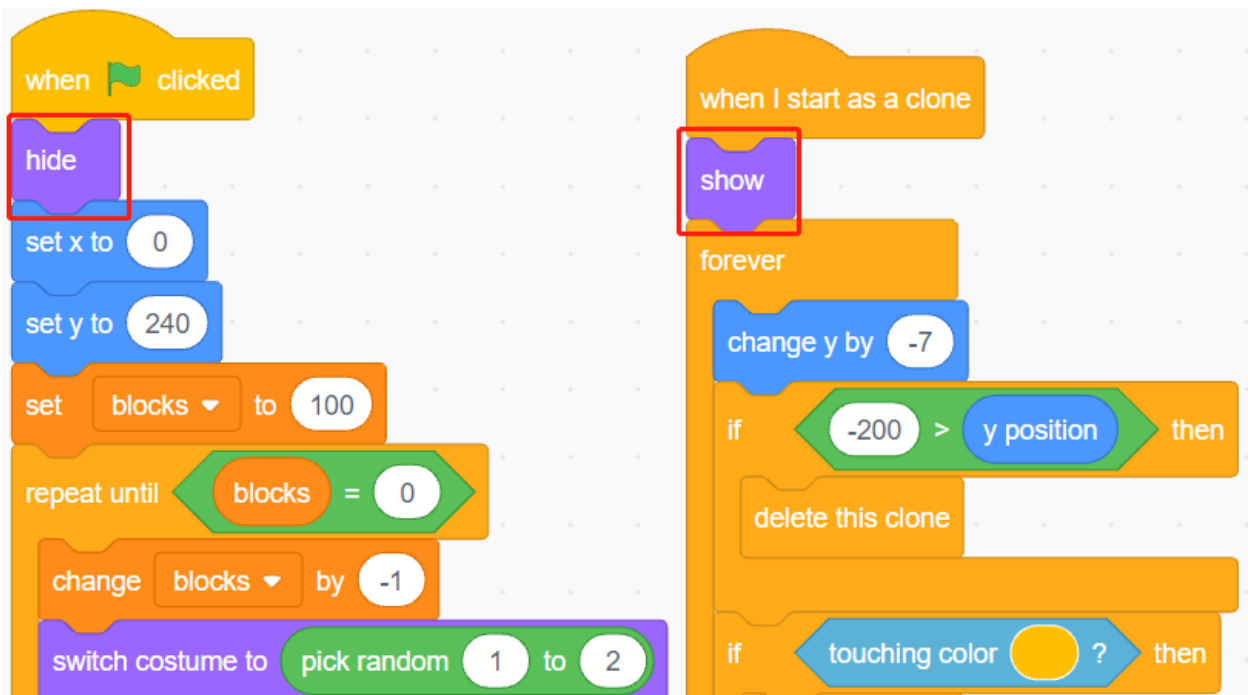
Erstellen Sie Klone des **Tile**-Sprites, während die Variable **blocks** abnimmt, und stoppen Sie das Skript, wenn blocks 0 ist. Hier werden zwei [wait () seconds]-Blöcke verwendet: Der erste begrenzt das Intervall zwischen **Tiles** Klonen und der zweite lässt die Variable blocks auf 0 sinken, ohne das Programm sofort zu stoppen, sodass dem letzten Tile-Sprite genügend Zeit zum Bewegen bleibt.



Nun soll der Klon des **Tile**-Sprites langsam nach unten bewegen und gelöscht werden, sobald er den unteren Bühnenrand erreicht. Die Änderung in der y-Koordinate beeinflusst die Fallgeschwindigkeit: Je größer der Wert, desto schneller die Geschwindigkeit.



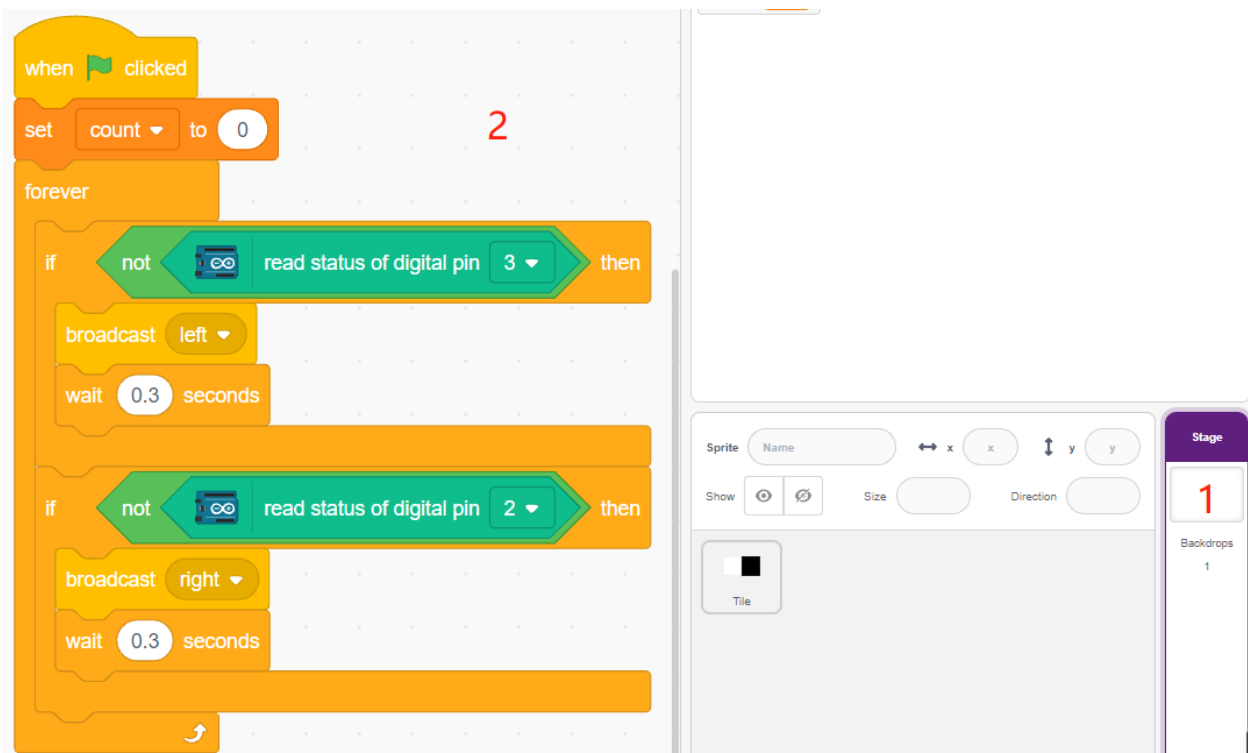
Das Original verstecken und den Klon anzeigen.



3. Die Werte der 2 IR-Module auslesen

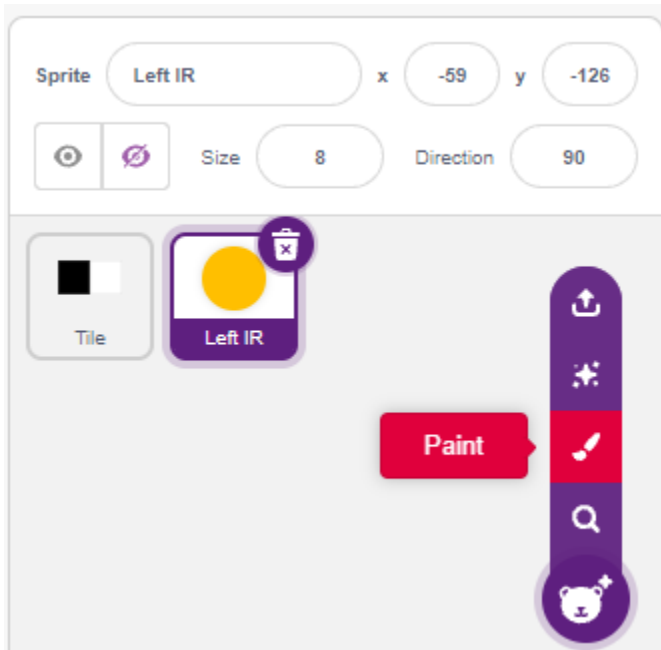
Im Hintergrund lesen Sie die Werte der 2 IR-Module aus und führen die entsprechenden Aktionen durch.

- Wenn das linke IR-Hindernisvermeidungsmodul Ihre Hand erkennt, senden Sie eine Nachricht - **left**.
- Wenn das rechte IR-Hindernisvermeidungsmodul Ihre Hand erkennt, senden Sie eine Nachricht - **right**.

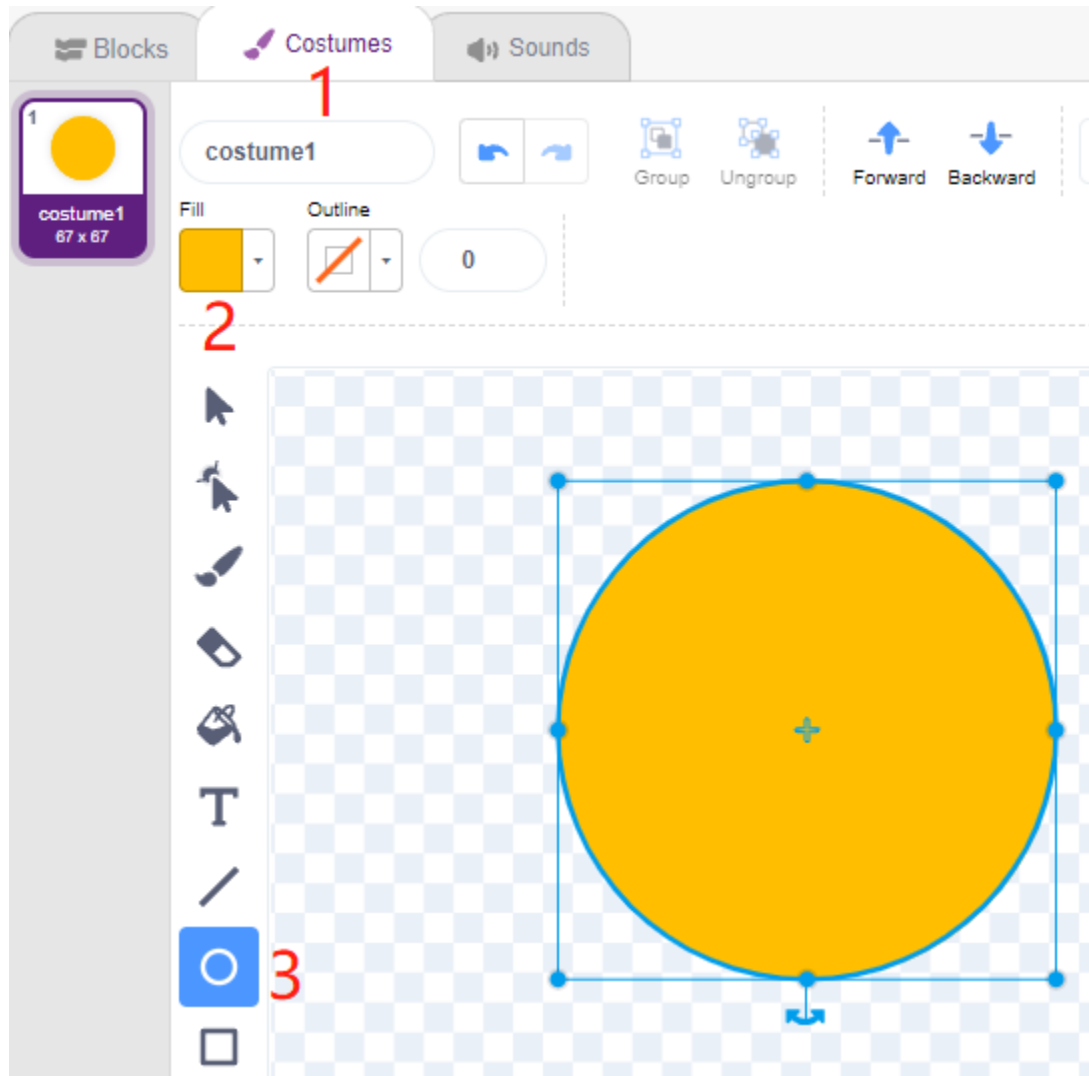


4. Left IR-Sprite

Fahren Sie erneut mit der Maus über das **Add sprite**-Symbol und wählen Sie **Paint**, um einen neuen Sprite namens **Left IR** zu erstellen.



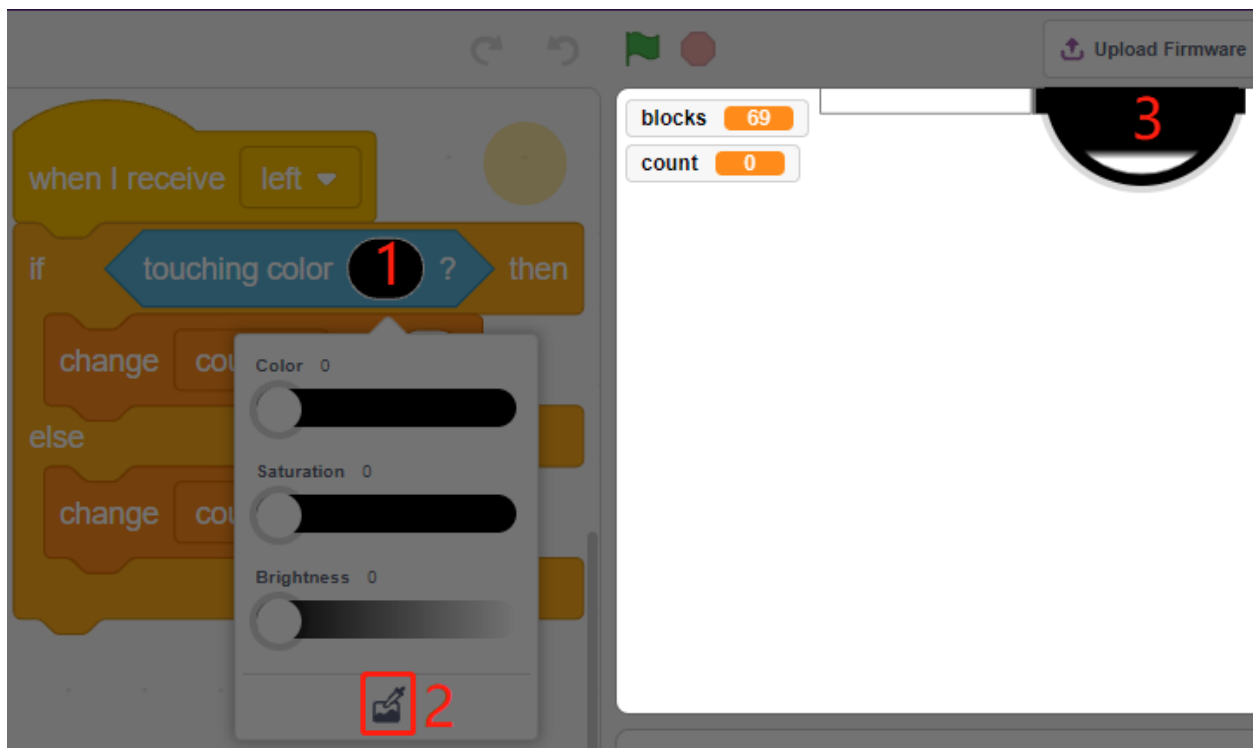
Gehen Sie zur **Costumes**-Seite des **Left IR**-Sprites, wählen Sie eine Füllfarbe (eine beliebige Farbe außer Schwarz und Weiß) und zeichnen Sie einen Kreis.



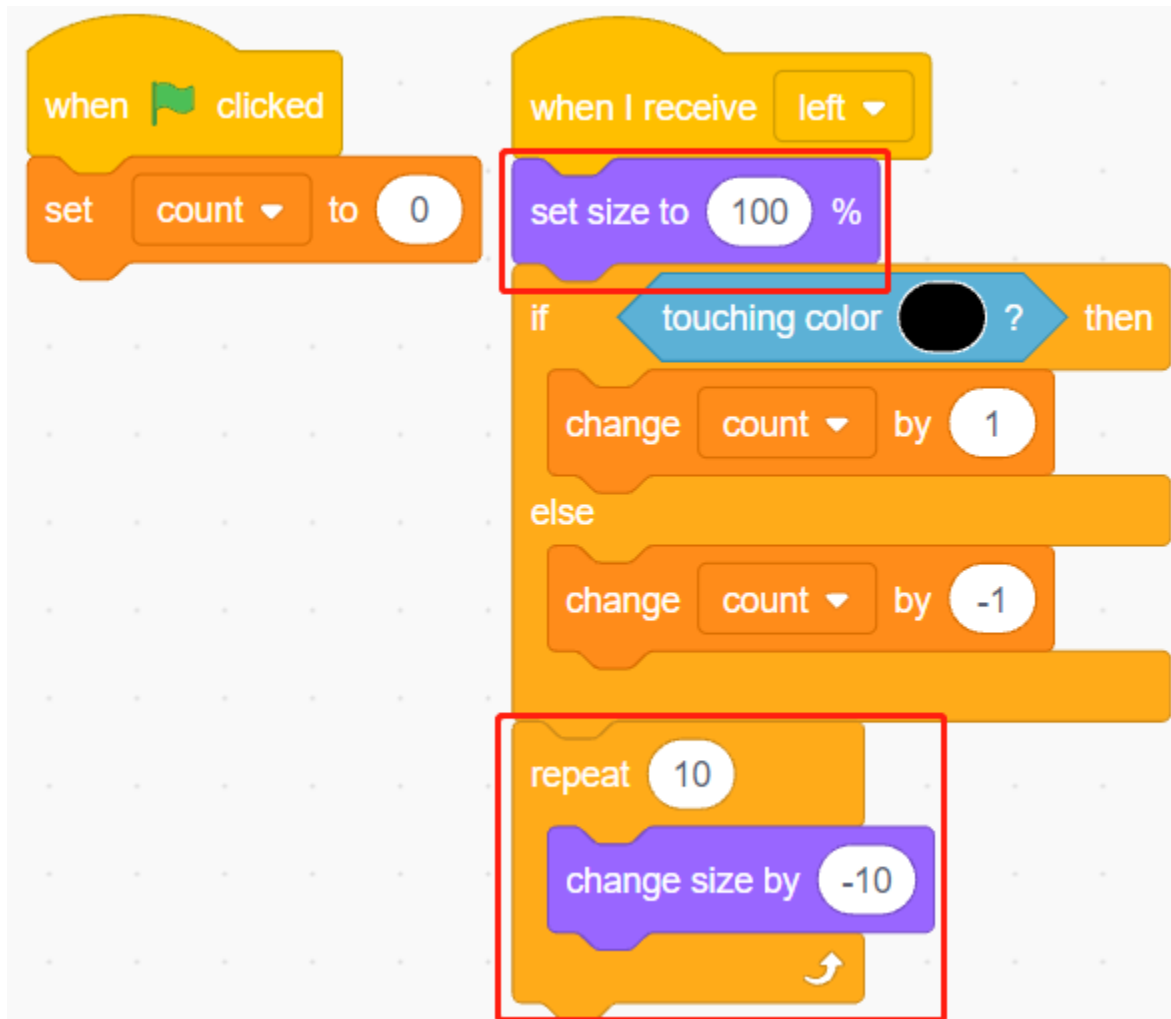
Beginnen Sie nun, den **Left IR**-Sprite zu scripten. Wenn die Nachricht - **left** empfangen wird (das IR-Empfängermodul links erkennt ein Hindernis), dann prüfen Sie, ob der schwarze Block des **Tile**-Sprites berührt wird, und wenn ja, erhöhen Sie die Variable **count** um 1, andernfalls verringern Sie sie um 1.



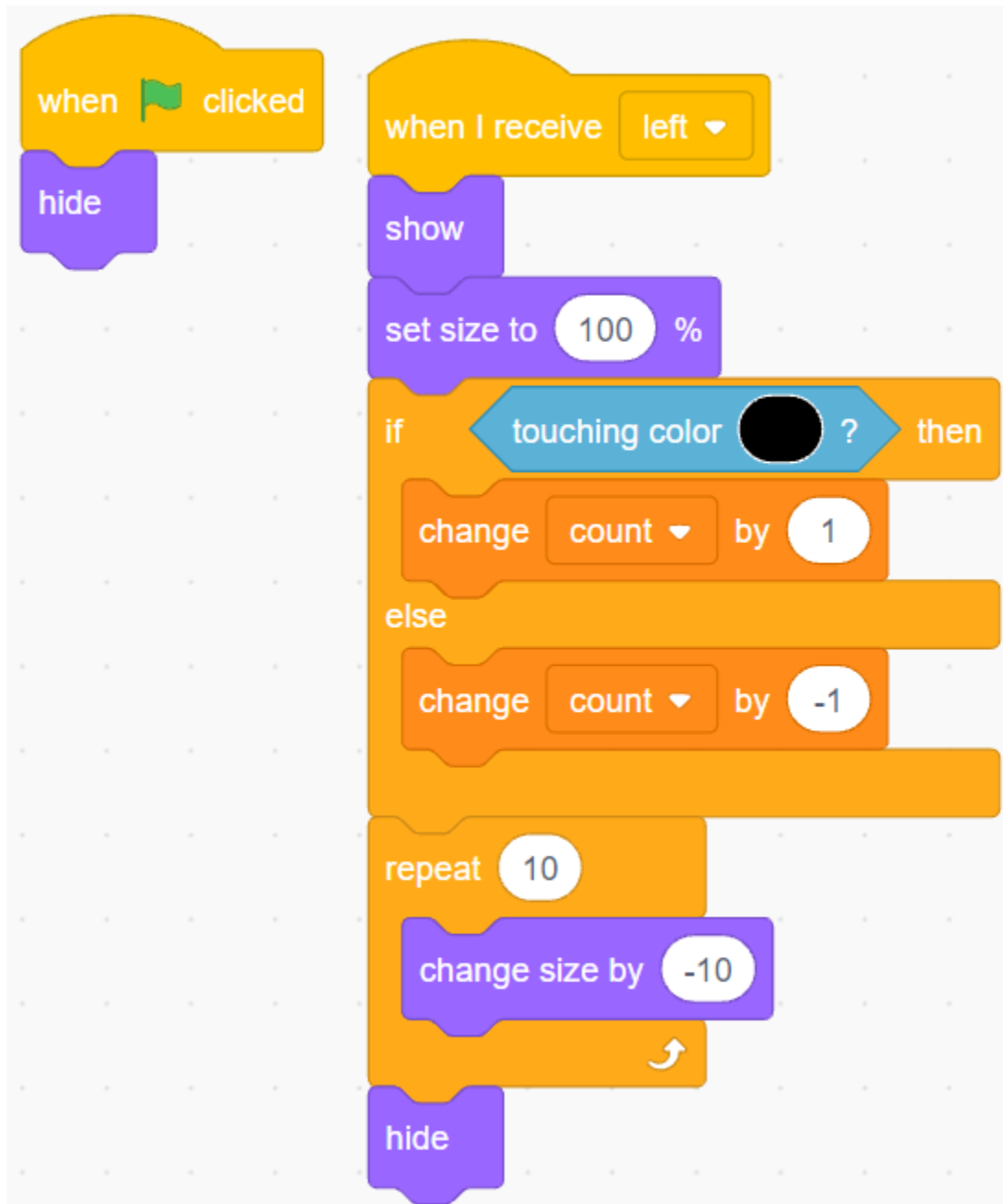
Bemerkung: Sie müssen den **Tile-Sprite** auf der Bühne anzeigen lassen und dann die Farbe des schwarzen Blocks im **Tile-Sprite** aufnehmen.



Führen Sie nun den Sensor-Effekt (Vergrößern/Verkleinern) für **Left IR** durch.

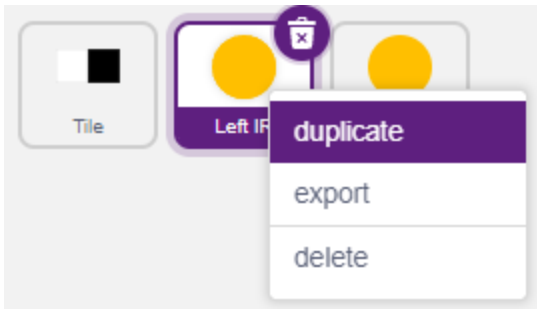


Lassen Sie den **Left IR**-Sprite verstecken, wenn auf die grüne Flagge geklickt wird, zeigen Sie ihn an, wenn die Nachricht - **left** empfangen wird, und verstecken Sie ihn schließlich wieder.

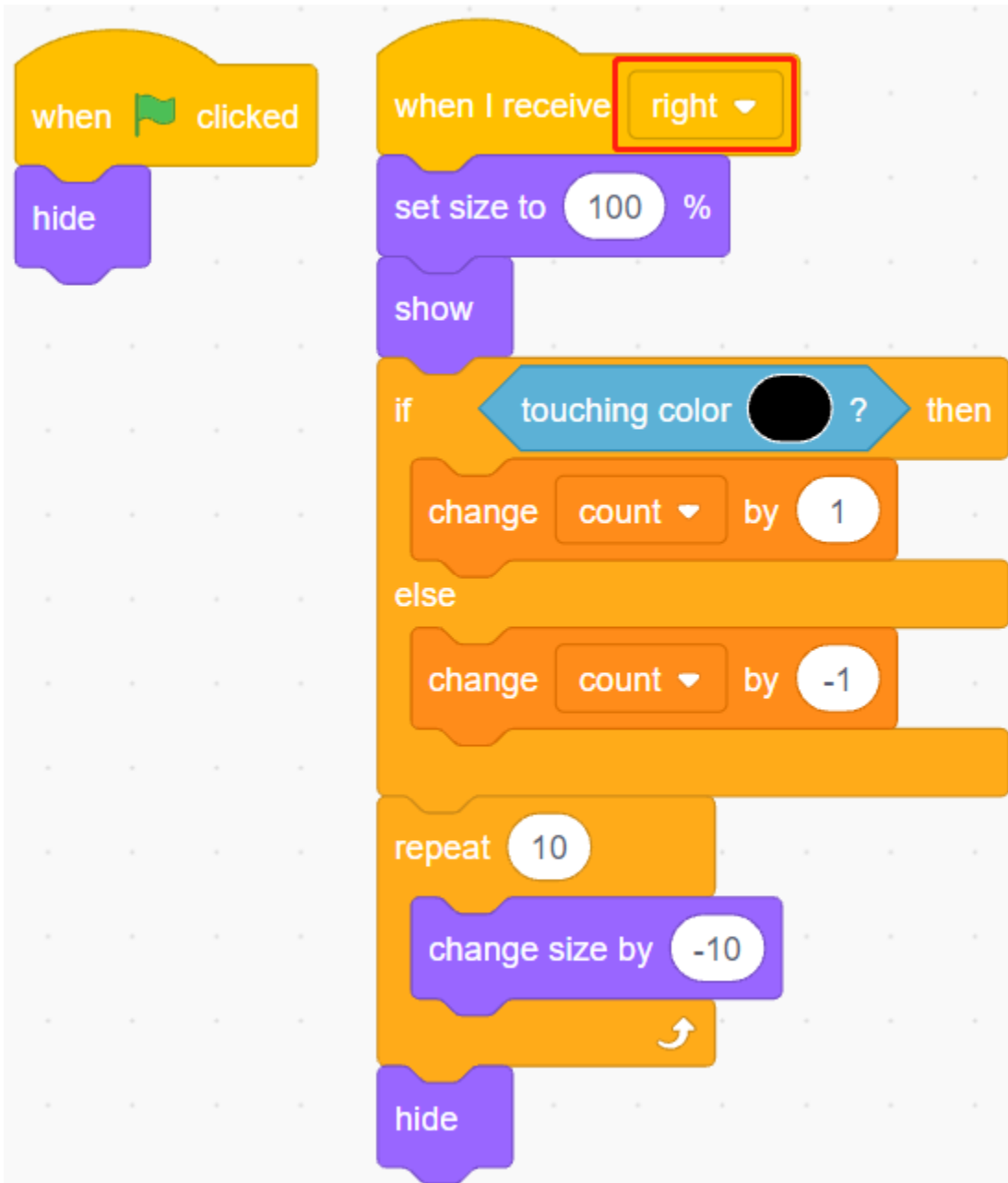


5. Right IR-Sprite

Kopieren Sie den **Left IR**-Sprite und benennen Sie ihn in **Right IR** um.



Ändern Sie anschließend die empfangene Nachricht in - **right**.



Nun sind alle Skripte fertig, und Sie können auf die grüne Flagge klicken, um das Skript auszuführen.

8.24 2.21 SPIEL - Schütze dein Herz

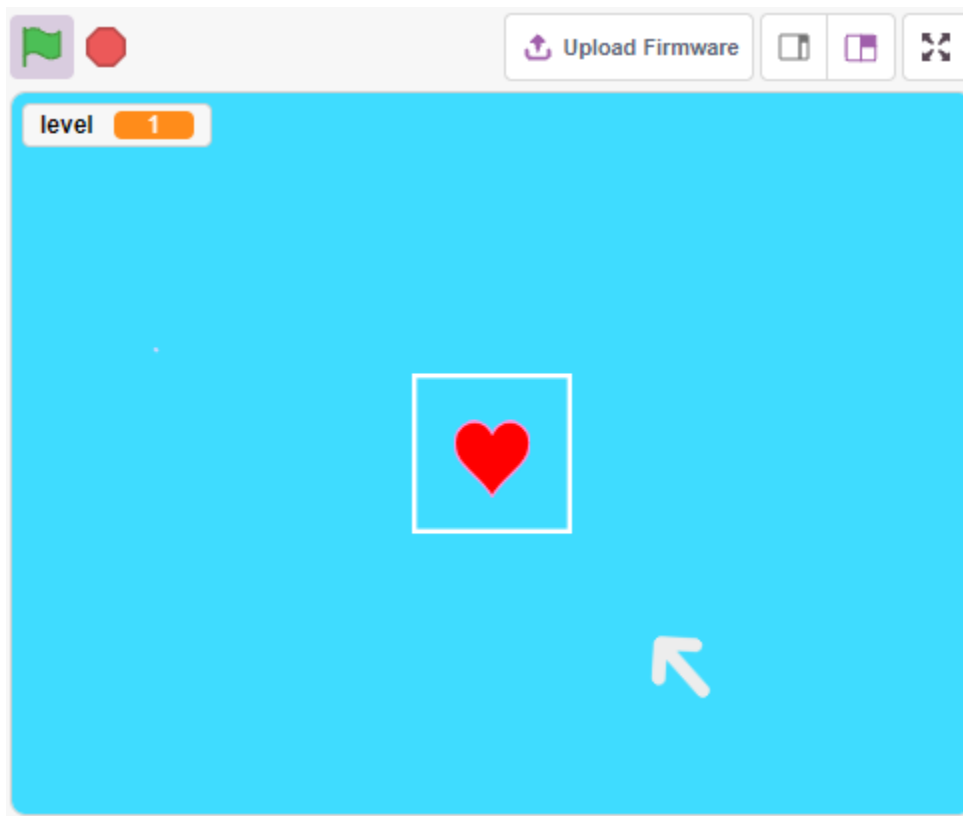
In diesem Projekt erstellen wir ein Spiel, das die Reaktionsgeschwindigkeit testet.

Auf der Bühne befindet sich ein Herz, das in einem rechteckigen Kasten geschützt ist. Pfeile fliegen aus beliebigen Positionen auf dieser Bühne auf dieses Herz zu. Die Farbe des Pfeils wechselt zufällig zwischen Schwarz und Weiß, und der Pfeil wird immer schneller fliegen.

Wenn die Farbe des rechteckigen Kastens und die Farbe des Pfeils gleich sind, wird der Pfeil blockiert und die Stufe erhöht sich um 1. Sind die Farben jedoch nicht gleich, wird der Pfeil das Herz durchschießen und das Spiel ist vorbei.

Hier wird die Farbe des rechteckigen Kastens vom Linienverfolgungsmodul gesteuert. Wenn das Modul auf einer schwarzen Oberfläche (einer reflektierenden Oberfläche) platziert wird, ist die Farbe des rechteckigen Kastens schwarz, ansonsten weiß.

Sie müssen also entscheiden, ob Sie das Linienverfolgungsmodul auf einer weißen oder einer schwarzen Oberfläche platzieren, je nach Farbe des Pfeils.



8.24.1 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+	

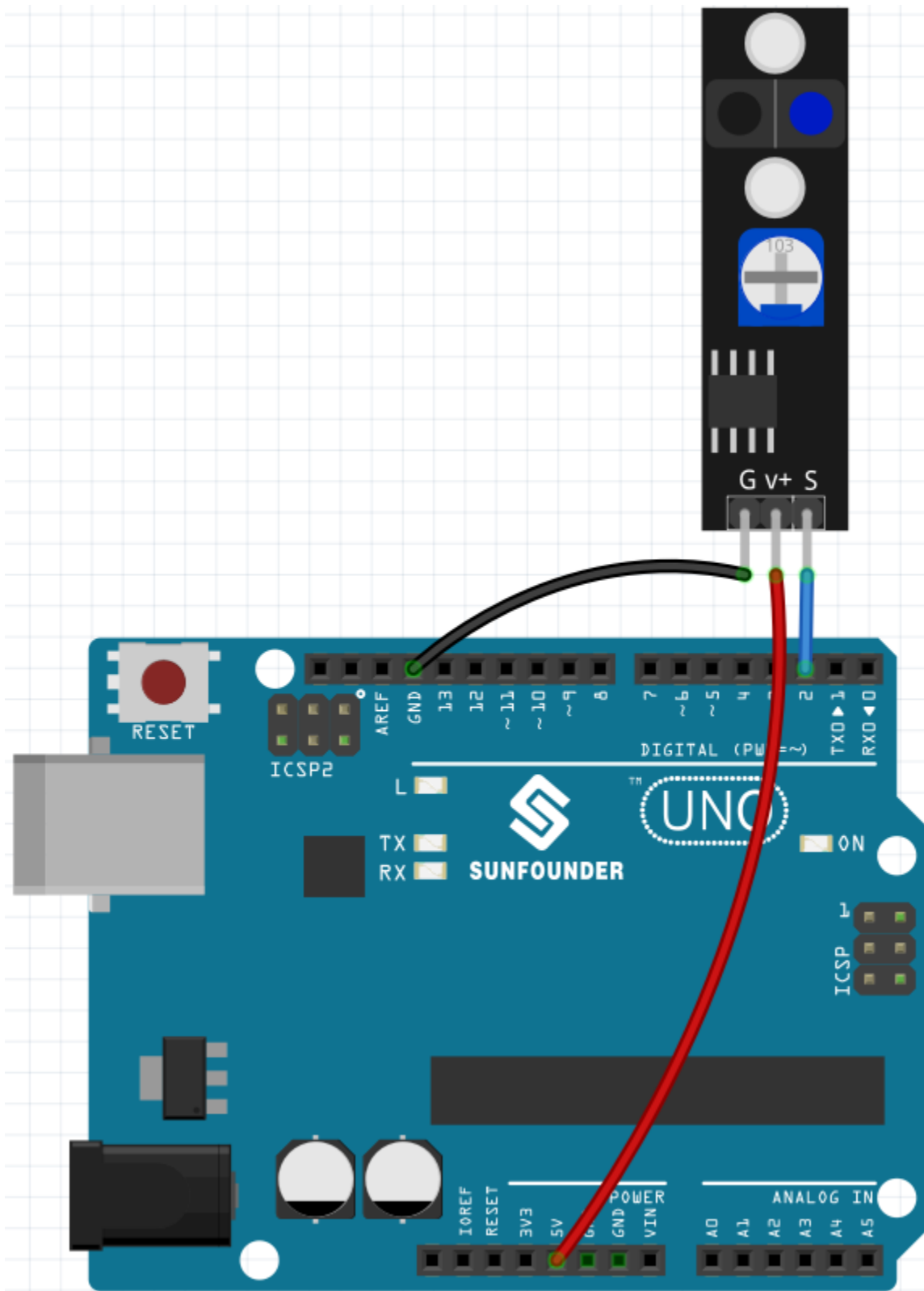
Sie können diese auch separat über die untenstehenden Links kaufen.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>Linienverfolgungsmodul</i>	

8.24.2 Schaltung aufbauen

Dies ist ein digitales Linienverfolgungsmodul. Wenn eine schwarze Linie erkannt wird, gibt es 1 aus; wenn eine weiße Linie erkannt wird, gibt es 0 aus. Zusätzlich können Sie den Erfassungsabstand über das Potentiometer auf dem Modul einstellen.

Bauen Sie den Schaltkreis jetzt gemäß dem untenstehenden Diagramm.



Bemerkung: Bevor Sie mit dem Projekt beginnen, müssen Sie die Empfindlichkeit des Moduls anpassen.

Verkabeln Sie gemäß dem obigen Diagramm und schalten Sie dann die R3-Platine ein (entweder direkt über das USB-

Kabel oder das 9V-Batteriekabel). Laden Sie den Code nicht hoch.

Kleben Sie nun ein schwarzes Elektroband auf den Schreibtisch und platzieren Sie das Linienverfolgungsmodul in einer Höhe von 2 cm vom Schreibtisch entfernt.

Mit dem Sensor nach unten gerichtet, beobachten Sie die Signalleuchte 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 erzielt.

8.24.3 Programmierung

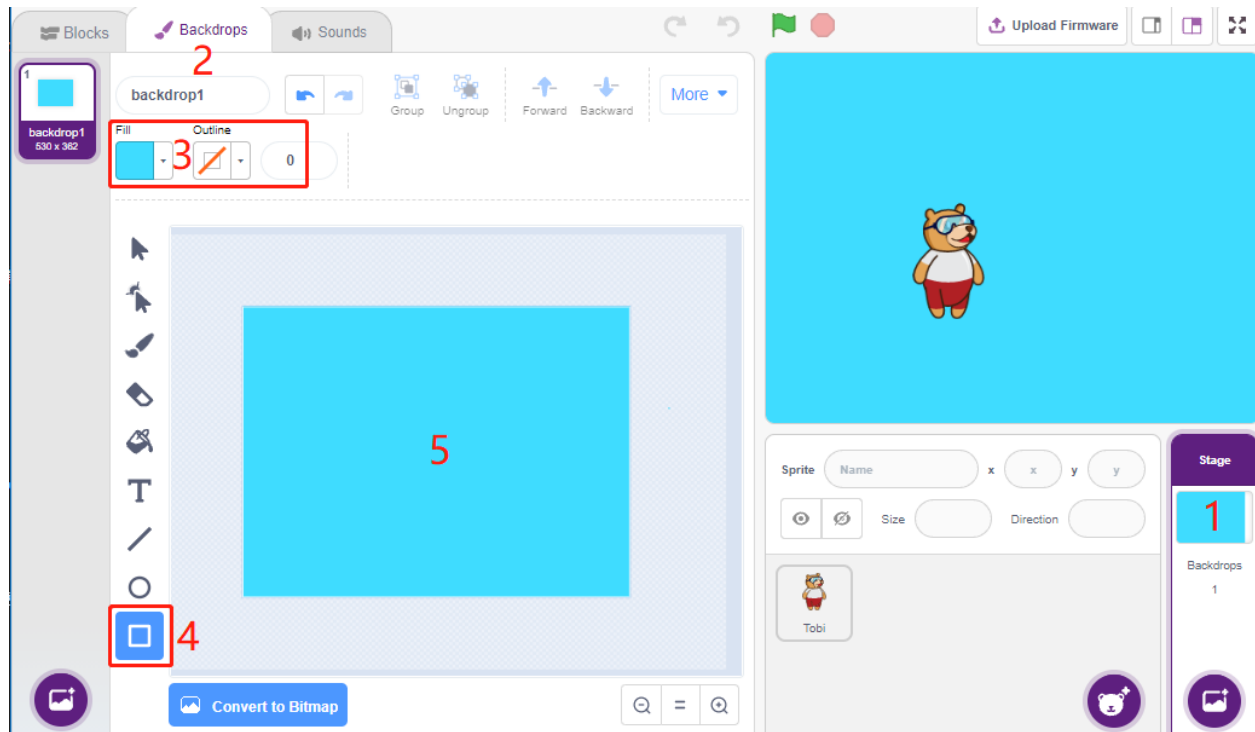
Hier müssen wir 3 Sprites erstellen: **Heart**, **Square Box** und **Arrow1**.

- **Heart**: bleibt in der Mitte der Bühne stehen. Wenn es vom **Arrow1**-Sprite berührt wird, ist das Spiel vorbei.
- **Square Box**: Hat zwei Arten von Kostümen, schwarz und weiß, und wechselt die Kostüme entsprechend dem Wert des Linienverfolgungsmoduls.
- **Arrow**: fliegt in Schwarz/Weiß aus irgendeiner Position auf die Mitte der Bühne zu; wenn seine Farbe mit der Farbe des **Square Box**-Sprites übereinstimmt, wird er blockiert und fliegt wieder aus einer zufälligen Position auf die Mitte der Bühne zu; stimmt seine Farbe nicht mit der Farbe des **Square Box**-Sprites überein, durchfliegt er das **Heart**-Sprite und das Spiel ist vorbei.

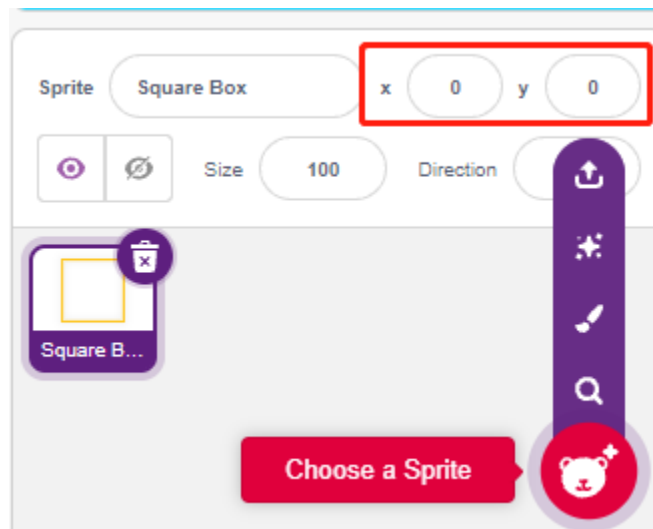
1. Sprite Rechteckiger Kasten hinzufügen

Da der Pfeil1 und das Sprite des rechteckigen Kastens beide weiße Kostüme haben, müssen sie, um auf der Bühne angezeigt zu werden, nun den Hintergrund mit einer Farbe füllen, die jede Farbe außer Schwarz, Weiß und Rot sein kann.

- Klicken Sie auf **Backdrop1**, um zu seiner **Backdrops**-Seite zu gelangen.
- Wählen Sie die gewünschte Füllfarbe aus.
- Verwenden Sie das **Rectangle**-Werkzeug, um ein Rechteck in der Größe des Zeichenbretts zu zeichnen.

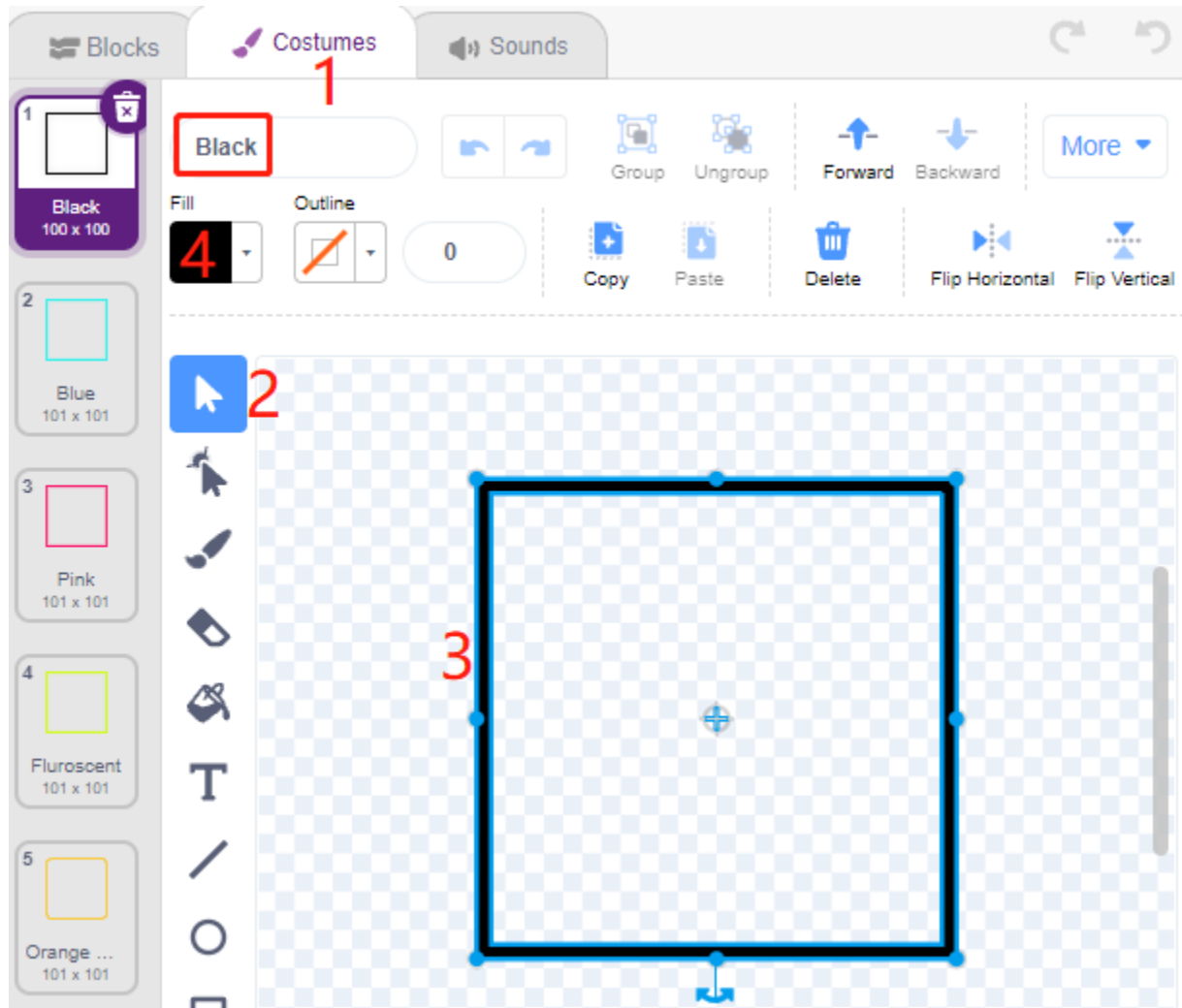


Löschen Sie das Standard-Sprite und verwenden Sie die Schaltfläche **Choose a Sprite**, um das **Square Box**-Sprite hinzuzufügen, und setzen Sie dessen x und y auf (0, 0).

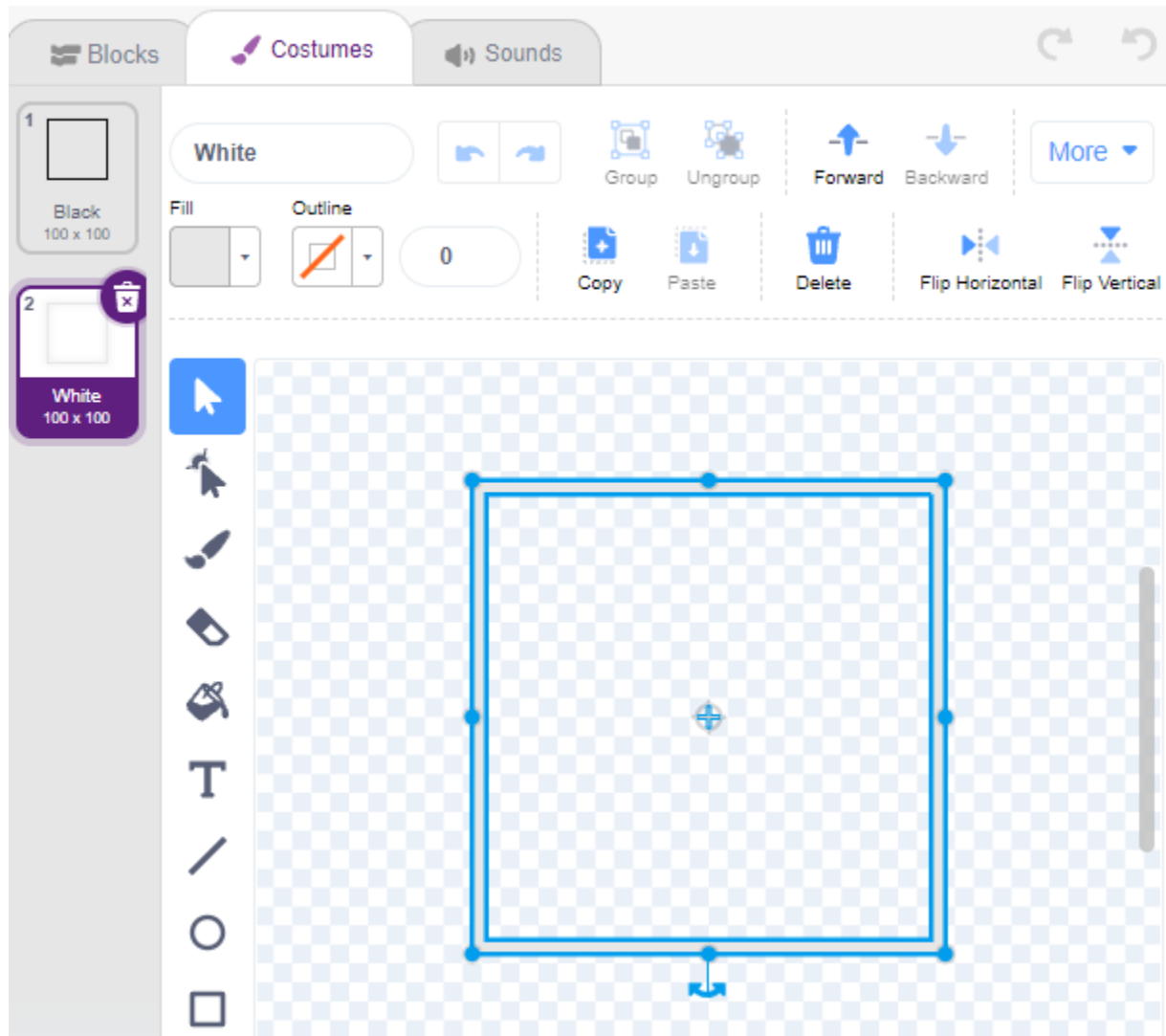


Gehen Sie zur **Costumes**-Seite des **Square Box**-Sprites und legen Sie die schwarz-weißen Kostüme fest.

- Klicken Sie auf das Auswahlinstrument
- Wählen Sie das Rechteck auf der Leinwand aus
- Wählen Sie die Füllfarbe Schwarz
- und nennen Sie das Kostüm **Black**

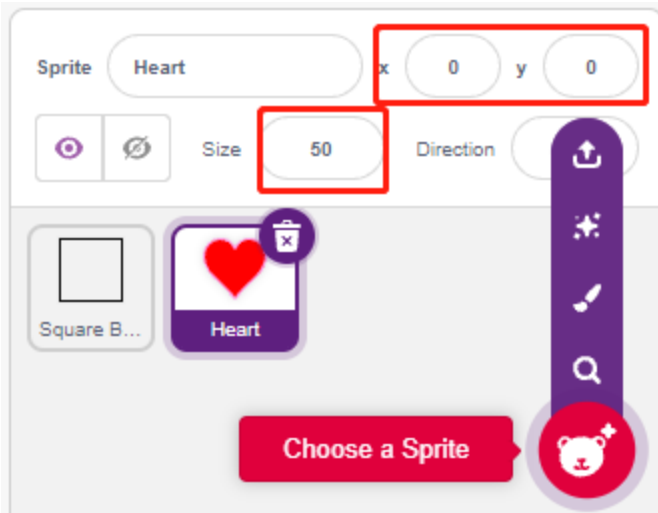


Wählen Sie das zweite Kostüm aus, stellen Sie die Füllfarbe auf Weiß, nennen Sie es Weiß und löschen Sie die restlichen Kostüme.

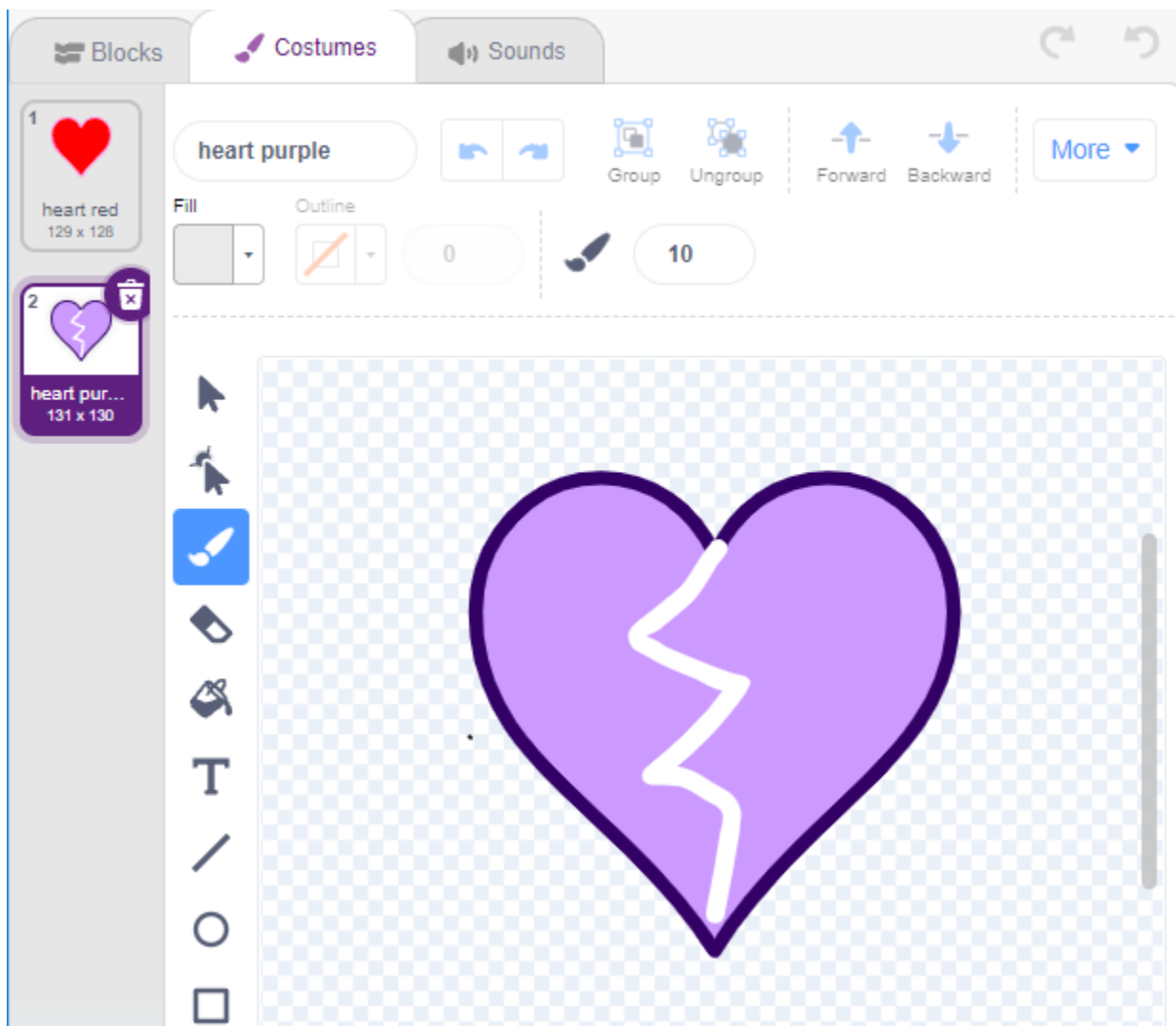


2. Herz-Sprite hinzufügen

Fügen Sie auch ein **Heart**-Sprite hinzu, setzen Sie seine Position auf (0, 0) und verkleinern Sie seine Größe, sodass es so aussieht, als ob es sich innerhalb des Quadrat-Kastens befindet.

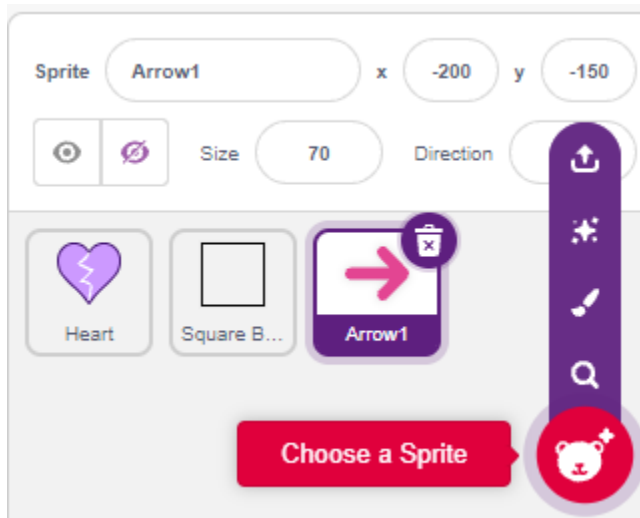


Auf der **Costumes**-Seite passen Sie das lila Herz-Kostüm an, damit es aussieht, als wäre es zerbrochen.

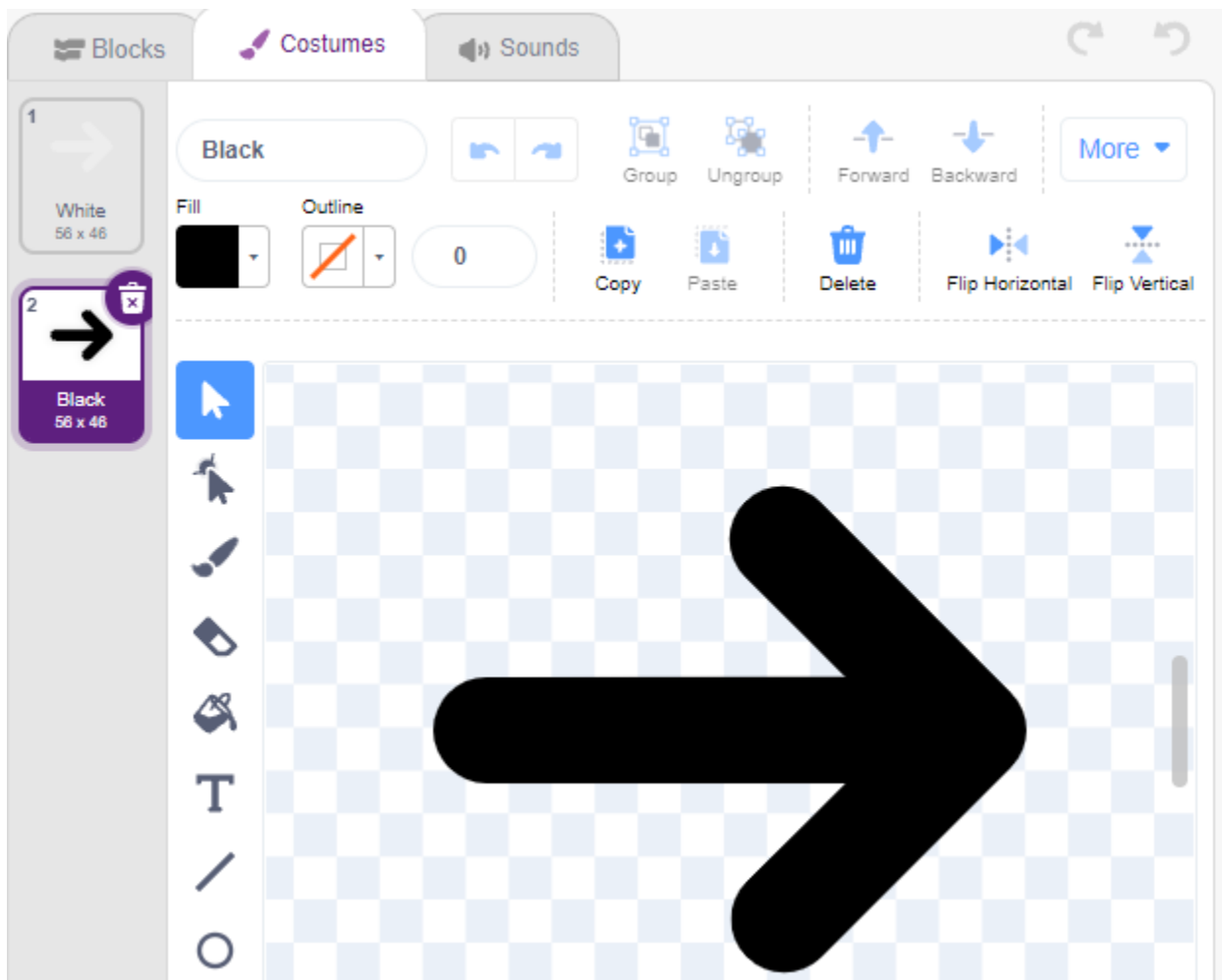


3. Arrow1-Sprite hinzufügen

Fügen Sie ein **Arrow1**-Sprite hinzu.



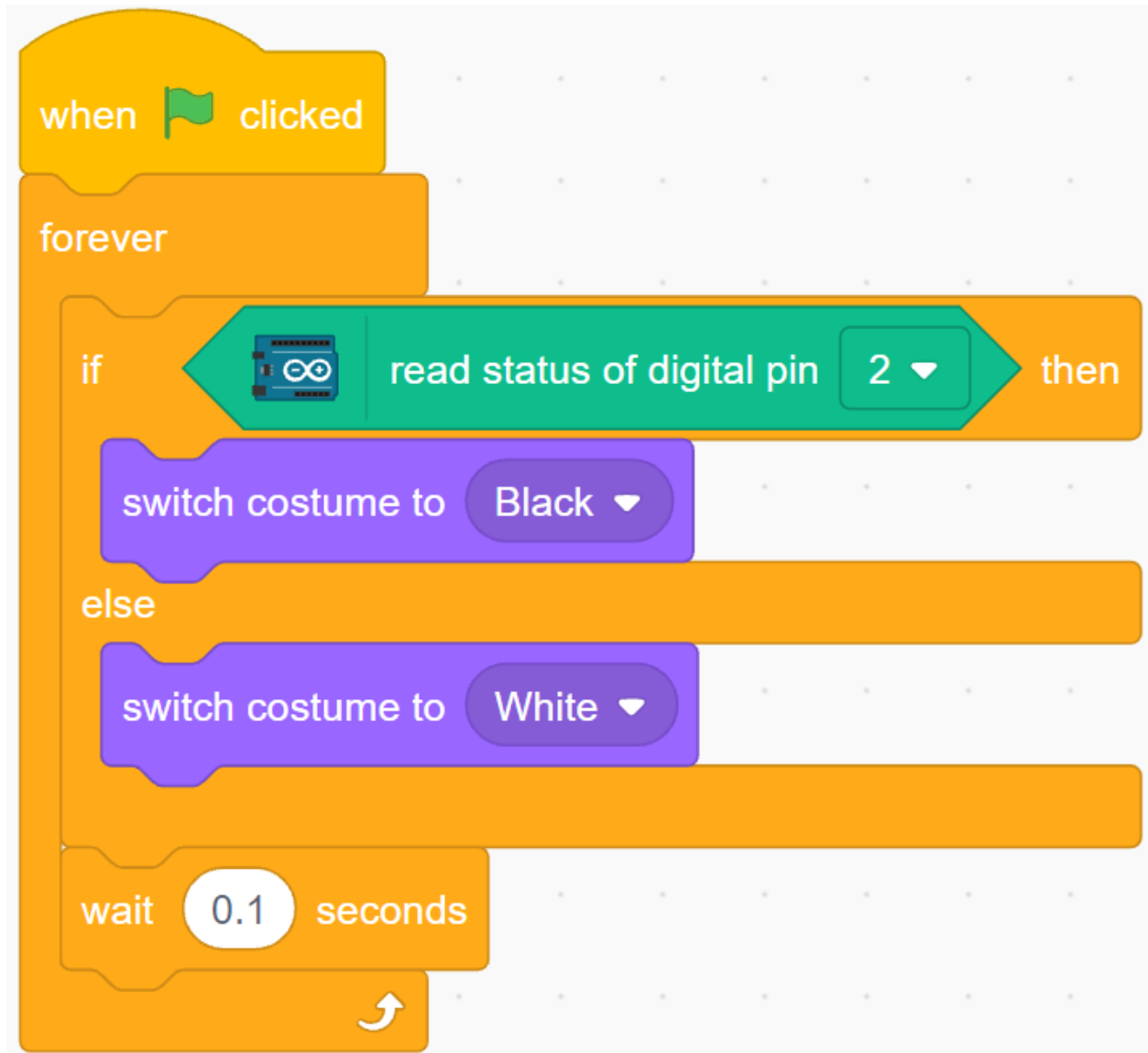
Auf der **Costumes**-Seite behalten und kopieren Sie das nach rechts gerichtete Kostüm und setzen Sie seine Farbe auf Schwarz und Weiß.



4. Skripterstellung für das Quadrat-Kasten-Sprite

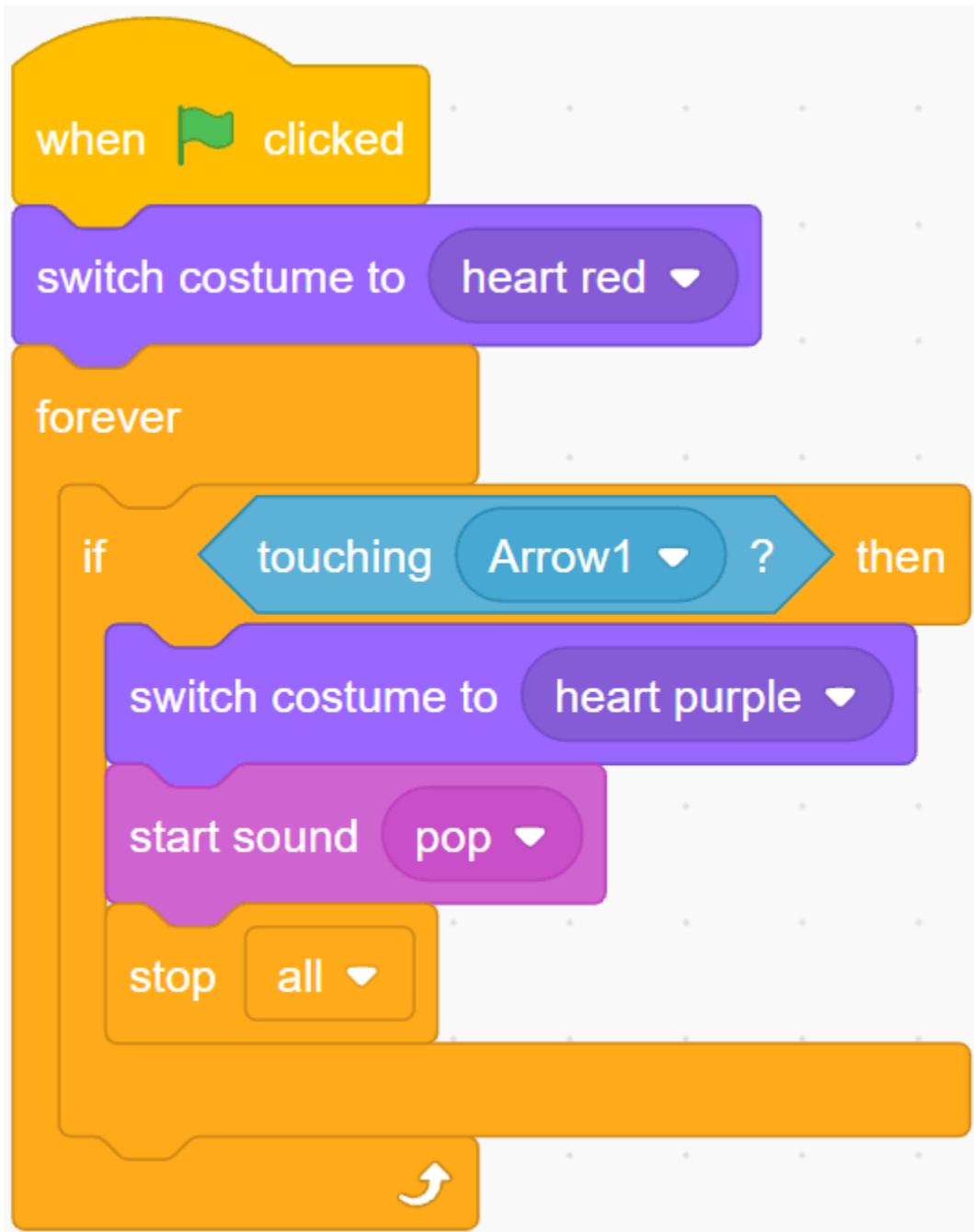
Gehen Sie zurück zur **Blocks**-Seite und scripten Sie das **Square Box**-Sprite.

- Wenn der Wert des digitalen Pins 2 (Linienverfolgungsmodul) 1 ist (schwarze Linie erkannt), wechseln Sie das Kostüm zu **Black**.
- Andernfalls wechseln Sie das Kostüm zu **White**.



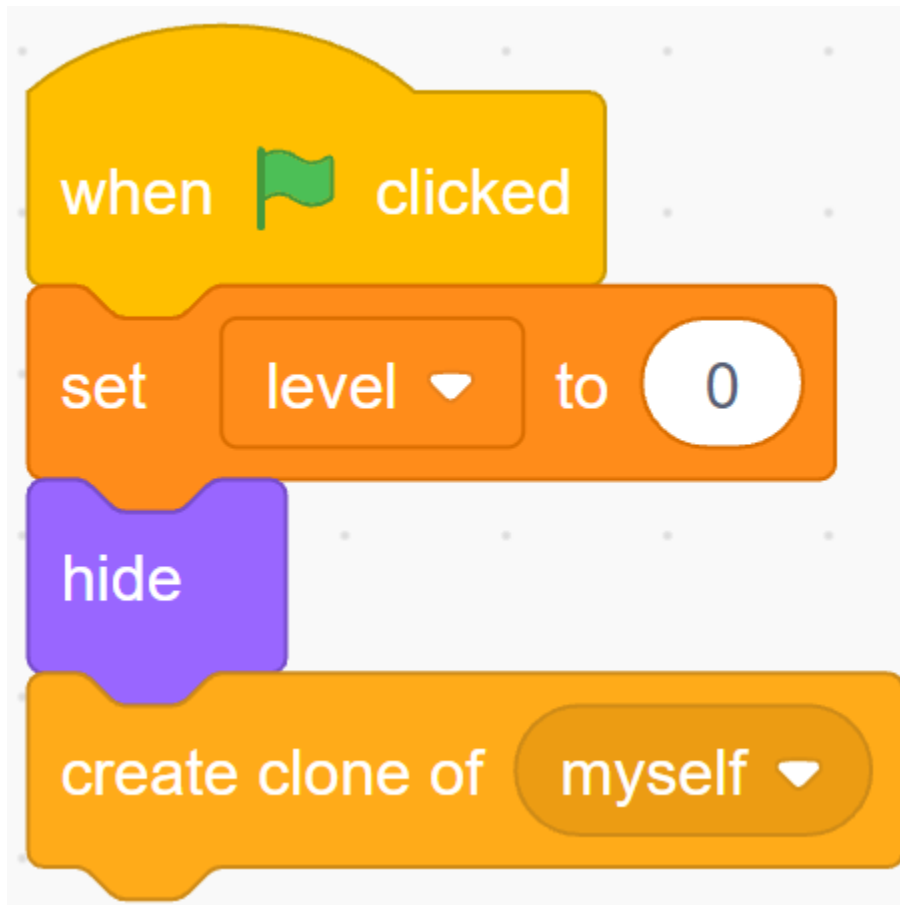
5. Skripterstellung für das Herz-Sprite

Das **Heart**-Sprite wird im **Square Box** geschützt und hat standardmäßig ein rotes Kostüm. Wenn das Arrow1-Sprite berührt wird, endet das Spiel.



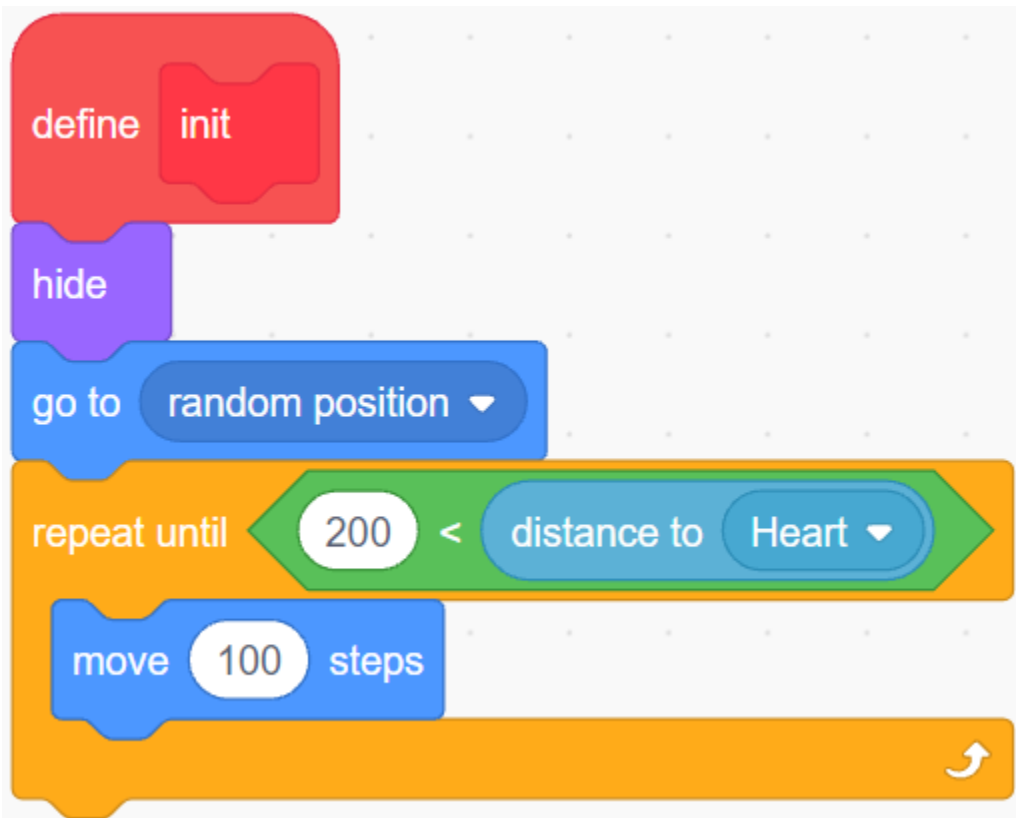
6. Skripterstellung für das Arrow1-Sprite

Lassen Sie das **Arrow1**-Sprite verstecken und einen Klon erstellen, wenn die grüne Flagge angeklickt wird.

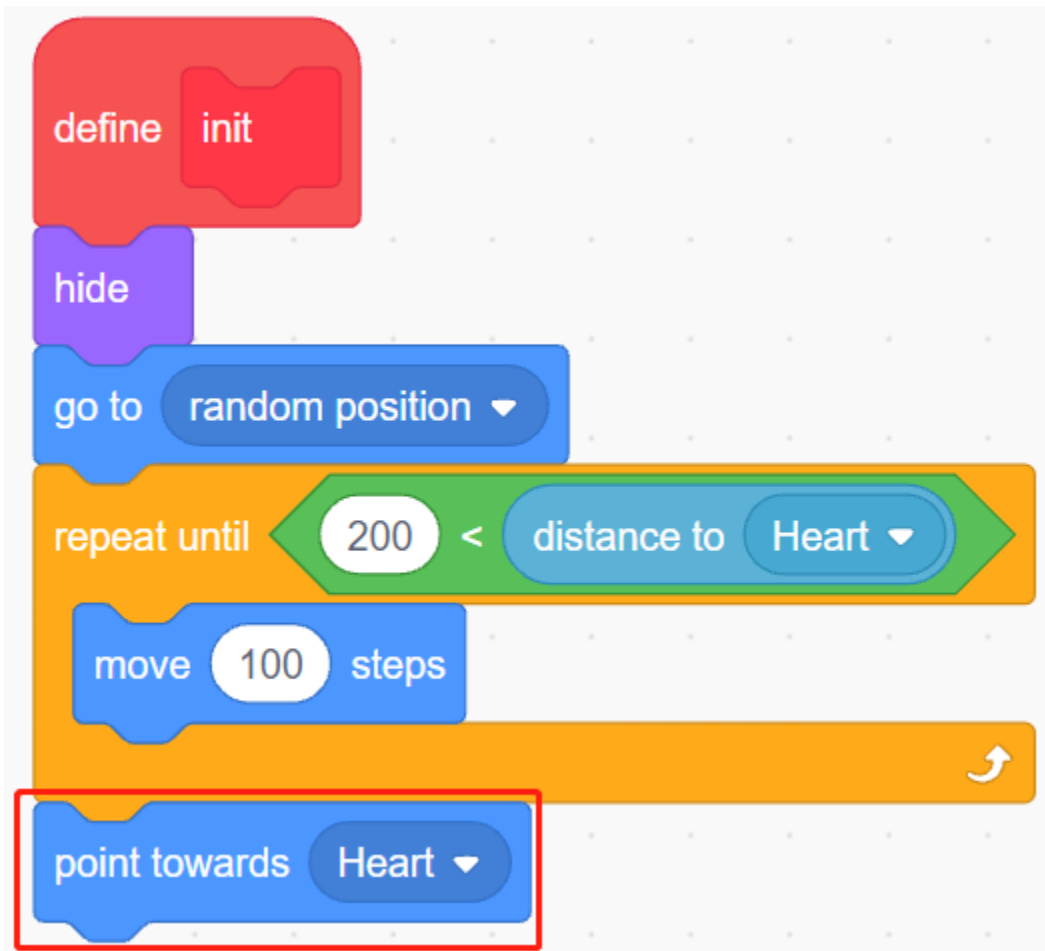


Erstellen Sie einen [init]-Block, um die Position, Orientierung und Farbe des **Arrow1**-Sprites zu initialisieren.

Es erscheint an einer zufälligen Position, und wenn der Abstand zwischen ihm und dem **Heart**-Sprite weniger als 200 beträgt, bewegt es sich nach außen, bis der Abstand größer als 200 ist.

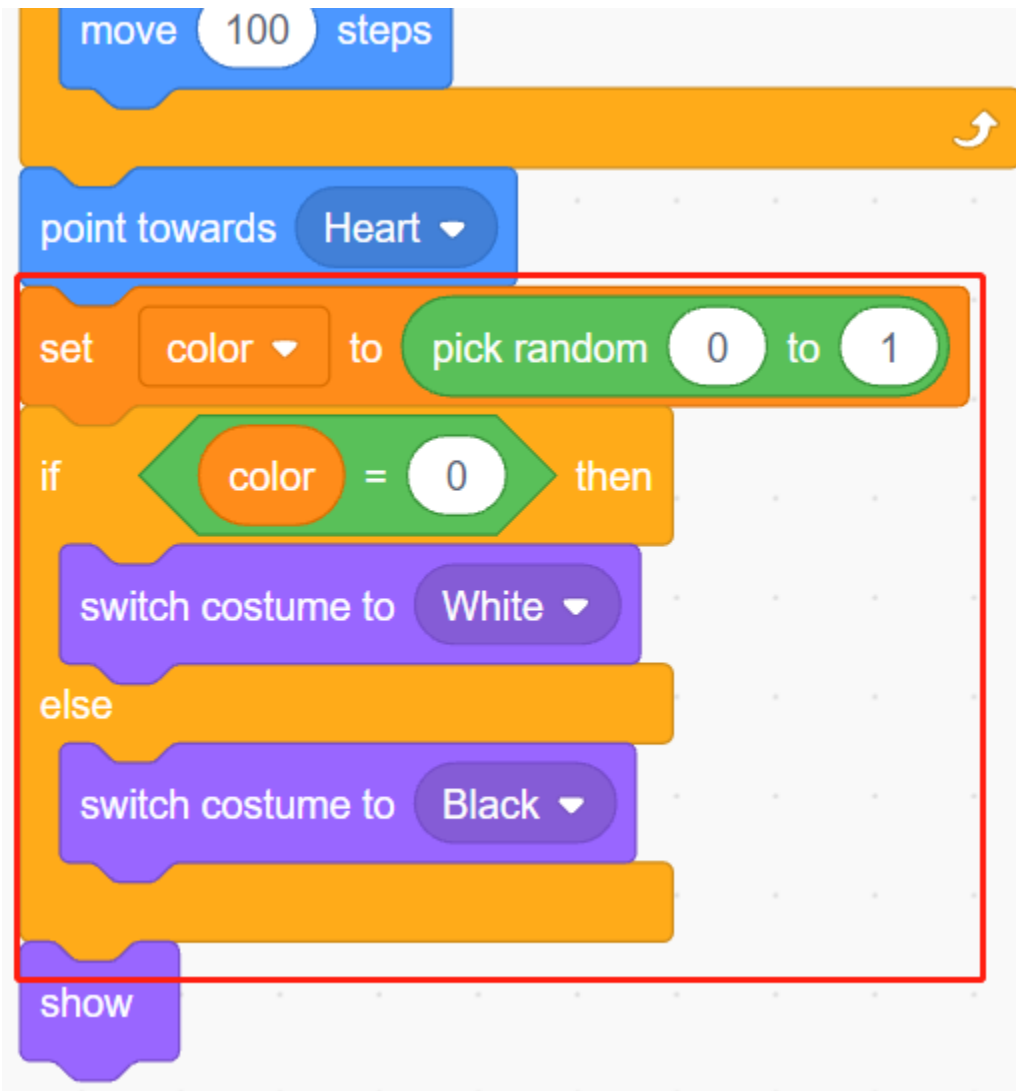


Richten Sie seine Richtung auf das **Heart**-Sprite aus.

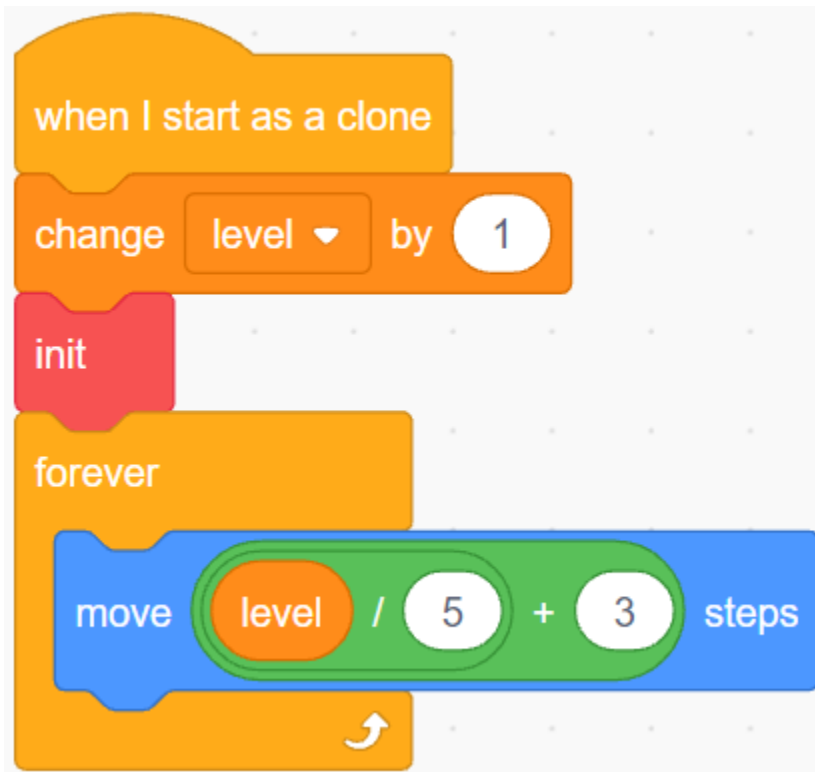


Lassen Sie seine Farbe zufällig zwischen Schwarz/Weiß wechseln.

- Ist die Variable Farbe 0, wechseln Sie das Kostüm zu **White**.
- Ist die Variable Farbe 1, wechseln Sie das Kostüm zu **Black**.

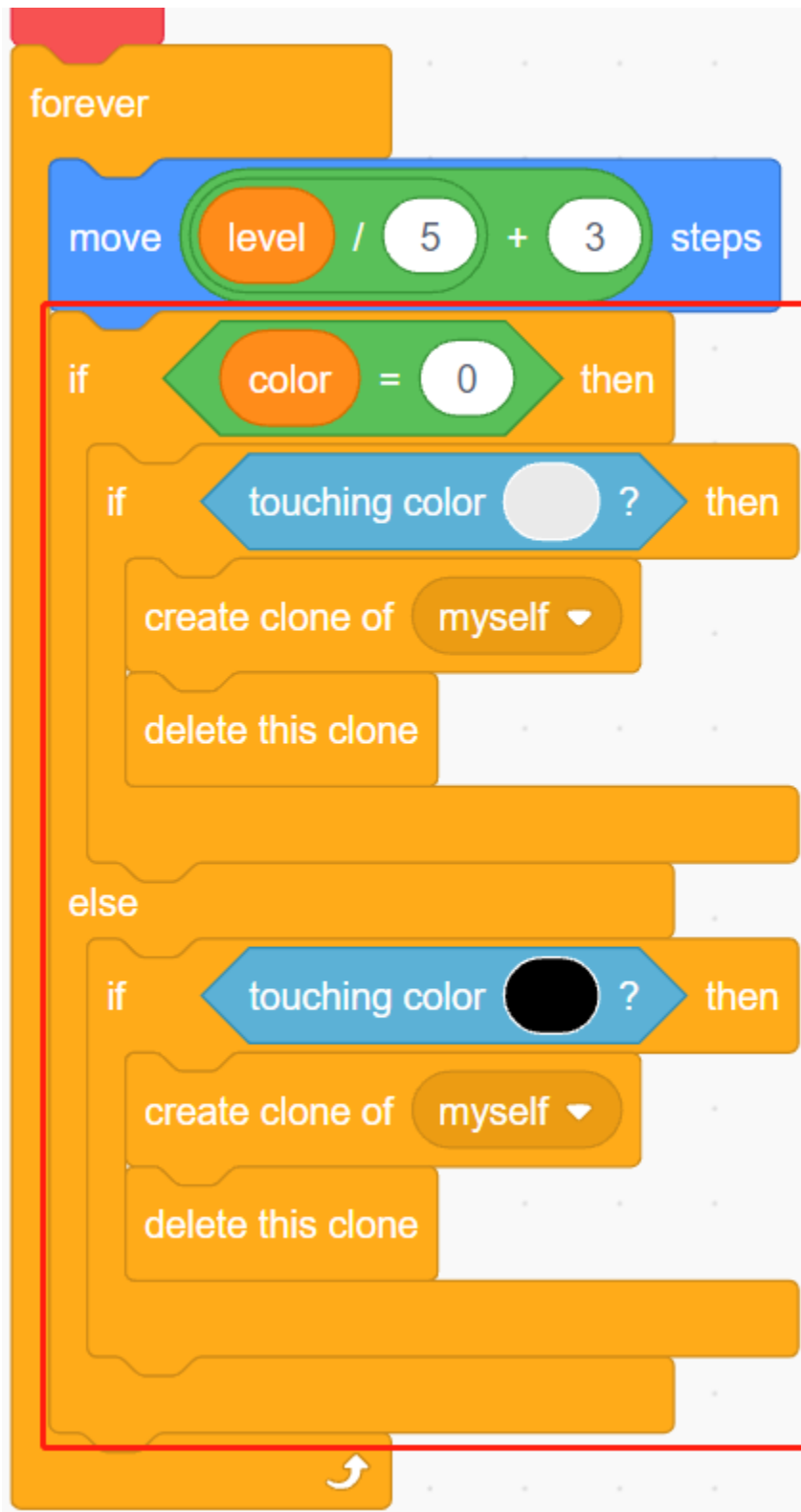


Lassen Sie es jetzt bewegen, es wird schneller bewegen, wenn der Wert der Variable **level** zunimmt.

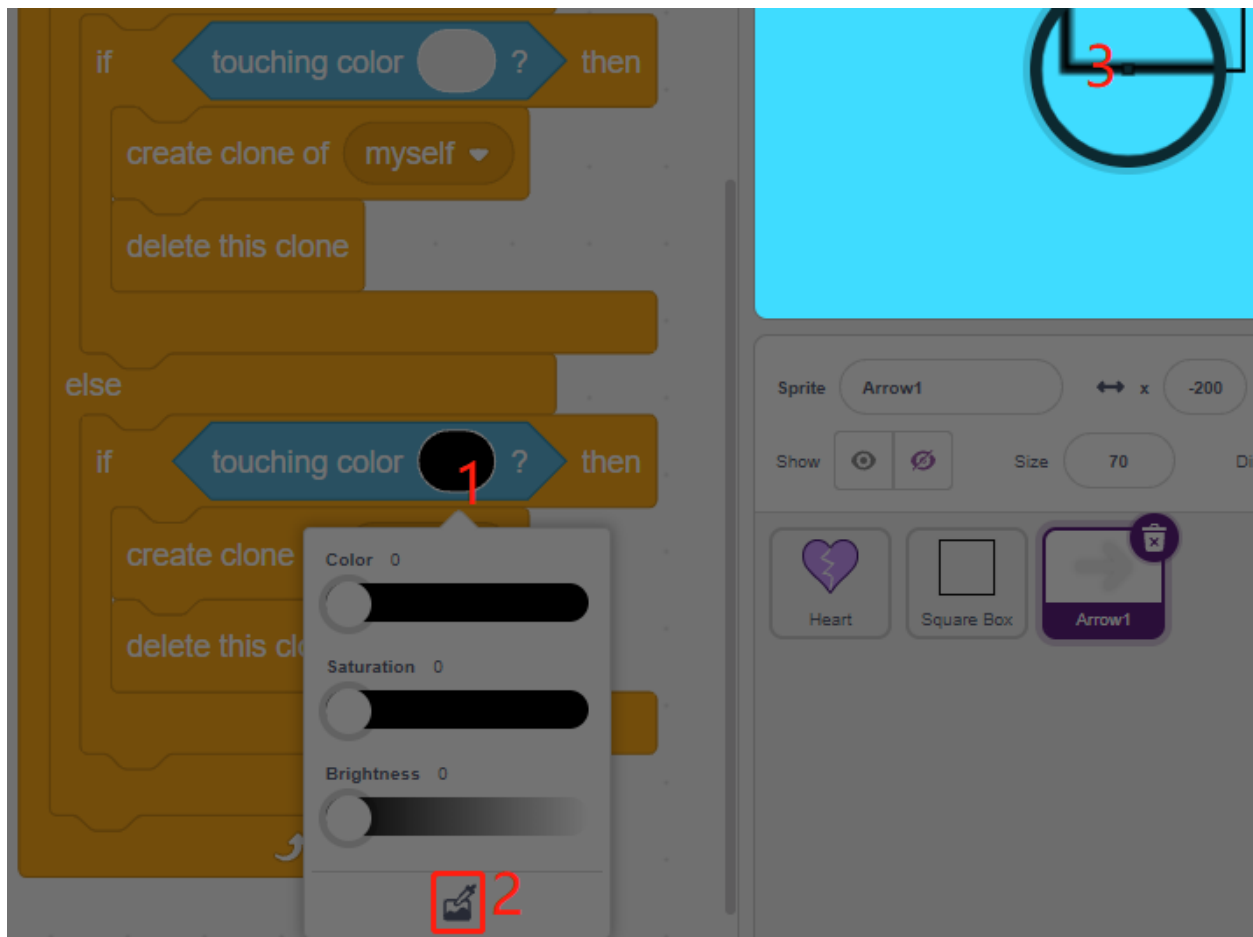


Legen Sie jetzt seinen Kollisions-Effekt mit dem **Square Box**-Sprite fest.

- Wenn das **Arrow1**-Sprite und das **Square Box**-Sprite dieselbe Farbe haben (die gemäß dem Wert des Linienverfolgungsmoduls geändert wird), entweder Schwarz oder Weiß, wird ein neuer Klon erstellt und das Spiel geht weiter.
- Stimmen ihre Farben nicht überein, bewegt sich das **Arrow1**-Sprite weiter und das Spiel endet, wenn es das **Heart**-Sprite trifft.



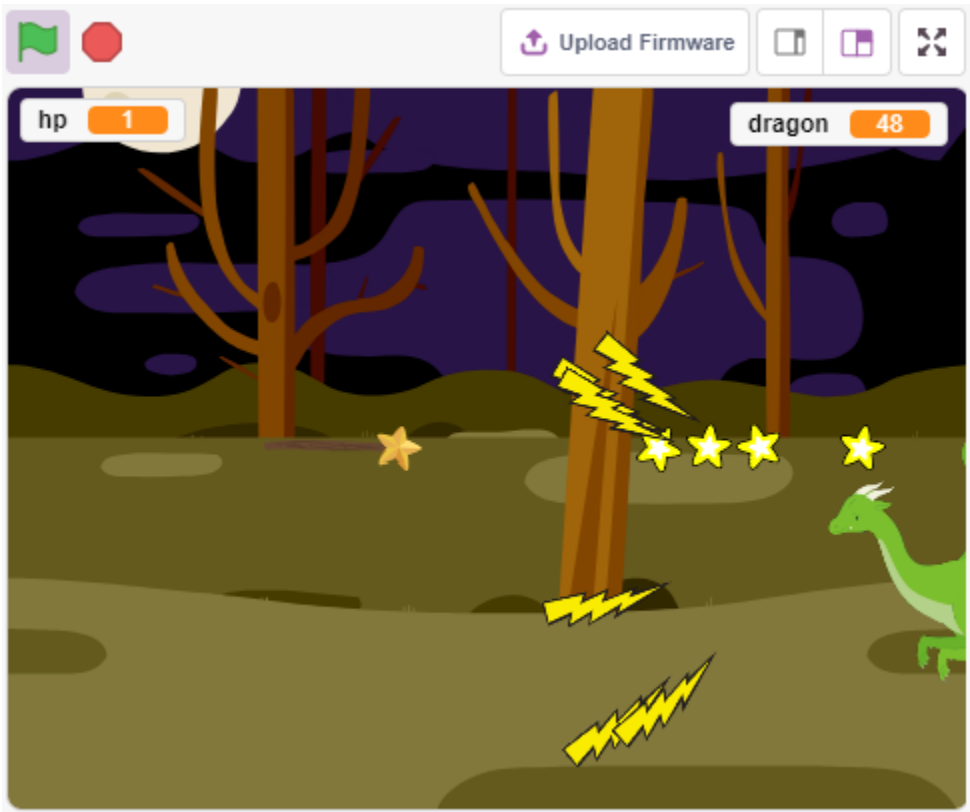
Bemerkung: Die beiden [touch color()]-Blöcke müssen die schwarz-weißen Kostüme von Quadrat-Kasten separat aufnehmen.



8.25 2.22 SPIEL - Drachen töten

In diesem Spiel verwenden wir den Joystick, um einen Drachen zu bekämpfen.

Beim Klicken auf Grün wird der Drache rechts auf und ab schweben und intermittierend Feuer speien. Sie müssen den Joystick verwenden, um die Bewegung des Zauberstabs zu steuern und Sternenangriffe auf den Drachen zu starten, während Sie den von ihm abgefeuerten Flammen ausweichen und ihn schließlich besiegen.



8.25.1 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.

KOMPONENTENVORSTELLUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>Jumper-Kabel</i>	
<i>Joystick Modul</i>	-

8.25.2 Schaltung aufbauen

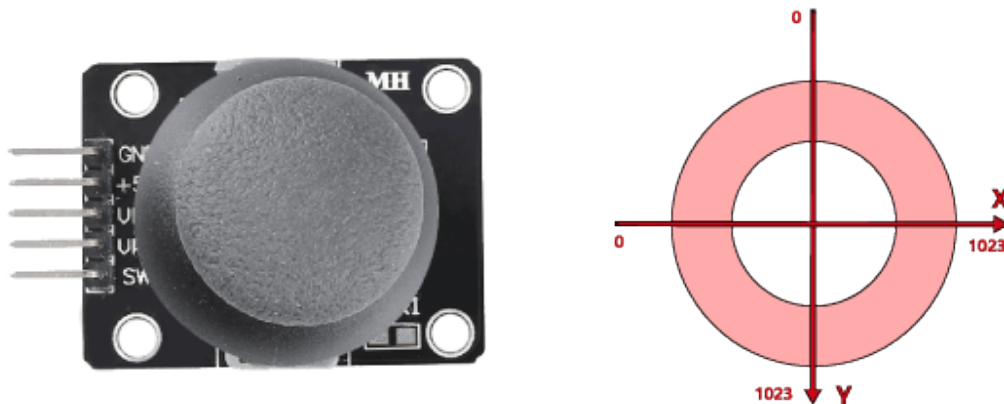
Ein Joystick ist ein Eingabegerät, das aus einem Stick besteht, der auf einer Basis schwenkt und den Winkel oder die Richtung an das zu steuernde Gerät meldet. Joysticks werden häufig verwendet, um Videospiele und Roboter zu steuern.

Um einem Computer einen vollen Bewegungsbereich mitzuteilen, muss ein Joystick die Stickposition auf zwei Achsen messen - der X-Achse (links nach rechts) und der Y-Achse (oben nach unten).

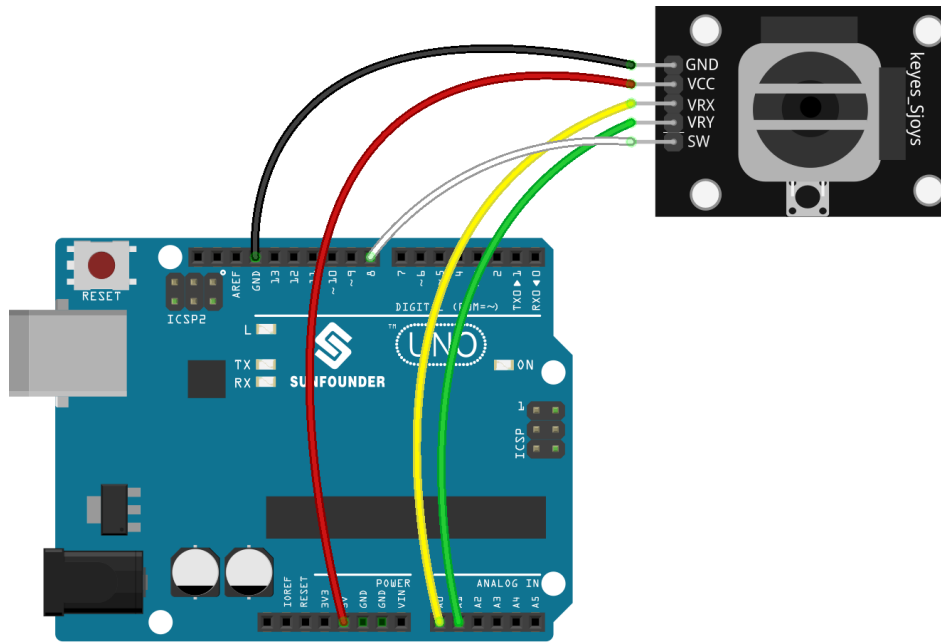
Die Bewegungskordinaten des Joysticks werden in der folgenden Abbildung gezeigt.

Bemerkung:

- Die x-Koordinate geht von links nach rechts, der Bereich liegt zwischen 0 und 1023.
 - Die y-Koordinate geht von oben nach unten, der Bereich liegt zwischen 0 und 1023.
-



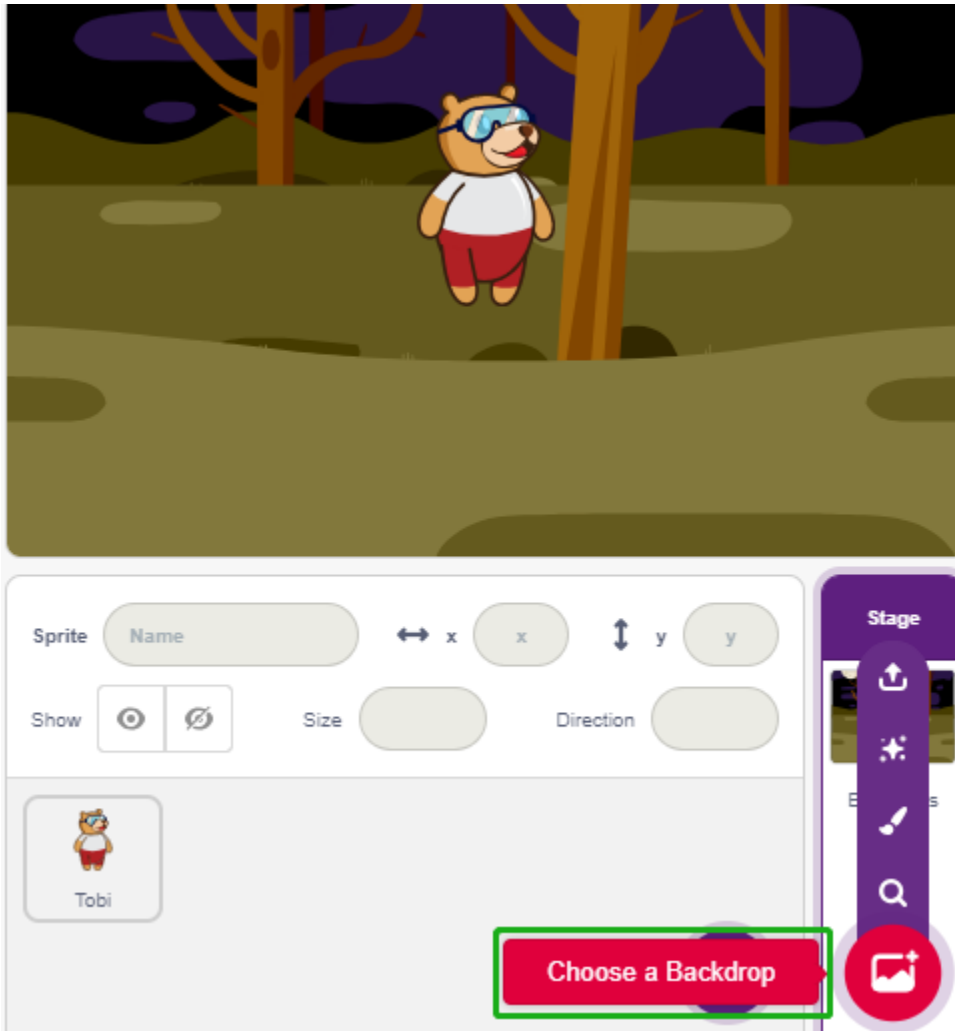
Bauen Sie nun die Schaltung gemäß dem folgenden Diagramm.



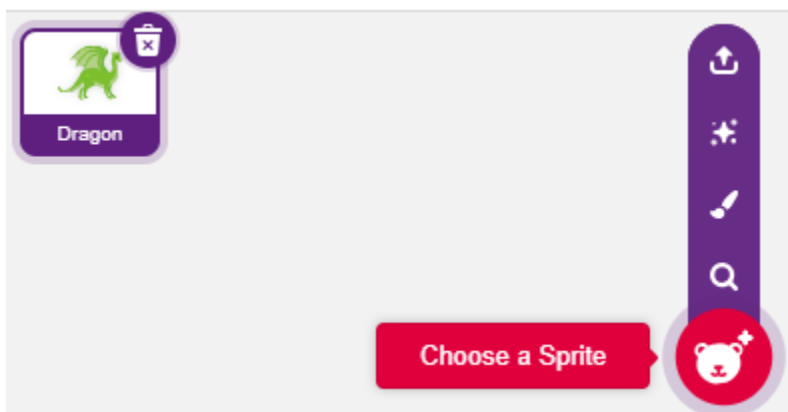
8.25.3 Programmierung

1. Drache

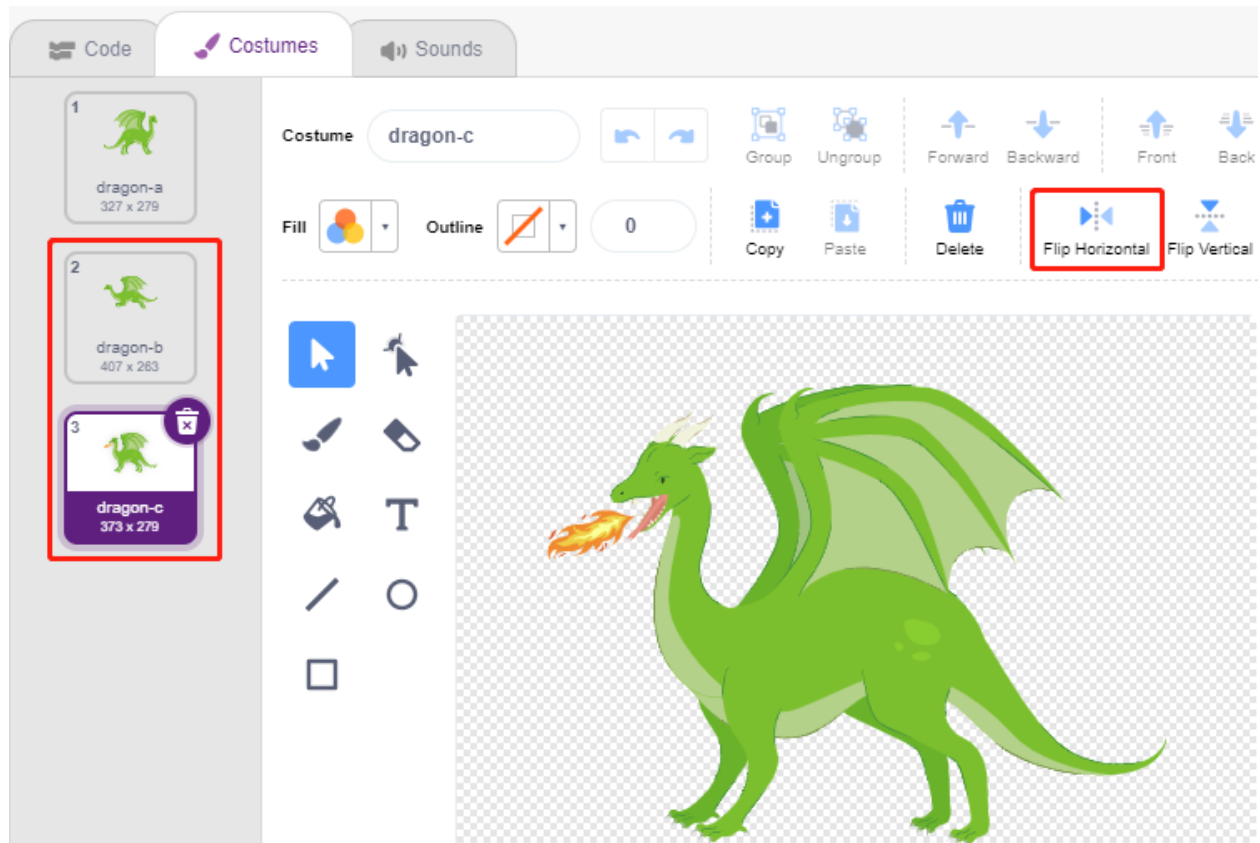
Woods-Hintergrund hinzugefügt über den **Choose a Backdrop**-Knopf.



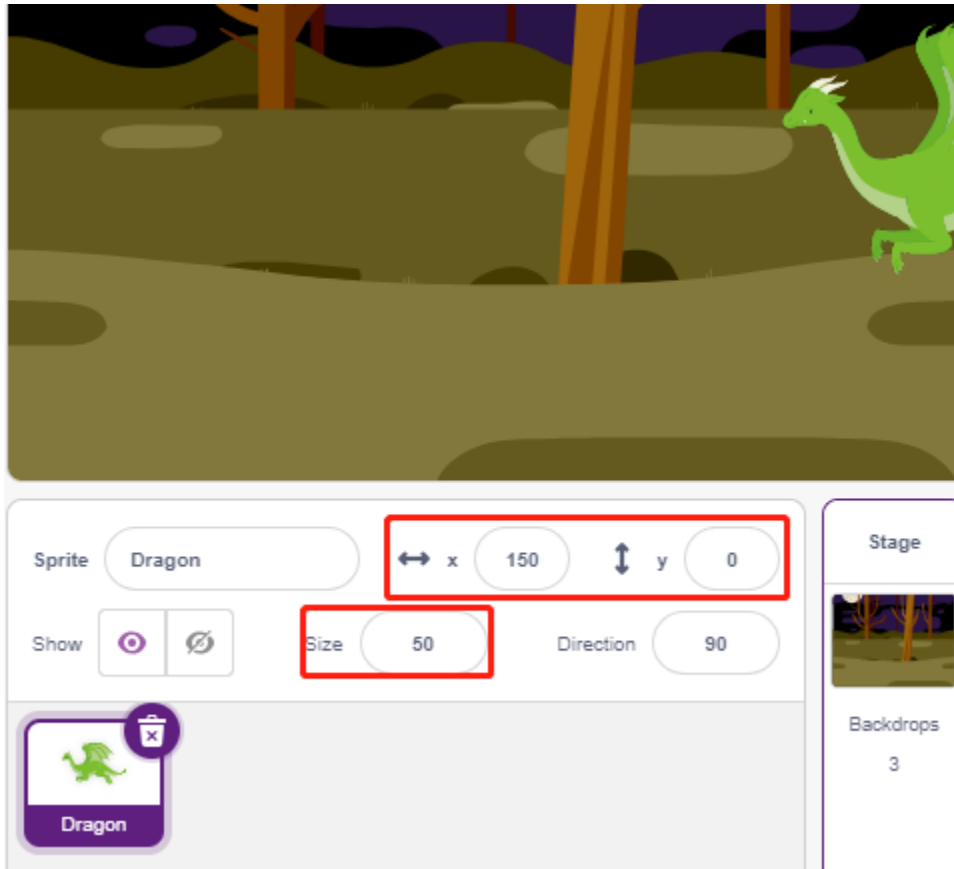
- Löschen Sie das Standard-Sprite und fügen Sie das **Dragon**-Sprite hinzu.



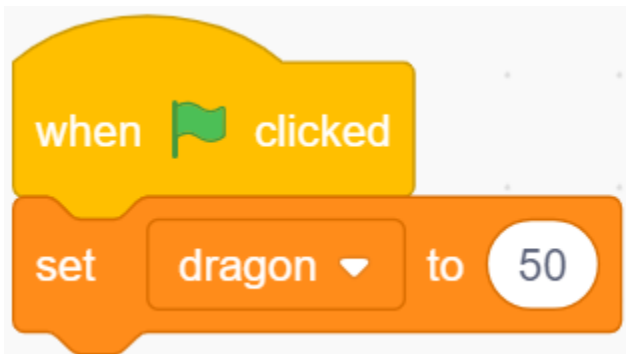
- Wechseln Sie zur **Costumes**-Seite und spiegeln Sie den Drachen-b und Drachen-c horizontal.



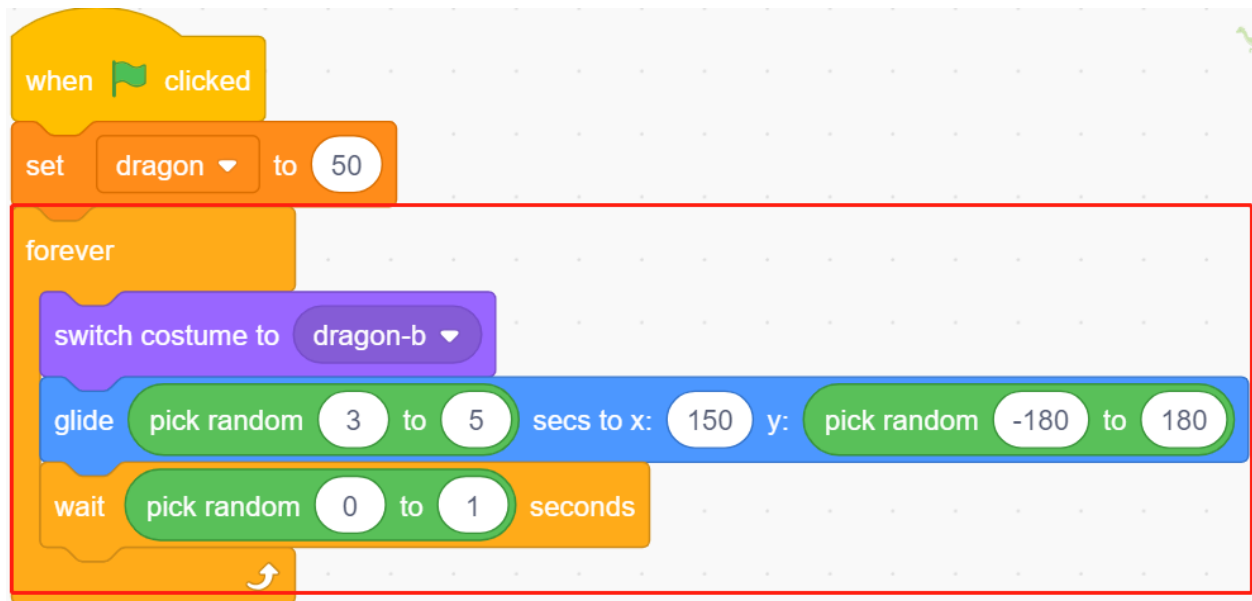
- Stellen Sie die Größe auf 50% ein.



- Erstellen Sie nun eine Variable - **dragon**, um die Lebenspunkte des Drachens aufzuzeichnen, und setzen Sie den Anfangswert auf 50.

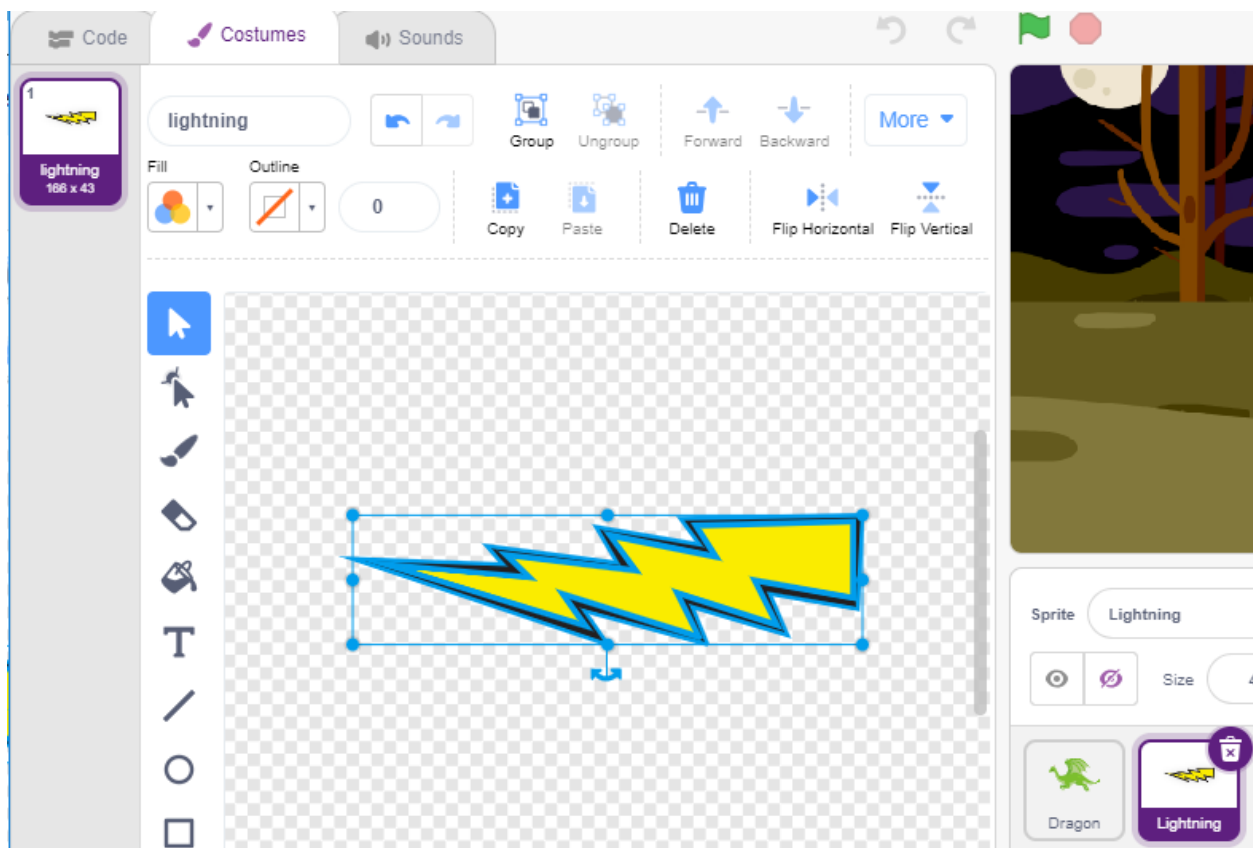


- Wechseln Sie als Nächstes das Kostüm des Sprites zu **dragon-b** und lassen Sie das **Dragon**-Sprite in einem bestimmten Bereich auf und ab bewegen.

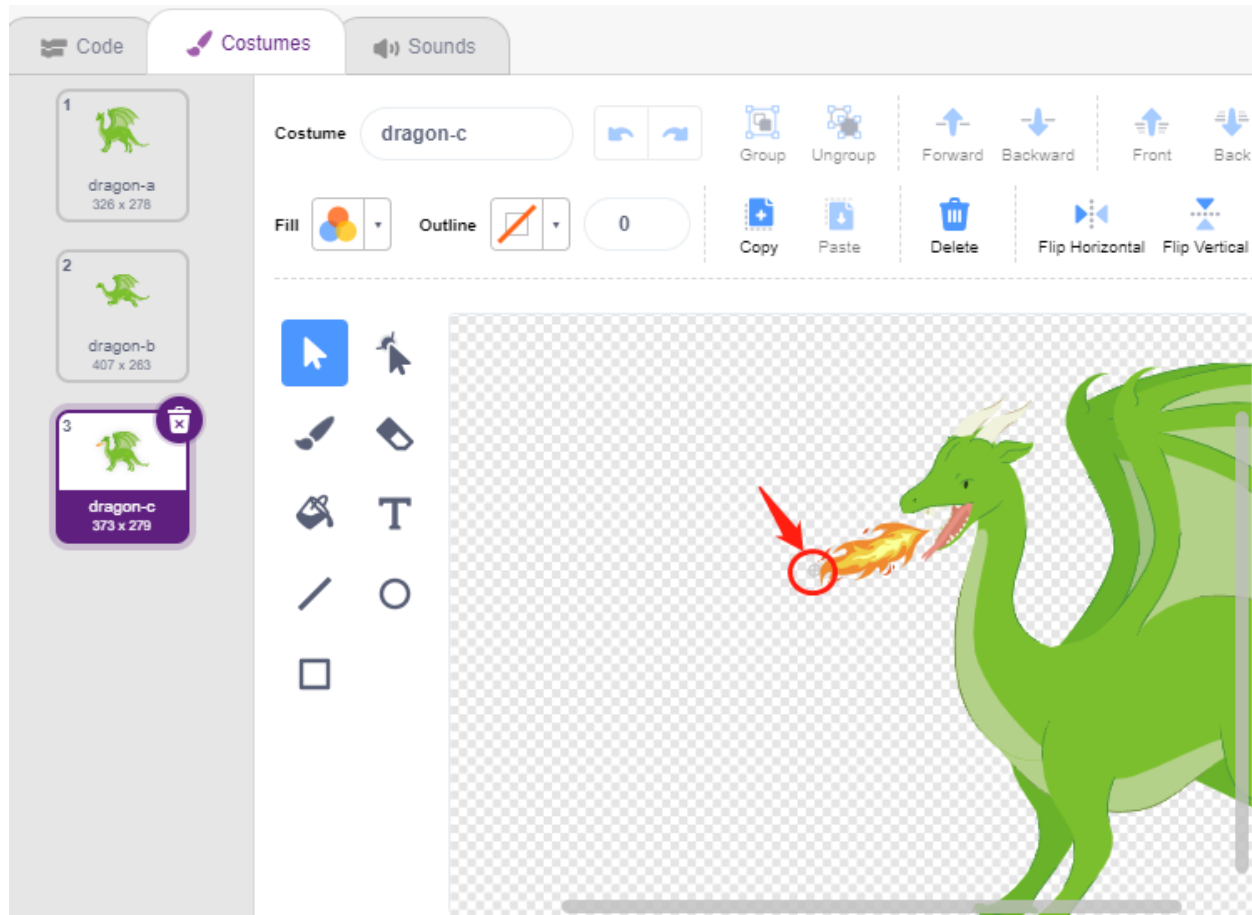


- Fügen Sie ein **Lightning**-Sprite als das vom **Dragon**-Sprite geblasene Feuer hinzu. Sie müssen es auf der Kostümseite um 90° im Uhrzeigersinn drehen, um das **Lightning**-Sprite in die richtige Richtung zu bewegen.

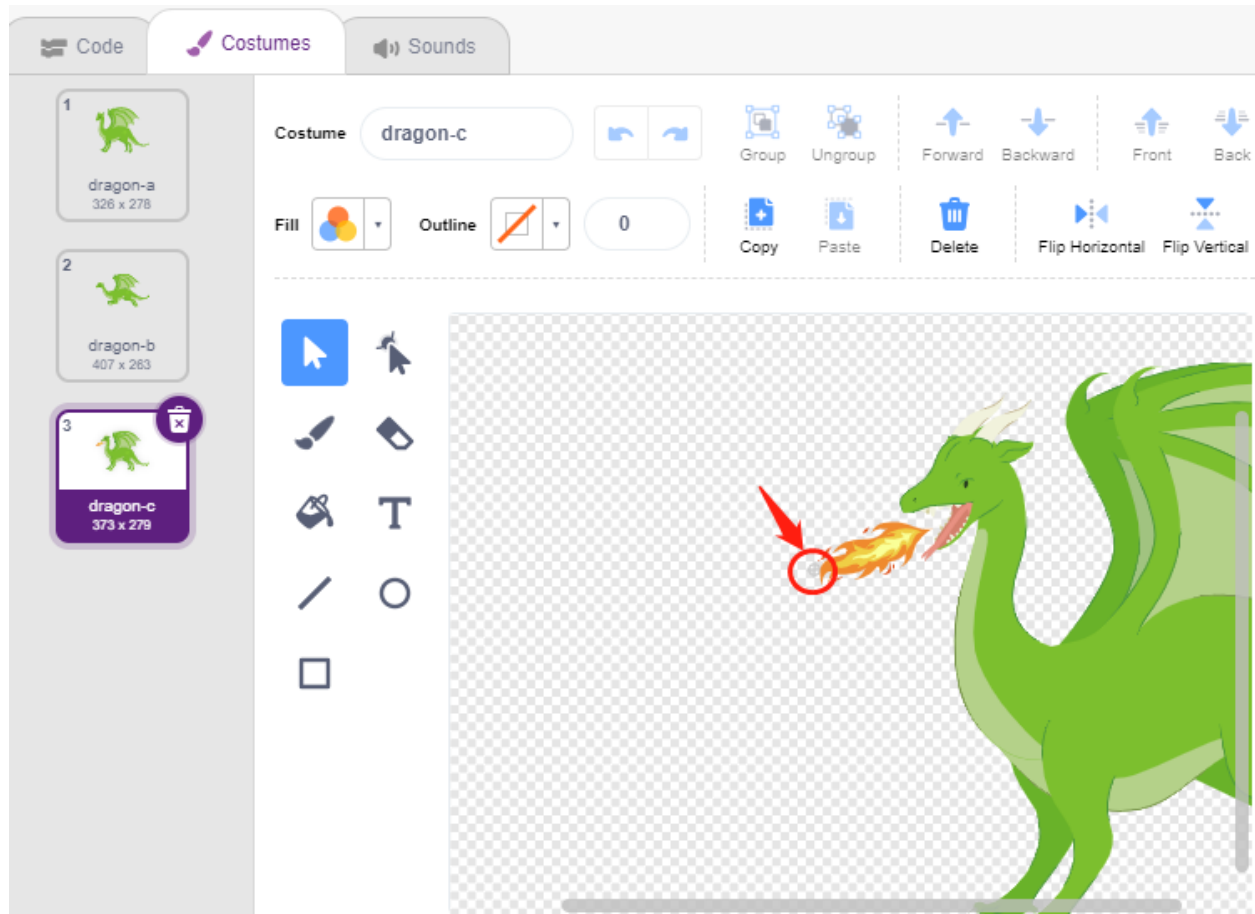
Bemerkung: Beim Anpassen des **Lightning**-Sprites können Sie es möglicherweise aus der Mitte bewegen, was vermieden werden muss! Der Mittelpunkt muss genau in der Mitte des Sprites liegen!



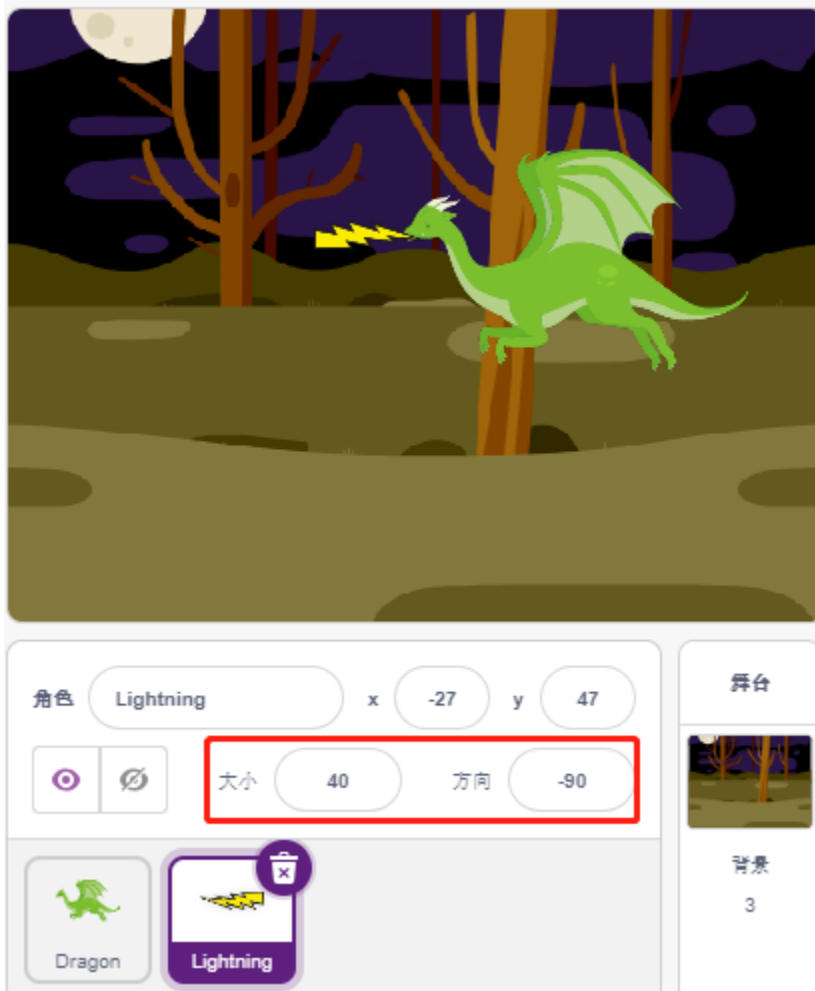
- Dann justiere das **dragon-c** Kostüm des **Dragon** Sprites, so dass sein Mittelpunkt am Schwanz des Feuers liegen sollte. Dies stellt sicher, dass die Positionen des **Dragon** Sprites und des **Lightning** Sprites korrekt sind und verhindert, dass der **Lightning** aus den Füßen des Drachen herausgeschossen wird.



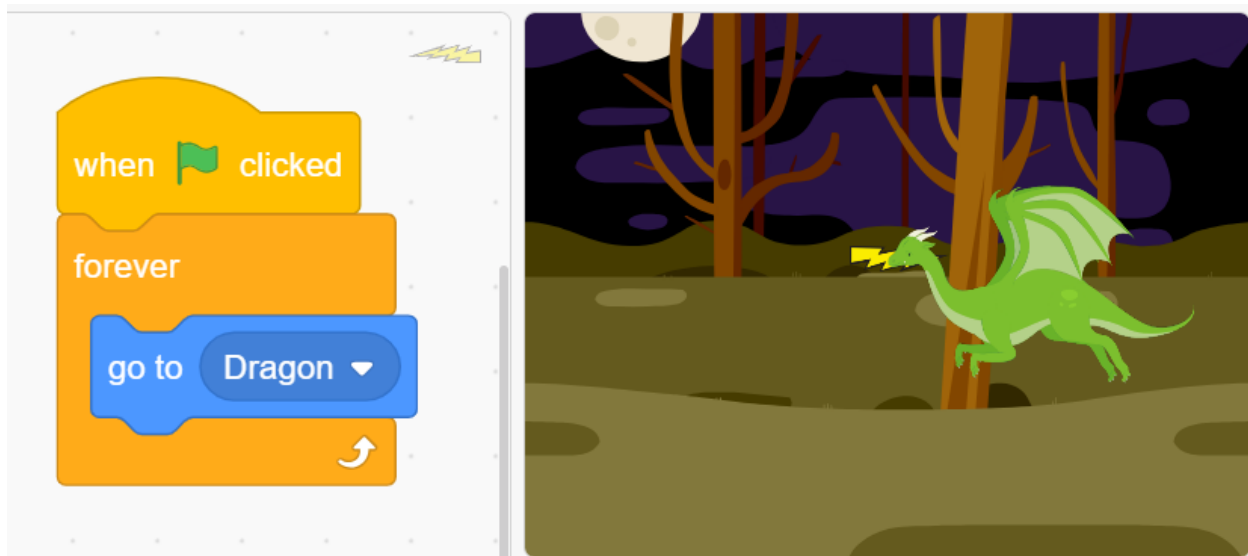
- Entsprechend muss bei **dragon-b** der Kopf des Drachen mit dem Mittelpunkt übereinstimmen.



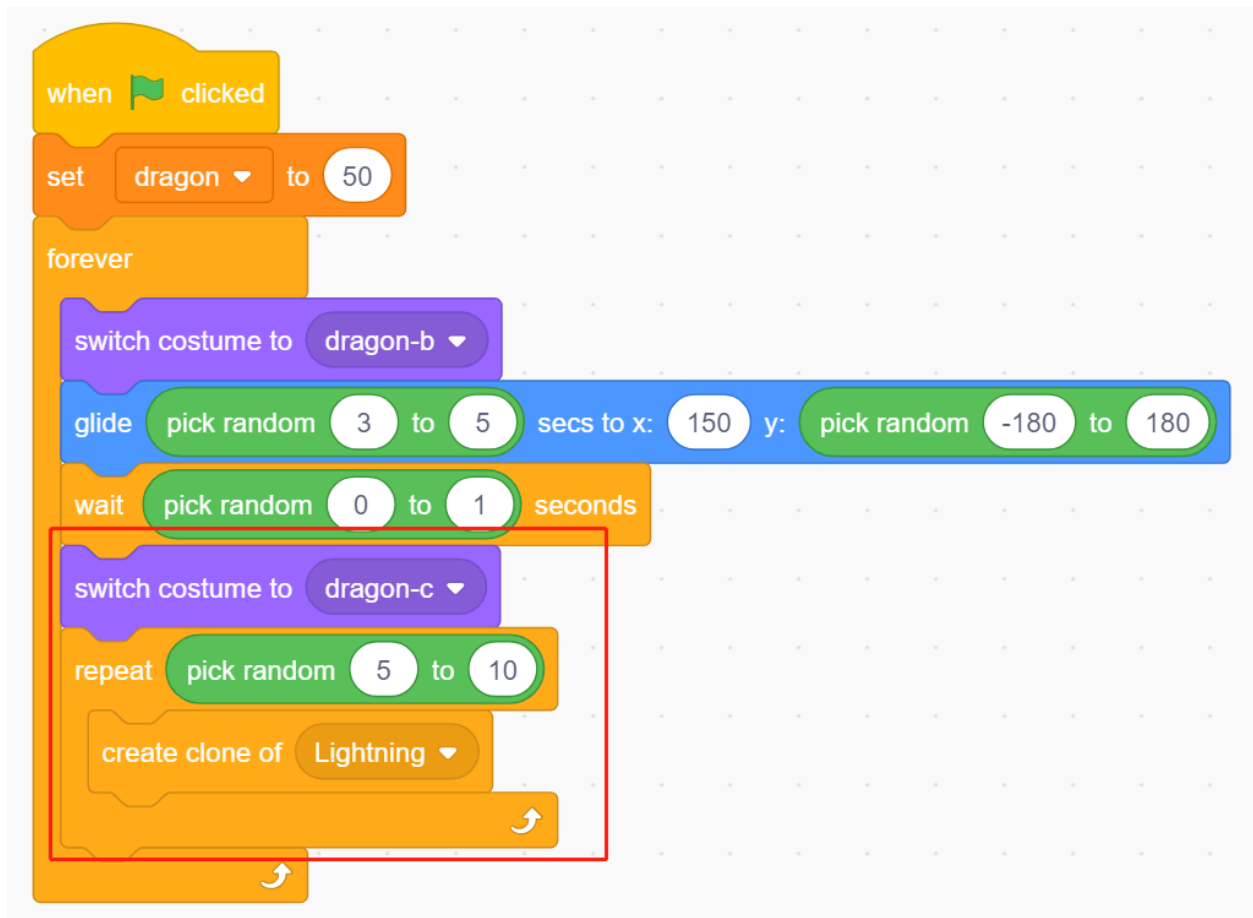
- Passen Sie die Größe und Ausrichtung des **Lightning** Sprites an, um das Bild harmonischer wirken zu lassen.



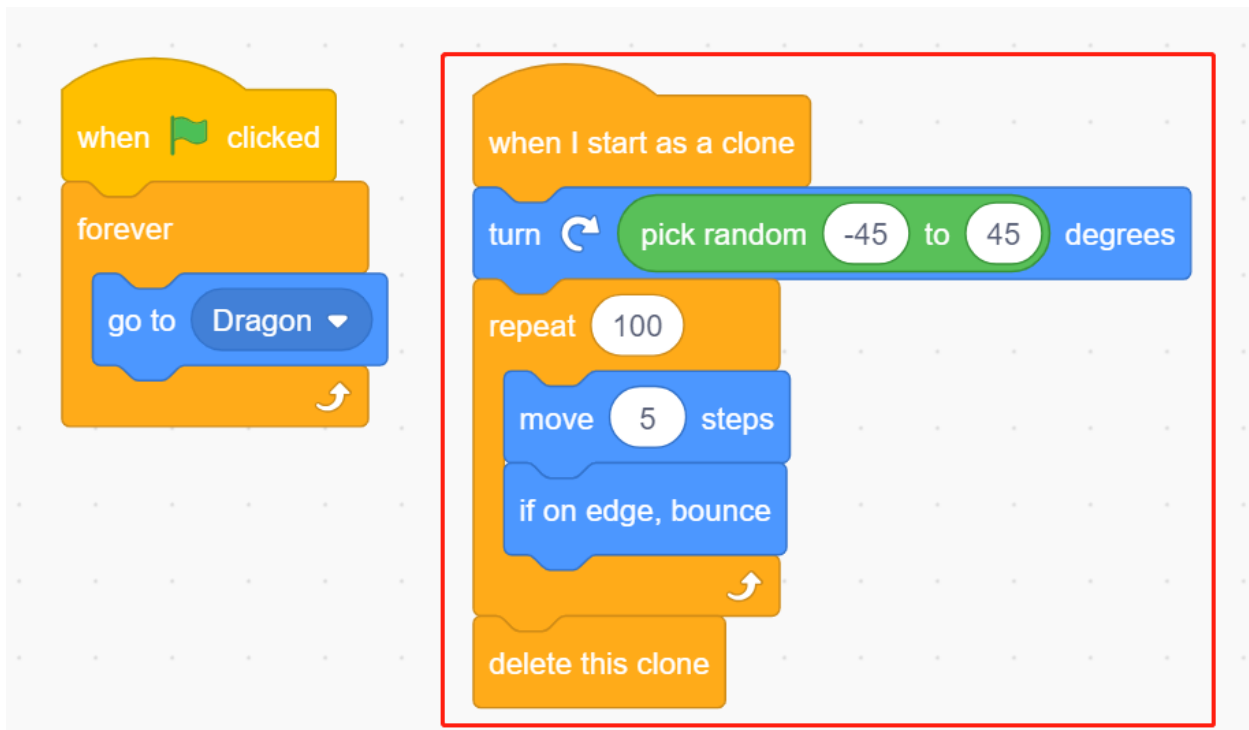
- Jetzt das Skript für den **Lightning** Sprite erstellen. Das ist einfach, er sollte dem **Dragon** Sprite immer folgen. Zu diesem Zeitpunkt, wenn Sie auf die grüne Flagge klicken, werden Sie sehen, wie der **Dragon** sich mit einem Blitz im Mund bewegt.



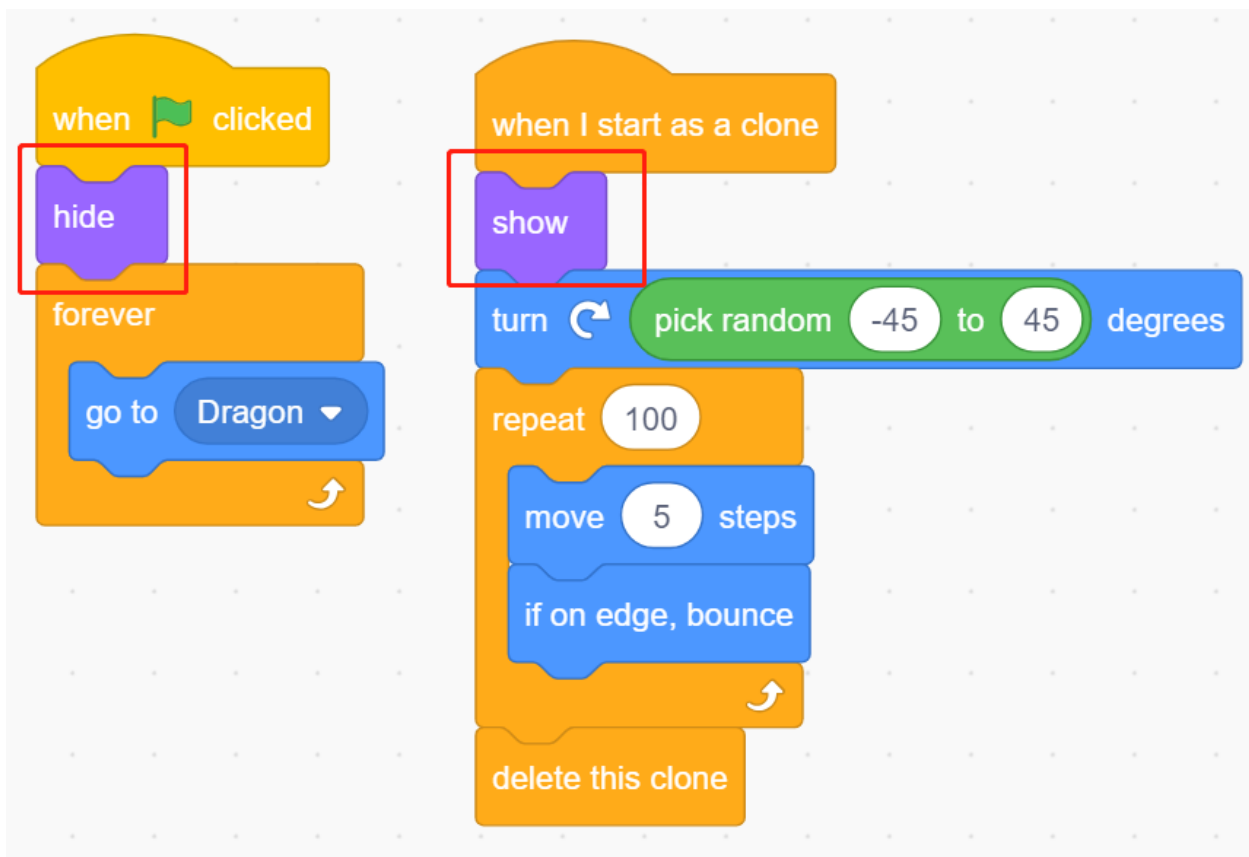
- Zurück zum **Dragon** Sprite, lassen Sie ihn jetzt Feuer speien, wobei Sie darauf achten müssen, dass das Feuer nicht aus seinem Mund schießt, sondern einen Klon für den **Lightning** Sprite erstellt.



- Klicken Sie auf den **Lightning** Sprite und lassen Sie den **Lightning** Klon in einem zufälligen Winkel schießen. Er wird von der Wand abprallen und nach einer bestimmten Zeit verschwinden.



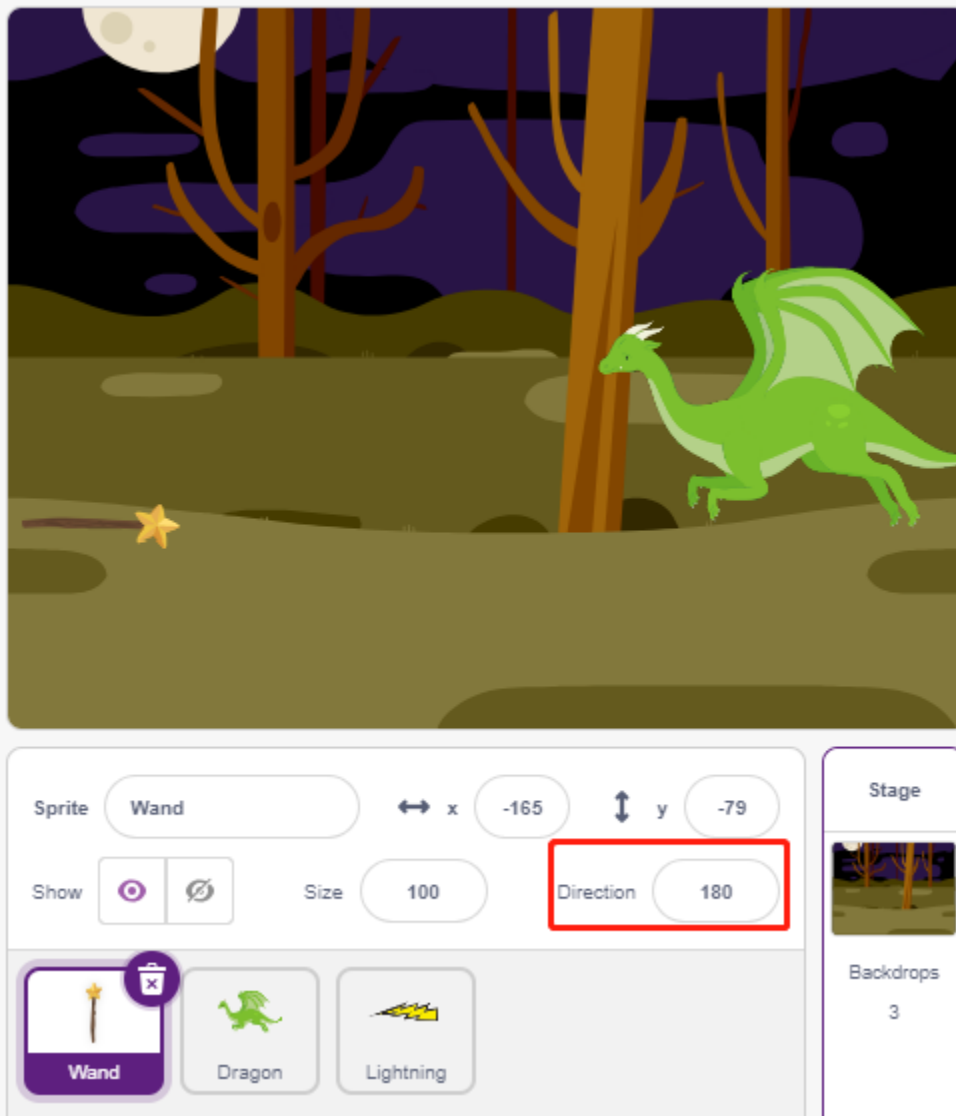
- Bei dem **Lightning** Sprite, verstecken Sie seinen Körper und zeigen den Klon.



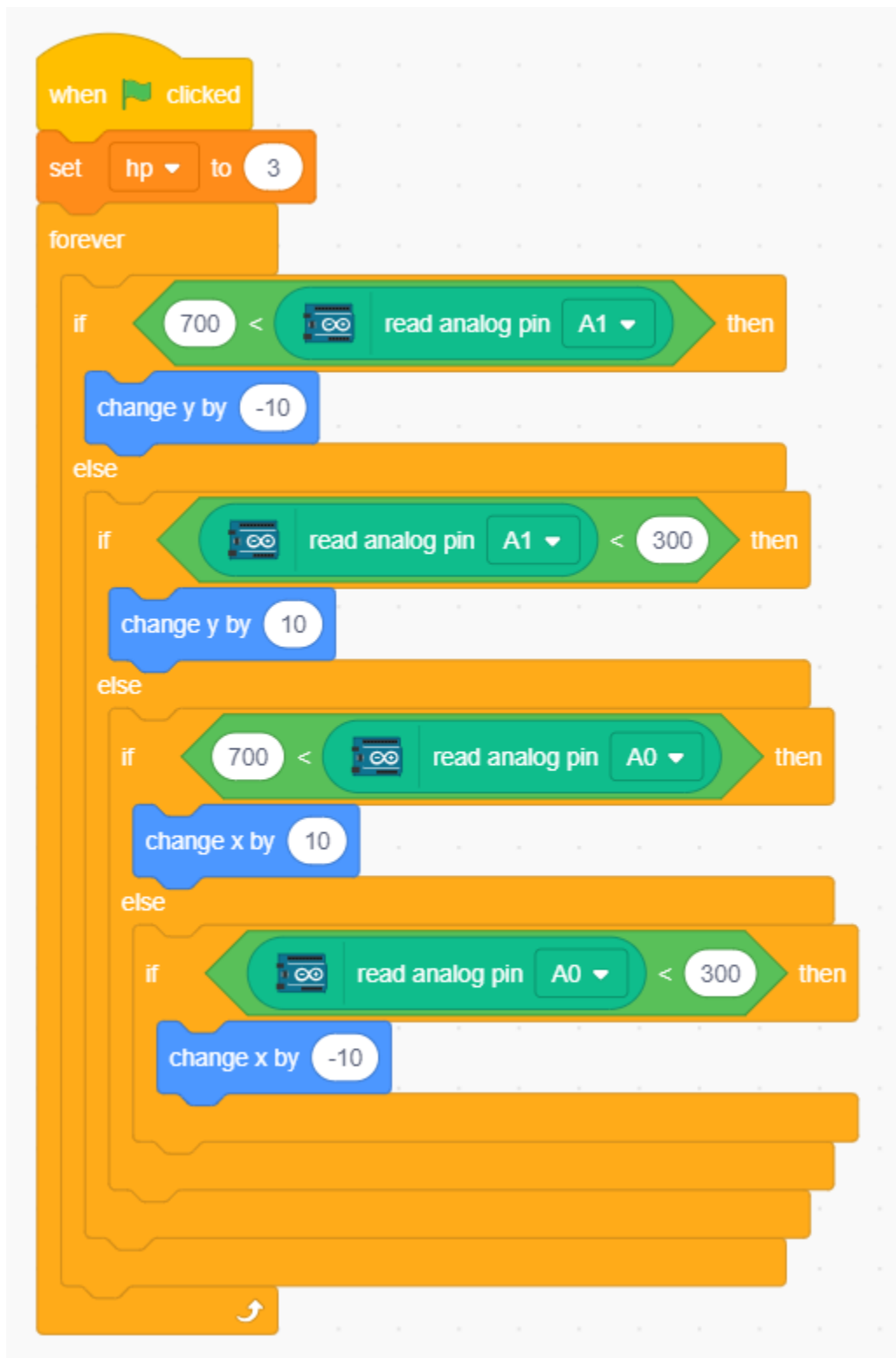
Jetzt kann der Drache auf und ab bewegen und Feuer speien.

2. Zauberstab

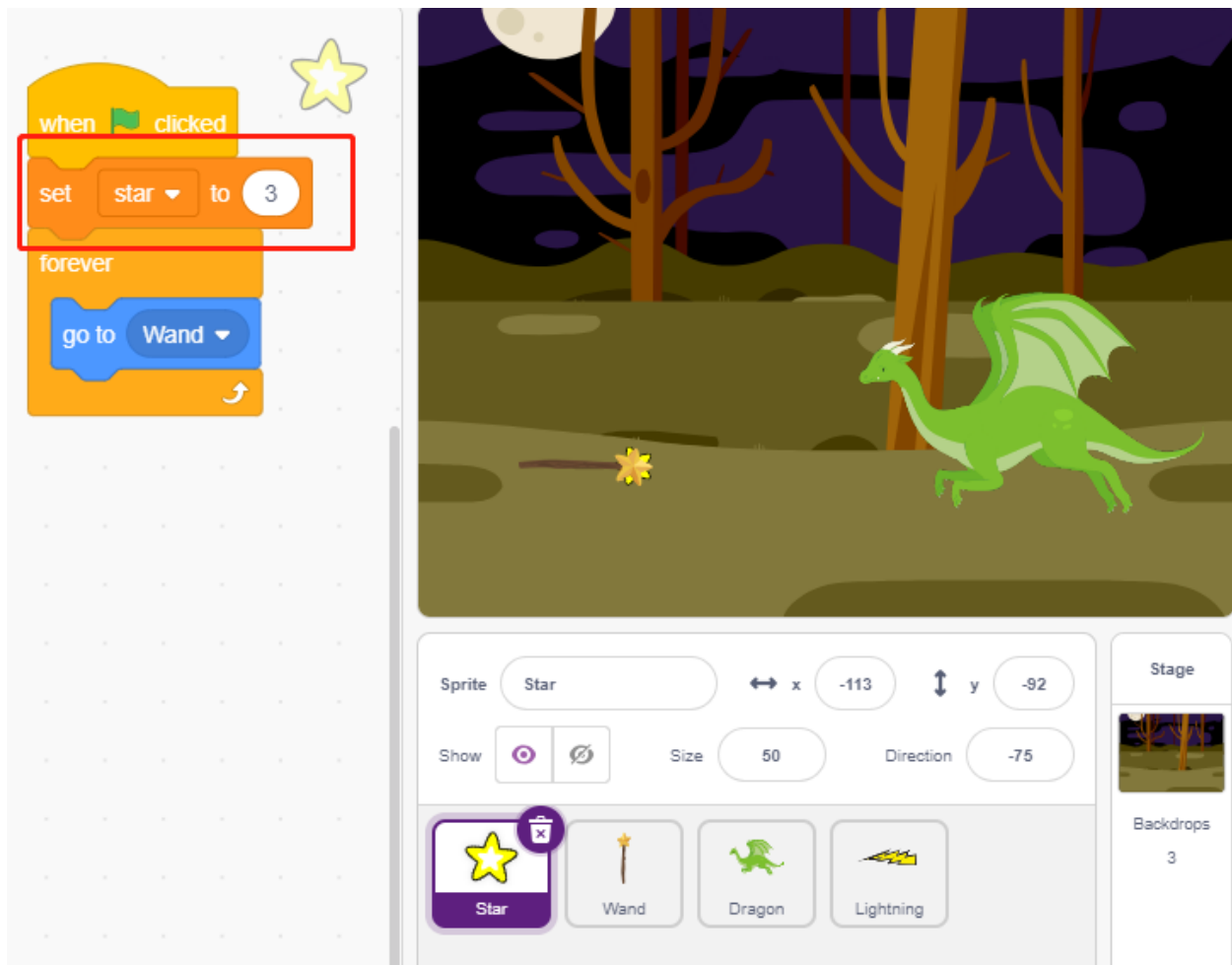
- Erstellen Sie einen **Wand** Sprite und drehen Sie seine Richtung auf 180 Grad, um nach rechts zu zeigen.



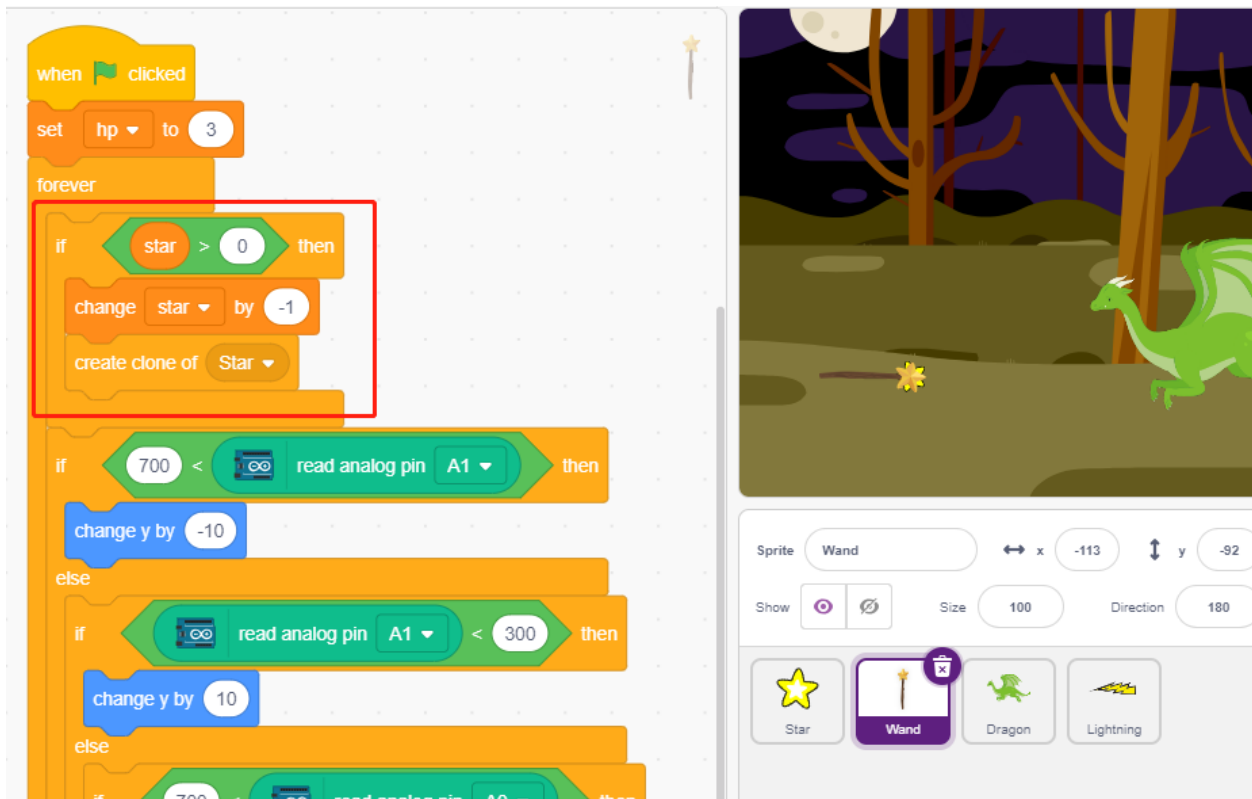
- Erstellen Sie jetzt eine Variable **hp**, um seinen Lebenswert zu erfassen, der zunächst auf 3 gesetzt wird. Dann lesen Sie den Wert des Joysticks, der zur Steuerung der Bewegung des Zauberstabs verwendet wird.



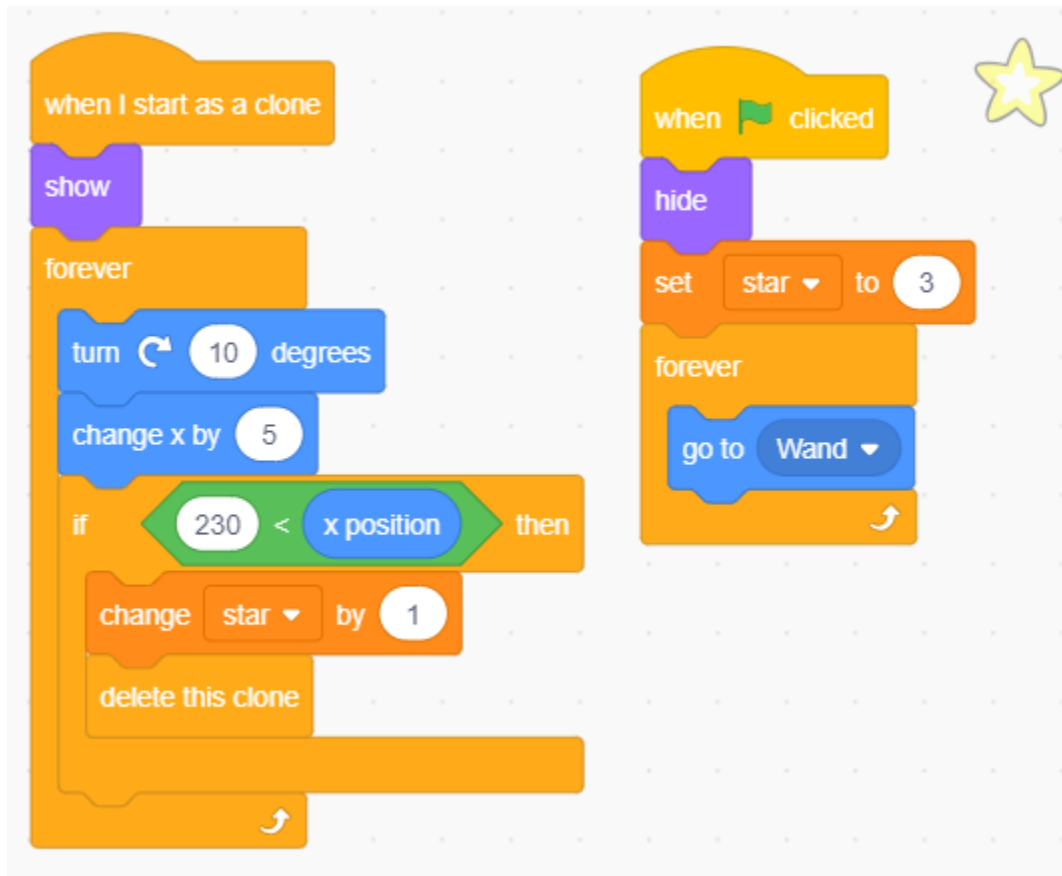
- Der Drache hat Blitze und der Zauberstab, der ihn zerstört, hat seinen „magischen Schuss“! Erstellen Sie einen **Star** Sprite, verkleinern Sie ihn und lassen Sie ihn immer dem **Wand** Sprite folgen, wobei die Anzahl der Sterne auf drei begrenzt ist.



- Lassen Sie den **Wand** Sprite automatisch Sterne schießen. Der **Wand** Sprite schießt Sterne auf die gleiche Weise, wie der Drache Feuer speit – indem Klone erstellt werden.



- Gehen Sie zurück zum **Star** Sprite und erstellen Sie ein Skript für seinen Klon, um sich drehend nach rechts zu schießen, verschwindet nach dem Verlassen der Bühne und stellt die Anzahl der Sterne wieder her. Wie beim **Lightning** Sprite, verstecken Sie den Körper und zeigen den Klon.



Jetzt haben wir einen Zauberstab, der Sternenkugeln schießt.

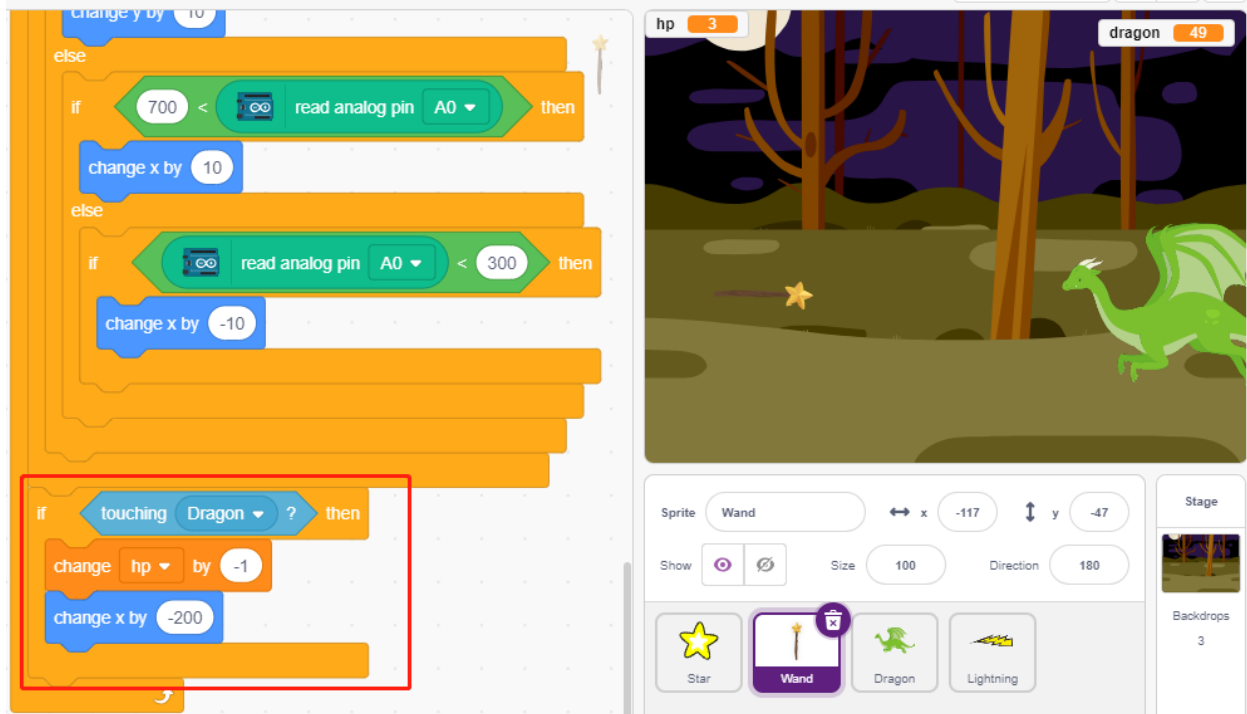
3. Kampf!

Der Zauberstab und der Drache stehen momentan noch im Widerspruch zueinander, und wir werden sie kämpfen lassen. Der Drache ist stark, und der Zauberstab ist der mutige Mann, der gegen den Drachen kämpft. Die Interaktion zwischen ihnen besteht aus den folgenden Teilen.

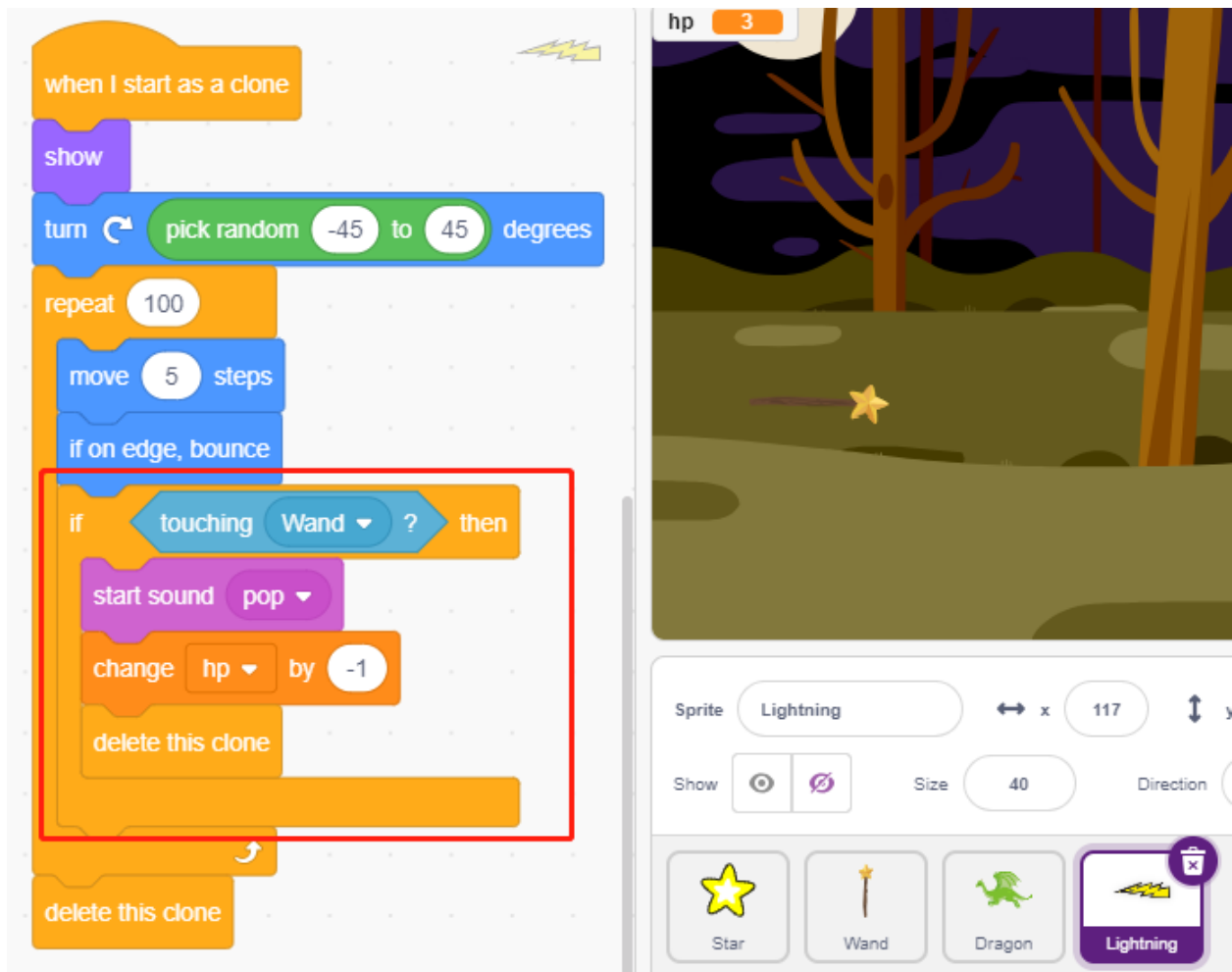
1. Wenn der Zauberstab den Drachen berührt, wird er zurückgestoßen und verliert Lebenspunkte.
2. Wenn ein Blitz den Zauberstab trifft, verliert der Zauberstab Lebenspunkte.
3. Wenn die Sternenkugel den Drachen trifft, verliert der Drache Lebenspunkte.

Sobald das geklärt ist, ändern wir die Skripte für jeden Sprite.

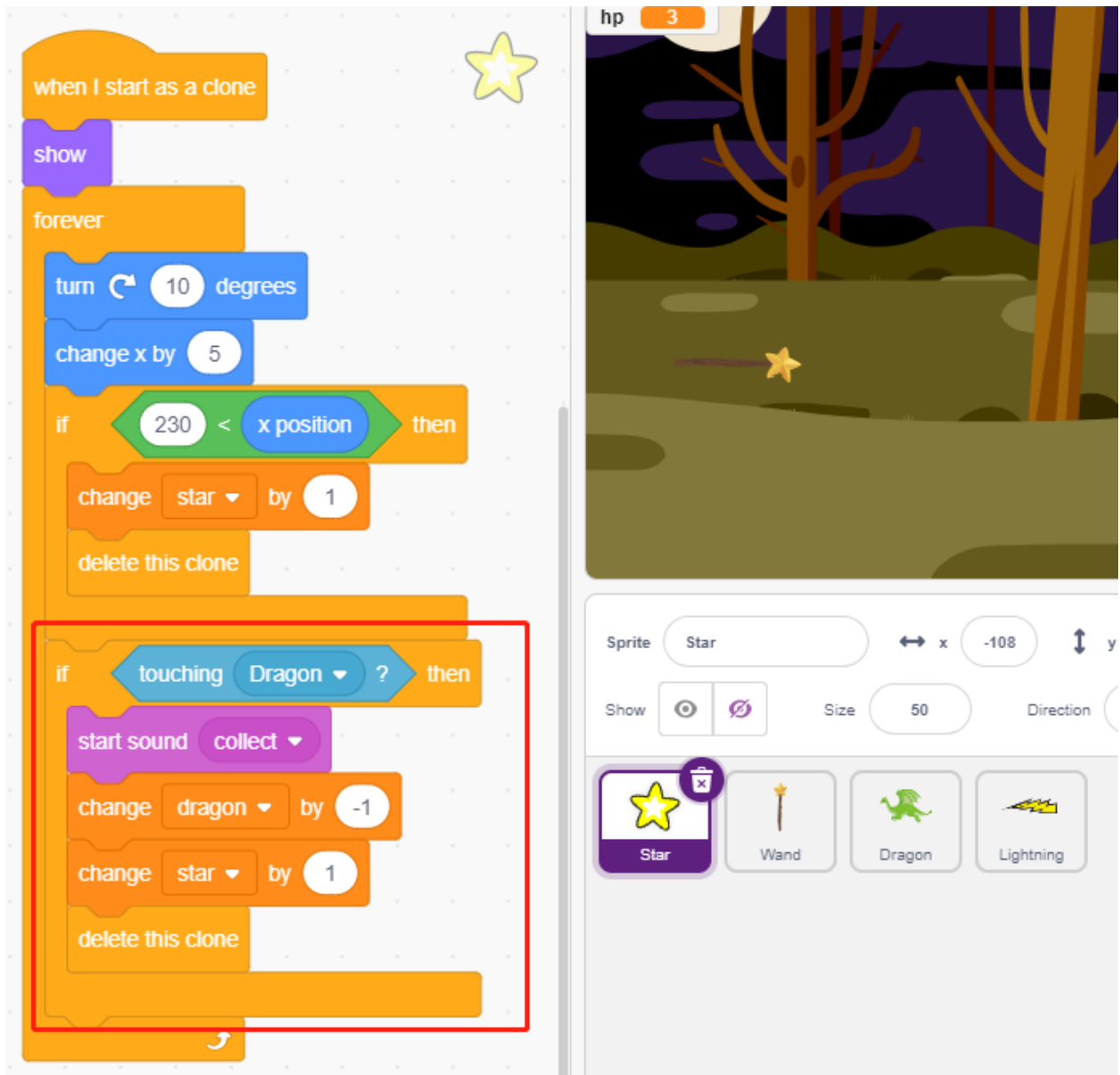
- Wenn der **Wand** den **Dragon** trifft, wird er zurückgestoßen und verliert Lebenspunkte.



- Wenn **Lightning** (ein **Lightning** Sprite Klon) den **Wand** Sprite trifft, gibt er ein Knallgeräusch von sich und verschwindet, und der **Wand** verliert Lebenspunkte.



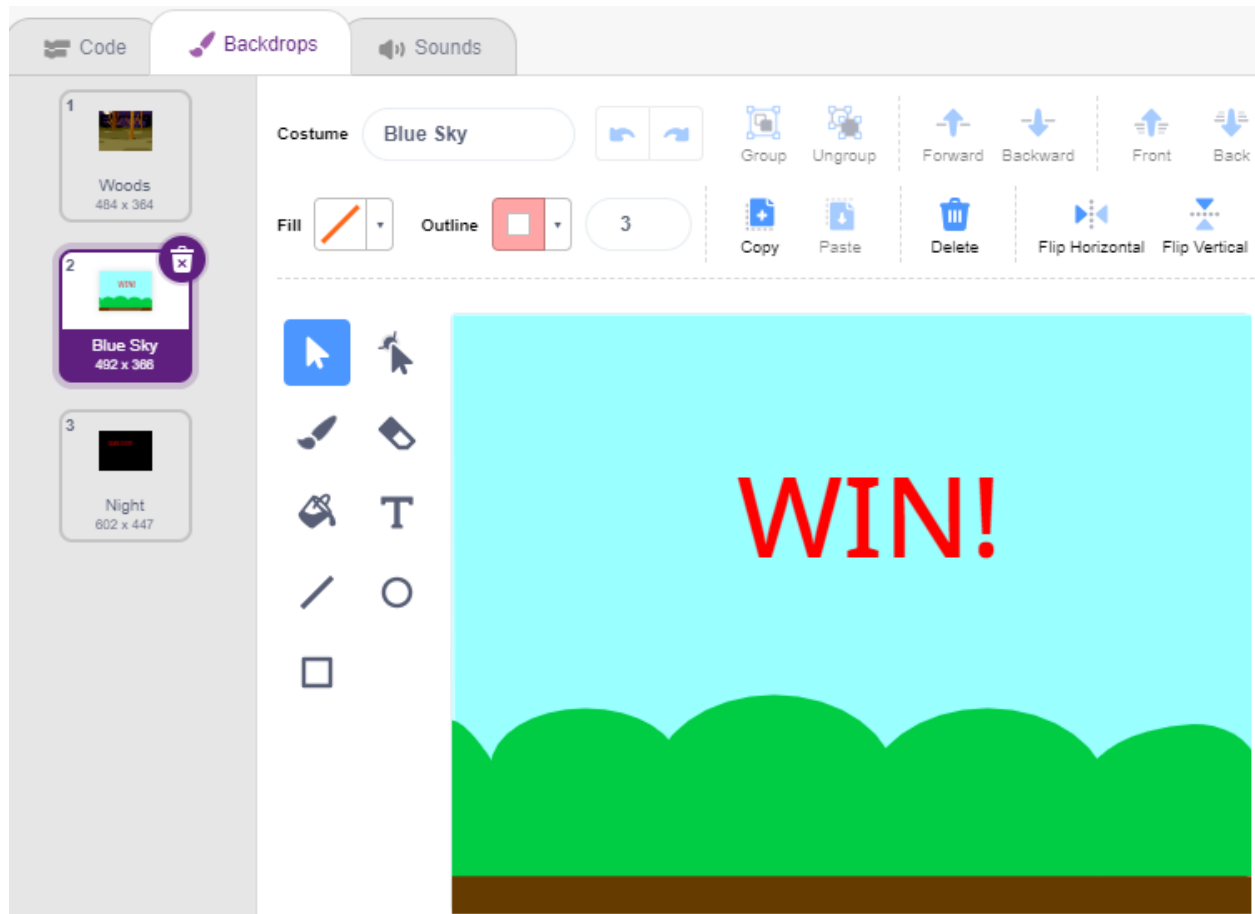
- Wenn ein **Star** (Klon des **Star** Sprites) den **Dragon** trifft, gibt er ein Sammelgeräusch von sich und verschwindet, stellt dabei die **Star** Anzahl wieder her, und der **Dragon** verliert Lebenspunkte.



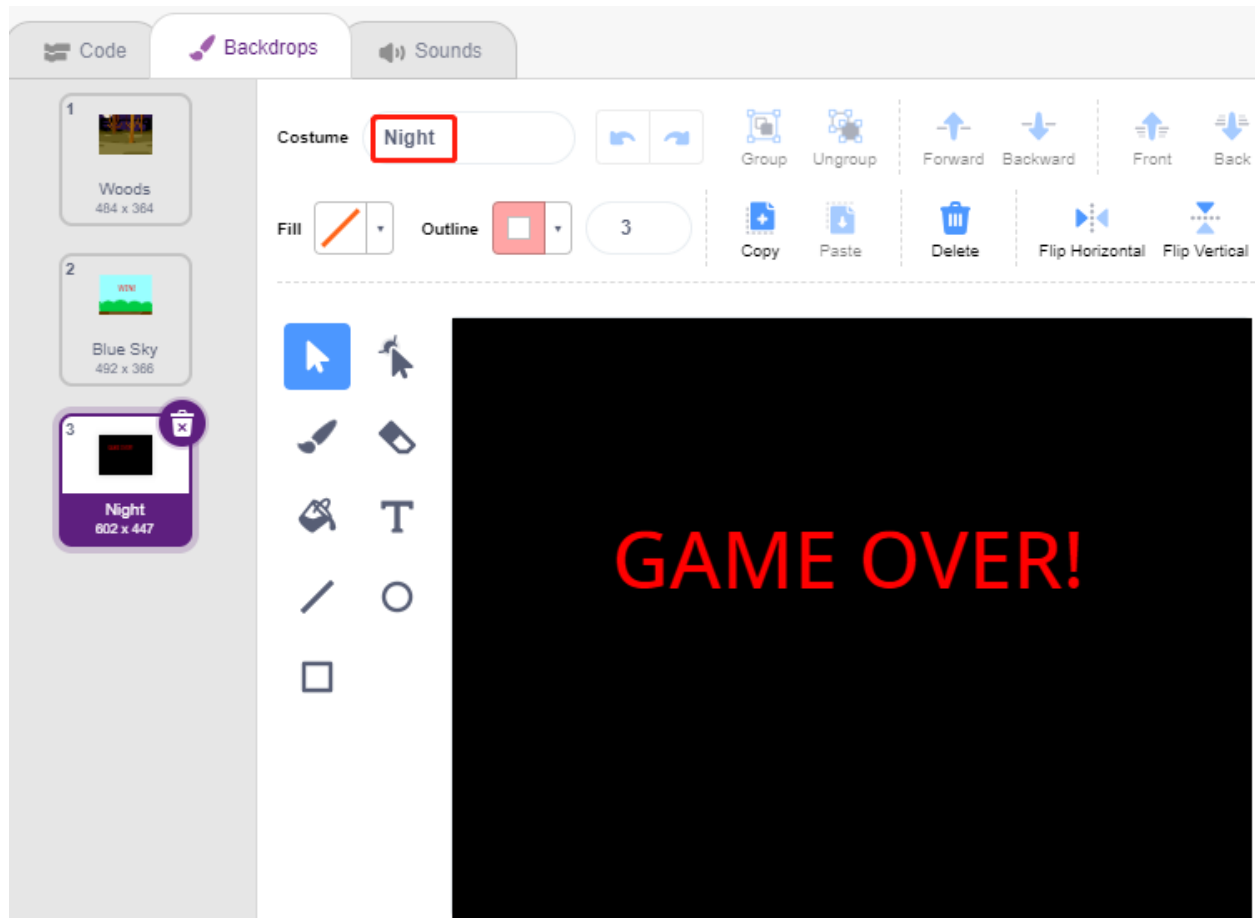
4. Bühne

Die Schlacht zwischen dem **Wand** und dem **Dragon** wird letztlich in Gewinner und Verlierer unterteilt, was wir mit der Bühne darstellen.

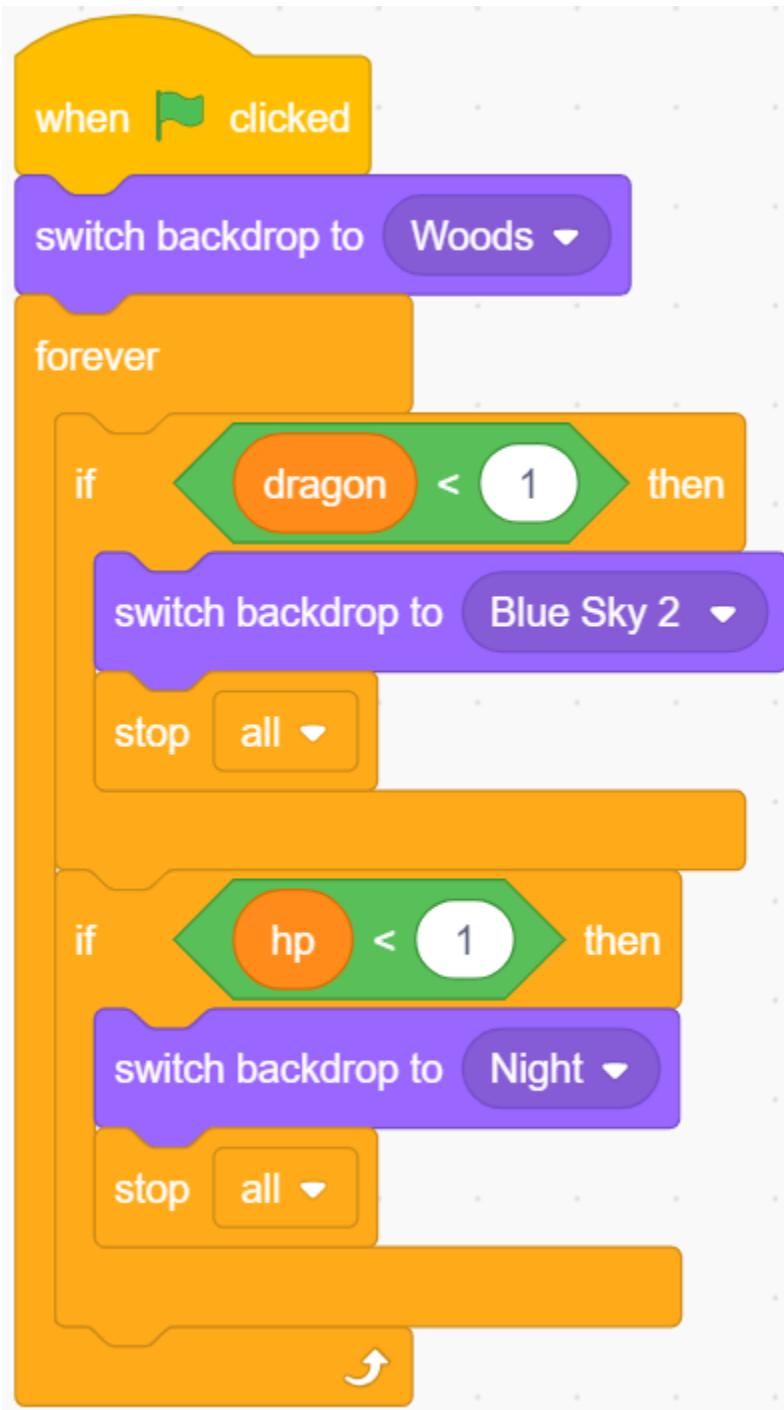
- Fügen Sie den **Blue Sky** Hintergrund hinzu und schreiben Sie das Wort „GEWONNEN!“ darauf, um zu zeigen, dass der Drache besiegt wurde und der Morgen angebrochen ist.



- Und ändern Sie den leeren Hintergrund wie folgt, um zu zeigen, dass das Spiel verloren wurde und alles in Dunkelheit liegt.



- Schreiben Sie jetzt ein Skript, um diese Hintergründe zu wechseln: Wenn die grüne Flagge angeklickt wird, wechseln Sie zum **Woods** Hintergrund; wenn die Lebenspunkte des Drachen weniger als 1 betragen, ist das Spiel gewonnen und wechselt zum **Blue Sky** Hintergrund; wenn die Lebenspunkte des **Wand** weniger als 1 betragen, wechseln Sie zum **Night** Hintergrund und das Spiel ist verloren.



3. Auto mit Scratch steuern

Die folgenden Projekte sind nach ihrer Programmierschwierigkeit geordnet. Es wird empfohlen, diese Bücher in der gegebenen Reihenfolge durchzugehen.

In jedem Projekt gibt es sehr detaillierte Schritte, die Ihnen beibringen, wie Sie den Schaltkreis aufbauen und wie Sie ihn Schritt für Schritt programmieren können, um das gewünschte Ergebnis zu erzielen.

Natürlich können Sie das Skript auch direkt ausführen. Stellen Sie jedoch sicher, dass Sie das relevante Material von [github](#) heruntergeladen haben.

Nach dem Download entpacken Sie es. Beachten Sie *Hochlade-Modus*, um einzelne Skripte direkt auszuführen.

8.26 3.1 Das Auto testen

Hier erfahren Sie, wie Sie Skripte schreiben, um das Auto vorwärts zu bewegen. Dennoch sollten Sie sich auf *Auto Projekte* beziehen, um das Auto zusammenzubauen und ein grundlegendes Verständnis dafür zu erlangen.

Bevor Sie mit dem Projekt beginnen, sollten Sie sich mit den Schritten vertraut machen, um PictoBlox im *Hochlade-Modus* zu verwenden.

8.26.1 Benötigte Komponenten

Für dieses Projekt benötigen Sie folgende Komponenten:

Ein komplettes Set zu erwerben, ist sicherlich praktisch. Hier ist der Link:

Name	ARTIKEL IM SET	LINK
3-in-1 Starter Set	380+	

Alternativ können Sie die Teile einzeln über die folgenden Links erwerben:

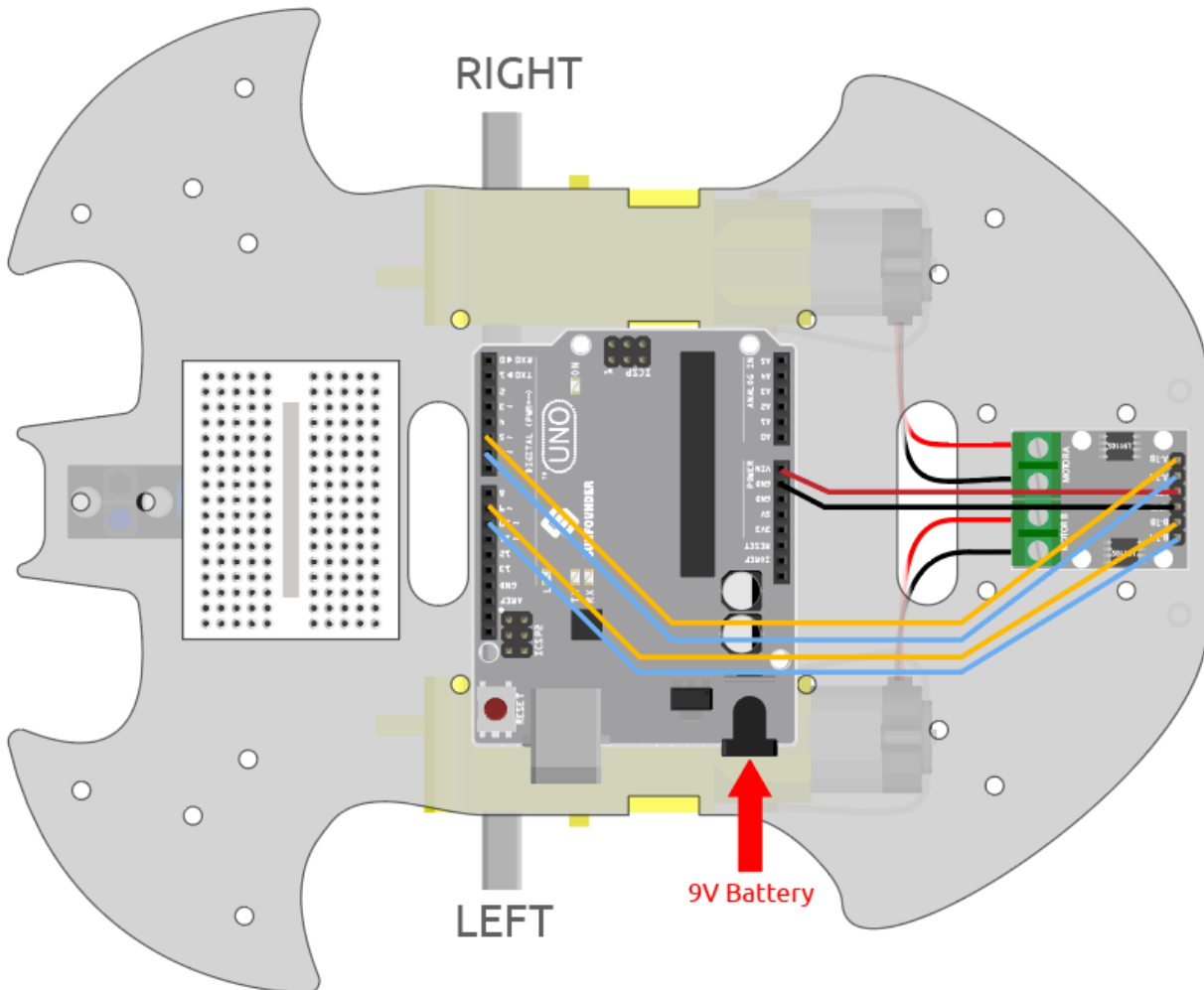
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-

8.26.2 Schaltkreis aufbauen

Das L9110 Motortreibermodul ist ein Hochleistungsmotortreiber für Gleichstrom- und Schrittmotoren. Mit dem L9110-Modul können bis zu 4 Gleichstrommotoren oder 2 Gleichstrommotoren mit Richtungs- und Geschwindigkeitskontrolle gesteuert werden.

Verbinden Sie die Drähte zwischen dem L9110-Modul und dem R3-Board gemäß dem untenstehenden Diagramm.

L9110 Modul	R3 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

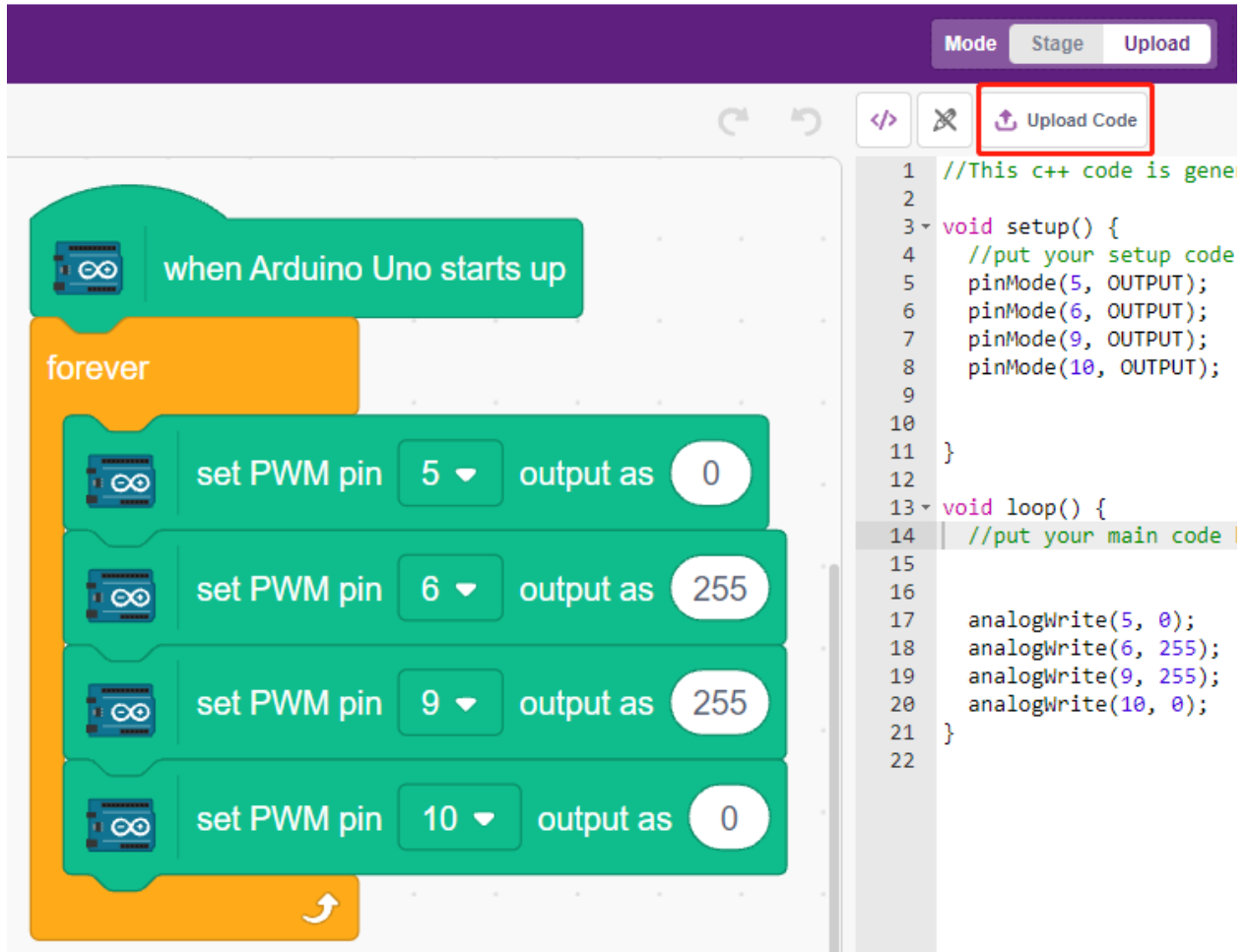


8.26.3 Programmierung

1. Das Auto vorwärts bewegen

Aufgrund der obigen Verkabelung wissen wir, dass die Pins 5 und 6 zur Steuerung der rechten Motorrotation und die Pins 9 und 10 für die linke Motorrotation verwendet werden. Schreiben Sie jetzt ein Skript, damit das Auto vorwärts fährt.

Nachdem Sie das Board als Arduino Uno ausgewählt haben, wechseln Sie in den *Hochlade-Modus* und schreiben das Skript entsprechend dem folgenden Diagramm.



Klicken Sie auf die Schaltfläche **Upload Code**, um den Code auf das R3-Board hochzuladen. Sobald dies erledigt ist, werden die beiden Motoren des Autos vorwärts bewegt (wenn Sie das Auto auf den Boden setzen, wird es in einer geraden Linie vorwärts fahren. Es kann jedoch sein, dass das Auto eine Kurve fährt, da die Geschwindigkeit der beiden Motoren leicht unterschiedlich ist).

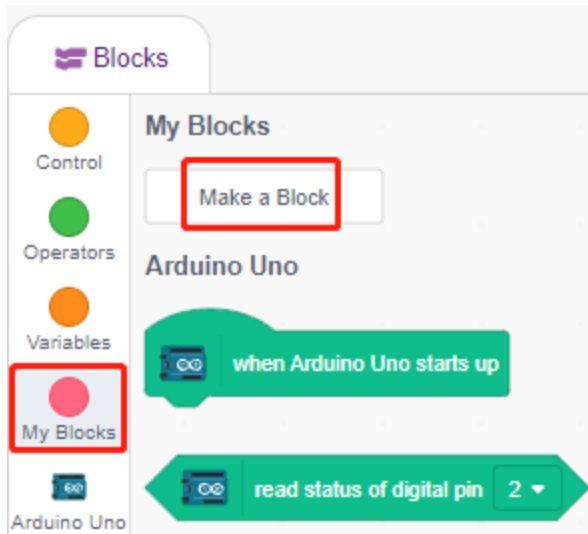
Wenn nicht beide Motoren vorwärts drehen, sondern die folgenden Situationen auftreten, müssen Sie die Verkabelung der beiden Motoren neu anpassen.

- Wenn beide Motoren gleichzeitig rückwärts drehen (linker Motor dreht 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.

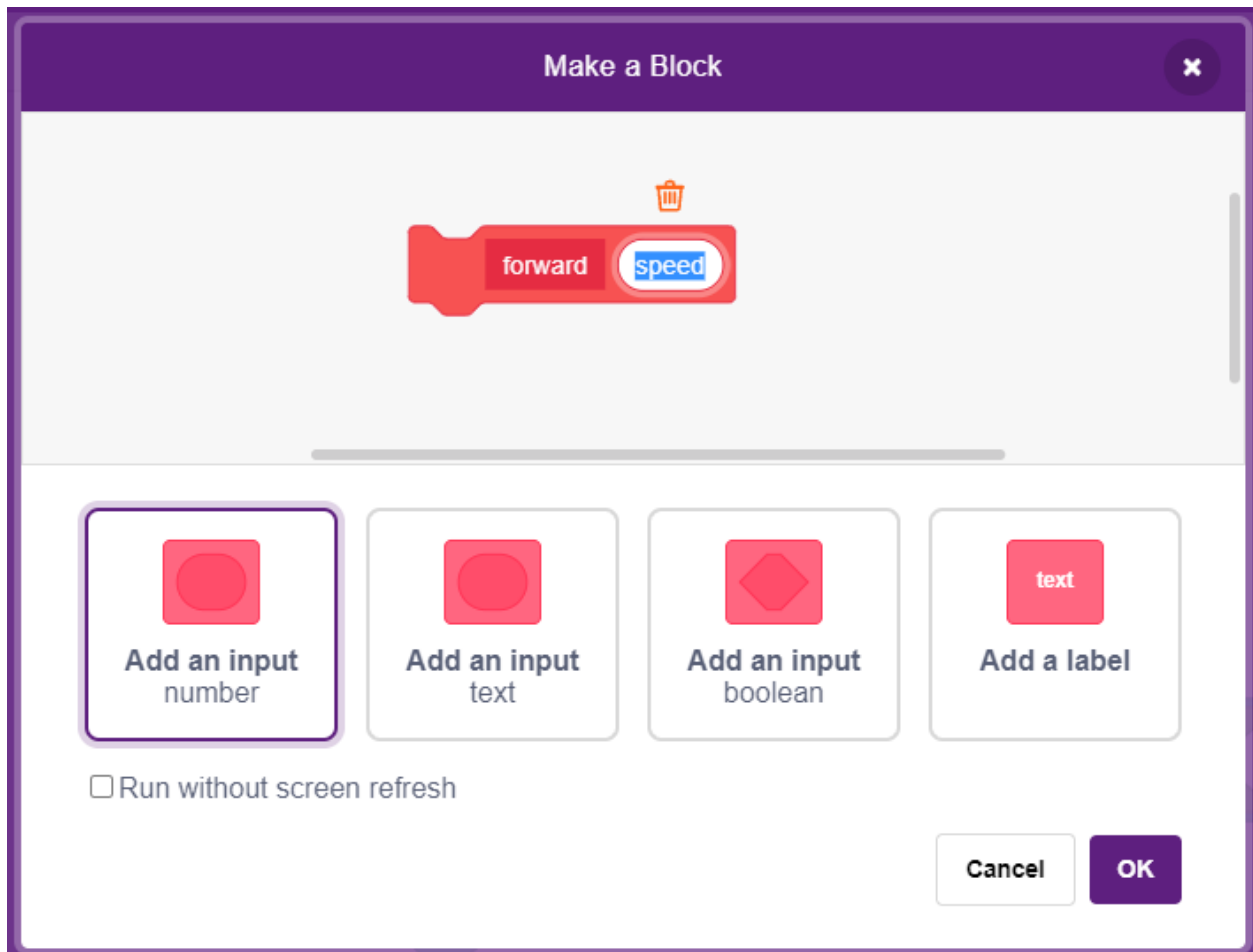
2. Block erstellen

Um das Skript übersichtlicher und einfacher zu gestalten, fügen wir hier alle Blöcke, die die Vorwärtsbewegung steuern, in einen Block ein. Wenn Sie ihn verwenden möchten, rufen Sie diesen Block direkt auf.

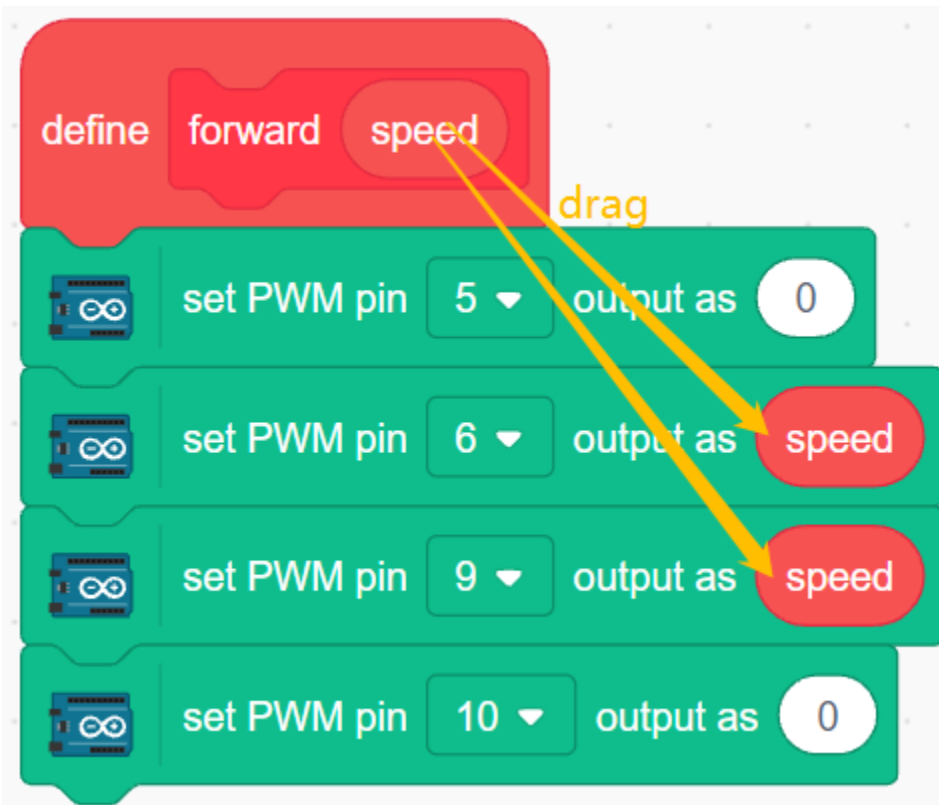
Klicken Sie in der **My Blocks**-Palette auf **My Blocks**.



Geben Sie den Namen des Blocks - **forward** ein und aktivieren Sie **Add an input**, setzen Sie den Eingabenamen auf **speed**.

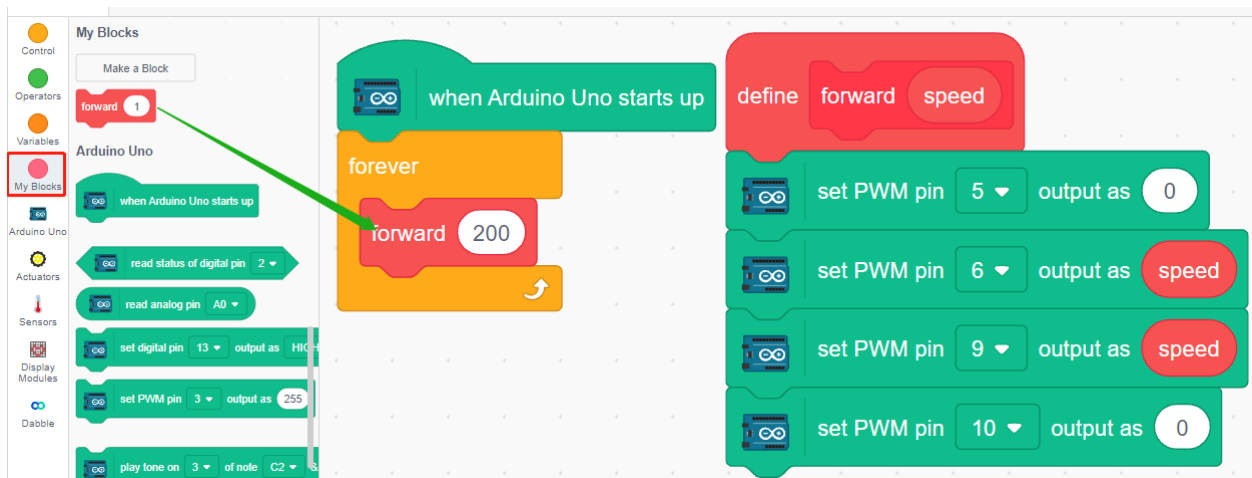


Ziehen Sie die Blöcke, die die Vorwärtsbewegung des Autos steuern, in **forward**. Beachten Sie, dass Sie den Parameter - **speed** zu den Pins 6 und 9 hinzufügen müssen.



Rufen Sie den erstellten Block im [Forward]-Block - **forward** auf. Im Upload-Modus muss am Anfang der Block [When Arduino Uno starts up] hinzugefügt werden.

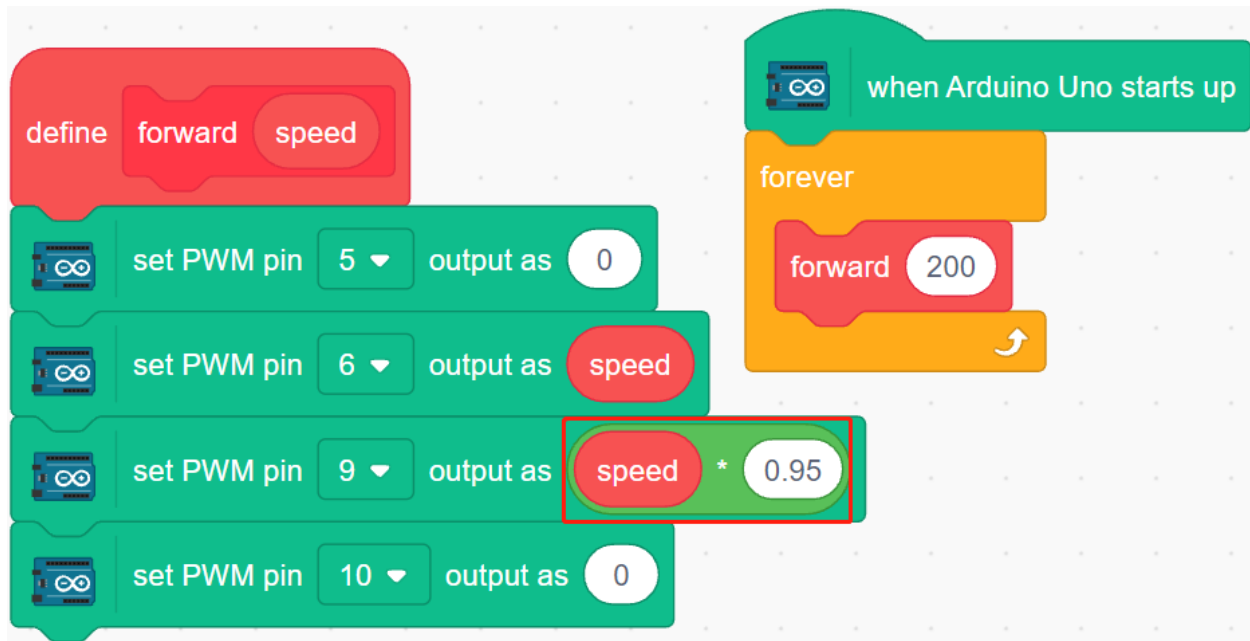
- Der Geschwindigkeitsbereich der Motorrotation liegt zwischen 100 und 255.



3. Geschwindigkeit der Motoren anpassen

Da es möglicherweise einen leichten Unterschied in der Geschwindigkeit der 2 Motoren gibt, wodurch das Auto nicht in einer geraden Linie fahren kann, können wir den linken und rechten Motoren unterschiedliche Geschwindigkeiten geben, um das Auto so weit wie möglich in einer geraden Linie zu bewegen.

Da mein Auto langsam nach rechts vorne fährt, reduziere ich hier die Geschwindigkeit des linken Motors.



8.27 3.2 Bewegung

Dieses Projekt basiert auf [3.1 Das Auto testen](#), um das Auto in alle Richtungen bewegen zu lassen.

Bevor wir mit der Programmierung beginnen, lassen Sie uns das Arbeitsprinzip des L9110-Moduls überprüfen.

Hier ist die Wahrheitstabelle für den Motor B:

B-1A	B-1B(B-2A)	Zustand des Motor B
1	0	Dreht im Uhrzeigersinn
0	1	Dreht gegen den Uhrzeigersinn
0	0	Bremse
1	1	Stopp

Hier ist die Wahrheitstabelle für den Motor A:

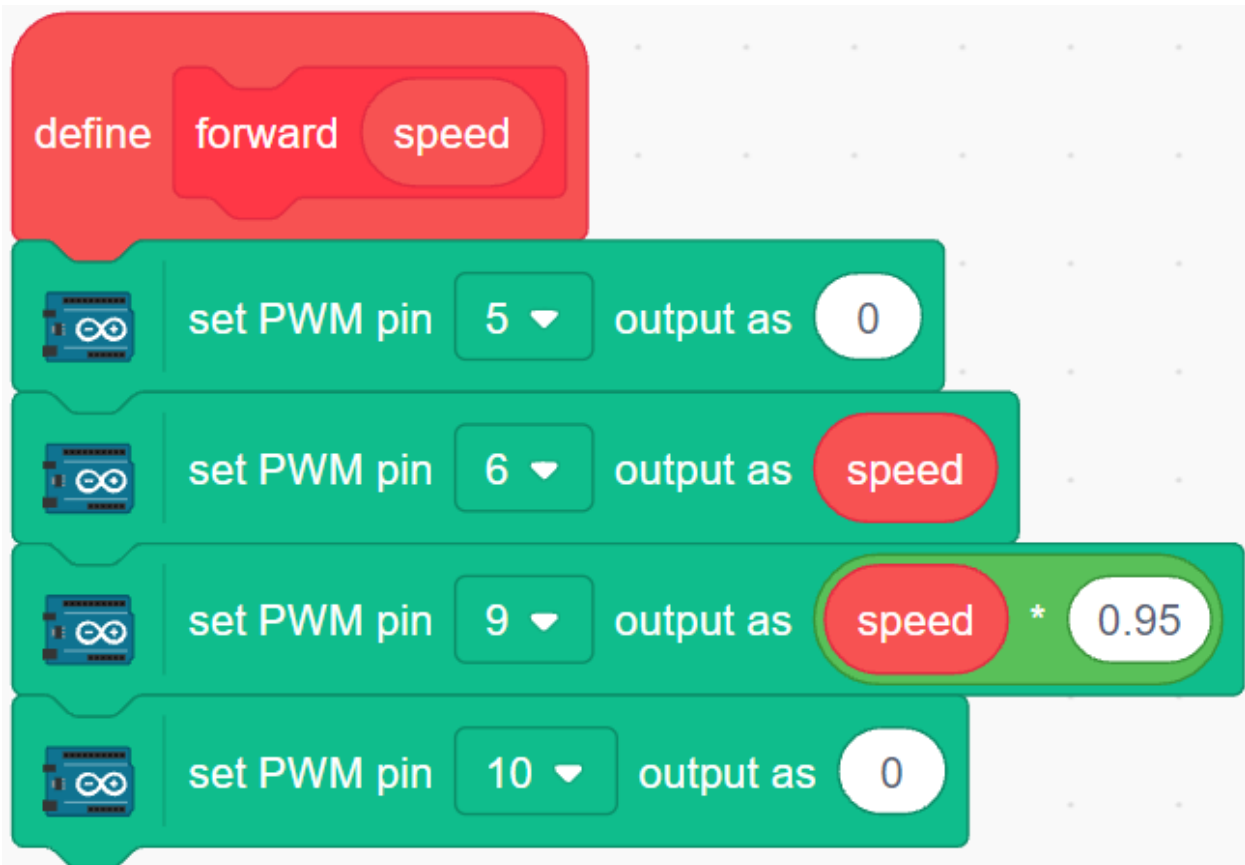
A-1A	A-1B	Zustand des Motor A
1	0	Dreht im Uhrzeigersinn
0	1	Dreht gegen den Uhrzeigersinn
0	0	Bremse
1	1	Stopp

8.27.1 Programmierung

Nun erstellen Sie Blöcke, um das Auto vorwärts, rückwärts, links und rechts zu bewegen und es jeweils anzuhalten.

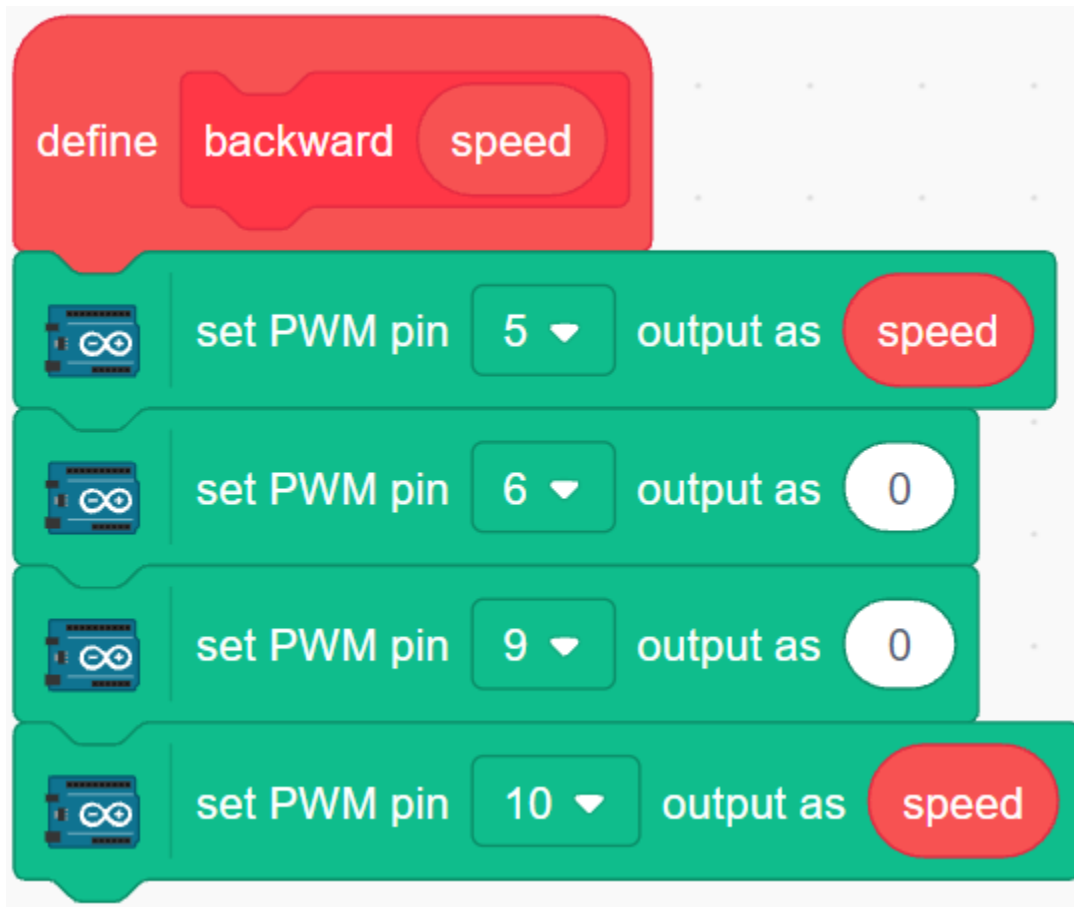
1. Vorwärts bewegen

Der rechte Motor dreht im Uhrzeigersinn und der linke Motor gegen den Uhrzeigersinn, um das Auto vorwärts zu bewegen.



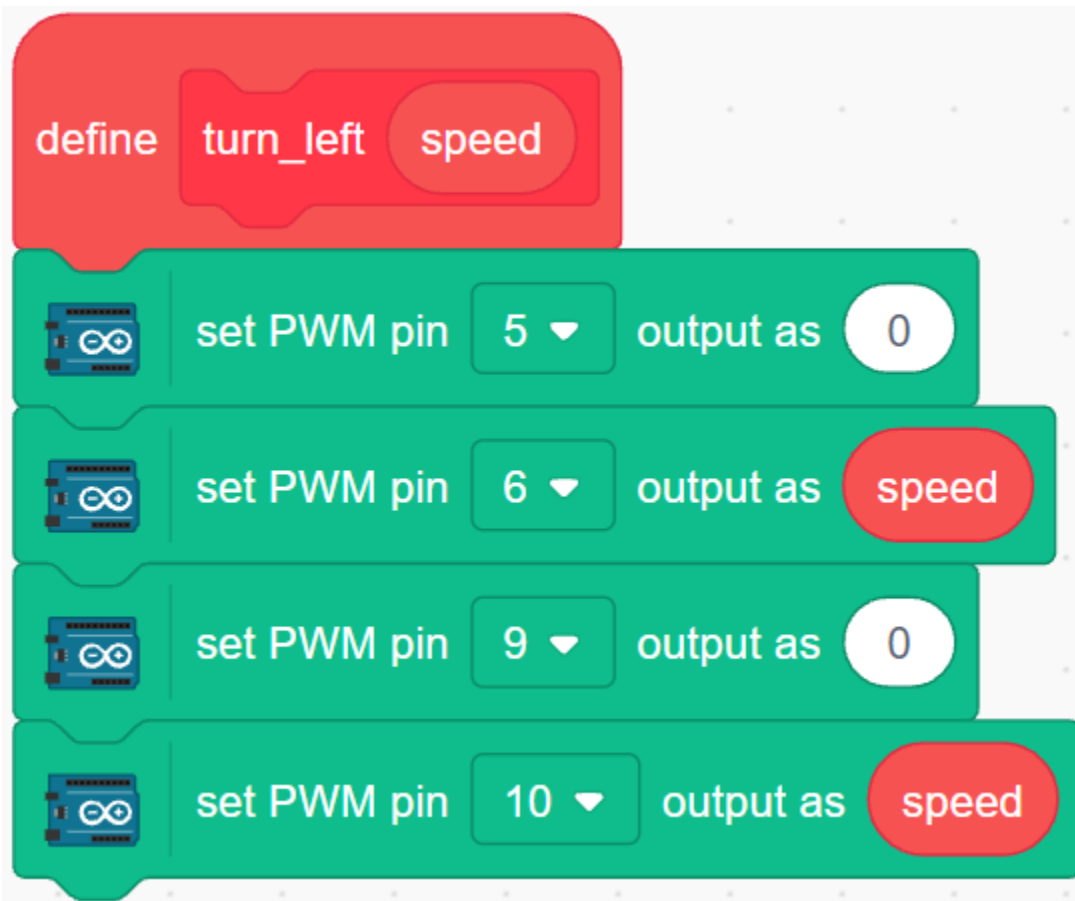
2. Rückwärts bewegen

Rückwärts ist genau das Gegenteil, der rechte Motor muss gegen den Uhrzeigersinn drehen, der linke Motor im Uhrzeigersinn.



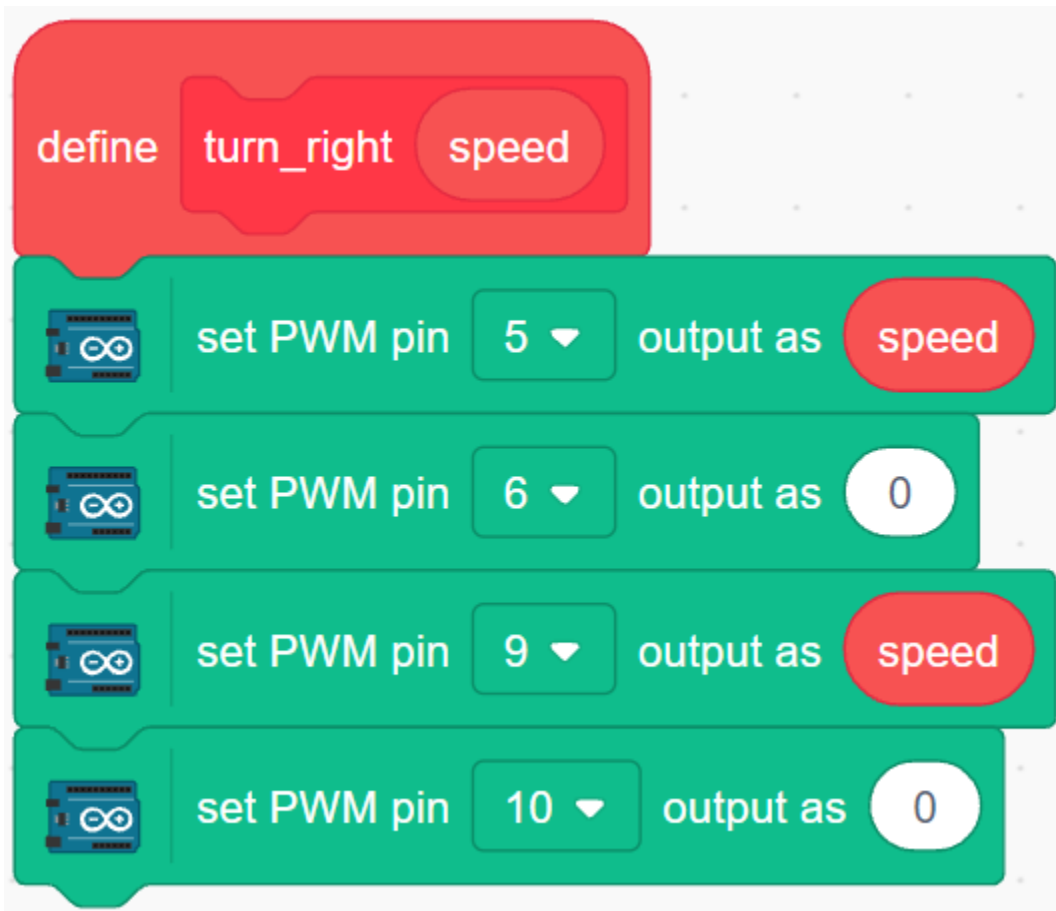
3. Links drehen

Die linken und rechten Motoren drehen gleichzeitig im Uhrzeigersinn, um das Auto nach links zu drehen.



4. Rechts drehen

Drehen Sie die linken und rechten Motoren in entgegengesetzter Richtung, um das Auto nach rechts zu drehen.



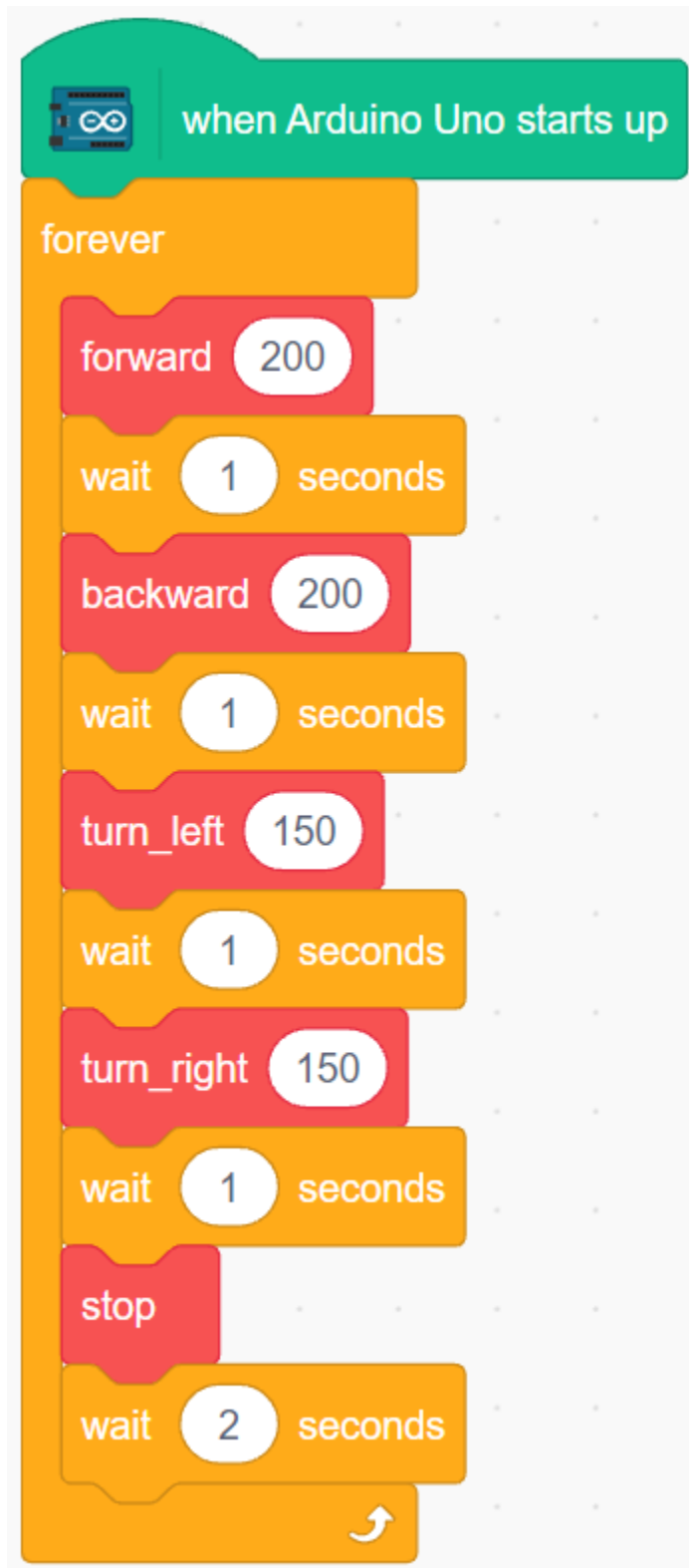
5. Stopp

Stoppen Sie das Auto, indem Sie alle Motoren auf 0 setzen.



6. Das Auto bewegen

Lassen Sie das Auto vorwärts, rückwärts, links und rechts für 1 Sekunde bewegen, dann stoppen. Da alle Blöcke im [Forever]-Block platziert sind, werden Sie sehen, dass das Auto die obigen Aktionen wiederholt.



8.28 3.3 Der Linie folgen

Das Auto ist mit einem Linienverfolgungsmodul ausgestattet, mit dem es der schwarzen Linie folgen kann.

Bevor Sie mit dem Projekt beginnen, sollten 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.

8.28.1 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 IM SET	LINK
3-in-1 Starter Set	380+	

Sie können die Teile auch einzeln über die folgenden Links erwerben:

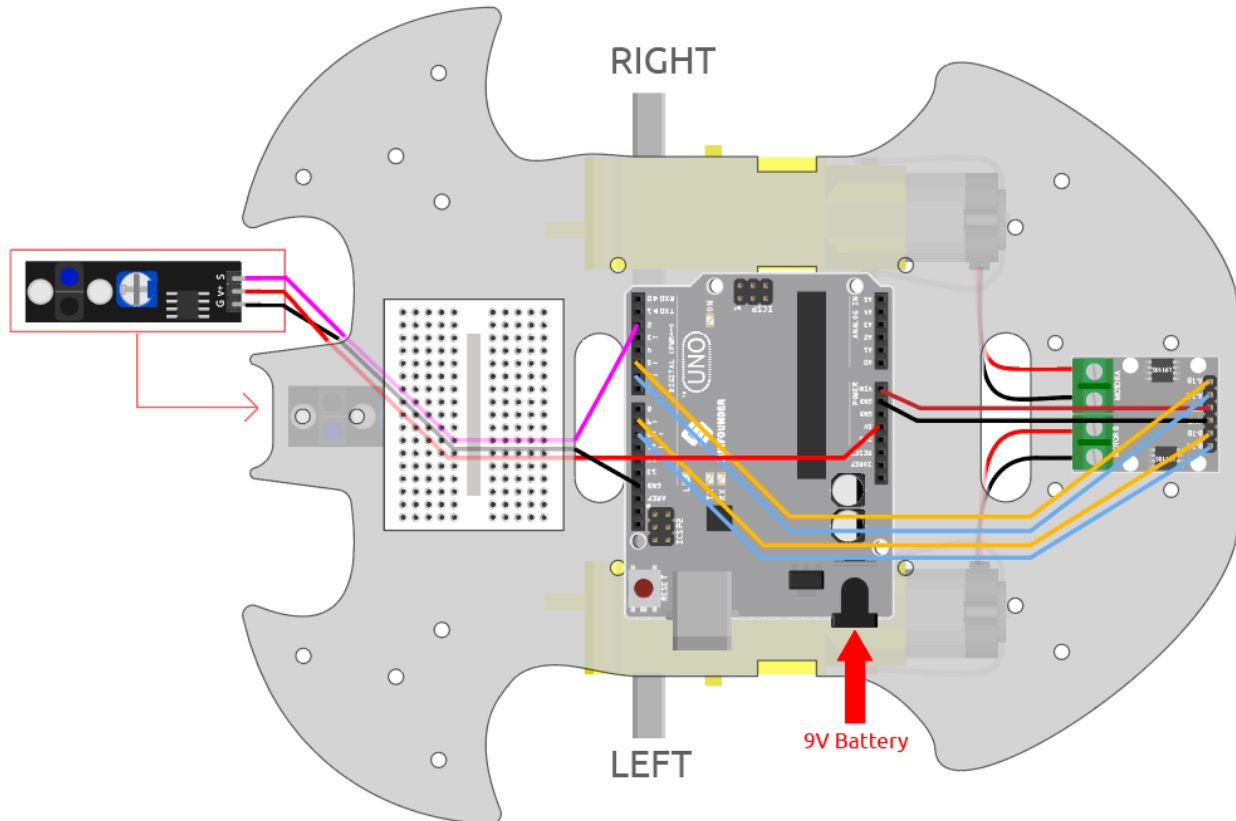
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Linienverfolgungsmodul</i>	

8.28.2 Schaltkreis aufbauen

Dies ist ein digitales Linienverfolgungsmodul. Wenn eine schwarze Linie erkannt wird, gibt es 1 aus; bei einer weißen Linie wird ein Wert von 0 ausgegeben. Zudem können Sie seinen Erfassungsabstand über das Potentiometer am Modul anpassen.

Bauen Sie den Schaltkreis entsprechend dem folgenden Diagramm auf.

Linienverfolgungsmodul	R3 Board
S	2
V+	5V
G	GND



8.28.3 Das Modul einstellen

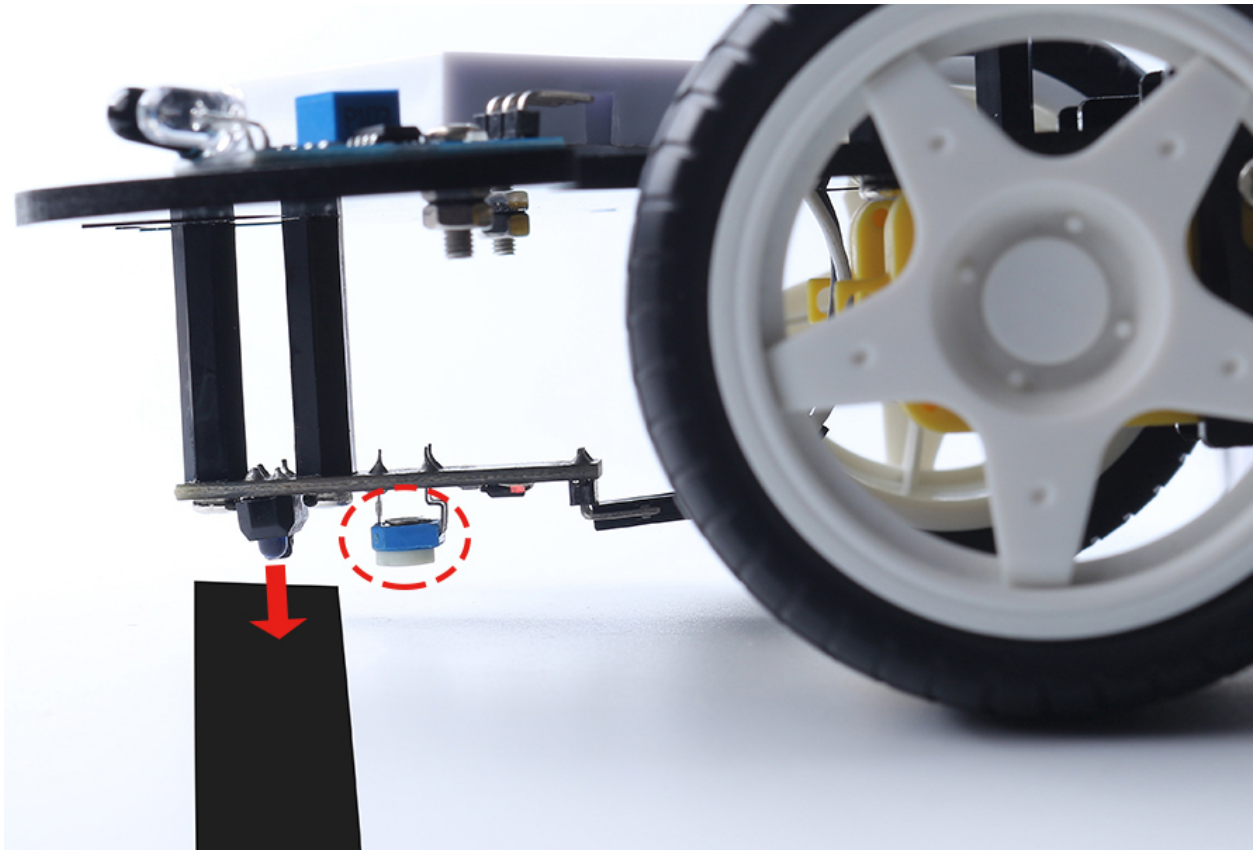
Bevor Sie mit dem Projekt beginnen, müssen Sie die Empfindlichkeit des Moduls anpassen.

Verdrahten Sie gemäß dem obigen Diagramm und schalten Sie dann das R3-Board ein (entweder direkt über das USB-Kabel oder das 9V-Batteriekabel). Laden Sie den Code noch nicht hoch.

Kleben Sie ein schwarzes Isolierband auf den Tisch und stellen Sie den Wagen darauf.

Beobachten Sie die Signalleuchte am Modul, um sicherzustellen, dass sie auf dem weißen Tisch leuchtet und auf dem schwarzen Band erlischt.

Wenn dies nicht der Fall ist, müssen Sie das Potentiometer am Modul so einstellen, dass der oben beschriebene Effekt erzielt wird.

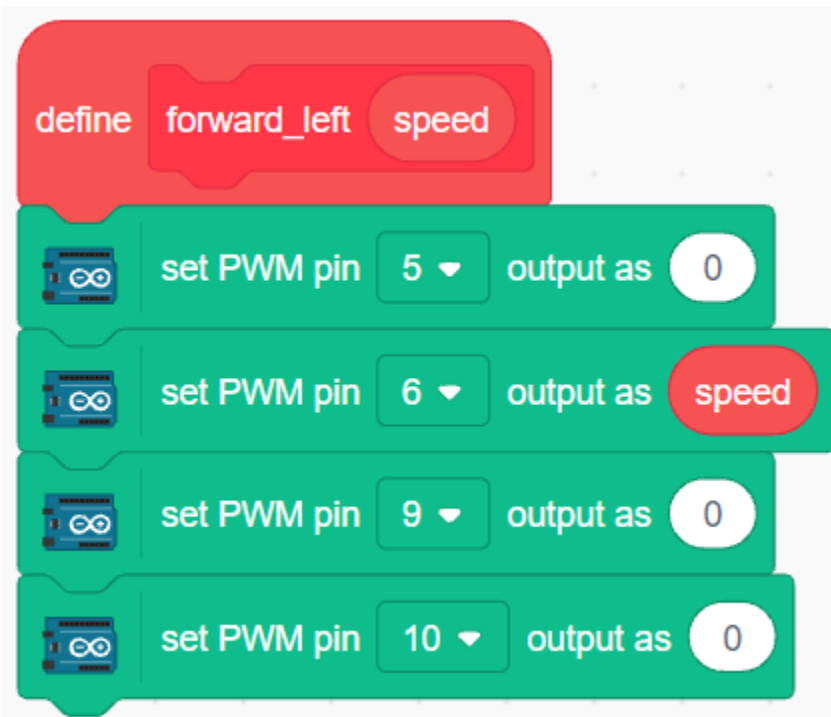


8.28.4 Programmierung

Erstellen Sie nun 2 Blöcke, die es dem Auto ermöglichen, sich entweder nach vorne links oder nach vorne rechts zu bewegen.

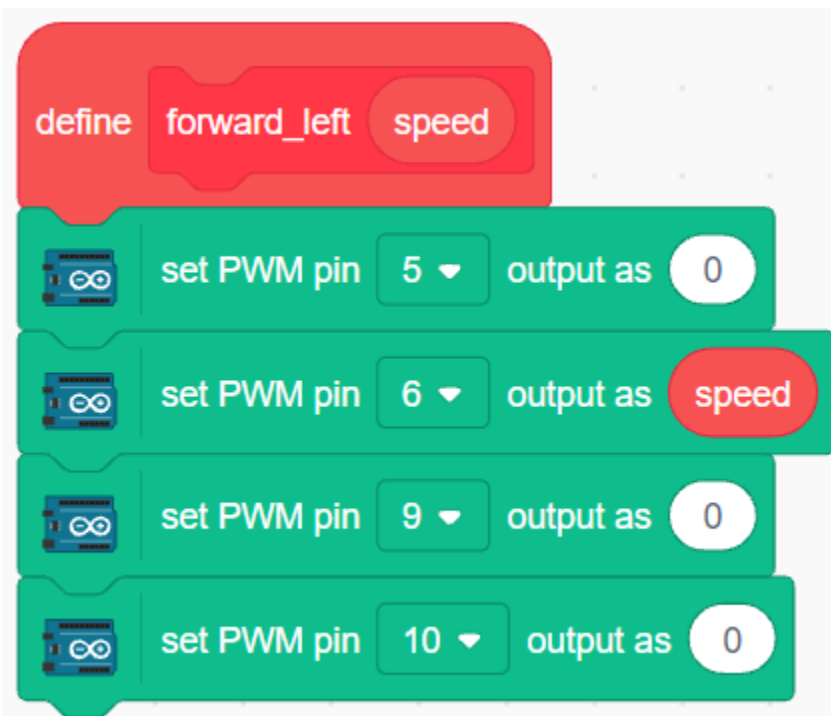
1. Nach vorne links bewegen

Wenn der rechte Motor im Uhrzeigersinn dreht und der linke Motor stillsteht, bewegt sich das Auto leicht nach vorne links.



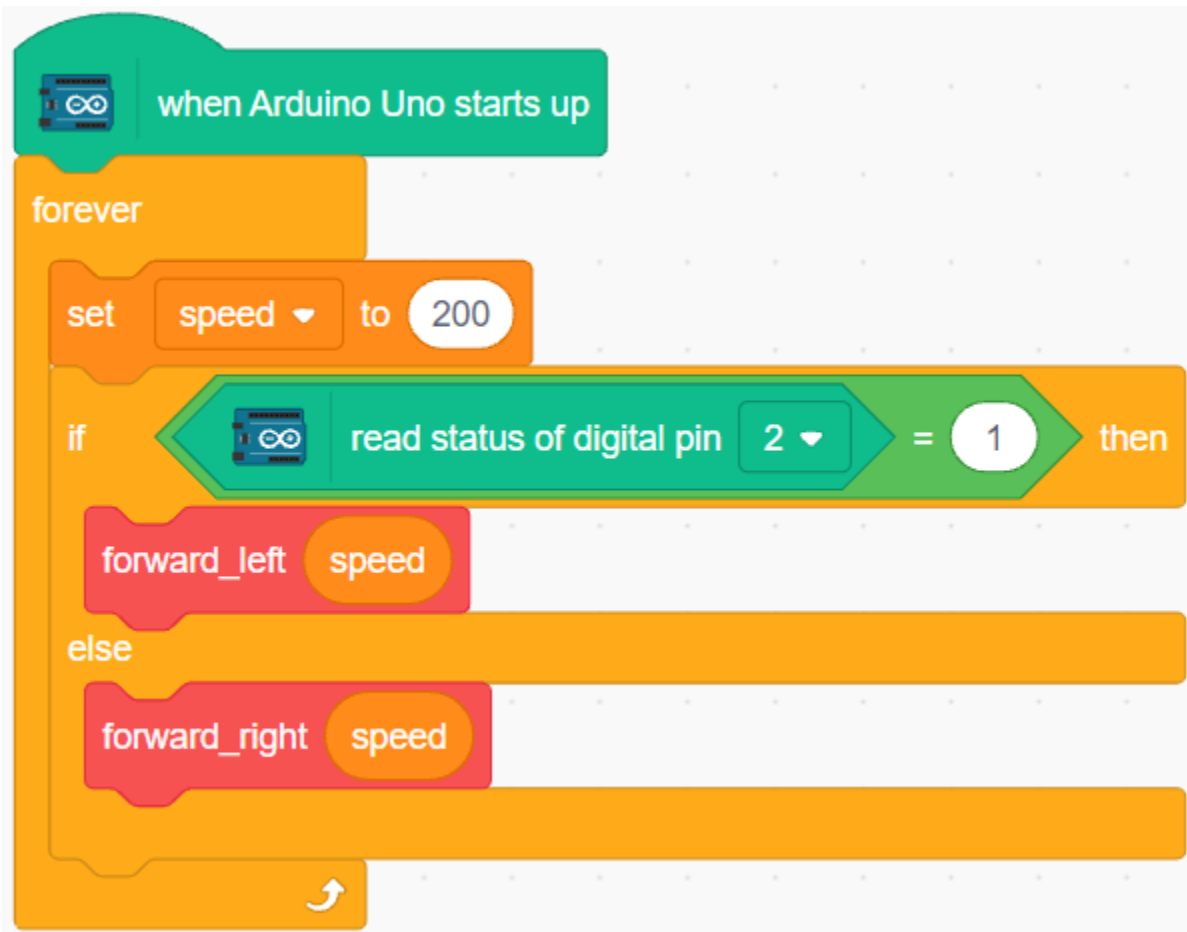
2. Nach vorne rechts bewegen

Ebenso bewegt sich das Auto leicht nach rechts, wenn der linke Motor gegen den Uhrzeigersinn dreht und der rechte Motor stillsteht.



3. Linienverfolgung

Lesen Sie den Wert des Linienverfolgungsmoduls. Wenn es 1 ist, wurde eine schwarze Linie erkannt. Lassen Sie das Auto nach vorne links fahren, andernfalls nach vorne rechts.



Nachdem Sie den Code auf das R3-Board hochgeladen haben, richten Sie das Linienverfolgungsmodul unter dem Auto auf die schwarze Linie aus, und Sie werden sehen, wie das Auto der Linie folgt.

8.29 3.4 Folge Deiner Hand

Stellen Sie sich dieses Auto hier als Ihr Haustier vor und wenn Sie ihm zuwinken, kommt es auf Sie zugelaufen.

8.29.1 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 IM SET	LINK
3-in-1 Starter Set	380+	

Sie können die Teile auch einzeln über die folgenden Links erwerben:

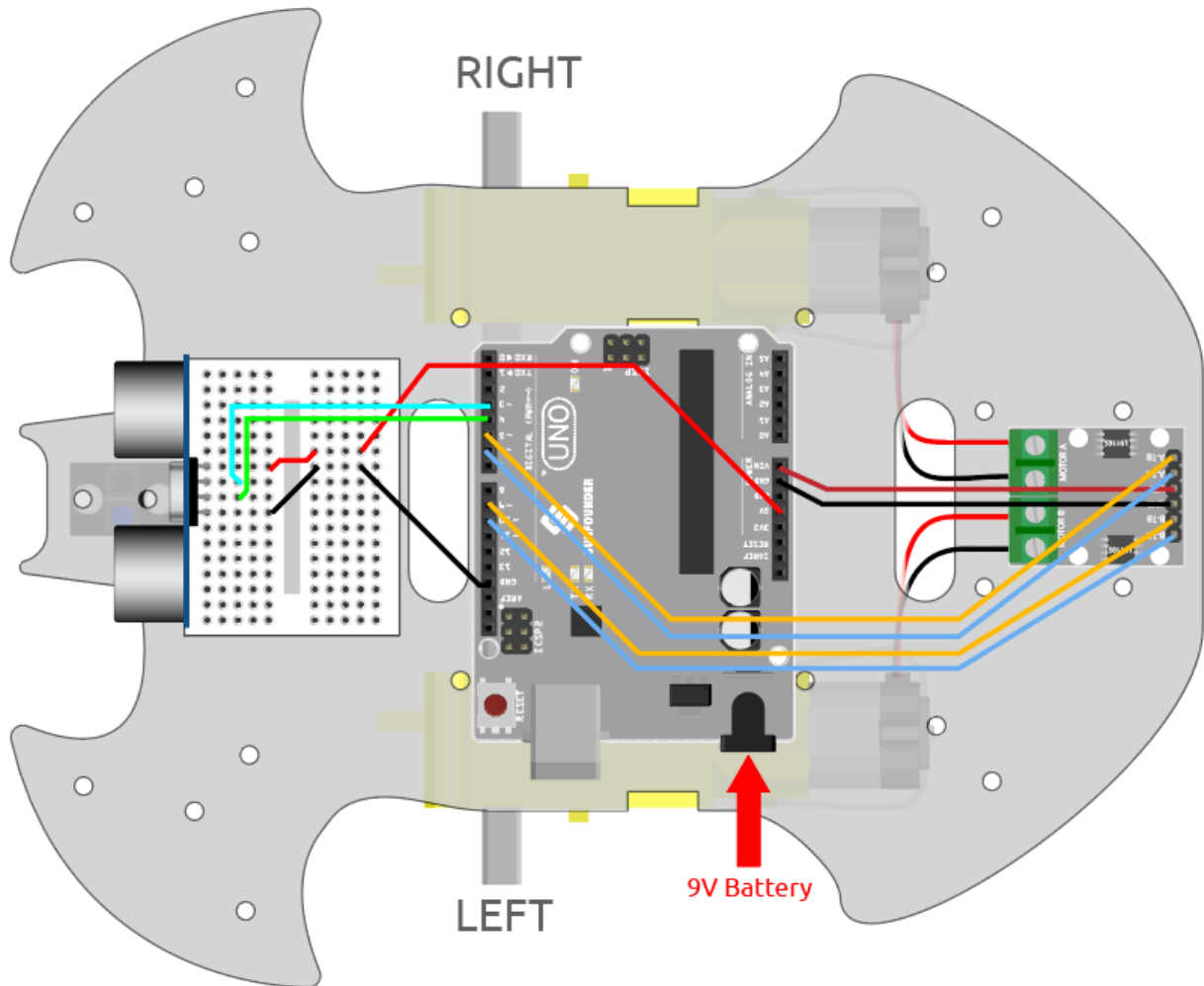
KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Ultraschall-Modul</i>	

8.29.2 Schaltkreis aufbauen

Ein Ultraschallsensormodul ist ein Instrument, das den Abstand zu einem Objekt mit Hilfe von Ultraschallwellen misst. Es verfügt über zwei Sonden. Eine dient zum Senden von Ultraschallwellen und die andere zum Empfangen der Wellen und zur Umwandlung der Sende- und Empfangszeit in eine Entfernung, wodurch der Abstand zwischen dem Gerät und einem Hindernis ermittelt wird.

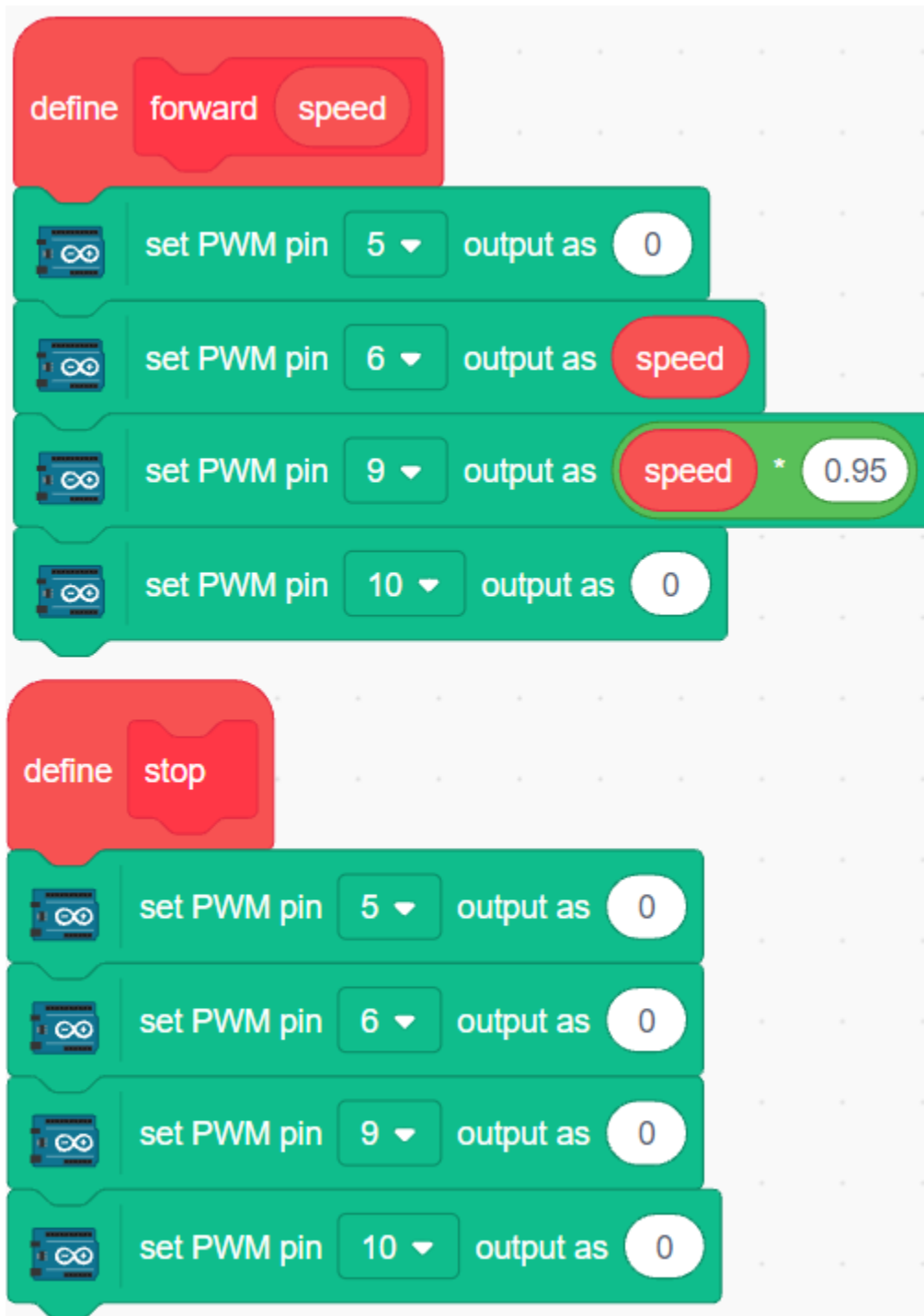
Bauen Sie nun den Schaltkreis entsprechend dem folgenden Diagramm auf.

Ultraschallmodul	R3 Board
Vcc	5V
Trig	3
Echo	4
Gnd	GND

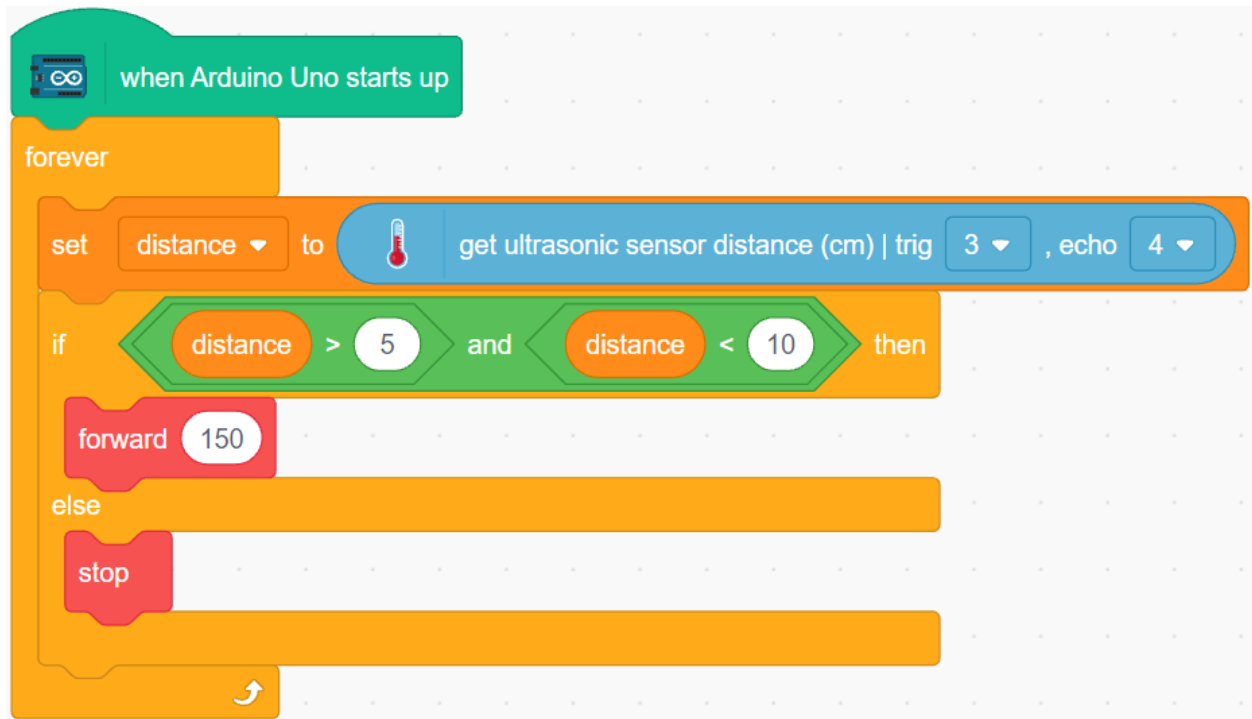


8.29.3 Programmierung

Erstellen Sie Blöcke, um das Auto vorwärts fahren und stoppen zu lassen.



Halten Sie Ihre Hand vor das Auto und lesen Sie den Wert des Ultraschallmoduls. Wenn der erkannte Abstand Ihrer Hand zwischen 5-10 cm liegt, lassen Sie das Auto vorwärts fahren, ansonsten stoppen Sie es.



8.30 3.5 Hindernisvermeidung

Am Auto sind zwei Infrarot-Hindernisvermeidungsmodule montiert, mit denen nahegelegene Hindernisse erkannt werden können.

In diesem Projekt darf das Auto frei vorwärts fahren und kann beim Erkennen eines Hindernisses ausweichen und in andere Richtungen weiterfahren.

8.30.1 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 IM SET	LINK
3-in-1 Starter Set	380+	

Sie können die Teile auch einzeln über die folgenden Links erwerben:

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Hindernisvermeidungsmodul</i>	

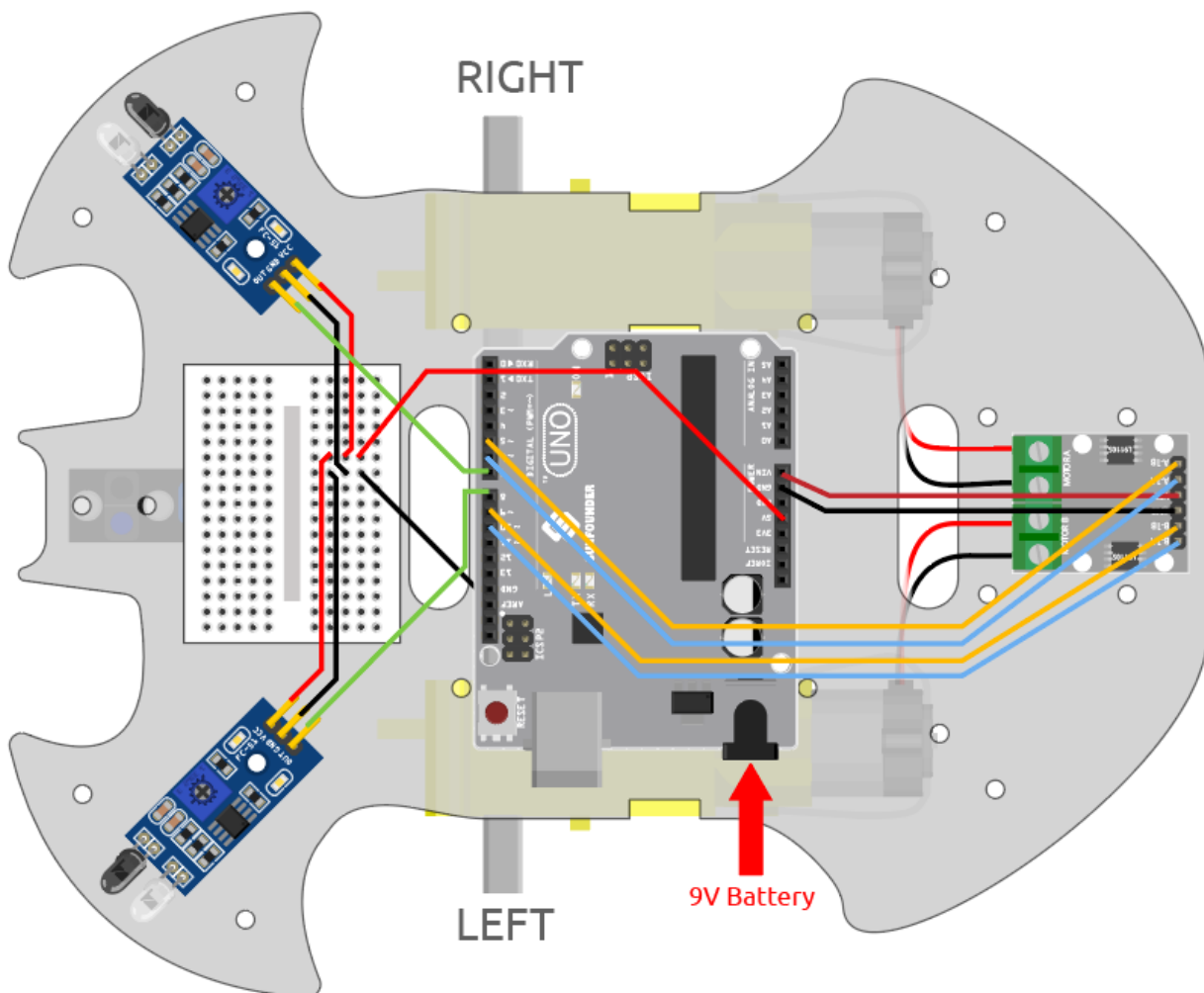
8.30.2 Schaltkreis aufbauen

Das Hindernisvermeidungsmodul ist ein infrarotgesteuerter Näherungssensor mit einstellbarer Distanz, dessen Ausgang normalerweise hoch ist und bei Erkennung eines Hindernisses niedrig wird.

Bauen Sie nun den Schaltkreis entsprechend dem untenstehenden Diagramm auf.

Linkes IR-Modul	R3 Board
OUT	8
GND	GND
VCC	5V

Rechtes IR-Modul	R3 Board
OUT	7
GND	GND
VCC	5V



8.30.3 Das Modul einstellen

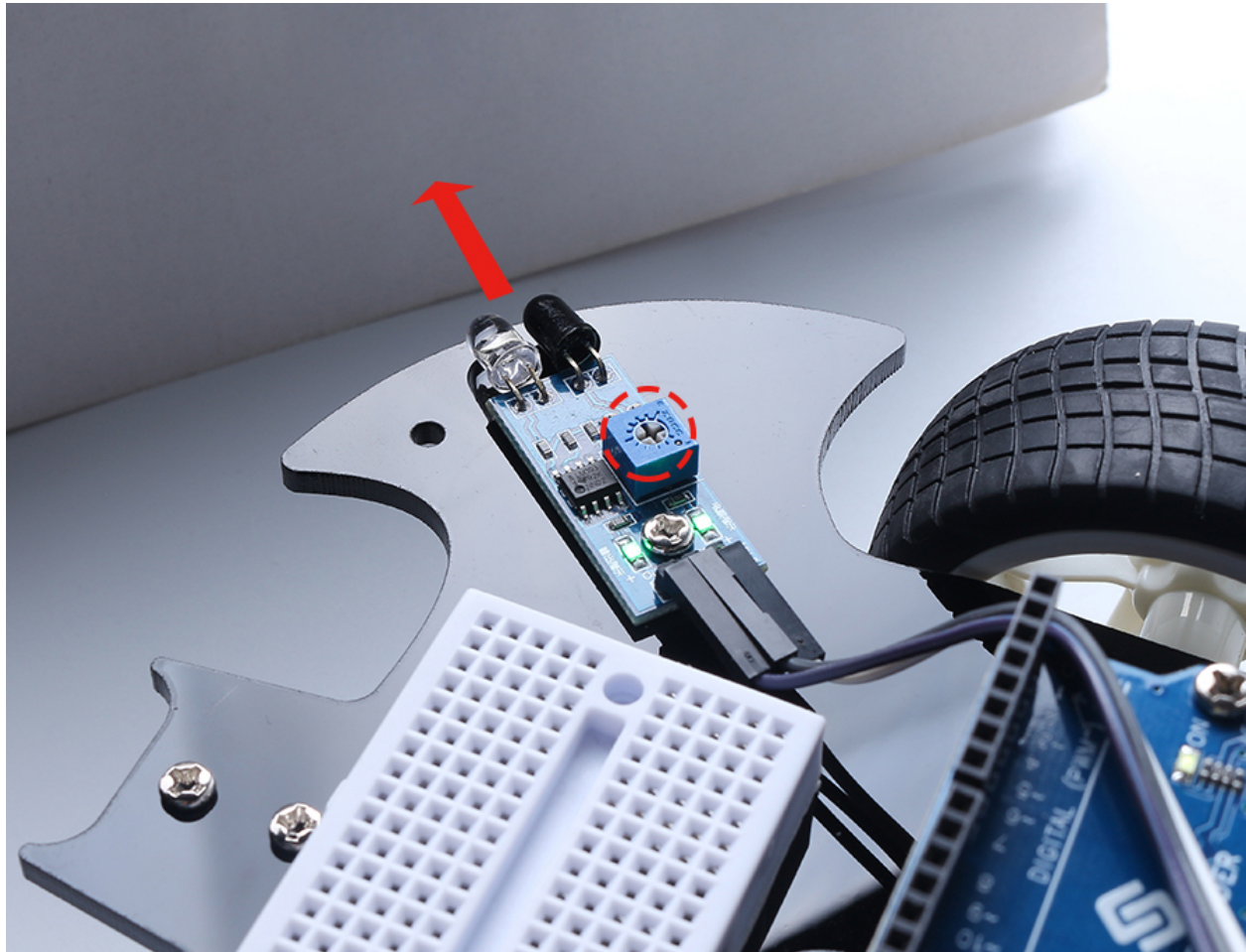
Vor Projektbeginn müssen Sie die Erfassungsdistanz des Moduls anpassen.

Verdrahten Sie entsprechend dem obigen Diagramm, schalten Sie das R3-Board ein (entweder durch direktes Einstecken des USB-Kabels oder durch Anschließen des 9V-Batteriekabels) und laden Sie den Code noch nicht hoch.

Platzieren Sie ein Notizbuch oder ein anderes flaches Objekt etwa 5 cm vor der IR-Hindernisvermeidung.

Dann verwenden Sie einen Schraubendreher, um das Potentiometer am Modul zu drehen, bis die Signalleuchte am Modul gerade leuchtet, um die maximale Erfassungsdistanz von 5 cm einzustellen.

Verfahren Sie auf die gleiche Weise, um ein weiteres Infrarot-Modul anzupassen.



8.30.4 Programmierung

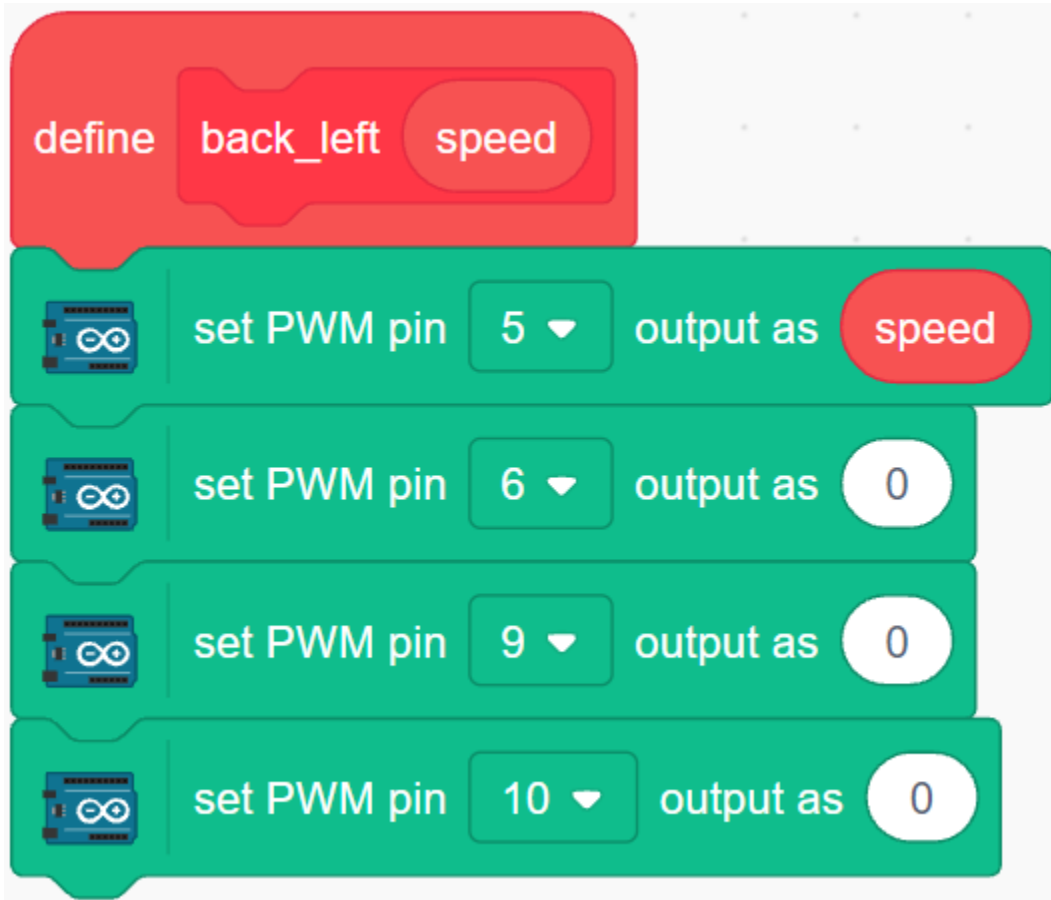
Das gewünschte Ergebnis:

- Wenn das linke IR-Modul ein Hindernis erkennt, fährt das Auto rückwärts nach links.
- Wenn das rechte IR-Modul ein Hindernis erkennt, fährt das Auto rückwärts nach rechts.
- Wenn beide IR-Module gleichzeitig ein Hindernis erkennen, fährt das Auto direkt rückwärts.
- Andernfalls fährt das Auto vorwärts.

Erstellen Sie jetzt die entsprechenden Blöcke.

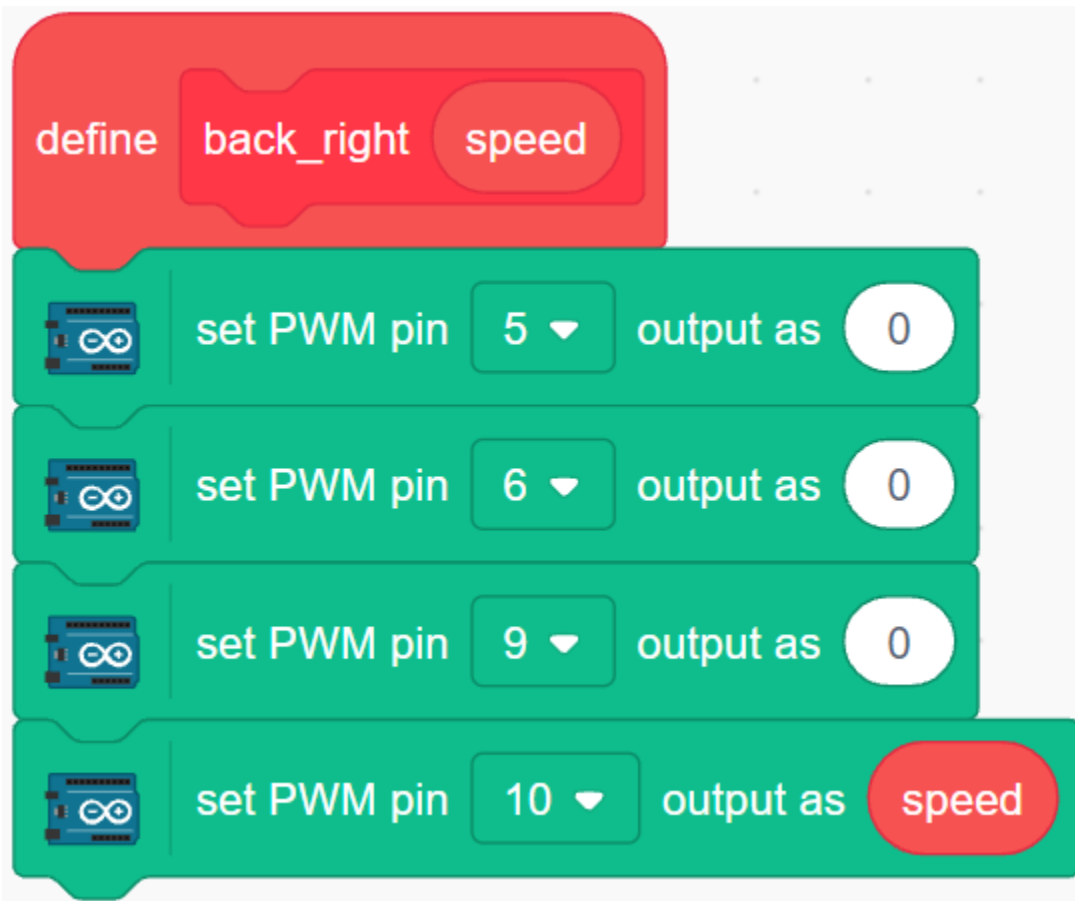
1. Das Auto fährt rückwärts nach links

Wenn der rechte Motor gegen den Uhrzeigersinn dreht und der linke Motor sich nicht dreht, wird das Auto rückwärts nach links fahren.

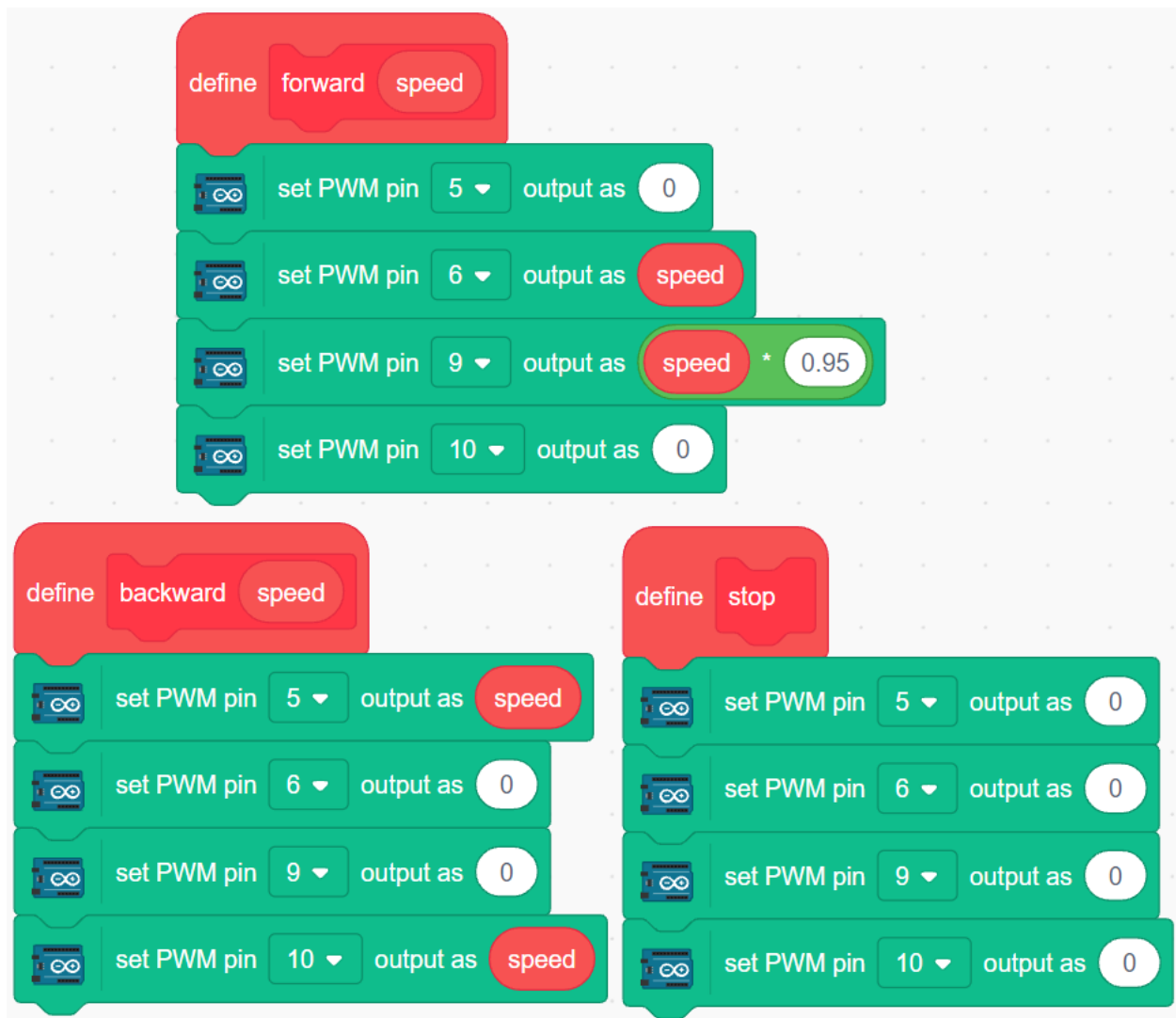


2. Das Auto fährt rückwärts nach rechts

Wenn der linke Motor im Uhrzeigersinn dreht und der rechte Motor sich nicht dreht, wird das Auto rückwärts nach rechts fahren.

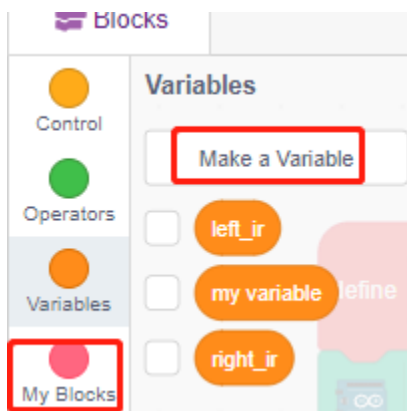


3. Das Auto fährt vorwärts, rückwärts und hält an

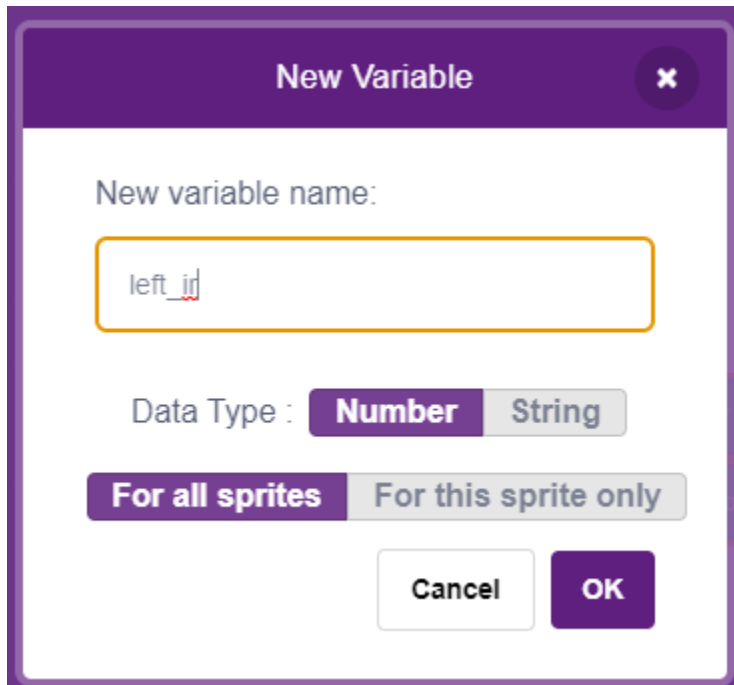


4. Werte der 2 IR-Module auslesen

Klicken Sie in der Palette **Variables** auf **Make a variable**.



Geben Sie den Variablennamen ein und klicken Sie auf **OK**, um eine neue Variable zu erstellen.

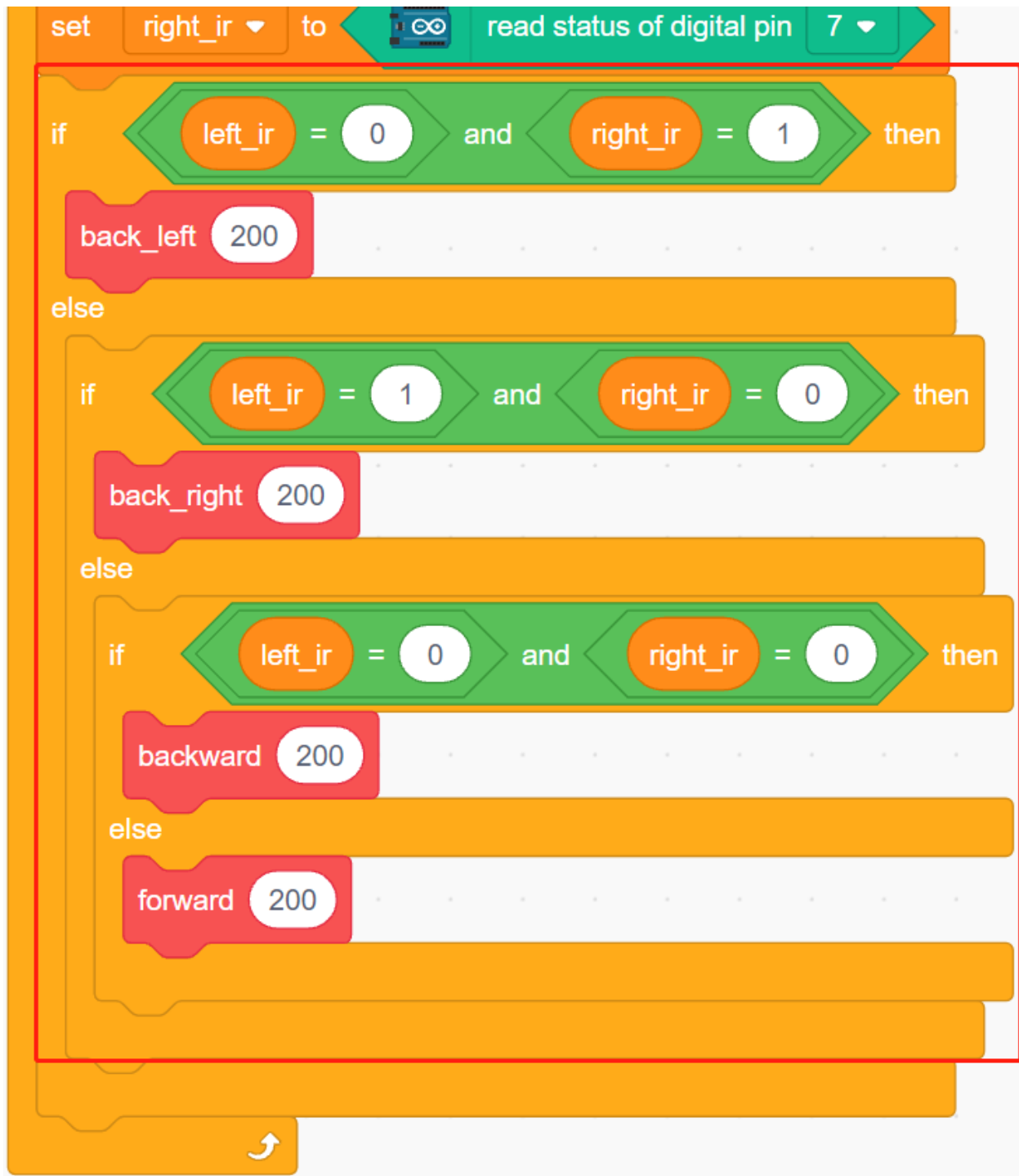


Lesen Sie die Werte der linken und rechten IR-Hindernisvermeidungsmodule und speichern Sie sie in den 2 neuen Variablen.



5. Hindernisvermeidung

- Wenn das linke IR-Modul 0 ist (Hindernis erkannt) und das rechte IR-Modul 1 ist, soll das Auto nach links zurückfahren.
- Wenn das rechte IR-Modul 0 ist (Hindernis erkannt), soll das Auto nach rechts zurückfahren.
- Wenn 2 IR-Module gleichzeitig ein Hindernis erkennen, fährt das Auto rückwärts.
- Ansonsten fährt das Auto weiter vorwärts.



8.31 3.6 Deiner Hand folgen 2

Im Projekt *3.4 Folge Deiner Hand* wird nur das Ultraschallmodul verwendet, wodurch das Auto nur vorwärts deiner Hand folgen kann.

In diesem Projekt verwenden wir gleichzeitig 2 IR-Hindernisvermeidungsmodule, sodass das Auto deiner Hand nach links oder rechts folgen kann.

8.31.1 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 IM SET	LINK
3-in-1 Starter Set	380+	

Sie können die Teile auch einzeln über die folgenden Links erwerben:

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Ultraschall-Modul</i>	
<i>Hindernisvermeidungsmodul</i>	

8.31.2 Schaltkreis aufbauen

Verbinden Sie das Ultraschallmodul und die beiden IR-Hindernisvermeidungsmodule gleichzeitig.

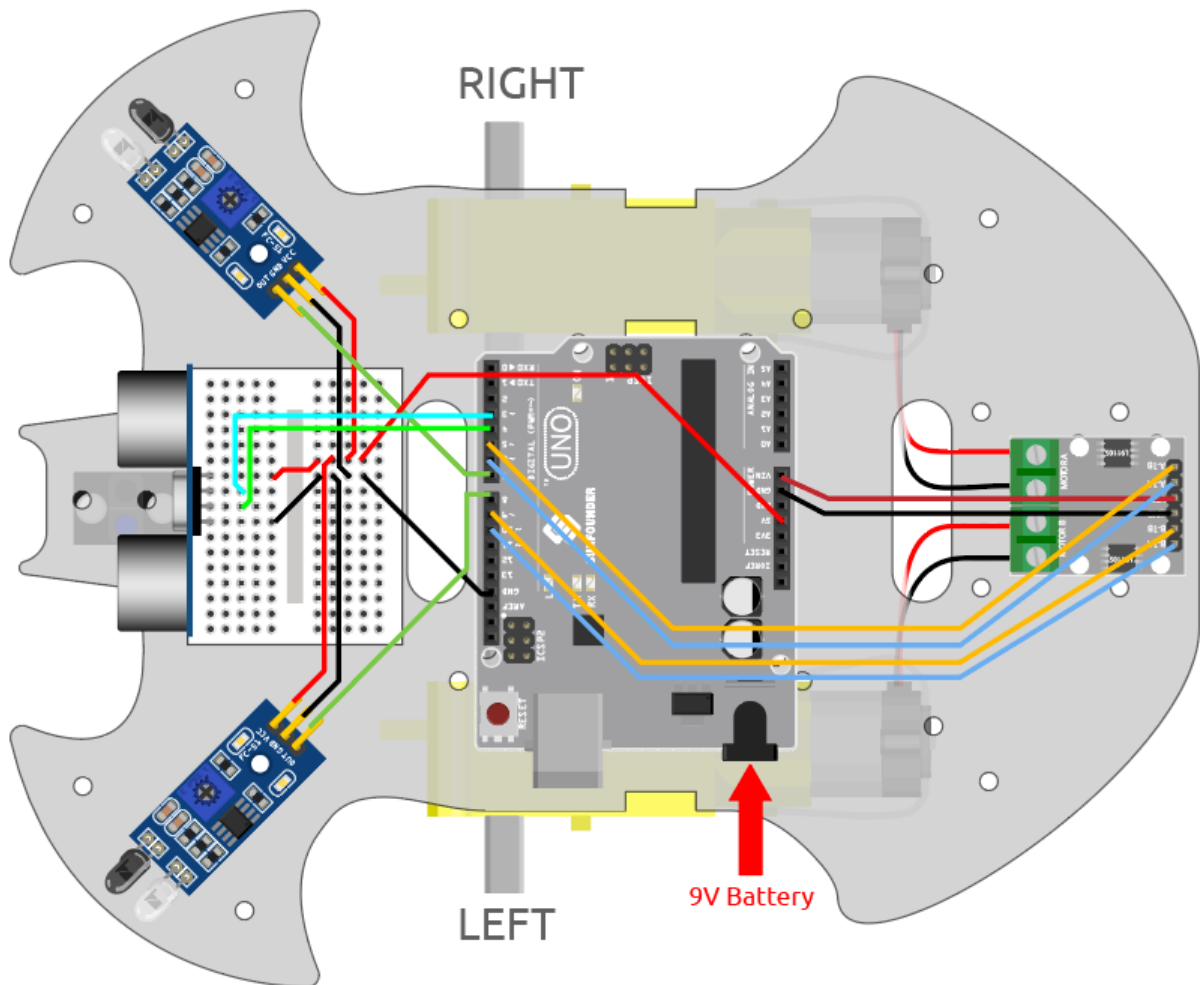
Die Verdrahtung zwischen Ultraschall und R3-Board ist wie folgt:

Ultraschall-Modul	R3-Board
Vcc	5V
Trig	3
Echo	4
Gnd	GND

Die Verdrahtung der 2 IR-Hindernisvermeidungsmodule zum R3-Board ist wie folgt:

Linkes IR-Modul	R3-Board
OUT	8
GND	GND
VCC	5V

Rechtes IR-Modul	R3-Board
OUT	7
GND	GND
VCC	5V



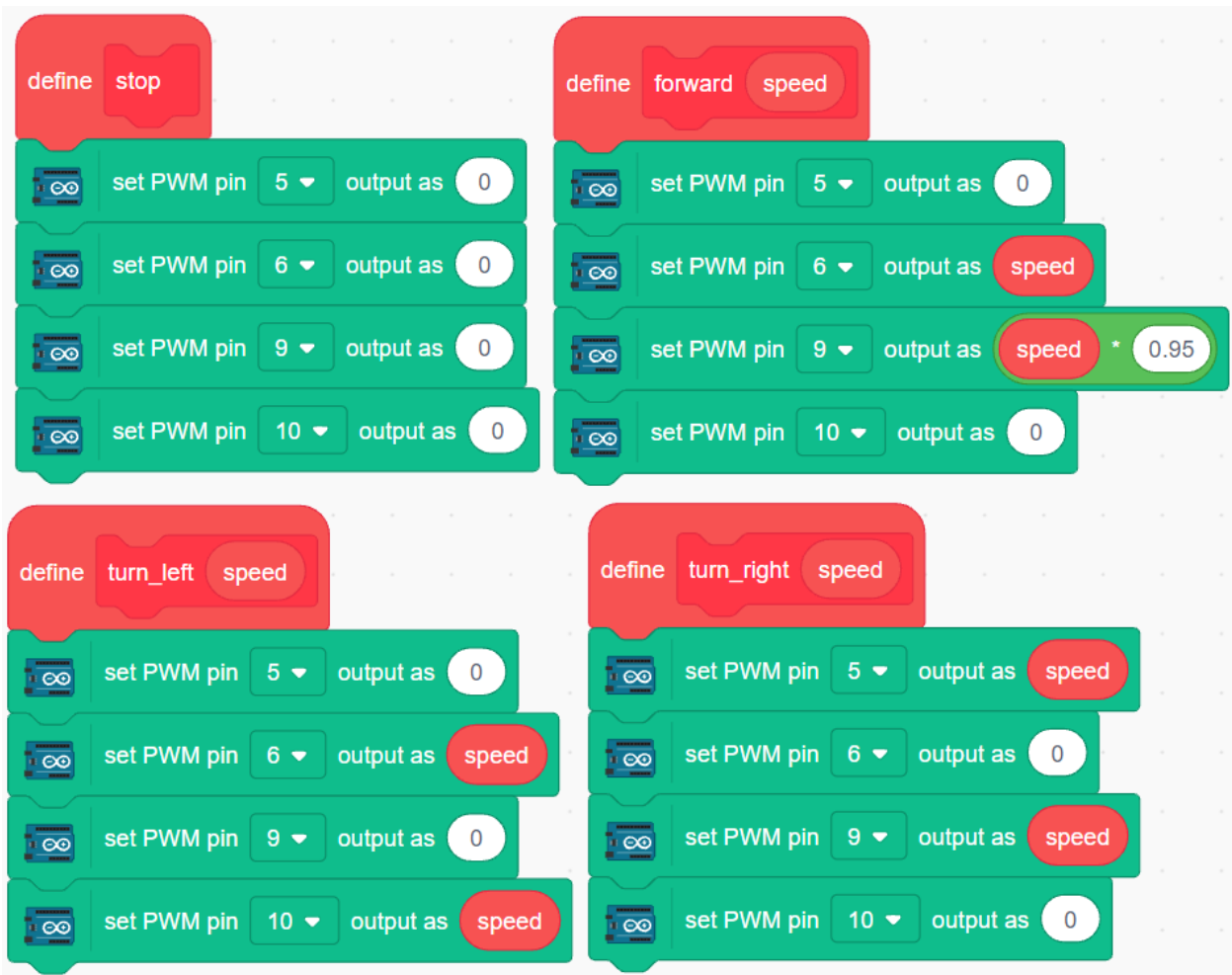
8.31.3 Programmierung

Das gewünschte Ergebnis dieses Projekts:

- Ultraschall erkennt deine Hand etwa 5-10 cm vor sich und lässt das Auto folgen.
- Das linke Infrarot-Modul erkennt deine Hand und dreht sich nach links.
- Das rechte IR-Modul erkennt deine Hand und dreht sich nach rechts.

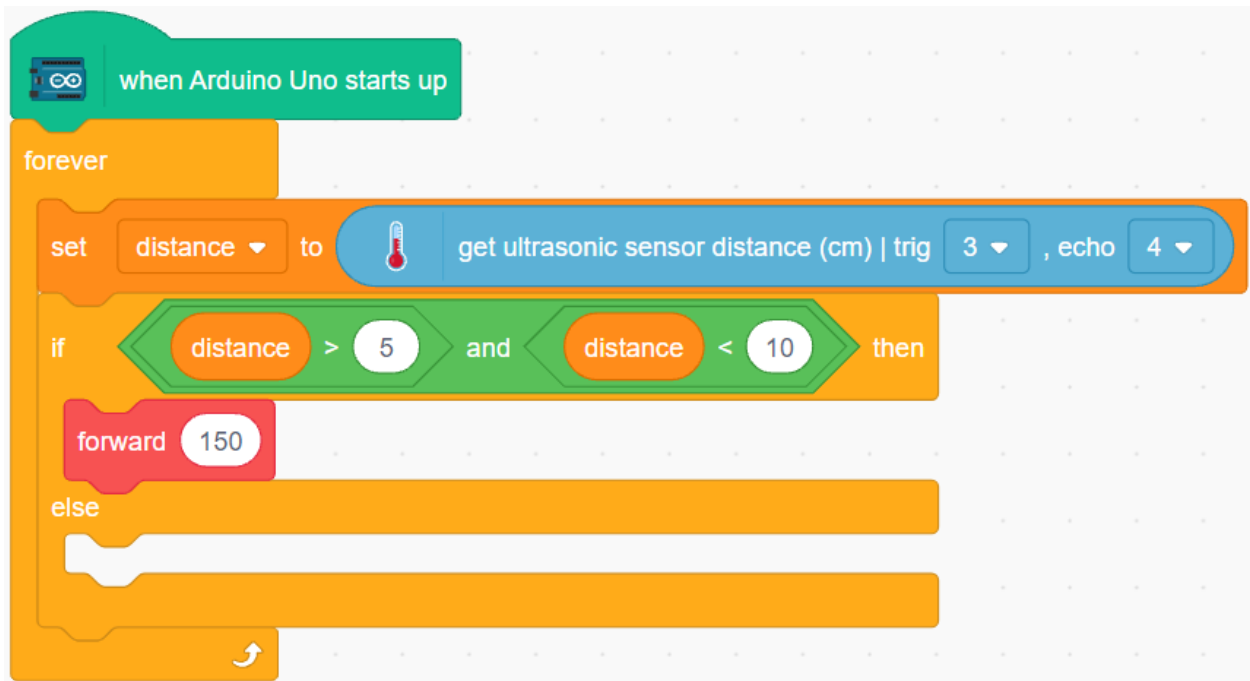
1. Block erstellen

Blöcke erstellen, die es dem Auto ermöglichen, vorwärts zu fahren, nach links und rechts abzubiegen und anzuhalten.



2. Zum Vorwärtsbewegen folgen

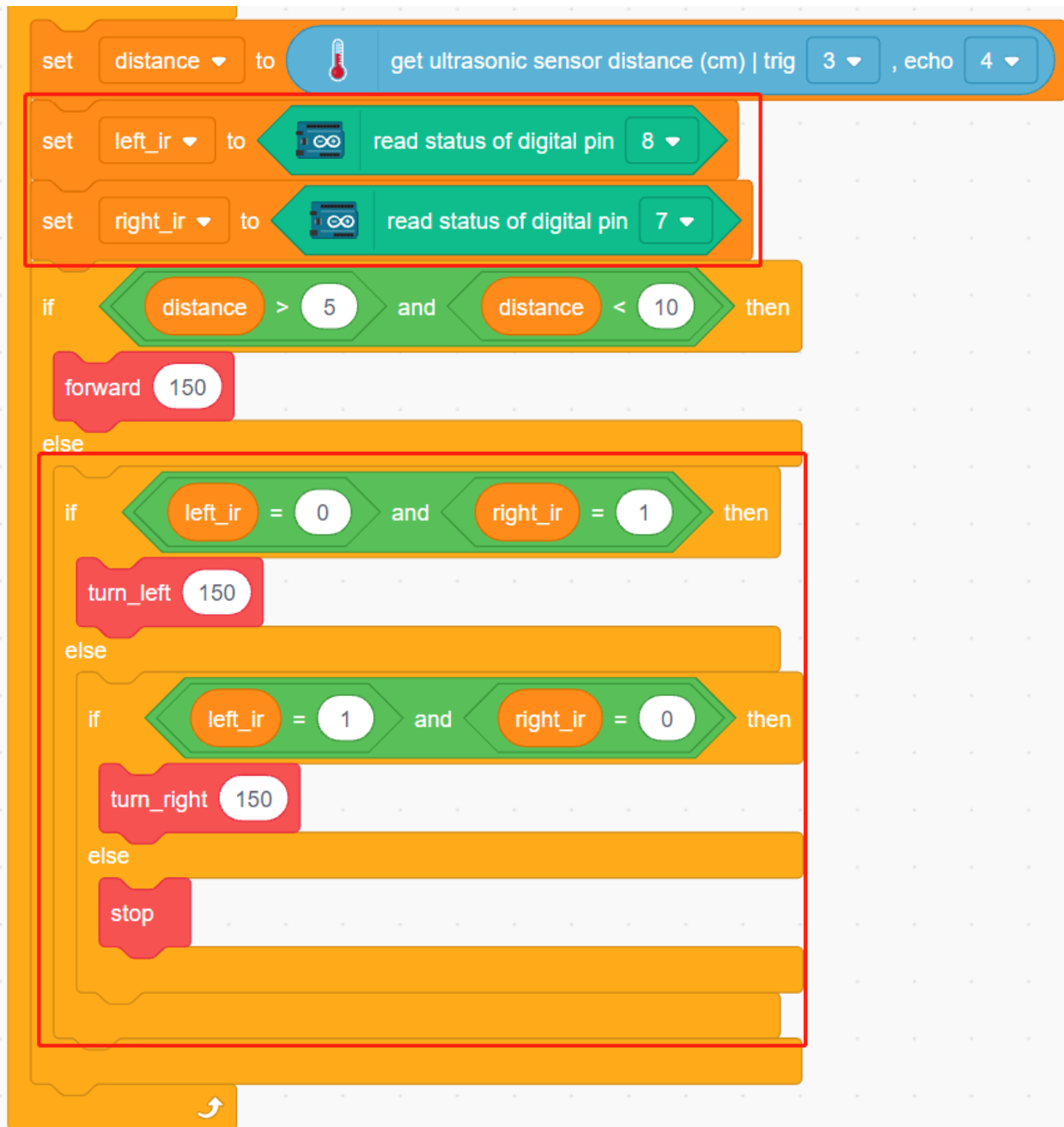
Lesen den Ultraschallwert und wenn deine Hand in einem Abstand von 5-10 cm erkannt wird, dann lass das Auto folgen.



3. Zum Links- und Rechtsdrehen folgen

Lese die Werte der linken und rechten IR-Module.

- Wenn das linke IR-Modul deine Hand erkennt, drehe dich nach links.
- Wenn das rechte IR-Modul deine Hand erkennt, drehe dich nach rechts.
- Wenn weder das IR-Modul noch das Ultraschallmodul deine Hand erkennen, lass das Auto stehen.



8.32 3.7 Hindernisvermeidung 2

Im Projekt [3.5 Hindernisvermeidung](#) werden lediglich 2 IR-Hindernisvermeidungsmodule zur Hindernisvermeidung eingesetzt. Doch die Erkennungsreichweite des IR-Hindernisvermeidungsmoduls ist kurz, was dazu führen kann, dass das Auto zu spät reagiert, um Hindernissen auszuweichen.

In diesem Projekt fügen wir außerdem ein Ultraschallmodul für die Fernerkennung hinzu, sodass das Auto Hindernisse aus größerer Entfernung wahrnehmen und eine Entscheidung treffen kann.

8.32.1 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 IM SET	LINK
3-in-1 Starter Set	380+	

Sie können die Teile auch einzeln über die folgenden Links erwerben:

KOMPONENTENBESCHREIBUNG	KAUF-LINK
<i>SunFounder R3 Platine</i>	
<i>L9110 Motortreibermodul</i>	-
<i>TT Motor</i>	-
<i>Ultraschall-Modul</i>	
<i>Hindernisvermeidungsmodul</i>	

8.32.2 Schaltkreis aufbauen

Verbinden Sie das Ultraschallmodul und die beiden IR-Hindernisvermeidungsmodule gleichzeitig.

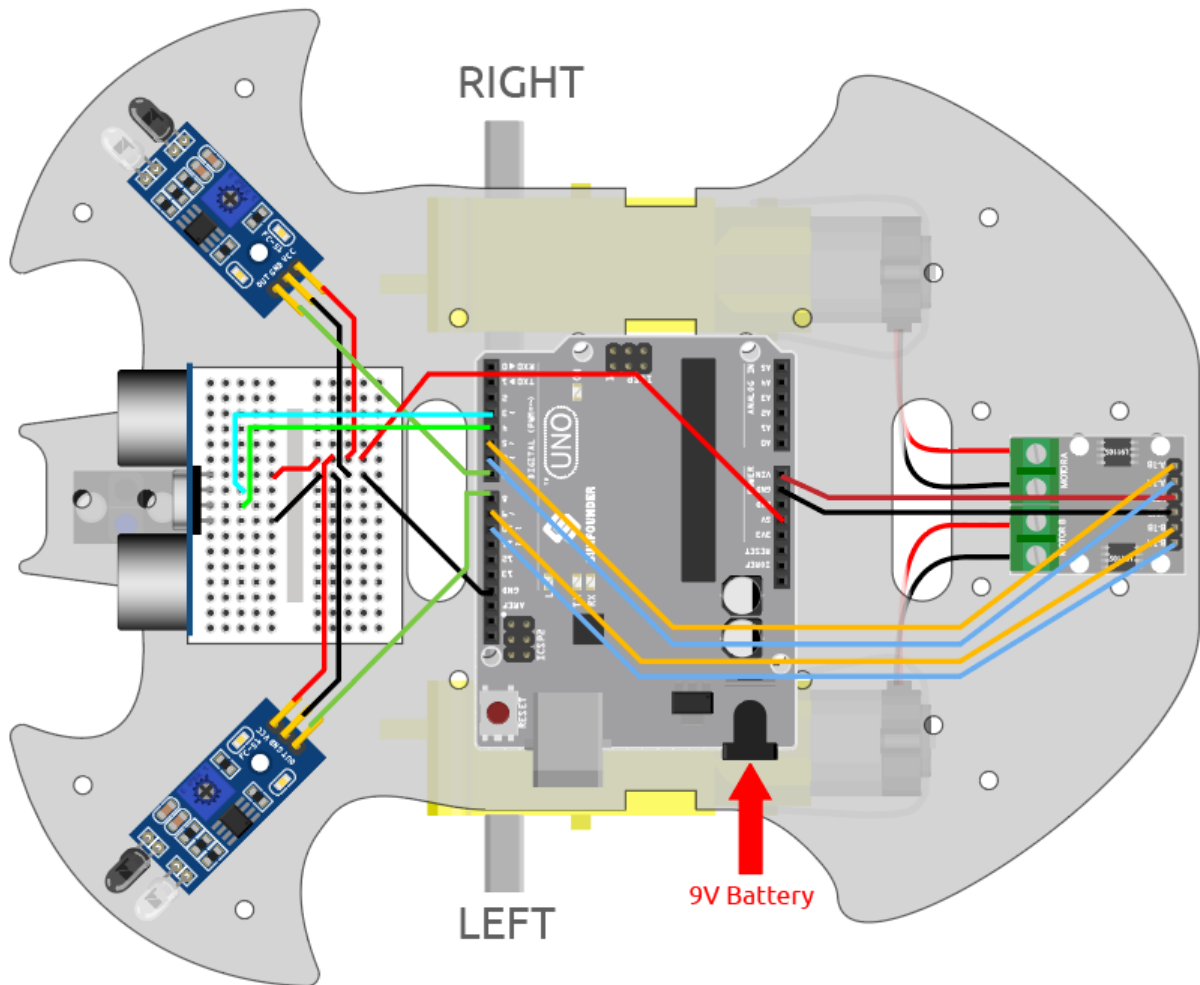
Die Verbindung des Ultraschallmoduls mit dem R3-Board ist wie folgt:

Ultraschall-Modul	R3-Board
Vcc	5V
Trig	3
Echo	4
Gnd	GND

Die Verbindung der beiden IR-Hindernisvermeidungsmodule mit dem R3-Board ist wie folgt:

Linkes IR-Modul	R3-Board
OUT	8
GND	GND
VCC	5V

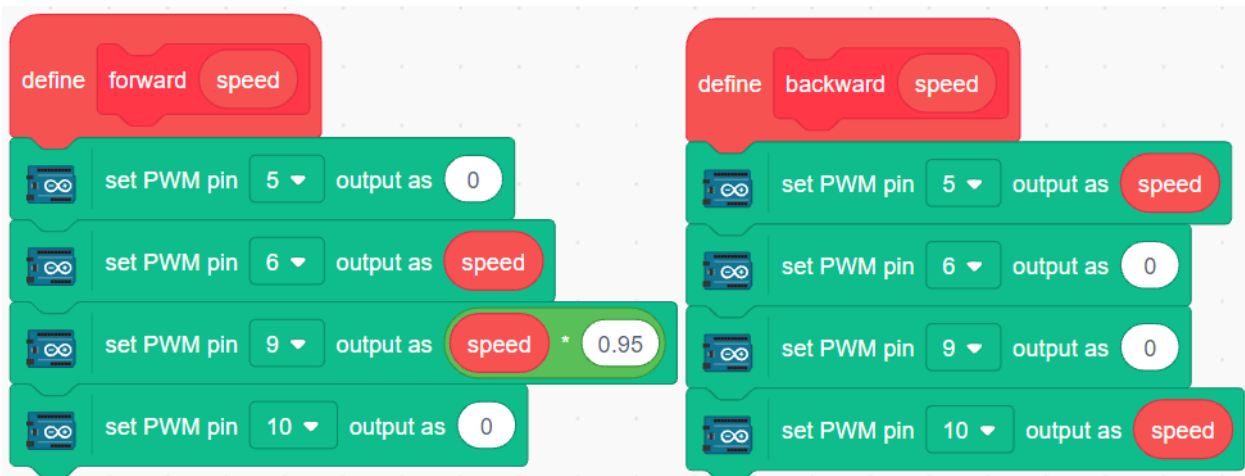
Rechtes IR-Modul	R3-Board
OUT	7
GND	GND
VCC	5V



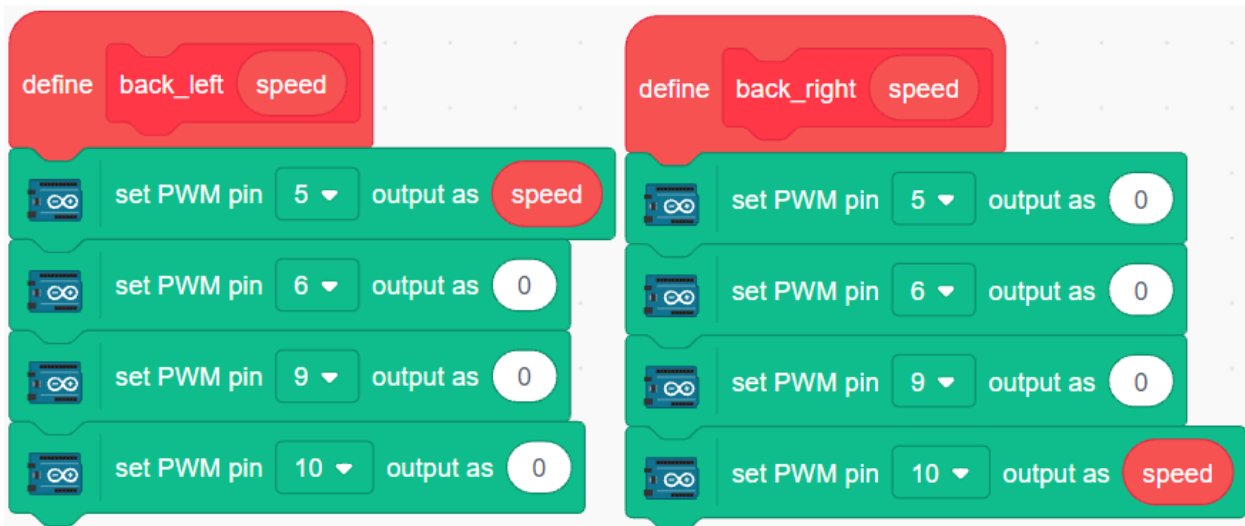
8.32.3 Programmierung

1. Funktion erstellen

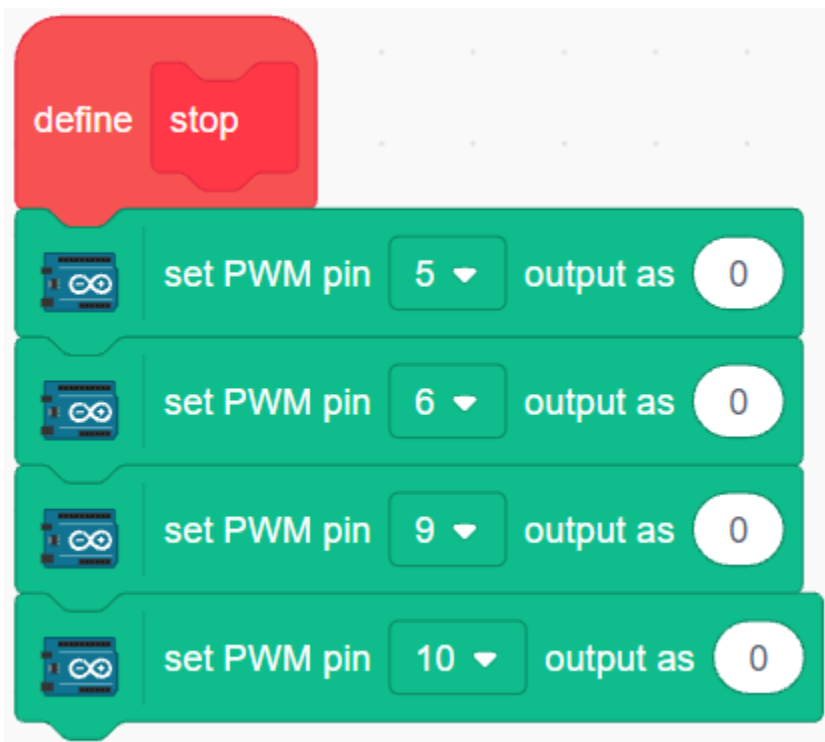
Lassen Sie das Auto vorwärts und rückwärts fahren.



Lassen Sie das Auto nach links hinten und nach rechts hinten fahren.



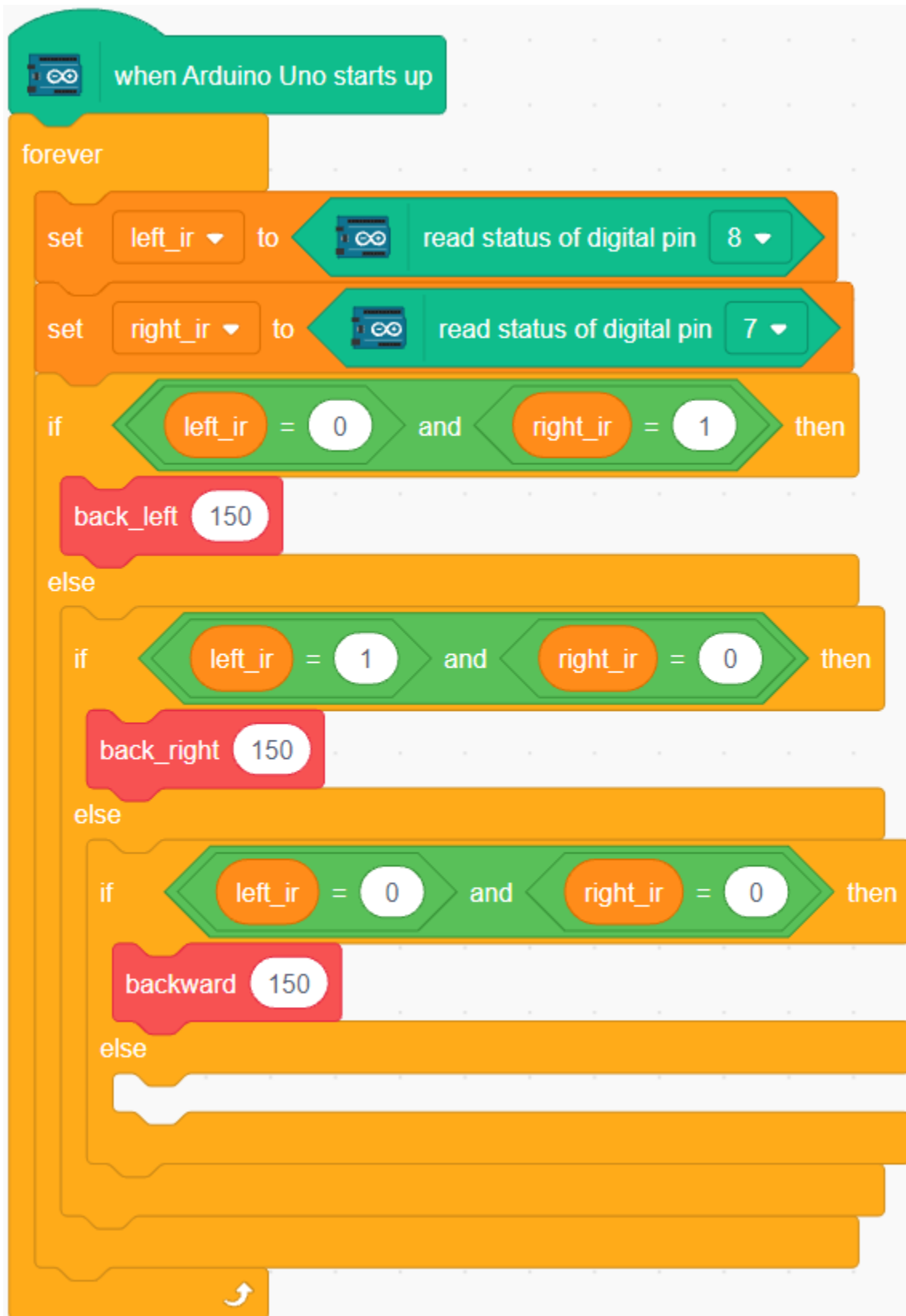
Lassen Sie das Auto stoppen.



2. Notfall-Hindernisvermeidung

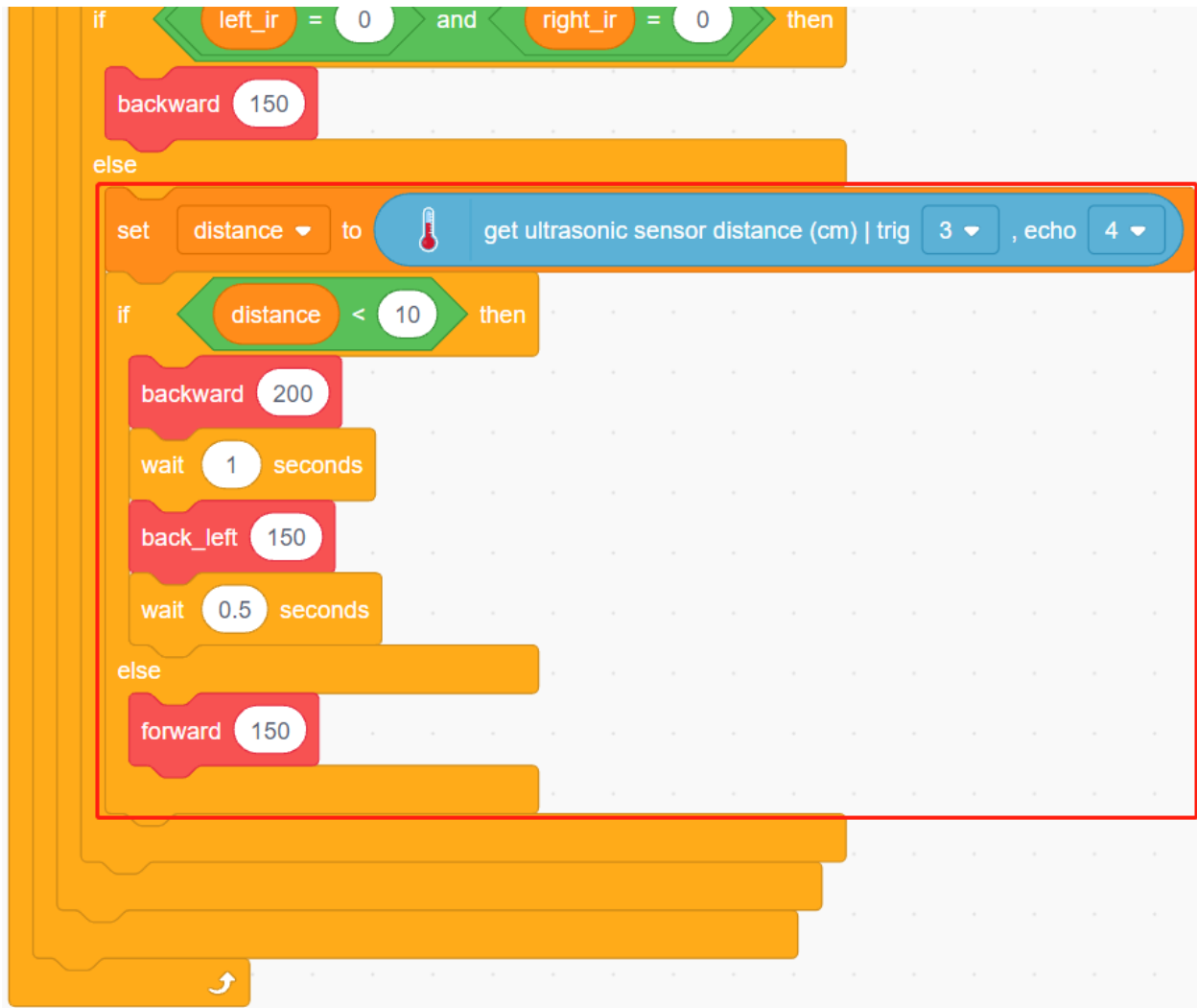
Die beiden IR-Hindernisvermeidungsmodule am Auto werden für die Notfall-Hindernisvermeidung verwendet, um Hindernisse in kurzer Entfernung, Ecken oder relativ kleine Hindernisse zu erkennen.

- Wenn das linke Infrarot-Modul ein Hindernis erkennt, fährt das Auto rückwärts nach links.
- Wenn das rechte IR-Modul ein Hindernis erkennt, fährt das Auto rückwärts nach rechts.
- Wenn beide Module gleichzeitig ein Hindernis erkennen, fährt das Auto direkt rückwärts.



3. Hindernisvermeidung über lange Strecken

Lesen Sie den Wert des Ultraschallmoduls. Wenn der erkannte Wert kleiner als 10 ist, fährt das Auto rückwärts; ansonsten fährt es weiter vorwärts.

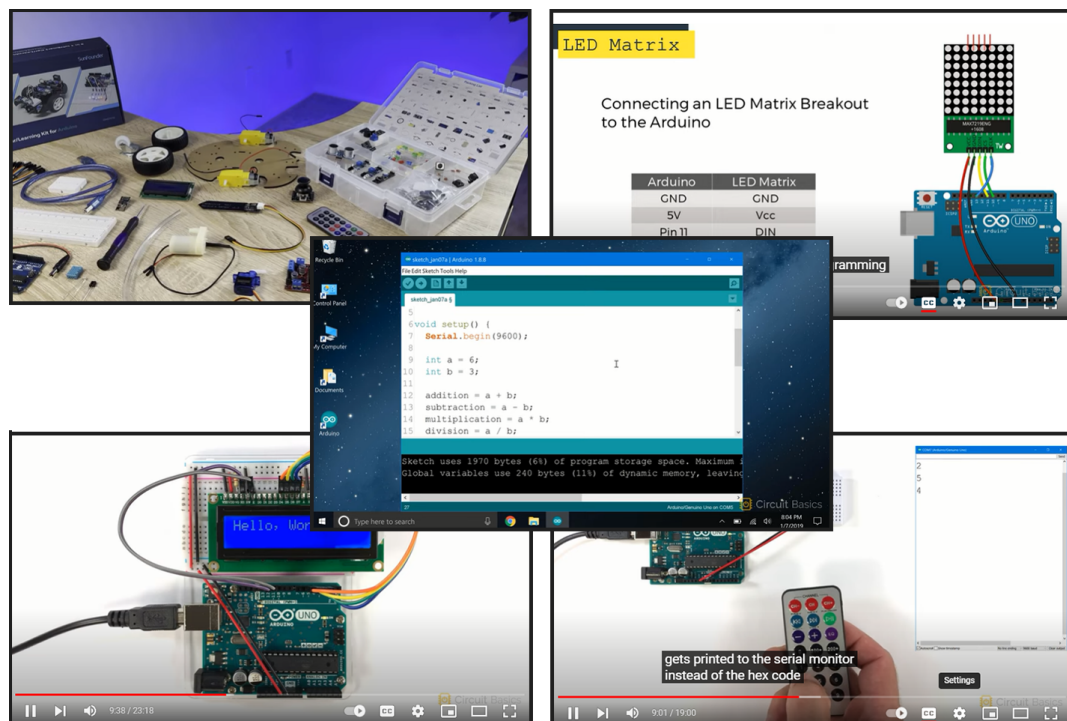


KAPITEL 9

Videokurse

- Für *Grundprojekte* Abschnitte

Wenn Sie den Inhalt der Online-Dokumentation schwer verständlich finden, können Sie Ihr Lernerlebnis durch einen aufbauenden Videokurs verbessern. Diese Videos werden Arduino für Sie ansprechender und visuell attraktiver gestalten.



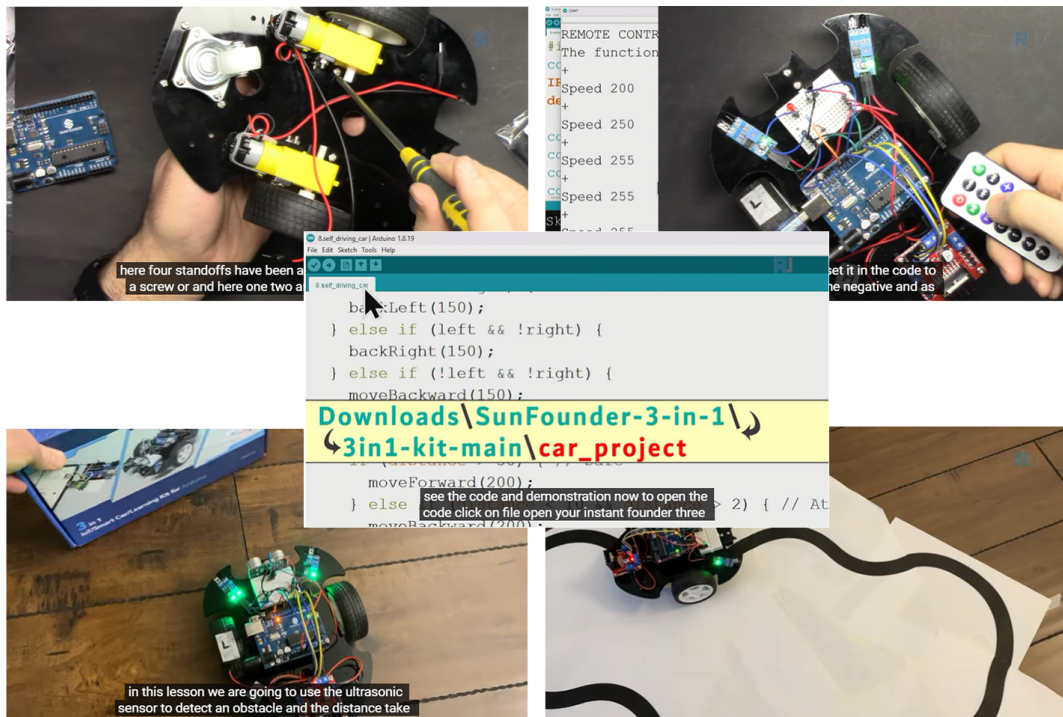
Sie können den Videokurs unter folgendem Link aufrufen: .

In diesen Videos werden Sie faszinierende Experimente und Projekte entdecken, die Arduino-Konzepte und -Prinzipien auf unterhaltsame und interaktive Weise vorstellen. Indem Sie die Videos

ansetzen und an praktischen Aktivitäten teilnehmen, werden Sie ein spannendes und angenehmes Lernerlebnis mit Arduino haben.

- Für die *Auto Projekte* Abschnitte

Ich empfehle, die folgenden YouTube-Playlists anzusehen, um mehr Anleitung und praktische Erfahrung zu sammeln: .

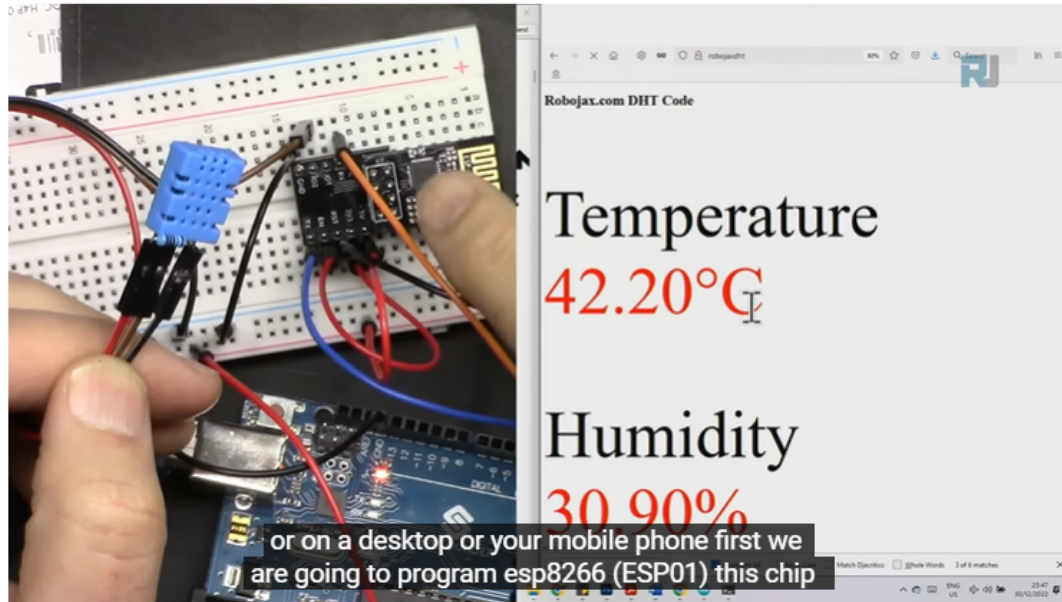


In diesen Videos lernen Sie die Grundlagen der Robotik und Arduino durch fesselnde Video-Lektionen kennen. Schritt für Schritt bauen Sie ein Roboterauto, während Sie die Funktionsweise von Motoren, Hindernisvermeidungsmodulen, Linienverfolgungsmodulen und Infrarotempfängern verstehen. Erforschen Sie, wie das Auto verschiedene Funktionen ausführt und entfalten Sie Ihre Kreativität in der Welt der Robotik und Technologie.

- Über die WiFi-Funktion

Im *IoT-Projekte* Abschnitt unseres Online-Tutorials erfahren Sie, wie Sie mit der IoT-Plattform Blynk kommunizieren können.

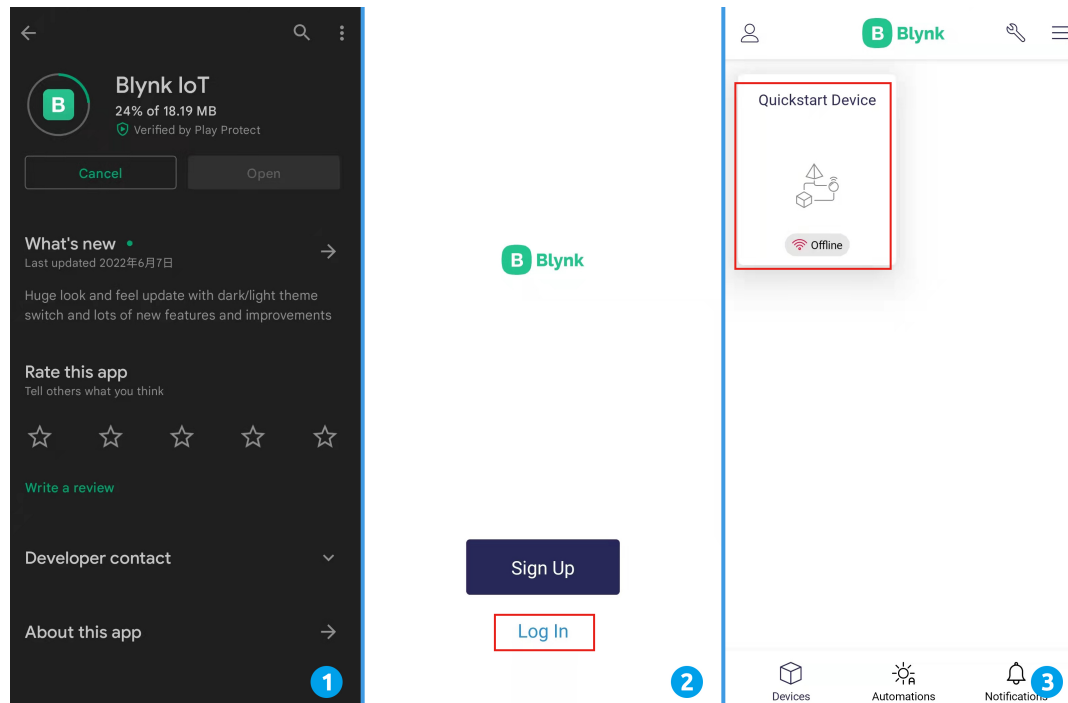
In erhalten Sie eine Anleitung zum Schreiben eines Webservers und zum Hochladen von Sensordaten darauf. Dieses Tutorial zeigt Ihnen, wie Sie eine Verbindung zwischen Ihrem Arduino-Projekt und einem Webserver über WiFi herstellen können.



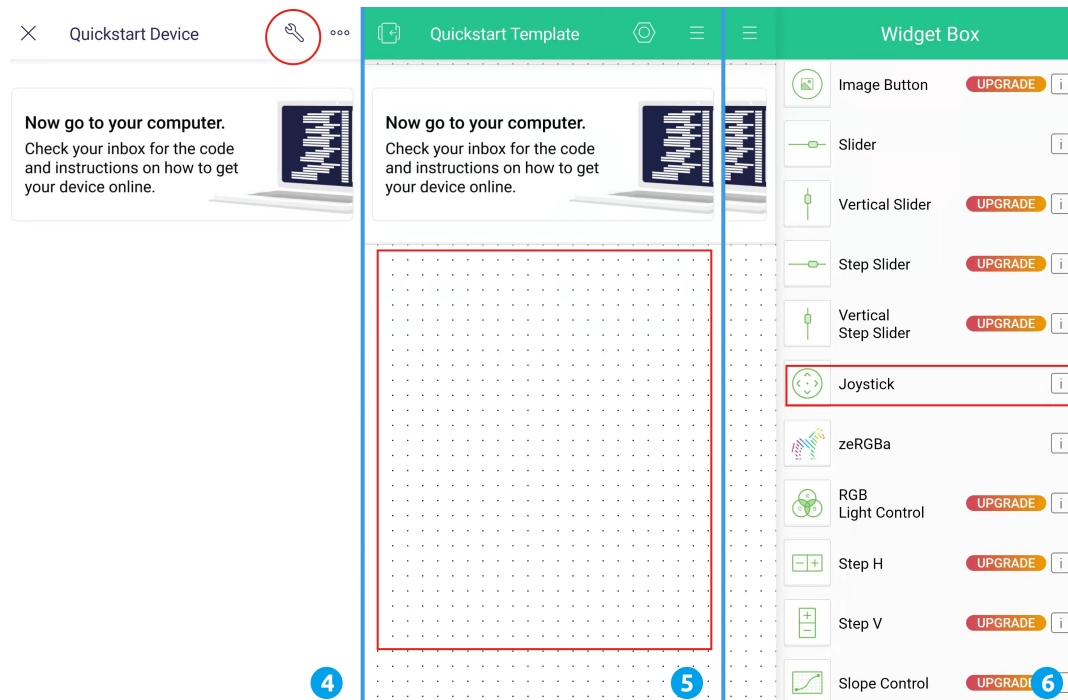
10.1 Wie verwendet man Blynk auf einem mobilen Gerät?

Bemerkung: Da Datenströme nur auf der Blynk-Webseite erstellt werden können, müssen Sie auf verschiedene Projekte verweisen, um Datenströme im Web zu erstellen. Folgen Sie anschließend dem untenstehenden Tutorial, um Widgets in Blynk auf Ihrem mobilen Gerät zu erstellen.

1. Öffnen Sie Google Play oder den APP Store auf Ihrem mobilen Gerät und suchen Sie nach „Blynk IoT“ (nicht Blynk(legacy)), um es herunterzuladen.
2. Nachdem Sie die APP geöffnet haben, melden Sie sich an. Dieses Konto sollte identisch mit dem Konto sein, das Sie im Webclient verwenden.
3. Gehen Sie dann zum **Dashboard** (falls Sie noch keines haben, erstellen Sie eines). Sie werden feststellen, dass das **Dashboard** für Mobilgeräte und Web unabhängig voneinander sind.

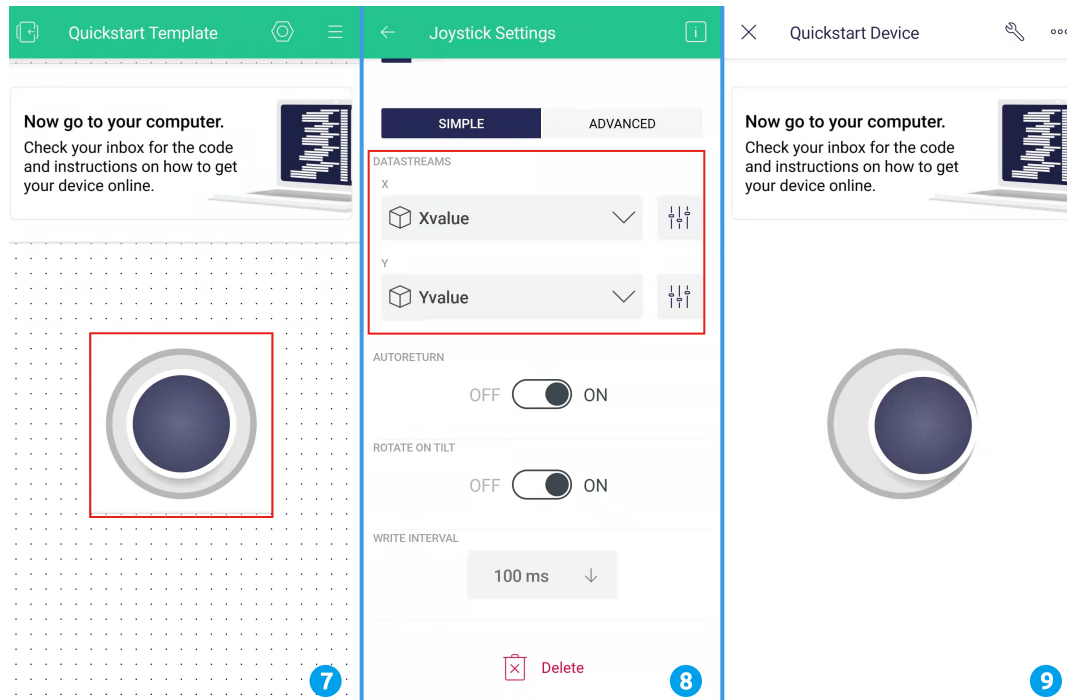


4. Klicken Sie auf das **Bearbeiten**-Symbol.
5. Klicken Sie auf den leeren Bereich.
6. Wählen Sie das gleiche Widget wie auf der Webseite, zum Beispiel ein **Joystick**-Widget.



7. Jetzt sehen Sie ein **Joystick**-Widget im leeren Bereich. Klicken Sie darauf.
8. Die **Joystick**-Einstellungen werden angezeigt. Wählen Sie die Datenströme **Xvalue** und **Yvalue**, die Sie gerade auf der Webseite eingerichtet haben. Beachten Sie, dass jedes Widget einem anderen Datenstrom in jedem Projekt entspricht.

9. Gehen Sie zurück zur **Dashboard**-Seite, und Sie können den **Joystick** nach Belieben bedienen.

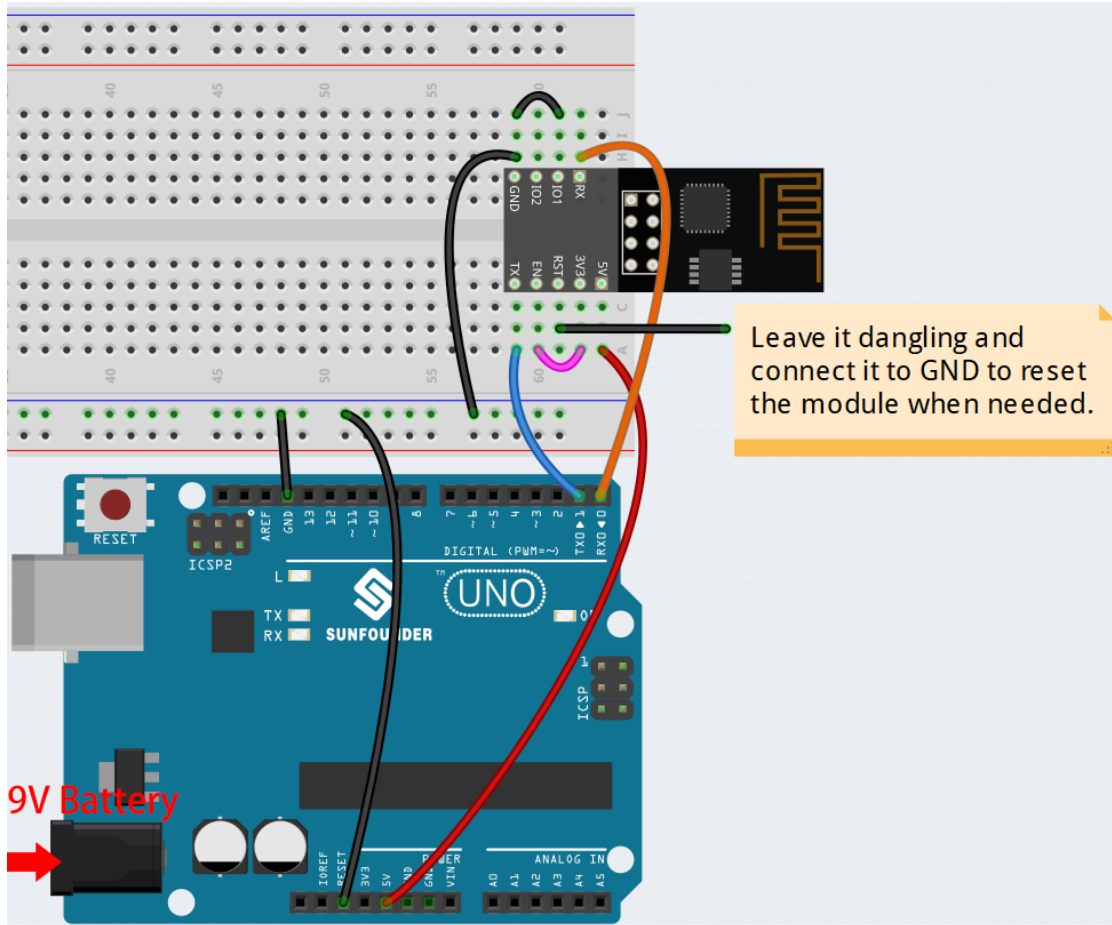


10.2 Wie kann die Firmware für das ESP8266-Modul neu geflasht werden?

10.2.1 Firmware mit R3 neu flashen

1. Schaltkreis aufbauen

Verbinden Sie das ESP8266 mit dem SunFounder R3 Board.



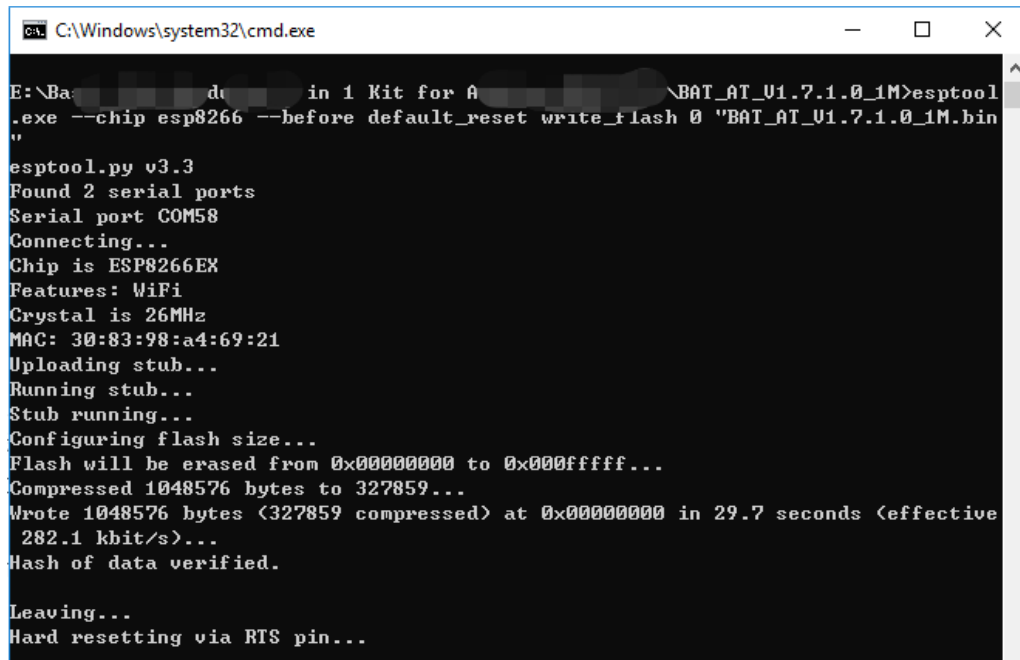
2. Firmware flashen

- Befolgen Sie die folgenden Schritte, um die Firmware unter **Windows** zu flashen.

1. Firmware und Flash-Tool herunterladen.
 - ESP8266 Firmware
2. 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 geflasht wird.
 - esptool.exe: Dies ist ein Kommandozeilen-Tool für Windows.
 - install_r3.bat: Dieses Kommandopak ist für das Windows-System. Ein Doppelklick auf diese Datei führt alle darin enthaltenen Befehle aus.
 - install_r4.bat: Ähnlich wie install_r3.bat, jedoch speziell für das UNO R4 Board.
3. Doppelklicken Sie auf install_r3.bat, um das Flashen 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 Flashen fehlschlägt, überprüfen Sie bitte die folgenden Punkte.

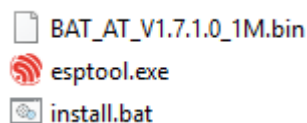
- Setzen Sie das ESP8266-Modul zurück, indem Sie RST am ESP8266-Adapter auf GND setzen und dann wieder entfernen.
- Überprüfen Sie, ob die Verkabelung korrekt ist.
- Stellen Sie sicher, dass der Computer Ihr Board korrekt erkannt hat und der Port nicht belegt ist.
- Öffnen Sie die install.bat-Datei erneut.

- Um die Firmware auf einem **Mac OS**-System zu flashen, befolgen Sie diese Schritte.

1. Verwenden Sie die folgenden Befehle, um Esptool zu installieren. Esptool ist ein Python-basiertes, Open-Source-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 korrekt installiert wurde, 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 3 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 geflasht wird.

- `esptool.exe`: Dies ist ein Kommandozeilen-Tool für Windows.
 - `install_r3.bat`: Dieses Kommandopakete ist für das Windows-System.
 - `install_r4.bat`: Ähnlich wie `install_r3.bat`, jedoch speziell für das UNO R4 Board.
5. Öffnen Sie ein Terminal und navigieren Sie mit dem Befehl `cd` in den gerade heruntergeladenen Firmware-Ordner. Führen Sie dann den folgenden Befehl aus, um die vorhandene Firmware zu löschen und die neue Firmware zu flashen.

```
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.

```
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.

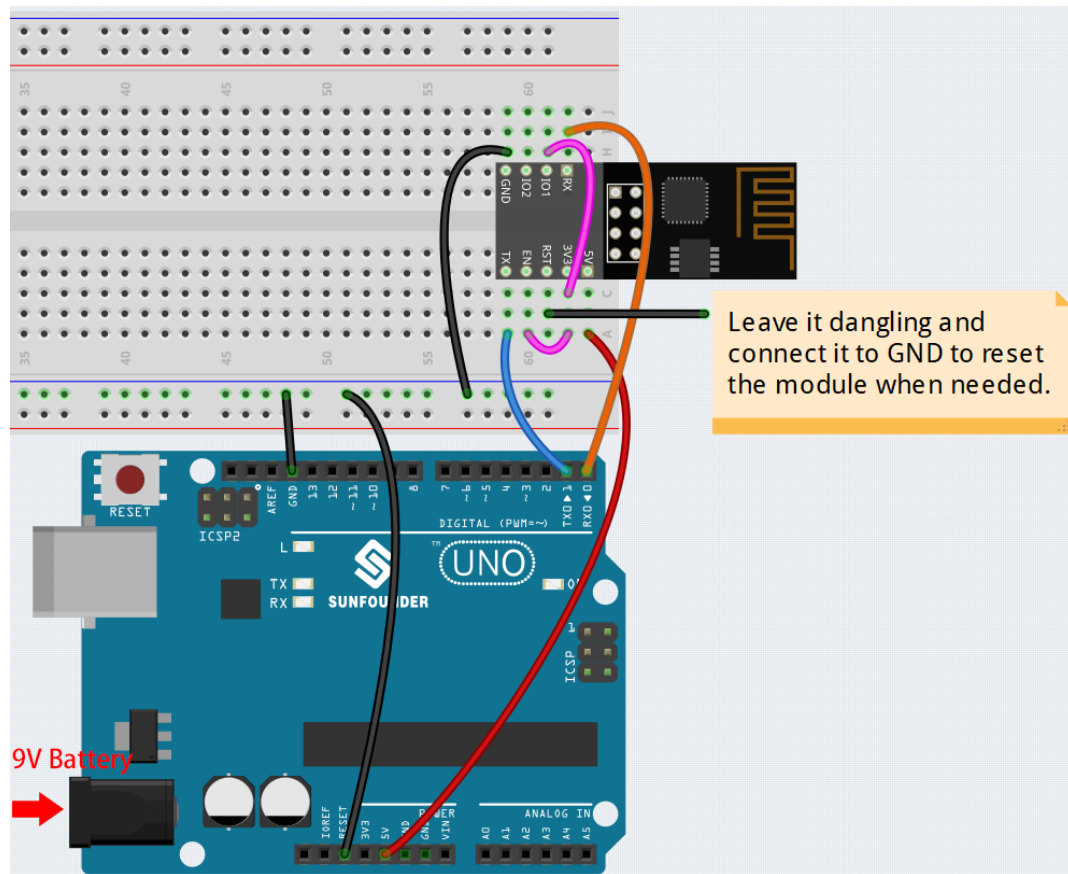
Leaving...
Hard resetting via RTS pin...
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$
```

Bemerkung: Wenn das Flashen fehlschlägt, überprüfen Sie bitte die folgenden Punkte.

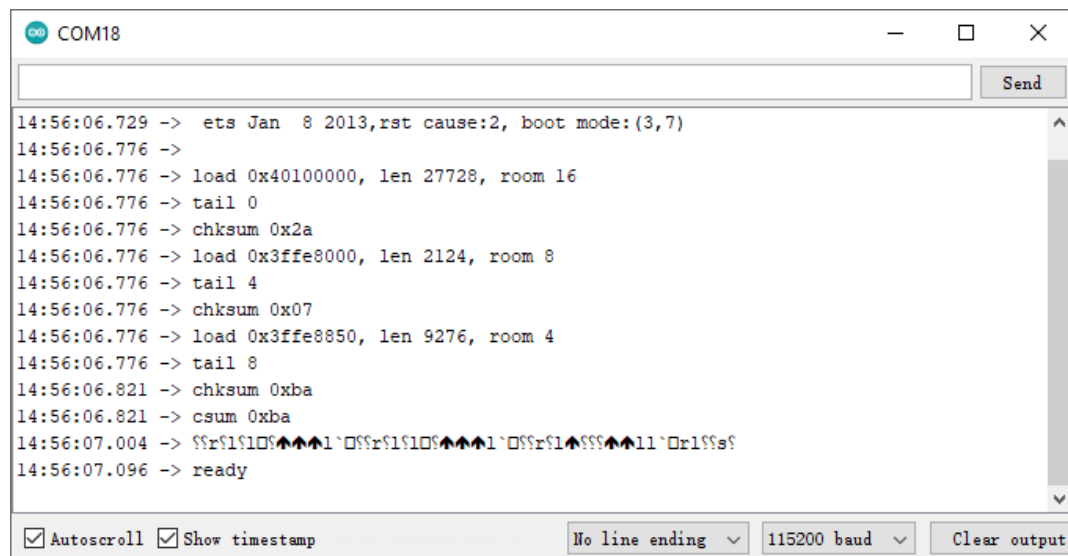
- Setzen Sie das ESP8266-Modul zurück, indem Sie RST am ESP8266-Adapter auf GND setzen und dann wieder entfernen.
- Überprüfen Sie, ob die Verkabelung korrekt ist.
- Stellen Sie sicher, dass der Computer Ihr Board korrekt erkannt hat und der Port nicht belegt ist.
- Öffnen Sie die `install.bat`-Datei erneut.

3. Test

1. Auf Basis der ursprünglichen Verkabelung verbinden Sie IO1 mit 3V3.



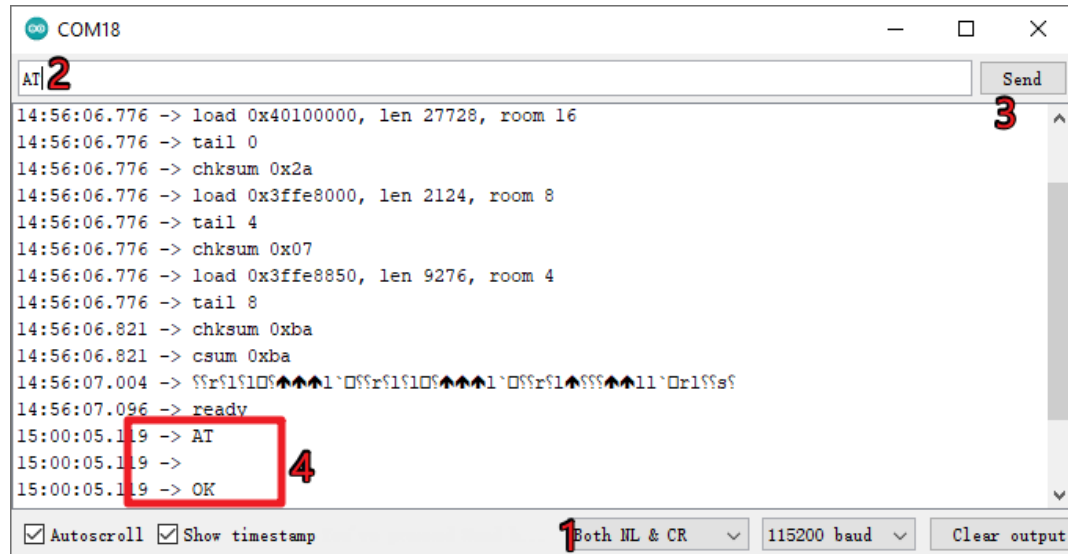
2. Sie können Informationen über das ESP8266-Modul sehen, wenn Sie auf das Lupensymbol (Serial Monitor) in der oberen rechten Ecke klicken und die Baudrate auf **115200** einstellen.



Bemerkung:

- Wenn ready nicht erscheint, können Sie versuchen, das ESP8266-Modul zurückzusetzen (verbinden Sie RST mit GND) und den Serial Monitor erneut zu öffnen.

3. Klicken Sie auf das **NEWLINE DROPDOWN BOX**, wählen Sie both NL & CR aus dem Dropdown-Menü aus, geben Sie AT ein. Wenn die Antwort OK lautet, bedeutet dies, dass das ESP8266 erfolgreich eine Verbindung mit dem R3-Board hergestellt hat.



Jetzt können Sie *1.1 Konfiguration des ESP8266* folgen, um den Arbeitsmodus und die Baudrate des ESP8266-Moduls einzustellen.

10.2.2 Wie man die Firmware mit R4 neu aufspielt

1. Den Schaltkreis aufbauen

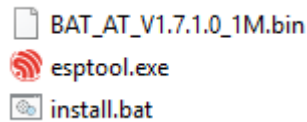
Verbinden Sie ESP8266 und das Arduino UNO R4 Board.



3. Firmware aufspielen

- Folgen Sie diesen Schritten, um die Firmware unter **Windows** aufzuspielen.

- Firmware und Brenn-Tool herunterladen.
 - ESP8266 Firmware
- Nach dem Entpacken werden Sie 4 Dateien sehen.



- BAT_AT_V1.7.1.0_1M.bin: Die auf das ESP8266 Modul aufzuspielende Firmware.
 - esptool.exe: Ein Kommandozeilen-Tool für Windows.
 - install_r3.bat: Ein Befehlspaket für das Windows-System. Ein Doppelklick auf diese Datei führt alle enthaltenen Befehle aus.
 - install_r4.bat: Ähnlich wie install_r3.bat, jedoch speziell für das UNO R4 Board.
- Doppelklicken Sie auf install_r4.bat, um das Aufspielen 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 Aufspielen fehlschlägt, überprüfen Sie bitte die folgenden Punkte:

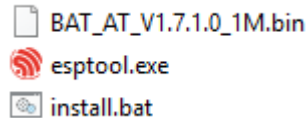
- Setzen Sie das ESP8266-Modul zurück, indem Sie RST am ESP8266-Adapter an GND anschließen und dann wieder entfernen.
- Überprüfen Sie, ob die Verkabelung korrekt ist.
- Stellen Sie sicher, dass Ihr Computer Ihr Board korrekt erkannt hat und der Port nicht belegt ist.
- Öffnen Sie die install.bat-Datei erneut.

- Um die Firmware unter **Mac OS** aufzuspielen, befolgen Sie diese Schritte:

1. Installieren Sie Esptool mit den folgenden Befehlen. Esptool ist ein Python-basiertes, Open-Source-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 korrekt installiert wurde, gibt es eine Nachricht wie [usage: esptool] aus, wenn Sie `python3 -m esptool` ausführen.
3. Firmware herunterladen.
 - ESP8266 Firmware
4. Nach dem Entpacken werden Sie 4 Dateien sehen.



- BAT_AT_V1.7.1.0_1M.bin: Die auf das ESP8266 Modul aufzuspielende Firmware.
 - esptool.exe: Ein Kommandozeilen-Tool für Windows.
 - install_r3.bat: Ein Befehlspaket für das Windows-System.
 - install_r4.bat: Ähnlich wie install_r3.bat, jedoch speziell für das UNO R4 Board.
5. Öffnen Sie ein Terminal und navigieren Sie mit dem `cd`-Befehl in den gerade heruntergeladenen Firmware-Ordner. Führen Sie dann den folgenden Befehl aus, um die vorhandene Firmware zu löschen und die neue Firmware aufzuspielen.

```
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.

```
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$ python3 -m esptool --chip esp8266 --before default_reset write_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.

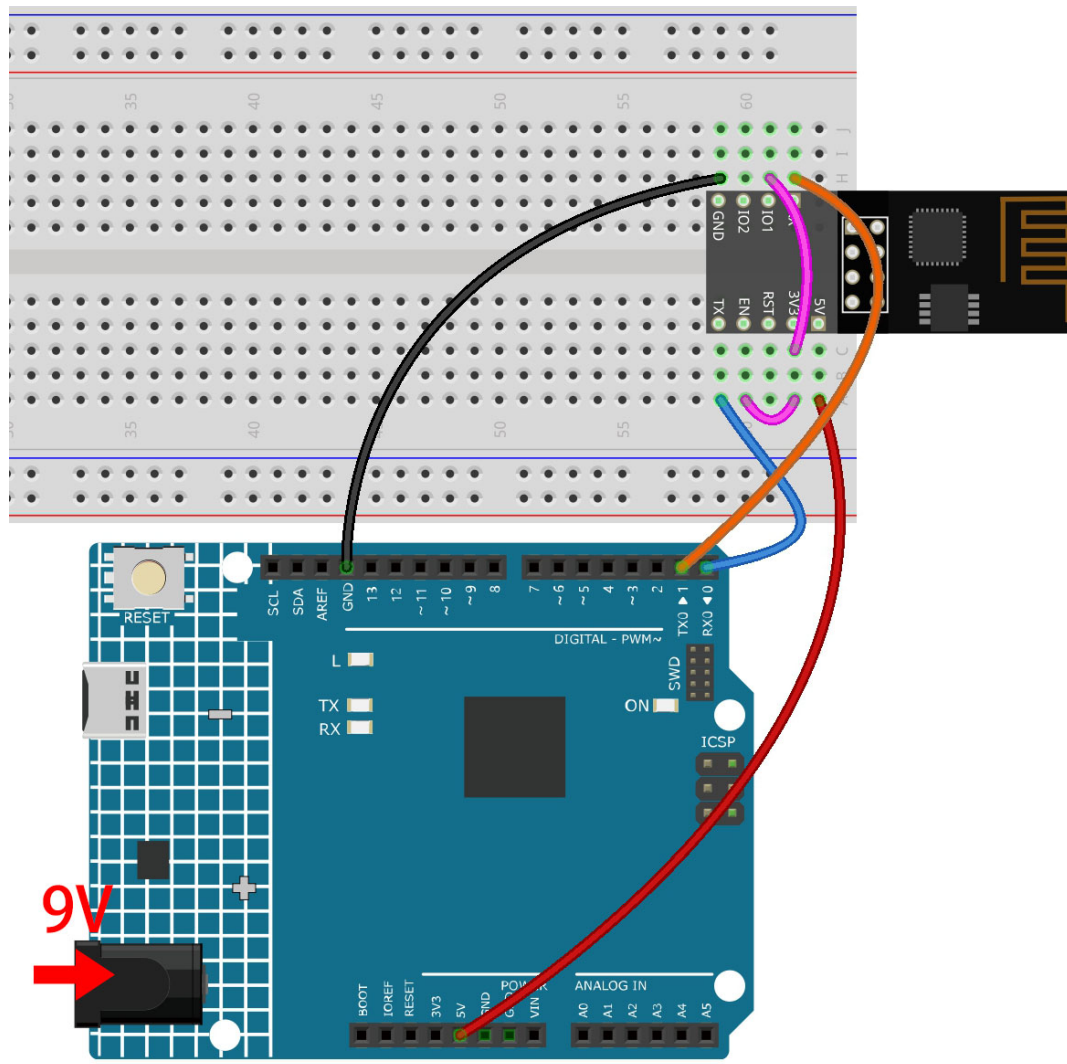
Leaving...
Hard resetting via RTS pin...
MacdeMacBook-Pro:BAT_AT_V1.7.1.0_1M mac$
```

Bemerkung: Wenn das Aufspielen fehlschlägt, überprüfen Sie bitte die folgenden Punkte:

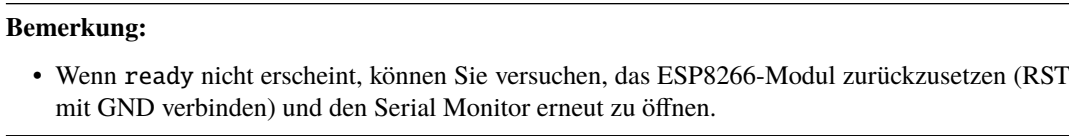
- Setzen Sie das ESP8266-Modul zurück, indem Sie RST am ESP8266-Adapter an GND anschließen und dann wieder entfernen.
- Überprüfen Sie, ob die Verkabelung korrekt ist.
- Stellen Sie sicher, dass Ihr Computer Ihr Board korrekt erkannt hat und der Port nicht belegt ist.
- Öffnen Sie die install.bat-Datei erneut.

4. Testen

1. Auf Basis der ursprünglichen Verkabelung verbinden Sie IO1 mit 3V3.



2. Sie werden Informationen über das ESP8266-Modul sehen, wenn Sie auf das Lupen-Icon (Serial Monitor) in der oberen rechten Ecke klicken und die Baudrate auf **115200** einstellen.



-
- COM18
- AT
- Send
- 14:56:06.776 -> load 0x40100000, len 27728, room 16
 14:56:06.776 -> tail 0
 14:56:06.776 -> checksum 0x2a
 14:56:06.776 -> load 0x3ffe8000, len 2124, room 8
 14:56:06.776 -> tail 4
 14:56:06.776 -> checksum 0x07
 14:56:06.776 -> load 0x3ffe8850, len 9276, room 4
 14:56:06.776 -> tail 8
 14:56:06.821 -> checksum 0xba
 14:56:06.821 -> csum 0xba
 14:56:07.004 -> \$rfl\$10\$^1`0\$rfl\$10\$^1`0\$rfl\$10\$^1`0r1\$se\$
 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

10.2. Wie kann die Firmware für das ESP8266-Modul neu geflasht werden? 655

KAPITEL 11

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 dieses Handbuchs, einschließlich, aber nicht beschränkt auf Texte, Bilder und Code, sind Eigentum der SunFounder Company. Sie dürfen diese nur für persönliches Lernen, Forschung, Vergnügen oder andere nicht-kommerzielle oder gemeinnützige Zwecke verwenden, gemäß den zugehörigen Vorschriften und Urheberrechtsgesetzen, ohne die gesetzlichen Rechte des Autors und der betreffenden Rechteinhaber zu verletzen. Für jede Einzelperson oder Organisation, die diese ohne Erlaubnis für kommerziellen Gewinn verwendet, behält sich das Unternehmen das Recht vor, rechtliche Schritte einzuleiten.