

## Danke, dass Sie sich für die

# Inventor Lab Kit

Beginnen Sie Ihre Elektronikreise mit dem Inventor Lab Kit, das für Lernende und Enthusiasten konzipiert ist. Im Mittelpunkt steht das Arduino Uno R3, und das Kit umfasst alles von grundlegenden Komponenten wie LEDs und Summern bis hin zu fortgeschrittenen Modulen wie RFID-Systemen und Ultraschallsensoren, sowie ein Mini-Multimeter zur Schaltungsanalyse.

Das Kit zeichnet sich durch eine klare pädagogische Struktur aus und führt Anfänger schrittweise in die Programmierung und Schaltungsdesign ein. Anstatt vorgefertigten Code zu verwenden, leitet es Sie Schritt für Schritt an, Ihre eigenen Skripte zu schreiben, was das Verständnis und die Behaltensleistung verbessert. Die Projekte reichen von einfachen Aufgaben wie dem Lichtanlassen von LEDs und der Verwendung eines Joysticks bis hin zu komplexen Anwendungen wie dem Bau eines Radarsystems und einem automatischen Seifenspender.

Ideal für Anfänger sowie für diejenigen, die ihre Fähigkeiten erweitern möchten, macht das Inventor Lab Kit das Lernen sowohl zugänglich als auch spannend und bietet alle notwendigen Werkzeuge, um die Welt der Elektronik zu erkunden und zu innovieren.

Bei Fragen oder Unterstützung wenden Sie sich bitte an [service@sunfounder.com](mailto:service@sunfounder.com). Tauchen Sie mit dem Beginner's Lab Kit in Ihre Lernreise ein und beginnen Sie mit dem Bauen, Programmieren und Erkunden der aufregenden Welt der Elektronik!

# Inhalt

Was in Ihrem Kit enthalten ist .....	1
Lektion 2: Ihr erster Schaltkreis .....	15
Lektion 3: Messen mit dem Multimeter .....	16
Lektion 4: Ohmsches Gesetz .....	17
Lektion 5: Serienschaltung vs. Parallelschaltung .....	18
Lektionen 6 : Blink-LED .....	22
Lektionen 7: Lass uns Ampeln bauen! .....	23
Lektionen 8 : Verkehrsampel mit Fußgängertaste .....	24
Lektionen 9: Dimmbare Schreibtischlampe .....	26
Lektion 10: Morsecode .....	27
Lektionen 11: Die Farben des Regenbogens .....	28
Lektion 12: Sirenenton .....	30
Lektion 13: Joystick-LED-Navigator .....	32
Lektionen 14: Dinosaurier-Spiel spielen .....	34
Lektion 15: Kalte oder Warme Farben .....	35
Lektion 16: Sommerventilator .....	36
Lektion17: Erforschung des I2C LCD1602 Displays .....	37
Lektion18: Ein/Aus Schreibtischlampe .....	38
Lektion19: Intelligente Mülltonne .....	40
Lektion 20: Automatischer Seifenspender .....	41
Lektionen 21: Temperaturalarm .....	42
Lektionen22: Fernbedienbare Farbenfrohe Beleuchtung .....	44
Lektion 23: Spiele "Blink, Blink, kleiner Stern" .....	45
Lektionen 24: Der Pomodoro-Timer .....	46
Lektionen25: Umgekehrtes Radarsystem .....	48
Lektionen26: Cyberwürfel .....	49
Lektion 27: Fließendes Licht mit 74HC595 .....	51
Lektion 28: Zahl anzeigen .....	52
Lektion 29: Pflanzenmonitor .....	54
Lektion 30: Arduino Radar System .....	55
Lektion 31: Zahl erraten .....	56
Lektionen 32: Stoppuhr .....	57
Lektion 33: Erkundung des RF522-RFID-Moduls .....	59
Lektionen 34: Zugangskontrollsystem .....	60

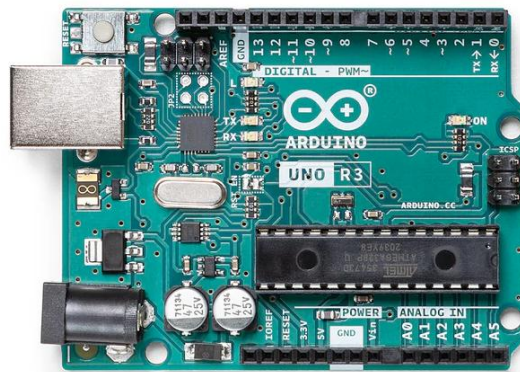
# Was in Ihrem Kit enthalten ist

In unserem Kit finden Sie eine Vielzahl von Komponenten und Teilen, die Sie während dieses Kurses zum Bau von Schaltungen verwenden werden. Hier ist eine kurze Übersicht über die enthaltenen Teile.

## Grundlegende Komponenten

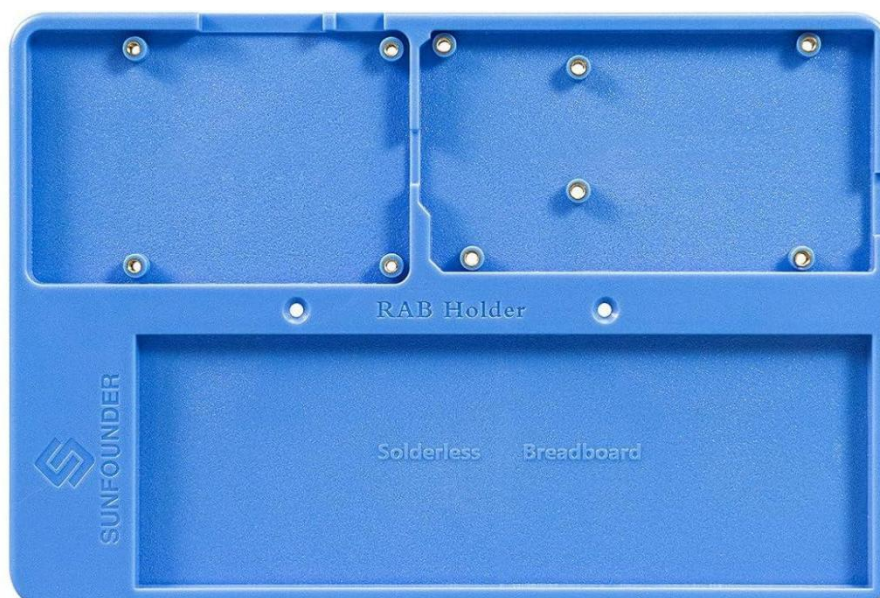
### 1 x Original Arduino Uno R3

Ein Mikrocontroller-Board, das das Gehirn Ihrer Schaltungen ist. Es enthält alles, was zur Unterstützung des Mikrocontrollers benötigt wird; schließen Sie es einfach mit einem USB-Kabel an Ihren Computer an oder speisen Sie es mit einem AC-zu-DC-Adapter oder einer Batterie, um loszulegen.



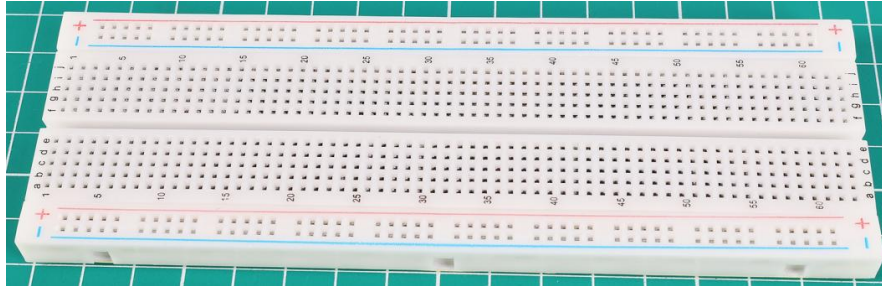
### 1 x RAB Holder

Entwickelt für Benutzer von Arduino und Raspberry Pi bietet es Stabilität für all Ihre Geräte. Neben der Halterung dient es als Werkzeug für Innovation und ist für jeden geeignet.



## 1 x 830-hole Breadboard

Eine lötfreie Platine, die es Ihnen ermöglicht, elektronische Schaltungen einfach zu erstellen. Sie ist mit Reihen von Löchern gefüllt, um Drähte und Komponenten zu verbinden.



## 1 x Multimeter mit roten und schwarzen Kabeln

Dies ist ein vielseitiges Multimeter, das in der Lage ist, Spannung, Strom und Widerstand zu messen sowie weitere elektrische Tests durchzuführen. Es ist ein unverzichtbares Werkzeug für die Elektronik- und Elektrotechnik.



## 1 x Breadboard Power Module

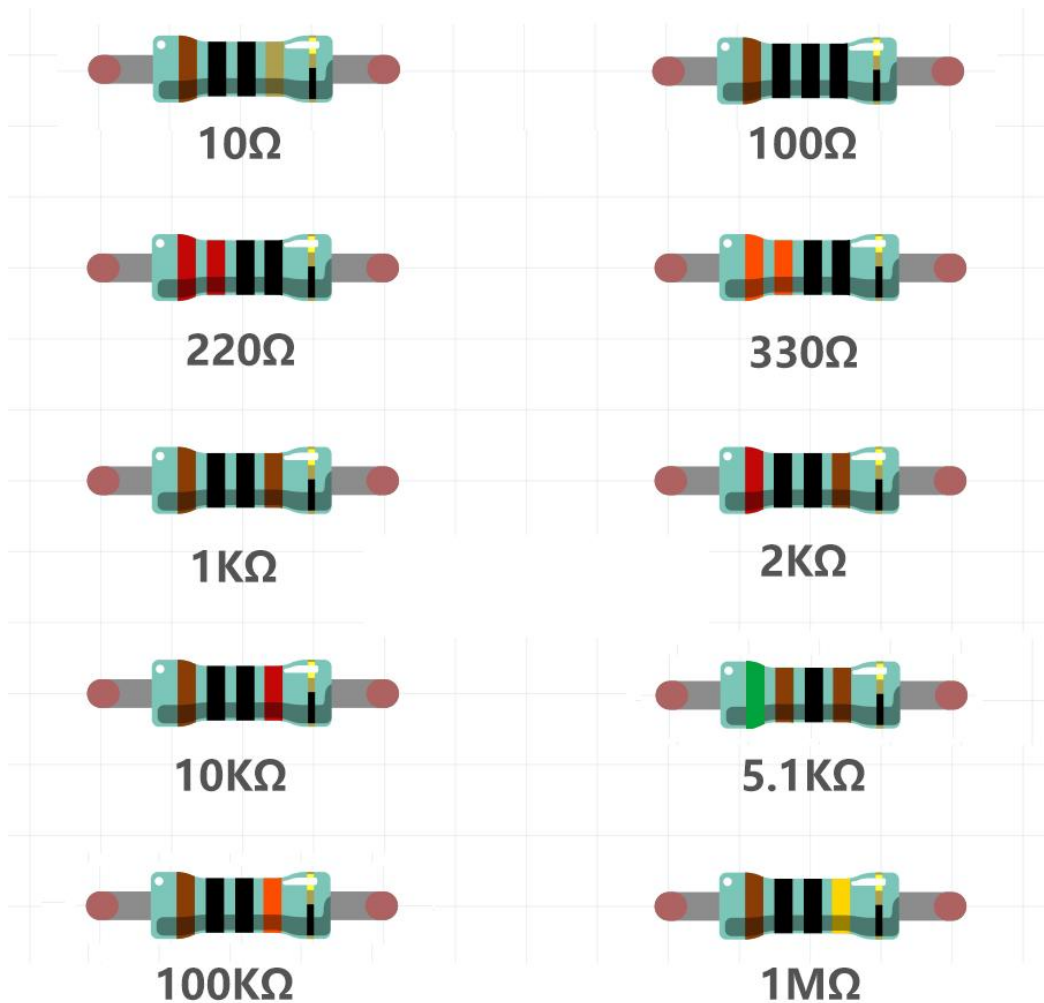
Das Breadboard-Strommodul ist ein praktisches Zubehör für Prototypen und bietet eine stabile 3,3V- oder 5V-Stromversorgung über einen DC-Adapter oder USB. Es passt auf Standard-Breadboards, wird in die Stromschienen gesteckt und verfügt über einen Ein/Aus-Schalter sowie Spannungsregler für eine konstante Ausgabe, was es zu einem unverzichtbaren Bestandteil von Elektronikprojekten macht.





## 120 x Widerstände (je 10 Stück von jedem, 30 Stück von 220Ω-Widerständen)

Ein Widerstand ist ein Bauteil, das den Fluss elektrischer Energie hemmt und somit die Spannung und den Strom in einem Stromkreis verändert. Der Wert eines Widerstands wird in Ohm gemessen, symbolisiert durch den griechischen Buchstaben Omega ( $\Omega$ ). Die farbigen Streifen auf einem Widerstand geben dessen Widerstandswert und Toleranz an.



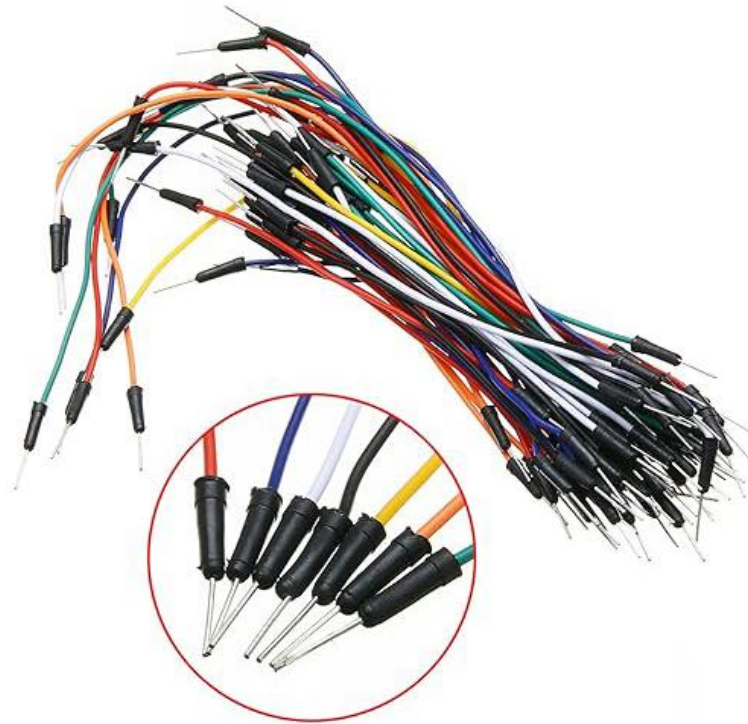
## 2 x 9V Battery

Dies ist eine nicht wiederaufladbare alkalische 9V-Batterie. Sie müssen sie im Multimeter installieren oder das Batteriekabel verwenden, um das Arduino Uno R3 oder das Breadboard-Netzmodul mit Strom zu versorgen.



## 65 x Jumperdrähte

Verbinden Sie die Komponenten auf dem Breadboard miteinander und mit der Arduino-Platine.



## 20 x Male-to-Female DuPont-Kabel

Male-to-Female DuPont-Kabel sind speziell für den Anschluss von Modulen mit männlichen Pinleisten, wie dem Ultraschallmodul, an das Breadboard konzipiert. Diese Kabel sind entscheidend für die Verbindung verschiedener Komponenten in Elektronikprojekten, bei denen Breadboard-kompatible Male-to-Female-Verbindungen erforderlich sind.



## 1 x USB Kabel

Verbindet das Arduino-Board mit einem Computer. Ermöglicht das Schreiben, Kompilieren und Übertragen von Programmen auf das Arduino-Board und versorgt das Board mit Strom.



## 1 x Batteriekabel

Dieses Kabel verbindet eine 9V-Batterie mit dem DC-Eingang eines Breadboard-Netzmoduls oder eines Arduino Uno R3. Es bietet eine praktische und tragbare Stromquelle für Ihre Elektronikprojekte.



## Displays

### 25 x LEDs (5 in jeder Farbe)

Diese bunte LED-Auswahl umfasst fünf Farben: Rot, Grün, Blau, Gelb und Weiß, die verschiedenen Beleuchtungs- und Signalisierungsbedürfnissen gerecht werden. Sie eignen sich für Anwendungen von einfachen Statusanzeigen bis hin zu komplexen dekorativen Beleuchtungsprojekten und bieten eine breite Farbpalette, um die visuelle Attraktivität jedes Elektronikprojekts zu steigern.



## 1 x RGB LED

Kombiniert rote, grüne und blaue LEDs in einem Gehäuse. Durch Anpassung der Eingangsspannung können verschiedene Farben angezeigt werden, wodurch Millionen von Farben entstehen.



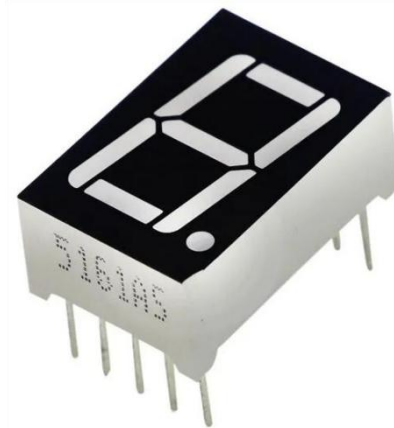
## 1 x 74HC595 Chip

Der 74HC594 ist ein Schieberegister, das verwendet wird, um die Ein- und Ausgangsports von digitalen Schaltungen zu erweitern, indem es serielle Eingaben in parallele Ausgaben umwandelt und somit die Anzahl der benötigten Verbindungspins reduziert. Dieser Chip eignet sich zur Steuerung einer Vielzahl von Ausgabegeräten, wie beispielsweise 7-Segment-Displays, ohne zu viele Pins des Mikrocontrollers zu belegen.



## 1 x 7-segment Display (Gemeinsame Kathode)

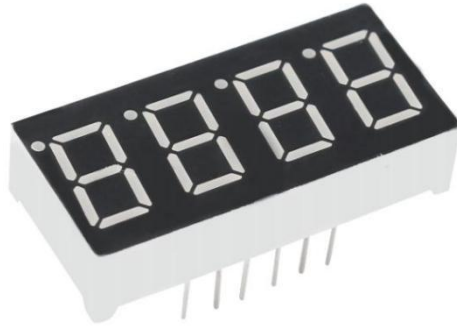
Ein 7-Segment-Display ist ein bauteilförmiger LED-Baustein, der sieben LEDs enthält. Jede LED wird als Segment bezeichnet – wenn sie aktiviert wird, bildet ein Segment einen Teil der anzuzeigenden Zahl.





## 1 x 4-digit 7-segment Display (Gemeinsame Kathode)

Ein 4-stelliges Display kombiniert vier 7-Segment-Displays, von denen jedes eine einzelne Ziffer darstellt. Um die benötigte Anzahl an Pins zu reduzieren, werden die Segmente jedes Displays multiplexed, was bedeutet, dass jeder Segment-Pin mit den entsprechenden Segment-Pins der anderen Displays verbunden ist.



## 1 x I2C LCD1602

Das I2C LCD1602 ist ein 16x2 Zeichen-Display-Modul, das das I2C-Kommunikationsprotokoll verwendet. Dieses Modul eignet sich hervorragend zur Anzeige von Text, wie z.B. Sensordaten oder Statusmeldungen, in Ihren Projekten.



## Aktuatoren

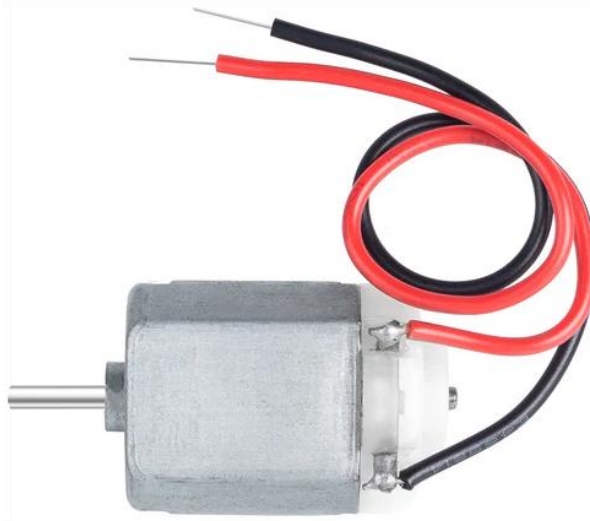
### 1 x L293D Chip

Der L293D ist ein dualer H-Brücken-Motortreiber-IC, der es Ihnen ermöglicht, die Geschwindigkeit und Richtung von zwei Gleichstrommotoren gleichzeitig zu steuern. Er eignet sich hervorragend für Robotik- und Automatisierungsprojekte und bietet eine zuverlässige sowie effiziente Motorsteuerung.



## 1 x Motor

Der 3V-Motor ist ein kompakter und effizienter Gleichstrommotor, der für Niederspannungsanwendungen konzipiert wurde. Er eignet sich hervorragend für kleine Elektronikprojekte, Spielzeuge und Hobbyrobotik und bietet zuverlässige Leistung bei niedrigem Stromverbrauch.



## 1 x Pump

Dies ist die DC 2,5-6V Mini-Unterwasserpumpe, die sich ideal für kleinere Projekte wie Tischbrunnen, Aquarien und Hydrokultursysteme eignet. Diese Pumpe nutzt zentrifugale Mechanik und verwendet einen Elektromotor, um rotierende Energie in fluiddynamische Energie umzuwandeln, wodurch Wasser effizient durch das System bewegt wird.



## 1 x Servo

Ein Servomotor ist ein präziser und vielseitiger Motor, der zur genauen Steuerung der Winkel- oder Linearposition, Geschwindigkeit und Beschleunigung eingesetzt wird. Er wird häufig in der Robotik, Automatisierung und Fernsteuersystemen verwendet und sorgt für zuverlässige und sanfte Bewegungen in verschiedenen Anwendungen.



## 1 x Tube

Dies ist ein 20 cm langer, 6 mm durchmessender transparenter Schlauch, der verwendet wird, um Wasser vom Auslass einer Wasserpumpe zu leiten.



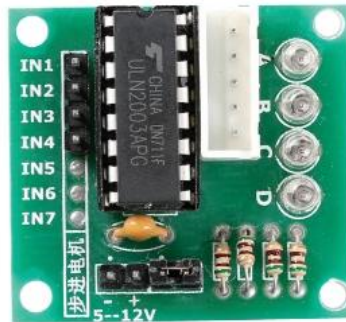
## 1 x Schrittmotor

Der 28BYJ-48 ist ein 5-poliger unipolarer Schrittmotor, der mit 5 V betrieben wird. Er eignet sich hervorragend für Anwendungen, die eine präzise Drehkontrolle erfordern, wie beispielsweise in der Robotik, 3D-Druckern und Automatisierungsprojekten.



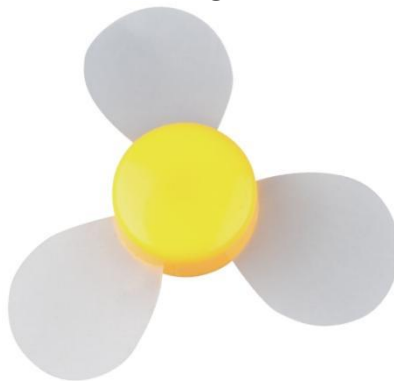
## 1 x ULN2003 Module

Das ULN2003-Modul ist ein Hochspannungs Hochstrom Darlington Transistorarray, das zum Ansteuern von Schrittmotoren, Relais und anderen induktiven Lasten verwendet wird. Es verfügt über sieben Open-Collector-Darlington-Paare, was es ideal für die Schnittstelle mit TTL- und CMOS-Logikpegeln in verschiedenen Steuerungsanwendungen macht.



## 1 x 3-blättrige Ventilatorblatt

Das weiche 3-blättrige Ventilatorblatt ist ein flexibles und sicheres Zubehörteil für 3V-Motoren. Hergestellt aus weichen, langlebigen Materialien minimiert es das Verletzungsrisiko.



## Sounds

### 1 x Active Buzzer & 1 x Passive Buzzer

Ein Summer, erhältlich in aktiven und passiven Varianten, ist ein akustisches Signalgerät, das Geräusche erzeugt, wenn elektrischer Strom angelegt wird. Er wird häufig in Alarmanlagen, Timern und Benachrichtigungssystemen eingesetzt.





## Sensors

### 1 x Fotowiderstand

Ein Fotowiderstand ist ein lichtempfindliches Bauteil, das seinen Widerstand je nach Lichtintensität ändert. Er ist ideal für die Erstellung von lichtaktivierten Steuerungen und Sensoren in elektronischen Projekten.



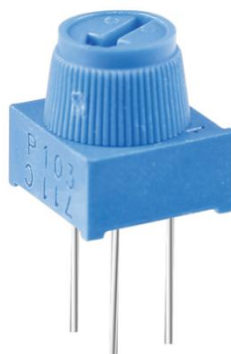
### 1 x NTC Thermistor

Ein Thermistor ist ein Widerstand, der empfindlich auf Temperaturänderungen reagiert. NTC-Thermistoren verringern ihren Widerstand, wenn die Temperatur steigt, während PTC-Thermistoren ihren Widerstand mit steigender Temperatur erhöhen.



### 1 x Potentiometer

Ein Potentiometer ist ein variabler Widerstand mit drei Anschlüssen. Zwei Anschlüsse verbinden die Enden eines Widerstands, während der mittlere Anschluss mit einem beweglichen Wischer verbunden ist, der den Widerstand in zwei Teile teilt. Potentiometer, die häufig zur Anpassung der Spannung in Schaltungen verwendet werden, sind vergleichbar mit den Lautstärkereglern an Radios.



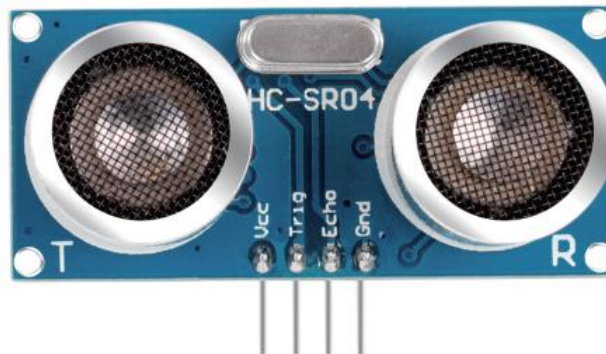
## 10 x Small Buttons

Ein kleiner Druckknopf wird verwendet, um beim Drücken eine physische Reaktion zu erzeugen. Er findet häufig Anwendung in elektronischen Geräten, um Aktionen auszulösen oder Befehle einzugeben.



## 1 x Ultrasonic Module

Dies ist ein Ultraschallmodul, das Ultraschallwellen zur Messung von Distanzen verwendet und die Position sowie den Abstand von Objekten präzise erkennt und misst. Es wird häufig in der Robotik, in Hindernisvermeidungssystemen und im Bereich der automatischen Steuerung eingesetzt und ist ein entscheidendes Element für die Umweltwahrnehmung und räumliche Navigation.



## 1 x Joystick Module

Ein Joystick-Modul, auch bekannt als Joystick-Sensor, ist ein Eingabegerät, das die Bewegung eines Knopfs in zwei Richtungen misst: horizontal (X-Achse) und vertikal (Y-Achse).



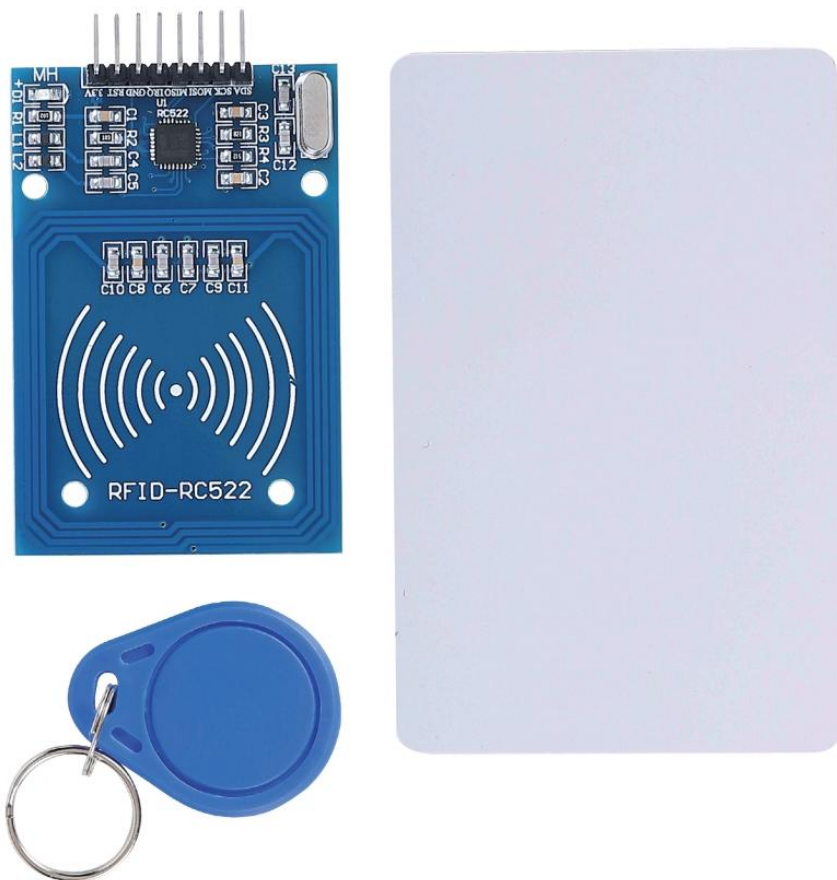
## 1 x Bodenfeuchtesensor-Modul

Ein kapazitiver Sensor zur Erfassung der Bodenfeuchtigkeit, korrosionsbeständig und arbeitet im Bereich von 3,3 V bis 5,5 V. Gibt den Feuchtigkeitswert aus, wobei nasser Boden einen kleineren analogen Wert ergibt.



## 1 x RC522-RFID Module with a Tag and a White Card

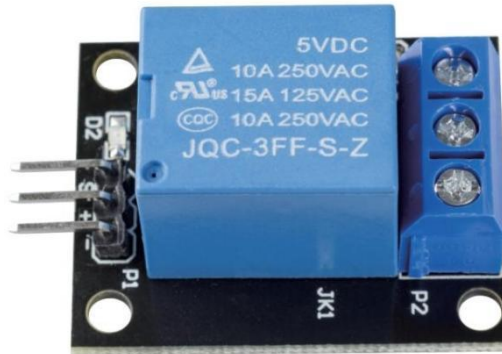
The RC522 RFID reader module, operating at a frequency of 13.56MHz, is designed to communicate with RFID tags adhering to the ISO 14443A standard. This compact and versatile device is ideal for applications in access control, inventory tracking, and contactless payment systems due to its ability to interface with microcontrollers via a 4-pin SPI connection, supporting data rates up to 10 Mbps.



## Andere

### 1 x Relaismodul

Ein Relaismodul ermöglicht es Mikrocontrollern, Hochspannungsgeräte zu steuern, indem es einen elektrisch isolierten Schalter bereitstellt. Es ist ideal für Anwendungen, die die Steuerung von Wechselstrom- oder Hochstromlasten mit Niederspannungs-Digitalsignalen erfordern.



### 1 x IR-Empfänger

Der SL838-Infrarotempfänger ist ein Bauteil, das Infrarotsignale empfängt und unabhängig Infrarotstrahlen empfangen sowie Signale im TTL-Niveau ausgeben kann. Er hat eine ähnliche Größe wie ein gewöhnlicher in Kunststoff verpackter Transistor und eignet sich für alle Arten von Infrarotfernbedienungen und Infrarotübertragungen.



### 1 x Fernbedienung

Diese kompakte Fernbedienung mit 21 Tasten misst 85x39x6 mm, hat eine Reichweite von 8 Metern und wird von einer 3V-Lithiumbatterie betrieben. Mit einer Infrarotfrequenz von 38 kHz und einer robusten PET-Oberfläche gewährleistet sie über 20.000 Einsätze und ist somit ideal für verschiedene Geräte.

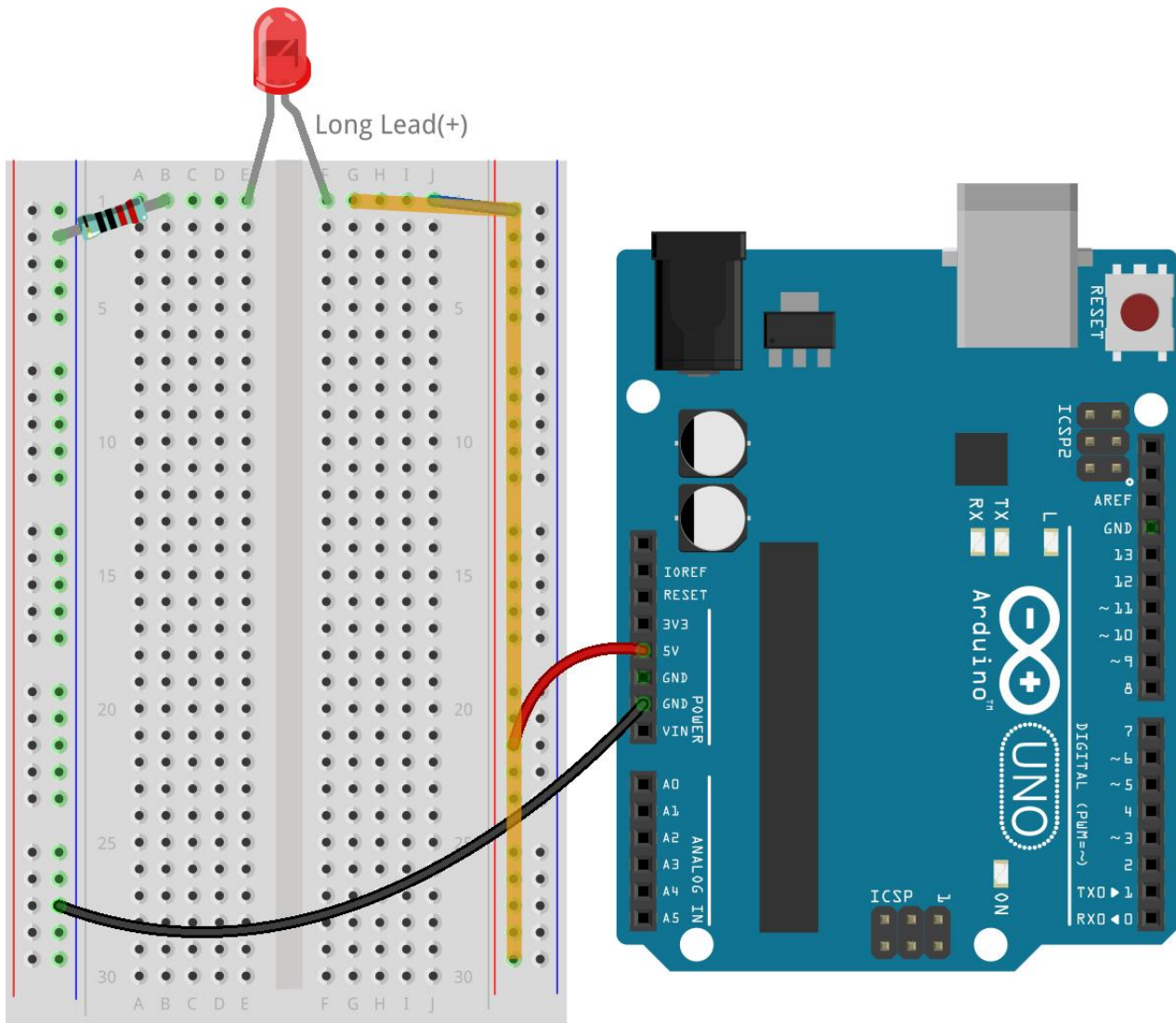




## Lektion 2: Ihr erster Schaltkreis

Beantworten Sie nach Abschluss der Lektion die folgenden Fragen.

1. Entfernen Sie den roten Draht von der Steckplatine und experimentieren Sie, indem Sie ihn in verschiedene Löcher der Steckplatine stecken. Beobachten Sie die Änderungen bei der LED. Skizzieren Sie die Lochpositionen, die es der LED ermöglichen, zu leuchten.



1. Was passiert, wenn Sie die Pins der LED umkehren? Wird sie leuchten? Warum oder warum nicht?

Die LED leuchtet nicht, da sie eine unidirektionale Leitfähigkeit aufweist; der Strom muss vom Anoden- zum Kathodenanschluss fließen, damit sie funktioniert.

## Lektion 3: Messen mit dem Multimeter

Beantworte diese Frage, nachdem du **“Mehr über das Multimeter erfahren!”** abgeschlossen hast.

---

Jetzt, da du ein detailliertes Verständnis dafür hast, wie man ein Multimeter verwendet, überlege, welche Multimeter-Einstellung du verwenden würdest, um die folgenden elektrischen Werte zu messen.

Messobjekt	Messbereich des Multimeters
9V volts DC	20V
1K ohms	2k $\Omega$
40 milliamps	200mA
110 volts AC	200V~

Füllen Sie diese Tabelle während des Themas **“Messen mit einem Multimeter”** aus.

---

Typ	Einheiten	Messwerte	Anmerkungen
Spannung	Volts	$\approx 5.13$ volts	
Strom	Milliamps	$\approx 13.54$ milliamps	
Widerstand	Ohms	$\approx 378.88$ ohms	

## Lektion 4: Ohmsches Gesetz

Füllen Sie die folgende Tabelle während „Ohmsches Gesetz durch praktische Experimente erkunden“ aus

1. Ersetzen Sie den 220-Ohm-Widerstand durch andere Widerstände mit unterschiedlichen Werten, wie unten aufgeführt. Notieren Sie die Helligkeitsänderungen der LED bei jedem Austausch, um zu beobachten, wie der Widerstand den Strom und somit die Lichtausgabe beeinflusst.

Widerstand	Beobachtungen
100Ω	Heller
1KΩ	Hell
10KΩ	Dunkler
1MΩ	Nahezu Aus

Sie werden feststellen, dass die LED nur mit dem 100Ω-Widerstand heller ist als mit dem vorherigen 220Ω-Widerstand. Bei höheren Widerständen nimmt die Helligkeit der LED ab, bis sie bei 1MΩ vollständig erlischt. Warum ist das so?

Laut Ohms Gesetz ( $I = V/R$ ) verringert sich der Strom durch die LED, während der Widerstand zunimmt und die Spannung konstant bleibt, was zu einer Abschwächung der LED führt. Bei 1MΩ ist der Strom zu gering, um die LED zum Leuchten zu bringen.

1. Nach der Beobachtung der Auswirkungen der Widerstandsänderung halten Sie den Widerstand auf 220 Ohm und ändern die Spannungsversorgung des Schaltkreises von 5V auf 3,3V. Notieren Sie alle Veränderungen in der Helligkeit der LED.

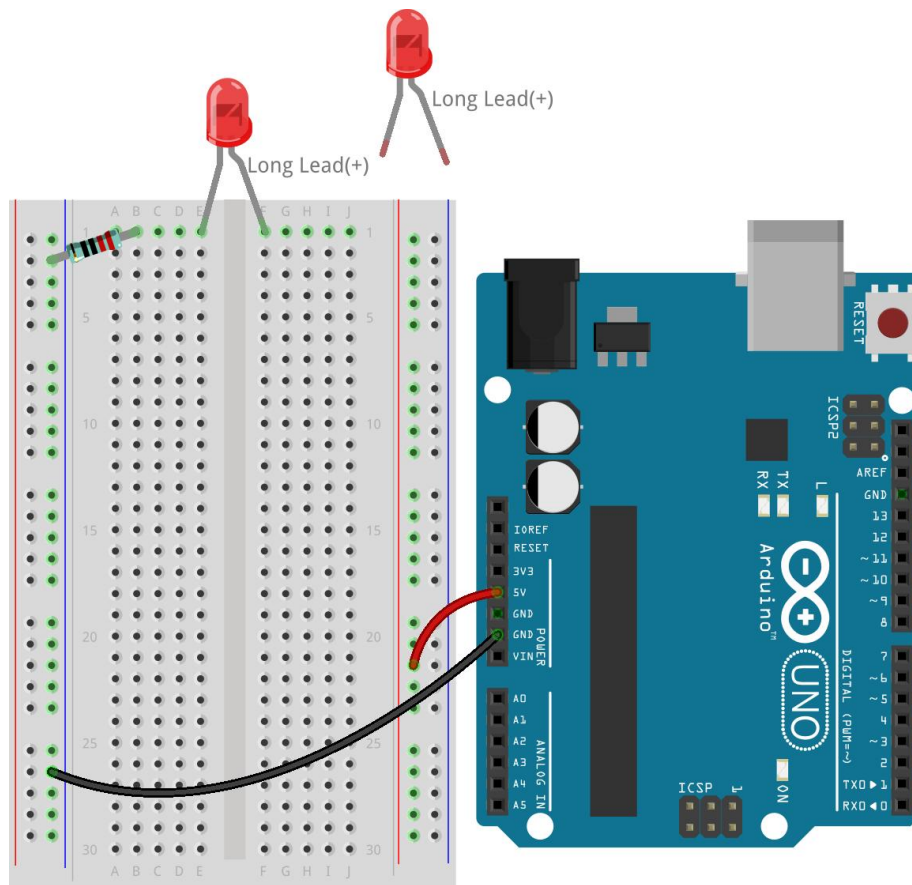
Sie werden feststellen, dass die LED bei 3,3V etwas dunkler ist als bei 5V. Warum ist das so?

Nach dem ohmschen Gesetz, wenn der Widerstand und die neue Spannung bekannt sind, sollte der Strom  $I = V/R$  betragen. Mit einer Verringerung der Spannung bei gleichbleibendem Widerstand nimmt der Strom ab, was die LED dimmt.

# Lektion 5: Serienschaltung vs. Parallelschaltung

Beantworten Sie die folgenden Fragen während des Themas “Eintauchen in Serienschaltungen”.

1. Was passiert, wenn Sie eine LED entfernen? Warum geschieht das?



In einem Serienschaltkreis, wenn Sie eine LED entfernen, wird die andere LED nicht leuchten. Das liegt daran, dass in einem Serienschaltkreis der Strom durch jede Komponente im Stromkreis fließen muss. Das Entfernen einer LED unterbricht den Stromkreis und verhindert, dass der Strom durch die verbleibende LED fließt.



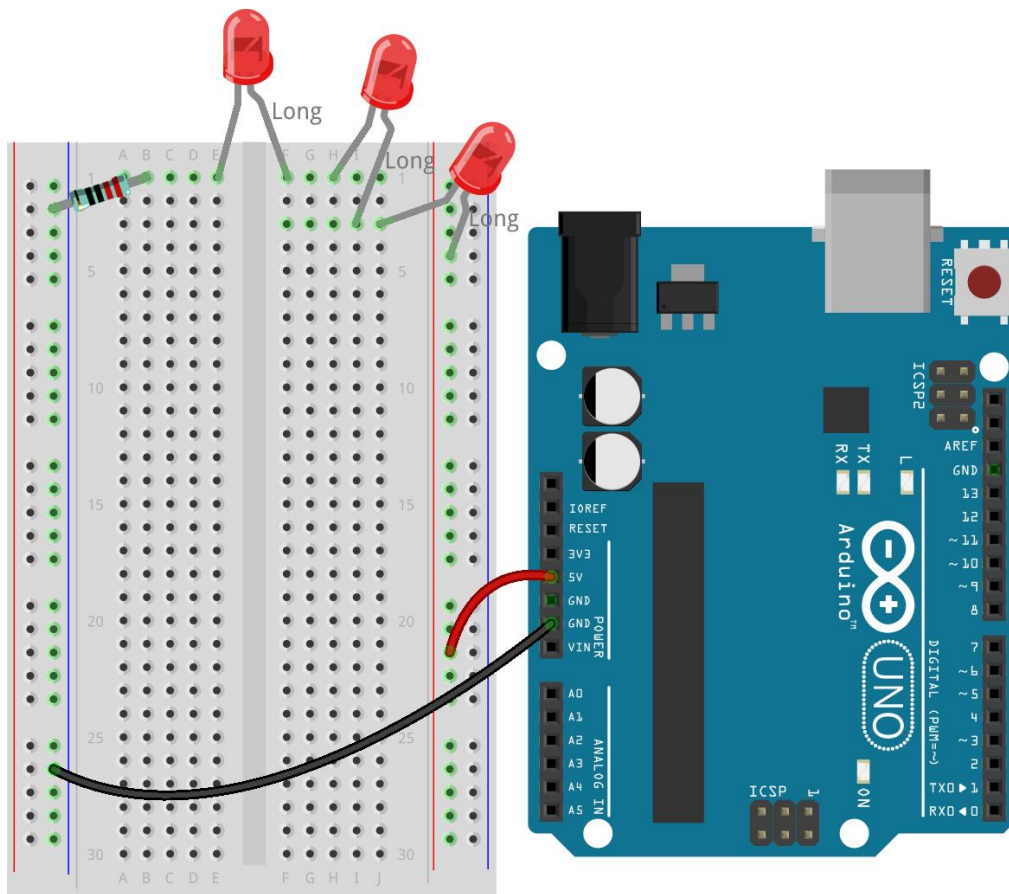
1. Messen Sie die Spannung jedes Bauteils im Serienschaltkreis.

Schaltkreis	Widerstands-Spannung	LED1 Voltage	LED2 Voltage	Total Voltage
2 LEDs	$\approx 1.13$ volts	$\approx 1.92$ volts	$\approx 1.92$ volts	$\approx 4.97$ volts

2. Messen Sie den Strom jedes Bauteils im Serienschaltkreis.

Schaltkreis	LED1-Strom	LED2-Strom
2 LEDs	$\approx 4.43$ milliamps	$\approx 4.43$ milliamps

3. Wenn eine weitere LED zu diesem Schaltkreis hinzugefügt wird, sodass drei LEDs entstehen, wie verändert sich die Helligkeit der LEDs? Warum? Wie ändern sich die Spannungen über den drei LEDs?



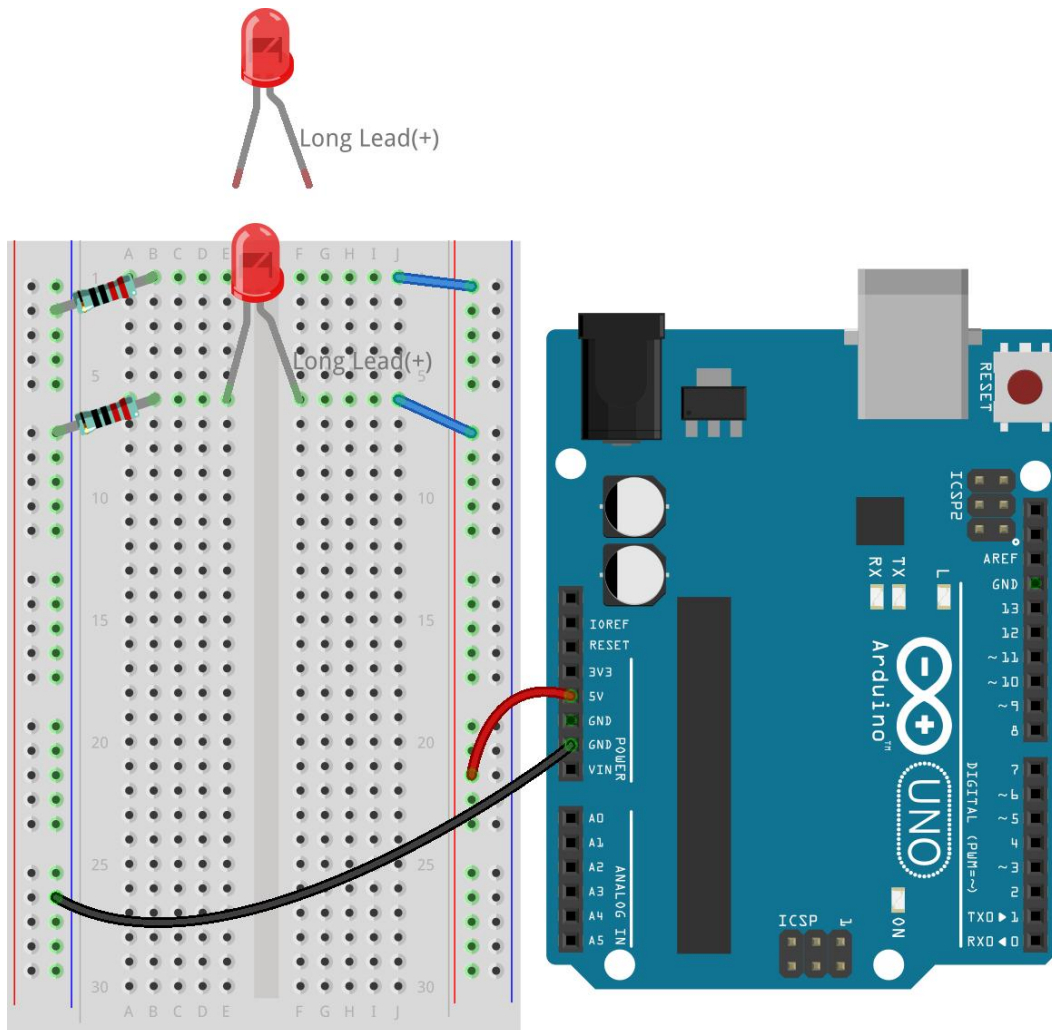
Das Hinzufügen einer weiteren LED zu einem bereits bestehenden seriellen Schaltkreis mit zwei LEDs führt in der Regel zu einer Verringerung der Helligkeit jeder LED. Dies geschieht, weil die Gesamtspannung der Stromquelle auf mehr Komponenten verteilt wird, was zu einem geringeren Spannungsabfall über jeder LED führt als bei nur zwei LEDs. Folglich fließt weniger Strom durch jede LED, was deren Helligkeit verringert.

Was die Spannungen über den drei LEDs betrifft, wird jede LED nun einen kleineren Anteil der gesamten Schaltungsspannung erhalten. Wenn die

Spannung der Stromquelle gleich bleibt, wird diese durch drei geteilt, vorausgesetzt, alle LEDs haben ähnliche elektrische Eigenschaften. Daher wird die Spannung über jeder LED im Schaltkreis ungefähr ein Drittel der Gesamtspannung betragen, die von der Stromquelle bereitgestellt wird.

Beantworte die folgenden Fragen während des Themas **“Eintauchen in Parallelkreise”**.

1. In diesem Parallelkreis, was passiert, wenn eine LED entfernt wird? Warum geschieht das?



In einem Parallelkreis bleibt die Beleuchtung der anderen LEDs im Schaltkreis erhalten, wenn eine LED entfernt wird. Dies geschieht, weil jede LED in einem Parallelkreis ihren eigenen unabhängigen Weg zur Stromquelle hat. Das Entfernen einer LED unterbricht nicht den Stromfluss zu den anderen LEDs, sodass diese unberührt bleiben und normal weiterleuchten. Diese Anordnung ermöglicht es jedem Bauteil in einem Parallelkreis, unabhängig von den anderen zu funktionieren.

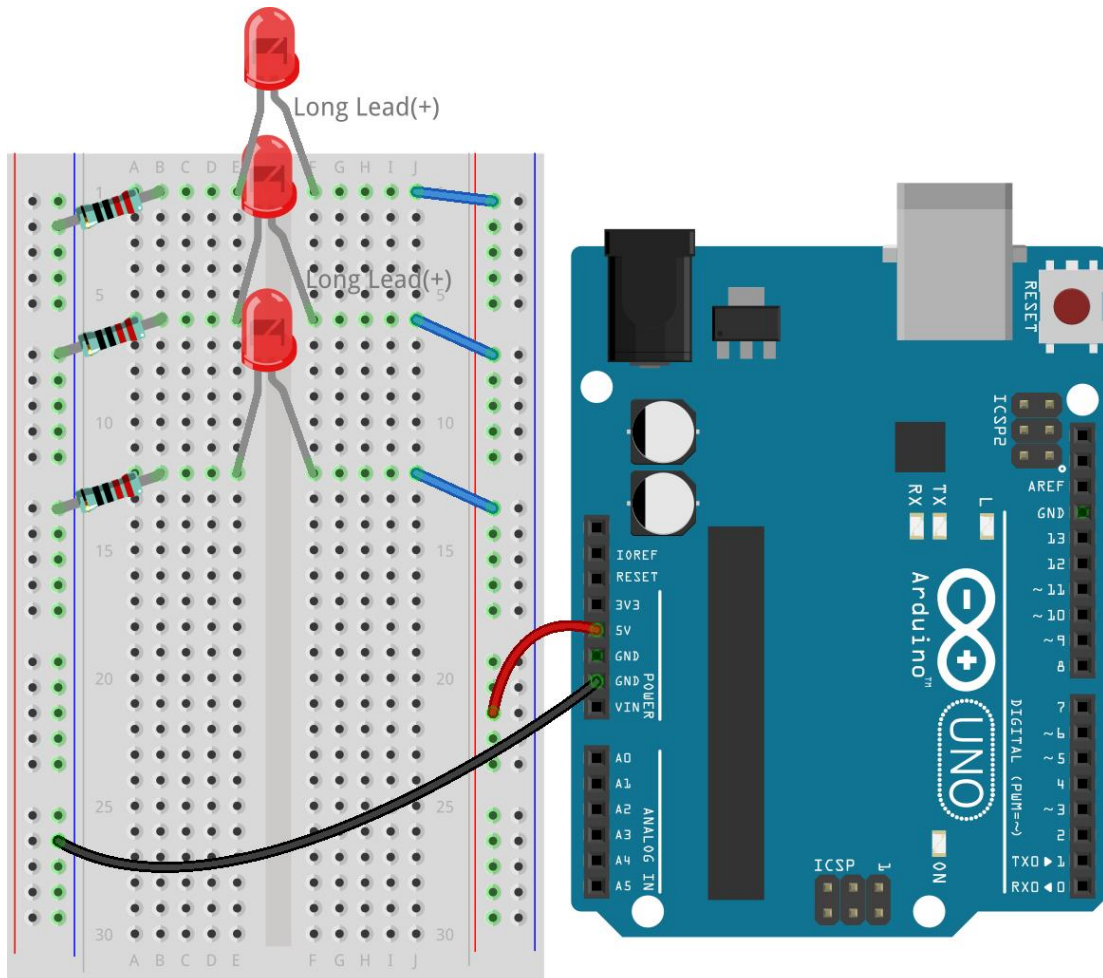
1. Tragen Sie die gemessene Spannung in die Tabelle ein.

Schaltkreis	Spannung Pfad 1	Spannung Pfad 2
2 LEDs	≈5.00 volts	≈5.00 volts

2. Tragen Sie den gemessenen Strom in die Tabelle ein.

Schaltkreis	LED1-Strom	LED2-Strom	Gesamtstrom
2 LEDs	$\approx 12.6$ milliamps	$\approx 12.6$ milliamps	$\approx 25.3$ milliamps

2. Wenn eine weitere LED zu diesem Schaltkreis hinzugefügt wird, wie verändert sich die Helligkeit der LEDs? Warum? Notieren Sie Ihre Antwort in Ihrem Handbuch.



Wenn eine weitere LED zu einem Parallelkreis hinzugefügt wird, bleibt die Helligkeit der bereits vorhandenen LEDs in der Regel unverändert. Dies liegt daran, dass jede LED in einem Parallelkreis ihren eigenen direkten Weg zur Stromquelle hat, sodass die Spannung über jeder LED konstant bleibt, unabhängig davon, wie viele LEDs hinzugefügt werden. Jede LED erhält die volle Spannung, die sie benötigt, um mit der vorgesehenen Helligkeit zu arbeiten. Daher hat das Hinzufügen weiterer LEDs keinen Einfluss auf die Helligkeit der bereits vorhandenen LEDs, vorausgesetzt, die Stromversorgung kann den gesamten Strombedarf des Schaltkreises decken. Notieren Sie dies in Ihrem Handbuch zur späteren Bezugnahme.

## Lektionen 6 : Blink-LED

Vervollständigen Sie die folgende Tabelle während des “**Lebendig-Machens von LEDs**“.

1. Tragen Sie die gemessene Spannung für Pin 3 in die Tabelle ein.

Zustand	Pin 3 Spannung
HIGH	≈4.95 volts
LOW	0.00 volts

Beantworte nach Abschluss der Lektion die folgende Frage.

1. Lade den obigen Code hoch, und du wirst feststellen, dass die LED in einem 3-Sekunden-Intervall blinkt. Wenn du möchtest, dass sie nur einmal ein- und ausschaltet, was solltest du tun?

Du kannst die Befehle, die die LED ein- und ausschalten, von der `loop()`-Funktion in die `setup()`-Funktion verschieben. Die `setup()`-Funktion wird nur einmal ausgeführt, wenn das Programm startet, sodass diese Änderung dazu führt, dass die LED einmal aufleuchtet und wieder ausgeht. So kannst du deinen Code anpassen:

```
void setup() {  
  // Setup code here, to run once:  
  pinMode(3, OUTPUT); // set pin 3 as output  
  
  digitalWrite(3, HIGH); // Light up the LED on pin 3  
  delay(3000);           // Wait for 3 seconds  
  digitalWrite(3, LOW);  // Switch off the LED on pin 3  
}  
  
void loop() {  
  // Main code here, to run repeatedly:  
}
```



## Lektionen 7: Lass uns Ampeln bauen!

Beantworte die folgende Frage während des **“Schreibens von Pseudocode für eine Ampel”**.

Überlege, was geschehen muss, damit dein Schaltkreis wie eine Ampel funktioniert. Schreibe im vorgesehenen Bereich deines Protokolls den Pseudocode auf, der beschreibt, wie deine Ampel funktionieren wird. Verwende einfache Sprache.

Um eine Ampel mit einem Arduino zu simulieren, benötigst du eine Konfiguration mit drei LEDs (rot, gelb und grün) und eine Abfolge, die die Beleuchtung steuert, um das Verhalten von echten Ampeln nachzuahmen. Hier ist eine einfache Pseudocode-Skizze, die du in deinem Protokoll festhalten kannst, um zu beschreiben, wie dieser Ampel-Schaltkreis funktionieren könnte:

Setup:

```
Define pins for the red, yellow, and green LEDs.  
Set all these pins as outputs.
```

Main Loop:

```
Turn on the red LED for 5 seconds.  
Turn off the red LED.  
Turn on the yellow LED for 2 seconds.  
Turn off the yellow LED.  
Turn on the green LED for 5 seconds.  
Turn off the green LED.  
Repeat the cycle.
```

Beantworte nach Abschluss der Lektion die folgende Frage:

Schau dir die Kreuzungen in deiner Umgebung an. Wie viele Verkehrsampeln gibt es normalerweise? Wie arbeiten sie miteinander zusammen?

In städtischen Gebieten verfügen Kreuzungen häufig über Verkehrsampeln, um den Fluss von Fahrzeugen und Fußgängern effizient zu steuern. Die Anzahl der Verkehrsampeln an einer Kreuzung kann je nach Größe und Komplexität stark variieren. Eine einfache vierstreifige Kreuzung hat normalerweise mindestens vier Verkehrsampeln, eine in jede Fahrtrichtung. Komplexere Kreuzungen können zusätzliche Ampeln für Abbiegespuren, Fußgängerüberwege und andere Verkehrsmanagementbedürfnisse aufweisen.

## Lektionen 8 : Verkehrsampel mit Fußgängertaste

Beantworte nach dem Abschluss von **“Schaltung aufbauen”** die folgende Frage

---

1. Ihre Verkehrsampel ist eine Mischung aus Serien- und Parallelschaltungen. Diskutieren Sie, welche Teile Ihrer Schaltung in Serie geschaltet sind und warum. Erklären Sie dann, welche Teile parallel geschaltet sind und warum.

In der Schaltung sind der Taster und sein 10K-Pull-Down-Widerstand in Reihe geschaltet. Diese Anordnung stellt sicher, dass beim Drücken des Tasters der Zustand von Pin 8 korrekt geändert wird, indem er im unbetätigten Zustand direkt mit dem Ground verbunden wird, um zu verhindern, dass der Eingang „floating“ ist.

Die drei LEDs, die an den Pins 3, 4 und 5 angeschlossen sind, sind parallel zueinander geschaltet. Jede LED funktioniert unabhängig, da sie mit separaten Steuerpins verbunden sind und eine gemeinsame Stromversorgung nutzen. Diese Anordnung ermöglicht es jeder LED, zu arbeiten, ohne die anderen zu beeinflussen, was für ein Verkehrssignalsystem entscheidend ist.

Füllen Sie diese Tabelle während der **“Code-Erstellung”** aus.

---

1. Tragen Sie in die Tabelle die gemessene Spannung an Pin 8 ein, wenn der Taster gedrückt und nicht gedrückt ist. Füllen Sie dann die entsprechenden Zustände für High und Low aus.

Tasterzustand	Pin 8 Spannung	Pin 8 Zustand
Loslassen	0.00 volts	LOW
Drücken	≈4.97 volts	HIGH

Beantworten Sie die folgende Frage nach Abschluss dieser Lektion.

1. Während der Tests werden Sie feststellen, dass die grüne LED nur blinkt, solange der Fußgängerknopf gedrückt bleibt, aber Fußgänger können die Straße nicht überqueren, während sie den Knopf kontinuierlich drücken. Wie können Sie den Code ändern, damit die grüne LED lange genug leuchtet, um ein sicheres Überqueren zu ermöglichen, ohne dass der Knopf kontinuierlich gedrückt werden muss? Bitte notieren Sie die Lösung in Pseudo-Code in Ihrem Handbuch.

Um sicherzustellen, dass die grüne LED für Fußgänger leuchtet, ohne dass der Knopf kontinuierlich gedrückt werden muss, und um danach den normalen Ampelzyklus fortzusetzen, können Sie Ihren Pseudo-Code anpassen, um den Knopfdruck zu überprüfen und den Betriebszustand basierend auf diesem Druck zu ändern. Hier ist eine optimierte und klarere Version des Pseudo-Codes, die diese Änderungen widerspiegelt:

Setup:

```
Define pins for red, yellow, and green LEDs as output  
Define the button pin as input
```

Main Loop:

```
Check if the button is pressed
```

```
If button is pressed:
```

```
    Turn off all LEDs
```

```
    Turn on green LED for pedestrians
```

```
    Delay 10 seconds
```

```
Else:
```

```
    Execute normal traffic light cycle:
```

```
        Turn on green LED (for vehicles), turn off other LEDs
```

```
        Delay 10 seconds
```

```
        Turn on yellow LED, turn off other LEDs
```

```
        Delay 3 seconds
```

```
        Turn on red LED, turn off other LEDs
```

```
        Delay 10 seconds
```

## Lektionen 9: Dimmbare Schreibtischlampe

Füllen Sie diese Tabelle während „Schaltung aufbauen“ aus.

1. Drehen Sie den Potentiometer im Uhrzeigersinn von Position 1 nach 3 und messen Sie den Widerstand an jedem Punkt. Halten Sie die Ergebnisse in der Tabelle fest.

Messpunkt	Widerstand (kiloohm)
1	1.52
2	5.48
3	9.01

2. Wie denkst du, würde sich die Spannung an A0 ändern, wenn der Potentiometer im Uhrzeigersinn und gegen den Uhrzeigersinn gedreht wird?

Man kann den Potentiometer als zwei in Reihe geschaltete Widerstände im Schaltkreis betrachten. Laut den Widerstandsmessungen steigt der Widerstand zwischen A0 und GND, wenn der Potentiometer im Uhrzeigersinn gedreht wird. Da der Strom in einem Serienschaltkreis konstant bleibt, führt eine Erhöhung des Widerstands gemäß dem Ohmschen Gesetz (Spannung = Strom  $\times$  Widerstand) zu einem Anstieg der Spannung an A0. Daher erhöht das Drehen des Potentiometers im Uhrzeigersinn die Spannung an A0, während das Drehen gegen den Uhrzeigersinn diese verringert, da der Widerstand abnimmt.

Beantworte die folgende Frage nach Abschluss dieser Lektion.

1. Wenn du die LED an einen anderen Pin, wie zum Beispiel Pin 8, anschließt und den Potentiometer drehst, wird sich die Helligkeit der LED dann immer noch ändern? Warum oder warum nicht?

Wenn du die LED an Pin 8 eines Arduino UNO anschließt und den Potentiometer drehst, wird sich die Helligkeit der LED nicht ändern. Das liegt daran, dass Pin 8 kein PWM (Pulsweitenmodulation) unterstützt, was für die Anpassung der Helligkeitsstufen mit der Funktion `analogWrite()` erforderlich ist. Beim Arduino UNO sind die Pins, die PWM unterstützen und somit zur Steuerung der Helligkeit einer LED über `analogWrite()` verwendet werden können, die Pins 3, 5, 6, 9, 10 und 11.

## Lektion 10: Morsecode

Beantworte die folgende Frage während des **“Aufbaus des Schaltkreises”**.

---

1. Was passiert, wenn du die Kathode eines aktiven Buzzers direkt mit GND und die Anode mit 5V verbindest? Warum?

Wenn du die Kathode eines aktiven Buzzers direkt mit GND und die Anode mit 5V verbindest, wird der Buzzer einen kontinuierlichen Ton ausgeben. Dies geschieht, weil der interne Oszillator im Buzzer durch die 5V-Stromversorgung aktiviert wird, wodurch er Geräusche erzeugt, bis der Stromkreis unterbrochen wird.

Beantworte nach Abschluss der Lektion die folgende Frage.

---

1. Verwende die bereitgestellte Morsecode-Tabelle, um einen Code zu schreiben, der die Nachricht „Hallo“ sendet.

Im Morsecode wird „Hallo“ wie folgt codiert, basierend auf den Zeichen:

- H: ....
- E: .
- L: .-..
- L: .-..
- O: ---

Zusammengefasst lautet der Morsecode für „Hallo“:

.... . -.. -.. ---

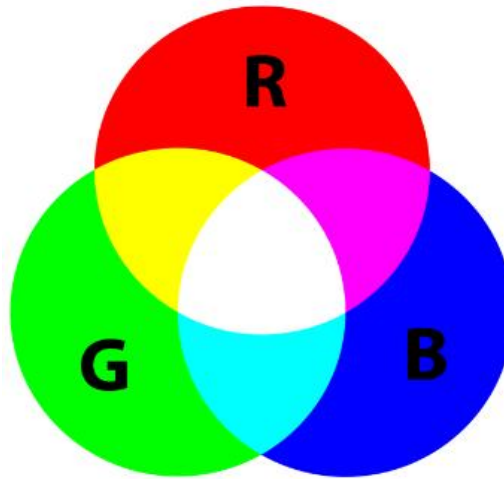
In der praktischen Kommunikation gibt es in der Regel eine längere Pause zwischen den Wörtern, um diese deutlich zu unterscheiden. Da „Hallo“ jedoch ein einzelnes Wort ist, bleibt der Code kontinuierlich, wobei nur Leerzeichen die einzelnen Buchstaben voneinander trennen.



# Lektionen 11: Die Farben des Regenbogens

Füllen Sie diese Tabelle während „Codeerstellung - RGB-LED zum Leuchten bringen“ aus

1. Wenn Sie andere Farben wünschen, was sollten Sie tun? Sehen Sie sich das untenstehende Diagramm an und notieren Sie Ihre Ideen in Ihrem Handbuch.



Farbe	Rot-Pin	Grün-Pin	Blau-Pin
Rot	HIGH	LOW	LOW
Grün	LOW	HIGH	LOW
Blau	LOW	LOW	HIGH
Gelb	HIGH	HIGH	LOW
Rosa	HIGH	LOW	HIGH
Cyan	LOW	HIGH	HIGH
Weiß	HIGH	HIGH	HIGH

Füllen Sie diese Tabelle während **“Code-Erstellung - Farben anzeigen”** aus.

1. Jetzt können Sie die Werte der Pins 9, 10 und 11 separat anpassen und die beobachteten Farben in Ihrem Handbuch festhalten.

Rot-Pin	Grün-Pin	Blau-Pin	Farbe
0	128	128	<i>Dunkelblau</i>
128	0	255	<i>Lila</i>
128	128	255	<i>Hellblau</i>
255	128	0	<i>Orange</i>

Füllen Sie diese Tabelle während „Codeerstellung - Parametrisierte Funktionen“ aus

1. Wählen Sie einige Ihrer Lieblingsfarben aus und füllen Sie die Tabelle mit ihren RGB-Werten aus.

Farbe	Rot	Grün	Blau

## Lektion 12: Sirenenton

Beantworte die folgende Frage während **“Schalte den Schaltkreis zusammen“**

---

1. Was passiert, wenn du die Kathode eines passiven Buzzers direkt mit GND und die Anode mit 5V verbindest? Warum?

Wenn du die Kathode eines passiven Buzzers direkt mit GND und die Anode mit 5V verbindest, wird der passive Buzzer, im Gegensatz zu einem aktiven Buzzer, von selbst keinen Ton erzeugen, da er keinen integrierten Oszillator hat. Ein passiver Buzzer benötigt ein externes Signal, um Geräusche zu erzeugen.

Typischerweise musst du ihn mit einer Rechteckwelle (oszillierende Spannung) bei der gewünschten Frequenz ansteuern, um hörbare Töne zu erzeugen.

Beantworte die folgenden Fragen während **“Code-Erstellung - Lasse den passiven Buzzer ertönen“**

---

1. Wenn du die Code- und Schaltkreis-Pins auf 7 oder 8 wechselst, die keine PWM-Pins sind, wird der Buzzer dann trotzdem einen Ton erzeugen? Du kannst es testen und deine Antwort im Handbuch aufschreiben.

Obwohl Pin 8 kein PWM-Pin ist, kann die Funktion `tone()` dennoch eine präzise Rechteckwelle darauf erzeugen, wodurch ein passiver Buzzer zum Erzeugen von Tönen angesteuert wird. Diese Flexibilität ermöglicht es dir, jeden digitalen Pin für die Tonausgabe zu nutzen, ohne auf PWM-fähige Pins beschränkt zu sein. Wenn du die Funktion `tone(pin, frequency)` aufrufst, konfiguriert Arduino einen Timer, um den Pin-Zustand (von HIGH nach LOW und zurück zu HIGH) mit der angegebenen Frequenz umzuschalten und so eine Rechteckwelle zu erzeugen. Diese Rechteckwelle treibt den passiven Buzzer an, sodass er den Ton in der Frequenz der erzeugten Welle ausgibt.

1. Um zu erkunden, wie Frequenz und Dauer in `tone(pin, frequency, duration)` den Klang des Buzzers beeinflussen, modifiziere bitte den Code unter zwei Bedingungen und halte die beobachteten Phänomene in deinem Handbuch fest:

- Halte die Frequenz bei 1000 und erhöhe schrittweise die Dauer von 100 über 500 bis 1000. Wie verändert sich der Klang des Buzzers und warum?
  - 100 ms Dauer: Der Klang ist ein kurzer Piepton.
  - 500 ms Dauer: Der Klang ist ein längerer Piepton, der deutlich hörbar ist und eine halbe Sekunde anhält.
  - 1000 ms Dauer: Der Klang ist noch länger und dauert eine volle Sekunde.

Wenn du die Dauer erhöhst, hält der Ton des Buzzers länger an. Die Tonhöhe oder Frequenz bleibt konstant (da sie auf 1000 Hz eingestellt ist), was bedeutet, dass sich die "Note" des Tons nicht ändert, aber die Hördauer zunimmt. Dies ist nützlich, um unterschiedliche Alarmdauern zu signalisieren, bei denen die Dringlichkeit oder Art des Alarms durch die Länge des Tons unterschieden werden kann.

- Halte die Dauer bei 100 und erhöhe schrittweise die Frequenz von 1000 über 2000 bis 5000. Wie verändert sich der Klang des Buzzers und warum?
  - 1000 Hz Frequenz: Der Klang ist ein mittelhoher Piepton.
  - 2000 Hz Frequenz: Der Klang hat eine höhere Tonhöhe im Vergleich zu 1000 Hz.
  - 5000 Hz Frequenz: Der Klang ist viel höher, wahrscheinlich als schärfer wahrgenommen und könnte in unmittelbarer Nähe unangenehm sein.

Eine Erhöhung der Frequenz bei konstanter Dauer führt zu einer Veränderung der Tonhöhe des Sounds. Höhere Frequenzen erzeugen höhere Töne. Dieses Prinzip ist nützlich, um verschiedene Arten von Benachrichtigungen oder Signalen je nach Dringlichkeit oder Wichtigkeit zu unterscheiden, wobei höhere Töne oft für dringlichere Alarme verwendet werden.

## Lektion 13: Joystick-LED-Navigator

Beantworte die folgende Frage während **“Code-Erstellung - Lese vom Joystick-Modul“**

---

1. Die X- und Y-Achsen des Joystick-Moduls geben analoge Werte zurück, während der SW-Pin einen digitalen Wert liefert. In vorherigen Schritten haben wir diese Werte bereits im Serial Monitor gesehen.

Bitte fasse die Unterschiede zwischen digitalen und analogen Werten in der Arduino-Programmierung zusammen.

In der Arduino-Programmierung:

- **Analoge Werte:** Kontinuierliche Werte im Bereich von 0 bis 1023, die 0V bis 5V entsprechen und vom Analog-Digital-Wandler (ADC) des Arduino erfasst werden. Diese Werte liefern detaillierte Informationen über die Signalstärke, wie bei den X- und Y-Achsen des Joysticks zu sehen ist.
- **Digitale Werte:** Diskrete Werte von 0 oder 1, die niedrige (0V) oder hohe (5V) Zustände repräsentieren. Der SW-Pin eines Joysticks verwendet ein digitales Signal, um anzuzeigen, ob er gedrückt ist oder nicht, was ideal für binäre Ein/Aus-Entscheidungen ist.

Analoge Eingänge bieten detaillierte Informationen über die Signalintensität, während digitale Eingänge zur einfachen Erkennung von Anwesenheit oder Abwesenheit verwendet werden.

Beantworte die folgende Frage während **“Code-Erstellung - Steuerung von LEDs basierend auf Joystick-Bewegungen“**

---

1. Im letzten Code haben wir die entsprechenden LEDs basierend auf den X- und Y-Werten des Joysticks gesteuert. Wie würdest du den Code anpassen, um die Helligkeit dieser LEDs während des Leuchtens zu regulieren?

Um die Helligkeit der LEDs basierend auf den X- und Y-Werten des Joysticks zu steuern, solltest du PWM (Pulsweitenmodulation) über die Funktion `analogWrite()` anstelle von `digitalWrite()` verwenden. Dieser Ansatz ermöglicht es dir, unterschiedliche Helligkeitsstufen der LEDs entsprechend der Position des Joysticks einzustellen, anstatt die LEDs nur ein- oder auszuschalten.

Hier ist ein verbessertes Code-Snippet, um dieses Konzept zu demonstrieren: So könntest du den Code anpassen:



```

void loop() {
  // Read the joystick values
  int xValue = analogRead(xPin);
  int yValue = analogRead(yPin);
  int swValue = digitalRead(swPin);

  // First, turn off all LEDs
  analogWrite(ledLeft, 0);
  analogWrite(ledRight, 0);
  analogWrite(ledUp, 0);
  analogWrite(ledDown, 0);

  // Check joystick positions and set LEDs accordingly
  if (yValue < 412) {
    // Joystick up
    analogWrite(ledUp, yValue / 4);
  } else if (yValue > 612) {
    // Joystick down
    analogWrite(ledDown, yValue / 4);
  } else if (xValue < 412) {
    // Joystick left
    analogWrite(ledLeft, xValue / 4);
  } else if (xValue > 612) {
    // Joystick right
    analogWrite(ledRight, xValue / 4);
  }
  // Add a small delay to stabilize readings
  delay(100);
}

```

1. Was ist der Unterschied im Verhalten der LED, die an Pin 8 angeschlossen ist, im Vergleich zu den LEDs an anderen Pins, und warum?

Die an Pin 8 des Arduino Uno R3 angeschlossene LED wechselt lediglich zwischen ein- und ausgeschaltet, ohne eine Helligkeitsvariation.

Dies liegt daran, dass Pin 8 PWM (Pulsweitenmodulation) nicht unterstützt. Beim Arduino Uno R3 ist PWM nur an den Pins 3, 5, 6, 9, 10 und 11 verfügbar. Das Fehlen der PWM-Funktionalität an Pin 8 bedeutet, dass die `analogWrite()`-Funktion, die die Helligkeit der LED durch variierende Spannungspegel anpasst, an diesem Pin nicht verwendet werden kann. Daher können Sie nur `digitalWrite()` verwenden, um die LED vollständig ein- oder auszuschalten, ohne Zwischenzustände.

# Lektionen 14: Dinosaurier-Spiel spielen

Beantworte die folgende Frage während „2. Servomotor vorbereiten“.

1. Wenn der Servomotor an Pin 8 oder einem anderen Nicht-PWM-Pin eines Arduino angeschlossen ist, wird er dann trotzdem korrekt funktionieren? Warum oder warum nicht?

Du kannst dies zuerst testen, bevor du antwortest.

Wenn du einen Servomotor an einen Pin eines Arduino anschließt, wird in der Regel empfohlen, einen PWM-Pin (Pulsweitenmodulation) zu verwenden.

Die Servo-Bibliothek des Arduino ist jedoch so konzipiert, dass sie die Steuerung von Servos über PWM-Pins hinaus ermöglicht, indem sie softwarebasierte Emulation verwendet, um die erforderlichen Signale zu erzeugen.

Dieser softwarebasierte Ansatz bedeutet, dass die Steuerungspulse für den Servomotor intern von der Bibliothek verwaltet werden und nicht von den hardwareseitigen PWM-Funktionen des Mikrocontrollers abhängen. Dadurch ermöglicht die Servo-Bibliothek die Verwendung jedes digitalen Pins – einschließlich Nicht-PWM-Pins wie Pin 8 – um einen Servomotor effektiv zu steuern.

Beantworte die folgende Frage während „3. Bereite den Fotowiderstand vor“.

1. Lese den Widerstandswert bei den aktuellen Umgebungslichtverhältnissen und trage ihn in die folgende Tabelle ein.

Umgebung	Widerstand (Kilohm)
Normales Licht	≈5.48
Helles Licht	≈0.16
Dunkelheit	≈1954

## Lektion 15: Kalte oder Warme Farben

Füllen Sie diese Tabelle während der „Code-Erstellung“ aus.

1. Öffnen Sie Paint oder ein beliebiges Farbauswahl-Tool, suchen Sie die Farben, die Sie als die wärmsten und kühlfsten empfinden, und notieren Sie deren RGB-Werte in Ihrem Handbuch.

Farbe	Rot	Grün	Blau
Warme Farbe	246	52	8
Kühle Farbe	100	150	255

Beantworte nach Abschluss der Lektion die folgende Frage.

1. Bedenke, dass die „unteren Grenzen“ eines Bereichs größer oder kleiner als die „oberen Grenzen“ sein können, sodass die `map()` Funktion verwendet werden kann, um einen Zahlenbereich umzukehren, zum Beispiel:

```
y = map(x, 1, 50, 50, 1);
```

Die Funktion verarbeitet auch negative Zahlen gut, sodass dieses Beispiel ebenfalls gültig ist und gut funktioniert.

```
y = map(x, 1, 50, 50, -100);
```

Für `y = map(x, 1, 50, 50, -100);`, wenn `x` 20 entspricht, was sollte `y` sein? Verwenden Sie die folgende Formel zur Berechnung.

te it.

$$\frac{\text{Value} - \text{From Low}}{\text{From High} - \text{From Low}} = \frac{\text{Y} - \text{To Low}}{\text{To High} - \text{To Low}}$$

$$\text{Y} = \frac{\text{Value} - \text{From Low}}{\text{From High} - \text{From Low}} \times (\text{To High} - \text{To Low}) + \text{To Low}$$

Für `x=20` mit der Abbildungsformel `y = map(x, 1, 50, 50, -100);` wäre der Wert von `y` ungefähr `-8.16`.

## Lektion 16: Sommerventilator

Füllen Sie diese Tabelle während „Schaltung aufbauen“ aus

1. Schließen Sie 1,2EN an 5V an und verwenden Sie ein Multimeter, um die Spannung an Pin 3 (1Y) zu messen, wenn Pin 2 (1A) mit 5V und wenn er mit GND verbunden ist. Halten Sie die Messwerte in der Tabelle fest.

1,2EN	1A	1Y
5V	5V	≈5.04V
5V	0V	≈0V

Nachdem Sie die Lektion abgeschlossen haben, beantworten Sie bitte die folgende Frage.

1. Wie sollte der Code geändert werden, wenn Sie auch die Drehrichtung des Motors steuern möchten?

- Ändern Sie die Funktion `motorRotate()`, um die Steuerung der Drehrichtung durch die Verwendung beider Motorsteuerepins einzuschließen.
- Fügen Sie der Funktion `motorRotate()` einen Parameter hinzu, um die Drehrichtung anzugeben.
- Passen Sie die Aufrufe von `motorRotate()` in der Hauptschleife an, um die Drehrichtung basierend auf den Tastendruckern zu steuern.

```
void loop() {
  if (digitalRead(button1) == LOW) {
    motorRotate(0, 0); // Turn off the motor
  } else if (digitalRead(button2) == LOW) {
    motorRotate(150, -1); // Motor runs backward at low speed
  } else if (digitalRead(button3) == LOW) {
    motorRotate(200, 1); // Motor runs forward at medium speed
  } else if (digitalRead(button4) == LOW) {
    motorRotate(250, 1); // Motor runs forward at high speed
  }
}

void motorRotate(int speed, int direction) {
  if (direction >= 0) {
    analogWrite(motor1A, speed);
    analogWrite(motor2A, 0);
  } else {
    analogWrite(motor1A, 0);
    analogWrite(motor2A, speed);
  }
}
```

# Lektion17: Erforschung des I2C LCD1602

## Displays

Beantworten Sie nach Abschluss der Lektion die folgende Frage.

1. Wenn Sie "Lass uns zählen" ab der fünften Spalte der ersten Zeile auf dem I2C LCD1602 anzeigen möchten, wie sollten Sie den Code anpassen, und welchen visuellen Effekt würden Sie beobachten?

Um "Lass uns zählen" ab der siebten Spalte der ersten Zeile auf dem I2C LCD1602 anzuzeigen, müssen Sie die Funktion `lcd.setCursor()` anpassen, indem Sie den Cursor auf die sechste Spalte setzen (da die Spaltenindizes bei 0 beginnen). So können Sie den Code ändern und was Sie dabei beobachten könnten:

```
void loop() {  
    lcd.setCursor(6, 0);           // Sets cursor to the fifth column (index  
6) of the first row.  
    lcd.print("Let's count");      // Displays "Let's count".  
    lcd.setCursor(0, 1);          // Moves cursor to the first column of  
the second line.  
    lcd.print("Count: ");          // Displays "Count: ".  
    lcd.print(count);              // Prints current count next to "Count:  
".  
    delay(1000);                  // Pauses for one second.  
    count++;                      // Increments counter.  
    lcd.clear();                  // Clears the display for the next  
iteration.  
}
```

Wenn Sie in der siebten Spalte beginnen, wird der Text "Lass uns zählen" weiter rechts auf dem Display angezeigt. Der Text "Lass uns zählen" besteht aus 11 Zeichen, sodass er bis zur siebzehnten Spalte reicht. Da das LCD1602 jedoch nur 16 Spalten hat, wird das letzte Zeichen nicht auf dem Display sichtbar sein. Diese Truncation führt dazu, dass nur "Lass uns zäh" sichtbar ist, wobei das letzte "l" am Rand des Displays abgeschnitten wird.

Daher ist eine sorgfältige Planung des Layouts des Displays entscheidend, um sicherzustellen, dass der gesamte gewünschte Text ordnungsgemäß auf dem LCD-Bildschirm angezeigt wird.



## Lektion18: Ein/Aus Schreibtischlampe

Füllen Sie diese Tabelle während „Schaltkreis erstellen“ aus.

1. Jetzt verwenden Sie ein Multimeter, um die Durchgängigkeit zwischen COM und NC zu messen, um das Funktionsprinzip des Relaismoduls zu überprüfen.

Zustand	NO oder NC mit dem COM-Anschluss verbunden?
Standard	NC
S-Pin Hoch	NO

Beantworten Sie die folgenden Fragen nach Abschluss dieser Lektion.

1. Was würde passieren, wenn Sie den digitalen Pin 7 nur auf INPUT setzen? Warum?

```
void setup() {
  pinMode(9, OUTPUT); // Set pin 9 as output
  pinMode(7, INPUT);  // Set pin 7 as input with an internal pull-up
resistor
  Serial.begin(9600); // Serial communication setup at 9600 baud
}
```

Die Konfiguration des digitalen Pins 7 im INPUT-Modus in Ihrem Arduino-Sketch, anstelle von INPUT\_PULLUP, kann zu einer potenziellen Instabilität des Signals führen, das von diesem Pin gelesen wird. Wenn ein Pin nur als INPUT konfiguriert ist und nicht über externe Schaltungen mit einer definitiven Hoch- oder Niederspannung verbunden ist, befindet er sich im sogenannten "schwebenden" Zustand. Ein schwebender Pin ist weder in einem stabilen Hoch- noch in einem stabilen Niedrigzustand; sein Zustand kann je nach elektrischen Störungen oder Umwelteinflüssen schwanken. Diese Schwankungen können zu unvorhersehbaren Werten führen, wenn Sie den Zustand des Pins über digitale Eingabefunktionen lesen, was dazu

führt, dass der Mikrocontroller fehlerhafte oder inkonsistente Daten erhält.

2. Welche Anpassungen müssten an der Schaltung vorgenommen werden, wenn Pin 7 nur auf `INPUT` eingestellt ist?

Wenn Pin 7 Ihres Arduino auf `INPUT`-Modus eingestellt ist und Sie stabile sowie vorhersagbare Werte sicherstellen möchten, sollten Sie einen externen Pull-up-Widerstand in die Schaltung einfügen. Dies erfordert, einen 10kΩ-Widerstand zwischen Pin 7 und der 5V-Stromversorgung des Arduino zu verbinden. Der Pull-up-Widerstand stellt sicher, dass der Eingabepin im hohen Zustand (Logiklevel 1) ist, wenn kein anderes Eingangssignal vorhanden ist.

## Lektion19: Intelligente Mülltonne

Beantworte die folgende Frage während “Code-Erstellung - Distanz messen”.

1. Wie solltest du den Code ändern, um die vom Gerät gemessene Distanz genauer auf Dezimalstellen anzuzeigen?

Um die Genauigkeit der Distanzmessungen in deinem Arduino-Code zu verbessern und Dezimalstellen einzuschließen, musst du die Berechnung in der Funktion `measureDistance()` anpassen. Derzeit wird die Distanz mit ganzzahliger Mathematik berechnet, wodurch Dezimalstellen abgeschnitten werden. Um Dezimalgenauigkeit zu erreichen, kannst du die Berechnung so ändern, dass sie Fließkommaarithmetik anstelle von Ganzzahlarithmetik verwendet.

Hier ist, wie du den Code anpassen kannst:

- Ändere den Datentyp von `distance` von `long` auf `float`, um Dezimalwerte zu ermöglichen.
- Passe die Formel an, um sicherzustellen, dass sie Gleitkommaberechnungen durchführt.

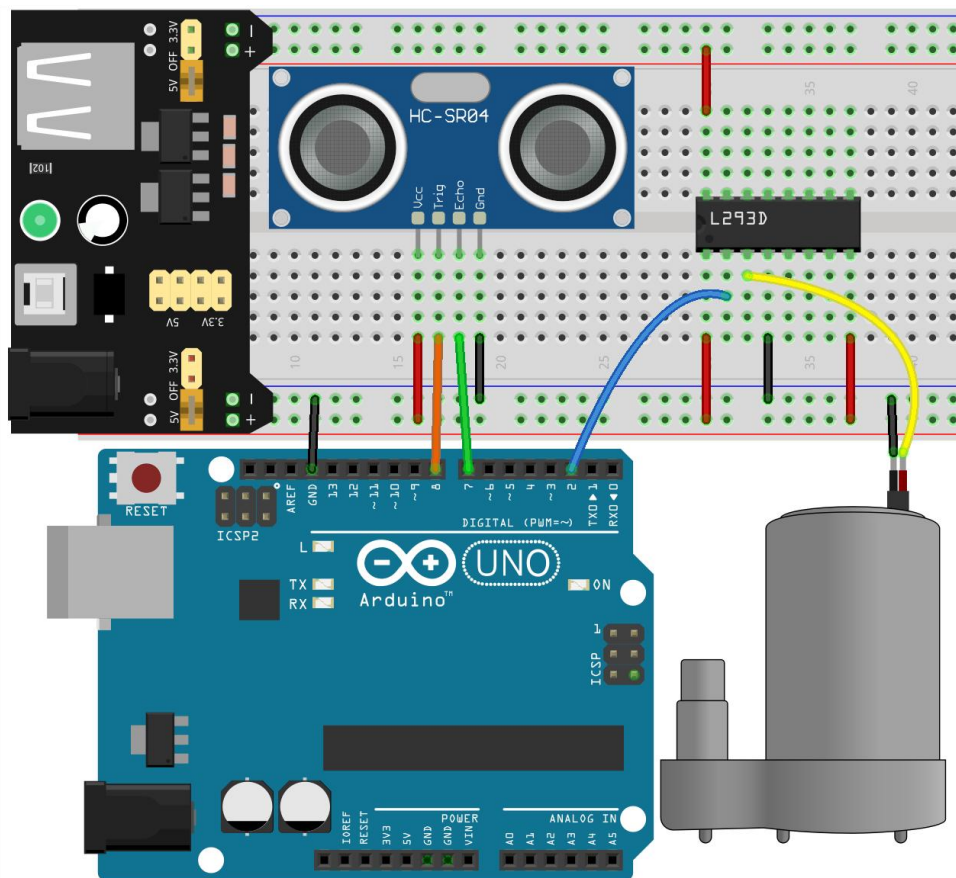
Hier ist der modifizierte Teil der `measureDistance()`-Funktion:

```
float measureDistance() {  
    digitalWrite(TRIGGER_PIN, LOW); // Ensure Trig pin is low before  
a pulse  
    delayMicroseconds(2);  
    digitalWrite(TRIGGER_PIN, HIGH); // Send a high pulse  
    delayMicroseconds(10);           // Pulse duration of 10  
microseconds  
    digitalWrite(TRIGGER_PIN, LOW); // End the high pulse  
  
    long duration = pulseIn(ECHO_PIN, HIGH); // Measure the duration  
of high level on Echo pin  
    float distance = duration * 0.034 / 2.0; // Calculate the distance  
(in cm) with decimal precision  
    return distance;  
}
```

## Lektion 20: Automatischer Seifenspender

Beantworten Sie die folgende Frage während „Codeerstellung - Die Wasserpumpe zum Laufen bringen“

1. In diesem Projekt haben Sie die Wasserpumpe mit einem bestimmten Treiber und einer speziellen Konfiguration verbunden. Was glauben Sie, würde passieren, wenn Sie die Anschlüsse der Pumpe umkehren? Würde die Pumpe rückwärts arbeiten, aufhören zu funktionieren oder etwas anderes? Probieren Sie es aus und reflektieren Sie über das Ergebnis.



Sie werden feststellen, dass die Pumpe in Ihrem Projekt weiterhin effektiv arbeiten und Wasser pumpen würde, selbst wenn Sie die Anschlüsse umkehren. Sie würde nicht rückwärts arbeiten, aufhören zu funktionieren oder Schaden verursachen.

Das Umkehren der Anschlüsse verändert die Funktion der Pumpe nicht, da sie über eine bidirektionale Fähigkeit verfügt. Die Pumpe wird weiterhin Wasser von ihrem Eingang zu ihrem Ausgang bewegen, wie vorgesehen.

## Lektionen 21: Temperaturalarm

Füllen Sie diese Tabelle während des "Aufbaus des Schaltkreises" aus.

1. Messen Sie den Widerstandswert bei den unterschiedlichen Temperaturen und tragen Sie ihn in die folgende Tabelle ein.

Umgebung	Widerstand (Kiloohm)
Aktuelle Temperatur	9.37
Höhere Temperatur	6.10
Niedrigere Temperatur	12.49

Beantworte nach Abschluss der Lektion die folgende Frage

1. Im Code werden die Temperaturen in Kelvin und Celsius berechnet. Wenn du auch die Temperatur in Fahrenheit wissen möchtest, was solltest du tun?

Dies ist die gängige Methode zur Umrechnung von Celsius in Fahrenheit und ermöglicht es dir, die Temperatur in Fahrenheit basierend auf dem Celsius-Wert zu erhalten, den du bereits aus deinen Berechnungen hast.

$$F = C * 1.8 + 32$$

2. Kannst du dir andere Situationen oder Orte vorstellen, an denen ein Temperaturüberwachungssystem wie das, das wir heute gebaut haben, nützlich sein könnte?

Temperaturüberwachungssysteme finden in alltäglichen Situationen und verschiedenen Umgebungen breite Anwendung. Hier sind einige vereinfachte Beispiele:

- **Hauskomfort:** Passen Sie Ihre Heizung oder Kühlung automatisch basierend auf aktuellen Temperaturmessungen an, um Ihren Wohnraum angenehm zu gestalten.
- **Gartenarbeit:** Überwachen Sie die Temperaturen im Gewächshaus, um sicherzustellen, dass die Pflanzen unter optimalen Bedingungen wachsen. Fügen Sie automatisierte Systeme hinzu, um die Temperaturen



bei Bedarf anzupassen.

- **Lebensmittelsicherheit:** Überwachen Sie die Temperaturen von Kühlschränken und Gefriertruhen, um sicherzustellen, dass Lebensmittel sicher zum Verzehr bleiben, insbesondere in Restaurants oder beim Transport von Lebensmitteln.

**Gesundheitswesen:** Überwachen und protokollieren Sie die Temperaturen in Lagerbereichen für temperaturempfindliche Medikamente und Impfstoffe, um sicherzustellen, dass diese wirksam bleiben.

# Lektionen22: Fernbedienbare Farbenfrohe Beleuchtung

Füllen Sie diese Tabelle während **“Codeerstellung - Ermitteln der Schlüsselwerte“** aus.

1. Bitte drücken Sie sorgfältig jede Taste auf der Fernbedienung und notieren Sie die entsprechenden Tastenwerte in der Tabelle Ihres Handbuchs.



Tastennamen	Tastencode	Tastennamen	Tastencode
POWER	0x45	0	0x16
MODE	0x46	1	0xC
MUTE	0x47	2	0x18
PLAY/PAUSE	0x44	3	0x5E
BACKWARD	0x40	4	0x8
FORWARD	0x43	5	0x1C
EQ	0x7	6	0x5A
-	0x15	7	0x42
+	0x9	8	0x52
CYCLE	0x19	9	0x4A
U/SD	0xD		

## Lektion 23: Spiele "Blink, Blink, kleiner Stern"

Beantworte die folgende Frage während der "Code-Erstellung - Array".

1. Du kannst auch Berechnungen mit den Elementen im Array durchführen, wie zum Beispiel die Änderung zu `Serial.println(melody[i] * 1.3)`. Welche Daten erhältst du und warum?

Die Zahl 1.3 ist eine Gleitkommazahl. Wenn eine Ganzzahl aus dem `melody`-Array (das vom Typ `int` ist) mit 1.3 multipliziert wird, wird das Ergebnis der Operation automatisch in eine Gleitkommazahl (`float`) umgewandelt.

Für jede Frequenz der Noten in diesem Array ergibt das Multiplizieren mit 1.3 und anschließendes Ausgeben des Ergebnisses:

```
340.6  
340.6  
509.6  
509.6  
572.0  
572.0  
509.6  
...
```

Beantworten Sie nach Abschluss der Lektion die folgende Frage:

1. Wenn Sie den passiven Summer im Schaltkreis durch einen aktiven Summer ersetzen, können Sie dann sicher "Twinkle Twinkle Little Star" spielen? Warum?

Wenn Sie den passiven Summer durch einen aktiven Summer ersetzen, um "Twinkle Twinkle Little Star" zu spielen, wird es nicht wie gewünscht funktionieren. Aktive Summer können nur einen einzelnen Ton erzeugen, da sie einen eingebauten Oszillator haben. Daher können Sie die Tonhöhe nicht steuern, um die Melodie genau zu spielen; Sie würden lediglich einen wiederholenden Piepton im Rhythmus des Liedes hören, nicht die tatsächlichen Noten.

## Lektionen 24: Der Pomodoro-Timer

Beantworten Sie die folgende Frage während der “Codierungserstellung - millis()”.

1. Wenn `delay(100)` in `delay(1000)` geändert wird, was wird dann mit dem Programm passieren? Warum?

```
if (currentMillis - previousMillis >= interval) {  
    previousMillis = currentMillis; // Save the last time the buzzer  
    beeped  
    digitalWrite(buzzerPin, HIGH); // Make a voice  
    delay(100);  
    digitalWrite(buzzerPin, LOW); // silence  
}
```

Im ursprünglichen Code piepst der Summer für etwa 100 Millisekunden alle 1000 Millisekunden (1 Sekunde, wie durch die `interval` Variable festgelegt), gefolgt von einer Stille von 900 Millisekunden. Nach der Änderung wird der Summer für 1000 Millisekunden alle 1000 Millisekunden piepsen und dann fast sofort wieder piepsen, da das nächste Intervall fast sofort beginnt. Dadurch verwandelt sich die Änderung der Verzögerung von 100 auf 1000 Millisekunden den Summer von kurzen Pieptönen in einen kontinuierlichen Ton, der störender und ungeeigneter für die ursprüngliche Absicht wird.

Die Änderung von `delay(100)` auf `delay(1000)` in deinem Code bewirkt, dass der Summer für eine volle Sekunde anstatt nur für einen kurzen Piepton ertönt, da die Pausenzeit, während der der Summer eingeschaltet ist, verlängert wird. Dies führt zu längeren Summergeräuschen und selteneren Programmschleifen, was das Programm während dieser Intervalle weniger reaktionsschnell gegenüber anderen Aufgaben machen kann.

Beantworte nach dem Abschluss der Lektion die folgende Frage.

Denke über andere Orte in deinem Leben nach, an denen du die Zeit „hören“ kannst. Liste einige Beispiele auf und schreibe sie in dein Handbuch!

Die Zeit zu hören ist ein faszinierendes Konzept, und es gibt mehrere alltägliche Situationen, in denen wir dies erleben können. Hier sind einige Beispiele, die du dir notieren könntest:

- **Uhren und Wecker:** Das Ticken von analogen Uhren oder die spezifischen Pieptöne digitaler Uhren, die jede vergangene Sekunde oder Minute signalisieren.
- **Küchentimer:** Das Ticken und das abschließende Signal eines mechanischen oder digitalen Küchentimers, der die Koch- oder Backzeit herunterzählt.
- **Schulglocken:** Das Läuten von Glocken in Schulen, das den Beginn und das Ende von Unterrichtsstunden oder Pausen markiert.
- **Durchsagen im öffentlichen Verkehr:** Die Pieptöne oder Glocken, die Ankündigungen an Bahnhöfen oder in Bussen vorausgehen und die bevorstehende Abfahrt oder Ankunft kennzeichnen.
- **Mikrowelle:** Der Piepton, wenn der Timer endet und signalisiert, dass der Erwärmungsprozess abgeschlossen ist.
- **Fitness-Tracker oder Sportuhren:** Die Pieptöne oder Alarme, die das Ende einer festgelegten Zeit während des Trainings oder der Intervalle anzeigen.

## Lektionen25: Umgekehrtes Radarsystem

Beantworte nach Abschluss der Lektion die folgende Frage.

---

1. In diesem Projekt haben wir einen aktiven Summer als Warnmechanismus verwendet, aber ein passiver Summer könnte ebenfalls ähnliche Funktionen erfüllen. Wenn du den aktiven Summer durch einen passiven Summer ersetzen würdest, wie müsste der Code geändert werden?

Wenn du dich entscheidest, den aktiven Summer in deinem Arduino-Projekt durch einen passiven Summer zu ersetzen, musst du den Code, der den Summer steuert, anpassen, da passive Summer ein wechselndes Signal benötigen, um Töne zu erzeugen, im Gegensatz zu aktiven Summern, die nur ein hochsignalisiertes Signal zum Piepen benötigen.

Hier ist, wie du die Funktion `beep()` in deinem Code anpassen kannst, damit sie mit einem passiven Summer funktioniert:

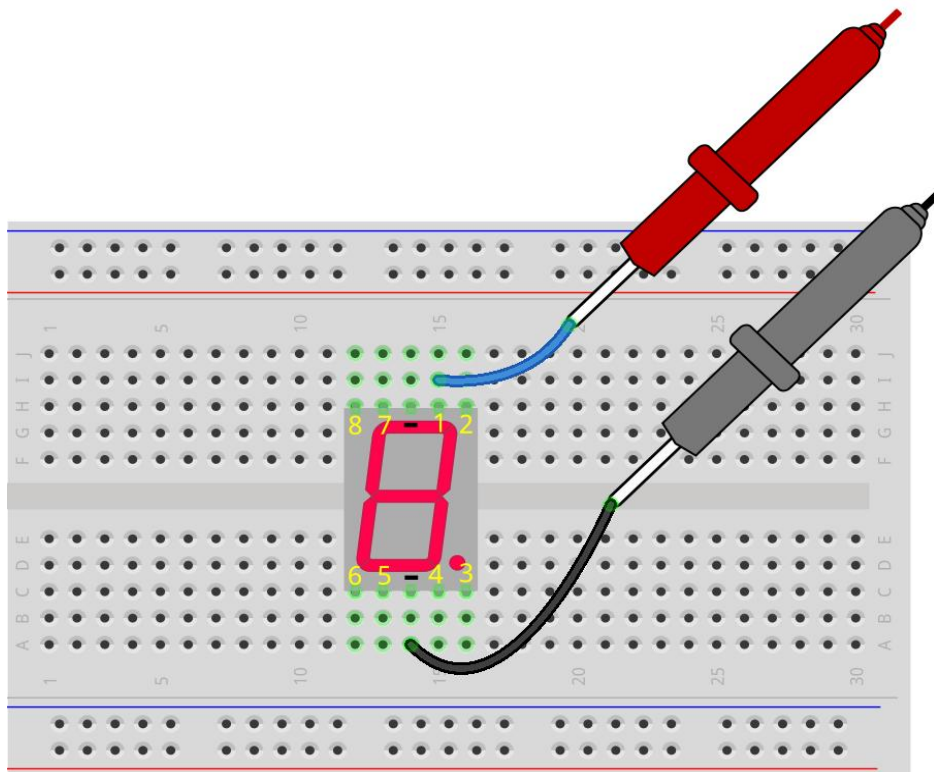
```
// Function to make passive buzzer beep
void beep() {
    // Generate a tone on the buzzer pin
    tone(BUZZER_PIN, 2000); // 2000 Hz frequency
    delay(100);             // Beep duration: 100 milliseconds
    noTone(BUZZER_PIN);     // Stop the tone
}
```



## Lektionen26: Cyberwürfel

Beantworten Sie die folgenden Fragen während des Themas **“Lernen des 7-Segment-Displays”**.

1. Wenn ein Segment leuchtet, beziehen Sie sich auf dieses Diagramm, um die Pin-Nummer des Segments und dessen ungefähre Position in der Tabelle des Handbuchs festzuhalten.



Pin	Segmentnummer	Position
1	A	Oben
2	B	Oben rechts
3	C	Unten rechts
4	D	Unten
5	E	Unten links
6	F	Oben links
7	G	Mitte
8	Dezimalpunkt	Der Punkt

1. Aus den obigen Tests ist bekannt, dass das Display im Kit eine gemeinsame Kathode hat, was bedeutet, dass Sie nur den gemeinsamen Pin mit GND verbinden und den anderen Pins eine hohe Spannung zuführen müssen, um die entsprechenden Segmente zum Leuchten zu bringen. Wenn Sie möchten, dass das Display die Zahl 2 anzeigt, welche Pins sollten dann mit einer hohen Spannung versorgt werden? Warum?



Für die Zahl 2 müssen die Segmente a, b, d, e und g aktiviert (auf hohe Spannung gesetzt) werden, da diese die Zahl 2 auf dem Display bilden. Die Segmente f, c und dp (Dezimalpunkt, falls vorhanden) sollten ausgeschaltet (auf niedrige Spannung) bleiben, da sie nicht zur Darstellung der Zahl 2 gehören.

Die Pins, die mit einer hohen Spannung versorgt werden sollten, sind also die, die mit den Segmenten a, b, d, e und g verbunden sind, um die Zahl 2 korrekt darzustellen.

## Lektion 27: Fließendes Licht mit 74HC595

Beantworte die folgende Frage während “Code Creation - LEDs zum Leuchten bringen”.

1. Was passiert, wenn wir `MSBFIRST` in `shiftOut(DS, SHcp, MSBFIRST, B11101110);` ändern? Warum?

Wenn Sie `MSBFIRST` in `LSBFIRST` ändern, wird die Reihenfolge der Bits umgekehrt, und das Byte wird beginnend mit dem wenigsten signifikanten Bit (dem rechten) herausgeschoben. Wenn Sie das Schieberegister zur Steuerung von LEDs verwenden, wird durch die Änderung der Bitreihenfolge auch die Reihenfolge, in der die LEDs aufleuchten, umgekehrt. Anstatt in der ursprünglich programmierten Sequenz aufzulodern, leuchten sie in umgekehrter Reihenfolge auf.

Beantworten Sie nach Abschluss der Lektion die folgende Frage.

1. Wie sollten die Elemente des `dataArray[]`-Arrays geändert werden, wenn wir gleichzeitig drei LEDs leuchten lassen und den Eindruck erwecken möchten, dass sie "fließen"?

Sie würden mit den ersten drei LEDs beginnen, die eingeschaltet sind, und dann in jedem nachfolgenden Muster eine LED nach rechts verschieben, bis die letzten drei LEDs leuchten. So könnten Sie diese Muster in binärer Form definieren:

```
B11100000: LEDs 1, 2, 3 are on; others are off.  
B01110000: LEDs 2, 3, 4 are on; others are off.  
B00111000: LEDs 3, 4, 5 are on; others are off.  
B00011100: LEDs 4, 5, 6 are on; others are off.  
B00001110: LEDs 5, 6, 7 are on; others are off.  
B00000111: LEDs 6, 7, 8 are on; others are off.
```

```
byte dataArray[] = { B11100000, B01110000, B00111000, B00011100,  
B00001110, B00000111 };
```

## Lektion 28: Zahl anzeigen

Füllen Sie diese Tabelle während **“Binärzahlen für die Ziffern 0 bis 9”** aus

1. Now that we know the binary representations for digits 0 and 2, please fill in the binary numbers for the remaining digits in the table below.

Zahl	Binär
0	<i>B00111111</i>
1	<i>B00000110</i>
2	<i>B01011011</i>
3	<i>B01001111</i>
4	<i>B01100110</i>
5	<i>B01101101</i>
6	<i>B01111101</i>
7	<i>B00000111</i>
8	<i>B01111111</i>
9	<i>B01101111</i>

Füllen Sie diese Tabelle während der **“Binärumwandlung”** aus.

1. Bitte wandeln Sie die Binärzahlen, die die Ziffern 0 bis 9 darstellen, mit einem Taschenrechner in Dezimal- und Hexadezimalzahlen um und füllen Sie die Tabelle aus. Dies dient Ihnen als schnelle Referenz für Basisumwandlungen.

Number	Binär	Dezimal	Hexadecimal
0	B00111111	63	0x3F
1	B00000110	6	0x06
2	B01011011	91	0x5B
3	B01001111	79	0x4F

4	B01100110	102	0x66
5	B01101101	109	0x6D
6	B01111101	125	0x7D
7	B00000111	7	0x07
8	B01111111	127	0x7F
9	B01101111	111	0x6F

## Lektion 29: Pflanzenmonitor

Beantworte die folgende Frage während „Codeerstellung - Bodenfeuchtigkeit lesen“.

1. Im bereitgestellten Code verstehen wir, dass ein höherer Feuchtigkeitsgehalt zu einem niedrigeren Sensormesswert führt, und Feuchtigkeit wird typischerweise als Prozentsatz angegeben. Wie können wir den Code ändern, um den Bodenfeuchtigkeitswert als Prozentsatz anzuzeigen?

Um den Bodenfeuchtigkeitswert als Prozentsatz anzuzeigen, müssen Sie zunächst den Wertebereich verstehen, den Ihr Feuchtigkeitssensor bei vollständiger Trockenheit und vollständiger Nässe zurückgibt. Diese Werte können dann verwendet werden, um die analoge Ausgabe des Sensors auf eine Prozentskala zuzuordnen, wobei 0 % vollständig trocken und 100 % vollständig nass bedeutet.

So können Sie Ihren Code ändern, um die Bodenfeuchtigkeit als Prozentsatz zu berechnen und anzuzeigen:

```
const int moisturePin = A1; // Define the pin where the soil moisture
sensor is connected

void setup() {
    Serial.begin(9600); // Initialize serial communication at 9600
    baud rate
}

void loop() {
    int moistureValue = analogRead(moisturePin); // Read the analog
    value from the moisture sensor
    Serial.print("Raw Moisture Value: ");
    Serial.println(moistureValue); // Output the raw sensor value to
    the serial monitor for observation

    // Map the moisture value from the sensor's possible range (0 to 1023)
    to a percentage (0% to 100%)
    float moisturePercent = map(moistureValue, 0, 1023, 100, 0);
    Serial.print("Moisture Percentage: ");
    Serial.print(moisturePercent);
    Serial.println("%"); // Output the calculated moisture percentage
    delay(1000); // Delay for one second before the next reading to
    reduce data flooding
}
```



## Lektion 30: Arduino Radar System

Beantworte die folgende Frage während "3. Vorbereitung des Ultraschallmoduls"

1.Im obigen Code nimmt das Ultraschallmodul jede Gradmessung vor. Wenn du der Meinung bist, dass die Messungen zu häufig sind und du alle 5 Grad eine Messung durchführen möchtest, wie sollte der Code angepasst werden?

Um den Code so anzupassen, dass das Ultraschallmodul alle 5 Grad anstelle von jedem Grad eine Messung vornimmt, musst du die Schleife ändern, die die Bewegung des Servos steuert. Konkret würdest du den Inkrementwert in der for-Schleife von 1 auf 5 ändern. Dadurch bewegt sich der Servo bei jeder Iteration der Schleife um 5 Grad, was die Häufigkeit der Messungen verringert.

Hier ist, wie du die Schleife im Code anpassen kannst:

```
void loop() {  
  // rotates the servo from 15 to 165 degrees in steps of 5 degrees  
  for (int i = 15; i <= 165; i += 5) {  
    ...  
  }  
  // rotates the servo from 165 to 15 degrees in steps of 5 degrees  
  for (int i = 165; i >= 15; i -= 5) {  
    ...  
  }  
}
```

## Lektion 31: Zahl erraten

Beantworten Sie nach Abschluss der Lektion die folgende Frage

---

Welche zusätzlichen Komponenten können hinzugefügt werden, um den Spielspaß zu erhöhen? Welche Rollen spielen sie im Spiel?

Um den Spaß am Zahlenratenspiel zu erhöhen, sollten folgende Komponenten hinzugefügt werden:

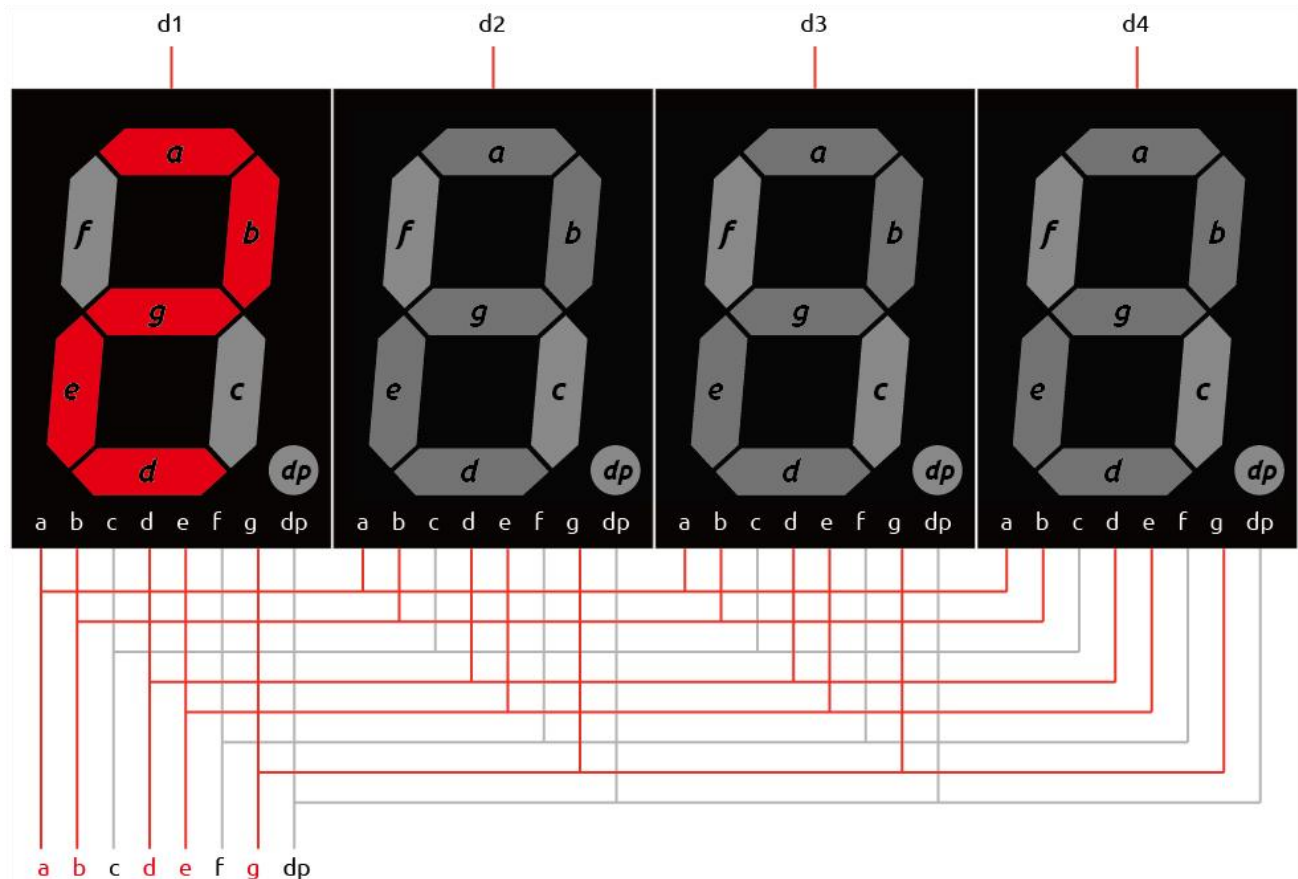
- **Soundeffekte mit einem Lautsprecher oder Buzzer:** Bieten Sie akustisches Feedback für richtige oder falsche Antworten, um das Spiel ansprechender zu gestalten.
- **RGB-LEDs oder LED-Streifen:** Nutzen Sie visuelle Effekte wie blinkende Farben für Feedback, um das Spiel visuell spannend zu machen.
- **Punktetafel mit zusätzlichem Display:** Zeigen Sie Punktzahlen oder einen Timer an, um den Wettbewerbsgeist zu fördern.
- **Wi-Fi- oder Bluetooth-Modul:** Ermöglichen Sie das Spielen aus der Ferne oder Multiplayer-Modi, damit Spieler von verschiedenen Standorten aus antreten können.
- **Touchscreen-Display:** Verbessern Sie die Benutzeroberfläche für ein interaktiveres und modernes Spielerlebnis.

Diese Ergänzungen können das Spiel interaktiver, wettbewerbsorientierter und unterhaltsamer machen.

## Lektionen 32: Stoppuhr

Beantworten Sie die folgende Frage während **“Lernen des 4-stelligen 7-Segment-Displays”**.

1. Wenn Sie möchten, dass die linkeste Ziffer (d1) des 4-stelligen 7-Segment-Displays „2“ anzeigt, welche Pegel sollten dann die Pins d1~d4 und a~g haben?



Pin	HIGH or LOW	7-segment Display	HIGH or LOW
d1	LOW	a	HIGH
d2	HIGH	b	HIGH
d3	HIGH	c	LOW
d4	HIGH	d	HIGH
		e	HIGH
		f	LOW

		g	HIGH
		dp	LOW

Beantworten Sie die folgende Frage während “**Code-Erstellung - Zahlen auf einer Ziffer scrollen**”.

In der Programmierung sind bitweise Operationen wie **AND** und **OR** entscheidend für die Manipulation einzelner Datenbits. Die bitweise **AND**-Operation (**&**) vergleicht jedes Bit ihrer Operanden und ergibt 1, wenn beide Bits 1 sind, andernfalls 0. Im Gegensatz dazu ergibt die bitweise **OR**-Operation (**|**) 1, wenn mindestens eines der Bits 1 ist, und 0 nur, wenn beide Bits 0 sind.

Angesichts dieser Informationen betrachten Sie den Ausdruck **(B01011011 >> 2) | 1**. Nachdem die Binärzahl **B01011011** um **2** Positionen nach rechts verschoben wurde, welches Ergebnis ergibt die bitweise **OR**-Verknüpfung mit 1?

1) **Rechtsverschiebung:** Die Operation **B01011011 >> 2** verschiebt die Bits von **B01011011** um 2 Positionen nach rechts:

Ursprünglich: **01011011**

Nach der Verschiebung: **00010110**

## Fazit

Das Endergebnis von **(B01011011 >> 2) | 1** ist **00010111**, was 23 im Dezimalsystem entspricht. Die **| 1**-Operation stellt sicher, dass das niedrigstwertige Bit (LSB) auf 1 gesetzt wird, was die Zahl leicht erhöht, falls das LSB ursprünglich 0 war. Dies zeigt, wie die Rechtsverschiebung den Wert verringert, indem sie seine Größe reduziert (durch Division durch Potenzen von 2), während die **OR**-Operation verwendet werden kann, um ein spezifisches Bitmuster sicherzustellen, wie zum Beispiel das Setzen des LSB.

## Lektion 33: Erkundung des RF522-RFID-Moduls

Beantworte nach Abschluss der Lektion die folgende Frage

---

1. Jetzt, wo du verstanden hast, wie man das RC522-RFID-Modul zum Lesen oder Schreiben von Karten- oder Tag-Informationen und zur Anzeige auf einem LCD verwendet, wie würdest du ein gängiges Zugangskontrollsystem für den täglichen Gebrauch entwerfen? Beschreibe deinen Entwurfsansatz.

Um ein praktisches Zugangskontrollsystem für den täglichen Gebrauch zu entwerfen, würde mein Ansatz auf Benutzerfreundlichkeit, Sicherheit und Skalierbarkeit ausgerichtet sein. Hier ist eine Übersicht meiner Designstrategie:

### Systemkomponenten

- **RFID-Leser (RC522):** Zur Benutzeridentifikation über RFID-Karten oder -Tags, die leicht zu verteilen und zu verwalten sind.
- **Mikrocontroller (Arduino):** Funktioniert als zentrale Steuereinheit zur Kontrolle des RFID-Lesers, zur Verarbeitung der Eingaben und zur Verwaltung der Ausgabefunktionen.
- **LCD-Display (I2C LCD1602):** Um den Benutzern Echtzeit-Feedback zu geben, wie z.B. den Zugangsstatus oder Anweisungen.
- **Summer:** Für akustisches Feedback, das anzeigt, ob der Zugang gewährt oder verweigert wurde, was das Benutzererlebnis verbessert.
- **Tastatur:** Ermöglicht eine alternative Eingabemethode über einen PIN-Code und bietet eine Backup-Option für den Fall, dass RFID-Tags vergessen oder verloren gehen.

### Benutzerinteraktion

- Benutzer können entweder ihre RFID-Karte scannen oder einen PIN-Code eingeben. Das System gibt sofort visuelles Feedback auf dem LCD und akustisches Feedback über den Summer.
- Bei erfolgreicher Authentifizierung wird die Tür für eine festgelegte Zeit entriegelt und anschließend automatisch wieder verriegelt.
- Bei verweigertem Zugang zeigt das System eine Nachricht an und gibt einen Alarmton von sich.

## Lektionen 34: Zugangskontrollsystem

Beantworten Sie die folgende Frage während **“Codeerstellung - Den Schrittmotor drehen lassen“**.

1. Wenn Sie eine vollständige Umdrehung in eine Richtung und dann eine vollständige Umdrehung in die entgegengesetzte Richtung erreichen möchten, die in diesem Zyklus fortgesetzt wird, wie sollte der Code geändert werden?

Um den Code so zu ändern, dass der Schrittmotor eine vollständige Umdrehung in eine Richtung und dann eine vollständige Umdrehung in die entgegengesetzte Richtung ausführt, wobei kontinuierlich zwischen den beiden Richtungen gewechselt wird, können Sie die Funktion `loop()` in Ihrem Arduino-Sketch anpassen. So können Sie es umsetzen:

- Legen Sie die Anzahl der Schritte fest, die für eine vollständige Umdrehung erforderlich sind. Da Ihr Schrittmotor 2048 Schritte pro Umdrehung benötigt (wie durch die Konstante `STEPS` definiert), verwenden Sie diese Zahl, um den Motor eine volle Umdrehung auszuführen.
- Verwenden Sie die Funktion `stepper.step()` mit positiven und negativen Werten. Positive Werte drehen den Motor in eine Richtung, während negative Werte ihn in die entgegengesetzte Richtung drehen.

Hier ist die modifizierte Version Ihrer `loop()` Funktion:

```
void loop() {  
  // Set the speed of the stepper  
  stepper.setSpeed(5);  
  
  // Rotate clockwise  
  stepper.step(STEPS); // Move 2048 steps (one full revolution)  
  delay(1000);         // Wait for 1 second  
  
  // Rotate counterclockwise  
  stepper.step(-STEPS); // Move -2048 steps (one full revolution in  
the opposite direction)  
  delay(1000);         // Wait for 1 second  
}
```



Beantworten Sie die folgende Frage während der “**Code-Erstellung - Zugangskontrollsystem**”.

---

1. Jetzt, da ein grundlegendes Zugangskontrollsystem eingerichtet ist, welche zusätzlichen Komponenten könnten hinzugefügt werden, um die Funktionalität und Flexibilität zu verbessern?

Hier sind einige zusätzliche Komponenten, die in Ihr Zugangskontrollsystem integriert werden können, um dessen Funktionalität und Flexibilität zu verbessern:

- **Buzzer für akustisches Feedback:** Liefert akustisches Feedback bei Zugriffsereignissen und verbessert die Interaktion der Benutzer durch Signale für gewährten oder verweigerten Zugriff sowie Systemwarnungen.
- **Kamera zur visuellen Identifizierung:** Ermöglicht die Eingabe von PIN-Codes als alternative Authentifizierungsmethode, nützlich für Benutzer ohne RFID-Karten oder für Systeme, die eine Zwei-Faktor-Authentifizierung erfordern.
- **Camera for Visual Identification:** Nimmt Bilder oder Videos zur zusätzlichen Sicherheitsüberprüfung auf und führt ein visuelles Protokoll von Zugriffsereignissen.
- **Bewegungsmelder:** Erfasst die Anwesenheit vor Erreichen des Kontrollpunkts, aktiviert das System oder alarmiert die Sicherheitskräfte bei unbefugten Zugriffsversuchen.