

Vielen Dank, dass Sie sich für das **Beginner's Lab Kit**

Willkommen beim Beginner's Lab Kit, einem umfassenden Starterpaket, das speziell für Neueinsteiger in die Welt der Elektronik und Programmierung entwickelt wurde. Dieses Kit enthält eine Reihe essenzieller Komponenten wie LEDs, Widerstände, einen Summer, Potentiometer, Fotowiderstände, Thermistoren, Drucktasten, digitale Anzeigen und ein Ultraschallmodul. Eines der herausragenden Merkmale dieses Kits ist das enthaltene Multimeter, ein unverzichtbares Werkzeug, mit dem Sie Strom, Spannung und Widerstand in Ihren Schaltungen messen können. Diese Ergänzung ist besonders nützlich, um Ihr Verständnis dafür zu vertiefen, wie jede Komponente funktioniert.

Die mit diesem Kit bereitgestellte Kursabfolge ist um die Arduino-Programmiersyntax strukturiert und gewährleistet einen logischen und lehrreichen Fortschritt. Diese Struktur ermöglicht es Ihnen, Schaltungen Schritt für Schritt zu erstellen und dabei zu lernen, wie Sie die Programme schreiben, die diese steuern. Im Verlauf des Kurses werden Sie auf Fehlersuche-Herausforderungen stoßen, die Ihr Verständnis des Materials vertiefen.

Für Fragen oder Unterstützung wenden Sie sich bitte an uns unter service@sunfounder.com. Tauchen Sie mit dem Beginner's Lab Kit in Ihre Lernreise ein und beginnen Sie, die aufregende Welt der Elektronik durch Bauen, Programmieren und Erkunden zu entdecken!

Inhaltsverzeichnis

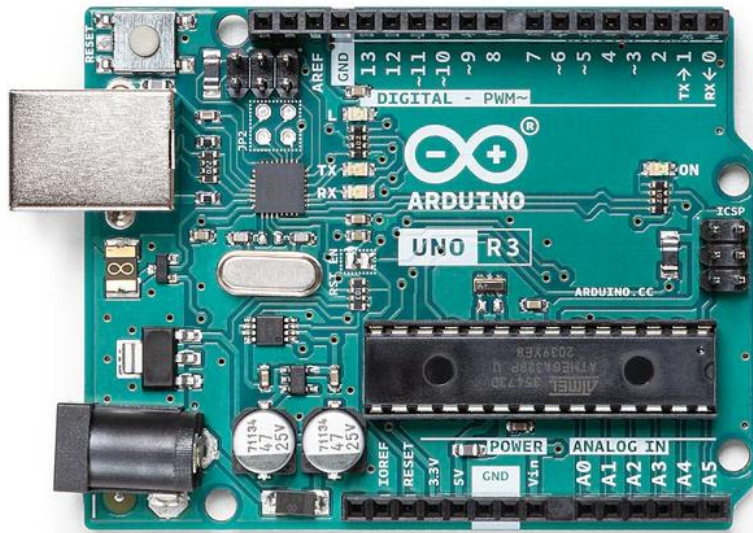
Was ist in Ihrem Kit enthalten	1
Lektion 2: Dein erster Schaltkreis	8
Lektion 3: Messen mit dem Multimeter	9
Lektion 4: Ohmsches Gesetz	10
Lektion 5: Reihenschaltung vs. Parallelschaltung	11
Lektion 6: LED blinken lassen	15
Lektion 7: Lassen Sie uns eine Ampel bauen!	16
Lektion 8: Ampel mit Fußgängertaster	17
Lektion 9: Dimmbare Schreibtischlampe	19
Lektion 10: Ein/Aus Schreibtischlampe	20
Lektion 11: Steuerung von LED-Arrays mit Potentiometer	21
Lektion 12: Die Farben des Regenbogens	22
Lektion 13: Das Spektrum des Sehens	23
Lektion 14: Zufällige Farben	24
Lektion 15: Kühle oder warme Farben	25
Lektion 16: Temperaturalarm	26
Lektion 17: Morsecode	27
Lektion 18: Lichtalarm	28
Lektion 19: Rückwärtseinparksystem mit Alarm	29
Lektion 20: Der Pomodoro-Timer	30
Lektion 21: Sirenenton	32
Lektion 22: Spielen Sie „Twinkle, Twinkle, Little Star “	34
Lektion 23: Cyber-Würfel	35
Lektion 24: Fließendes Licht mit 74HC595	37
Lektion 25: Zahl anzeigen	38

Was ist in Ihrem Kit enthalten

In unserem Kit finden Sie eine Vielzahl von Bauteilen und Komponenten, die Sie im Laufe dieses Kurses zum Bau von Schaltkreisen verwenden werden. Hier ist eine kurze Übersicht über den Inhalt.

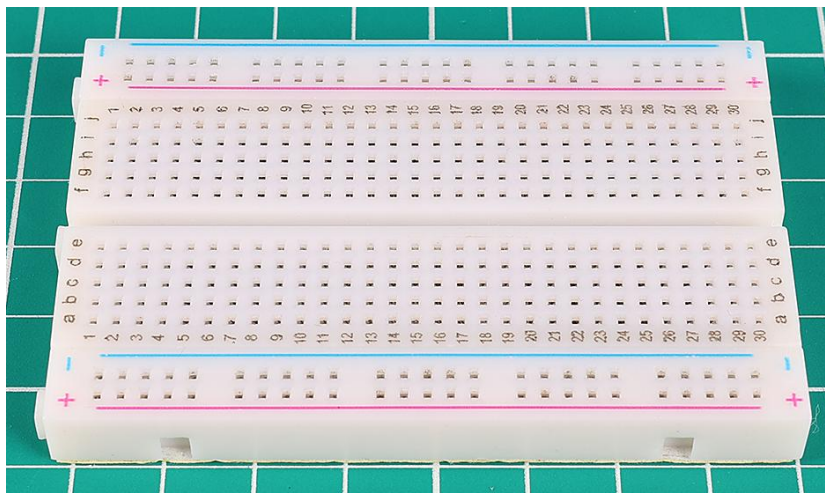
1 x Original Arduino Uno R3

Eine Mikrocontroller-Platine, die das Gehirn deiner Schaltungen ist. Sie enthält alles, was nötig ist, um den Mikrocontroller zu unterstützen; einfach mit einem USB-Kabel an deinen Computer anschließen oder mit einem AC-zu-DC-Adapter oder einer Batterie betreiben, um loszulegen.



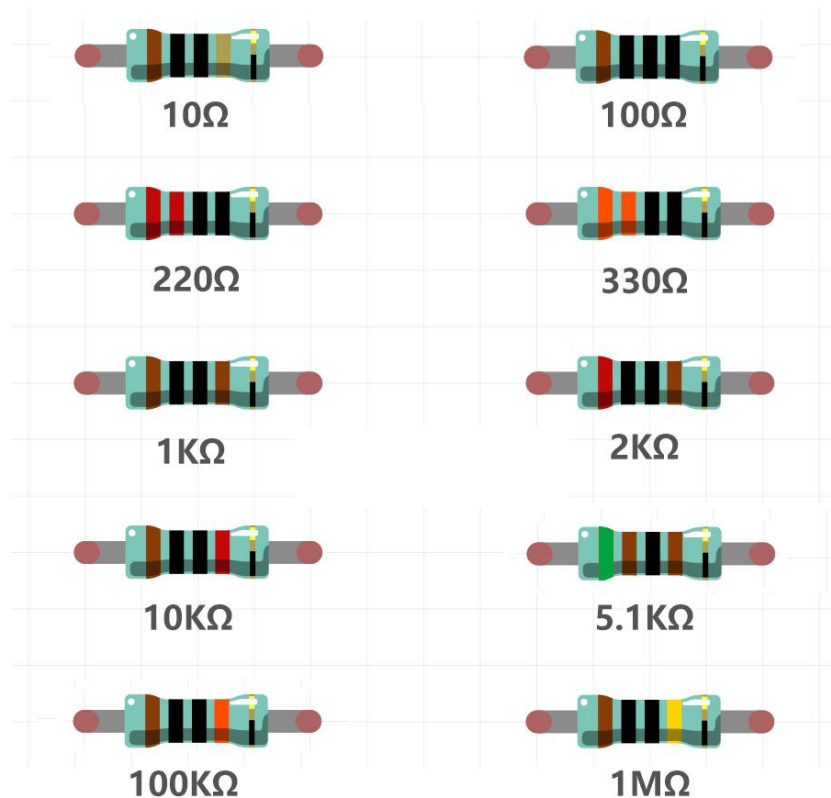
1 x 400-hole Breadboard

Eine lötfreie Platine, mit der du ganz einfach elektronische Schaltungen aufbauen kannst. Sie ist mit Reihen von Löchern versehen, um Drähte und Komponenten miteinander zu verbinden.



120 x Widerstände (je 10 Stück, 30 Stück 220Ω-Widerstände)

Ein Widerstand ist ein Bauteil, das den Fluss von elektrischer Energie behindert 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 (Ω). Die farbigen Streifen auf einem Widerstand geben seinen Widerstandswert und seine Toleranz an.



25 x LEDs (5 pro Farbe)

Diese bunte Auswahl an LEDs umfasst fünf Farben: Rot, Grün, Blau, Gelb und Weiß, die verschiedene Beleuchtungs- und Signalanforderungen abdecken. Geeignet für Anwendungen, die von einfachen Statusanzeigen bis hin zu komplexen dekorativen Beleuchtungsprojekten reichen, bieten diese LEDs eine reiche Farbauswahl, um die visuelle Attraktivität jedes Elektronikprojekts zu verbessern.



2 x RGB LEDs

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



1 x Photoresistor

Ein Fotowiderstand ist ein lichtempfindliches Bauteil, das seinen Widerstand in Abhängigkeit von der Lichtintensität ändert, der es ausgesetzt ist. Er eignet sich ideal zur Erstellung von lichtaktivierten Steuerungen und Sensoren in elektronischen Projekten.



1 x NTC Thermistor

Ein Thermistor ist ein temperaturabhängiger Widerstand. NTC-Thermistoren verringern ihren Widerstand bei steigender Temperatur, während PTC-Thermistoren ihren Widerstand mit zunehmender Temperatur erhöhen.



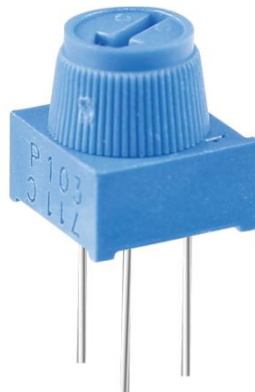
1 x Active Buzzer & 1 x Passive Buzzer

Ein Summer, erhältlich in aktiver und passiver Ausführung, ist ein akustisches Signalgerät, das einen Ton erzeugt, wenn elektrischer Strom angelegt wird. Er wird häufig in Alarmanlagen, Timern und Benachrichtigungssystemen verwendet.



1 x Potentiometer

Ein Potentiometer ist ein verstellbarer Widerstand mit drei Anschlüssen. Zwei Pins sind mit den Enden eines Widerstands verbunden, während der mittlere Pin an einen beweglichen Schleifer angeschlossen ist, der den Widerstand in zwei Teile teilt. Potentiometer, die häufig zur Spannungsanpassung in Schaltungen verwendet werden, funktionieren ähnlich wie Lautstärkeregler bei Radios.



10 x Small Buttons

Ein kleiner Druckknopf wird verwendet, um bei Betätigung eine physische Rückmeldung zu geben. Er wird häufig in elektronischen Geräten eingesetzt, um Aktionen auszulösen oder Befehle einzugeben.



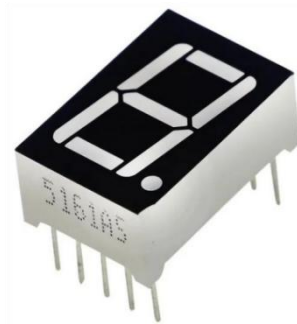
1 x 74HC595 Chip

Der 74HC594 ist ein Schieberegister, das zur Erweiterung der Ein-/Ausgabeports von digitalen Schaltungen verwendet wird, indem serielle Eingaben in parallele Ausgaben umgewandelt werden. Dadurch wird die Anzahl der benötigten Verbindungspins reduziert. Dieser Chip eignet sich besonders zum Steuern einer großen Anzahl von Ausgabegeräten, wie z. B. 7-Segment-Anzeigen, ohne zu viele Mikrocontroller-Pins zu belegen.



1 x 7-segment Display

Ein 7-Segment-Display ist ein komponentenförmiges Bauteil in Form einer „8“, das 7 LEDs enthält. Jede LED wird als Segment bezeichnet – wenn sie aktiviert wird, bildet ein Segment einen Teil einer anzuzeigenden Zahl.



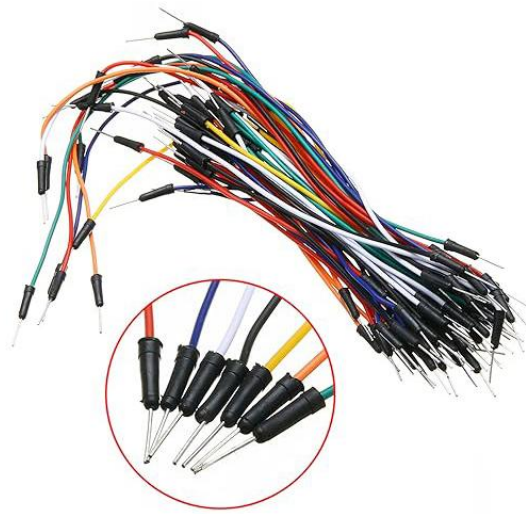
1 x Ultrasonic Module

Dies ist ein Ultraschallmodul, das Ultraschallwellen zur Abstandsmessung verwendet und damit die Position und Entfernung von Objekten präzise erfasst und misst. Es wird häufig in der Robotik, in Hindernisvermeidungssystemen und in der Automatisierungstechnik eingesetzt.



65 x Jumper Wires

Verbinden Sie die Komponenten auf dem Steckbrett miteinander und mit dem Arduino-Board.



10 x Male-to-Female DuPont-Kabel

Male-to-Female DuPont-Kabel sind speziell für die Verbindung von Modulen mit männlichen Stiftleisten, wie dem Ultraschallmodul, mit dem Steckbrett ausgelegt. Diese Kabel sind unerlässlich, um verschiedene Komponenten in Elektronikprojekten zu verbinden.



1 x USB Cable

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



1 x 9V Battery

Dies ist eine nicht wiederaufladbare 9V-Alkalibatterie. Sie müssen sie im Multimeter installieren.



1 x Multimeter mit roten und schwarzen Messleitungen

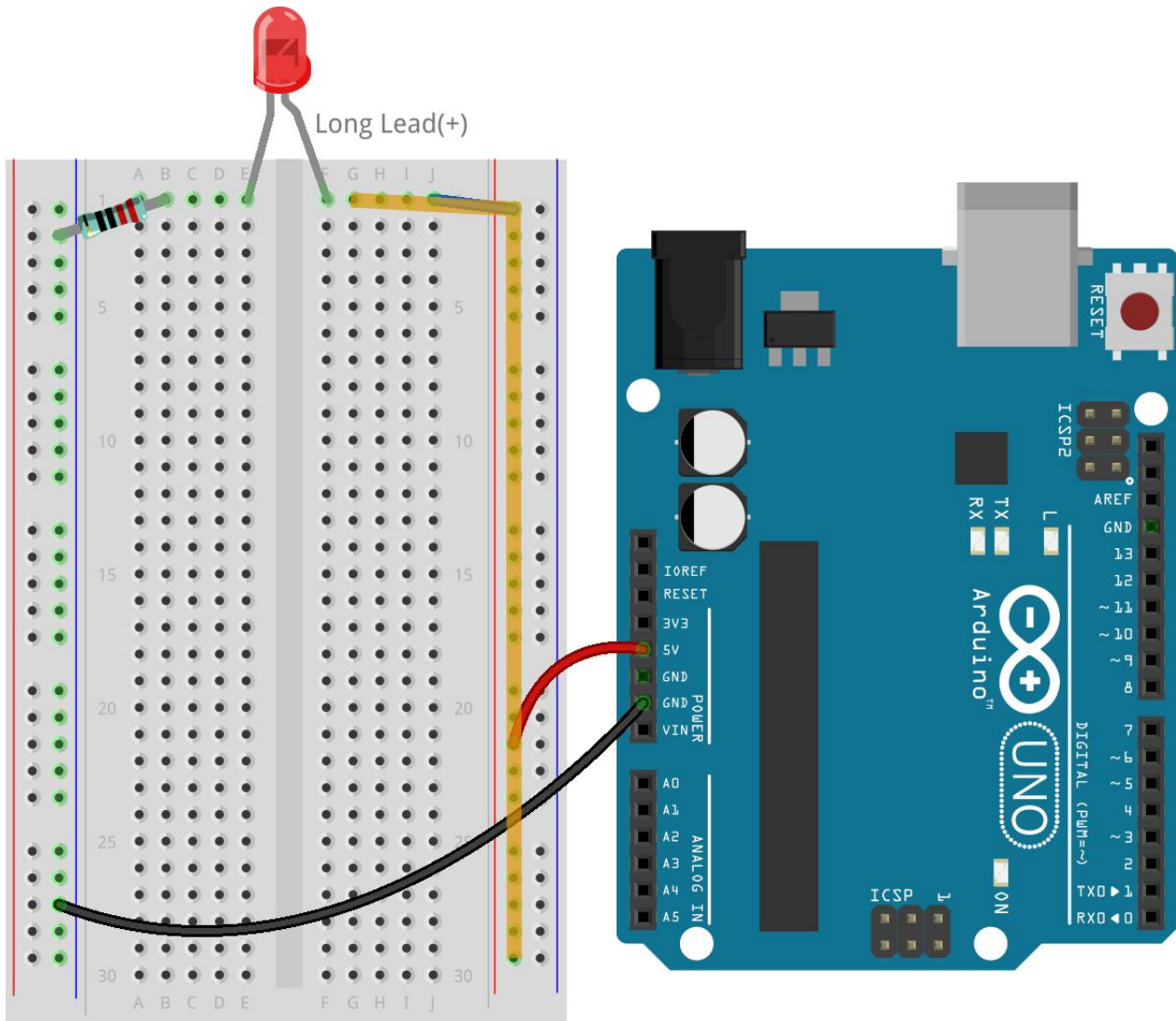
Dies ist ein vielseitiges Multimeter, das in der Lage ist, Spannung, Strom und Widerstand zu messen sowie andere elektrische Tests durchzuführen. Es ist ein unverzichtbares Werkzeug für Elektronik- und Elektroarbeiten.



Lektion 2: Dein erster Schaltkreis

Nach Abschluss der Lektion beantworte die folgenden Fragen

1. Entferne das rote Kabel vom Steckbrett und experimentiere, indem du es in verschiedene Löcher auf dem Steckbrett steckst. Beobachte dabei mögliche Veränderungen der LED. Skizziere die Lochpositionen, die das Leuchten der LED ermöglichen.



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

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

Lektion 3: Messen mit dem Multimeter

Beantworten Sie diese Frage, nachdem Sie "Mehr über das Multimeter erfahren" abgeschlossen haben!

1. Da du nun 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	Multimetereinstellung
9V Gleichspannung (DC)	<i>20V</i>
1K ohms	<i>2kΩ</i>
40 Milliampere	<i>200mA</i>
110V Wechselspannung (AC)	<i>200V~</i>

Fill out this table during "Measuring with a Multimeter"

Typ	Einheiten	Messergebnisse	Anmerkungen
Spannung	Volts	<i>≈ 5.13 volts</i>	
Strom	Milliampere	<i>≈ 13.54 milliamps</i>	
Widerstand	Ohms	<i>≈ 378.88 ohms</i>	

Lektion 4: Ohmsches Gesetz

Füllen Sie die folgende Tabelle aus während "Das ohmsche Gesetz anhand praktischer Experimente erkunden"

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

Widerstand	Beobachtungen
100Ω	<i>Heller</i>
1KΩ	<i>Hell</i>
10KΩ	<i>Dunkler</i>
1MΩ	<i>Fast aus</i>

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?

Nach dem Ohmschen Gesetz ($I = V/R$) verringert sich der Strom durch die LED, wenn der Widerstand steigt und die Spannung konstant gehalten wird, wodurch die LED dunkler wird. Bei 1MΩ ist der Strom zu gering, um die LED zum Leuchten zu bringen.

1. Nachdem Sie die Auswirkungen der Änderung des Widerstands beobachtet haben, belassen Sie den Widerstand bei 220 Ohm und ändern Sie die Spannungsversorgung des Stromkreises von 5V auf 3,3V. Notieren Sie alle Veränderungen in der Helligkeit der LED.

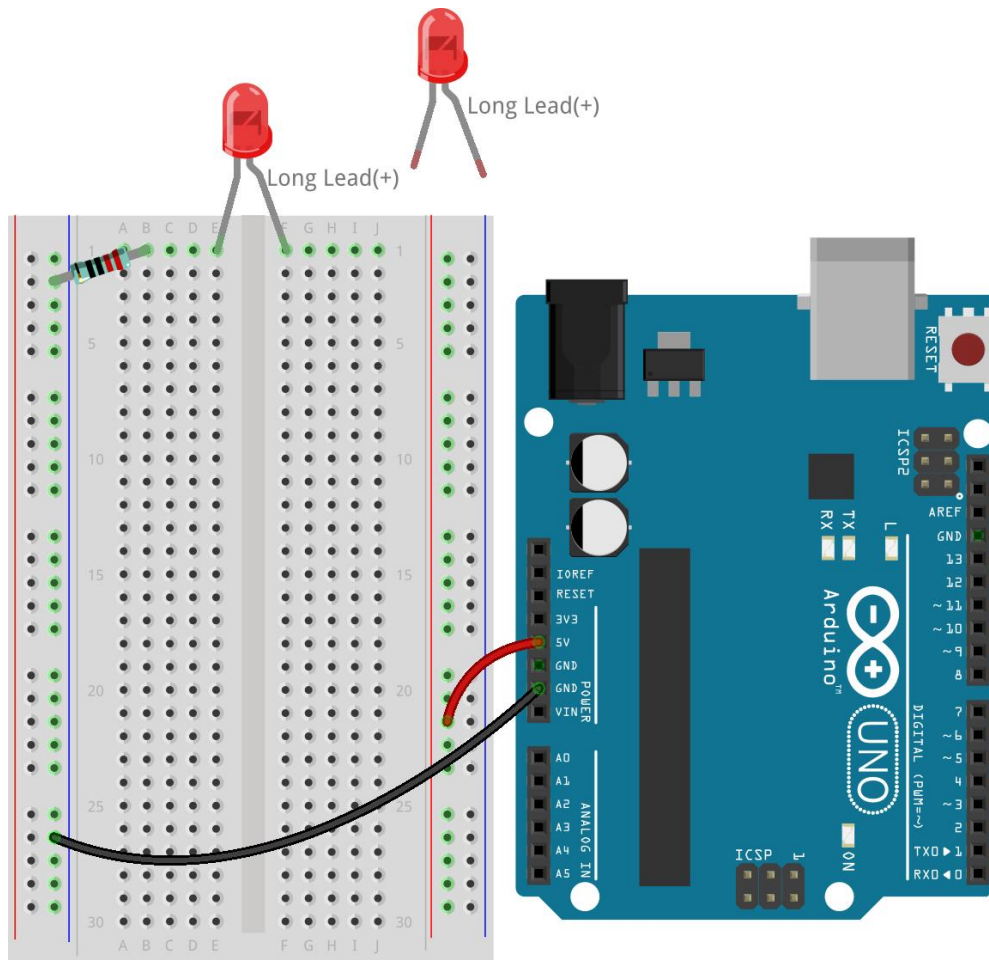
Sie werden feststellen, dass die LED bei 3,3V etwas dunkler ist als bei 5V. Warum ist das so?

Nach dem Ohmschen Gesetz, bei bekanntem Widerstand und neuer Spannung, sollte der Strom $I = V/R$ sein. Bei einer Verringerung der Spannung, während der Widerstand gleich bleibt, verringert sich der Strom, was die LED dunkler erscheinen lässt.

Lektion 5: Reihenschaltung vs. Parallelschaltung

Beantworten Sie die folgenden Fragen während des Abschnitts "Eintauchen in Reihenschaltungen".

1. Was passiert, wenn Sie eine LED entfernen? Warum tritt dies auf?



In einer Reihenschaltung leuchtet die andere LED nicht, wenn Sie eine LED entfernen. Dies liegt daran, dass in einer Reihenschaltung der Strom durch jede Komponente im Stromkreis fließen muss. Durch das Entfernen einer LED wird der Stromkreis unterbrochen, sodass kein Strom mehr durch die verbleibende LED fließen kann.

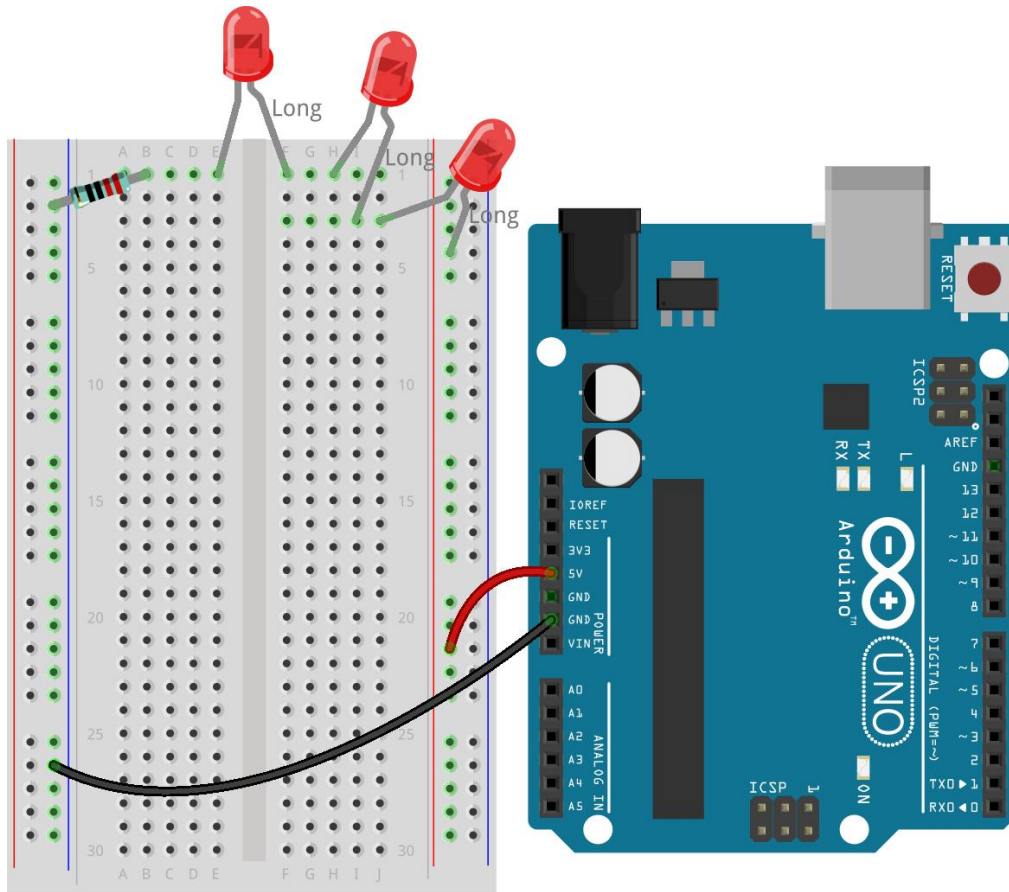
1. Messen Sie die Spannung jeder Komponente in der Reihenschaltung.

Schaltkreis	Widerstandsspannung	Spannung LED1	Spannung LED2	Gesamtspannung
2 LEDs	$\approx 1.13 \text{ volts}$	$\approx 1.92 \text{ volts}$	$\approx 1.92 \text{ volts}$	$\approx 4.97 \text{ volts}$

1. Messen Sie den Strom jeder Komponente in der Reihenschaltung.

Schaltkreis	Strom LED1	Strom LED2
2 LEDs	$\approx 4.43 \text{ milliamps}$	$\approx 4.43 \text{ milliamps}$

1. Wenn eine weitere LED zu diesem Stromkreis hinzugefügt wird, sodass insgesamt drei LEDs vorhanden sind, wie ändert sich die Helligkeit der LEDs? Warum? Wie ändern sich die Spannungen über den drei LEDs?

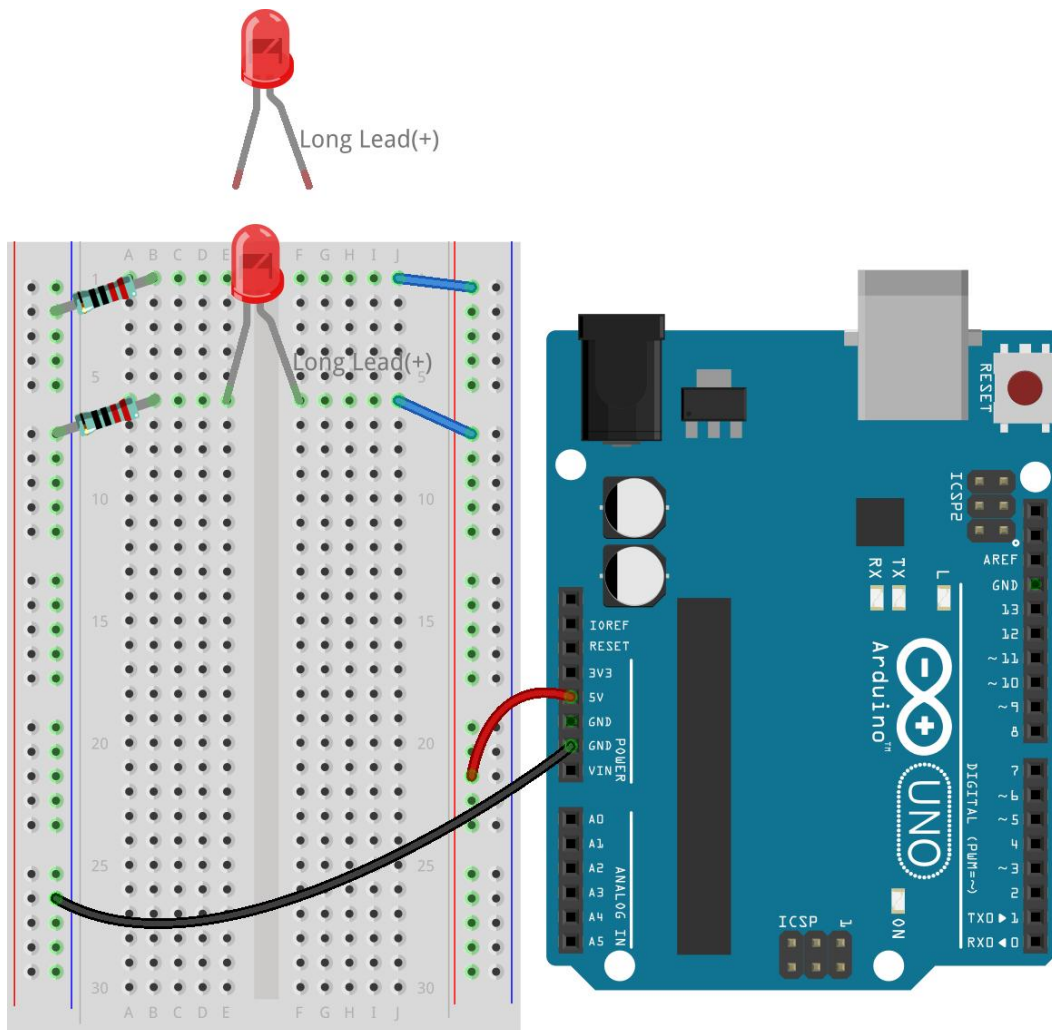


Das Hinzufügen einer weiteren LED zu einer Reihenschaltung mit bereits zwei LEDs führt in der Regel zu einer Verringerung der Helligkeit jeder LED. Dies liegt daran, dass die Gesamtspannung der Stromquelle auf mehr Komponenten aufgeteilt wird, wodurch der Spannungsabfall über jede LED geringer ist als bei nur zwei LEDs. Folglich fließt weniger Strom durch jede LED, was ihre Helligkeit verringert.

Bezüglich der Spannungen über die drei LEDs: Jede LED wird nun einen kleineren Anteil der Gesamtspannung des Stromkreises haben. Bleibt die Spannung der Stromquelle gleich, wird diese Spannung durch drei geteilt, vorausgesetzt, alle LEDs haben ähnliche elektrische Eigenschaften. Daher wird die Spannung über jede LED im Stromkreis etwa ein Drittel der Gesamtspannung der Stromquelle betragen.

Beantworten Sie die folgenden Fragen während des Abschnitts **“Eintauchen in Parallelschaltungen”**.

1. Was passiert in dieser Parallelschaltung, wenn eine LED entfernt wird? Warum tritt dies auf?



In einer Parallelschaltung leuchten die anderen LEDs weiterhin, wenn eine LED entfernt wird. Dies liegt daran, dass jede LED in einer Parallelschaltung ihren eigenen unabhängigen Pfad zur Stromquelle hat. Das Entfernen einer LED unterbricht den Stromfluss zu den anderen LEDs nicht, sodass sie unbeeinflusst bleiben und normal weiter funktionieren. Diese Anordnung ermöglicht es jeder Komponente in einer Parallelschaltung, unabhängig von den anderen zu arbeiten.

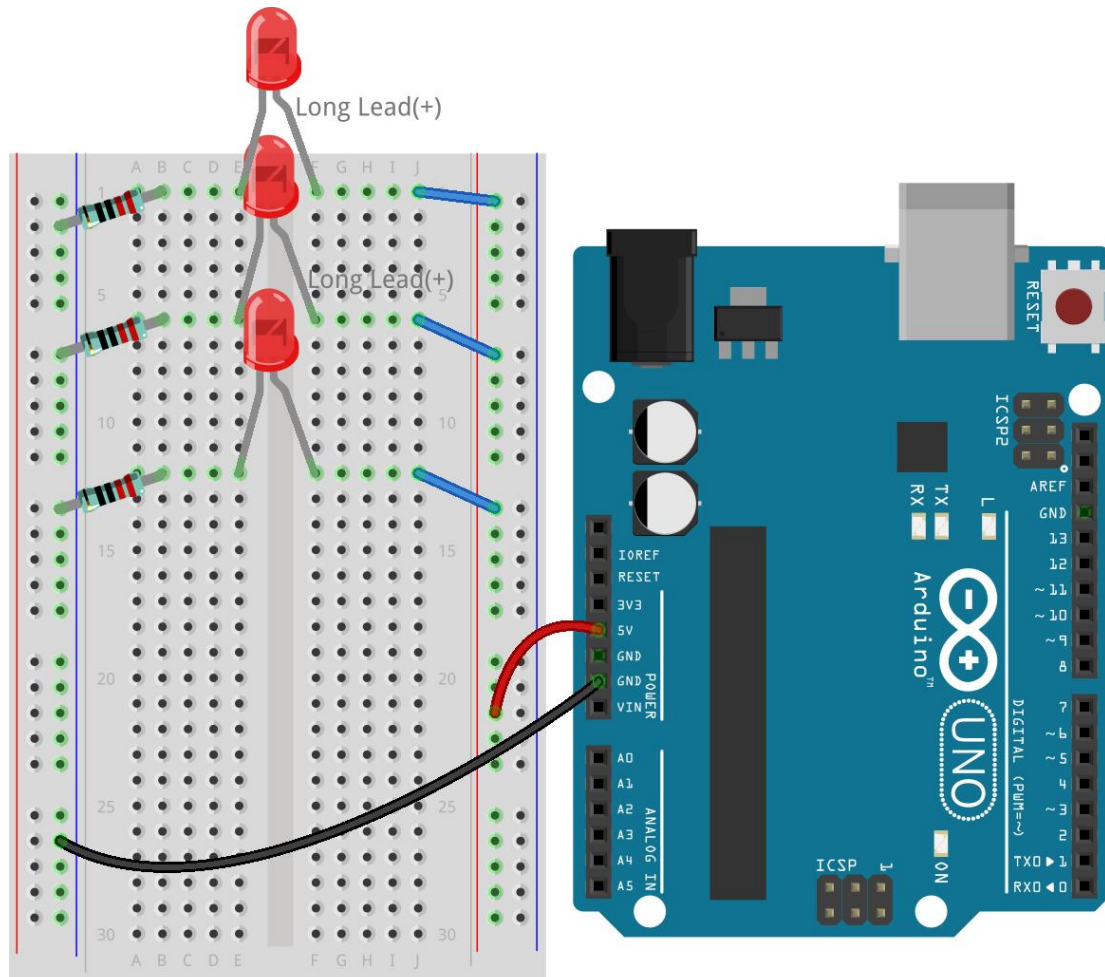
1. Füllen Sie die gemessene Spannung in die Tabelle ein.

Schaltkreis	Spannung Pfad1	Spannung Pfad2
2 LEDs	<i>≈5.00 volts</i>	<i>≈5.00 volts</i>

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

Circuit	Strom LED1	Strom LED2	Gesamtstrom
2 LEDs	$\approx 12.6 \text{ milliamps}$	$\approx 12.6 \text{ milliamps}$	$\approx 25.3 \text{ milliamps}$

1. Wenn eine weitere LED zu diesem Stromkreis hinzugefügt wird, was passiert mit der Helligkeit der LEDs? Warum? Notieren Sie Ihre Antwort in Ihrem Handbuch.



Wenn eine weitere LED zu einer Parallelschaltung hinzugefügt wird, bleibt die Helligkeit der vorhandenen LEDs in der Regel unverändert. Dies liegt daran, dass jede LED in einer Parallelschaltung ihren eigenen direkten Pfad 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 leuchten. 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 für zukünftige Referenzen.

Lektion 6: LED blinken lassen

Füllen Sie die folgende Tabelle während des Abschnitts “LEDs zum Leben erwecken” aus.

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

Zustand	Spannung Pin 3
HIGH	<i>≈4.95 volts</i>
LOW	<i>0.00 volts</i>

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

1. Laden Sie den obigen Code hoch, und Sie werden feststellen, dass die LED in einem 3-Sekunden-Intervall wiederholt blinkt. Wenn Sie möchten, dass sie sich nur einmal ein- und ausschaltet, was sollten Sie tun?

Sie können 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. Diese Änderung bewirkt, dass die LED nur einmal aufleuchtet und sich dann wieder ausschaltet. So können Sie Ihren 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:  
}
```

Lektion 7: Lassen Sie uns eine Ampel bauen!

Beantworten Sie die folgende Frage während des Abschnitts **“Pseudocode für eine Ampel schreiben”**.

Überlegen Sie, was geschehen muss, damit Ihr Schaltkreis wie eine Ampel funktioniert. Schreiben Sie in den dafür vorgesehenen Abschnitt Ihres Protokolls den Pseudocode auf, der beschreibt, wie Ihre Ampel funktionieren wird. Verwenden Sie dabei einfaches Deutsch.

Um eine Ampel mit einem Arduino zu simulieren, benötigen Sie eine Schaltung mit drei LEDs (rot, gelb und grün) und eine Sequenz, die die Beleuchtung so steuert, dass sie einer echten Ampel ähnelt. Hier ist eine einfache Pseudocode-Übersicht, die Sie in Ihr Protokoll schreiben können, um zu beschreiben, wie dieser Ampelschaltkreis 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.

Beantworten Sie die folgende Frage während des Abschnitts **“Pseudocode für eine Ampel schreiben”**.

Schauen Sie sich die Kreuzungen in Ihrer Umgebung an. Wie viele Ampeln gibt es dort normalerweise? Wie sind sie aufeinander abgestimmt?

In städtischen Gebieten gibt es an Kreuzungen oft Ampeln, um den Verkehrsfluss von Fahrzeugen und Fußgängern effizient zu steuern. Die Anzahl der Ampeln an einer Kreuzung kann stark variieren, abhängig von deren Größe und Komplexität. Eine einfache Kreuzung mit vier Fahrtrichtungen hat typischerweise mindestens vier Ampeln, eine für jede Fahrtrichtung. Komplexere Kreuzungen können zusätzliche Ampeln für Abbiegespuren, Fußgängerüberwege und andere verkehrssteuernde Anforderungen haben.

Lektion 8: Ampel mit Fußgängertaster

Nachdem Sie **“Den Schaltkreis aufbauen”** abgeschlossen haben, beantworten Sie die folgende Frage.

1. Ihre Ampel besteht aus einer Mischung aus Reihen- und Parallelschaltungen. Diskutieren Sie, welche Teile Ihres Schaltkreises in Reihe geschaltet sind und warum. Erklären Sie anschließend, welche Teile parallel geschaltet sind und warum.

In dem Schaltkreis 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 nicht gedrückten Zustand direkt mit Masse verbunden wird, um schwebende Eingänge zu verhindern.

Die drei LEDs, die an die Pins 3, 4 und 5 angeschlossen sind, sind parallel zueinander geschaltet. Jede LED arbeitet unabhängig, da sie an separate Steuereingänge angeschlossen sind und eine gemeinsame Stromversorgung teilen. Diese Anordnung ermöglicht es jeder LED, ohne Beeinflussung der anderen zu funktionieren, was für ein Ampelsystem entscheidend ist.

Füllen Sie diese Tabelle während der „Codeerstellung“ aus.

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

Tasterzustand	Spannung Pin 8	Zustand Pin 8
Freigabe	<i>0.00 volts</i>	<i>LOW</i>
Drücken	<i>≈4.97 volts</i>	<i>HIGH</i>

Beantworten Sie die folgende Frage nach Abschluss dieser Lektion.

1. Während des Testens werden Sie möglicherweise feststellen, dass die grüne LED nur blinkt, solange der Fußgängertaster gedrückt wird, aber Fußgänger können die Straße nicht überqueren, während sie den Taster dauerhaft drücken. Wie können Sie den Code ändern, um sicherzustellen, dass die grüne LED lange genug leuchtet, um einen sicheren Überweg zu ermöglichen, ohne dass ein ständiges Drücken erforderlich ist? Bitte notieren Sie die Pseudocode-Lösung in Ihrem Handbuch.

Um sicherzustellen, dass die grüne LED für Fußgänger leuchtet, ohne dass der Taster dauerhaft gedrückt werden muss, und um danach den normalen Ampelzyklus fortzusetzen, können Sie Ihren Pseudocode anpassen, um den Tastendruck zu überprüfen und dann den Betriebszustand basierend auf diesem Druck zu ändern. Hier ist eine optimierte und klarere Version des Pseudocodes, 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

Lektion 9: Dimmbare Schreibtischlampe

Füllen Sie diese Tabelle während „Den Schaltkreis aufbauen“ aus.

1. Drehen Sie den Potentiometer im Uhrzeigersinn von Position 1 auf 3 und messen Sie den Widerstand an jedem Punkt. Tragen Sie die Ergebnisse in die Tabelle ein.

Messpunkt	Widerstand (Kilohm)
1	1.52
2	5.48
3	9.01

1. Wie denken Sie, 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. Gemäß der Widerstandsmessung steigt der Widerstand zwischen A0 und GND, wenn der Potentiometer im Uhrzeigersinn gedreht wird. Da der Strom in einer Reihenschaltung konstant bleibt, führt gemäß dem Ohmschen Gesetz (Spannung = Strom \times Widerstand) eine Erhöhung des Widerstands zu einer Erhöhung 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.

Beantworten Sie die folgende Frage nach Abschluss dieser Lektion.

1. Wenn Sie die LED an einen anderen Pin, wie zum Beispiel Pin 8, anschließen und den Potentiometer drehen, wird sich die Helligkeit der LED dann weiterhin ändern? Warum oder warum nicht?

Wenn Sie die LED an Pin 8 eines Arduino UNO anschließen und den Potentiometer drehen, wird sich die Helligkeit der LED nicht ändern. Das liegt daran, dass Pin 8 keine PWM (Pulsweitenmodulation) unterstützt, die erforderlich ist, um die Helligkeitsstufen mit der Funktion `analogWrite()` anzupassen. Auf einem 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: Ein/Aus Schreibtischlampe

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 Einstellung des digitalen Pins 7 auf den Modus INPUT in Ihrem Arduino-Sketch, im Gegensatz zu INPUT_PULLUP, kann zu potenzieller 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 hohen oder niedrigen Spannung verbunden ist, wird er als "schwebend" bezeichnet. Ein schwebender Pin befindet sich nicht in einem stabilen hohen oder niedrigen Zustand; sein Zustand kann aufgrund von elektrischen Störungen oder Interferenzen aus der Umgebung schwanken. Diese Schwankungen können zu unvorhersehbaren Messwerten führen, wenn Sie versuchen, den Zustand des Pins über digitale Eingabefunktionen auszulesen, was zu fehlerhaften oder inkonsistenten Daten führt, die vom Mikrocontroller empfangen werden.

1. Wenn Pin 7 nur auf INPUT eingestellt ist, welche Anpassungen müssten am Schaltkreis vorgenommen werden?

Wenn Pin 7 Ihres Arduino auf den Modus INPUT eingestellt ist und Sie stabile und vorhersehbare Messwerte sicherstellen möchten, sollten Sie einen externen Pull-up-Widerstand in den Schaltkreis einfügen. Dazu verbinden Sie einen 10k Ω -Widerstand zwischen Pin 7 und der 5V-Stromversorgung des Arduino. Der Pull-up-Widerstand stellt sicher, dass der Eingangspin im hohen Zustand (logisches Level 1) ist, wenn kein anderes Eingangssignal vorhanden ist.

Lektion 11: Steuerung von LED-Arrays mit Potentiometer

Beantworten Sie die folgenden Fragen, bevor Sie mit der “Codeerstellung” fortfahren.

1. Schreiben Sie Ihren Pseudocode für das LED-Array.

Pseudocode dient als Programmskizze, die in einfacher Sprache verfasst ist, um das Verständnis zu erleichtern. Ihre Aufgabe ist es, Pseudocode für ein LED-Array zu erstellen, das auf einen Potentiometer reagiert. Wenn der Wert des Potentiometers steigt, leuchten mehr LEDs auf.

Hier ist ein vereinfachter Pseudocode, der die Steuerung eines LED-Arrays basierend auf den Eingaben eines Potentiometers umreißt:

```
Declare readValue variable.  
Setup.  
Declare 3 digital pin outputs.  
Main Loop.  
If the potentiometer's value is below 200, all LEDs should be off.  
If the value is between 200 and 600, the first LED should be on.  
If the value is between 600 and 1000, the first two LEDs should be on.  
If the value exceeds 1000, all LEDs should be on.  
Delay for a short time.
```

Beantworten Sie nach Abschluss der Lektion die folgende Frage.

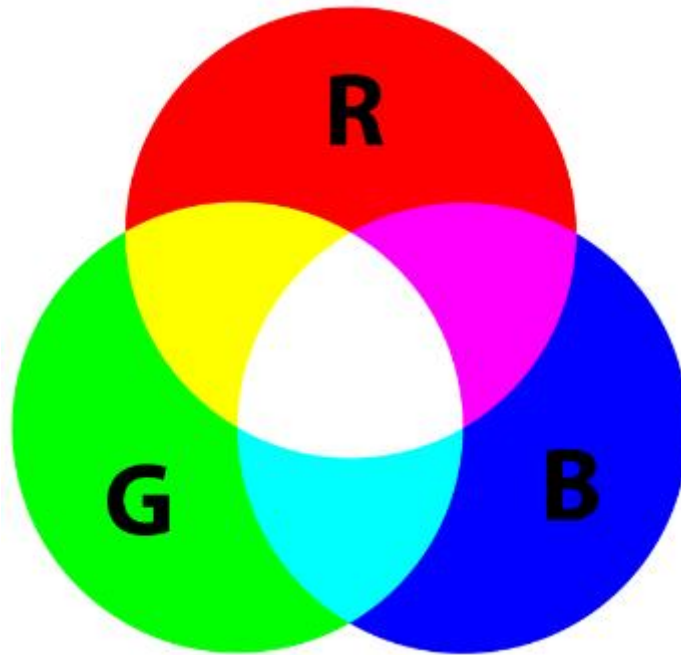
1. Im letzten Code bestimmen wir die Anzahl der LEDs, die leuchten, basierend auf dem Wert des Potentiometers. Wie können wir den Code ändern, damit sich beim Einschalten der LEDs deren Helligkeit entsprechend dem Potentiometer ändert?

Um Ihren Code so zu ändern, dass sich die Helligkeit der LEDs entsprechend dem Wert des Potentiometers ändert, können Sie die Funktion `analogWrite()` anstelle von `digitalWrite()` verwenden. Die Funktion `analogWrite()` ermöglicht es Ihnen, die Helligkeit der LEDs über PWM (Pulsweitenmodulation) zu steuern.

Lektion 12: Die Farben des Regenbogens

Füllen Sie diese Tabelle während der „Codeerstellung“ aus.

1. Wenn Sie andere Farben möchten, was sollten Sie tun? Sehen Sie sich das folgende Diagramm an und tragen Sie Ihre Ideen in Ihr Handbuch ein.



Farbe	Rot Pin	Grün Pin	Blau Pin
Rot	<i>HIGH</i>	<i>LOW</i>	<i>LOW</i>
Grün	<i>LOW</i>	<i>HIGH</i>	<i>LOW</i>
Blau	<i>LOW</i>	<i>LOW</i>	<i>HIGH</i>
Gelb	<i>HIGH</i>	<i>HIGH</i>	<i>LOW</i>
Pink	<i>HIGH</i>	<i>LOW</i>	<i>HIGH</i>
Cyan	<i>LOW</i>	<i>HIGH</i>	<i>HIGH</i>
Weiß	<i>HIGH</i>	<i>HIGH</i>	<i>HIGH</i>

Lektion 13: Das Spektrum des Sehens

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

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

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>

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

Farbe	Rot	Grün	Blau

Lektion 14: Zufällige Farben

Nach Abschluss der Lektion beantworten Sie bitte die folgenden Fragen.

1. Wenn Sie den Code von `randomSeed(analogRead(A0))` auf `randomSeed(0)` ändern, wie werden sich die Farben der RGB-LED ändern und warum?

Wenn Sie den Code von `randomSeed(analogRead(A0))` auf `randomSeed(0)` ändern, wird die Zufälligkeit der RGB-LED-Farben beeinflusst. Die Funktion `randomSeed(seed)` wird verwendet, um den Pseudo-Zufallszahlengenerator in Arduino zu initialisieren, was die Sequenz der von Funktionen wie `random()` erzeugten Zufallszahlen beeinflusst.

Die Verwendung von `analogRead(A0)` als Seed-Wert liest einen mehr oder weniger zufälligen Wert vom analogen Pin A0 aus, der typischerweise durch Umgebungsruschen und andere Faktoren beeinflusst wird. Dies führt dazu, dass bei jedem Programmstart unterschiedliche Seed-Werte entstehen. Das bedeutet, dass die Sequenz der Zufallszahlen (und somit auch die Farben) bei jedem Neustart des Arduinos variiert.

Im Gegensatz dazu initialisiert das Setzen des Seed-Werts auf einen festen Wert wie `randomSeed(0)` den Zufallszahlengenerator jedes Mal mit demselben Ausgangspunkt, wenn das Programm läuft. Dies führt dazu, dass dieselbe Sequenz von Zufallszahlen generiert wird, und folglich zeigt die RGB-LED bei jedem Neustart oder Einschalten des Arduinos dasselbe Farbmuster. Dadurch entfällt die Zufälligkeit bei den Farbwechseln der LED.

1. In welchen Situationen wird Zufälligkeit im Alltag verwendet, um Probleme zu lösen, abgesehen von der zufälligen Auswahl von Farben zur Dekoration und der Ziehung von Lotteriezahlen?

Zufälligkeit wird in verschiedenen Alltagssituationen eingesetzt, um Probleme zu lösen, darunter:

- **Brettspiele:** Würfeln, um die Züge zu bestimmen, wodurch jedes Spiel anders und fair bleibt.
- **Musik-Playlists:** Lieder zufällig abspielen, um das Hörerlebnis abwechslungsreich und unvorhersehbar zu gestalten.
- **Essenswahl:** Das zufällige Auswählen eines Restaurants oder Gerichts erleichtert die Entscheidungsfindung und macht es unterhaltsamer, wenn man unschlüssig ist, was man essen soll.
- **Sitzplatzverteilung:** Zufälliges Ziehen von Sitzplätzen bei Veranstaltungen, um Gäste zu durchmischen und soziale Interaktionen zu fördern.
- **Filmabende:** Zufälliges Auslosen eines Films, wenn die Vorlieben der Anwesenden unterschiedlich sind.

Lektion 15: Kühle oder warme Farben

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

1. Öffnen Sie Paint oder ein beliebiges Farbwahl-Tool, suchen Sie die Farben, die Sie für die wärmsten und kühlest halten, und notieren Sie deren RGB-Werte in Ihrem Handbuch.

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

Nach Abschluss der Lektion beantworten Sie bitte die folgende Frage.

1. Beachten Sie, 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 einwandfrei funktioniert.

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

Für $y = \text{map}(x, 1, 50, 50, -100)$, wenn x gleich 20 ist, was sollte y sein? Verwenden Sie die folgende Formel zur Berechnung.

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

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

Für $x=20$ und unter Verwendung der Mapping-Formel $y = \text{map}(x, 1, 50, 50, -100)$; beträgt der Wert von y ungefähr -8.16 .

Lektion 16: Temperaturalarm

Füllen Sie diese Tabelle während des “Schaltkreisaufbaus” aus.

1. Lesen Sie den Widerstandswert bei den unterschiedlichen Temperaturen ab und notieren Sie ihn in der untenstehenden Tabelle.

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

Nach Abschluss der Lektion beantworten Sie bitte die folgende Frage.

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

Dies ist die Standardmethode zur Umrechnung von Celsius in Fahrenheit und liefert Ihnen die Temperatur in Fahrenheit basierend auf dem Celsius-Wert, den Sie bereits aus Ihren Berechnungen haben.

$$F = C * 1.8 + 32$$

1. Fallen Ihnen andere Situationen oder Orte ein, an denen ein Temperaturüberwachungssystem wie das, das wir heute gebaut haben, nützlich sein könnte?

Temperaturüberwachungssysteme sind in vielen Alltagssituationen und verschiedenen Umgebungen anwendbar. Hier einige vereinfachte Beispiele:

- **Wohnkomfort:** Automatische Anpassung der Heizung oder Klimaanlage basierend auf Echtzeit-Temperaturmessungen, um den Wohnraum angenehm zu halten.
- **Gartenbau:** Überwachung der Temperaturen in Gewächshäusern, um sicherzustellen, dass die Pflanzen unter optimalen Bedingungen wachsen. Automatisierte Systeme können hinzugefügt werden, um die Temperaturen nach Bedarf anzupassen.
- **Lebensmittelsicherheit:** Überwachung der Temperaturen in Kühlschränken und Gefriergeräten, um sicherzustellen, dass Lebensmittel sicher verzehrt werden können, besonders in Restaurants oder während des Lebensmitteltransports.
- **Gesundheitswesen:** Überwachung und Protokollierung der Temperaturen in Lagerräumen für temperaturempfindliche Medikamente und Impfstoffe, um deren Wirksamkeit zu gewährleisten.

Lektion 17: Morsecode

Beantworten Sie die folgende Frage während des „Schaltkreisaufbaus“.

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

Wenn Sie die Kathode eines aktiven Buzzers direkt mit GND und die Anode mit 5V verbinden, gibt der Buzzer einen kontinuierlichen Ton ab. Dies geschieht, weil der interne Oszillator im Buzzer durch die 5V-Spannung aktiviert wird, wodurch er Ton erzeugt, bis der Stromkreis unterbrochen wird.

Nach Abschluss der Lektion beantworten Sie bitte die folgende Frage.

1. Verwenden Sie die bereitgestellte Morsecode-Tabelle, um einen Code zu schreiben, der die Nachricht „Hello“ sendet.

Im Morsecode wird „Hello“ basierend auf den Buchstaben wie folgt codiert:

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

Zusammengesetzt ergibt der Morsecode für „Hello“:

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

In der praktischen Kommunikation gibt es normalerweise eine längere Pause zwischen den Wörtern, um sie klar zu unterscheiden, aber da „Hello“ ein einzelnes Wort ist, ist der Code kontinuierlich, wobei nur Leerzeichen die einzelnen Buchstaben trennen.

Lektion 18: Lichtalarm

Füllen Sie diese Tabelle während des „Schaltkreisaufbaus“ aus.

1. Lesen Sie den Widerstandswert bei der aktuellen Umgebungsbeleuchtung ab und notieren Sie ihn in der untenstehenden Tabelle.

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

Nach Abschluss der Lektion beantworten Sie bitte die folgende Frage.

1. Gerissene Diebe könnten sich entscheiden, nachts zu stehlen, und wenn ein Gemälde verschwindet, könnte der Fotowiderstand möglicherweise keine Veränderung des Lichts erkennen und somit keinen Alarm auslösen. Was kann unternommen werden, um diesen Mangel zu beheben?

Eine Lösung besteht darin, eine Lichtquelle vor dem Gemälde zu installieren. Dies stellt nicht nur sicher, dass das Gemälde gut sichtbar und beleuchtet ist, sondern auch, dass jede Störung oder Entfernung des Gemäldes sofort den Alarm des Fotowiderstands auslöst, indem der erkannte Lichtpegel verändert wird.

Lektion 19: Rückwärtseinparksystem mit Alarm

Nach Abschluss der Lektion beantworten Sie bitte die folgende Frage.

1. Wenn Sie möchten, dass der vom Gerät erfasste Abstand genauer in Dezimalstellen angegeben wird, wie sollten Sie den Code anpassen?

Um die Distanzmessung genauer auf Dezimalstellen zu machen, können Sie die Berechnung in der Funktion `measureDistance` so anpassen, dass Gleitkommaarithmetik anstelle von Ganzzahlarithmetik verwendet wird. Dadurch kann das Ergebnis Dezimalwerte enthalten, was präzisere Messungen ermöglicht.

```
float distance = duration * 0.034 / 2.0;
```

Die Variable `distance` in der Funktion `measureDistance` wird in float geändert, um Dezimalwerte zu unterstützen. Die Berechnung der `distance` verwendet Gleitkommaarithmetik, indem durch `2.0` anstelle von `2` geteilt wird.

Lektion 20: Der Pomodoro-Timer

Beantworten Sie die folgende Frage während der "Coding-Erstellung - millis()"

1. Wenn `delay(100)` auf `delay(1000)` geändert wird, was passiert dann mit dem Programm? 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 piept der Buzzer etwa 100 Millisekunden lang alle 1000 Millisekunden (1 Sekunde, wie durch die Variable `interval` festgelegt), gefolgt von 900 Millisekunden Stille. Nach der Änderung piept der Buzzer 1000 Millisekunden lang alle 1000 Millisekunden und beginnt dann fast sofort wieder zu piepen, da das nächste Intervall fast sofort startet. Das Ändern der Verzögerung von 100 auf 1000 Millisekunden verwandelt den Buzzer von kurzen Pieptönen zu einem kontinuierlichen Ton, was störender und ungeeignet für den ursprünglichen Zweck wird.

Die Änderung von `delay(100)` auf `delay(1000)` in Ihrem Code führt dazu, dass der Buzzer für eine volle Sekunde anstatt nur für einen kurzen Piepton ertönt, da die Pausenzeit, während der der Buzzer eingeschaltet ist, verlängert wird. Dies resultiert in längeren Buzzer-Geräuschen und weniger häufigen Programmdurchläufen, wodurch das Programm möglicherweise weniger reaktionsfähig für andere Aufgaben während dieser Intervalle wird.

Nach Abschluss der Lektion beantworten Sie bitte die folgende Frage.

Denken Sie an andere Orte in Ihrem Leben, an denen Sie die Zeit „hören“ können. Listen Sie einige Beispiele auf und schreiben Sie diese in Ihr Handbuch!

Die Zeit zu „hören“ ist ein faszinierendes Konzept, und es gibt mehrere Alltagssituationen, in denen wir dies erleben können. Hier sind einige Beispiele, die Sie in Betracht ziehen

könnten, zu notieren:

- **Uhren:** Das Ticken von analogen Uhren oder die speziellen Pieptöne von Digitaluhren, die jede verstrichene Sekunde oder Minute signalisieren.Kitchen
- **Küchentimer:** Das Ticken und der abschließende Alarm 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.
- **Öffentliche Verkehrsdurchsagen:** Die Pieptöne oder Glockenspiele, die den Durchsagen an Bahnhöfen oder in Bussen vorausgehen und die bevorstehende Abfahrt oder Ankunft anzeigen.
- **Mikrowellenofen:** Der Piepton, wenn der Timer abläuft und signalisiert, dass der Heizvorgang abgeschlossen ist.
- **Fitness-Tracker oder Sportuhren:** Die Pieptöne oder Alarme, die das Ende einer festgelegten Zeit während des Trainings oder von Intervallen anzeigen.

Lektion 21: Sirenenton

Beantworten Sie die folgende Frage während des **“Schaltkreisaufbaus”**.

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

Wenn Sie die Kathode eines passiven Buzzers direkt mit GND und die Anode mit 5V verbinden, wird der passive Buzzer im Gegensatz zu einem aktiven Buzzer keinen Ton von sich geben, da er keinen eingebauten Oszillator besitzt.

Ein passiver Buzzer benötigt ein externes Signal, um Geräusche zu erzeugen. Üblicherweise muss er mit einer Rechteckwelle (wechselnde Spannung) bei der gewünschten Frequenz angesteuert werden, um hörbare Töne zu erzeugen.

Beantworten Sie die folgenden Fragen während der **“Code-Erstellung - Den passiven Buzzer ertönen lassen”**.

1. Wenn Sie den Code und die Schaltung auf die Pins 7 oder 8 umschalten, die keine PWM-Pins sind, wird der Buzzer trotzdem einen Ton erzeugen? Sie können dies testen und dann Ihre Antwort im Handbuch notieren.

Obwohl Pin 8 kein PWM-Pin ist, kann die `tone()`-Funktion dennoch eine präzise Rechteckwelle darauf erzeugen, was ausreicht, um einen passiven Buzzer anzusteuern und einen Ton zu erzeugen. Diese Flexibilität ermöglicht es, jeden digitalen Pin für die Tonausgabe zu verwenden, ohne auf PWM-fähige Pins beschränkt zu sein. Wenn Sie die Funktion `tone(pin, frequency)` aufrufen, konfiguriert Arduino einen Timer, um den Pin-Zustand (von HIGH zu LOW und wieder zurück zu HIGH) mit der angegebenen Frequenz zu wechseln und so eine Rechteckwelle zu erzeugen. Diese Rechteckwelle steuert den passiven Buzzer an, sodass er einen Ton mit der Frequenz der erzeugten Welle abgibt.

1. Um zu untersuchen, wie die Frequenz und die Dauer in tone(pin, frequency, duration) den Klang des Buzzers beeinflussen, modifizieren Sie bitte den Code unter zwei Bedingungen und tragen Sie die beobachteten Phänomene in Ihr Handbuch ein:

- Behalten Sie die Frequenz bei 1000 und erhöhen Sie schrittweise die Dauer von 100 über 500 bis 1000. Wie verändert sich der Klang des Buzzers und warum?
- 100 ms Dauer: Der Ton ist ein kurzes Piepen.
- 500 ms Dauer: Der Ton ist ein längeres Piepen, deutlich hörbar und dauert eine halbe Sekunde.
- 1000 ms Dauer: Der Ton ist noch länger und dauert eine volle Sekunde.

Wenn Sie die Dauer erhöhen, hält der vom Buzzer erzeugte Ton länger an. Die Tonhöhe oder Frequenz des Tons bleibt konstant (da sie auf 1000 Hz eingestellt ist), was bedeutet, dass sich die „Note“ des Tons nicht ändert, aber die Zeit, in der Sie ihn hören, verlängert wird. Dies ist nützlich, um verschiedene Alarmdauern zu signalisieren, bei denen die Dringlichkeit oder der Typ des Alarms durch die Länge des Tons unterschieden werden kann.

- Behalten Sie die Dauer bei 100 und erhöhen Sie schrittweise die Frequenz von 1000 über 2000 bis 5000. Wie verändert sich der Klang des Buzzers und warum?
- 1000 Hz Frequenz: Der Ton ist ein mittelhohes Piepen.
- 2000 Hz Frequenz: Der Ton hat eine höhere Tonlage im Vergleich zu 1000 Hz.
- 5000 Hz Frequenz: Der Ton ist viel höher und wird wahrscheinlich als schriller und möglicherweise unangenehm in der Nähe wahrgenommen.

Die Erhöhung der Frequenz bei gleichbleibender Dauer führt zu einer Änderung der Tonhöhe. Höhere Frequenzen erzeugen höher klingende Töne. Dieses Prinzip ist nützlich, um zwischen verschiedenen Arten von Benachrichtigungen oder Signalen basierend auf ihrer Dringlichkeit oder Wichtigkeit zu unterscheiden, wobei höhere Tonhöhen oft für dringlichere Alarmer verwendet werden.

Lektion 22: Spielen Sie „Twinkle, Twinkle, Little Star“

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

1. Sie können auch Operationen an den Elementen im Array durchführen, wie zum Beispiel `Serial.println(melody[i] * 1.3)`, ändern. Welche Daten erhalten Sie und warum?

Die Zahl 1,3 ist eine Gleitkommazahl. Wenn eine Ganzzahl aus dem Array `melody` (vom Typ `int`) mit 1,3 multipliziert wird, wird das Ergebnis der Operation automatisch in eine Gleitkommazahl (`float`) umgewandelt.

Für jede Notenfrequenz in diesem Array ergibt das Multiplizieren mit 1,3 und das anschließende Ausgeben des Ergebnisses:

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

Nach Abschluss der Lektion beantworten Sie bitte die folgende Frage.

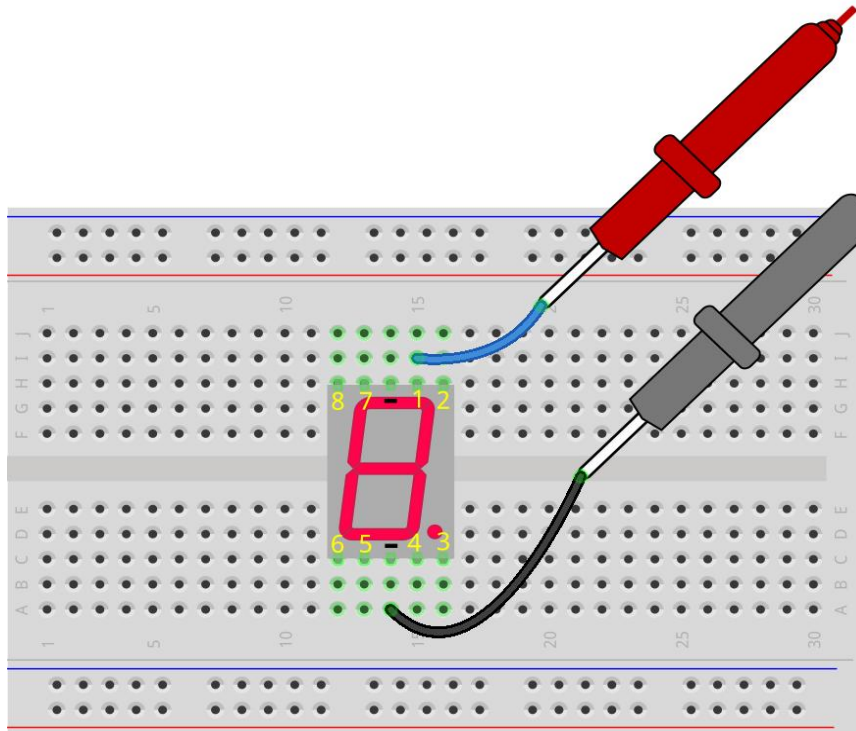
1. Wenn Sie den passiven Buzzer in der Schaltung durch einen aktiven Buzzer ersetzen, können Sie dann „Twinkle Twinkle Little Star“ korrekt abspielen? Warum?

Wenn Sie den passiven Buzzer durch einen aktiven Buzzer ersetzen, um „Twinkle Twinkle Little Star“ abzuspielen, wird es nicht wie beabsichtigt funktionieren. Aktive Buzzer können nur einen einzelnen Ton erzeugen, da sie einen eingebauten Oszillator haben. Daher können Sie die Tonhöhe nicht steuern, um die Melodie korrekt abzuspielen; Sie würden lediglich ein sich wiederholendes Piepen im Rhythmus des Liedes hören, aber nicht die tatsächlichen Noten.

Lektion 23: Cyber-Würfel

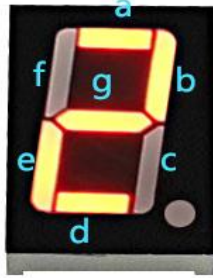
Beantworten Sie die folgenden Fragen während des Abschnitts "Das 7-Segment-Display verstehen".

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



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	decimal point	Der Punkt

1. Aus den oben durchgeführten Tests ist bekannt, dass das Display im Kit eine gemeinsame Kathode hat. Das bedeutet, dass der gemeinsame Pin nur mit GND verbunden werden muss und die anderen Pins mit einer hohen Spannung versorgt werden, um die entsprechenden Segmente zu beleuchten. Wenn das Display die Zahl 2 anzeigen soll, welche Pins müssen dann mit einer hohen Spannung versorgt werden? Warum?



Um die Zahl 2 anzuzeigen, müssen die Segmente a, b, d, e und g aktiviert (mit hoher Spannung versorgt) werden, da diese die Zahl 2 auf dem Display bilden. Die Segmente f, c und dp (Dezimalpunkt, falls vorhanden) sollten abgeschaltet (mit niedriger Spannung) bleiben, da sie nicht zur Anzeige der Zahl 2 gehören.

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

Lektion 24: Fließendes Licht mit 74HC595

Beantworten Sie die folgende Frage während der **“Code-Erstellung - LEDs zum Leuchten bringen”**.

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

Wenn Sie `MSBFIRST` in `LSBFIRST` ändern, wird die Reihenfolge der Bits umgekehrt, und das Byte wird beginnend mit dem am wenigsten signifikanten Bit (dem ganz rechten) ausgegeben. Wenn Sie das Schieberegister zur Steuerung von LEDs verwenden, wird sich die Reihenfolge ändern, in der die LEDs leuchten. Anstatt in der ursprünglich programmierten Sequenz aufleuchten, werden sie in umgekehrter Reihenfolge leuchten.

Beantworten Sie nach Abschluss der Lektion die folgende Frage.

1. Wenn wir drei LEDs gleichzeitig leuchten lassen und sie den Anschein erwecken wollen, zu „fließen“, wie sollten die Elemente des `dataArray[]` Arrays geändert werden?

Sie würden mit den ersten drei leuchtenden LEDs beginnen und dann in jedem nachfolgenden Muster eine LED nach rechts verschieben, bis die letzten drei LEDs leuchten. So könnten Sie diese Muster in binär 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 25 Zahl anzeigen

Füllen Sie diese Tabelle während „Binärzahlen für die Ziffern 0 bis 9“ aus.

1. Jetzt, da wir die binären Darstellungen für die Ziffern 0 und 2 kennen, tragen Sie bitte die binären Zahlen für die verbleibenden Ziffern in die folgende Tabelle ein.

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 „Binärkonvertierung“ aus.

1. Bitte wandeln Sie die binären Zahlen, die die Ziffern 0 bis 9 darstellen, mit einem Taschenrechner in Dezimal- und Hexadezimalzahlen um und füllen Sie die Tabelle aus. Dies wird Ihnen einen schnellen Referenzleitfaden für Basisumwandlungen bieten.

Zahl	Binär	Dezimal	Hexadezimal
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