SPORT-ROBOTER

Mikrocontroller werden häufig bei Robotern verwendet, um zu steuern, wie sich ein Roboter bewegt und arbeitet. In dieser Lektion werdet ihr einen Servomotor verwenden, um einen einfachen Sportroboter zu bauen. Ihr programmiert den Roboter so, dass er von euch gesteuert wird, um einen Ball zu schlagen, zu treten oder zu werfen. Schließlich führt ihr ein Experiment durch, um die athletischen Fähigkeiten eures Roboters zu testen.

VORAUSSICHTLICHE ZEIT **90-135 min**

SIE WERDEN ETWAS DARÜBER LERNEN

BIBLIOTEKENS FUNKTION, MAGNETISCHES FELD, KONDENSATOREN, KONSTANTE, FUNKTION MAP(), DIE LEISTUNG MESSEN

Übersicht

In den Lektionen 1-5 habt ihr Schaltungen gebaut, die einen Mikrocontroller zur Steuerung von LED-Leuchten verwenden. LED-Schaltungen sind für eine Reihe verschiedener Anwendungen nützlich, aber Mikrocontroller, wie der auf eurem Arduino UNO R3 Board, sind neben LEDs auch für Steuerungen nützlich. Mikrocontroller werden häufig bei Robotern verwendet, um zu steuern, wie sich ein Roboter bewegt und arbeitet. In dieser Lektion werdet ihr einen Servomotor verwenden, um einen einfachen Sportroboter zu bauen. Ihr programmiert den Roboter so, dass er von euch gesteuert wird, um einen Ball zu schlagen, zu treten oder zu werfen. Schließlich führt ihr ein Experiment durch, um die athletischen Fähigkeiten eures Roboters zu testen.

TEACHER NOTES

Mit dieser Lektion beginnt die zweite Hälfte des Kurses. Die erste Hälfte des Kurses konzentrierte sich auf einige der grundlegenden Konzepte und Prinzipien der Programmierung von Arduino Boards zur Steuerung

elektrischer Schaltkreise. Die zweite Hälfte des Kurses befasst sich mit einigen der verschiedenen Sensoren und Aktoren, die an Arduino Boards angeschlossen werden können. Diese Geräte erweitern die Einsatzmöglichkeiten und Anwendungen von Arduino Boards erheblich.

In dieser Lektion lernen die Schülerinnen und Schüler, einen Servomotor mit ihrem Arduino UNO R3 Board zu steuern. Die Schülerinnen und Schüler bauen eine Schaltung, mit der die Position des Servos durch ein Potentiometer gesteuert werden kann. Es werden mehrere neue Kodierungskonzepte vorgestellt, wie z.B. Codebibliotheken, Klassen und Objekte, Konstanten und die Map-Funktion zur Skalierung eines Datenbereichs auf einen anderen. Die Schülerinnen und Schüler entscheiden, welche Art von Sportroboter sie herstellen wollen, und erstellen eine Befestigung für ihren Servo, mit der sie einen Ball schlagen, treten oder werfen können. Die Schülerinnen und Schüler verwenden dann das Potentiometer, um den Drehwinkel für ihre Befestigung einzustellen, und fügen dem Schaltkreis einen Tastschalter hinzu, damit der Roboter den Schlag, den Kick oder den Wurf ausführen kann. Schließlich führen die Schülerinnen und Schüler mit ihrem Roboter ein Experiment durch, bei dem sie untersuchen, wie sich die Einstellung des Drehwinkels auf die Entfernung auswirkt, die der Ball zurücklegt.

Das Servo wird sich höchstwahrscheinlich nicht schnell genug bewegen, um den Ball wie ein Geschoss zu werfen. Wenn die Schülerinnen und Schüler einen Wurfroboter bauen wollen, sollten sie sich darüber im Klaren sein, dass das Wort "Wurf" in der Lektion im Zusammenhang mit dem Schieben des Balls verwendet wird, so dass er auf dem Tisch oder Boden rollt.

ZEITAUFWAND FÜR DIE LEKTION

Die hier aufgeführten Zeiten sind nur Richtwerte und müssen möglicherweise an die jeweilige Lernsituation angepasst werden.

Übersicht und Fachbegriffe	3 minuten
Aufladen eines Kondensators	10 minuten
Sport-Roboter	
Benötigte Materialien	2 minuten
Aufbau der Schaltung	10 minuten
Code-Erstellung - Servo-Steuerung	15 minuten
Code-Erstellung - Serielle Kommunikation	10 minuten
Den Roboter bauen	8 minuten
Testen und ändern	

Benötigte Materialien	2 minuten
Ändern des Schaltkreises	5 minuten
Ändern des Codes	10 minuten
Testen des Schaltkreises	15 minuten
Gesamte Lektion	**90 minuten**

Wenn Sie diese Lektion in zwei Unterrichtsstunden absolvieren, sollten die Schülerinnen und Schüler den Abschnitt Code-Erstellung - Serielle Kommunikation bis zum Ende der ersten Unterrichtsstunde abgeschlossen haben.

LERNZIELE

- Die Beziehung zwischen elektrischen und magnetischen Feldern untersuchen.
- Bestimmen, wie ein Standardservo funktioniert.
- Die Verwendung von Kondensatoren erforschen und wie sie Ladung speichern.
- Spannung messen, wenn ein Kondensator entladen wird.
- Bei der Kodierung zwischen Konstanten und Variablen unterscheiden.
- Die Rolle verstehen, die Bibliotheken und Klassen bei der Erweiterung der Fähigkeiten der Software und des Arduino UNO R3 Boards spielen.
- Die Funktion map() in der Arduino IDE auf die Skalierung beziehen.
- Ein Experiment durchführen, um die Leistung eines Roboters zu messen.

FURTHER NOTES

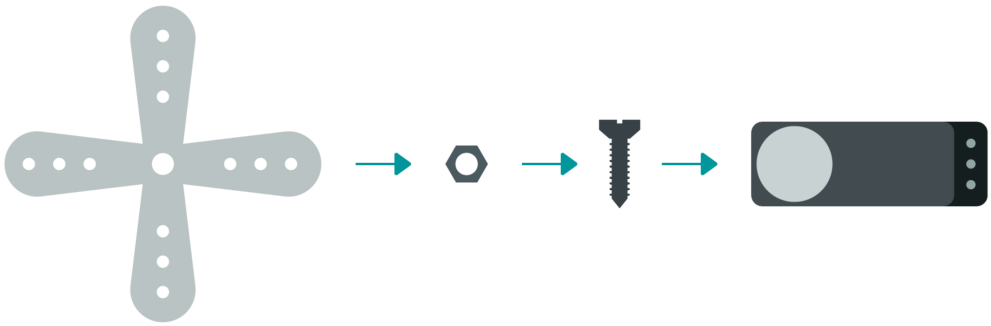
Wir müssen die folgende zusätzliche Materialien zur Verfügung stellen:

- Pappreste
- Scheren
- Klebeband (durchsichtiges Klebeband funktioniert gut)
- Ein kleiner Ball wie z.B. ein Ping-Pong-Ball oder Golfball
- Ein Messgerät wie ein Zollstock oder ein Maßband

SERVOHORNAUFSATZ

Wenn wir zum ersten Mal an dieser Lektion teilnehmen, müssen wir die mitgelieferten Servohörner an die Servos anschließen. Obwohl wir diese Aufgabe überlassen können, wird empfohlen, dass wir dies ausführen, um zu verhindern, dass die Schrauben verloren gehen oder die Servos beschädigt werden.

Jeder Bausatz enthält einen Servo, der mit einer kleinen Tasche mit Servohörnern geliefert wird. Suchen wir das Servohorn, das wie ein Kreuz aussieht, und setzen wir es auf die Abtriebswelle des Servos. Befestigen wir die Schraube mit einem kleinen Kreuzschlitzschraubendreher durch das Loch im Servohorn am Servo.



Fachbegriffe

- **Aktor –** eine Vorrichtung, die von einem Mikrocontroller gesteuert wird, um eine bestimmte Aufgabe auszuführen
- **Kapazität -** die Einheit für die Fähigkeit eines Bauteils, elektrische Ladung zu speichern
- **Kondensator -** eine elektrische Komponente, die eine elektrische Ladung speichert
- **Klasse -** eine Vorlage, die ein Objekt beschreibt; sie enthält Variablen, die Eigenschaften eines Objekts enthalten, und Methoden, die beschreiben, was das Objekt tun kann.
- **Konstante -** beim Programmieren ein Wert, der sich während der Programmausführung nie ändert
- **Dielektrikum -** das Material in einem Kondensator, das eine Platte von der anderen isoliert
- **Bibliothek -** bei der Programmierung eine Gruppe von Funktionen oder Befehlen, die importiert werden müssen; die Funktionen und Befehle bieten zusätzliche Funktionalität und werden häufig mit Sensoren und Aktoren verwendet.
- **Methode -** eine Prozedur, die das Verhalten eines Objekts in einer Klasse definiert
- **Objekt -** ein spezifisches Beispiel, das zu einer Klasse gehört
- **Sensor -** ein Gerät, das dazu dient, etwas anzuzeigen oder zu messen, und das diese Messung in ein elektrisches Signal umwandelt, das ein Mikrocontroller lesen kann; Sensoren liefern Input für das Arduino UNO R3 Board.
- **Servo -** ein Motortyp, der einen eigenen Treiber enthält und zur präzisen Positionierung in beide Richtungen verwendet wird.

TEACHER NOTES

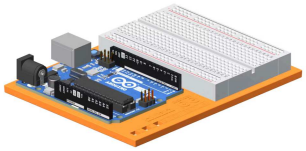
Es gibt viele Übungsmöglichkeiten zu den Fachbegriffen für die Schülerinnen und Schüler. Allerdings ist die Zeit dafür im 90-minütigen Rahmen für diese

Lektion nicht berücksichtigt. Diese Übungen können vielleicht in den normalen Unterricht einfließen oder in Eigenarbeit erledigt werden.

Aufladen eines Kondensators

In dieser Aktivität werdet ihr untersuchen, wie ein Kondensator geladen und entladen werden kann. Ihr werdet eine Batterie zum Laden eines Kondensators verwenden und mit dem Multimeter messen, wie die Kondensatorspannung beim Entladen abfällt.

Benötigtes Material



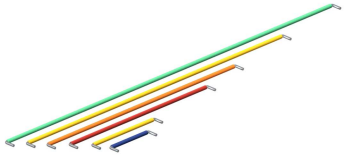
1 ARDUINO PROJEKT BOARD

1 100 MIKROFARAD KONDENSATOR

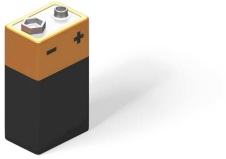
1 10K Ω WIDERSTAND

1 TASTSCHALTER

1 MULTIMETER MIT PRÜFLEITUNGEN



1 STECKVERBINDER



1 9V-BATTERIE

1 9V-BATTERIEANSCHLUSS

TEACHER NOTES

Die Farbbänder des 10K-Ohm-Widerstandes sind:

- **4 Bänder:** braun, schwarz, orange, gold
- **5 Bänder:** braun, schwarz, schwarz, rot, gold

Prüfung des Kondensators

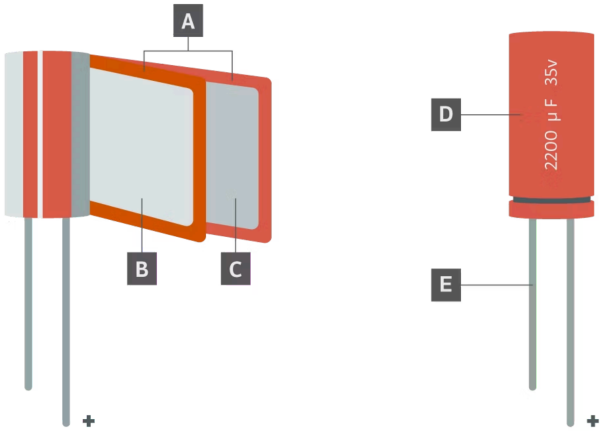
Kondensatoren

Ein **Kondensator** ist ein elektrisches Bauteil, das wie eine Batterie eine elektrische Ladung speichert. Allerdings speichern Kondensatoren Ladung auf eine ganz andere Weise als Batterien. Batterien nutzen eine chemische Reaktion, um an einem Pol Elektronen zu erzeugen und am anderen Pol Elektronen zu absorbieren. Kondensatoren produzieren keine Elektronen wie Batterien. Kondensatoren sind Speichereinheiten, in denen Elektronen gespeichert und über den gleichen Anschluss wieder abgegeben werden, über den sie hereinkamen.

Kondensatoren haben zwei Anschlüsse. Einige Kondensatoren, wie die in eurem Bausatz, sind polarisiert. Sie haben eine Anode und eine Kathode. Bei polarisierten Kondensatoren fließen die Elektronen nur in eine Richtung. Daher werden polarisierte Kondensatoren im Allgemeinen bei Gleichstromanwendungen verwendet, bei denen der Strom in eine Richtung fließt. Bei nicht polarisierten Kondensatoren können Elektronen in beide Richtungen fließen. Dadurch eignen sie sich hervorragend für Wechselstromanwendungen, bei denen der Strom in wechselnde Richtungen fließt. Sie können jedoch auch in Gleichstromkreisen verwendet werden.

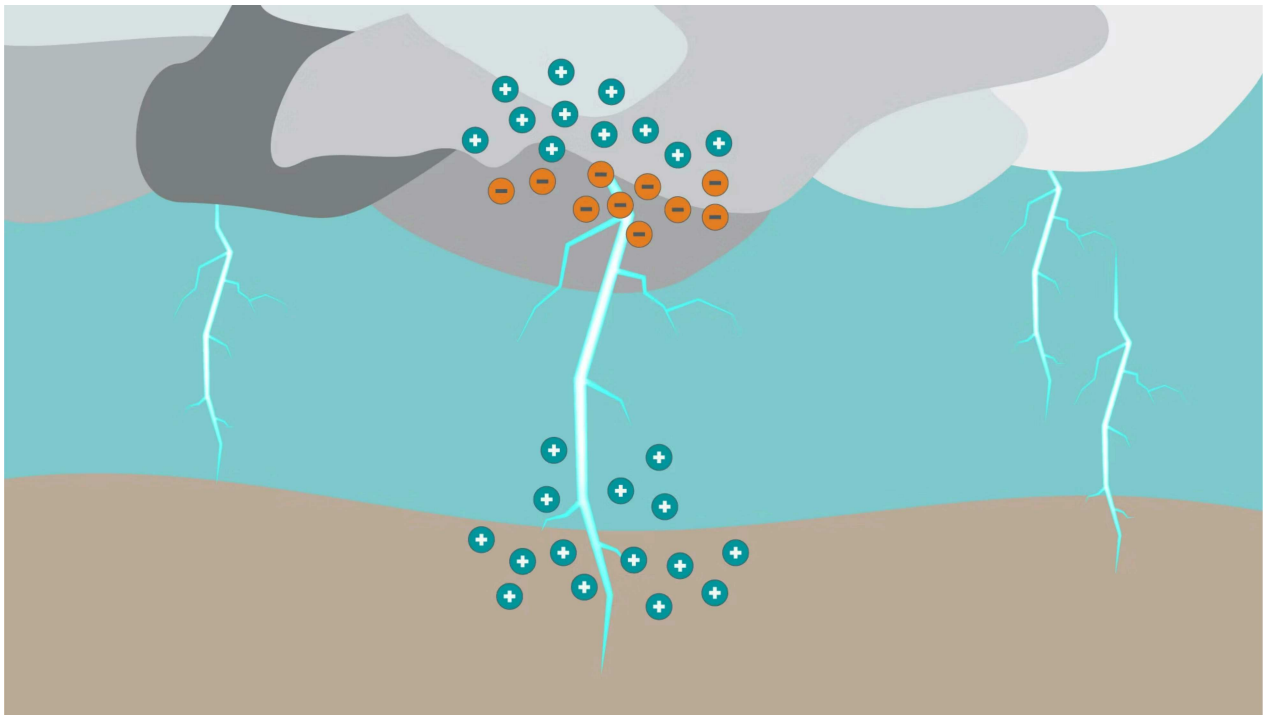
> **Hinweis:** Bei polarisierten Kondensatoren haben die Kathoden üblicherweise einen kürzeren Draht und sind normalerweise mit einem Pfeil, einem Streifen oder einem negativen Vorzeichen gekennzeichnet.

Im Inneren eines Kondensators sind die Anschlüsse mit leitenden Metallplatten verbunden, die durch einen Isolator getrennt sind. Der Isolator, der die Platten voneinander trennt, wird als **Dielektrikum** bezeichnet und kann aus Keramik, Porzellan, Glas und sogar Luft bestehen. Das Dielektrikum ist das Material, das den Kondensatoren ein breites Spektrum an Typen und Verwendungsmöglichkeiten verleiht.

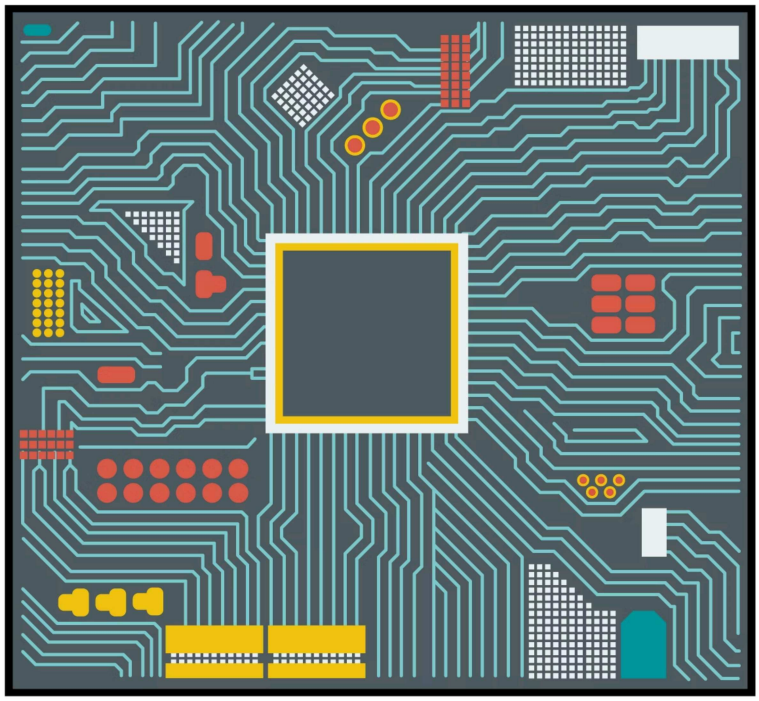


A) Trennzeichen B) Anodenfolie (+) C) Kathodenfolie (-) D) Behälter E) Terminal

In gewisser Weise kann man sich einen Gewitterblitz wie einen Kondensator vorstellen. Die Wolken sind eine leitende Platte und der Boden ist die andere. Sie sind durch die Luft getrennt, die als Dielektrikum wirkt. Wenn sich genügend Ladung in den Wolken aufbaut, überbrückt der Blitz den Luftspalt zwischen den Wolken und der Erde und entlädt den Kondensator.

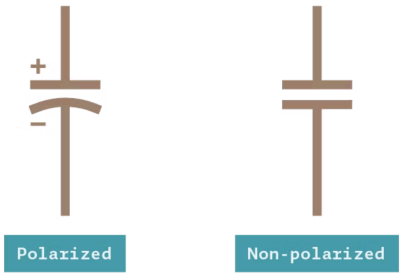


Blitze führen zu einem großen Unterschied zwischen Batterien und Kondensatoren. Kondensatoren können alle ihre gespeicherten Elektronen sehr schnell entladen, während sich Batterien langsam entladen. Dadurch sind die Einsatzmöglichkeiten von Batterien und Kondensatoren sehr unterschiedlich. Batterien sind gut in der Lage, Strom über eine gewisse Zeit zu liefern, während Kondensatoren gut in der Lage sind, schnelle Stromänderungen zu bewältigen.



Die **Kapazität** ist ein Maß dafür, wie viel Ladung ein Kondensator sammeln und speichern kann. Sie wird in Einheiten gemessen, die Farad genannt werden und nach Michael Faraday benannt sind. (Ihr könnt über Michael Faraday im Spotlight über Erfindungen in dieser Lektion lesen.) Im Allgemeinen gilt: je größer die physikalische Größe eines Kondensators, desto mehr Kapazität hat er. Zum Beispiel wäre ein Ein-Farad-Kondensator ziemlich groß - vielleicht so groß wie eine Zwei-Liter-Flasche Limo. Er könnte genug Ladung aufnehmen, um euch möglicherweise unter Strom zu setzen. Aber keine Sorge, die Kondensatoren, die ihr verwenden werdet, liegen im Mikrofarad-Bereich. Das ist ein Millionstel eines Farad. Diese Kondensatoren halten viel weniger Ladung und sind viel sicherer.

Die Schaltsymbole für Kondensatoren sind unten dargestellt:

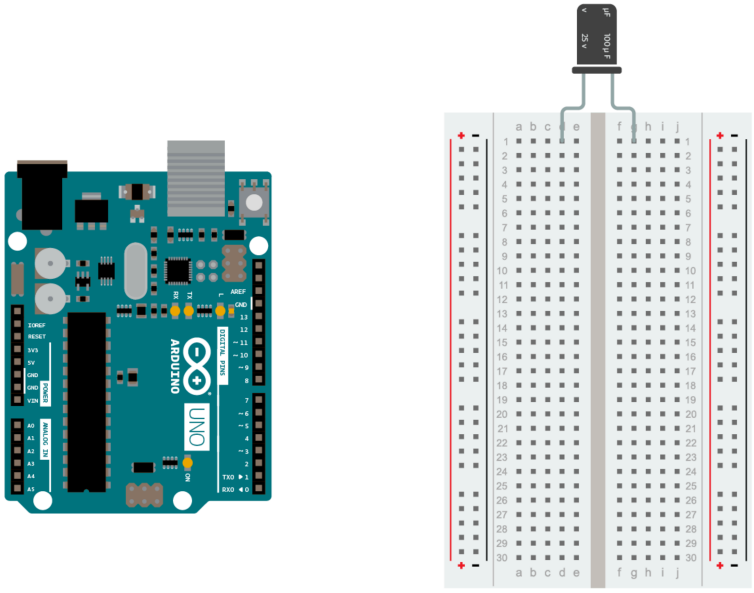


Step 1

Sucht einen 100 Mikrofarad (uf)-Kondensator.

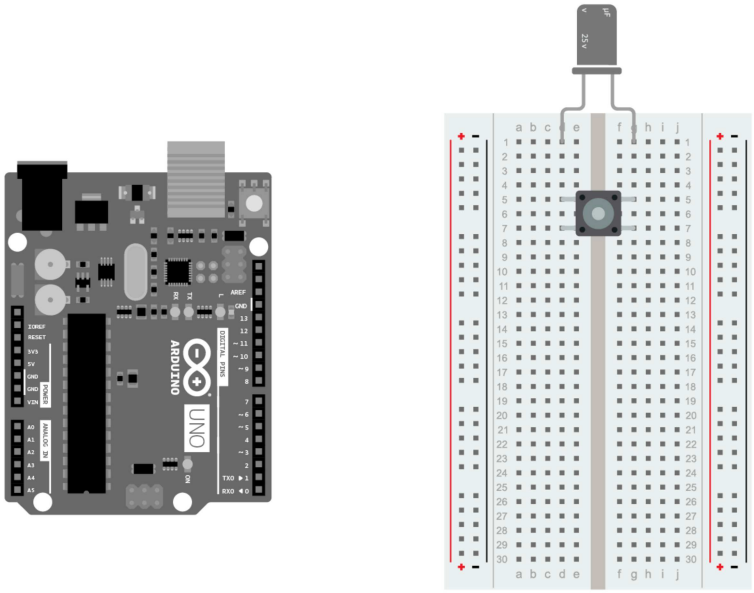
Step 2

Befestigt den Kondensator so an der Lochrasterplatte, dass er die Lücke in der Lochrasterplatte überbrückt. Dadurch werden Anode und Kathode voneinander getrennt, so dass sie nicht miteinander verbunden sind.



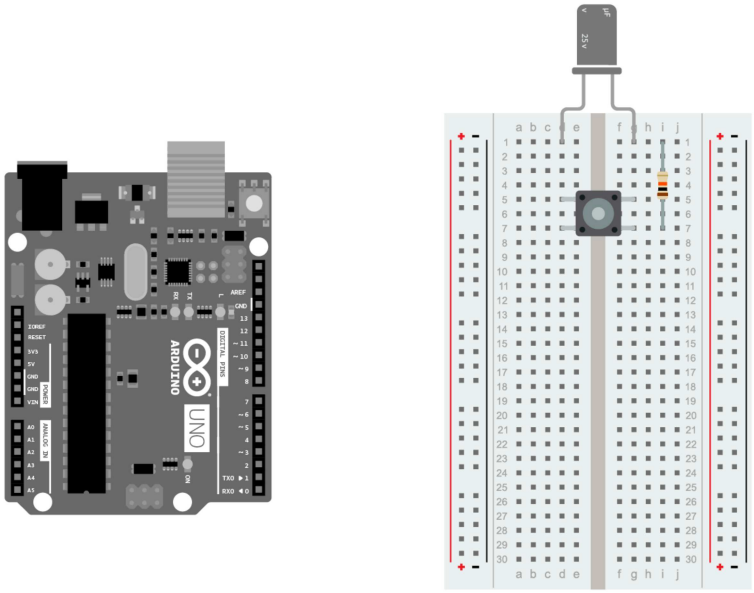
Step 3

Befestigt einen Tastschalter am Steckbrett, so dass er die Lücke des Steckbretts überspannt. Bringt den Schalter einige Reihen vom Kondensator entfernt an.



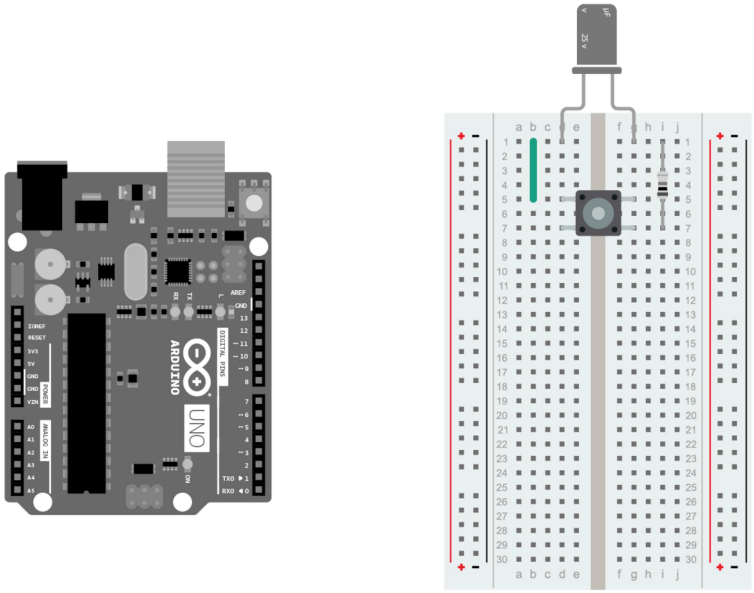
Step 4

Verwendet einen 10K-Ohm-Widerstand, um die Kathode des Kondensators mit dem rechten unteren Pin des Schalters zu verbinden.



Step 5

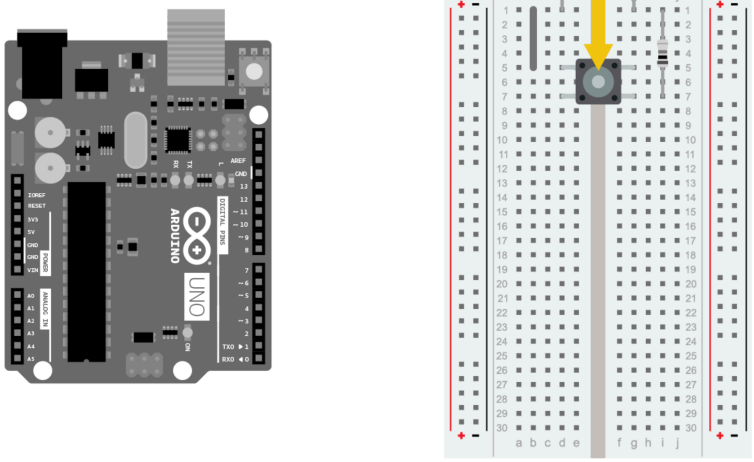
Verwendet einen kurzen Steckverbinder, um den oberen linken Stift des Schalters mit der Anode des Kondensators zu verbinden.



Step 6

Ihr habt gerade eine einfache Schaltung erstellt. Wenn die Taste gedrückt wird, wirkt der Widerstand wie eine Last, die den Strom aus dem Kondensator verbraucht. Haltet

den Knopf etwa 10 Sekunden lang gedrückt. Dadurch wird der Stromkreis geschlossen und der Kondensator entladen.

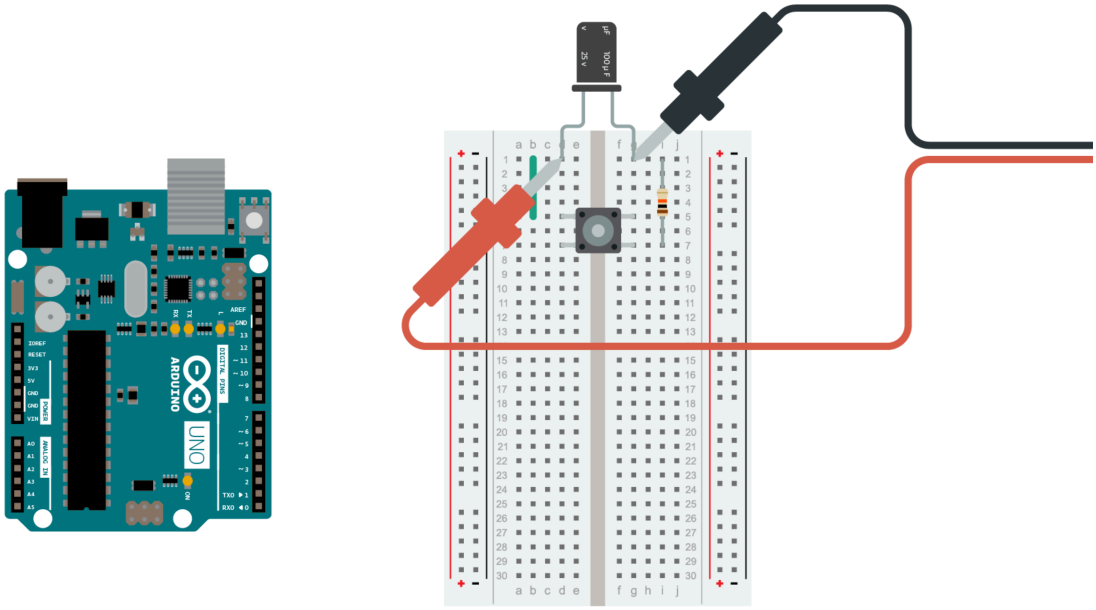


Step 7

Dreht das Multimeter auf die Einstellung 20V DC.

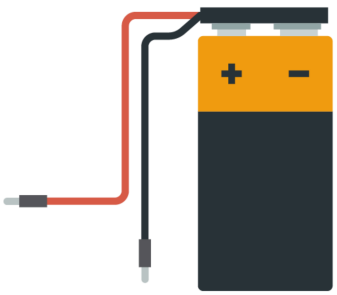
Step 8

Messt die Spannung zwischen der Anode und der Kathode des Kondensators. Berührt mit der roten Messleitung vom Multimeter die Anode und mit der schwarzen Messleitung die Kathode. Wenn euer Kondensator vollständig entladen ist, sollte das Multimeter Null oder fast Null Volt anzeigen.



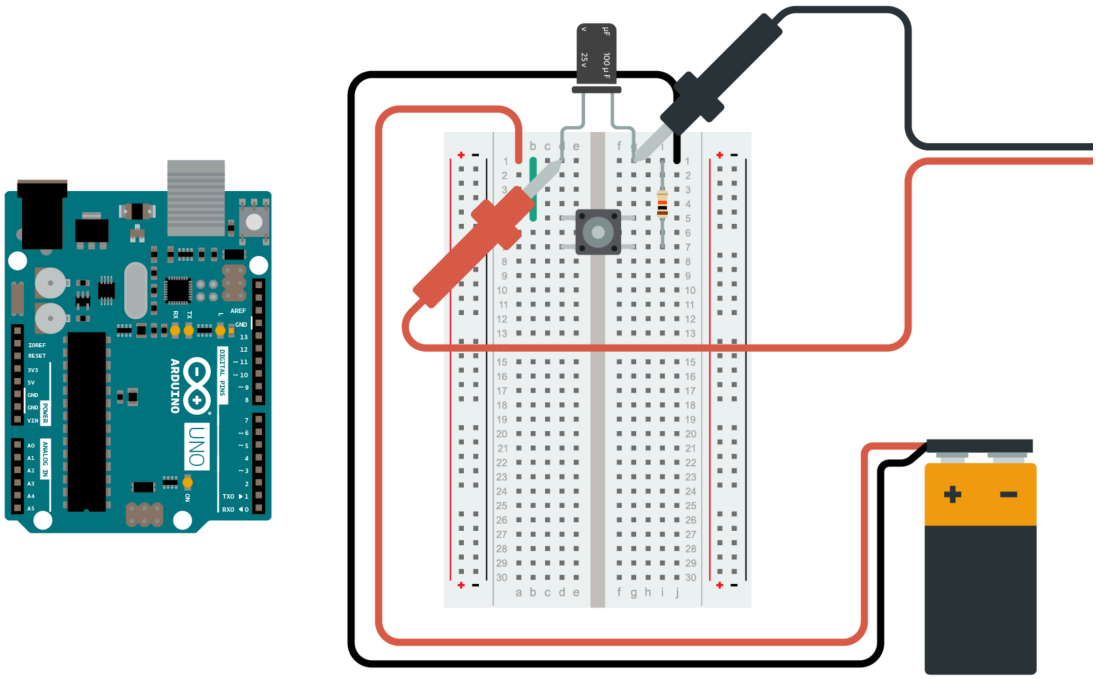
Step 9

Schließt den Batterieschnappverschluss an die Batterie an.



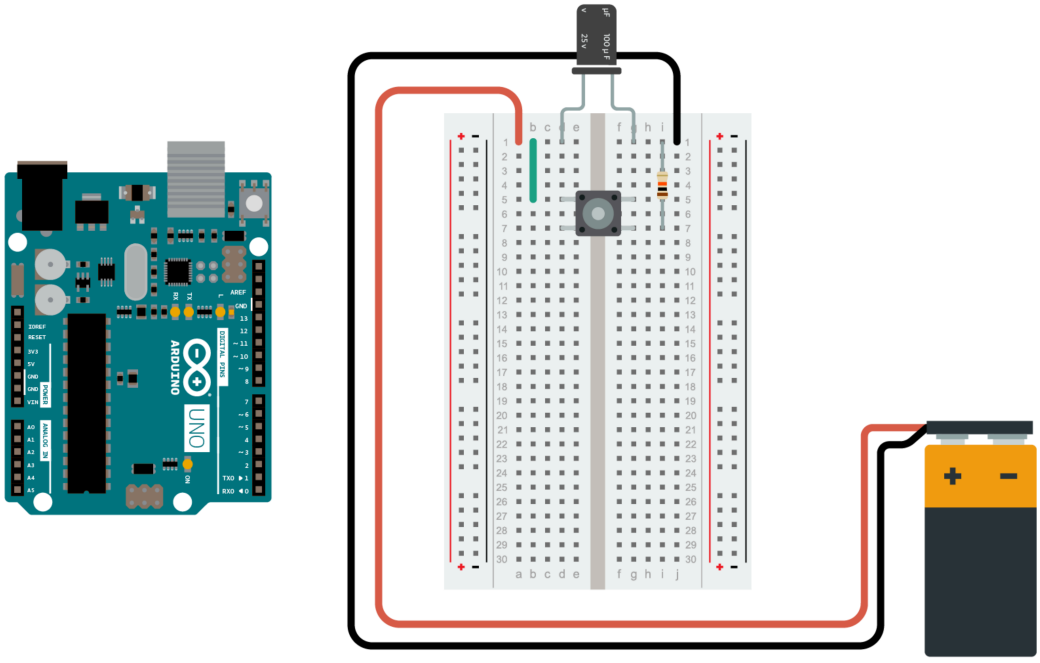
Step 10

Während ihr noch die Spannung über dem Kondensator messt, ladet ihr den Kondensator, indem ihr die Batterie an seine Anschlüsse anschließt. Befestigt die negative Leitung der Batterie (schwarzer Draht) an der Lochrasterplatine, so dass sie mit der Kathode des Kondensators verbunden ist. Befestigen Sie das Pluskabel der Batterie (roter Draht) an dem Steckbrett, so dass es mit der Anode des Kondensators verbunden ist. Beachtet den Spannungswert auf dem Multimeter. Der Kondensator sollte mit etwa neun Volt geladen werden.



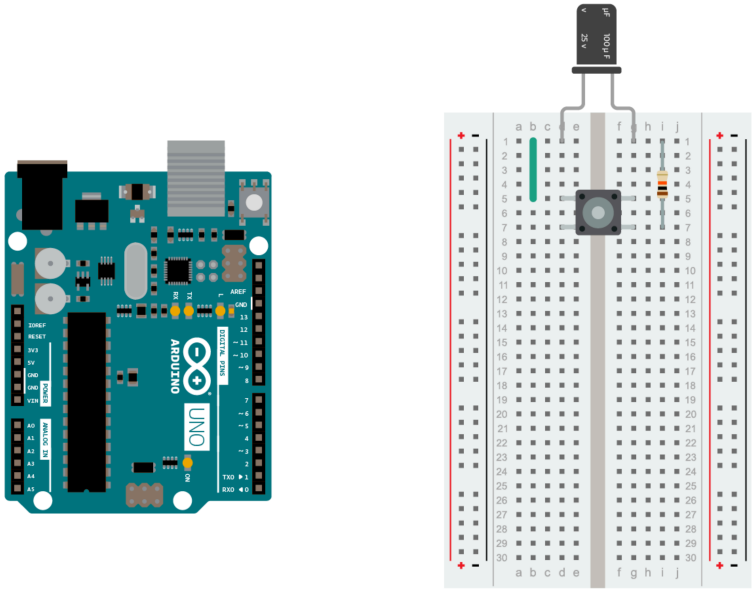
Step 11

Entfernt die Messleitungen des Multimeters vom Kondensator.



Step 12

Trennt die Batterie vom Steckbrett und schiebt sie aus dem Weg. Dadurch bleibt der Kondensator als einzige Stromquelle für die Schaltung übrig.



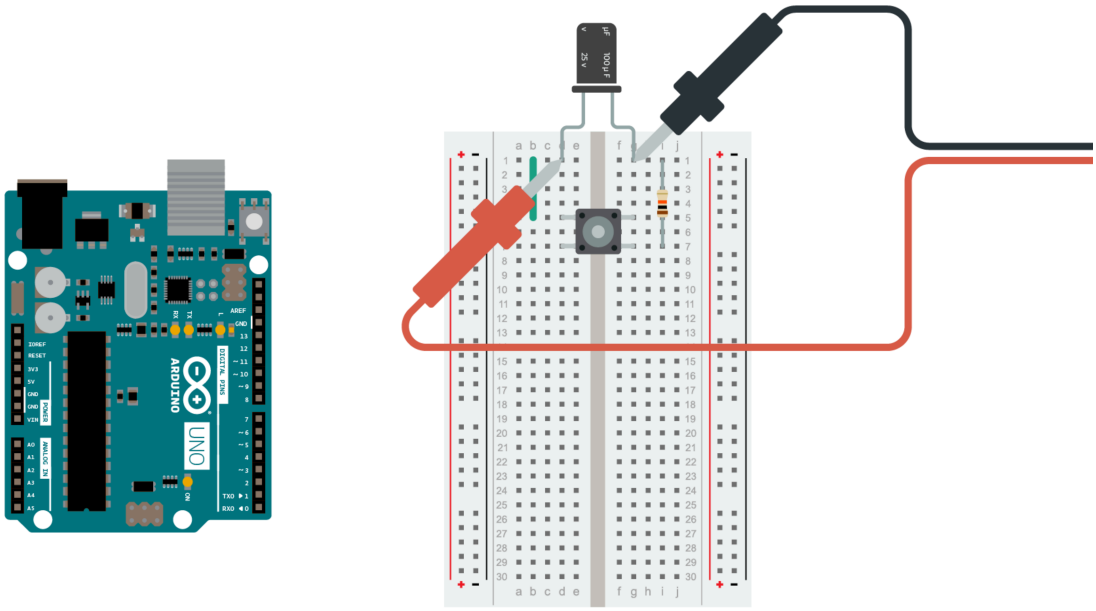
Step 13

Messt die Spannung über dem Kondensator. Berührt mit der roten Messleitung vom Multimeter die Anode und mit der schwarzen Messleitung die Kathode. Das

Multimeter sollte bei fast neun Volt, der Spannung der Batterie, beginnen. Ihr solltet jedoch bemerken, dass die Spannung abnimmt, wenn der Kondensator seine gespeicherte Energie entlädt.

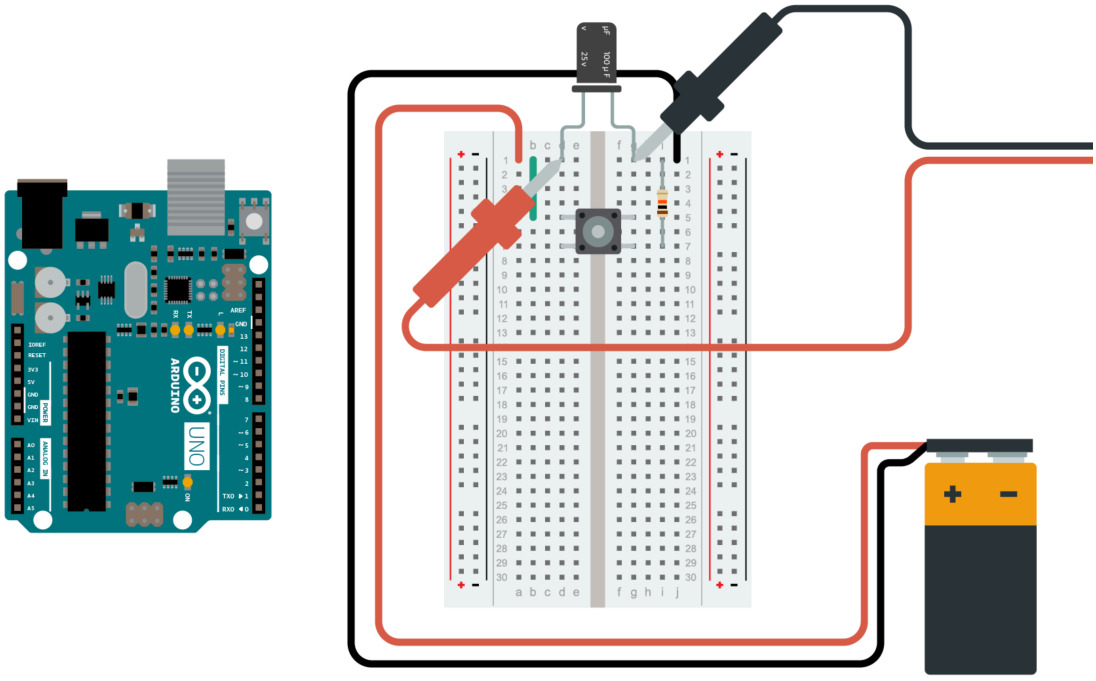
Der Kondensator entlädt sich, weil das Multimeter mit einem Widerstand von einem Megaohm (1.000.000 Ohm) als Last auf die Schaltung wirkt. Bei einem so hohen Widerstand entlädt sich der Kondensator langsam (etwa 500 Sekunden).

Bei dieser Aktivität braucht ihr nicht darauf zu warten, dass sich der Kondensator vollständig entlädt. Es ist wichtiger, die Entladungsgeschwindigkeit des Kondensators zu beachten.



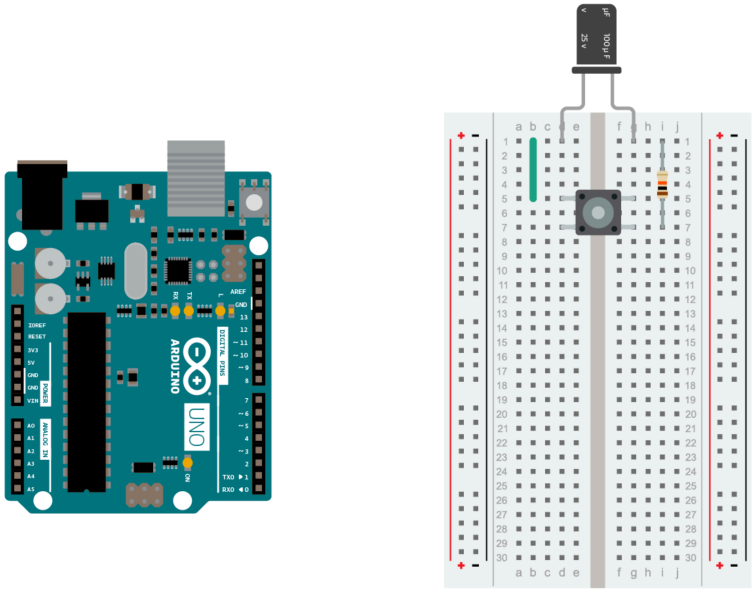
Step 14

Schließt die Batterie wieder an, sodass der Kondensator wieder vollständig geladen wird. Denkt daran: Minusleitung zur Kathode des Kondensators und Plusleitung zur Anode des Kondensators.



Step 15

Entfernt die Messleitungen vom Multimeter und trennt die Batterie ab. Der Kondensator sollte nun vollständig geladen sein.

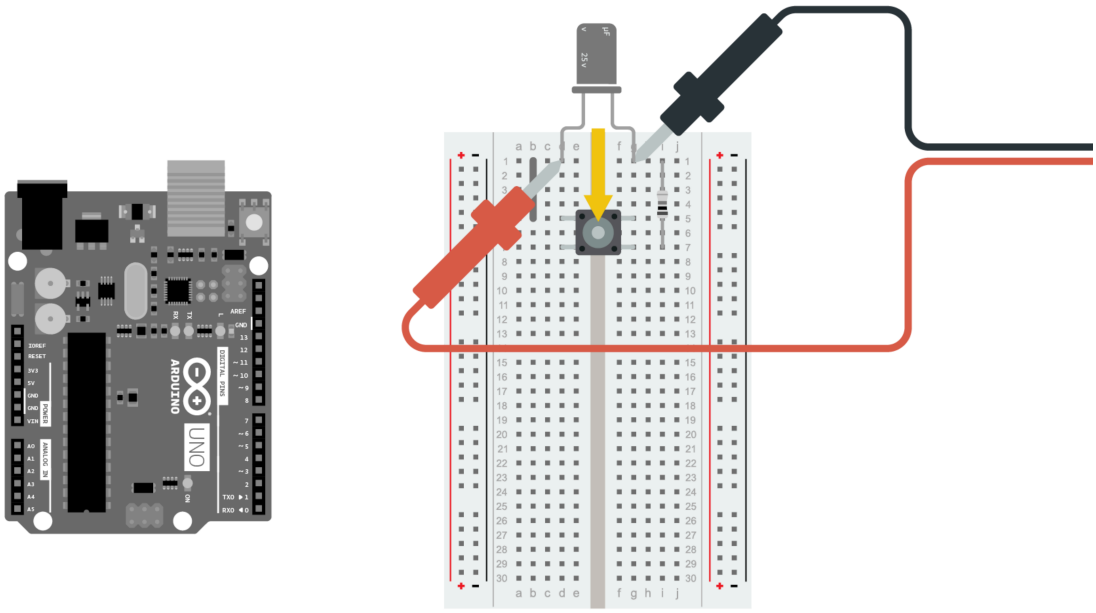


Step 16

Schließt die Multimeter-Prüfleitungen wieder an den Kondensator an (rote Leitung zur Anode und schwarze Leitung zur Kathode). Der Kondensator beginnt, sich zu

entladen. Haltet den Druckknopf gedrückt und beobachtet die Geschwindigkeit, mit der sich der Kondensator entlädt. Haltet den Knopf gedrückt, bis sich der Kondensator vollständig entlädt und das Multimeter Null (oder fast Null) Volt anzeigt.

Strom fließt vom Kondensator durch den 10.000-Ohm-Widerstand, durch den Tastschalter und zurück zum Kondensator. Diese Schaltung hat nur einen Widerstand von 10.000 Ohm; mit weniger Widerstand entlädt sich der Kondensator viel schneller. Es sollte etwa 10 Sekunden dauern, bis sich der Kondensator entladen hat.



Hinweis: Ihr könntet von Leuten hören, die einen Kondensator durch Kurzschluss entladen. Das bedeutet, dass die Anschlüsse des Kondensators direkt verbunden werden, ohne dass der Stromkreis belastet wird. Dies ist eine schlechte Praxis, die euch oder euren Geräten möglicherweise Schaden zufügen kann. Kurzschlüsse können Funken erzeugen, die zu Verbrennungen oder zu Elektroschocks führen können.

Obwohl bei der Verwendung der 100 µF Kondensatoren in eurem Bausatz nur ein geringes Risiko für euch besteht, können größere Kondensatoren ein ziemlich großes Risiko darstellen. Wenn ihr an einer Schaltung mit Kondensatoren arbeitet, ist es immer ratsam zu prüfen, ob die Kondensatoren mit einem Multimeter aufgeladen sind, und wenn ja, sie mit einer Last zu entladen, so wie ihr es bei dieser Übung getan habt.

17) Schaltet das Multimeter aus und legt alle Komponenten wieder an ihren richtigen Lagerort zurück.

TEACHER NOTES

VERSTÄNDNIS-ÜBERPRÜFUNG

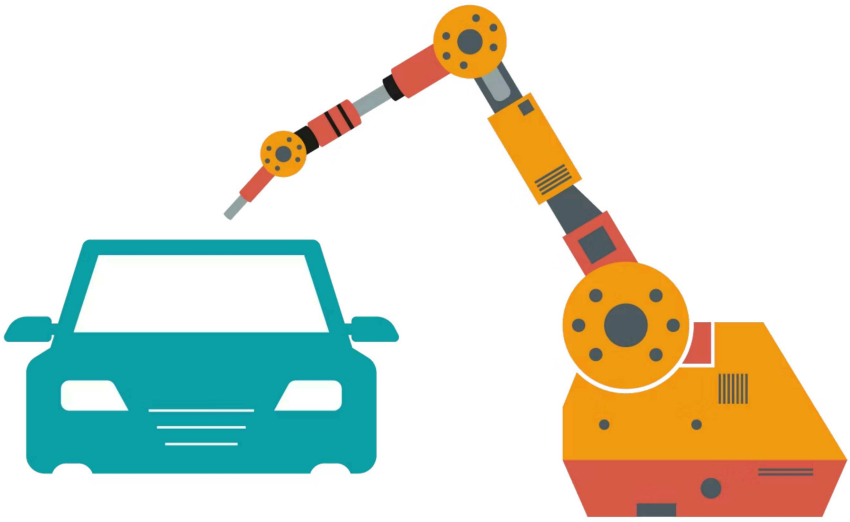
Fragen Sie die Schülerinnen und Schüler, ob sich der Kondensator schneller oder langsamer entlädt, wenn der 10.000-Ohm-Widerstand durch einen 220-Ohm-Widerstand ersetzt wird.

Der Kondensator wird sich viel schneller entladen. Je niedriger der Widerstand in der Schaltung, desto mehr Strom kann durch die Schaltung fließen.

ERFAHREN SIE MEHR: **Zeitkonstante des Kondensators**

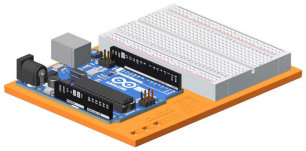
Sport-Roboter

Der Bereich der Robotik ist eine der am schnellsten wachsenden Technologiebranchen. Vielleicht wisst ihr bereits darüber Bescheid, wie Roboter eingesetzt werden, um Dinge wie Autos herzustellen, Pakete in Lagerhallen zu sortieren oder sogar Lebensmittel in einigen Restaurants zu kochen. Gegenwärtig setzen Ingenieure Roboter ein, um Probleme zu lösen, z.B. in selbstfahrenden Fahrzeugen, bei der Paketzustellung und der Sicherheitsüberwachung. Roboter werden sogar in der Sportindustrie zur Unterhaltung eingesetzt.



Es wurden Fernsehsendungen geschaffen, in denen Roboter einander gegenüberstehen und kämpfen und versuchen, sich gegenseitig zu zerstören. Immer beliebter werden Drohnenflugwettbewerbe, bei denen Menschen Drohnen durch einen Hindernis-Parcours fliegen und verschiedene Aufgaben erfüllen. Auch Wettbewerbe für Robotik-Ingenieurskunst und -Design erfreuen sich wachsender Beliebtheit. Roboter dringen sogar in den Bereich der traditionellen Sportarten vor. Gegenwärtig werden Roboter gebaut, die Fußball, Golf, Pingpong und sogar Basketball spielen. In dieser Übung werdet ihr einen Sport-Roboter bauen und programmieren, der einen Ball treten, schlagen oder werfen kann.

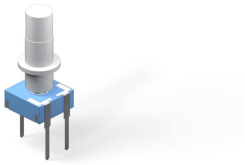
Benötigtes Material



1 ARDUINO PROJEKT BOARD

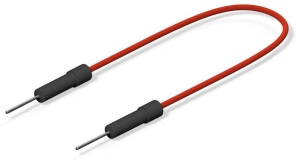
1 100 MIKROFARAD KONDENSATOR

1 SERVO

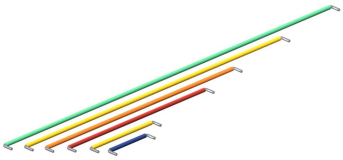


1 POTENTIOMETER

1 SCHWARZE STROMKABEL



1 ROT STROMKABEL



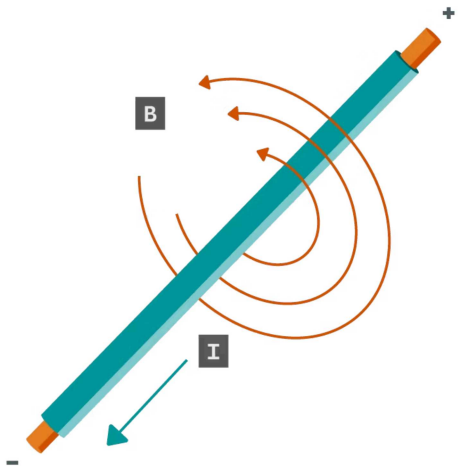
1 STECKVERBINDER

1 MULTIMETER MIT PRÜFLEITUNGEN

1 USB-KABEL

Motoren und Generatoren

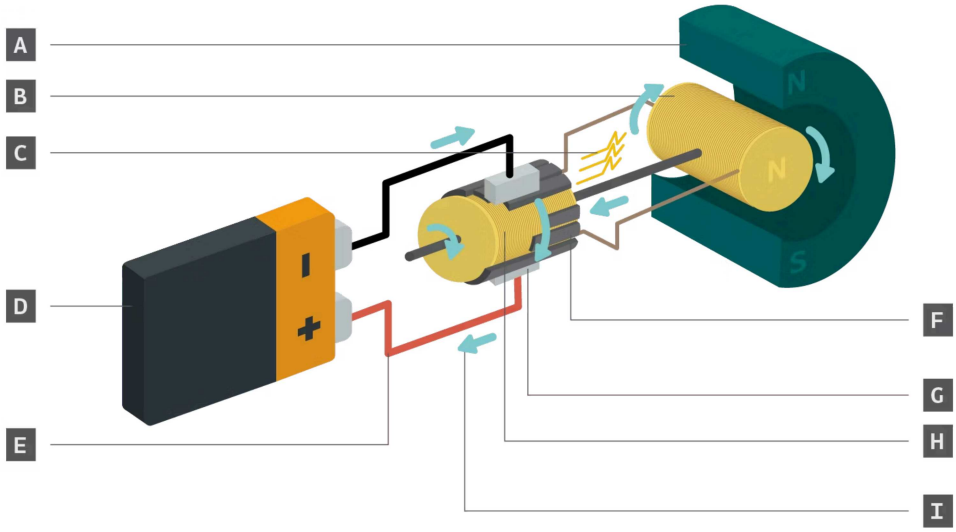
Es besteht eine besondere Beziehung zwischen Elektrizität und Magnetismus. Wenn sich Strom durch einen leitenden Pfad wie einen Draht bewegt, erzeugt er ein Magnetfeld. Wenn ihr in der Nähe eines Drahtes ein sich bewegendes Magnetfeld habt, kann es Strom durch den Draht schicken. Diese elektromagnetischen Eigenschaften sind es, die Motoren zum Drehen und Generatoren zur Erzeugung von Elektrizität veranlassen.



I) Strom B) Magnetisches Feld

Im Inneren eines Elektromotors befindet sich ein Magnet. Magnete haben zwei Pole - einen Nordpol und einen Südpol. Auch im Inneren eines Motors befinden sich mehrere Drahtspulen. Wenn Elektrizität durch den Draht fließt, erzeugt sie einen Elektromagneten, der ebenfalls einen Nord- und einen Südpol hat. Wie ihr wahrscheinlich bereits wisst, stoßen sich solche Pole ab und entgegengesetzte Pole ziehen sich an. Dies führt dazu, dass sich der Motor dreht, während er versucht, sich auf das Magnetfeld auszurichten. Wenn sich der Motor dreht, ändert der durch die Spulen fließende Strom seine Richtung. Dadurch werden die Pole des Elektromagneten umgedreht, und da sich gleiche Pole abstoßen und entgegengesetzte Pole anziehen, dreht sich der Motor weiter. Der Motor dreht sich, weil sich dieser Vorgang immer wieder wiederholt. Eine Änderung der Spannung an

der Spule verändert die Stärke des Elektromagneten und beeinflusst, wie schnell der Motor sich dreht.



A) Permanentmagnet B) Spule C) Elektrizität D) Batterie E) Stromleitung F) Magnet G) Bürsten H) Gewickelter Kupferdraht I) Richtung des Elektronenstroms

Ein Generator ist ähnlich aufgebaut wie ein Motor. Er funktioniert jedoch genau umgekehrt. In einem Generator bringt eine Turbine die Spulen zum Drehen. Dies kann mit Wasser in einem hydroelektrischen Damm, mit Wind in einer Windturbine oder mit Dampf in Kohle- und Kernkraftwerken geschehen. Wenn die Turbine die Spulen dreht, wird im Draht Strom erzeugt und so Elektrizität generiert.

ERFAHREN SIE MEHR: Arten von Motoren

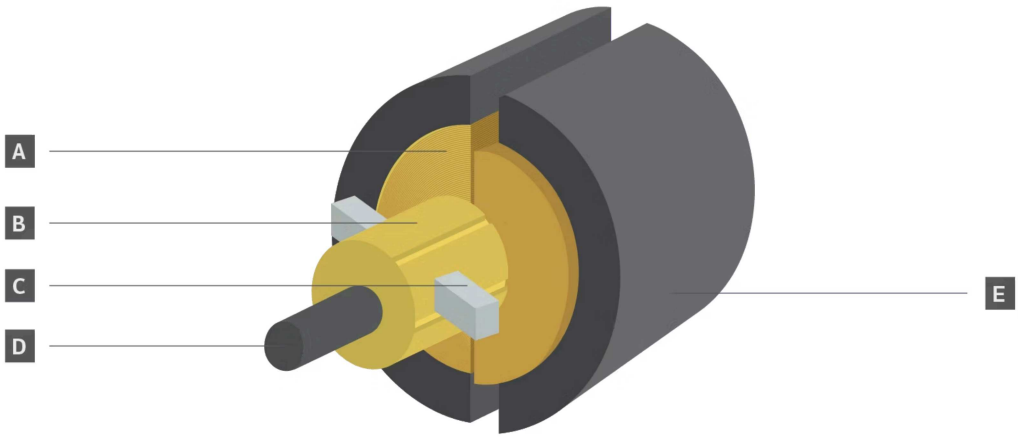
Blickpunkt Erfindungen

Elektrische Motoren

Elektromotoren gibt es seit mehr als einem Jahrhundert. Sie treiben unsere Ventilatoren, unsere Waschmaschinen, unsere Computerlaufwerke und viele andere Dinge an. Aber es ist der Gasmotor, der die Phantasie der Menschen beflügelt hat. Warum? Weil er der Motor der Wahl für die Automobilindustrie war. Neben einem brüllenden Achtzylindermotor, der einen Stahllastwagen antreiben kann, wirkt der Motor, der euren Mixer antreibt, bescheiden.

Aber Elektromotoren bekommen jetzt den Respekt, den sie verdienen. Modernste Technik hat das Elektroauto zum Mainstream gemacht. Heutzutage sind Elektromotoren den Gasmotoren regelmäßig überlegen.

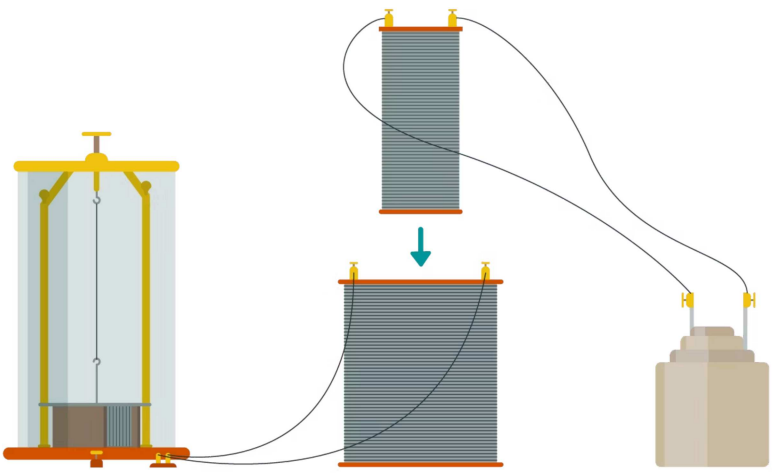
Elektromotoren sind sehr unterschiedlich. Sie alle nutzen jedoch die Vorteile einer Beziehung zwischen Elektrizität und Magnetismus. In einem Induktionsmotor, wie er von Tesla-Fahrzeugen verwendet wird, erzeugt ein elektrischer Strom ein rotierendes Magnetfeld, das eine Metallachse dreht. Diese Drehbewegung treibt das Auto an.



A) Rotorspulen B) Kommutator C) Bürsten D) Welle E) Stator (Magnete)

Der erste Elektromotor wird Michael Faraday, einem Wissenschaftler aus dem 19. Jahrhundert in London, zugeschrieben. Zu Faradays Zeit waren die meisten Wissenschaftler wohlhabende Männer. Faraday war der arme Sohn eines Schmiedes, der darum kämpfen musste, von seinen Kollegen aus der Oberschicht akzeptiert zu werden. Aber sein brilliantes, unkonventionelles Denken führte ihn zu einer der größten Entdeckungen der Geschichte.

Faraday war der erste, der die Beziehung zwischen Elektrizität und Magnetismus verstanden hat. Er erkannte, dass beide Kräfte unsichtbare Felder erzeugen. Diese Felder bewegen sich in gekrümmten Mustern, üben Druck aufeinander aus und bewirken eine Rotation. Faraday erkannte, wie diese Felddrehung Objekte bewegen konnte, die in ihrem Strom gefangen waren. Diese Bewegung konnte für bestimmte Aufgaben genutzt werden.



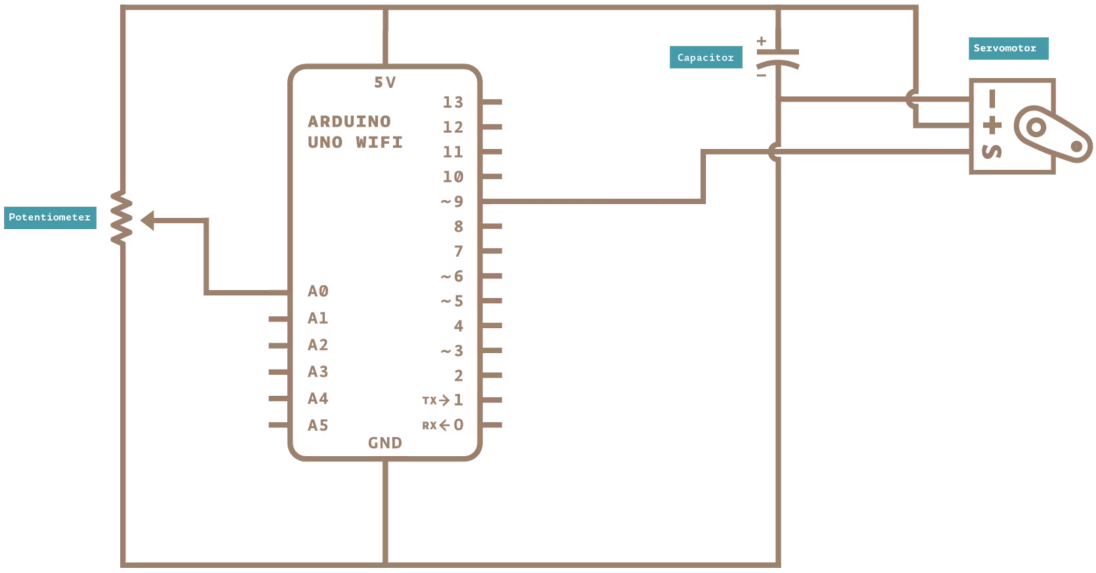
Der von Faraday geschaffene Motor war primitiv. Er diente nur dazu, einen Kupferdraht kreisförmig um einen Magneten schwingen zu lassen. Aber nach wenigen Jahren inspirierte dieser nette Trick zahlreiche Erfinder. Sie schufen verbesserte Motoren auf der Grundlage von Faraday's Konstruktion.

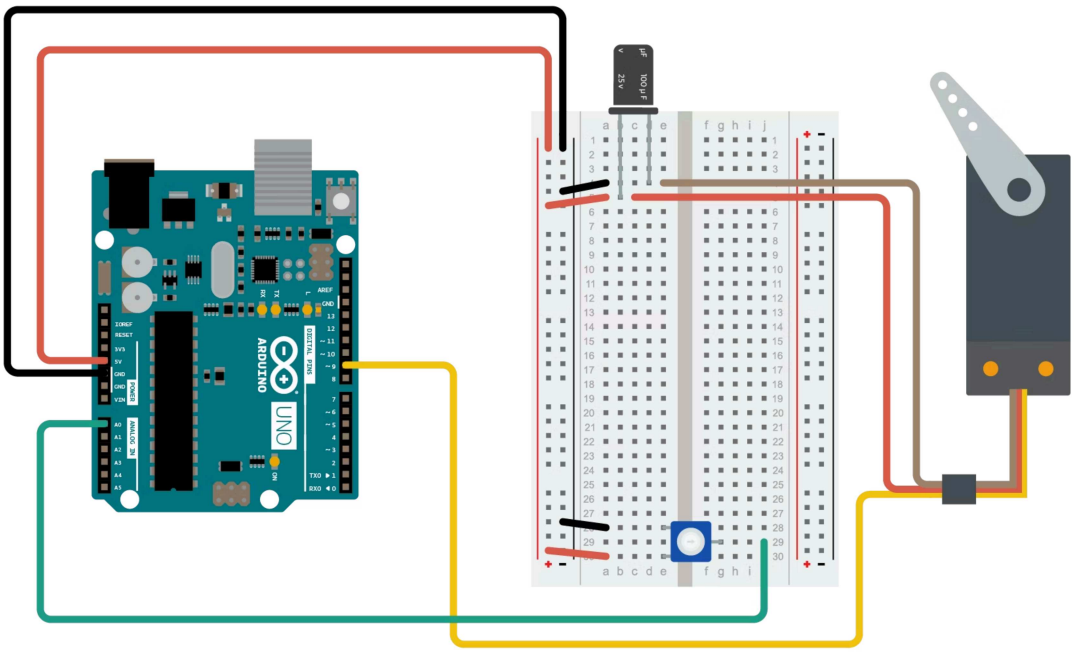
Ein solches Erfinderpaar war ein armes amerikanisches Ehepaar - die Davenports. Thomas Davenport war ein Schmied. Er kaufte einen Elektromagneten. Er und seine Frau Emily zerlegten ihn, um seine Konstruktion zu verstehen. Sie konstruierten Weitere auf der Grundlage dessen, was sie gelernt hatten. Sie stießen auf eine Hürde, als sie isolierten Draht benötigten, aber nur blanken Kupferdraht hatten. Da Emily erkannte, dass Seide ein Isolator ist, schnitt sie ihr Hochzeitskleid aus Seide in

Streifen, um den Draht einzuwickeln. Indem sie ihr Gerät an einer schwachen chemischen Batterie befestigten, schufen die Davenports den ersten funktionierenden Gleichstrommotor der Welt.

Heute sind Gleichstrommotoren eines der am weitesten verbreiteten Geräte auf der Welt. In wenigen Jahren könnte der Gasmotor veraltet sein. Auch im aufstrebenden Bereich der Robotik finden Elektromotoren neue Anwendungen. Motoren, die auch als Aktoren bezeichnet werden, verwenden Elektromagnete und Permanentmagnete, um kinetische Energie zu erzeugen. Betrachtet man das Innenleben der meisten Roboter, so treiben Aktoren die Bewegung an. Ihr findet Aktoren überall im Roboter, von den beweglichen Gelenken eines Arms und Greifers bis hin zu den Hubbeinen oder drehenden Rädern, die für die Fortbewegung sorgen. Der Sport-Roboter, den ihr in der nächsten Aktivität bauen werdet, wird einen Aktor verwenden, um seine Tritt-, Schlag- oder Wurfbewegung auszuführen.

Schaltplan


Capacitor
Servomotor
Potentiometer
5V
ARDUINO
UNO WIFI
13
12
11
10
~9
8
7
~6
~5
4
~3
2
TX→1
RX←0
A0
A1
A2
A3
A4
A5
GND
S

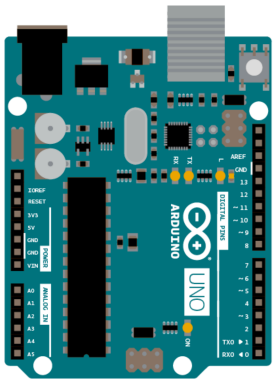


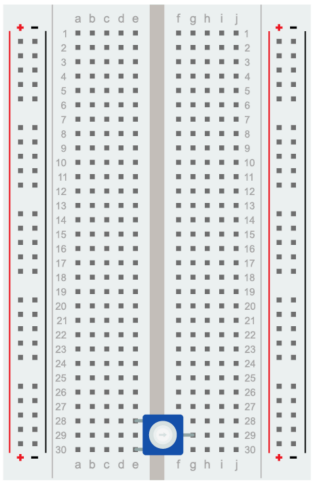
Aufbau der Schaltung

Verwendet den Schaltplan und den Verdrahtungsplan oder die folgende Diashow, um die Schaltung aufzubauen.

Step 1

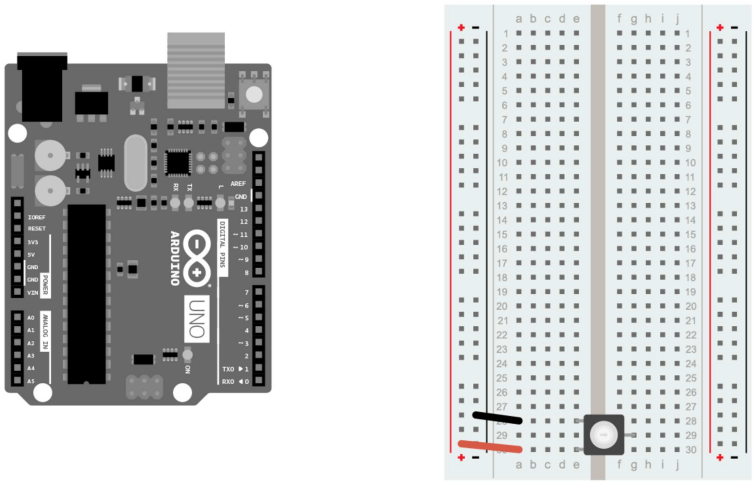
Befestigt das Potentiometer so an dem Steckbrett, dass es den Spalt in der Mitte des Steckbretts überbrückt. Bringt das Potentiometer so an, dass die Power- und Ground Pins zum Arduino UNO R3 Board zeigen und der Schleifer-Pin von dem Arduino UNO R3 Board weg zeigt. Das Potentiometer sollte nur in einer Richtung auf das Steckbrett passen. Wenn die Potentiometer-Pins nicht auf das Steckbrett passen, dreht ihr das Potentiometer um 90 Grad.





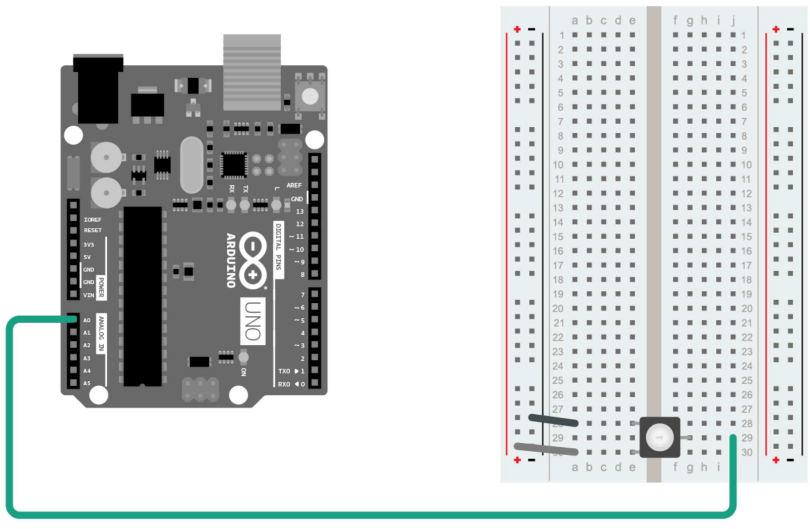
Step 2

Verwendet einen kurzen Steckverbinder, um den Power Pin des Potentiometers mit dem positiven Anschluss des Steckbretts zu verbinden. Verwendet einen weiteren kurzen Steckverbinder, um den Ground Pin des Potentiometers mit dem negativen Anschluss des Steckbretts zu verbinden.



Step 3

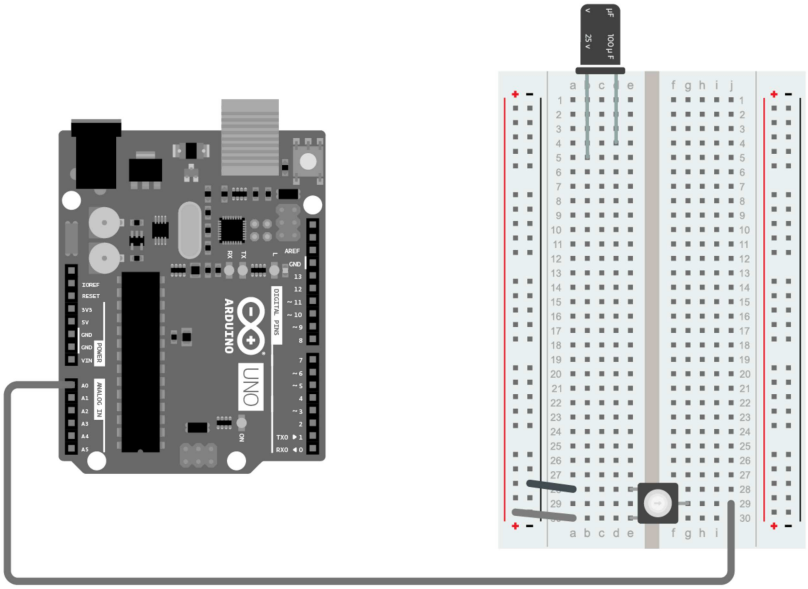
Verwendet einen langen Steckverbinder, um den Schleifer-Pin des Potentiometers mit dem Analog-Pin A0 des Arduino UNO R3 Boards zu verbinden.



Step 4

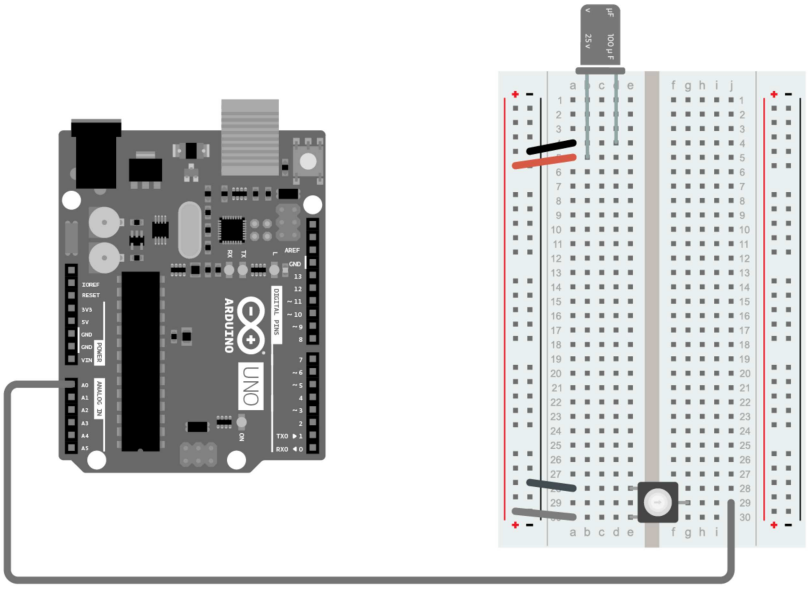
Schließt einen 100 µf Kondensator in der Nähe der oberen Seite des Steckbretts an. Schließt den Kondensator so an, dass die Anode von euch gewendet ist und die

Kathode näher bei euch liegt. Erinnert euch daran, dass die Kathode der kurze Draht ist und durch den Streifen auf der Seite des Kondensators gekennzeichnet wird.



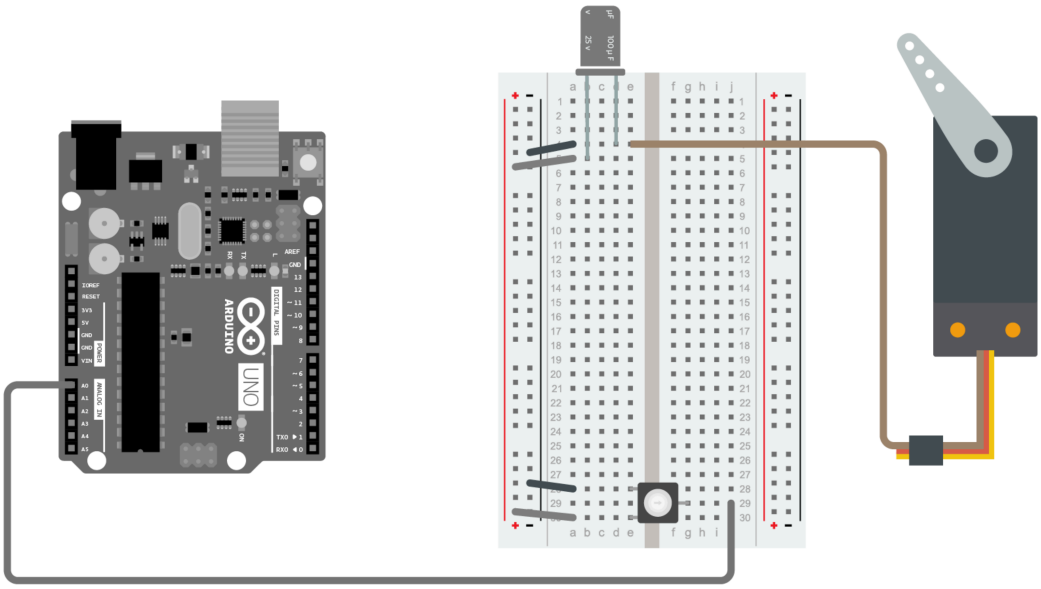
Step 5

Verwendet einen kurzen Steckverbinder, um die Anode des Kondensators mit dem positiven Anschluss des Steckbretts zu verbinden. Verwendet einen weiteren kurzen Steckverbinder, um die Kathode des Kondensators mit dem negativen Anschluss des Steckbretts zu verbinden.



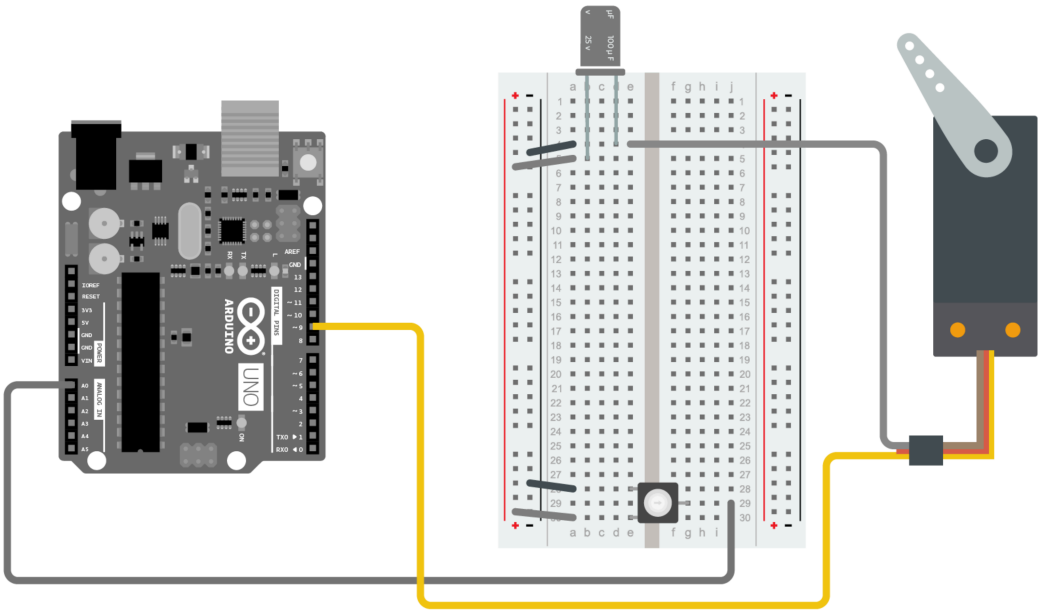
Step 6

Verwendet einen langen Steckverbinder, um den schwarzen Draht von der Buchse des Servos mit der Kathode des Kondensators zu verbinden. Achtet darauf, dass das Servohorn am Servo angeschlossen und mit der kleinen Schraube gesichert ist.



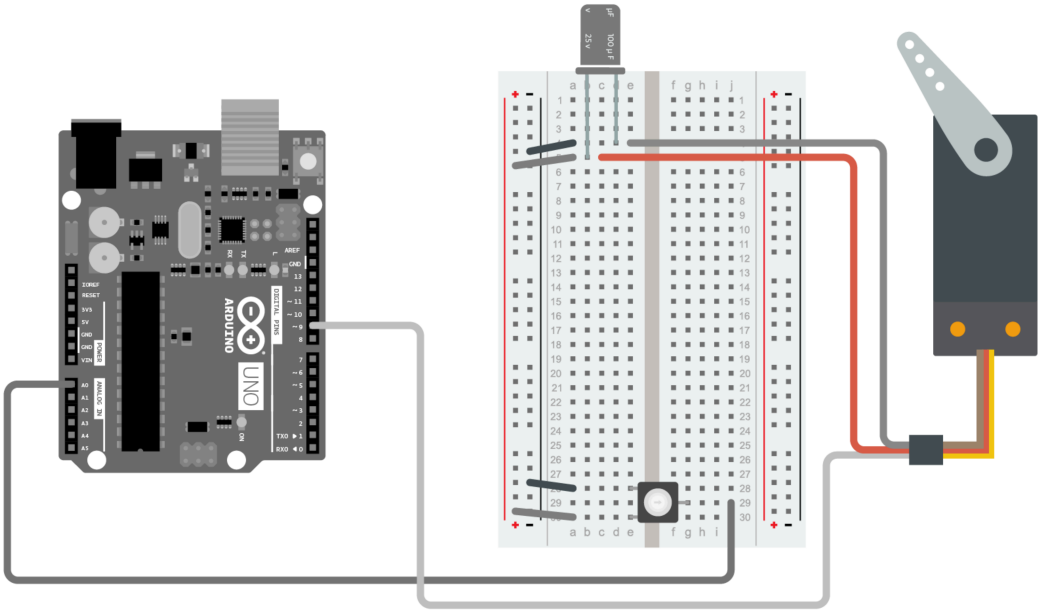
Step 7

Verbindet den weißen Draht von der Buchse des Servos mit Pin 9 des Arduino UNO R3 Boards. Dies ist der Draht, der den Servo steuert.



Step 8

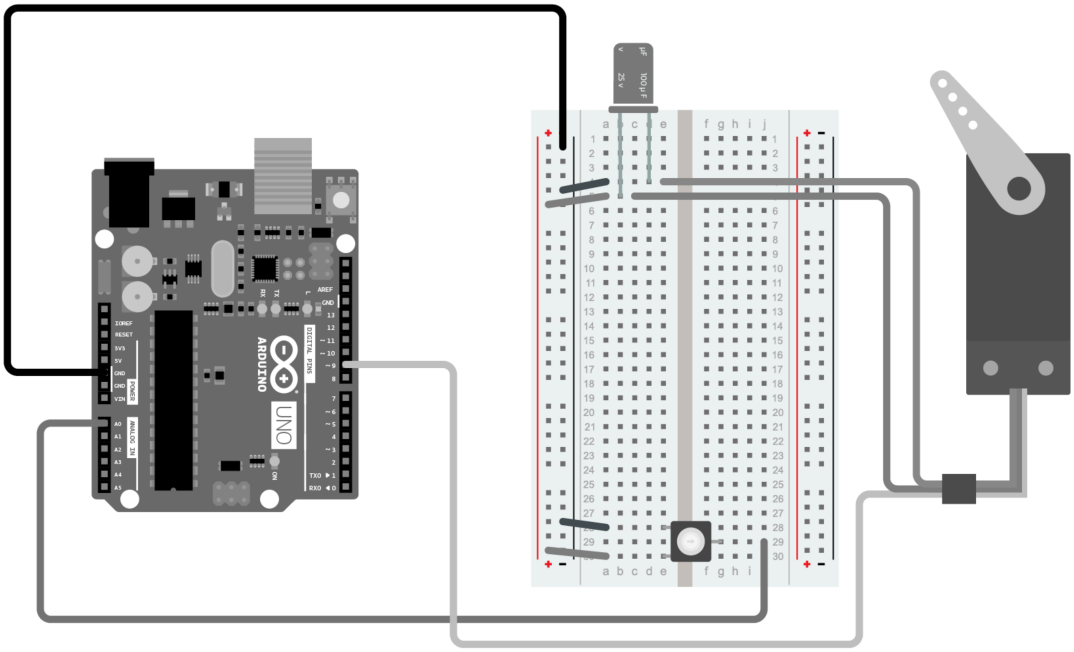
Verwendet einen dritten langen Steckverbinder, um den roten Draht von der Buchse des Servos mit der Anode des Kondensators zu verbinden.



Step 9

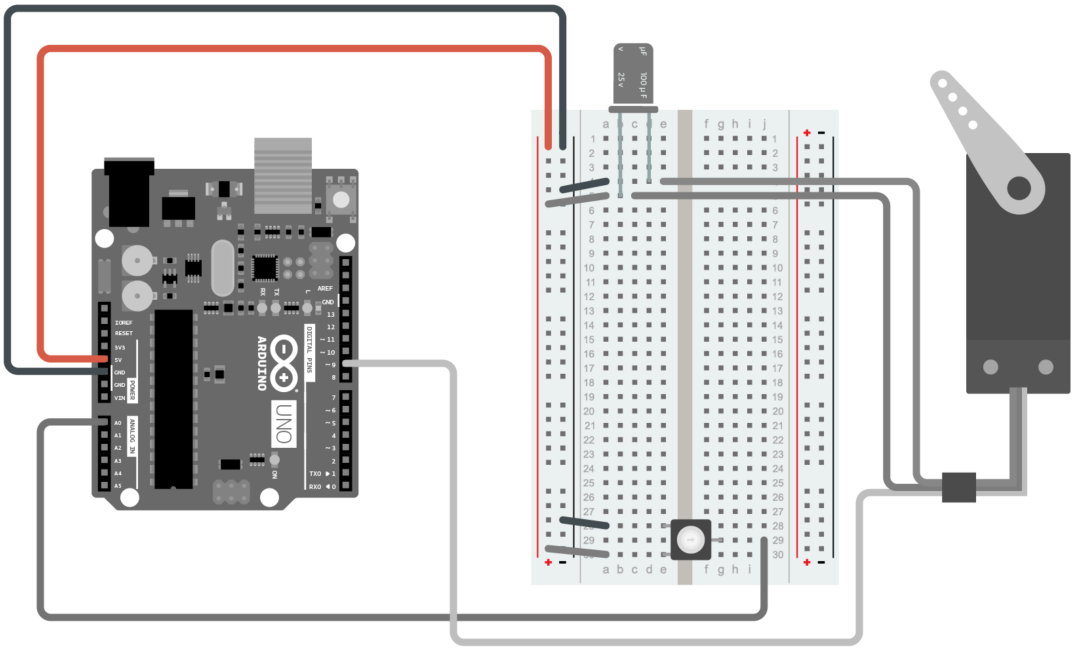
Schließt das schwarze Stromkabel an einen Ground-Pin auf dem Arduino UNO R3 Board an. Schließt das andere Ende des Stromkabels an den Minuspol des

Steckbretts an. Dadurch werden der Servo und das Potentiometer geerdet.



Step 10

Schließt das rote Stromkabel an den 5V-Pin auf dem Arduino UNO R3 Board an. Verbindet das andere Ende des Stromkabels mit dem positiven Anschluss des Steckbretts. Dadurch werden Servo und Potentiometer mit fünf Volt Strom versorgt.



Code-Erstellung - Servo-Steuerung

In dieser Übung werdet ihr einen Sketch schreiben, der es euch ermöglicht, den Winkel des Servos durch Drehen des Potentiometers zu steuern.

TEACHER NOTES

Bei dieser Übung beginnen die Schülerinnen und Schüler mit einem Sketch, der einen analogen Wert vom Potentiometer liest und den Wert auf einen Bereich skaliert, der zur Steuerung eines Standard-Servos verwendet werden kann. Während dieser Übung werden die Schülerinnen und Schüler mit mehreren neuen Codierungskonzepten konfrontiert. Wenn es ihnen überwältigend erscheint, ermutigen Sie sie, weiterzumachen, und versichern Sie ihnen, dass sie bei der Bearbeitung dieser und anderer Übungen in dieser Lektion Gelegenheit haben werden, ihr Verständnis weiter zu entwickeln.

In dieser Übung werden die Schülerinnen und Schüler mit folgenden Kodierungskonzepten vertraut gemacht:

- **#include < library.h> –** Importiert eine installierte Bibliothek in den Sketch
- **Objekterstellung –** Benennt ein Objekt, das mit der Bibliothek verwendet werden soll
- **Konstantendeklarationen –** Deklariert den Typ und den Namen einer Konstanten
- **servoName.attach() –** Ein Befehl aus der Servo-Bibliothek, der ein Servo-Objekt mit einem physikalischen Pin auf dem Arduino UNO R3 Board verbindet
- **servoName.write(angle)–** Ein Befehl aus der Servo-Bibliothek, der ein Servo-Objekt auf einen bestimmten Winkel bewegt
- **map()–** Skaliert einen Wert von einem Bereich in einen anderen Bereich

Die Schülerinnen und Schüler werden folgende Konzepte überprüfen:

- **Variablen-Deklarationen –** Deklariert den Typ und den Namen einer Variable
- **Variablenzuweisung –** Weist einer Variablen einen Wert zu

1) Startet die Arduino IDE auf eurem Computer oder Gerät.

2) Wählt im Dateimenü die Option **Neu**. Dadurch wird ein neuer Sketch in einem neuen Fenster gestartet. Ihr könnt das andere Sketchfenster schließen.

3) Ihr könnt entweder das Sketchfenster maximieren und zwischen ihm und diesen Anweisungen hin- und herschalten oder ihr könnt beide Fenster nebeneinander anordnen, so dass ihr beide Fenster gleichzeitig sehen könnt. Verwendet die Methode, die für euch am besten geeignet ist.

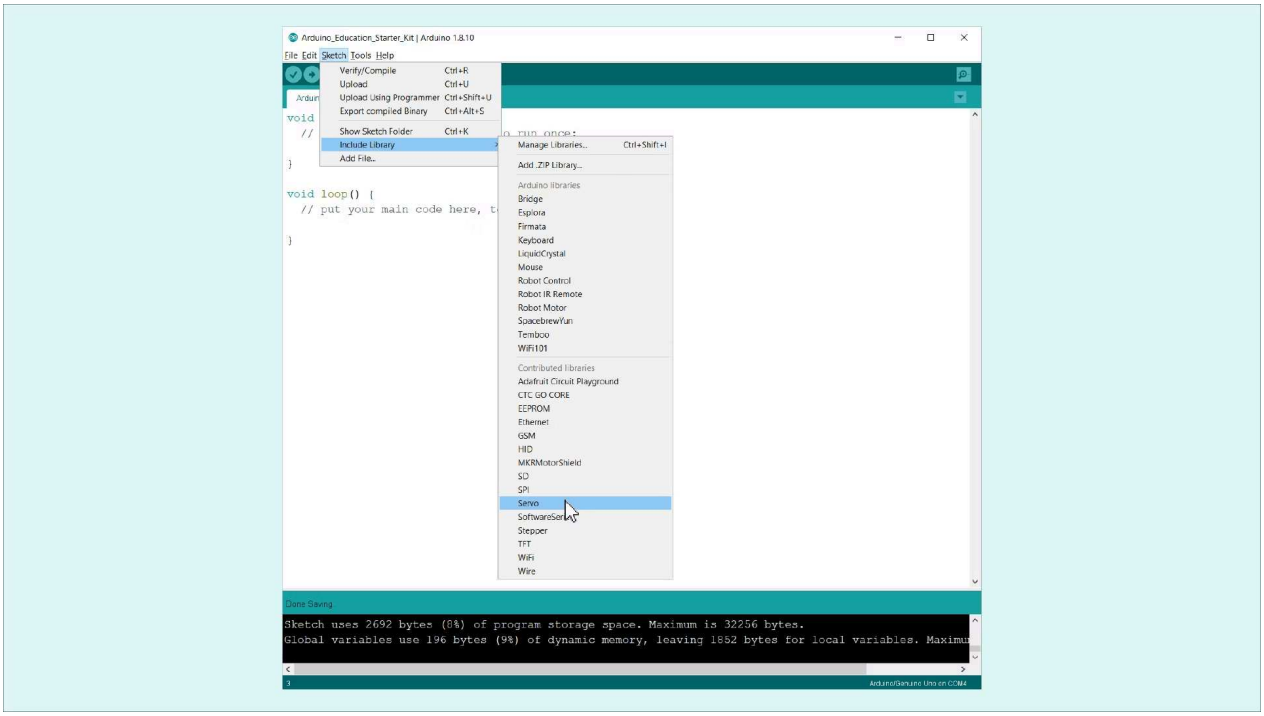
4) Speichert die Skizze unter einem neuen Namen. Wählt im Dateimenü die Option **Speichern unter...** Der Speicherort des Sketches sollte standardmäßig das Arduino Sketchbook sein. Wenn dies nicht der Fall ist, fragt euren Lehrer, wo ihr den Sketch auf eurem Computer speichern sollt. Benennt die Datei "Lektion6*V1*", gefolgt von euren Initialen und denen eures Partners. Klickt auf **Speichern**.

5) Bisher habt ihr etwas über Inputs und Outputs und ihre Beziehung zum Arduino UNO R3 Board erfahren. Elektronische Komponenten, die Eingänge bereitstellen, werden **Sensoren** genannt. Sensoren messen etwas aus der Welt um sie herum und wandeln diese Messung in Elektrizität um, die das Arduino UNO R3 Board lesen kann. Das Potentiometer ist zum Beispiel ein Sensor, der den Widerstand misst. Outputkomponenten werden durch das Arduino UNO R3 Board gesteuert. Diese Ausgabegeräte werden oft als **Aktoren** bezeichnet. In den Lektionen 1 bis 4 habt ihr LEDs als Aktoren verwendet. In dieser Lektion ist der Servo, den ihr an eure Schaltung angeschlossen habt, ein Aktor.

Einige Sensoren und Aktoren, wie z.B. der Servo, sind komplex und erfordern spezielle Befehle, um sie zu steuern. Diese Befehle sind in **Bibliotheken** gruppiert und bieten zusätzliche Funktionalität für das Arduino UNO R3 Board. Es gibt alle Arten von Bibliotheken, die Menschen erstellt haben. In diesem Sketch werdet ihr eine Bibliothek namens Servo verwenden. Diese Bibliothek ist eine der Standardbibliotheken, die bereits in der Arduino IDE installiert sein sollte. Um die Servo-Bibliothek zu verwenden, müsst ihr die Bibliothek in den Sketch importieren.

Hinweis: Ihr könnt andere Bibliotheken installieren, indem ihr in der Arduino IDE im Menü Tools auf **Manage Libraries...**. klickt. Für die Lektionen dieses Kurses müsst ihr jedoch keine Bibliotheken installieren. Für weitere Informationen über die Installation zusätzlicher Arduino-Bibliotheken klickt ihr auf den folgenden Link **https://www.arduino.cc/en/Guide/Libraries**

Um die Servobibliothek zu importieren, damit sie in dem Sketch verwendet werden kann, geht ihr zum Menü **Sketch** und klickt auf **Bibliothek einfügen**. Dieses Pop-Out-Menü zeigt euch die Bibliotheken an, die derzeit in der IDE installiert sind. Wählt **Servo** aus dem Pop-Out-Menü.



Ihr solltet bemerken, dass am Anfang eures Sketches eine neue Codezeile hinzugefügt wurde. Dieser Befehl importiert die Servo-Bibliothek und ermöglicht euch die Verwendung aller ihrer Befehle und Funktionen.

6) Fügt der der Include-Anweisung einen Code-Kommentar hinzu, der euch daran erinnert, wozu dieser Befehl dient.

7) Bei der Programmierung arbeiten Bibliotheken oft mit Klassen. Eine **Klasse** ist wie eine Schablone, die eine Menge Informationen in verschiedenen Variablen oder Eigenschaften enthält. Diese Variablen oder Eigenschaften beschreiben ein Objekt. Ein **Objekt** ist ein spezifisches Beispiel, das zu der Klasse gehört. Mit anderen Worten, wenn eine Klasse eine Vorlage ist, ist ein Objekt eine fertige Vorlage, die alle

Eigenschaften eines bestimmten Beispiels beschreibt. Klassen enthalten auch den Code für Objekte, um bestimmte Verhaltensweisen oder Aktionen auszuführen. Diese Verhaltensweisen werden als **Methoden** bezeichnet.

Zum Beispiel könnte ein Hund als eine Klasse angesehen werden. Eigenschaften für die Hundeklasse könnten Rasse, Gewicht, Alter und Farbe sein. Methoden, die in die Hundeklasse aufgenommen werden könnten, sind fressen, schlafen, laufen, sitzen, apportieren und sich umdrehen. Ein Objekt der Hundeklasse wäre ein **Labrador Retriever**, der 40 Pfund (18 Kilogramm) wiegt, sieben Jahre alt und braun ist.

Servos sind eine Klasse von Objekten, die ihr in eurem Sketch verwenden werdet. Ihr müsst ein bestimmtes Servo-Objekt definieren, indem ihr eurem Servo einen Namen gebt. Wenn ihr ein Servo-Objekt mit einem Namen definiert, könnt ihr alle Methoden verwenden, die zu der Servo-Klasse gehören.

Um ein Servo-Objekt zu erstellen, gebt ihr den folgenden Befehl nach der Include-Bibliothek-Anweisung ein: **Servo myServo;**. Dieser Befehl erzeugt ein Objekt namens **myServo** der Klasse Servo. Fügt einen Codekommentar ein, der den Befehl erläutert.

Hinweis: Dieselben Regeln für die Benennung von Variablen gelten auch für die Benennung von Objekten. Ihr könnt das Servo-Objekt nach allem benennen, was nicht bereits ein Schlüsselwort oder ein anderer bereits verwendeter Name ist. Anstatt das Objekt myServo zu nennen, könntet ihr auch Jack oder Jill oder euren eigenen Namen verwenden. Ihr könnt das Objekt jedoch nicht Servo nennen, da dies der Name der Bibliothek ist. Beachtet auch, dass dieser Befehl mit einem Semikolon endet.

TEACHER NOTES

Klassen, Objekte und Methoden können für Schülerinnen und Schüler schwer verständliche Konzepte sein. Es ist an diesem Punkt nicht wichtig, dass sie ein vollständiges Verständnis der Konzepte haben. Sie müssen nur genug wissen, um in der Lage zu sein, ein Objekt zu erstellen und die Bibliotheksbefehle für dieses Objekt zu verwenden.

8) In Lektion 4 habt ihr etwas über Variablen gelernt und wie sie Informationen speichern können. Der Wert einer Variablen kann sich im Laufe eines Programms oder eines Sketches ändern. In Lektion 4 habt ihr eine Variable verwendet, um den Wert vom Potentiometer zu verfolgen. Als ihr das Potentiometer manipuliert habt, hat sich auch die Variable geändert.

Eine **Konstante** ist wie eine Variable, außer dass sie sich nicht ändert. Konstanten bleiben in dem gesamten Sketch gleich. Genau wie Variablen gibt es verschiedene Arten von Konstanten. Einige Beispiele für die verschiedenen Arten von Konstanten sind ganze Zahlen (Zahlen ohne Dezimalkomma), Gleitkommazahlen (Zahlen mit Dezimalkomma), Zeichen (Buchstaben), Zeichenketten (Wörter oder Phrasen) und Boole'sche Zeichen (wahr oder falsch). In diesem Kurs werdet ihr nur Konstanten vom Typ Ganzzahl verwenden. Zum Beispiel ist der Arduino-Pin, an den das Potentiometer in eurer Schaltung angeschlossen ist, Pin A0. Obwohl die ID dieses Pins ein Buchstabe gefolgt von einer Zahl ist, erkennt die IDE diesen Pin und alle analogen Pins automatisch als ganze Zahlen. Da ihr den Pin, an den das Potentiometer angeschlossen ist, während der Sketch läuft, nicht ändern werdet, könnt ihr diesen Pin als Konstante deklarieren. Dies ist aus mehreren Gründen von Vorteil:

- Jedes Mal, wenn ihr euch auf den Pin beziehen müsst, könnt ihr ihn mit dem Namen der Konstanten und nicht mit der eigentlichen Pin-Nummer bezeichnen. Dadurch wird der Code sauberer und für andere leichter verständlich.

- Wenn ihr jemals die Arduino-Pin-Nummer physisch ändern müsst, braucht ihr nur den Code in der Deklaration am Anfang zu ändern und nicht alle Stellen im Code, an denen dieser Pin genannt wird.

Die Benennung einer Konstanten folgt den gleichen Regeln wie die Benennung einer Variablen. Ihr könnt ihr einen beliebigen Namen geben, der nicht bereits ein Schlüsselwort oder ein in der Arduino IDE verwendeter Befehl ist.

Das Schlüsselwort **const** wird verwendet, um eine Konstante zu erzeugen. Erzeugt nach eurer Servo-Objektzeile und vor der Funktion **void setup()** eine Konstante namens **potPin** und setzt sie gleich Pin A0. Gebt den folgenden Befehl ein:

9) Ihr werdet den Wert des Potentiometers ablesen und einen Winkelwert an das Servo ausgeben, der von dem Wert des Potentiometers abhängig ist. Dazu benötigt ihr zwei Variablen. Ertsellt eine ganzzahlige Variable namens **potValue** und setzt sie gleich Null. Diese Variable wird den Analogwert (von 0 bis 1.023) des Potentiometers verfolgen. Erstellt eine zweite ganzzahlige Variable namens **servoAngle** und setzt diese ebenfalls auf Null. Die Variable **servoAngle** verfolgt den dem Servo zugewiesenen Winkel (von 0 bis 179).

10) Verifiziert euren Code und debugged ihn, falls erforderlich.

FURTHER NOTES

Wenn ihr bei der Verifizierung eures Codes Fehler erhaltet, überprüft ihr, ob ihr die Konstante **potPin** als *A* gefolgt von einer Null und nicht als *A* gefolgt vom Buchstaben *O* deklariert habt.

11) Ihr seid bereit, zur Funktion **void setup()** des Sketches überzugehen. Als erstes müsst ihr dem Arduino Board mitteilen, an welchem Pin der Servo namens **myServo** angeschlossen ist. Die Servo-Bibliothek enthält einen Befehl, um genau dies zu tun - **servoName.attach(pin)**. In eurem Sketch lautet der **Servoname myServo**. Was den Pin betrifft, so steuert der weiße Draht den Servo und ist an den digitalen Pin 9 auf

dem Arduino UNO R3 Board angeschlossen. Gebt den folgenden Befehl in den Abschnitt **void setup()** ein.

Dieser Befehl tritt an die Stelle des Befehls **pinMode()**, den ihr in anderen Lektionen für einfache digitale Geräte wie LEDs verwendet habt.

12) Wenn ihr zur Funktion **void loop()** übergeht, müsst ihr als Erstes den Wert von dem an Pin A0 angeschlossenen Potentiometer lesen. Ihr müsst den Wert des Potentiometers in der Variablen **potValue** speichern. Gebt diesen Befehl am Anfang der **void loop()**-Funktion ein.

Beachtet, dass in diesem Befehl die Konstante **potPin** anstelle von A0 verwendet wird. Dadurch könnt ihr den analogen Pin am Anfang des Sketches ändern, ohne ihn hier ändern zu müssen.

13) Erinnert euch, dass analoge Signale einen Bereich von 0 bis 1.023 haben. Der Servo kann sich jedoch nur in einem Bereich von 0 bis 179 Grad bewegen. Dies ähnelt dem Problem, dem ihr in Lektion 4 begegnet sind, wo der Analogeingang einen Bereich von 0 bis 1.023 hatte, der Analogausgang jedoch nur einen Bereich von 0 bis 255. Ruft diesen Code aus dem Sketch in Lektion 4 wieder auf:

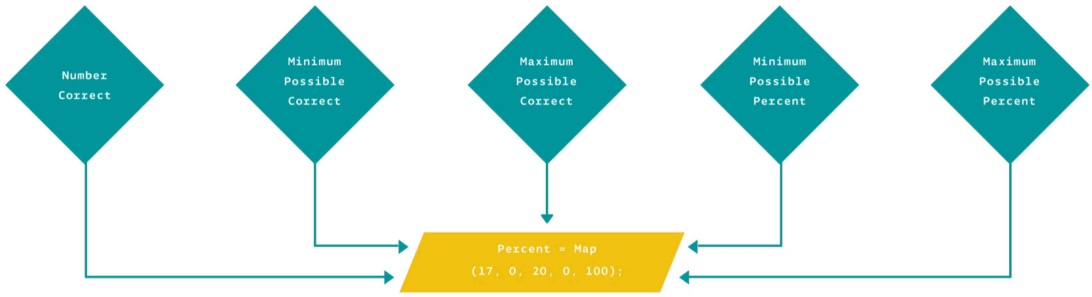
In Lektion 4 habt ihr **readValue**, das einen Bereich von 0 bis 1.023 hatte, durch vier geteilt, um **writeValue** in den Bereich von 0 bis 255 zu bekommen. Ihr könntet etwas Ähnliches für euren Sketch aus Lektion 6 tun. Die Arduino IDE hat jedoch eine praktische Funktion für die Behandlung von Unterschieden in Bereichen wie diesem. Mit der Funktion **map()** könnt ihr einen Datenbereich auf einen anderen skalieren. Die map-Funktion sieht wie folgt aus:

Beachtet, dass sich innerhalb der Klammern dieser Funktion fünf Werte befinden:

- **Abzubildender Wert -** Dies ist der Inputwert für die Funktion. In vielen Fällen handelt es sich um eine Variable, die von einem Sensor gelesen wird.
- **Eingangsminimum -** Um den Inputwert zu skalieren, muss die Funktion den niedrigsten Wert kennen, der die Eingabe sein kann.
- **Eingangsmaximum -** Durch Kenntnis sowohl des Eingangsminimums als auch des Eingangsmaximums kann die Funktion den Bereich der möglichen Werte bestimmen.
- **Ausgangsminimum -** Dies ist der niedrigste Wert, der im Ausgabebereich möglich ist.
- **Ausgangsmaximum -** Dies ist der höchstmögliche Wert im Ausgabebereich.

Hinweis: Der Befehl **map()** kann nur mit ganzzahligen Werten von -2.147.483.648 bis 2.147.483.647 ausgeführt werden. Mit anderen Worten: Ihr könnt in dem Befehl keine Dezimalzahlen verwenden. Wenn ihr Dezimalzahlen von einem Datenbereich auf einen anderen abbilden möchtet, muss dies mit Variablen und Gleichungen erfolgen.

Es mag euch vielleicht nicht bewusst sein, aber ihr verwendet in der Schule wahrscheinlich ziemlich oft etwas Ähnliches wie die Mapfunktion. Nehmen wir zum Beispiel an, ihr habt einen Test mit 20 Fragen gemacht. Ihr habt 17 Fragen richtig beantwortet und möchtet eure Prozentzahlen wissen. Wenn ihr die Kartenfunktion zur Durchführung dieser Berechnung verwenden würdet, würdet ihr die Funktion so eingeben:



Dieser Befehl würde eure Punktzahl von 17 im Bereich von 0 bis 20 möglichen richtigen Fragen dem Bereich von 0 bis 100 Prozent zuordnen. Es würde einen Wert von 85 Prozent ergeben.

Für eure **map()**-Funktion möchtet ihr die Variable **servoAngle** auf die Variable **potValue** (das ist der Wert vom Potentiometer) abbilden. Die **PotValue**-Variable ist ein Analogwert, der zwischen 0 und 1.023 liegen kann. Für den Outputbereich hat das Servo einen Bereich von 0 bis 179 Grad. Gebt den folgenden Befehl einschließlich eines Codekommentars am unteren Ende der Funktion **void loop()** ein.

TEACHER NOTES

Die Schülerinnen und Schüler könnten sich fragen, warum der maximale Winkel 179 Grad und nicht 180 Grad beträgt. Erinnern Sie sie daran, dass der Bereich bei 0 statt bei 1 beginnt, wodurch der Servo einen vollen 180-Grad-Bereich erhält.

TEACHER NOTES

Die **map()**-Funktion kann verwirrend sein, weil es fünf Parameter gibt, die innerhalb der Klammern stehen. Erinnern Sie die Kursteilnehmer daran, dass sie jederzeit zu den Ressourcen gehen können, wenn sie einen der in diesem Kurs vorgestellten Befehle oder eine der Funktionen wiederholen müssen.

14) Das Letzte, was passieren muss, ist, den Servo zu bewegen. Der Servobibliotheksbefehl zur Steuerung des Servos lautet **servoName.write(angle)**. In eurem Sketch lautet der **ServoName myServo**. Der **Winkel*** kann ein beliebiger Wert zwischen 0 und 179 Grad für ein Standardservo sein. Ihr verwendet die **map()**-Funktion, um die Variable **servoAngle** auf einen Wert in diesem Bereich zu setzen. Euer Befehl zum Bestimmen des Servowinkels sollte wie folgt aussehen:

15) Es wird dem Servo wahrscheinlich helfen, wenn ihr eine kurze Verzögerung am Ende der **void loop()**-Funktion hinzufügt. Diese Verzögerung hält das Programm an und gibt dem Servo Zeit, seine Position zu erreichen, bevor der Loop erneut beginnt. Fügt am Ende der **void loop()**-Funktion eine kurze Verzögerung von 100 Millisekunden hinzu.

16) Verifiziert euren Code und debugged, falls erforderlich.

17) Schließt das Arduino UNO R3 über das USB-Kabel an den Computer an.

18) Geht in der Arduino IDE zum Menü **Tools** und klickt dann auf **Port**. Wählt den mit eurem Arduino UNO R3 Board verbundenen Port aus.

19) Klickt auf die Schaltfläche **Hochladen**, um den Sketch auf das Arduino UNO R3 Board hochzuladen.

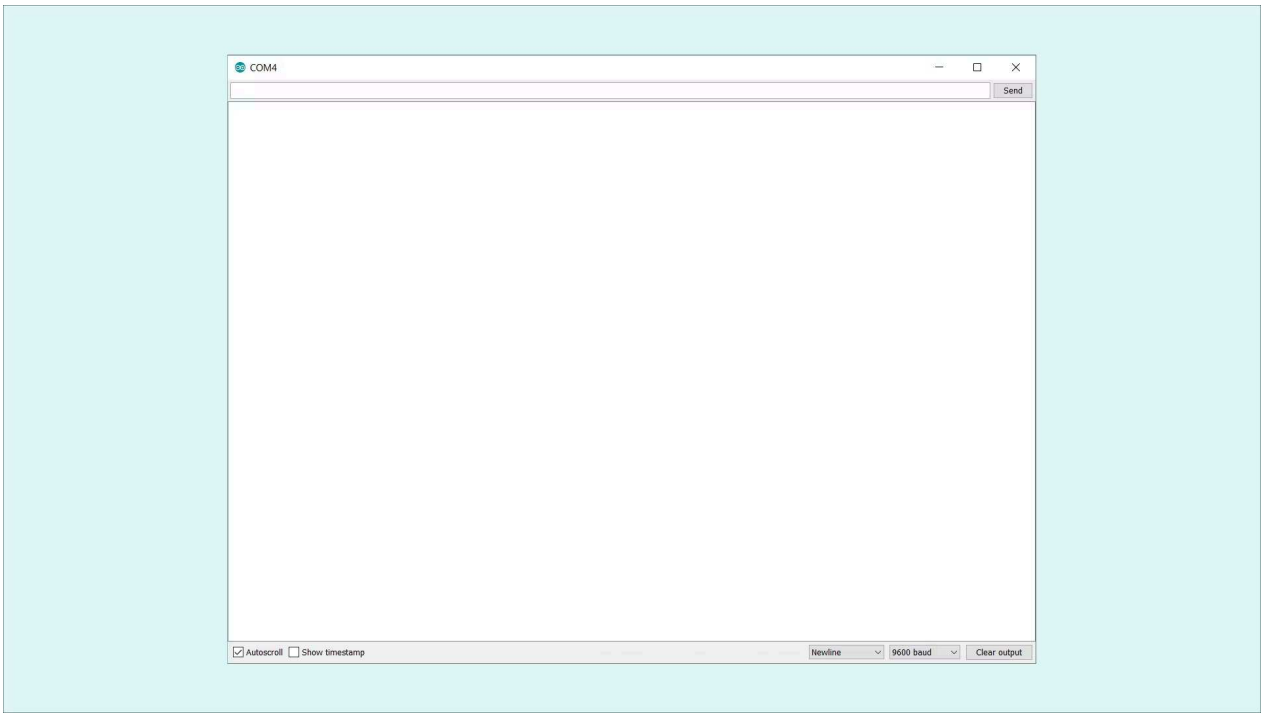
20) Nachdem die Skizze auf das Arduino UNO R3 Board übertragen wurde, dreht ihr das Potentiometer und beobachtet das Verhalten des Servomotors. Dreht das Potentiometer hin und her und beobachtet, wie der analoge Bereich des Potentiometers von 0 bis 1.023 auf den Bereich des Servomotors von 0 bis 179 Grad verkleinert wird.

21) Wenn ihr fertig seid, zieht ihr das USB-Kabel vom Arduino UNO R3 Board ab. Dadurch wird der Stromkreis abgeschaltet.

22) Klickt auf **Speichern**, um euren Sketch zu speichern.

Code-Erstellung - Serielle Kommunikation

Nun, da ihr über einen funktionierenden Schaltkreis verfügt, könnt ihr den seriellen Monitor verwenden, um weiter zu untersuchen, wie die Funktion **map()** Daten von einem Bereich in einen anderen ändert. In dieser Übung werdet ihr den seriellen Monitor verwenden, um die analogen Daten vom Potentiometer und die Inputdaten zum Servo zu verfolgen.



TEACHER NOTES

Bei dieser Übung verwenden die Schülerinnen und Schüler den seriellen Monitor, um ihre Eingabe- und Ausgabedaten zu überwachen und zu beobachten. Die Schülerinnen und Schüler überprüfen diese Konzepte:

- **Serial.begin()** – startet den seriellen Monitor
- **Serial.print()** und **Serial.println()** – gibt Daten von dem Arduino UNO R3 Board auf den seriellen Monitor des Computers aus

1) Öffnet gegebenenfalls euren Sketch zu Lektion6_V1 in der Arduino IDE.

2) Speichert den Sketch unter einem neuen Namen. Wählt im Dateimenü die Option **Speichern unter...** Benennt die Datei "Lektion6_V2_", gefolgt von euren Initialen und denen eures Partners. Klickt auf **Speichern**.

3) Schließt euren Lektion6*V1 Sketch. Ordnet das Sketchfenster von Lektion6*V2 und dieses Fenster so an, wie es für euch am besten geeignet ist.

4) Um den seriellen Monitor zu verwenden, müsst ihr eine Anweisung hinzufügen, die das Arduino UNO R3 Board anweist, den seriellen Monitor zu starten. Diese Anweisung geht normalerweise in den Abschnitt **void setup()** des Sketches. Gebt den folgenden Befehl in die **void setup()**-Funktion ein. Denkt daran, dass es eine gute Praxis ist, Code-Kommentare einzufügen.

5) Es gibt zwei Befehle, die zum Drucken von Informationen auf dem seriellen Monitor verwendet werden können:

- **Serial.print()** – druckt jeden Variablenwert oder Text, der in Anführungszeichen gesetzt ist, in dieselbe Zeile wie alle vorherigen Druckanweisungen
- **Serial.println()** – druckt Variablen oder Text, der in Anführungszeichen gesetzt ist, in dieselbe Zeile, beginnt aber nach dem Drucken der Informationen eine neue Zeile

Gebt die folgenden Befehle vor der **delay()**-Anweisung am Ende des Loops ein.

> **Hinweis:** Stellt sicher, dass euer letzter Druckbefehl **ln** vor den Klammern enthält, damit der serielle Monitor eine neue Zeile beginnt, nachdem er die Variable **servoAngle** gedruckt hat.

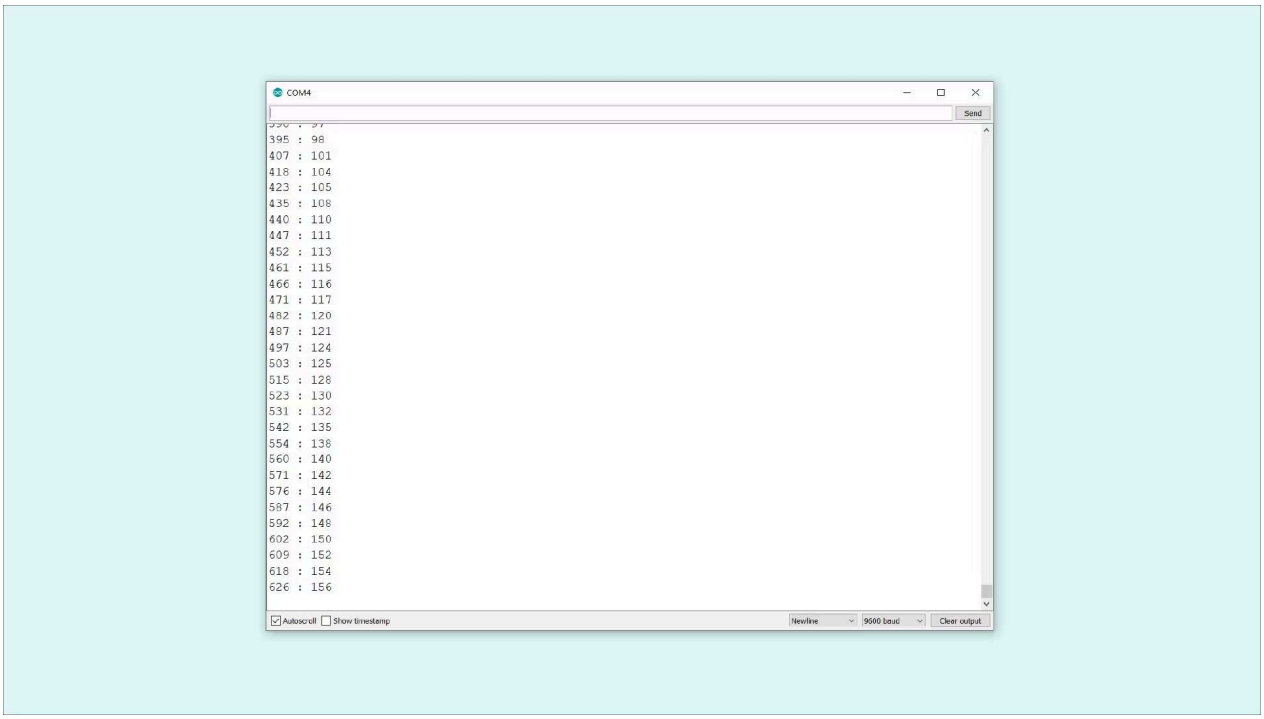
6) Überprüft euren Code und debugged, falls erforderlich.

7) Schließt das Arduino UNO R3 über das USB-Kabel an den Computer an.

8) Klickt auf die Schaltfläche **Hochladen**, um den Sketch auf euer Arduino UNO R3 Board hochzuladen.

9) Klickt auf die Schaltfläche **Serial Monitor** in der oberen rechten Ecke der Arduino IDE. Der serielle Monitor wird in einem neuen Fenster geöffnet.

10) Stellt in der rechten unteren Ecke sicher, dass die Baudrate auf 9600 eingestellt ist. Verwendet das Dropdown-Menü, um die Baudrate bei Bedarf zu ändern. Klickt auf **Ausgabe löschen**, um das serielle Monitorfenster zu löschen.



11) Dreht das Potentiometer, um den Winkel des Servos einzustellen. Beachtet beim Drehen des Potentiometers, wie sich die **PotValue**-Variable und die **ServoAngle**-Variable im seriellen Monitor ändern.

TEACHER NOTES

VERSTÄNDNIS-ÜBERPRÜFUNG

Die Funktion **map()** skaliert den Potentiometerbereich von 0 bis 1.023 auf den Servobereich von 0 bis 179. Ohne Verwendung des seriellen Monitors oder des Potentiometers sollten Sie prüfen, ob die Schülerinnen und Schüler den Servowinkel bestimmen können, wenn das Potentiometer auf einen Wert von 512 eingestellt ist. Lassen Sie sie dann den seriellen Monitor und das Potentiometer verwenden, um ihre Antwort zu überprüfen.

Ein Potentiometerwert von 512 liegt auf halbem Wege im analogen Bereich von 0 bis 1.023. Daher sollte der Servowinkel etwa auf halbem Wege im

Servobereich von 0 bis 179 liegen. Der Servowinkel sollte etwa 89 oder 90 Grad betragen.

12) Wenn ihr fertig seid, zieht ihr das USB-Kabel vom Arduino UNO R3 Board ab. Dadurch wird der Stromkreis abgeschaltet.

13) Klickt auf **Speichern**, um euren Sketch zu speichern.

Aufbau des Roboters

Jetzt, da ihr einen funktionierenden Servo habt, den ihr steuern könnt, ist es an der Zeit, eure Schaltung in etwas Nützlicheres zu verwandeln - in einen Roboter. In dieser Übung werdet ihr ein Gerät erstellen und an euren Servo anbringen, das einen Ball schlagen, treten oder werfen kann.

FURTHER NOTES

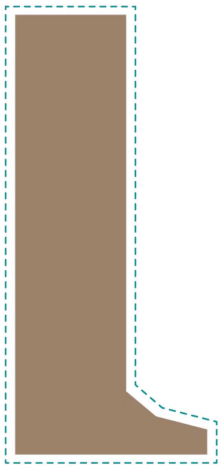
Für diese Übung müssen Sie den Schülern Pappe, Schere, Klebeband und eine Art Kugel, die sie mit ihrem Servo bewegen können, zur Verfügung stellen. Tischtennisbälle funktionieren aufgrund ihres geringen Gewichts gut. Sie können jedoch alles verwenden, was Ihnen zur Verfügung steht.

1) Falls noch nicht geschehen, trennt ihr das Arduino UNO R3 Board vom USB-Kabel. Dadurch wird die Stromversorgung des Schaltkreises unterbrochen.

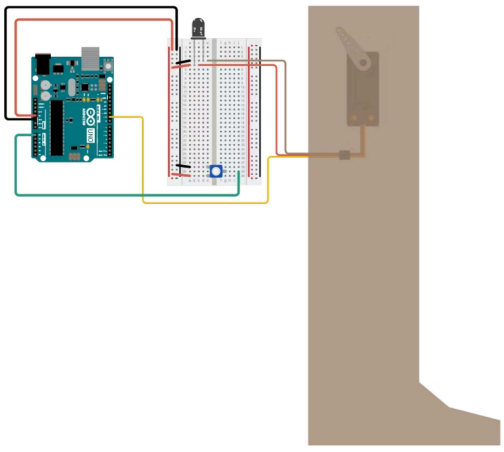
2) Entscheidet zusammen mit einem Partner, welche Art von Sport-Roboter ihr erschaffen möchtet. Hier sind einige Ideen, die ihr berücksichtigen solltet:

- Ein Fußball- oder Footballroboter, der den Ball kickt
- Ein Hockey-Roboter, der den Ball wie einen Puck schlägt
- Ein Baseball- oder Softball-Roboter, der den Ball schlägt oder wie ein Katapult wirft
- Ein Basketball-Roboter, der den Ball schießt (Dies könnte aufgrund der Geschwindigkeit des Servos schwierig sein).

3) Euer Lehrer wird euch Karton, Schere, Klebeband und einen Ball zur Verfügung stellen. Basierend auf der von euch gewählten Sportart erstellt ihr mit dem Karton und der Schere eine Servobefestigung, die den Ball bewegen kann. Wenn ihr euch zum Beispiel für Fußball entscheidet, können ihr ein Bein mit einem Fuß erstellen, das den Ball treten kann.

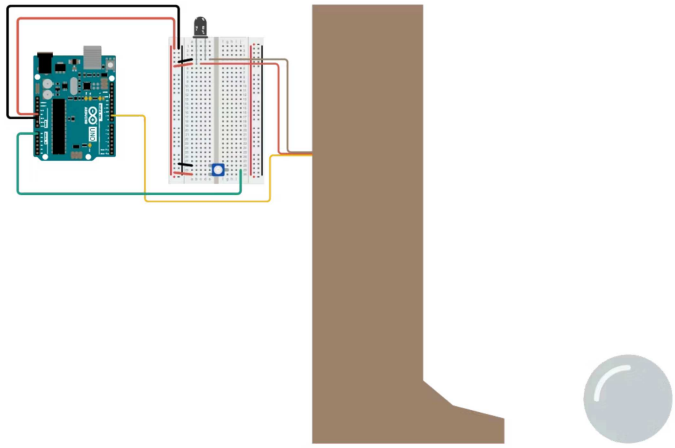


4) Befestigt euer Gerät mit Klebeband am Servohorn des Servos.



5) Wenn euer Servoanbau abgeschlossen ist, schließt ihr das USB-Kabel an das Arduino UNO R3 Board an. EUer Servosteuerungs-Sketch sollte automatisch ablaufen.

6) Übt das Treten, Schlagen oder Werfen des Balls mit Hilfe des Potentiometers, um die Bewegung des Servos zu steuern.



TEACHER NOTES

KLASSENDISKUSSION

Lassen Sie die Schülerinnen und Schüler über die Leistung ihres Sportroboters diskutieren:

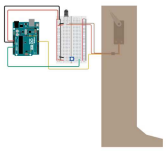
- Ist es schwierig, den Roboter dazu zu bringen, den Ball genau zu schlagen, zu treten oder zu werfen? Warum?
- Welche Änderungen könnten an dem Roboter vorgenommen werden, um seine Leistung zu verbessern?
- Können sie sich einen Roboter vorstellen, der einen Menschen in der gewählten Sportart schlagen könnte?

Die Schülerinnen und Schüler sollten feststellen, dass ihr Roboter weit davon entfernt ist, einen Menschen in der von ihnen gewählten Sportart zu ersetzen. Wenn die Schülerinnen und Schüler darüber diskutieren, wie sie ihren Roboter verbessern könnten, könnten sie die Verwendung von Sensoren zur Erkennung des Balls oder der Gegner und die Verwendung von Motoren zur Bewegung des Roboters auf dem Spielfeld erwähnen.

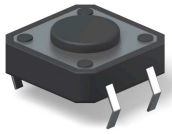
Testen und ändern

In der vorherigen Übung habt ihr einen Roboter gebaut, der einen Ball schlägt, tritt oder wirft. Ihr habt wahrscheinlich bemerkt, dass ihr nicht viel Kontrolle darüber habt, wie er dies tut. In dieser Übung werdet ihr euren Roboter modifizieren, um seine Spielfähigkeit zu verbessern. Dazu fügt ihr einen Tastschalter hinzu, der die Bewegung des Roboters steuert, und stellt mit dem Potentiometer ein, wie weit sich der Roboter beim Schlagen, Schiessen oder Werfen dreht.

Benötigtes Material

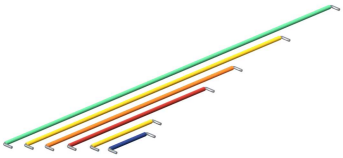


1 ROBOTERSCHALTUNG AUS VORHERIGER ÜBUNG



1 TASTSCHALTER

1 10K Ω WIDERSTAND



1 STECKVERBINDER

1 USB-KABEL

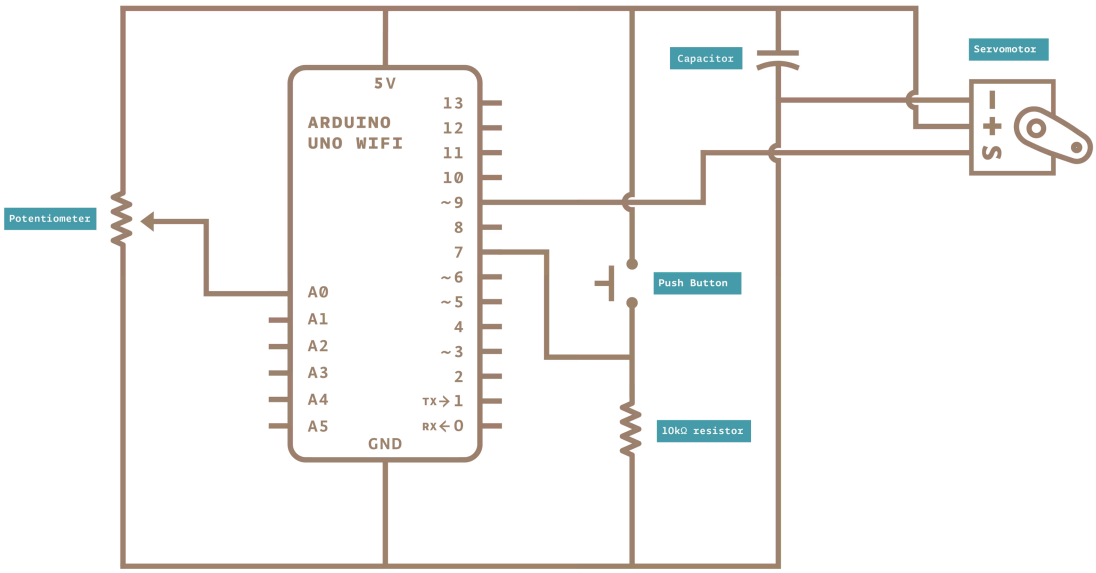
FURTHER NOTES

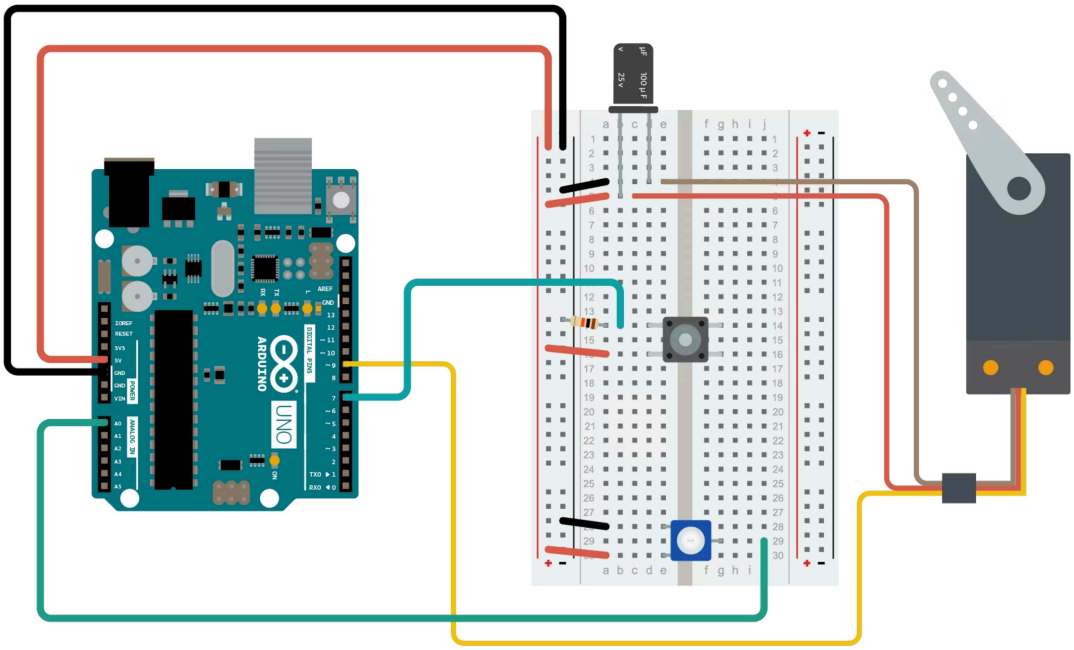
Die Farbbänder des 10K-Ohm-Widerstandes sind:

- **4 Bänder:** braun, schwarz, orange, gold

- **5 Bänder:** braun, schwarz, schwarz, rot, gold

Schaltplan



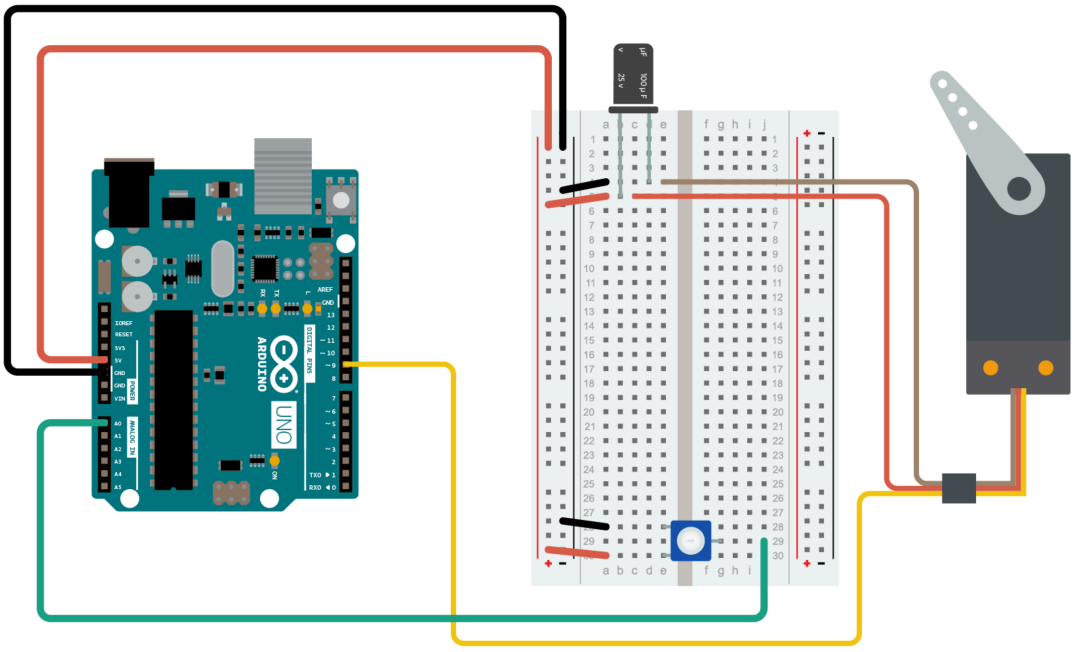


Ändern des Schaltkreises

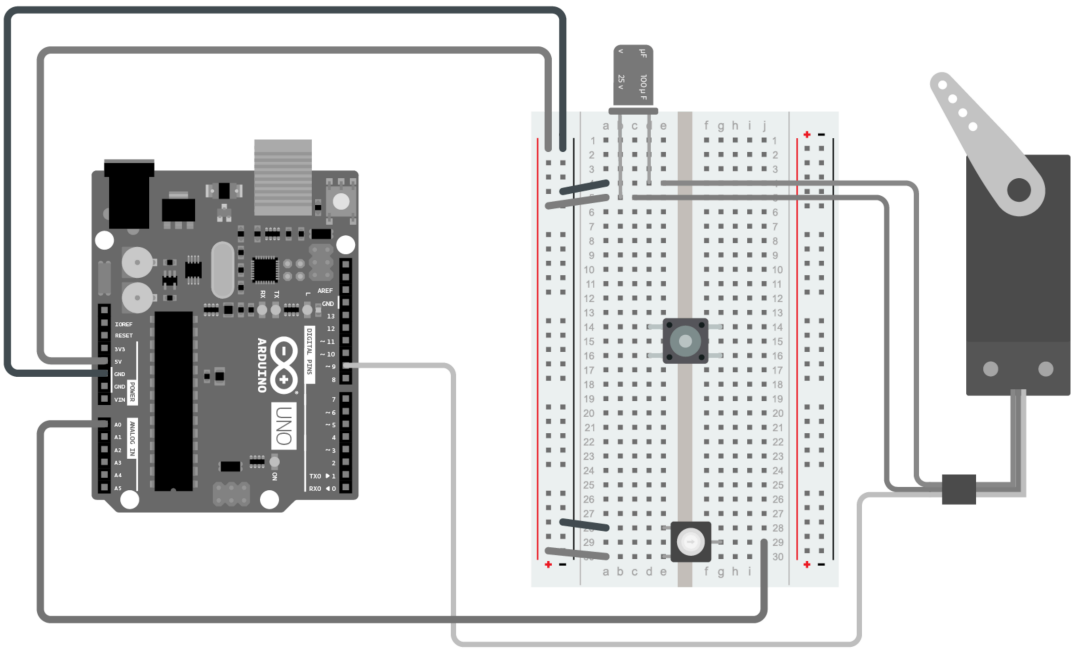
Verwendet das Schaltbild und den Schaltplan oder die folgende Diashow, um eure Sportroboterschaltung zu modifizieren.

Step 1

Beginnt mit dem Roboter aus der vorherigen Aktivität.

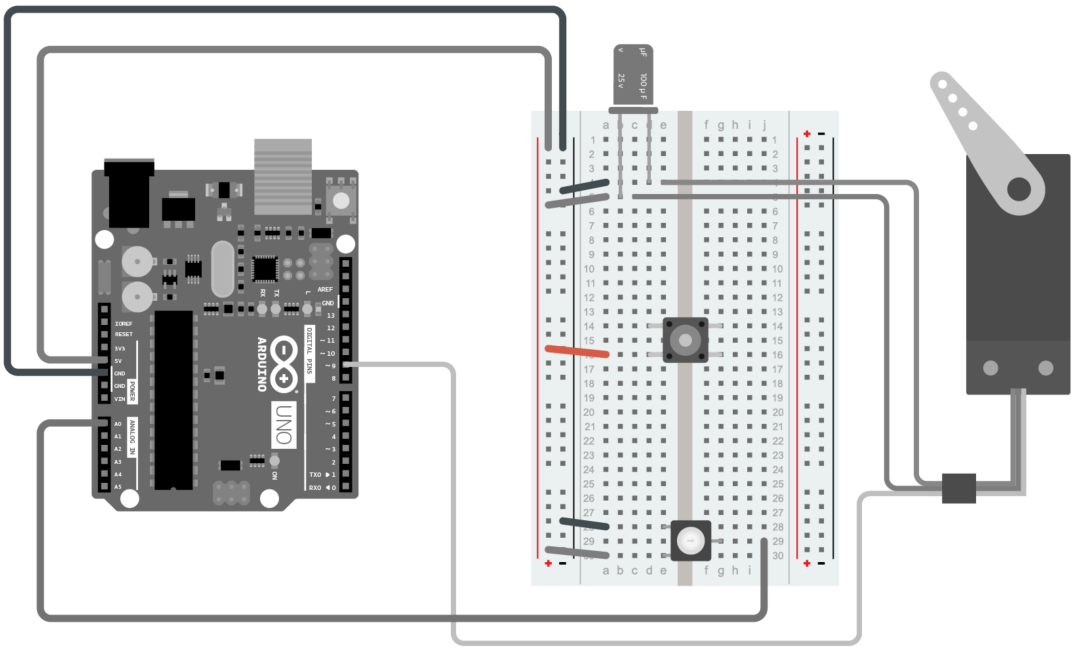


*Step 2 * Fügt einen Tastschalter hinzu, der die Lücke im Steckbrett überbrückt. Bringt den Tastschalter irgendwo in der Mitte des Steckbretts an.



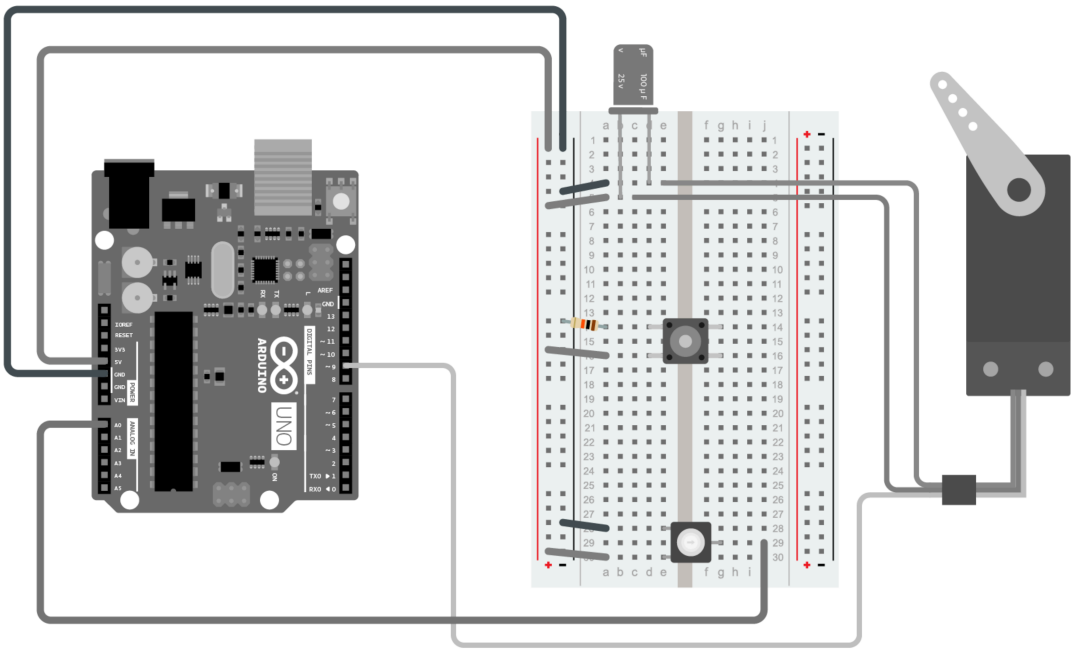
Step 3

Verwendet einen kurzen Steckverbinder, um den unteren Stift des Druckknopfes mit dem positiven Anschluss zu verbinden. Dadurch wird der Taster mit fünf Volt Spannung versorgt.



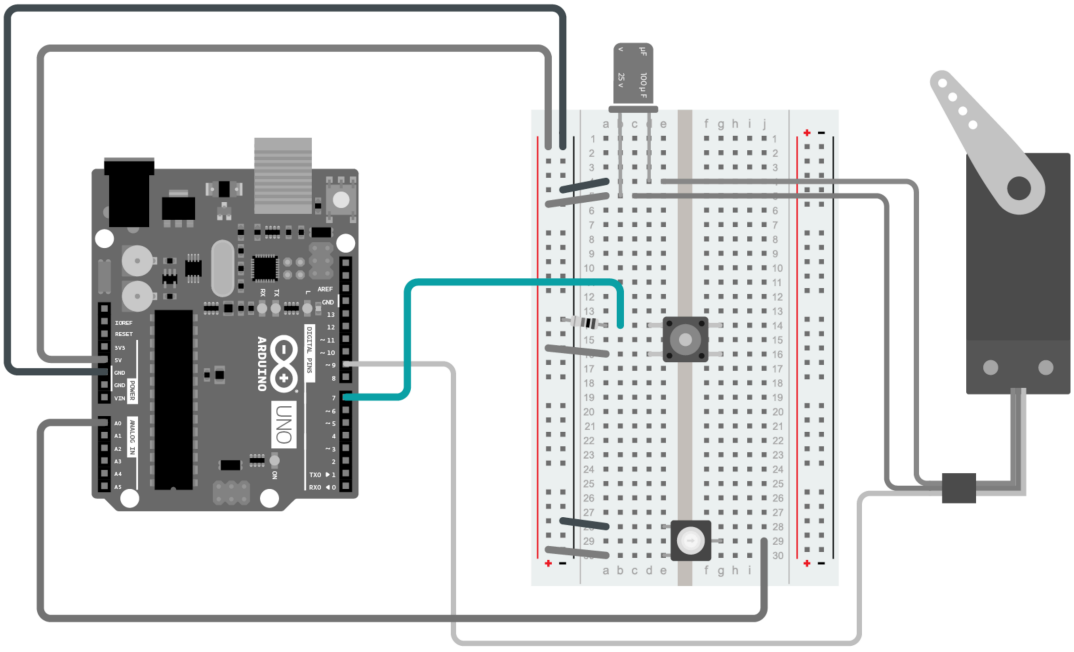
Step 4

Schließt einen 10K-Ohm-Widerstand an das Steckbrett an, so dass der obere Stift des Tastschalters über den Widerstand mit Ground verbunden ist. Dieser Widerstand fungiert als Pulldown-Widerstand.



Step 5

Verwendet einen langen Steckverbinger, um den oberen Pin des Tastschalters mit Pin 7 auf dem Arduino UNO R3 Board zu verbinden. Auf diese Weise könnt ihr den Status von Pin 7 ablesen und feststellen, ob der Knopf gedrückt ist oder nicht.



Ändern des Codes

Nun, da ihr eurer Schaltung einen Tastschalter hinzugefügt habt, müsst ihr euren Code so ändern, dass ihr lesen könnt, ob der Tastschalter gedrückt ist oder nicht.

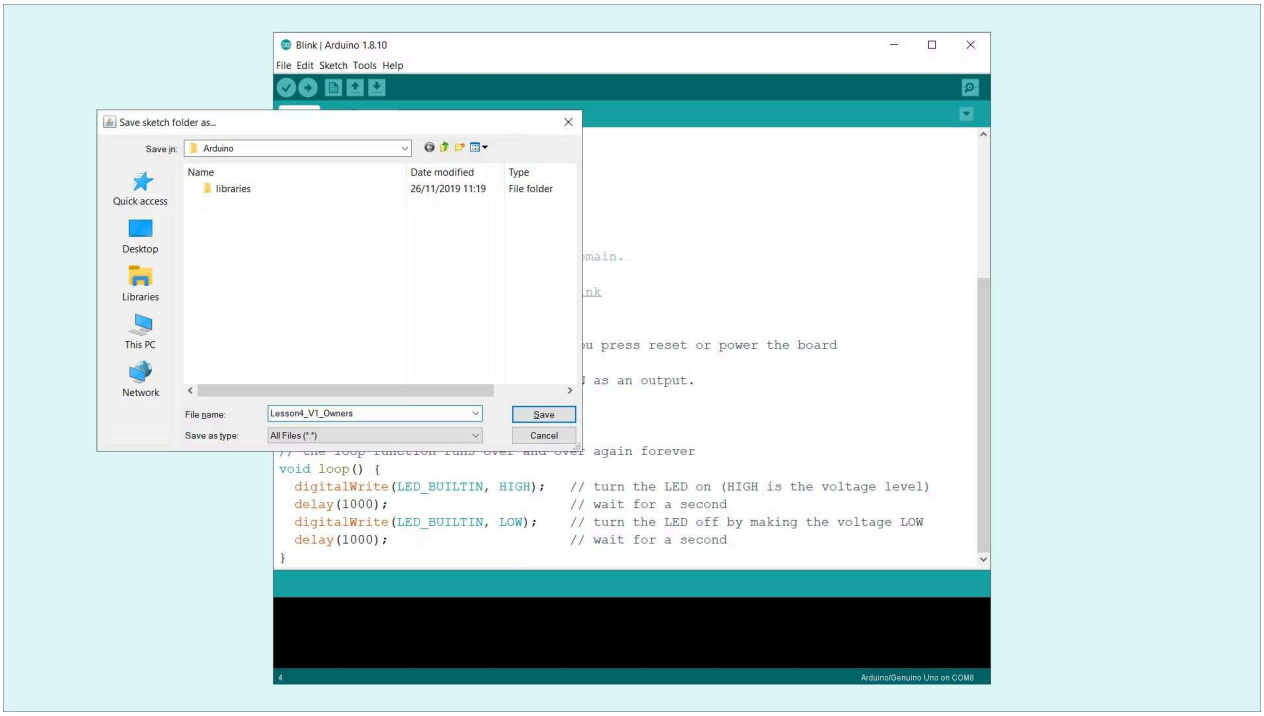
TEACHER NOTES

Während dieser Übung überprüfen die Schülerinnen und Schüler diese Programmierbefehle und -funktionen:

- **if(Bedingung){} else {}** – Wenn-sonst bedingte Struktur
- **Konstantendeklarationen –** Deklariert den Typ und den Namen einer Konstanten
- **servoName.attach()** – Ein Befehl aus der Servo-Bibliothek, der ein Servo-Objekt mit einem physikalischen Pin auf dem Arduino UNO R3 Board verbindet

1) Öffnet gegebenenfalls euren Sketch zu Lektion6_V2 in der Arduino IDE.

2) Speichert die Skizze unter einem neuen Namen. Wähl im Dateimenü die Option **Speichern unter…** Benennt die Datei "Lektion6_V3_", gefolgt von euren Initialen und denen eures Partners. Klickt auf **Speichern**.



3) Schließt euren Lektion6*V2-Sketch. Ordnet das Sketchfenster von Lektion6*V3 und dieses Fenster so an, wie es für euch am besten geeignet ist.

4) Genau wie für den Potentiometer-Pin könnt ihr eine Konstante bestimmen, die den Pin auf dem Arduino UNO R3 Board beschreibt, an den der Tastschalter

angeschlossen ist. Auf diese Weise könnt ihr den Pin später bei Bedarf leicht ändern. Sie gibt dem Pin auch einen aussagekräftigeren Namen, so dass jeder andere, der sich euren Code ansieht, weiß, welche Art von Gerät mit diesem Pin verbunden ist.

In eurer Schaltung sollte euer Tastschalter mit Pin 7 auf dem Arduino UNO R3 Board verbunden sein. Erstellt oben in eurem Sketch vor der Funktion **void setup()** eine ganzzahlige Konstante namens **buttonPin** und weist ihr für Pin 7 den Wert 7 zu.

TEACHER NOTES

Erinnern Sie die Schülerinnen und Schüler daran, bei jeder neuen Codezeile, die sie ihrer Skizze hinzufügen, Codekommentare hinzuzufügen, die ihren Code erklären.

5) Geht zur Funktion **void loop()** in eurem Sketch. Für diese Schaltung müsst ihr den Setch so programmieren, dass der Servo beim Drücken der Taste bewegt wird. Als Erstes müsst ihr also feststellen, ob der Taster gedrückt wird oder nicht. Dazu ist ein "Wenn-Sonst"-bedingtes Statement erforderlich. Erstellt die if-else-Bedingungsstruktur am Anfang eurer **void loop()**-Funktion. Sie sollte wie folgt aussehen:

6) Wenn in eurem Schaltkreis der Tastschalter gedrückt wird, wird der digitale Pin 7 als HIGH (Spannung größer als 3 Volt) gelesen. Wenn der Tastschalter nicht gedrückt wird, wird er als LOW (Spannung unter 1,5 Volt) angezeigt. Der Befehl **digitalRead(pin)** wird verwendet, um festzustellen, ob ein digitaler Pin HIGH oder LOW ist.

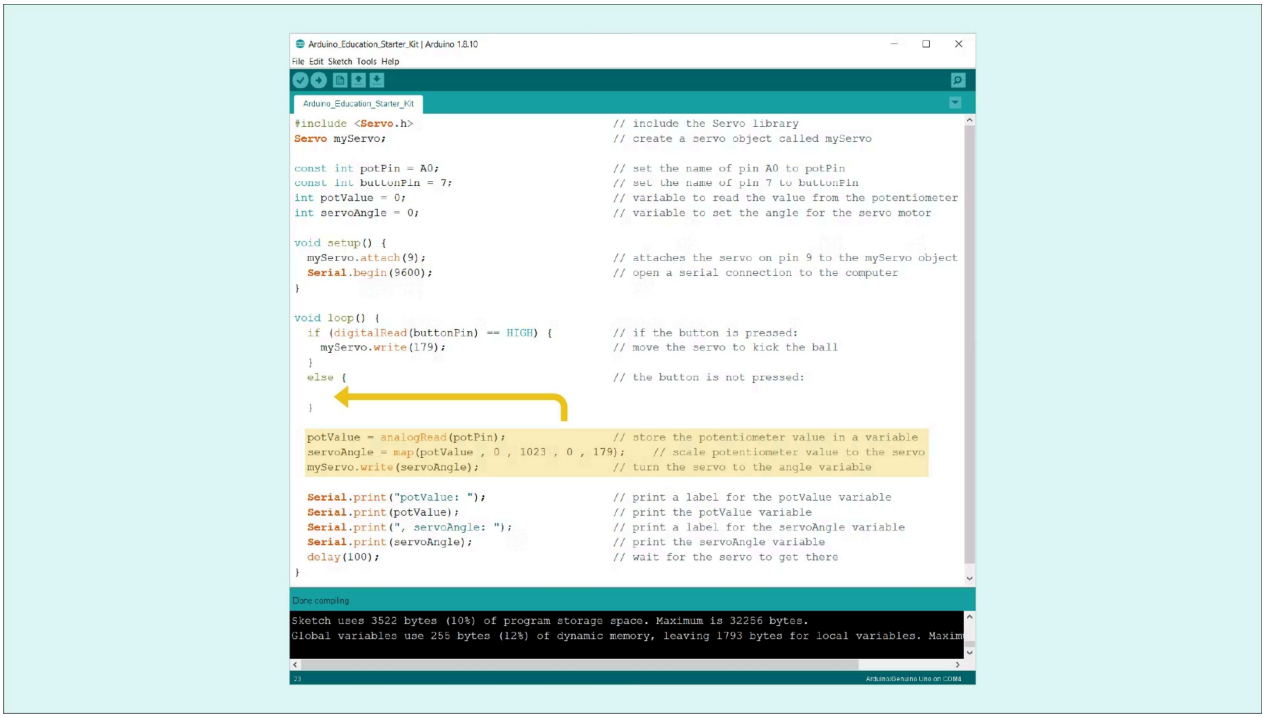
Erinnert euch daran, dass der Bedingungsteil der "Wenn-Sonst"-Anweisung innerhalb der Klammern steht. Die Bedingung vergleicht immer zwei Werte. Ihr müsst den Wert des digitalen Pins 7 lesen und ihn mit HIGH vergleichen, um festzustellen, ob der Taster gedrückt ist. Denkt auch daran, dass ihr Pin 7 **buttonPin** am Anfang des Sketches als Konstante benannt habt. Gebt die Bedingung für das if-Statement wie unten gezeigt ein.

Hinweis: Achtet besonders auf eure Klammern. Hier werden zwei Sätze verwendet - einer enthält den Bedingungsteil der if-Anweisung und einer den **ButtonPin**-Namen für den Befehl **digitalRead()**.

7) Wenn der Taster gedrückt wird, muss sich der Servo in seine äußerste Position bewegen, die 179 Grad beträgt. Fügt einen Servo-Schreibbefehl zwischen den geschweiften Klammern für das if-Statement ein. Dies sollte den if-Teil eures if-else-Statements vervollständigen.

8) Geht zum anderen Teil eures if-Statements über. Dies ist der Teil des Sketches, der die Befehle für den Fall enthält, dass der Taster nicht gedrückt wird. In diesem Fall wird das Potentiometer verwendet, um die Entfernung einzustellen, die das Servo bei Tritten, Schlägen oder Würfen zurücklegt. Die gute Nachricht ist, dass ihr die dafür benötigten Befehle bereits habt. Sie müssen nur noch in eurem Sketch verschoben werden.

Verschiebt die Befehle, die die Variable **potValue**, die Variable **servoAngle** und die Servoposition angeben, zwischen die geschweiften Klammern für den anderen Teil eurer bedingten Anweisung. Ihr könnt diese drei Befehle markieren und ziehen, durch Ausschneiden und Einfügen verschieben oder einfach in den anderen Teil der Bedingung neu eingeben. Wenn ihr sie jedoch neu eintippt, stellt sicher, dass ihr sie im unteren Teil löscht.



9) Verifiziert euren Code und debugged, falls erforderlich. Euer Sketch sollte ähnlich wie dieser aussehen:

10) Schließt das Arduino UNO R3 über das USB-Kabel an den Computer an.

11) Geht in der Arduino IDE zum Menü **Tools** und klickt dann auf **Port**. Wählt den mit eurem Arduino UNO R3 Board verbundenen Port aus.

12) Klickt auf die Schaltfläche **Hochladen**, um den Sketch auf das Arduino UNO R3 Board hochzuladen.

13) Nachdem der Sketch auf das Arduino UNO R3 Board übertragen wurde, dreht ihr das Potentiometer, um den Abstand der Schlag/Tritt/Wurfbewegung eures Roboters einzustellen. Drückt die Taste, um die Bewegung auszuführen.

14) Klickt auf **Speichern**, um euren Sketch zu speichern.

Testen des Schaltkreises

Nun, da ihr euren Sport-Roboter gebaut und programmiert habt, ist es an der Zeit, mit ihm zu experimentieren. Für diese Übung werdet ihr euren Sportroboter verwenden, um zu testen, wie sich eine Änderung des Drehwinkels des Servos auf die

Entfernung auswirkt, die der Ball zurücklegt, wenn euer Roboter den Ball schlägt, tritt oder wirft. Ihr braucht euren Sport-Roboter, etwas, um die Entfernung zu messen, zum Beispiel ein Lineal, Zollstock oder Maßband und einen kleinen Ball, den ihr schlagen, treten oder werfen könnt. Außerdem benötigt ihr euer Arbeitsheft.

TEACHER NOTES

Bei dieser Übung führen die Schülerinnen und Schüler ein Experiment durch, um zu sehen, wie sich eine Änderung der Entfernung, um die sich das Servo dreht, auf die Entfernung auswirkt, um die sich der Ball auf dem Boden bewegt. Dieses Experiment funktioniert am besten, wenn es auf einer weichen Oberfläche durchgeführt wird, die mehr Reibung bietet, um den Ball abzubremsen und ihn daran zu hindern, lange Strecken zu rollen. Ein kurzer Teppich, eine Schaumstoffmatte oder ein Stück Stoff, wie ein flach ausgebreitetes Badetuch, sollten gut funktionieren. Die Schülerinnen und Schüler müssen diese Übung möglicherweise auf dem Boden durchführen.

Badetücher sind eine einfache Möglichkeit, die harte Oberfläche eines Schreibtisches in eine weiche Oberfläche zu verwandeln, die genügend Reibung bietet, um die Bälle daran zu hindern, sich zu weit zu bewegen.

Sie müssen den Schülerinnen und Schülern auch ein Messgerät zur Verfügung stellen. Zollstöcke oder Maßbänder werden wahrscheinlich am besten funktionieren, insbesondere wenn die Bälle weiter rollen als der Abstand eines Lineals, aber Sie können Lineale verwenden, wenn sie die einzige verfügbare Option sind. Schaumstoff-Golfbälle oder Ping-Pong-Bälle eignen sich gut für diese Übung.

1) Zuerst müsst ihr eure Experimentierbahn einrichten.

2) Bildet an einem Ende der Bahn eine Startlinie für den Ball. Dies kann durch Hinlegen eines Stiftes oder Bleistifts geschehen. Auf diese Weise wird ein einheitlicher Startplatz für den Ball geschaffen. Der Stift oder Bleistift sollte während des Versuchs nicht bewegt werden.

3) Stellt sicher, dass euer Sport-Roboter an den USB-Anschluss eures Computers angeschlossen ist und euer Sketch läuft.

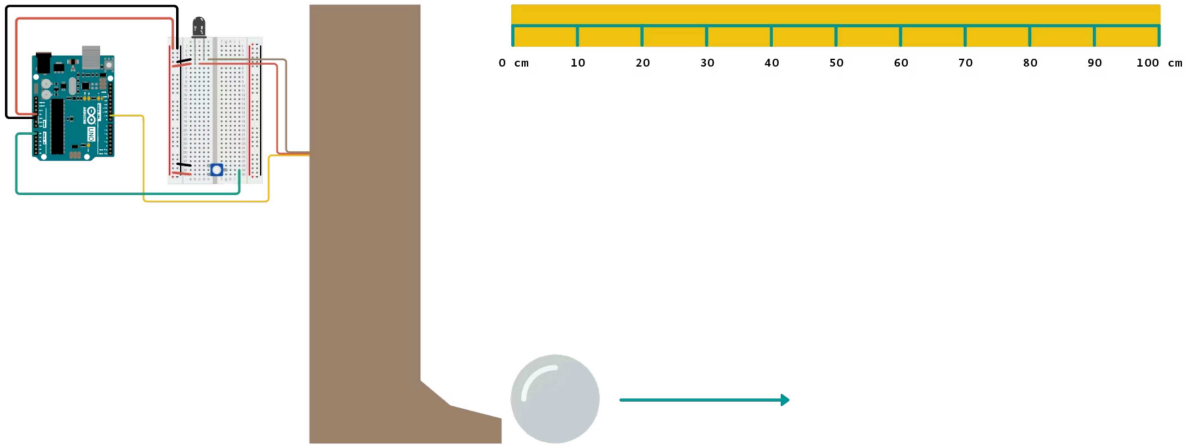
4) Startet den seriellen Monitor, damit ihr den Potentiometerwert und den Servowinkel beobachten könnt.

5) Euer Arbeitsheft enthält eine Datentabelle, die ihr bei der Durchführung des Experiments ausfüllen werdet. Dreht das Potentiometer und lest den Wert auf dem

seriellen Monitor ab. Stellt den Potentiometerwert und den Servowinkel so ein, dass sie in dem in der Datentabelle angegebenen Bereich liegen.

6) Platziert den Ball an der Startlinie.

7) Platziert den Roboterarm/-bein so, dass er den Ball kaum berührt.



8) Drückt den Taster eures Roboters, um den Ball in Richtung des anderen Endes der Experimentierbahn zu schlagen/zu treten/zu werfen.

9) Messt die Entfernung, die der Ball zurücklegt. Verwendet die Einheiten, die aufgrund eures Messgeräts und der Entfernung, die der Ball zurücklegt, am Sinnvollsten sind. Tragt die Entfernung (einschließlich der von euch verwendeten Einheiten) in euer Arbeitsheft ein.

10) Zeichnet alle Beobachtungen über die Leistung des Roboters oder die Bewegung des Balls im vorgesehenen Raum auf.

11) Wiederholt die Schritte 5-10, um die Datentabelle zu vervollständigen.

12) Welche Schlussfolgerungen könnt ihr aus den Ergebnissen eures Experiments ziehen? Besteht ein Zusammenhang zwischen dem Servowinkel und der Entfernung, die der Ball zurückgelegt hat? Wenn ja, wie ist diese Beziehung? Tragt eure Schlussfolgerungen in den dafür vorgesehenen Platz in eurem Arbeitsheft ein.

13) Welche Faktoren hätten einen größeren Einfluss darauf, wie weit der Roboter den Ball schlägt, kickt oder wirft? Überlegt, wie ihr diese Faktoren mit Code in einem Sketch kontrollieren könntet. Notiert eure Ideen an der dafür vorgesehenen Stelle in eurem Arbeitsheft.

14) Schließt alle Fenster der Arduino IDE.

15) Räumt euren Arbeitsbereich auf und lagert alle Geräte in einem geeigneten Lagerbereich.

TEACHER NOTES

WEITERE ÜBUNGEN

- **Servogeschwindigkeit** – Fordern Sie die Schülerinnen und Schüler auf, ihren Code zu ändern, um die Geschwindigkeit des Servos zu verlangsamen. Eine Möglichkeit, dies zu tun, wäre, entweder einen while-Loop oder einen for-Loop zu verwenden, um den Servo mit kurzen Verzögerungen zwischen den einzelnen Schritten zu bewegen. Möglicherweise müssen die Schülerinnen und Schüler nach while loop und for loop forschen, da diese erst in Lektion 7 eingeführt werden.
- **Forschung** – Lassen Sie die Schülerinnen und Schüler über Roboter, die für den Sport entwickelt werden, forschen und darüber berichten.
- **Brainstorming** – Lassen Sie die Schülerinnen und Schüler per Brainstorming eine Liste anderer Anwendungen neben Sport-Robotern erstellen, die Servos zur Bewegungssteuerung verwenden könnten.