

Super Starter Kit

ebook

Super Starter Kit

mit atmega328-Mikrocontroller-Platine



Anwendungsbereiche

Bildung und Lehre: Einsatz in Schulen, Hochschulen und Ausbildungseinrichtungen zur Vermittlung von Grundlagen der Elektronik, Programmierung und eingebetteten Systemen. Forschung und Entwicklung: Verwendung in Forschungs- und Entwicklungsprojekten zur Erstellung von Prototypen und Experimenten in den Bereichen Elektronik und Informatik. Prototypenentwicklung: Einsatz in der Entwicklung und Erprobung neuer elektronischer Schaltungen und Geräte. Hobby und Maker-Projekte: Verwendung durch Elektronikenthusiasten und Hobbyisten zur Entwicklung und Umsetzung von DIY-Projekten.

Erforderliche Kenntnisse und Fähigkeiten

Grundlegendes Verständnis der Elektronik und Elektrotechnik. Kenntnisse in der Programmierung, insbesondere in der Programmiersprache C/C++. Fähigkeit, Schaltpläne zu lesen und einfache Schaltungen zu entwerfen. Erfahrung im Umgang mit elektronischen Komponenten und Löten.

Betriebsbedingungen

Das Produkt darf nur mit den im Datenblatt spezifizierten Spannungen betrieben werden, um Beschädigungen zu vermeiden. Eine stabilisierte Gleichstromquelle ist zum Betrieb erforderlich. Bei der Verbindung mit anderen elektronischen Komponenten und Schaltungen sind die maximalen Strom- und Spannungsgrenzen zu beachten, um Überlastungen und Schäden zu vermeiden.

Umweltbedingungen

Das Produkt sollte in einer sauberen, trockenen Umgebung verwendet werden, um Schäden durch Feuchtigkeit oder Staub zu vermeiden. Schützen Sie das Produkt vor direkter Sonneneinstrahlung (UV)

Bestimmungsgemäße Verwendung

Das Produkt ist für den Einsatz in Bildungs-, Forschungs- und Entwicklungsumgebungen konzipiert. Es dient zur Entwicklung, Programmierung und Prototypenentwicklung von elektronischen Projekten und Anwendungen. Das Sensorprodukt ist nicht als fertiges Verbraucherprodukt gedacht, sondern als Werkzeug für technisch versierte Nutzer, darunter Ingenieure, Entwickler, Forscher und Studenten.

Nicht bestimmungsgemäße vorhersehbare Verwendung

Das Produkt eignet sich nicht für den industriellen Einsatz oder sicherheitsrelevante Anwendungen. Eine Verwendung des Produkts in Medizingeräten oder für Zwecke der Luft- und Raumfahrt ist nicht zulässig

Entsorgung

Nicht mit dem Hausmüll entsorgen! Ihr Produkt ist entsprechend der europäischen Richtlinie über Elektro- und Elektronik-Altgeräte umweltgerecht zu entsorgen. Die darin enthaltenen, wertvollen Rohstoffe können so der Wiederverwendung zugeführt werden. Die Anwendung dieser Richtlinie trägt zum Umwelt- und Gesundheitsschutz bei. Nutzen Sie die von Ihrer Kommune eingerichtete Sammelstelle zur Rückgabe und Verwertung elektrischer und elektronischer Altgeräte. WEEE-Reg.-Nr.: DE 62624346

Elektrostatische Entladung

Achtung: Elektrostatische Entladungen können das Produkt beschädigen. Hinweis: Erden Sie sich, bevor Sie das Produkt berühren, indem Sie beispielsweise ein antistatisches Armband tragen oder eine geerdete Metalloberfläche berühren.

Sicherheitshinweise

Obwohl unser Produkt den Anforderungen der RoHS-Richtlinie (2011/65/EU) entspricht und keine gefährlichen Stoffe in über den Grenzwerten zulässigen Mengen enthält, können dennoch Rückstände vorhanden sein. Beachten Sie die folgenden Sicherheitshinweise, um chemische Gefährdungen zu vermeiden: Achtung: Beim Löten können Dämpfe entstehen, die gesundheitsschädlich sein können. Hinweis: Verwenden Sie einen Lötdampfabsauger oder arbeiten Sie in einem gut belüfteten Bereich. Tragen Sie gegebenenfalls eine Atemschutzmaske. Achtung: Einige Personen könnten empfindlich auf bestimmte Materialien oder Chemikalien reagieren, die im Produkt enthalten sind. Hinweis: Sollten Hautreizungen oder allergische Reaktionen auftreten, unterbrechen Sie die Nutzung und suchen Sie gegebenenfalls einen Arzt auf. Achtung: Halten Sie das Produkt außerhalb der Reichweite von Kindern und Haustieren, um versehentlichen Kontakt und Verschlucken von Kleinteilen zu vermeiden. Hinweis: Bewahren Sie das Produkt in einem sicheren, geschlossenen Behälter auf, wenn es nicht verwendet wird. Achtung: Vermeiden Sie den Kontakt des Produkts mit Nahrungsmitteln und Getränken. Hinweis: Lagern und verwenden Sie das Produkt nicht in der Nähe von Lebensmitteln, um eine Kontamination zu verhindern. Obwohl unser Produkt den

Anforderungen der RoHS-Richtlinie (2011/65/EU) entspricht und keine gefährlichen Stoffe in über den Grenzwerten zulässigen Mengen enthält, können dennoch Rückstände vorhanden sein. Beachten Sie die folgenden Sicherheitshinweise, um chemische Gefährdungen zu vermeiden: **Achtung:** Beim Lötten können Dämpfe entstehen, die gesundheitsschädlich sein können. **Hinweis:** Verwenden Sie einen Lötdampfabsauger oder arbeiten Sie in einem gut belüfteten Bereich. Tragen Sie gegebenenfalls eine Atemschutzmaske. **Achtung:** Einige Personen könnten empfindlich auf bestimmte Materialien oder Chemikalien reagieren, die im Produkt enthalten sind. **Hinweis:** Sollten Hautreizungen oder allergische Reaktionen auftreten, unterbrechen Sie die Nutzung und suchen Sie gegebenenfalls einen Arzt auf. **Achtung:** Halten Sie das Produkt außerhalb der Reichweite von Kindern und Haustieren, um versehentlichen Kontakt und Verschlucken von Kleinteilen zu vermeiden. **Hinweis:** Bewahren Sie das Produkt in einem sicheren, geschlossenen Behälter auf, wenn es nicht verwendet wird. **Achtung:** Vermeiden Sie den Kontakt des Produkts mit Nahrungsmitteln und Getränken. **Hinweis:** Lagern und verwenden Sie das Produkt nicht in der Nähe von Lebensmitteln, um eine Kontamination zu verhindern. Das Produkt enthält empfindliche elektronische Komponenten und scharfe Kanten. Unsachgemäßer Umgang oder Montage kann zu Verletzungen oder Beschädigungen führen. Beachten Sie die folgenden Sicherheitshinweise, um mechanische Gefährdungen zu vermeiden: **Achtung:** Die Platine und die Anschlüsse des Produkts können scharfe Kanten aufweisen. Gehen Sie vorsichtig vor, um Schnittverletzungen zu vermeiden. **Hinweis:** Tragen Sie bei der Handhabung und Montage des Produkts geeignete Schutzhandschuhe. **Achtung:** Vermeiden Sie übermäßigen Druck oder mechanische Belastung der Platine und der Komponenten. **Hinweis:** Montieren Sie das Produkt nur auf stabilen und ebenen Oberflächen. Verwenden Sie geeignete Abstandshalter und Gehäuse, um mechanische Belastungen zu minimieren. **Achtung:** Stellen Sie sicher, dass das Produkt sicher befestigt ist, um unbeabsichtigtes Verrutschen oder Herunterfallen zu verhindern. **Hinweis:** Verwenden Sie passende Unterlage oder eine sichere Befestigung in Gehäusen oder auf Montageplatten. **Achtung:** Achten Sie darauf, dass alle Kabelverbindungen sicher und korrekt angeschlossen sind, um Zugbelastungen und versehentliches Herausziehen zu vermeiden. **Hinweis:** Führen Sie Kabel so, dass sie nicht unter Spannung stehen und keine Stolpergefahr darstellen. Das Produkt arbeitet mit elektrischen Spannungen und Strömen, die bei unsachgemäßem Gebrauch zu elektrischen Schlägen, Kurzschlüssen oder anderen Gefahren führen können. Beachten Sie die folgenden Sicherheitshinweise, um elektrische Gefährdungen zu vermeiden: **Achtung:** Verwenden Sie das Produkt nur mit den spezifizierten Spannungen. **Hinweis:** Die Leistungsgrenzen des Produkts finden Sie im dazugehörigen Datenblatt **Achtung:** Vermeiden Sie Kurzschlüsse zwischen den Anschlüssen und Komponenten des Produkts **Hinweis:** Achten Sie darauf, dass keine leitenden Objekte die Platine berühren oder überbrücken. Verwenden Sie isolierte Werkzeuge und beachten Sie die Anordnung der Verbindungen. **Achtung:** Führen Sie keine Arbeiten am Produkt durch, wenn es mit einer Stromquelle verbunden ist. **Hinweis:** Trennen Sie das Produkt von der Stromversorgung, bevor Sie Änderungen an der Schaltung vornehmen oder Komponenten anschließen bzw. entfernen. **Achtung:** Überschreiten Sie nicht die spezifizierten Stromstärken für die Ein- und Ausgänge des Produkts. **Hinweis:** Die Leistungsgrenzen des Produkts finden sich in den technischen Spezifikationen oder im Datenblatt **Achtung:** Stellen Sie sicher, dass die verwendeten Stromquellen stabil und korrekt dimensioniert sind. **Hinweis:** Verwenden Sie nur geprüfte und geeignete Netzteile, um Spannungsschwankungen und Überlastungen zu vermeiden. **Achtung:** Halten Sie ausreichenden Abstand zu spannungsführenden Teilen ein, um unabsichtlichen Kontakt zu vermeiden. **Hinweis:** Sorgen Sie entsprechend der verwendeten Spannung für eine sichere und übersichtliche Anordnung der Verkabelung. **Achtung:** Verwenden Sie isolierende Gehäuse oder Schutzabdeckungen, um das Produkt vor direktem Kontakt zu schützen. **Hinweis:** Setzen Sie das Produkt in ein nicht leitendes Gehäuse ein, um versehentliche Berührungen und Kurzschlüsse zu vermeiden. Das Produkt und die darauf befindlichen Komponenten können sich während des Betriebs erwärmen. Unsachgemäßer Umgang oder eine Überlastung des Produkts kann zu Verbrennungen, Beschädigungen oder Bränden führen. Beachten Sie die folgenden Sicherheitshinweise, um thermische Gefährdungen zu vermeiden: **Achtung:** Stellen Sie sicher, dass das Produkt innerhalb der empfohlenen Betriebstemperaturen verwendet wird. **Hinweis:** Der empfohlene Betriebstemperaturbereich liegt typischerweise zwischen -40°C und +85°C. Überprüfen Sie die spezifischen Angaben im Datenblatt des Produkts. **Achtung:** Platzieren Sie das Produkt nicht in der Nähe von externen Wärmequellen wie Heizkörpern oder direkter Sonnestrahlung. **Hinweis:** Sorgen Sie dafür, dass das Produkt in einem kühlen und gut belüfteten Bereich betrieben wird. **Achtung:** Stellen Sie sicher, dass das Produkt gut belüftet ist, um eine Überhitzung zu vermeiden. **Hinweis:** Verwenden Sie Lüfter oder Kühlkörper, wenn das Produkt in einem geschlossenen Gehäuse betrieben wird oder in einer Umgebung mit eingeschränkter Luftzirkulation. **Achtung:** Montieren Sie das Produkt auf hitzebeständigen Oberflächen und in hitzebeständigen Gehäusen. **Hinweis:** Verwenden Sie Materialien für Gehäuse, die hohe Temperaturen aushalten können, um eine Beschädigung oder Feuergefahr zu vermeiden. **Achtung:** Implementieren Sie eine Überwachung der Temperatur bei Verwendung eines Gehäuses und gegebenenfalls Schutzmechanismen, die das Produkt abschalten, wenn es überhitzt. **Hinweis:** **Hinweis:** Verwenden Sie Temperaturfühler und entsprechende Software, um die Temperatur des Produkts zu überwachen und das System bei Bedarf abzuschalten. **Achtung:** Vermeiden Sie Überlastungen, die zu übermäßiger Erwärmung der Komponenten führen können. **Hinweis:** Überschreiten Sie nicht die spezifizierten Grenzwerte für Strom und Spannung, um eine Überhitzung zu verhindern. **Achtung:** Kurzschlüsse können erhebliche Hitze entwickeln und Brände verursachen. **Hinweis:** Stellen Sie sicher, dass alle Verbindungen korrekt und sicher sind und dass keine leitenden Objekte unbeabsichtigt Kurzschlüsse verursachen können.

Inhaltsverzeichnis

Umfang der Lieferung	5
Einführung	8
Installation der Arduino IDE	8
Bibliotheken hinzufügen und seriellen Monitor öffnen	12
Installieren einer Bibliothek mit dem Bibliotheksmanager	12
Importieren einer .zip-Bibliothek	13
Arduino Serieller Monitor	14
Blink-Beispiel	18
LED-Beispiel	22
Schematische Darstellung	26
Schaltplan	27
RGB-LED	28
Farbe	30
Theorie PWM	31
Schematische Darstellung	33
Schaltplan	34
Beispiel-Code	35
Digitale Eingänge	38
Schematische Darstellung	39
Schaltplan	40
Beispiel Code	41
Neigungskugelschalter	43
Schematische Darstellung	45
Schaltplan	46
Beispiel-Code	47
Aktiver Buzzer	48
Schematische Darstellung	49
Schaltplan	50
Beispiel-Code	51
	2

Super Starter Kit

Passiver Buzzer	52
Schematische Darstellung	53
Beispiel für eine Verdrahtung	54
Beispiel-Code	55
DHT11-Sensor	56
Schematische Darstellung	58
Schaltplan	59
Beispiel-Code	60
IR-Empfängermodul	62
Schematische Darstellung	64
Schaltplan	65
Beispielcode	65
Servo-Motor	66
Schematische Darstellung	67
Schaltplan	68
Beispiel-Code	68
Joystick-Modul	69
Schematische Darstellung	70
Schaltplan	71
Beispiel-Code	72
Modul zur Wasserstandserkennung	74
Schematische Darstellung	76
Schaltplan	77
Beispiel-Code	78
Ultraschall-Sensormodul	79
Schematische Darstellung	80
Schaltplan	81
Beispiel-Code	82
Tonsensormodul	83
Schematische Darstellung	85
Schaltplan	86

Super Starter Kit

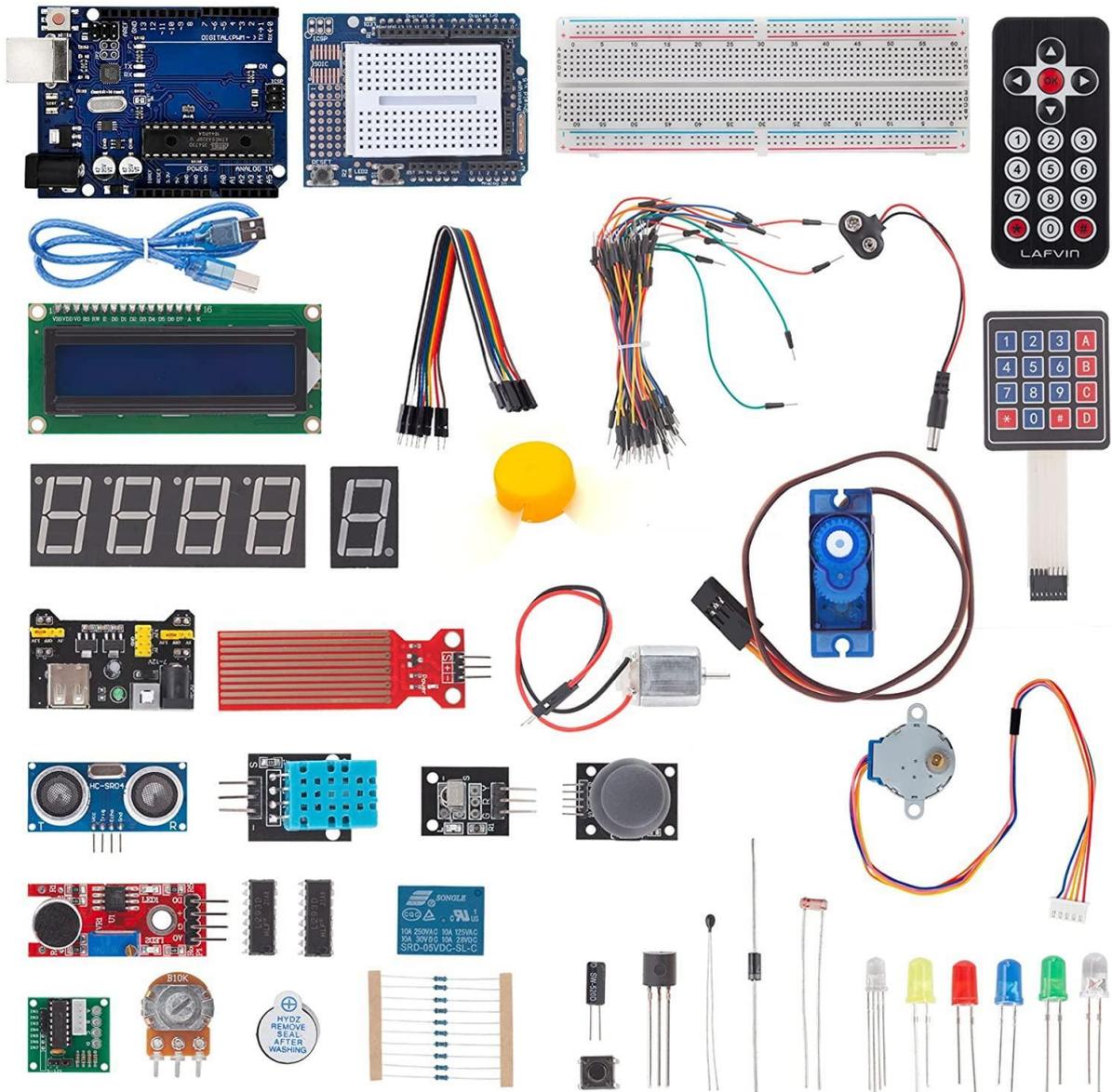
Beispiel-Code	87
Membranschalter-Modul	88
Schematische Darstellung	89
Schaltplan	90
Beispiel-Code	91
8 LEDs mit 74HC595	93
Schematische Darstellung	96
Schaltplan	97
Beispiel-Code	98
Der serielle Monitor	100
Beispiel-Code	100
Fotowiderstand	106
Schematische Darstellung	107
Schaltplan	108
Beispiel-Code	109
LCD1602 I2C-Modul	110
Schematische Darstellung	111
Verdrahtungsschema	112
Beispiel-Code	113
Thermometer	114
Schematische Darstellung	115
Schaltplan	116
Beispiel-Code	117
7 Segmentanzeige und 74HC595	118
Schematische Darstellung	120
Schaltplan	121
Beispiel-Code	122
7-Segment-Anzeige 4 Ziffern	124
Schematische Darstellung	125
Schaltplan	126
Beispiel-Code	127

Super Starter Kit

Gleichstrommotor	128
Einstellen der Ausgangsspannung	129
L293D	130
Schematische Darstellung	132
Schaltplan	133
Beispiel-Code	134
Relais	136
Schematische Darstellung	138
Schaltplan	139
Beispiel-Code	140
Schrittmotor	142
Schematische Darstellung	145
Schaltplan	146
Beispiel-Code	147
Steuerung eines Schrittmotors mit einer Fernbedienung	148
Schematische Darstellung	149
Schaltplan	150
Beispiel-Code	151

Super Starter Kit

Lieferumfang



- 65 Stück Jumper Kabel
- Schrittmotor
- Servomotor SG90
- LCD1602 mit I2C-Adapter
- Joystick-Modul
- Membranschalter-Modul
- 5V-Relais
- Schallsensor-Modul

Super Starter Kit

- Potentiometer 10K
- F-M Dupont-Drähte
- DHT11-Sensormodul
- 9V Batterieclip
- Diode 1N4007
- 5 Stück kleiner Knopf
- Thermistor
- Transistor PN2222
- RGB-LED
- rote, gelbe, blaue, grüne und weiße LEDs je 5 Stück
- atmega328 Mikrocontroller mit Kabel
- Prototyping-Schild
- 830-poliges Breadboard
- IR-Fernbedienung
- MB102 Breadboard Stromversorgung
- 100 Stück Widerstände
- ULN2003 Schrittmotor-Treiberplatine
- HC-SR04 Ultraschallmodul
- Wasserstandsdetektor
- Lüfterrad mit DC-Motor
- 1-stellige 7-Segment-Anzeige
- 4-stellige 7-Segment-Anzeige
- IR-Empfängermodul
- L293D-Motortreiber
- 74HC595
- Kippschalter
- Aktiver Buzzer
- Passiver Summer
- 2 Stück Fotowiderstände

Einführung

Dieses Tutorial ist für Anfänger gedacht. Sie lernen die Grundlagen der Arduino-IDE kennen und erfahren, wie Sie den Mikrocontroller atmega328 mit Sensoren und anderen Komponenten verwenden.

Installation der Arduino IDE

Sie können die kostenlose Entwicklungsumgebung Arduino IDE über den folgenden Link herunterladen: <https://www.arduino.cc/en/Main/Software>

Windows-Benutzer sollten auf jeden Fall eine der ersten beiden Download-Optionen für die Arduino-IDE verwenden. Die "Windows App"-Version aus dem Windows Store verursacht Verbindungsprobleme, insbesondere bei der Verwendung von Board-Definitionen von Drittanbietern.

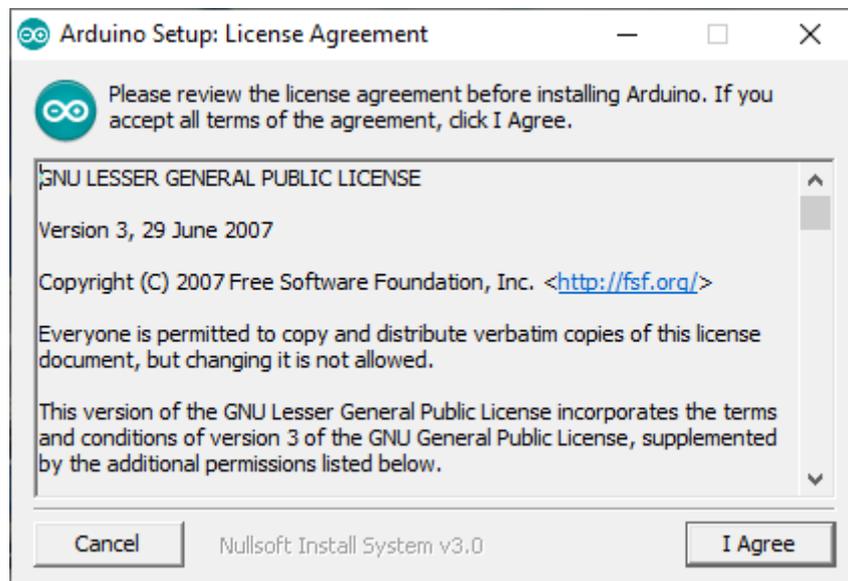
Download the Arduino IDE



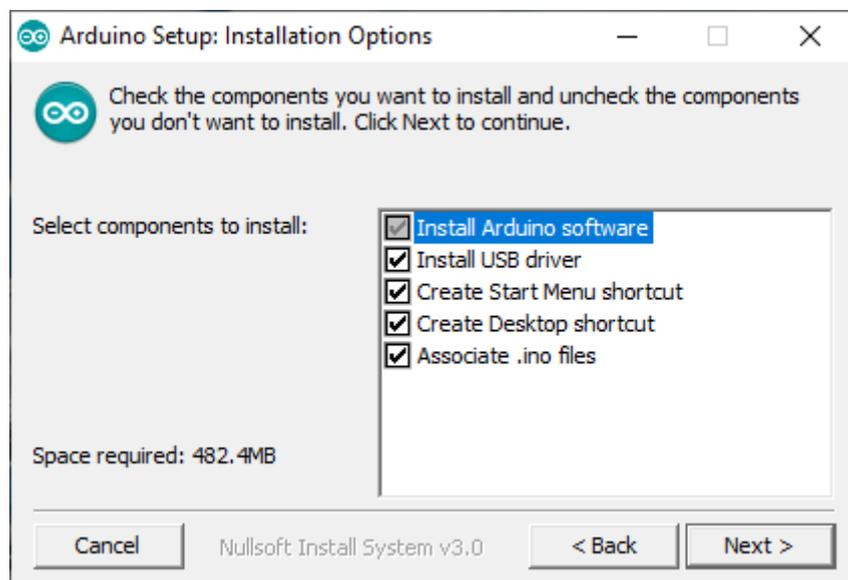
The screenshot shows the Arduino IDE download page. On the left, there is a circular logo with a minus sign and a plus sign. To its right, the text reads: **ARDUINO 1.X.X**. Below this, it says: "The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software. This software can be used with any Arduino board. Refer to the [Getting Started](#) page for Installation instructions." A green arrow points from this text to the right-hand side of the page. On the right, there is a teal sidebar with several options: "Windows Installer, for Windows XP and up", "Windows ZIP file for non admin install", "Windows app, Requires Win 8.1 or 10" (with a red 'X' over it and a "Get" button), "Mac OS X 10.8 Mountain Lion or newer", "Linux 32 bits", "Linux 64 bits", "Linux ARM 32 bits", and "Linux ARM 64 bits". At the bottom of the sidebar, there are links for "Release Notes", "Source Code", and "Checksums (sha512)".

Super Starter Kit

Nach dem Start der Arduino IDE Installationsdatei "arduino-1.X.X-windows.exe" müssen die Lizenzbedingungen der Software gelesen und akzeptiert werden:



Im nächsten Schritt können verschiedene Optionen für die Installation ausgewählt werden.

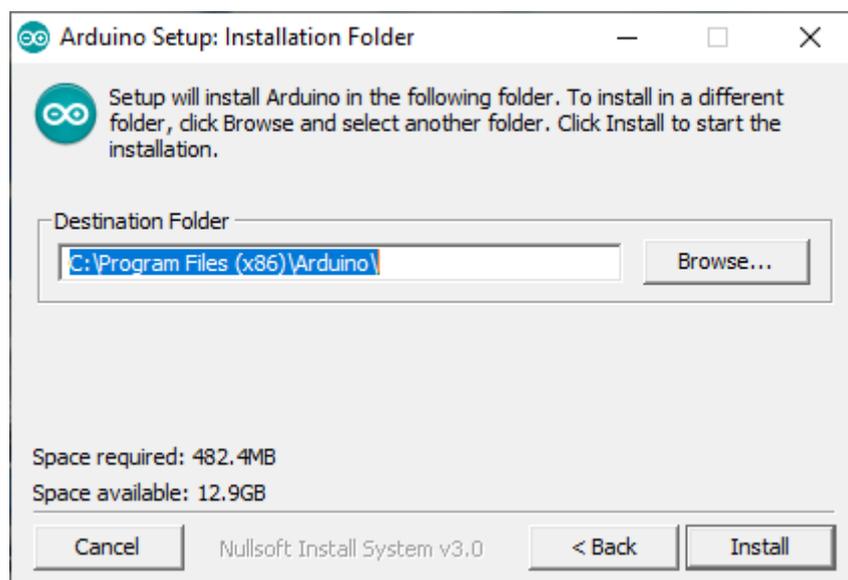


Super Starter Kit

Es folgt ein kurzer Überblick über die verschiedenen Optionen mit einer kurzen Erläuterung zu jeder Option:

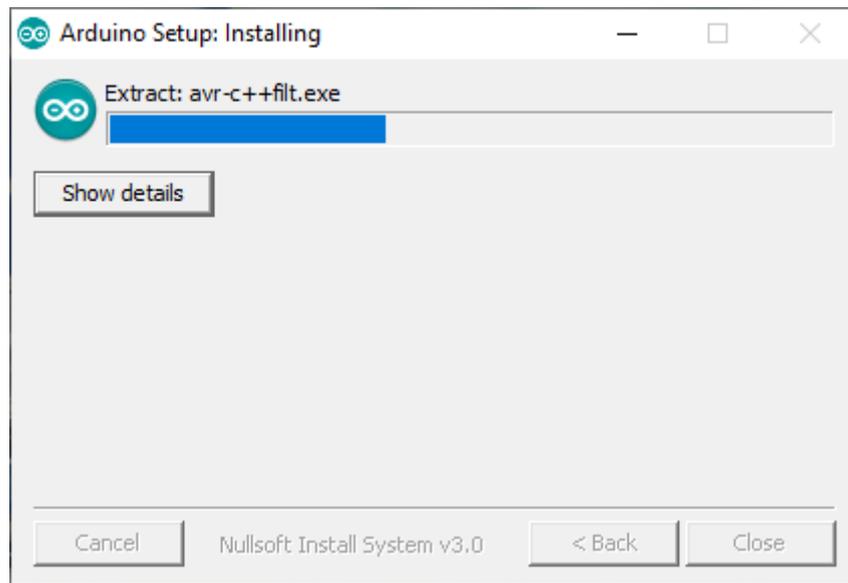
Option	Erklärung
Arduino-Software installieren	Installiert die Arduino IDE - Diese Option kann nicht abgewählt werden
USB-Treiber installieren	Installiert USB-Treiber für verschiedene andere Mikrocontroller. Diese sind für die Verwendung der Software mit dem D1 mini nicht erforderlich, aber wir empfehlen dringend, sie zu installieren, wenn Sie auch andere Mikrocontroller verwenden
Startmenü-Verknüpfung erstellen	Erzeugt eine Verknüpfung im Windows-Startmenü (optional)
Desktop-Verknüpfung erstellen	Erstellt eine Verknüpfung auf der Arbeitsstation (Optional)
Zuordnen von .ino-Dateien	Erzeugt eine Dateinamenerweiterung für Dateien mit der Endung .ino und verknüpft sie mit der Arduino IDE

Schließlich muss der Zielordner angegeben werden. Für die Installation werden etwa 500 MB freier Speicherplatz benötigt.

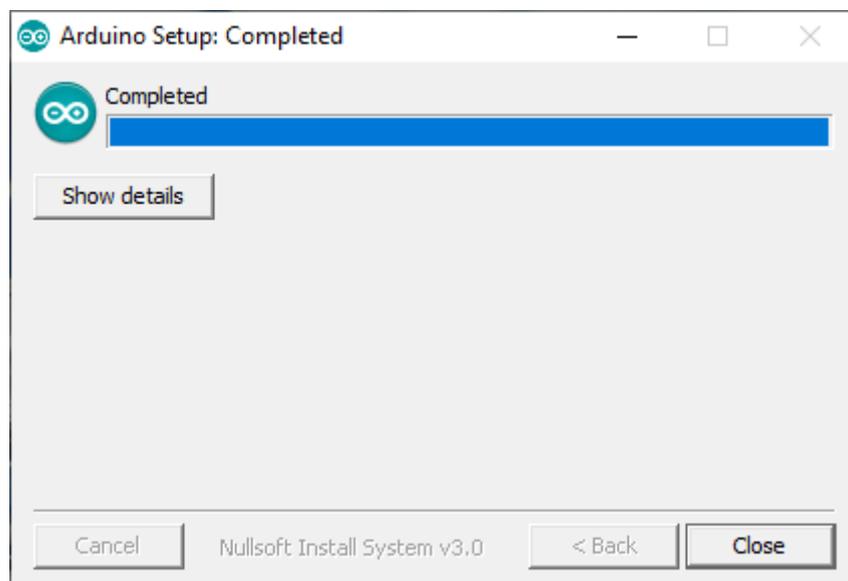


Super Starter Kit

Klicken Sie auf "Installieren", um die Installation zu starten.



Nach erfolgreicher Installation kann das Installationsprogramm über die Schaltfläche "Schließen" beendet werden:



Super Starter Kit

Bibliotheken hinzufügen und seriellen Monitor öffnen

Wenn Sie sich mit der Arduino-Software und den eingebauten Funktionen vertraut gemacht haben, möchten Sie vielleicht die Möglichkeiten Ihres Arduinos mit zusätzlichen Bibliotheken erweitern

Bibliotheken sind Codesammlungen, die den Anschluss an einen Sensor, ein Display, ein Modul usw. erleichtern. Die eingebaute LiquidCrystal-Bibliothek macht es zum Beispiel einfach, mit LCD-Anzeigen zu kommunizieren. Es gibt Hunderte von zusätzlichen Bibliotheken, die im Internet zum Download bereitstehen. Die eingebauten Bibliotheken und einige dieser zusätzlichen Bibliotheken sind in der Referenz aufgeführt. Um die zusätzlichen Bibliotheken zu verwenden, müssen Sie sie installieren.

Installieren einer Bibliothek mit dem Bibliotheksmanager

Um eine neue Bibliothek in Ihrer Arduino IDE zu installieren, können Sie den Bibliotheksverwalter verwenden. Öffnen Sie die IDE und klicken Sie auf das Menü "Sketch" und dann auf "Include Library" > "Manage Libraries".

Dann öffnet sich der Bibliotheksmanager und Sie finden eine Liste von Bibliotheken, die bereits installiert oder zur Installation bereit sind. In diesem Beispiel werden wir die Bridge-Bibliothek installieren. Blättern Sie in der Liste, um sie zu finden, und wählen Sie die Version der Bibliothek aus, die Sie installieren möchten. Wenn Sie sie nicht finden, können Sie die Suchleiste verwenden. Manchmal ist nur eine Version der Bibliothek verfügbar. Wenn das Menü zur Versionsauswahl nicht erscheint, ist das normal.

Klicken Sie schließlich auf Installieren und warten Sie, bis die IDE die neue Bibliothek installiert hat. Das Herunterladen kann je nach Geschwindigkeit Ihrer Verbindung einige Zeit dauern. Sobald der Download abgeschlossen ist, sollte neben der Bridge-Bibliothek der Vermerk Installiert erscheinen. Sie können den Bibliotheksmanager schließen.

Super Starter Kit

Importieren einer .zip-Bibliothek

Bibliotheken werden oft als ZIP-Datei oder Ordner verteilt. Der Name des Ordners ist der Name der Bibliothek. Innerhalb des Ordners befinden sich eine .cpp-Datei, eine .h-Datei und oft eine keywords.txt-Datei, ein Ordner mit Beispielen und andere für die Bibliothek erforderliche Dateien. Ab Version 1.0.5 können Sie Bibliotheken von Drittanbietern in der IDE installieren. Entpacken Sie die heruntergeladene Bibliothek nicht, sondern lassen Sie sie so wie sie ist. Navigieren Sie in der Arduino IDE zu Sketch > Include Library. Wählen Sie oben in der Dropdown-Liste die Option "Add .ZIP Library".

Sie werden aufgefordert, die Bibliothek auszuwählen, die Sie hinzufügen möchten. Navigieren Sie zu dem Ort, an dem sich die ZIP-Datei befindet, und öffnen Sie sie.

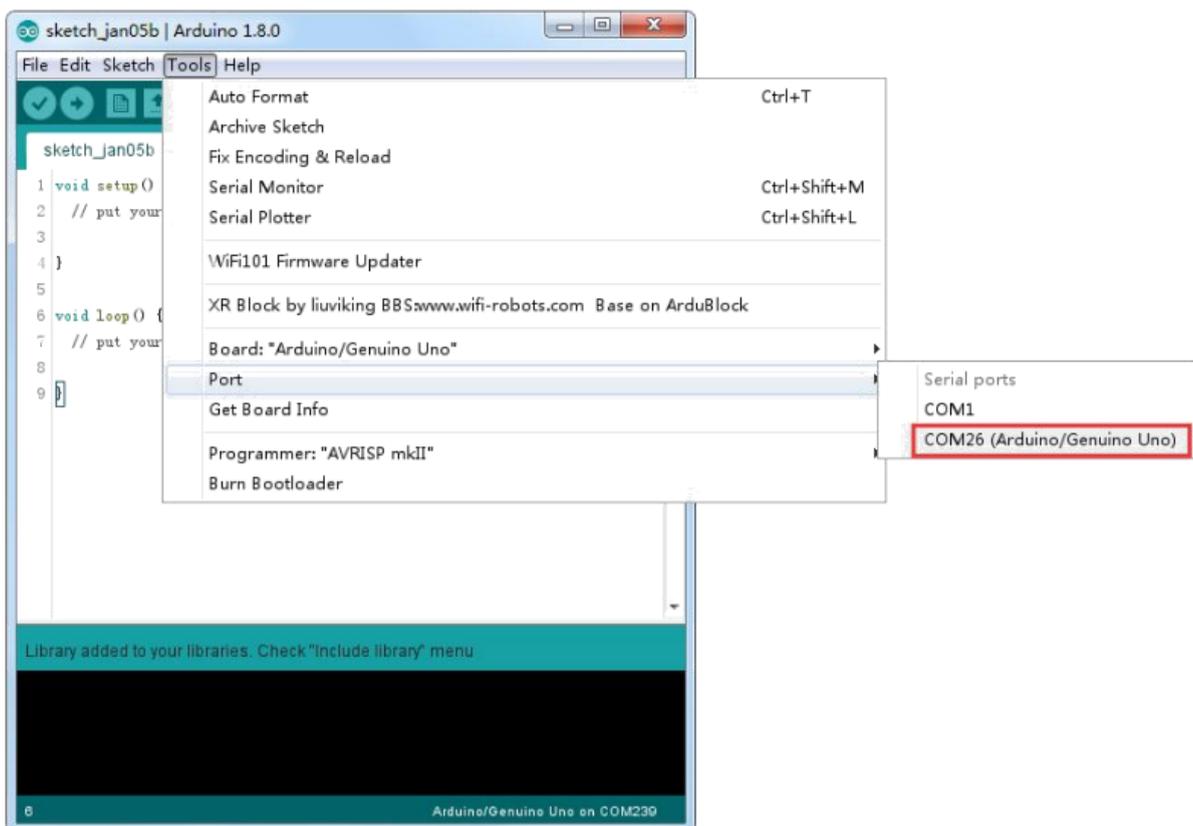
Kehren Sie zum Menü Skizze > Bibliothek importieren zurück. Die Bibliothek sollte nun unten im Dropdown-Menü angezeigt werden. Sie kann nun in Ihrer Skizze verwendet werden. Die Zip-Datei wurde im Ordner libraries in Ihrem Arduino-Sketch-Verzeichnis erweitert.

Super Starter Kit

Arduino Serieller Monitor

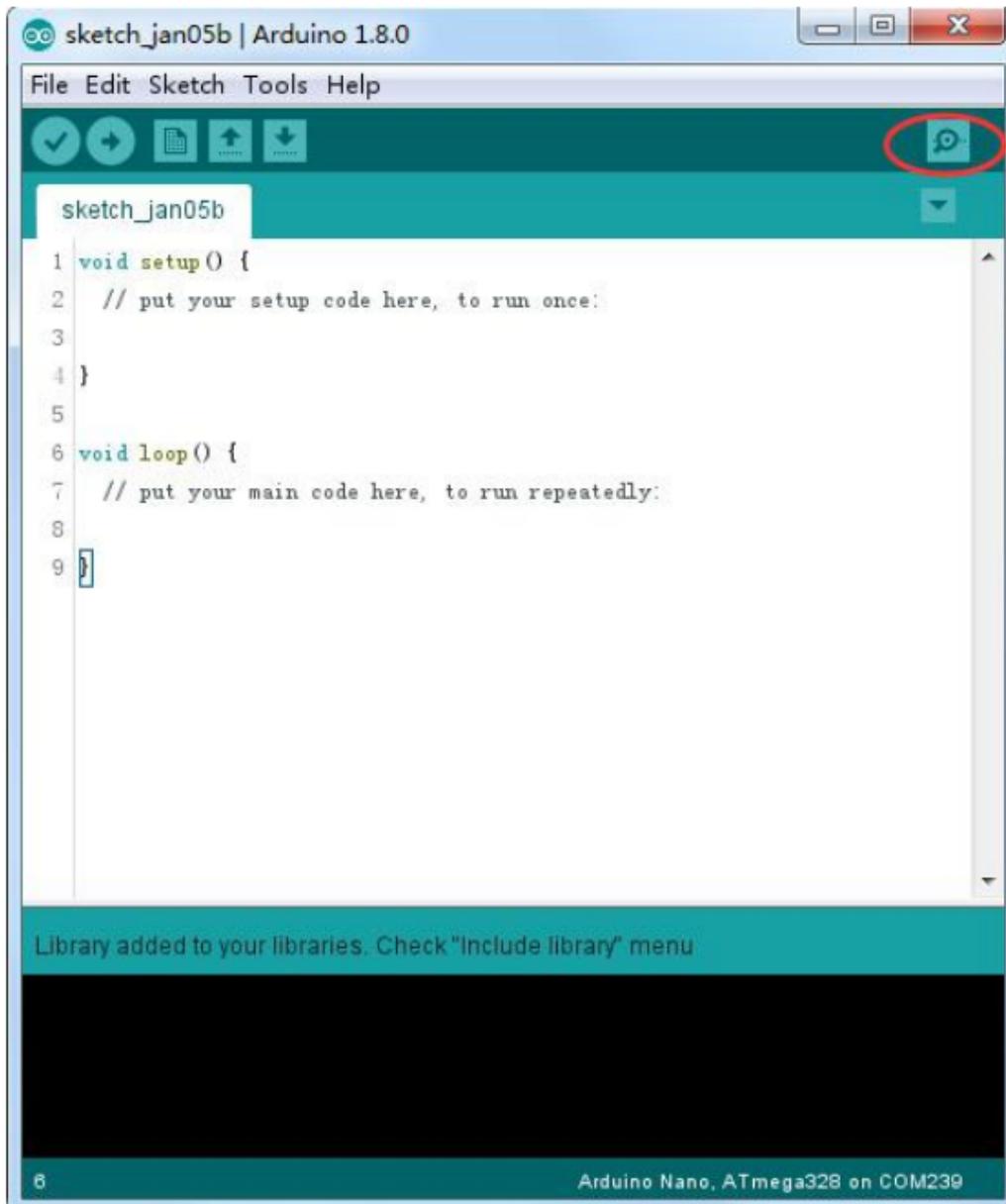
Die integrierte Entwicklungsumgebung (IDE) von Arduino ist die Software-Seite der Arduino-Plattform. Da die Verwendung eines Terminals ein wichtiger Bestandteil der Arbeit mit Arduinos und anderen Mikrocontrollern ist, wurde beschlossen, ein serielles Terminal in die Software zu integrieren. In der Arduino-Umgebung wird dies als Serial Monitor bezeichnet

Die Auswahl des Ports, der im Serial Monitor geöffnet werden soll, entspricht der Auswahl eines Ports für das Hochladen von Arduino-Code. Gehen Sie zu Tools -> Port, und wählen Sie den richtigen Port. Tipps: Wählen Sie denselben COM-Anschluss, den Sie im Gerätemanager haben.



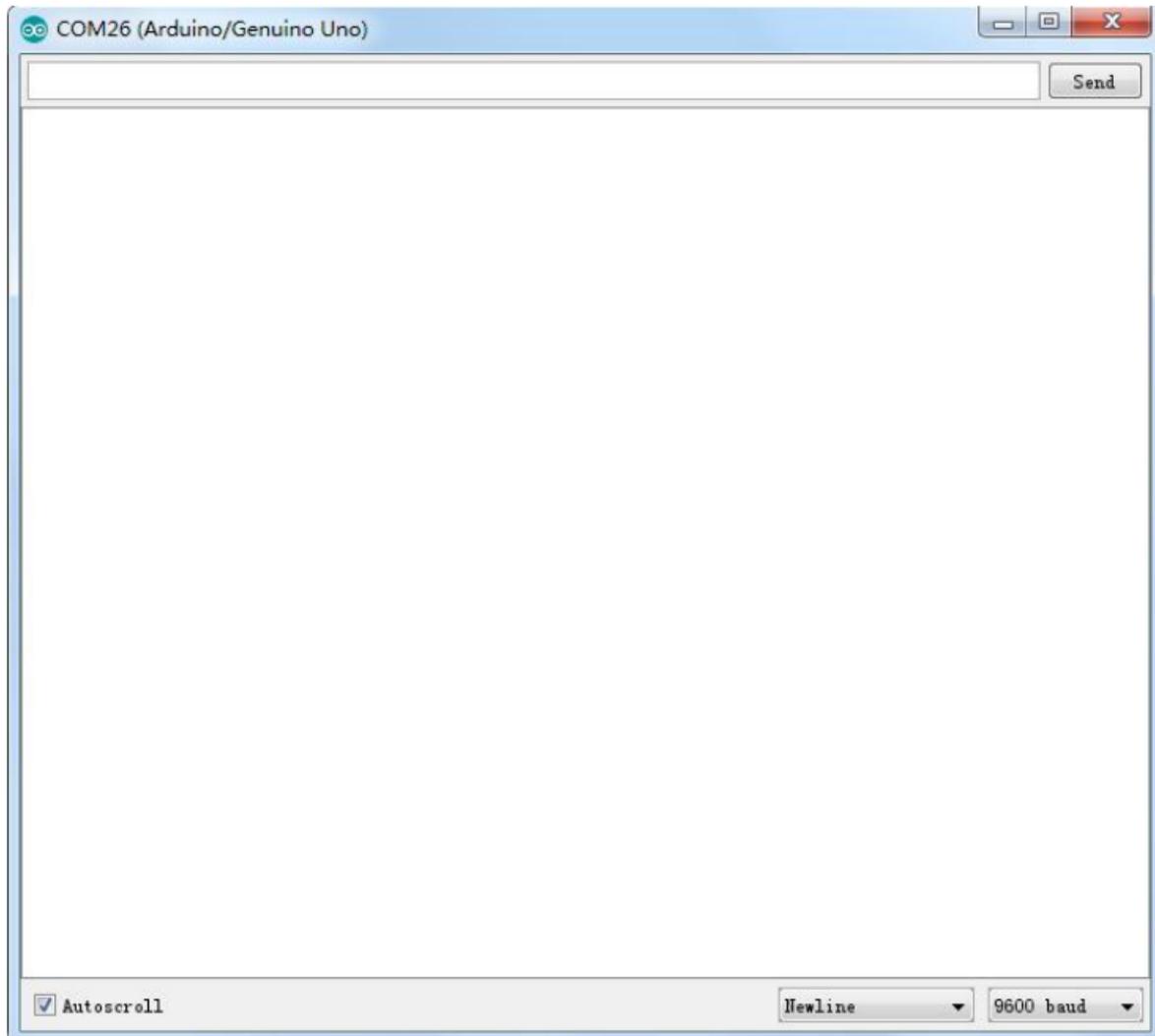
Super Starter Kit

Um den Serial Monitor zu öffnen, klicken Sie einfach auf das Symbol Serial Monitor.



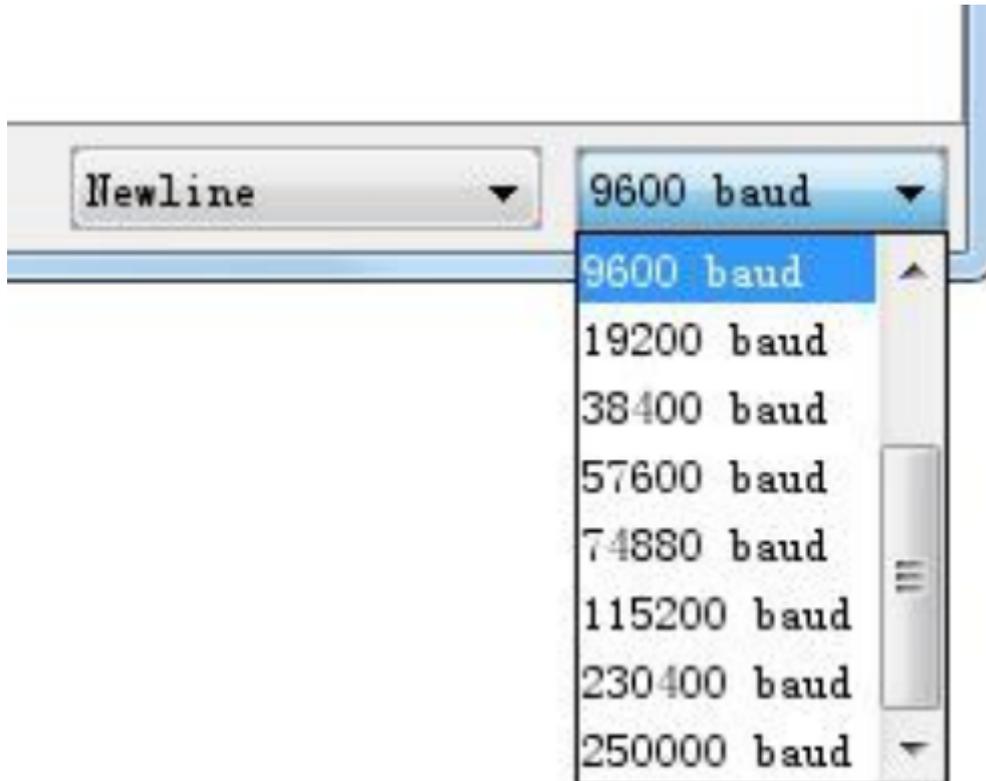
Super Starter Kit

Nach dem Öffnen sollten Sie etwa Folgendes sehen:



Super Starter Kit

Der Serial Monitor verfügt über begrenzte Einstellungen, die jedoch ausreichen, um die meisten Ihrer Anforderungen an die serielle Kommunikation zu erfüllen. Die erste Einstellung, die Sie ändern können, ist die Baudrate. Klicken Sie auf das Dropdown-Menü Baudrate, um die richtige Baudrate auszuwählen. (9600 Baud).



Zuletzt können Sie das Terminal auf Autoscroll einstellen oder nicht, indem Sie das Kästchen in der unteren linken Ecke markieren.



Super Starter Kit

Blink-Beispiel

Sie haben also die Arduino IDE installiert, jetzt ist es Zeit für Ihr erstes Projekt. Das Blink-Beispiel ist im Grunde ein "Hello World", es ist sehr einfach und Sie können auch überprüfen, ob alles so funktioniert, wie es sollte. Das Ergebnis sollte eine blinkende LED auf deinem Board sein.

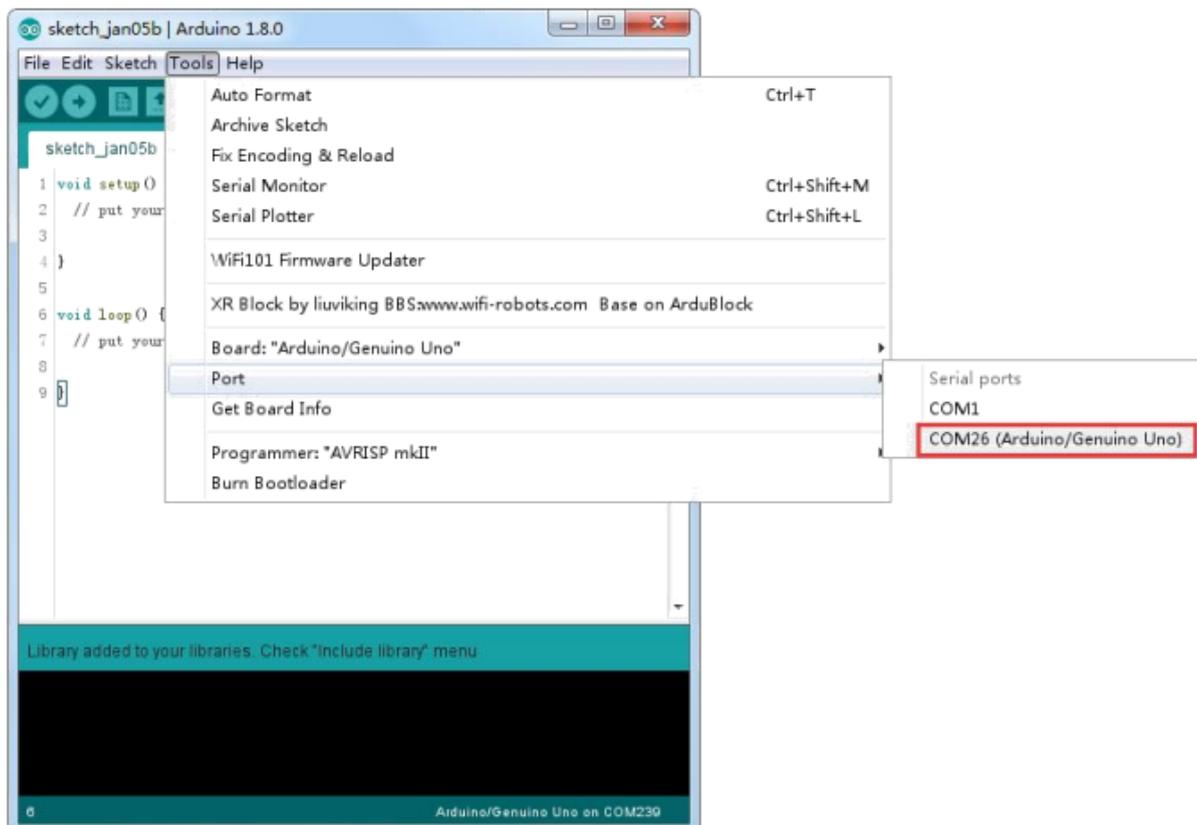
Für dieses Beispiel brauchst du nur den Mikrocontroller und das USB-Kabel.

Öffne die Arduino IDE und gehe zu Datei > Beispiele > Grundlagen > Blinken.

Verbinde nun den Mikrocontroller mit deinem PC über das USB-Kabel.

In der Arduino IDE gehen Sie zu Werkzeuge > Board > Arduino AVR Boards > Arduino Uno.

Sie können jetzt auch den Port auswählen.



Super Starter Kit

Sie können den Beispielcode über die Schaltfläche Upload hochladen



Wenn Sie sich den Statusbereich der IDE ansehen, sehen Sie einen Fortschrittsbalken und eine Reihe von Meldungen. Zunächst steht dort 'Sketch kompilieren...'. Dadurch wird der Sketch in ein Format konvertiert, das zum Hochladen auf das Board geeignet ist



Als Nächstes ändert sich der Status auf "Hochladen". Zu diesem Zeitpunkt sollten die LEDs am Arduino zu flackern beginnen, da der Sketch übertragen wird.

Schließlich ändert sich der Status in "Done" (Fertig).



Super Starter Kit

Wenn der Upload fehlschlägt kann es bedeuten, dass Ihre Karte gar nicht angeschlossen ist, dass die Treiber nicht installiert wurden (falls erforderlich) oder dass der falsche serielle Anschluss gewählt wurde.

Nachdem der Upload abgeschlossen ist, können Sie die LED blinken sehen.

Beachten Sie, dass ein großer Teil dieses Sketches aus Kommentaren besteht. Dies sind keine eigentlichen Programmanweisungen, sondern sie erklären nur, wie das Programm funktioniert. Sie sind zu Ihrem Nutzen da. Alles, was zwischen `/*` und `*/` am Anfang des Sketches steht, ist ein Blockkommentar; er erklärt, wofür der Sketch gedacht ist.

Einzeilige Kommentare beginnen mit `//` und alles bis zum Ende dieser Zeile wird als Kommentar betrachtet.

Als nächstes kommt die Funktion "setup". Wie der Kommentar schon sagt, wird diese Funktion ausgeführt, wenn die Reset-Taste gedrückt wird. Sie wird auch immer dann ausgeführt, wenn das Board aus irgendeinem Grund zurückgesetzt wird, z. B. wenn es zum ersten Mal mit Strom versorgt wird, oder nachdem ein Sketch hochgeladen wurde.

```
void setup() {  
  // initialize the digital pin as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}
```

Jeder Arduino-Sketch muss eine "Setup"-Funktion haben, und der Ort, an dem Sie eigene Anweisungen hinzufügen möchten, befindet sich zwischen dem `{` und dem `}`. In diesem Fall gibt es dort nur einen Befehl, der, wie der Kommentar sagt, dem Arduino-Board mitteilt, dass wir den Onboard-LED-Pin als Ausgang verwenden werden (die Onboard-LED ist an Pin 13 angeschlossen). Außerdem muss ein Sketch unbedingt eine Schleifenfunktion haben. Im Gegensatz zur "Setup"-Funktion, die nur einmal nach einem Reset ausgeführt wird, startet die "Loop"-Funktion sofort wieder, nachdem sie ihre Befehle ausgeführt hat.

Super Starter Kit

```
void loop() {  
    digitalWrite(LED_BUILTIN, HIGH);  
    delay(1000);  
    digitalWrite(LED_BUILTIN, LOW);  
    delay(1000);  
}
```

Innerhalb der Schleifenfunktion schalten die Befehle zunächst den LED-Pin ein (HIGH), dann 'verzögern' sie für 1000 Millisekunden (1 Sekunde), schalten dann den LED-Pin aus und halten eine weitere Sekunde lang an. Jetzt lässt du deine LED schneller blinken. Wie Sie vielleicht schon erraten haben, liegt der Schlüssel dazu in der Änderung des Parameters in () für den Befehl "delay".

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

Diese Verzögerungszeit wird in Millisekunden angegeben. Wenn Sie also möchten, dass die LED doppelt so schnell blinkt, ändern Sie den Wert von 1000 auf 500. Dies würde dann für eine halbe Sekunde jede Verzögerung statt eine ganze Sekunde pausieren. Laden Sie den Sketch erneut hoch und Sie sollten sehen, dass die LED schneller zu blinken beginnt.

Super Starter Kit

LED-Beispiel

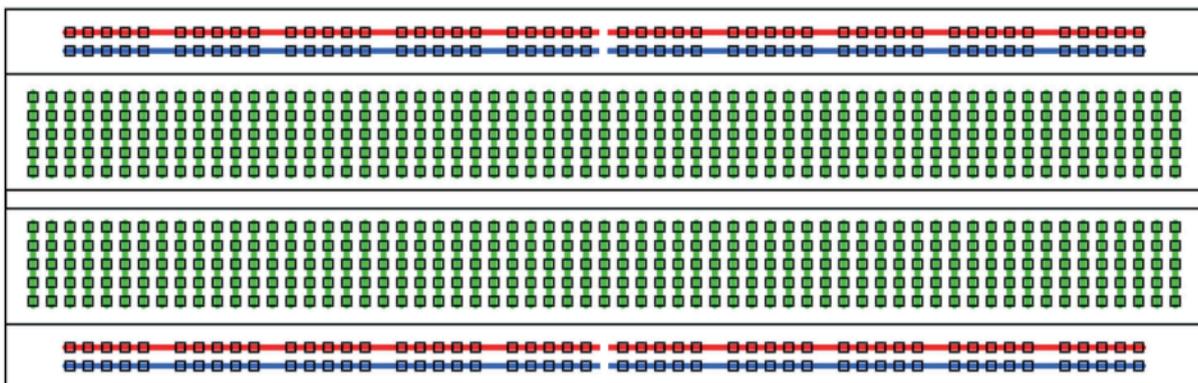
In diesem Beispiel benutzt du ein Breadboard, eine rote LED und verschiedene Widerstände, um die Helligkeit der LED zu verändern. Die Arduino IDE wird dafür nicht benötigt.

Für dieses Beispiel benötigst du:

- Mikrocontroller mit Kabel
- 830-polige Lochrasterplatine
- 5mm rote LED
- 220-Ohm-Widerstand
- 1k-Ohm-Widerstand
- 10k-Ohm-Widerstand
- 2 Überbrückungsdrähte

Mit einem Breadboard können Sie schnell Prototypen von Schaltungen erstellen, ohne dass Sie die Verbindungen löten müssen. Unten sehen Sie ein Beispiel.

Bitte beachten Sie, dass die Stromschienen (rot und blau) in der Mitte unterbrochen sind.



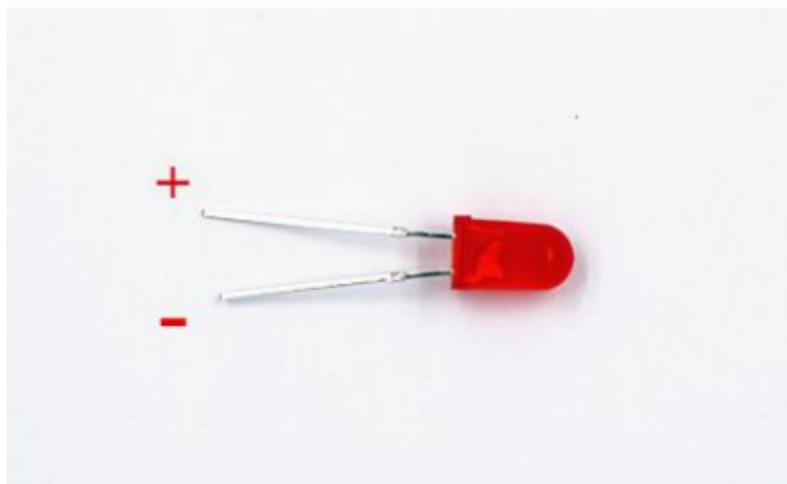
Breadboards gibt es in verschiedenen Größen und Konfigurationen. Die einfachste Variante ist ein Gitter aus Löchern in einem Kunststoffblock. Darin befinden sich Metallstreifen, die die elektrische Verbindung zwischen den Löchern in den kürzeren Reihen herstellen.

Super Starter Kit

Wenn man die Beine von zwei verschiedenen Bauteilen in dieselbe Reihe schiebt, werden sie elektrisch miteinander verbunden. Ein tiefer Kanal, der in der Mitte verläuft, zeigt an, dass die Verbindungen dort unterbrochen sind, d. h., Sie können einen Chip mit den Beinen auf beiden Seiten des Kanals einstecken, ohne sie miteinander zu verbinden.

Einige Lochrasterplatinen haben zwei Lochstreifen, die entlang der langen Kanten der Platine verlaufen und vom Hauptgitter getrennt sind. Diese Streifen verlaufen über die gesamte Länge der Platine und bieten eine Möglichkeit, eine gemeinsame Spannung anzuschließen. Sie sind in der Regel paarweise für +5 Volt und Masse angeordnet. Diese Leisten werden als Schienen bezeichnet und ermöglichen den Anschluss von Strom an viele Komponenten oder Punkte auf der Platine.

Breadboards eignen sich zwar hervorragend für das Prototyping, haben aber auch einige Einschränkungen. Da die Verbindungen durch Einstecken hergestellt werden und nur vorübergehend bestehen, sind sie nicht so zuverlässig wie gelötete Verbindungen. Wenn Sie intermittierende Probleme mit einer Schaltung haben, könnte das an einer schlechten Verbindung auf einem Breadboard liegen.



LEDs eignen sich hervorragend als Anzeigeleuchten. Sie verbrauchen sehr wenig Strom und halten so gut wie ewig. Hier werden Sie die rote 5 mm-LED verwenden.

Super Starter Kit

5 mm bezieht sich auf den Durchmesser der LED. Andere gängige Größen sind 3 mm und 10 mm. Sie können eine LED nicht direkt an eine Batterie oder eine Spannungsquelle anschließen, weil

- 1) die LED einen positiven und einen negativen Anschluss hat und nicht leuchtet, wenn sie falsch herum angeschlossen wird und
- 2) eine LED muss mit einem Widerstand verwendet werden, um die Stromstärke zu begrenzen oder zu drosseln.
zu begrenzen, da sie sonst durchbrennt.

Es gibt zwei Möglichkeiten, um festzustellen, welches der Pluspol der LED ist und welches der negativ ist.

Erstens ist der positive Anschluss länger. Zweitens befindet sich an der Stelle, an der die negative Leitung in den LED-Körper eintritt, eine flache Kante am Gehäuse der LED. Wenn Sie zufällig eine LED haben, die eine flache Seite neben der längeren Leitung hat, sollten Sie davon ausgehen, dass die längere Leitung positiv ist.

Nun zu den Widerständen.

Wie der Name schon sagt, widerstehen Widerstände dem Stromfluss. Je höher der Wert des Widerstands ist, desto mehr Widerstand leistet er und desto weniger Strom fließt durch ihn.

Damit wollen wir steuern, wie viel Strom durch die LED fließt und wie hell sie daher leuchtet.



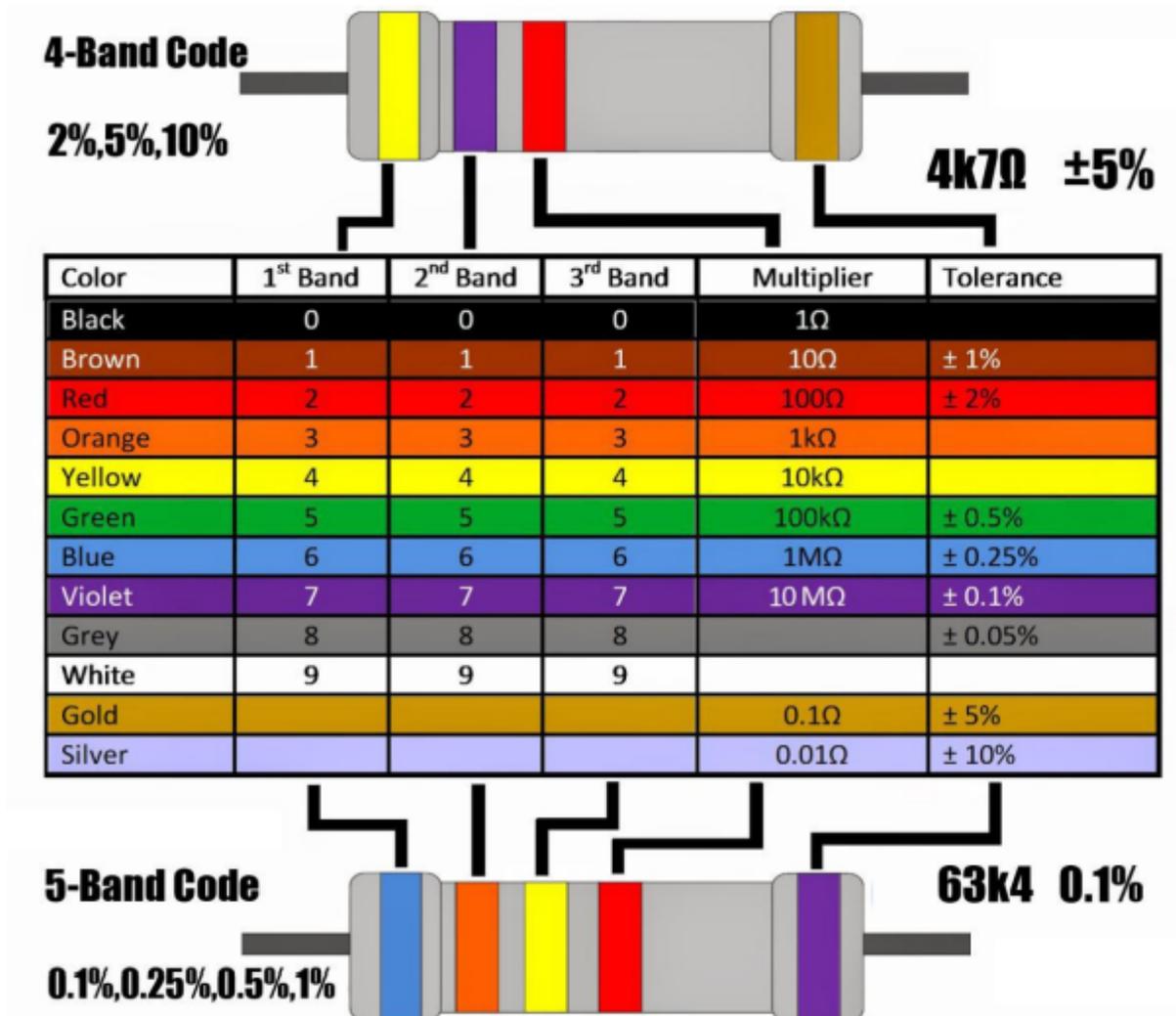
Doch zunächst mehr über Widerstände... Die Einheit des Widerstands heißt Ohm, was gewöhnlich mit Ω , dem griechischen Buchstaben Omega, abgekürzt wird.

Da ein Ohm ein niedriger Widerstandswert ist (er widersteht nicht viel), werden die Werte von Widerständen auch in $k\Omega$ (1.000 Ω) und $M\Omega$ (1.000.000 Ω) angegeben.

Diese werden Kilo-Ohm und Mega-Ohm genannt.

Super Starter Kit

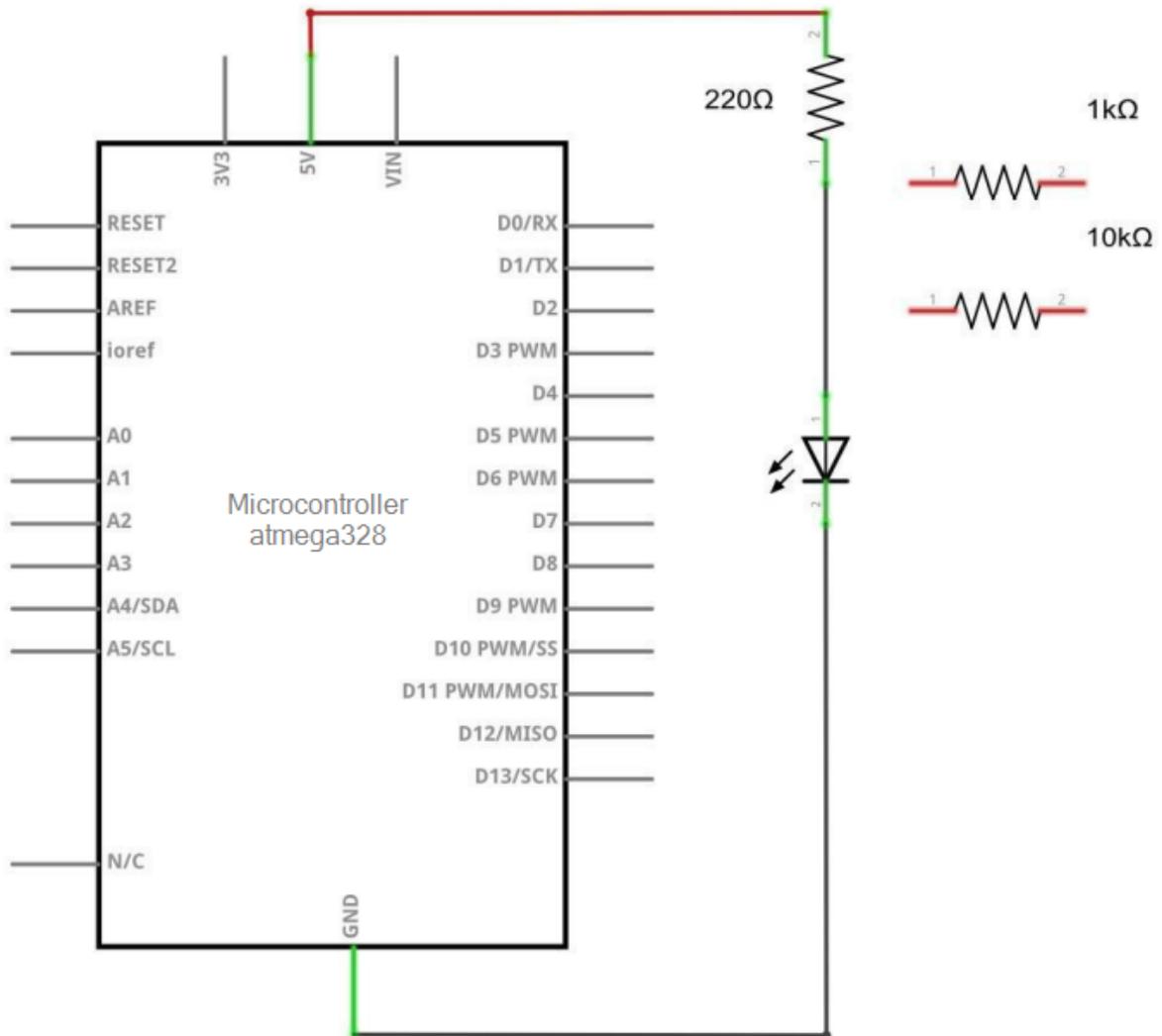
Diese Widerstände sehen alle gleich aus, sind aber mit verschiedenfarbigen Streifen versehen. Diese Streifen geben den Wert des Widerstands an. Der Farbcode des Widerstands besteht aus drei farbigen Streifen und einem goldenen Streifen an einem Ende.



Im Gegensatz zu LEDs haben Widerstände keinen positiven und negativen Anschluss. Sie können in beide Richtungen angeschlossen werden.

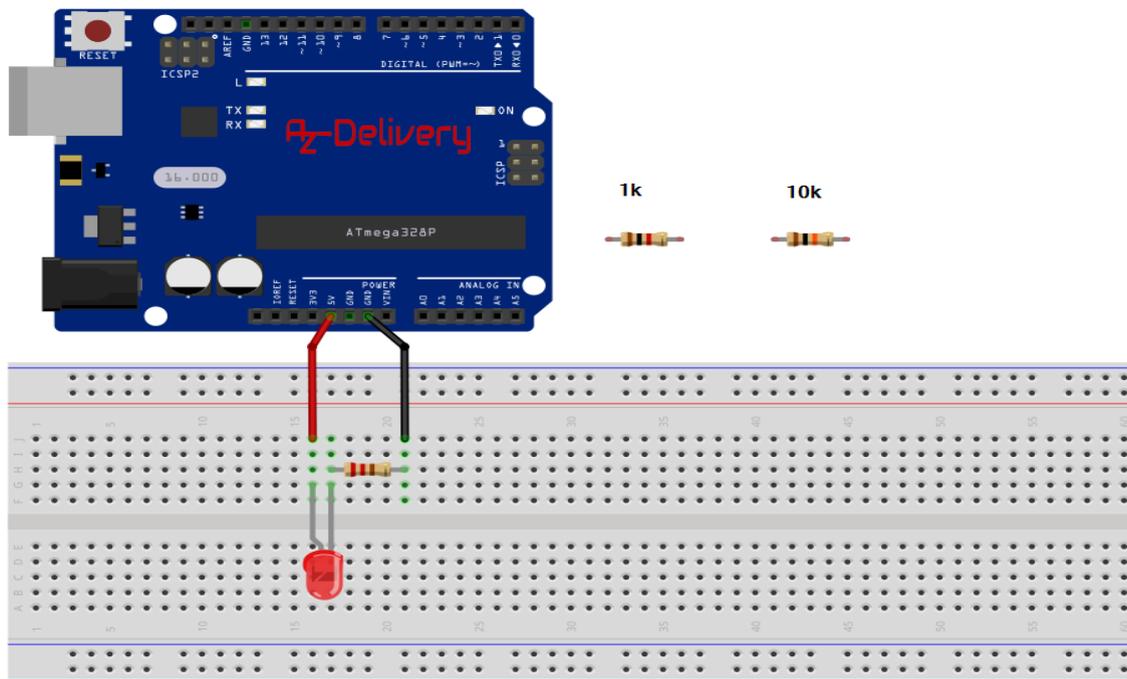
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Der rote Draht ist mit dem 5V-Pin des Mikrocontrollers und der positiven Leitung der LED verbunden. Die negative Leitung der LED ist mit dem 220-Ohm-Widerstand verbunden und die andere Leitung des Widerstands ist mit dem GND-Pin (schwarzer Draht) des Mikrocontrollers verbunden.

Der Mikrocontroller ist eine praktische 5-Volt-Quelle, mit der wir die LED und den Widerstand mit Strom versorgen werden. Sie müssen mit Ihrem Mikrocontroller nichts weiter tun, als ihn mit einem USB-Kabel anzuschließen. Mit dem 220- Ω -Widerstand an der richtigen Stelle sollte die LED recht hell sein.

Wenn Sie den 220- Ω -Widerstand gegen den 1-k Ω -Widerstand austauschen, wird die LED etwas dunkler erscheinen.

Mit dem 10 k Ω -Widerstand schließlich ist die LED gerade noch sichtbar.

Wenn wir den Widerstand so verschieben, dass er vor die LED kommt, leuchtet die LED immer noch. Wahrscheinlich werden Sie den 220- Ω -Widerstand wieder einsetzen wollen. Es spielt keine Rolle, auf welcher Seite der LED wir den Widerstand anbringen, Hauptsache, er ist irgendwo.

RGB-LED

RGB-LEDs sind eine lustige und einfache Möglichkeit, Farbe in Ihre Projekte zu bringen. Da sie wie 3 normale LEDs in einem sind, ist die Verwendung und der Anschluss nicht viel anders.

Es gibt sie meist in 2 Versionen: Gemeinsame Anode oder gemeinsame Kathode. Gemeinsame Anode verwendet 5 V auf dem gemeinsamen Pin, während gemeinsame Kathode mit Masse verbunden ist.

Wie bei jeder LED müssen wir einige Widerstände in Reihe schalten (insgesamt 3), um den Strom zu begrenzen.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- 830-poliges Breadboard
- 4 Jumper-Drähte
- RGB-LED
- 3 220-Ohm-Widerstände

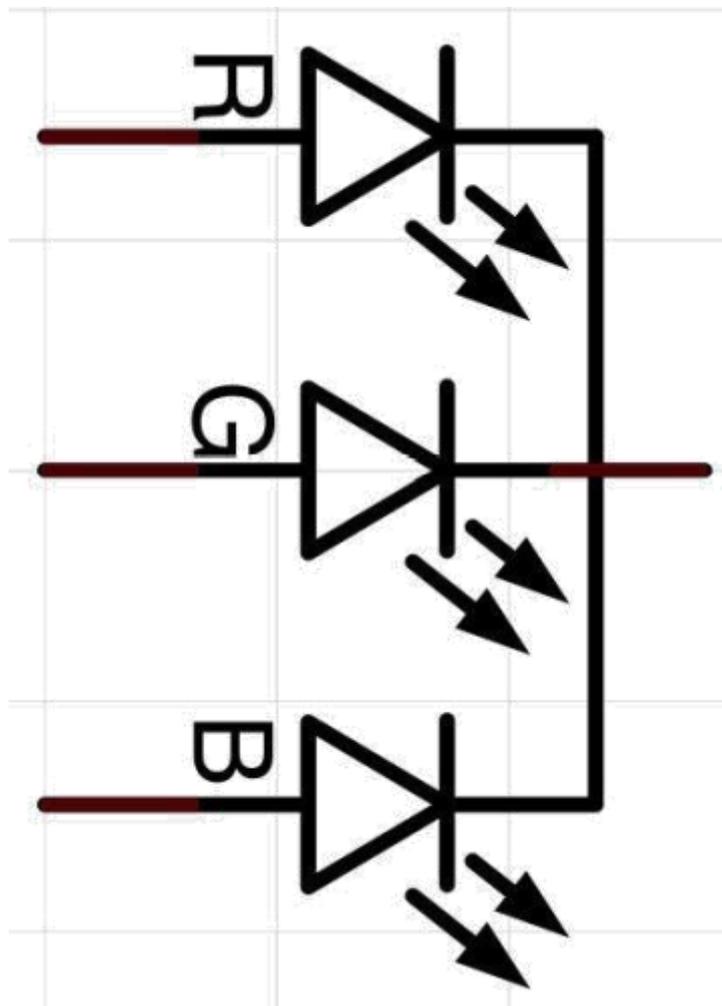
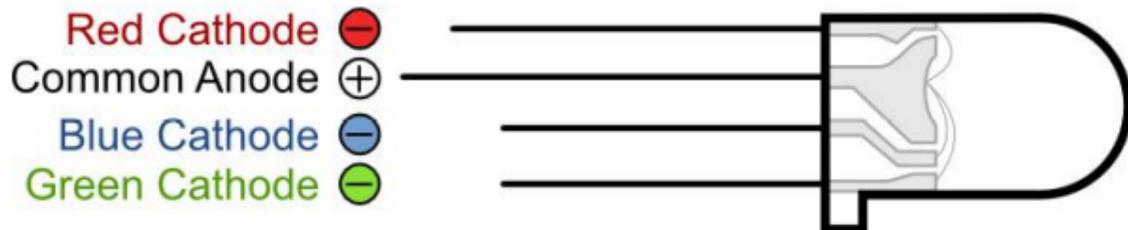
Auf den ersten Blick sehen RGB-LEDs (Rot, Grün und Blau) wie normale LEDs aus. Im Inneren des üblichen LED-Gehäuses befinden sich jedoch tatsächlich drei LEDs, eine rote, eine grüne und eine blaue. Indem man die Helligkeit jeder einzelnen LED steuert, kann man so ziemlich jede Farbe mischen, die man möchte.

Wir mischen die Farben auf die gleiche Weise, wie man Farbe auf einer Palette mischen würde - indem wir die Helligkeit jeder der drei LEDs einstellen. Der schwierige Weg, dies zu tun, wäre die Verwendung von Widerständen mit unterschiedlichen Werten (oder variablen Widerständen), wie wir es zuvor getan haben, aber das ist eine Menge Arbeit!

Zu unserem Glück verfügt die Mikrocontrollerplatine über eine AnalogWrite-Funktion, die Sie mit den mit ~ gekennzeichneten Pins verwenden können, um eine variable Stromstärke an die entsprechenden LEDs auszugeben.

Super Starter Kit

Die RGB-LED hat vier Leitungen. Eine Leitung geht an den positiven Anschluss jeder einzelnen LED im Gehäuse und eine Leitung ist mit allen drei negativen Seiten der LEDs verbunden.



Hier im Bild sehen Sie eine LED mit 4 Elektroden (gemeinsame Kathode).

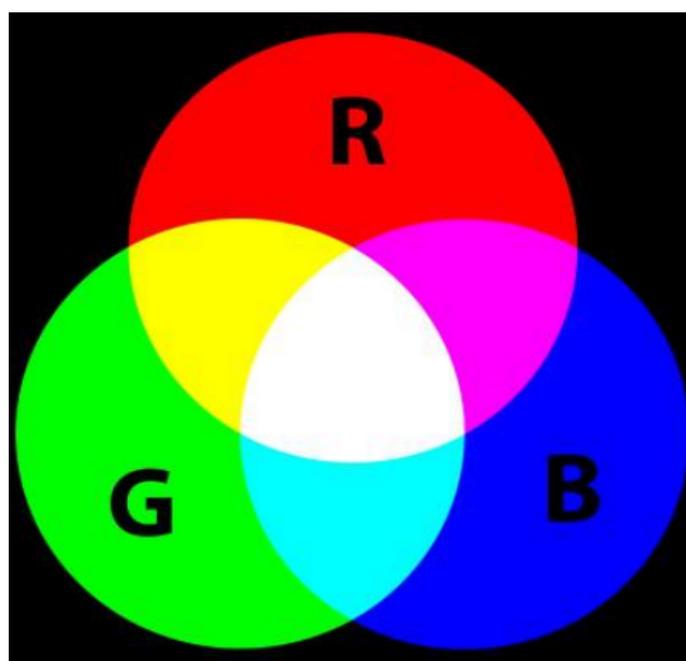
Super Starter Kit

Jeder einzelne Pin für die Farben Grün, Blau oder Rot wird Anode genannt. Sie werden immer "+" an sie anschließen. Die Kathode geht an "-" (Masse). Wenn Sie sie andersherum anschließen, leuchtet die LED nicht. Der gemeinsame negative Anschluss des LED-Gehäuses ist der zweite Stift von der flachen Seite. Er ist auch der längste der vier Anschlüsse und wird mit der Masse verbunden. Jede LED im Gehäuse benötigt einen eigenen 220-Ω-Widerstand, damit nicht zu viel Strom durch sie fließt. Die drei positiven Leitungen der LEDs (eine rote, eine grüne und eine blaue) werden über diese Widerstände mit den Ausgangspins des Mikrocontrollers verbunden.

Farbe

Der Grund dafür, dass Sie jede beliebige Farbe mischen können, indem Sie die Mengen an rotem, grünem und blauem Licht variieren, ist, dass Ihr Auge drei Arten von Lichtrezeptoren besitzt (rot, grün und blau).

Ihr Auge und Ihr Gehirn verarbeiten die Anteile an rotem, grünem und blauem Licht und wandeln sie in eine Farbe des Spektrums um. In gewisser Weise spielen wir dem Auge einen Streich, indem wir die drei LEDs verwenden. Die gleiche Idee wird bei Fernsehern verwendet, bei denen die LCD-Anzeige rote, grüne und blaue Farbpunkte nebeneinander hat, die jedes Pixel bilden.



Super Starter Kit

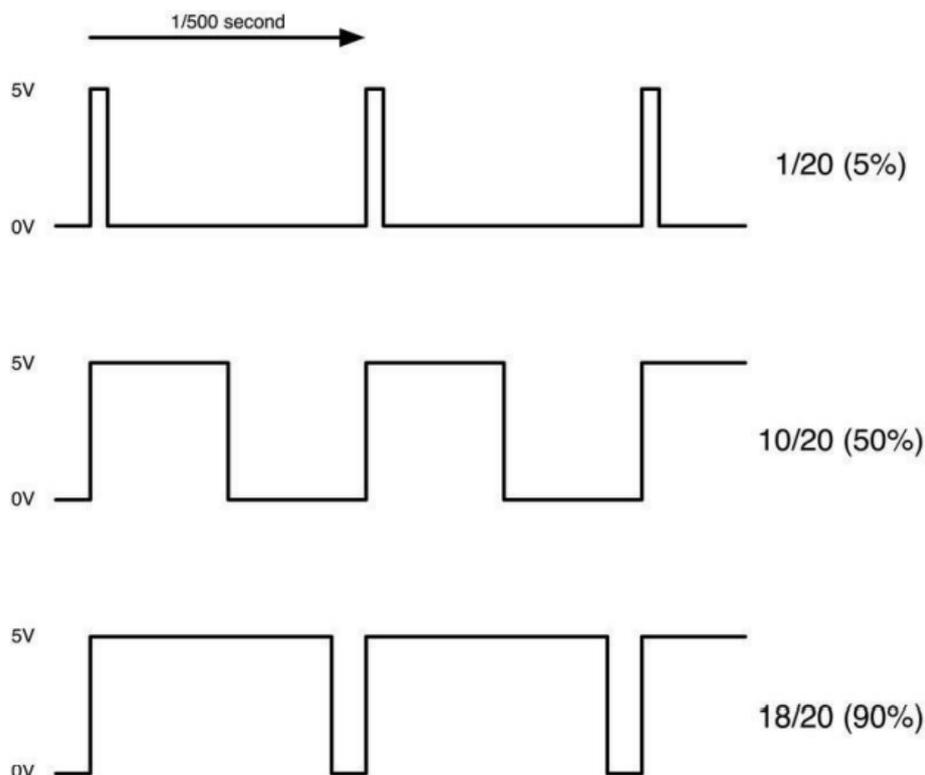
Wenn wir die Helligkeit aller drei LEDs gleich einstellen, dann ist die Gesamtfarbe des Lichts weiß. Wenn wir die blaue LED ausschalten, so dass nur die rote und die grüne LED die gleiche Helligkeit haben, wird das Licht gelb erscheinen.

Wir können die Helligkeit jedes roten, grünen und blauen Teils der LED separat steuern, so dass wir jede beliebige Farbe mischen können.

Schwarz ist weniger eine Farbe als vielmehr die Abwesenheit von Licht. Daher kommt man der Farbe Schwarz mit unserer LED am nächsten, wenn man alle LEDs ausschaltet.

Theorie PWM

Pulsweitenmodulation (PWM) ist eine Technik zur Steuerung der Leistung. Wir verwenden sie hier auch, um die Helligkeit der einzelnen LEDs zu steuern. Das Diagramm unten zeigt das Signal von einem der PWM-Pins am Mikrocontroller.

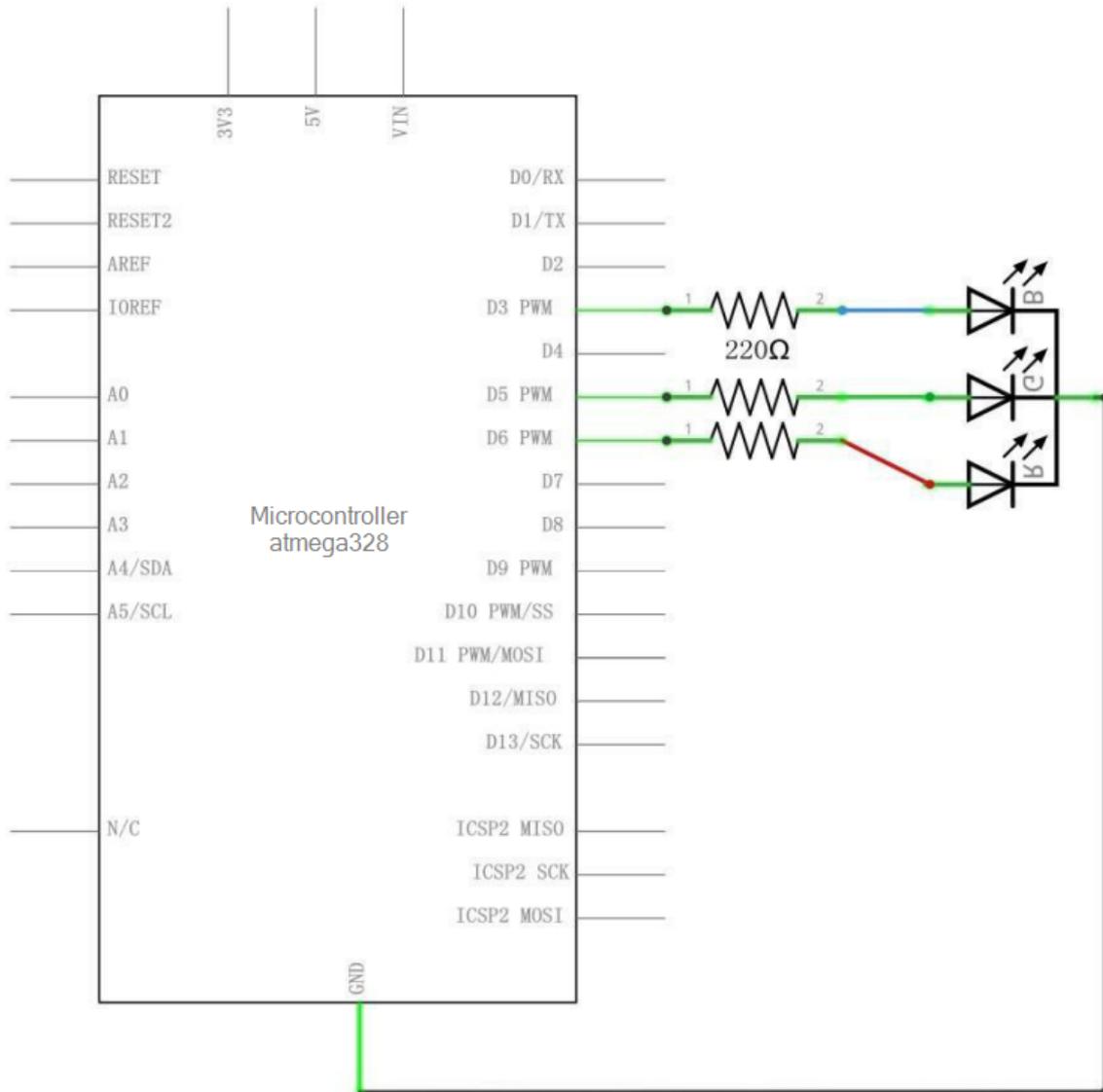


Super Starter Kit

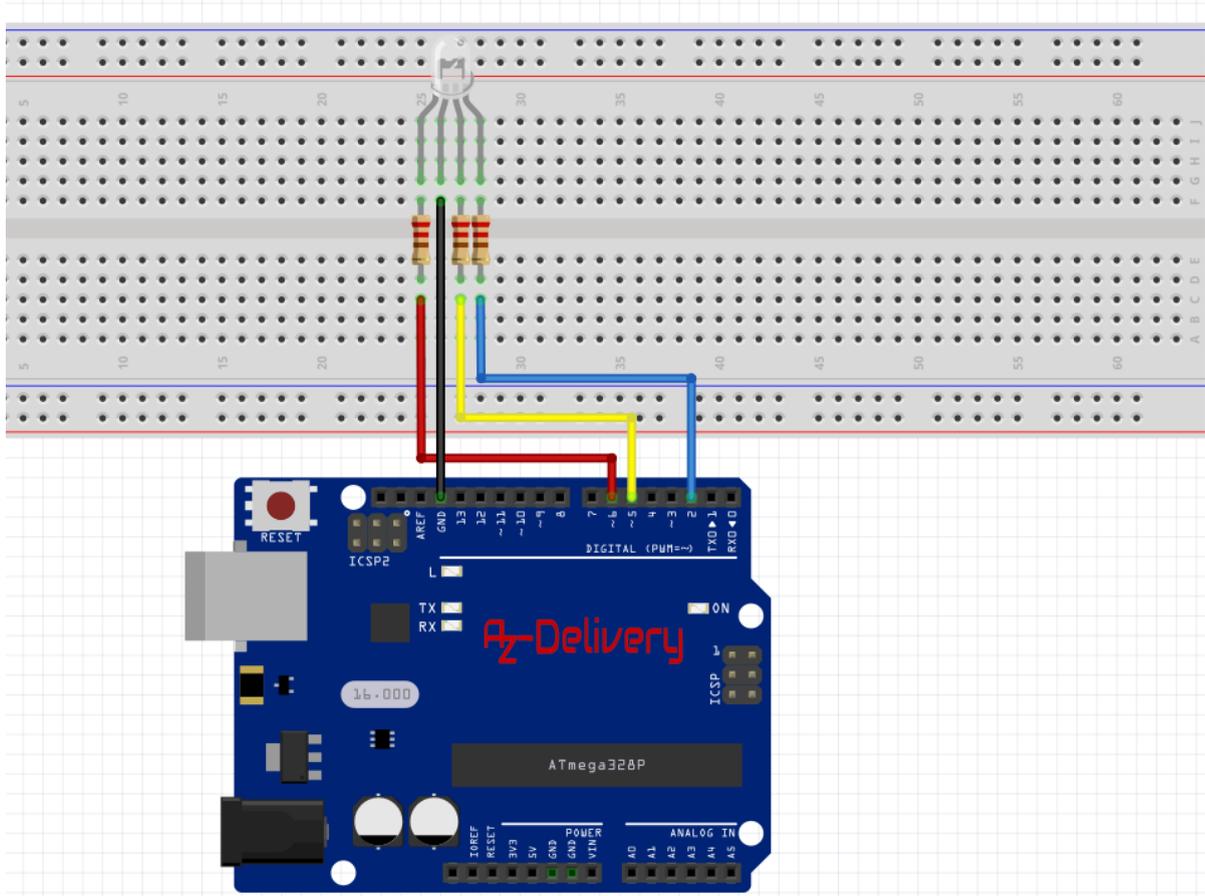
Ungefähr alle 1/500 Sekunde erzeugt der PWM-Ausgang einen Impuls. Die Länge dieses Impulses wird durch die Funktion 'analogWrite' gesteuert. So erzeugt "analogWrite(0)" überhaupt keinen Impuls und "analogWrite(255)" erzeugt einen Impuls, der bis zum nächsten fälligen Impuls anhält, so dass der Ausgang eigentlich die ganze Zeit eingeschaltet ist.

Wenn wir einen Wert in analogWrite angeben, der zwischen 0 und 255 liegt, erzeugen wir einen Impuls. Wenn der Ausgangsimpuls nur 5 % der Zeit hoch ist, erhält das Gerät, das wir ansteuern, nur 5 % der vollen Leistung. Liegt der Ausgang jedoch 90 % der Zeit auf 5 V, so erhält die Last 90 % der ihr zugeführten Leistung. Wir können nicht sehen, wie sich die LEDs mit dieser Geschwindigkeit ein- und ausschalten, also sieht es für uns nur so aus, als würde sich die Helligkeit ändern.

Schematische Darstellung



Schaltplan



Super Starter Kit

Beispiel-Code

```
// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6

#define delayTime 10 // fading time between colors

void setup()
{
  pinMode(RED, OUTPUT);
  pinMode(GREEN, OUTPUT);
  pinMode(BLUE, OUTPUT);
  digitalWrite(RED, HIGH);
  digitalWrite(GREEN, LOW);
  digitalWrite(BLUE, LOW);
}

// define variables
int redValue;
int greenValue;
int blueValue;

// main loop
void loop()
{
  redValue = 255; // choose a value between 1 and 255 to change the color.
  greenValue = 0;
  blueValue = 0;

  // this is unnecessary as we've either turned on RED in SETUP
  // or in the previous loop ... regardless, this turns RED off
  // analogWrite(RED, 0);
  // delay(1000);

  for (int i = 0; i < 255; i += 1) // fades out red bring green full when i=255
  {
    redValue -= 1;
    greenValue += 1;
  }
}
```

Super Starter Kit

```

// The following was reversed, counting in the wrong directions
// analogWrite(RED, 255 - redValue);
// analogWrite(GREEN, 255 - greenValue);
analogWrite(RED, redValue);
analogWrite(GREEN, greenValue);
delay(delayTime);
}

redValue = 0;
greenValue = 255;
blueValue = 0;

for (int i = 0; i < 255; i += 1) // fades out green bring blue full when i=255
{
    greenValue -= 1;
    blueValue += 1;
    // The following was reversed, counting in the wrong directions
    // analogWrite(GREEN, 255 - greenValue);
    // analogWrite(BLUE, 255 - blueValue);
    analogWrite(GREEN, greenValue);
    analogWrite(BLUE, blueValue);
    delay(delayTime);
}

redValue = 0;
greenValue = 0;
blueValue = 255;

for (int i = 0; i < 255; i += 1) // fades out blue bring red full when i=255
{
    // The following code has been rearranged to match the other two similar
sections
    blueValue -= 1;
    redValue += 1;
    // The following was reversed, counting in the wrong directions
    // analogWrite(BLUE, 255 - blueValue);
    // analogWrite(RED, 255 - redValue);
    analogWrite(BLUE, blueValue);
    analogWrite(RED, redValue);
    delay(delayTime);
}
}

```

Super Starter Kit

Sie können diesen Code in die Arduino IDE kopieren und in den Mikrocontroller hochladen.

Unser Code wird FOR-Schleifen verwenden, um die Farben zu durchlaufen. Die erste FOR-Schleife wird von ROT nach GRÜN gehen. Die zweite FOR-Schleife geht von GRÜN nach BLAU. Die letzte FOR-Schleife geht von BLAU nach ROT.

Die Skizze beginnt mit der Angabe der Pins, die für die einzelnen Farben verwendet werden sollen:

```
// Define Pins
#define BLUE 3
#define GREEN 5
#define RED 6
```

Wir legen auch die Überblendzeit zwischen den Farben fest:

```
#define delayTime 10 // fading time between colors
```

Der nächste Schritt besteht darin, die Funktion "setup" zu schreiben. Wie wir bereits gelernt haben, wird die Setup-Funktion nur einmal ausgeführt, nachdem der Arduino zurückgesetzt wurde. In diesem Fall muss sie nur die drei Pins, die wir verwenden, als Ausgänge definieren.

In der 'loop'-Funktion kannst du die FOR-Schleifen sehen, die die Farben überblenden.

Experimentieren Sie mit den Werten und der Verzögerungszeit und beobachten Sie die Wirkung auf Ihre LED.

Super Starter Kit

Digitale Eingänge

In diesem Beispiel lernen Sie, wie Sie Drucktasten mit digitalen Eingängen verwenden können, um eine LED ein- und auszuschalten.

Das Drücken der ersten Taste schaltet die LED ein, das Drücken der zweiten Taste schaltet die LED aus.

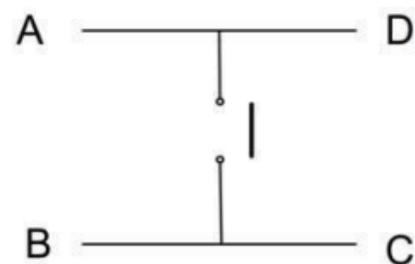
Für dieses Beispiel benötigst du:

- Mikrocontroller-Platine
- 830-poliges Breadboard
- 1 Stück 5mm rote LED
- 1 Stück 220-Ohm-Widerstand
- 2 Stck. Drucktasten
- 7 St. Überbrückungsdrähte

Schalter sind wirklich einfache Bauteile.

Wenn du einen Knopf drückst oder einen Hebel umlegst, verbinden sie zwei Kontakte miteinander, so dass Strom durch sie fließen kann.

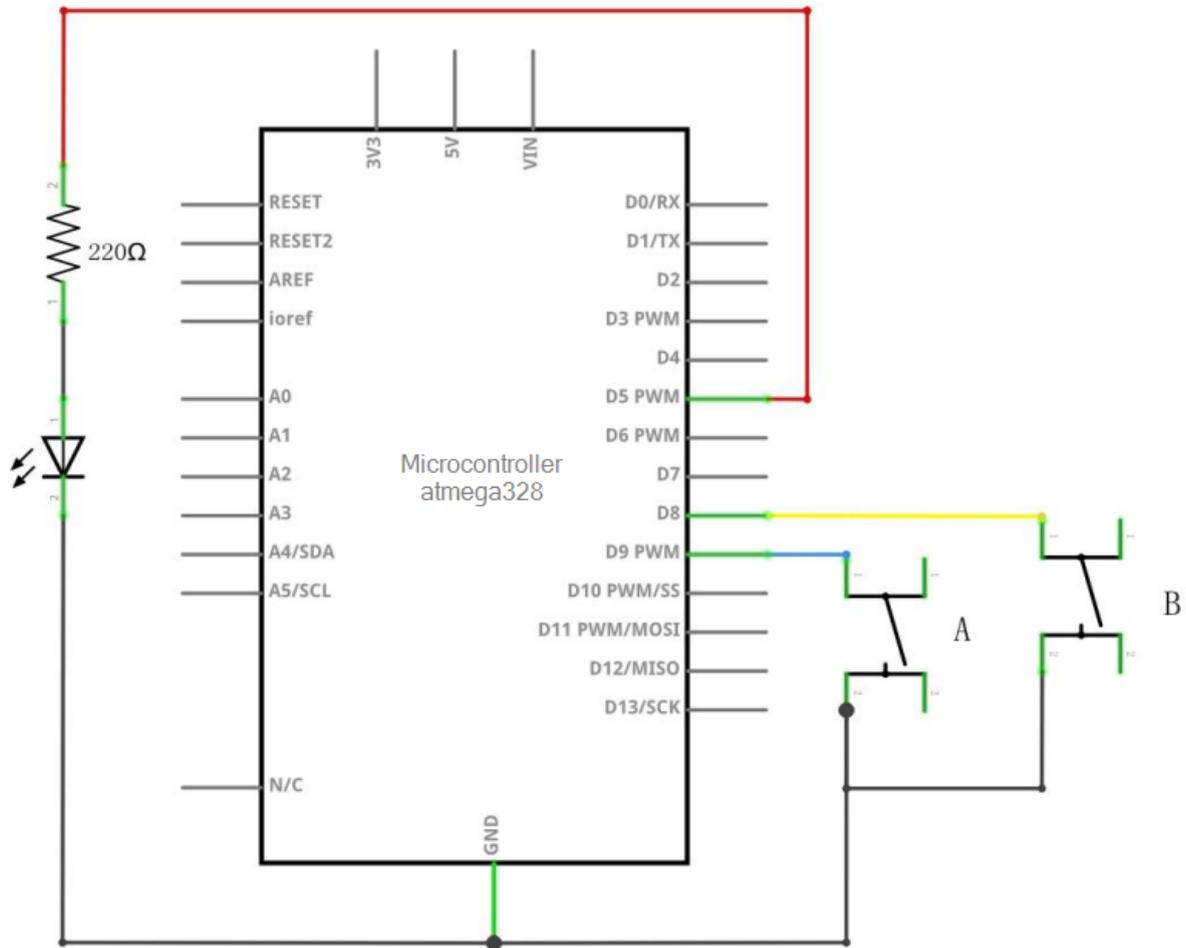
Die kleinen taktilen Schalter, die in diesem Beispiel verwendet werden, haben vier Anschlüsse, was ein wenig verwirrend sein kann.



Eigentlich gibt es nur zwei elektrische Verbindungen. Im Inneren des Schaltergehäuses sind die Stifte B und C miteinander verbunden, ebenso wie A und D.

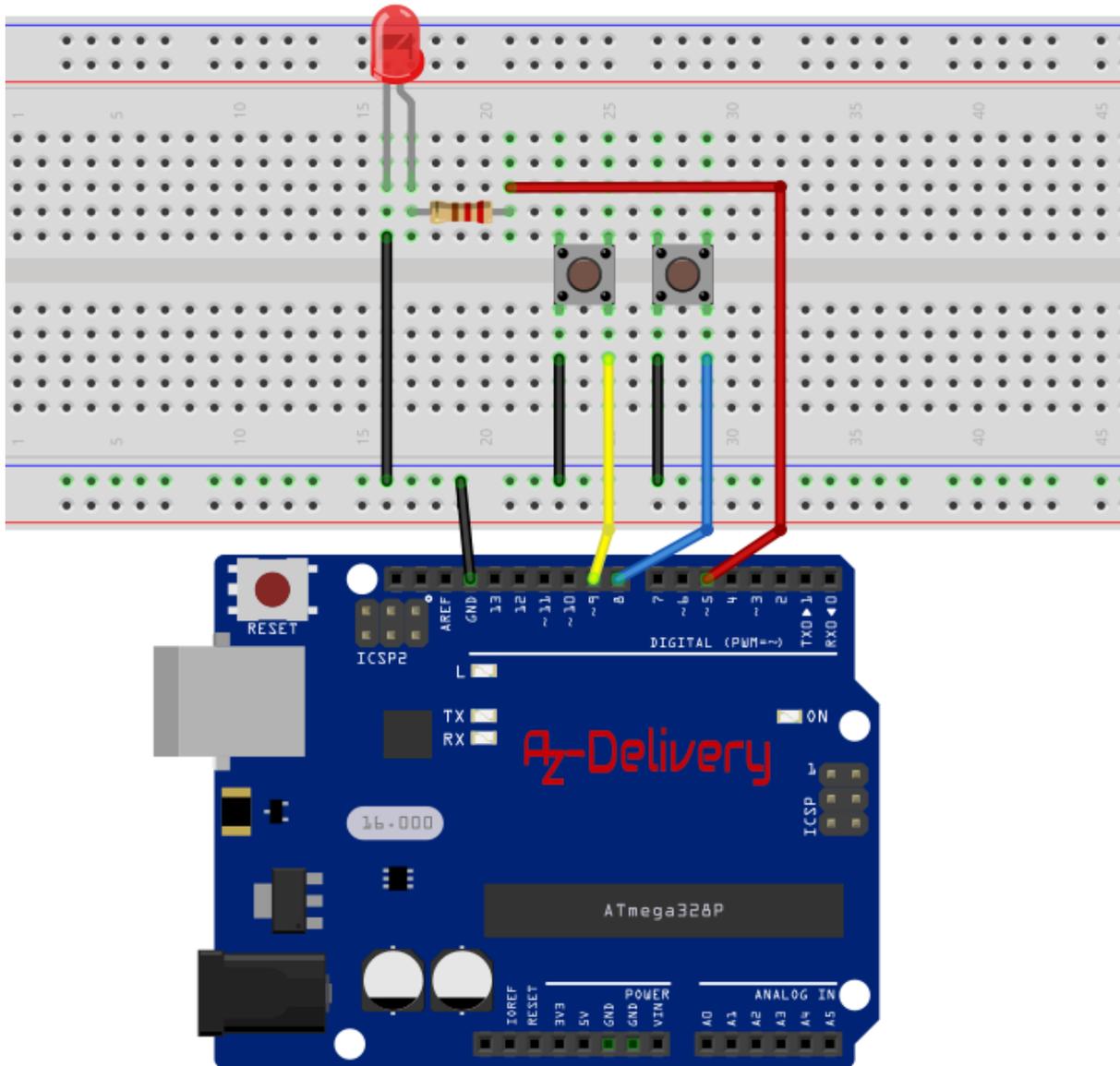
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Super Starter Kit

Beispiel Code

```
int ledPin = 5;
int buttonApin = 9;
int buttonBpin = 8;

byte leds = 0;

void setup()
{
  pinMode(ledPin, OUTPUT);
  pinMode(buttonApin, INPUT_PULLUP);
  pinMode(buttonBpin, INPUT_PULLUP);
}

void loop()
{
  if (digitalRead(buttonApin) == LOW)
  {
    digitalWrite(ledPin, HIGH);
  }
  if (digitalRead(buttonBpin) == LOW)
  {
    digitalWrite(ledPin, LOW);
  }
}
```

Kopieren Sie diesen Code in die Arduino IDE und laden Sie ihn auf den Mikrocontroller hoch.

Der erste Teil der Skizze definiert drei Variablen für die drei zu verwendenden Pins.

Der "ledPin" ist der Ausgangspin und "buttonApin" bezieht sich auf den Schalter, der näher an der Oberseite des Breadboards liegt und "buttonBpin" auf den anderen Schalter.

Super Starter Kit

Die Funktion "setup" definiert den ledPin wie üblich als OUTPUT, aber jetzt haben wir es mit zwei Eingängen zu tun. In diesem Fall setzen wir den pinMode auf 'INPUT_PULLUP' wie folgt:

```
pinMode(buttonApin, INPUT_PULLUP);  
pinMode(buttonBpin, INPUT_PULLUP);
```

Der Pin-Modus INPUT_PULLUP bedeutet, dass der Pin als Eingang verwendet werden soll, aber wenn nichts anderes an den Eingang angeschlossen ist, sollte er auf HIGH "gezogen" werden.

Mit anderen Worten: Der Standardwert für den Eingang ist HIGH, es sei denn, er wird durch das Drücken der Taste auf LOW gezogen. Aus diesem Grund sind die Schalter mit GND verbunden. Wenn ein Schalter gedrückt wird, verbindet er den Eingangspin mit GND, so dass er nicht mehr HIGH ist. Da der Eingang normalerweise HIGH ist und nur dann auf LOW geht, wenn die Taste gedrückt wird, ist die Logik ein wenig auf den Kopf gestellt.

In der Funktion "Schleife" gibt es zwei "if"-Anweisungen. Eine für jede Taste. Jede führt ein 'digitalRead' am entsprechenden Eingang aus. Denken Sie daran, dass, wenn die Taste gedrückt wird, der entsprechende Eingang auf LOW steht. Wenn die Taste A auf LOW steht, schaltet ein 'digitalWrite' auf den ledPin diesen ein. Ähnlich verhält es sich, wenn Knopf B gedrückt wird, wird ein LOW an den ledPin geschrieben.

Neigungskugelschalter

In diesem Beispiel lernen Sie, wie man einen Kipp-Kugelschalter verwendet, um einen kleinen Neigungswinkel zu erkennen.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- 2 Stück M-F-Drähte (weibliche auf männliche Dupont-Drähte)
- Kipp-Kugelschalter



Neigungssensoren (Neigungskugelschalter) ermöglichen die Erkennung von Orientierung oder Neigung.

Sie sind klein, preiswert, stromsparend und einfach zu bedienen. Bei ordnungsgemäßer Verwendung verschleiben sie nicht. Aufgrund ihrer Einfachheit werden sie gerne für Spielzeug, Gadgets und Geräte verwendet.

Manchmal werden sie aus offensichtlichen Gründen auch als "Neigungsschalter" oder "Rollkugelsensoren" bezeichnet.

Sie bestehen in der Regel aus einer Art Hohlraum (häufig, aber nicht immer, zylindrisch) mit einer leitfähigen freien Masse im Inneren, z. B. einer rollenden Kugel.

Ein Ende des Hohlraums hat zwei leitende Elemente (Pole). Wenn der Sensor so ausgerichtet wird, dass das Ende nach unten zeigt, rollt die Masse auf die Pole und schließt sie kurz, was wie ein Schalter wirkt.

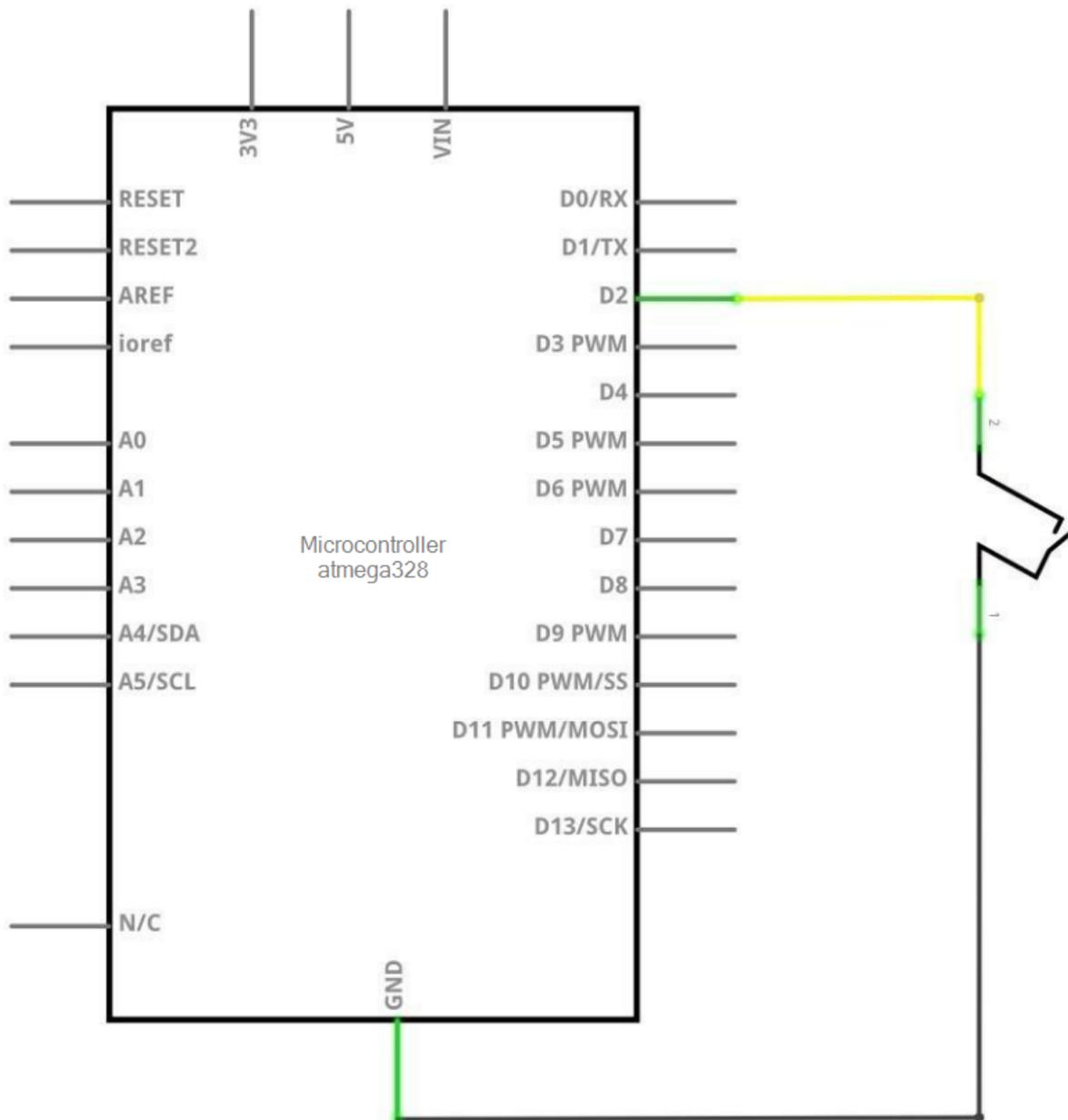
Super Starter Kit

Neigungsschalter sind zwar nicht so präzise oder flexibel wie ein vollständiger Beschleunigungssensor, können aber Bewegungen oder Ausrichtungen erkennen.

Ein weiterer Vorteil ist, dass die großen Schalter selbständig Strom schalten können. Beschleunigungssensoren hingegen geben digitale oder analoge Spannungen aus, die dann mit zusätzlichen Schaltungen analysiert werden müssen.

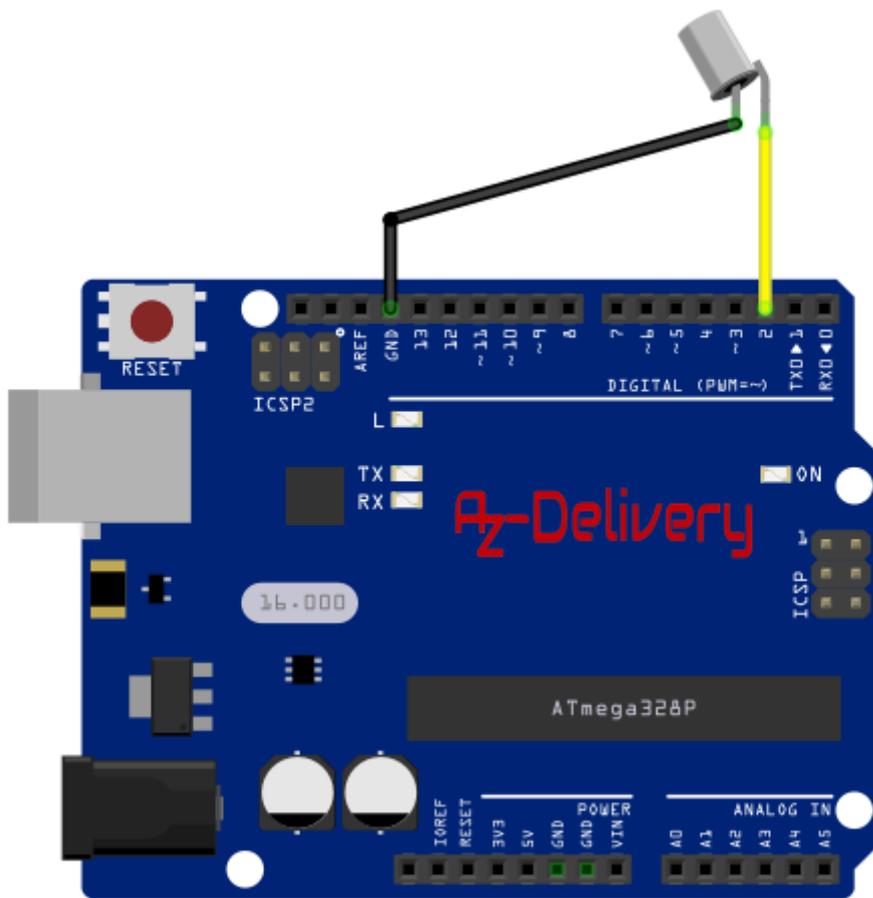
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Super Starter Kit

Beispiel-Code

```
#define tilt 2

void setup()
{
  pinMode(LED_BUILTIN,OUTPUT);
  pinMode(tilt,INPUT_PULLUP);
}

void loop()
{
  if(digitalRead(tilt)==HIGH)
  {
    digitalWrite(LED_BUILTIN,LOW);//turn the led off
  }
  else
  {
    digitalWrite(LED_BUILTIN,HIGH);//turn the led on
  }
}
```

Kopieren Sie diesen Code in die Arduino IDE und laden Sie ihn auf den Mikrocontroller hoch.

Wenn du den Schalter kippst, schaltet sich die integrierte LED des Mikrocontrollers ein.

Super Starter Kit

Aktiver Buzzer

In diesem Beispiel lernen Sie, wie Sie mit einem aktiven Summer einen Ton erzeugen können.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- Aktiver Buzzer
- 2 Stück F-M-Drähte (Buchsen-zu-Männchen-Dupont-Drähte)

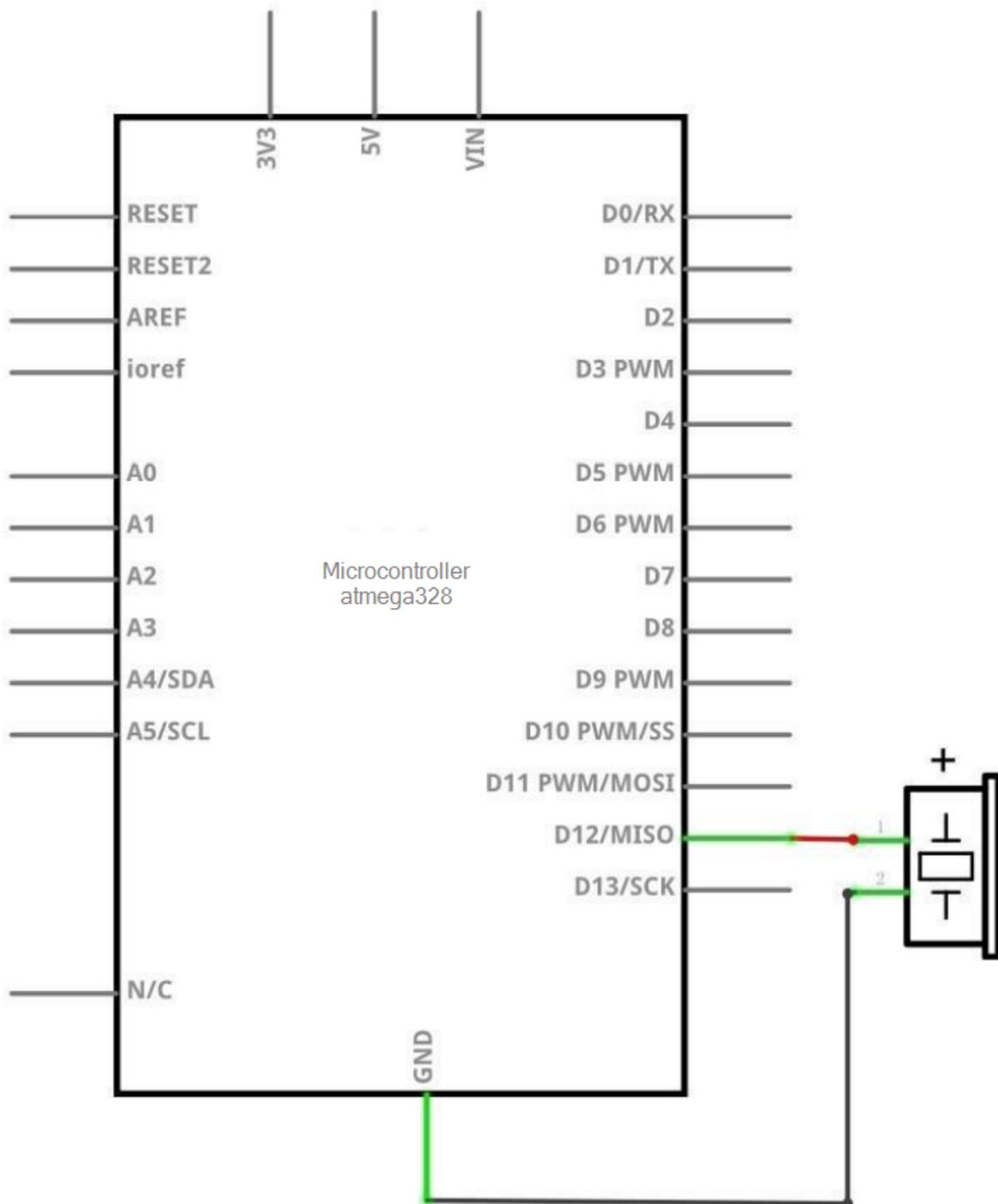


Elektronische Buzzer werden mit Gleichstrom betrieben und sind mit einem integrierten Schaltkreis ausgestattet. Sie sind weit verbreitet in Computern, Druckern, Fotokopierern, Alarmanlagen, elektronischem Spielzeug, elektronischen Geräten für die Automobilindustrie, Telefonen, Zeitschaltuhren und anderen elektronischen Produkten für Sprachgeräte. Buzzer können in aktive und passive unterteilt werden. Drehen Sie die Stifte von zwei Buzzern nach oben. Der eine mit einer grünen Platine ist ein passiver Summer, während der andere, der mit einem schwarzen Klebeband umschlossen ist, ein aktiver Summer ist.

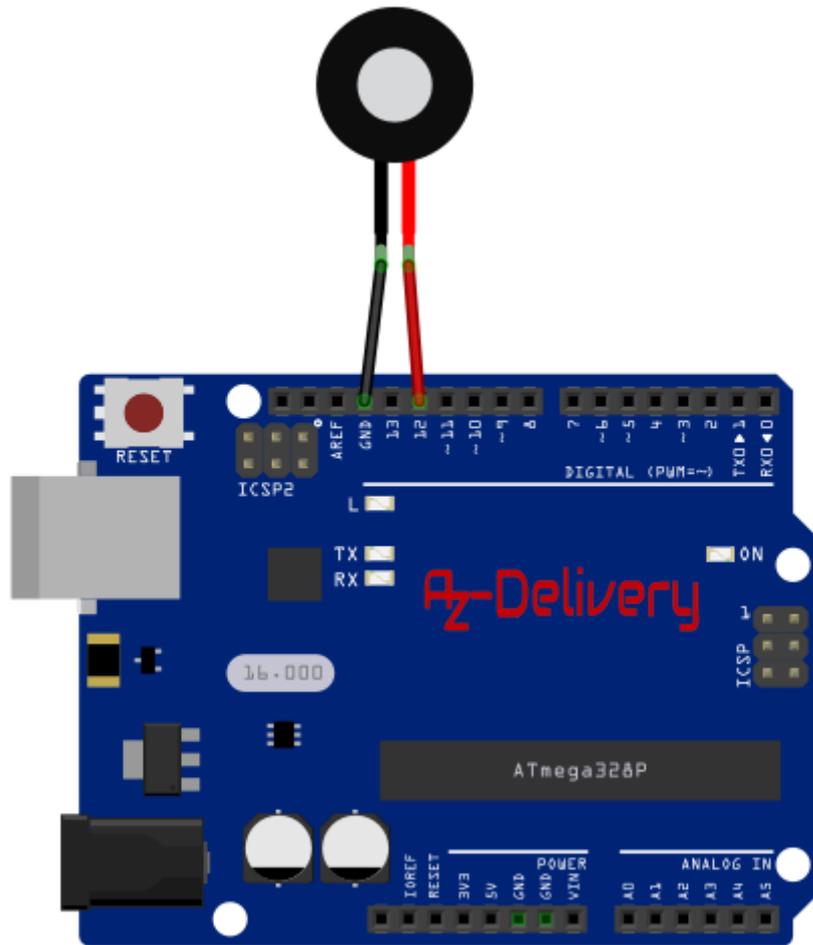
Der Unterschied zwischen den beiden besteht darin, dass ein aktiver Summer eine eingebaute Schwingungsquelle hat, so dass er einen Ton erzeugt, wenn er unter Strom steht. Ein passiver Summer hat keine solche Quelle, so dass er nicht zwitschert, wenn Gleichstromsignale verwendet werden; stattdessen müssen Sie Rechteckwellen verwenden, deren Frequenz zwischen 2K und 5K liegt, um ihn zu betreiben. Ein aktiver Summer ist oft teurer als ein passiver, da mehrere Schwingkreise eingebaut sind.

Super Starter Kit

Schematische Darstellung



Schaltplan



Beispiel-Code

```
int buzzer = 12;//the pin of the active buzzer
void setup()
{
  pinMode(buzzer, OUTPUT);
}
void loop()
{
  //output an frequency
  for (int i = 0; i < 80; i++)
  {
    digitalWrite(buzzer, HIGH);
    delay(1);//wait for 1ms
    digitalWrite(buzzer, LOW);
    delay(1);//wait for 1ms
  }
  delay(500);
  //output another frequency
  for (int i = 0; i < 100; i++)
  {
    digitalWrite(buzzer, HIGH);
    delay(2);//wait for 2ms
    digitalWrite(buzzer, LOW);
    delay(2);//wait for 2ms
  }
  delay(500);
}
```

Kopieren Sie diesen Code in die Arduino IDE und laden Sie ihn auf den Mikrocontroller hoch.

Er wird zwei Töne mit einer Pause von 0,5 Sekunden dazwischen erzeugen.

Experimentieren Sie mit den Werten, um verschiedene Töne zu erzeugen.

Passiver Buzzer

In diesem Beispiel lernen Sie, wie man einen passiven Summer verwendet. Ziel des Beispiels ist es, acht verschiedene Töne mit einer Dauer von jeweils 0,5 Sekunden zu erzeugen: von Alt Do (523Hz), Re (587Hz), Mi (659Hz), Fa (698Hz), So (784Hz), La (880Hz), Si (988Hz) bis zu Alt Do (1047Hz).

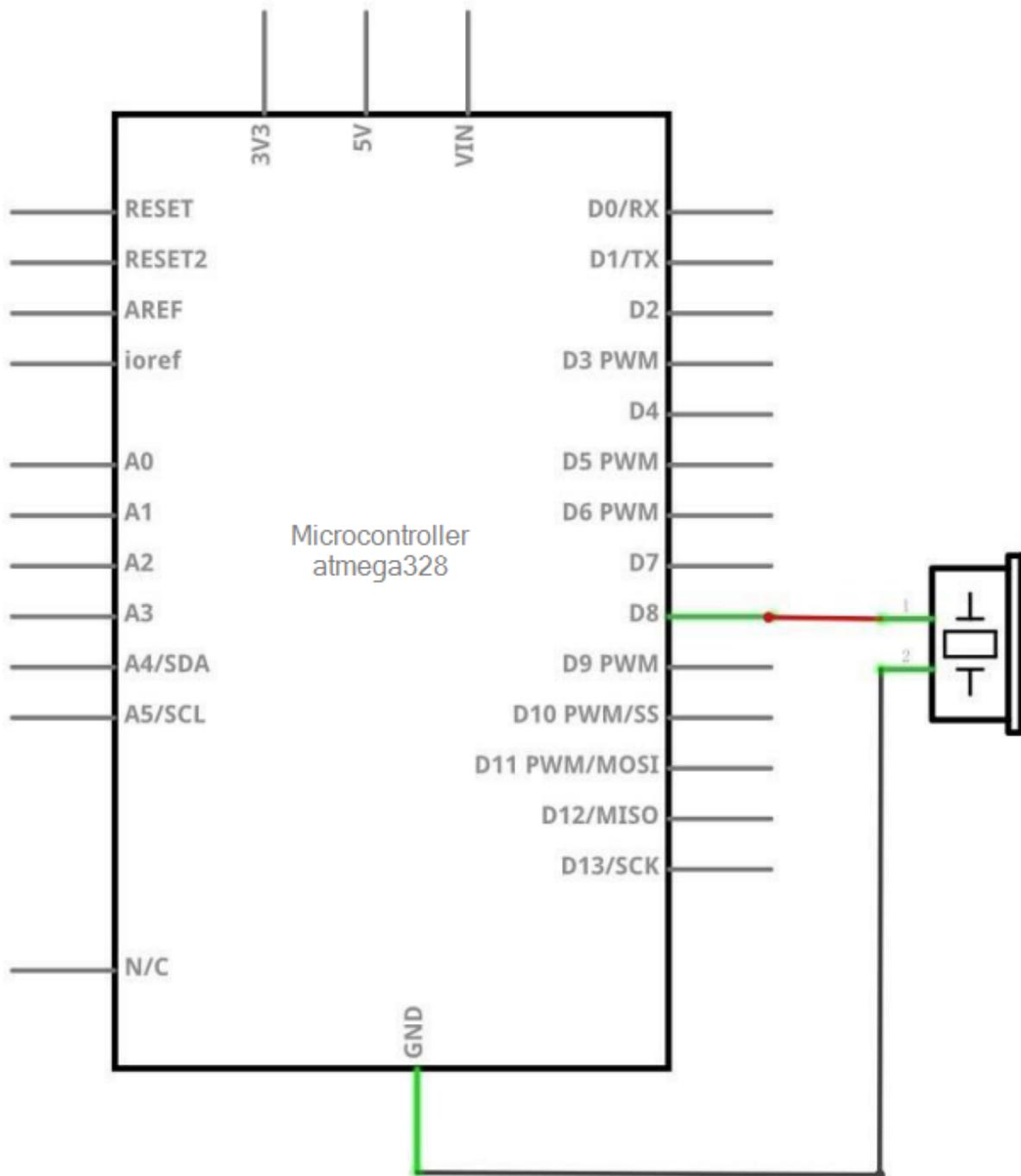
Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- Passiver Buzzer
- 2 Stück F-M-Drähte (weibliche und männliche Dupont-Drähte)

Das Funktionsprinzip des passiven Summers ist die Verwendung von PWM zur Erzeugung von Audio, um die Luft zum Vibrieren zu bringen. Solange die Vibrationsfrequenz entsprechend geändert wird, können verschiedene Töne erzeugt werden. Zum Beispiel, das Senden eines Impulses von 523Hz, kann es Alto Do erzeugen, Impuls von 587Hz, kann es Midrange Re erzeugen, Impuls von 659Hz, kann es Midrange Mi erzeugen. Mit dem Buzzer kann man ein Lied spielen. Wir sollten vorsichtig sein, nicht zu verwenden, die analoge Write () Funktion, um einen Impuls an den Summer zu erzeugen, weil der Impulsausgang des analogen Write () ist fest (500Hz).

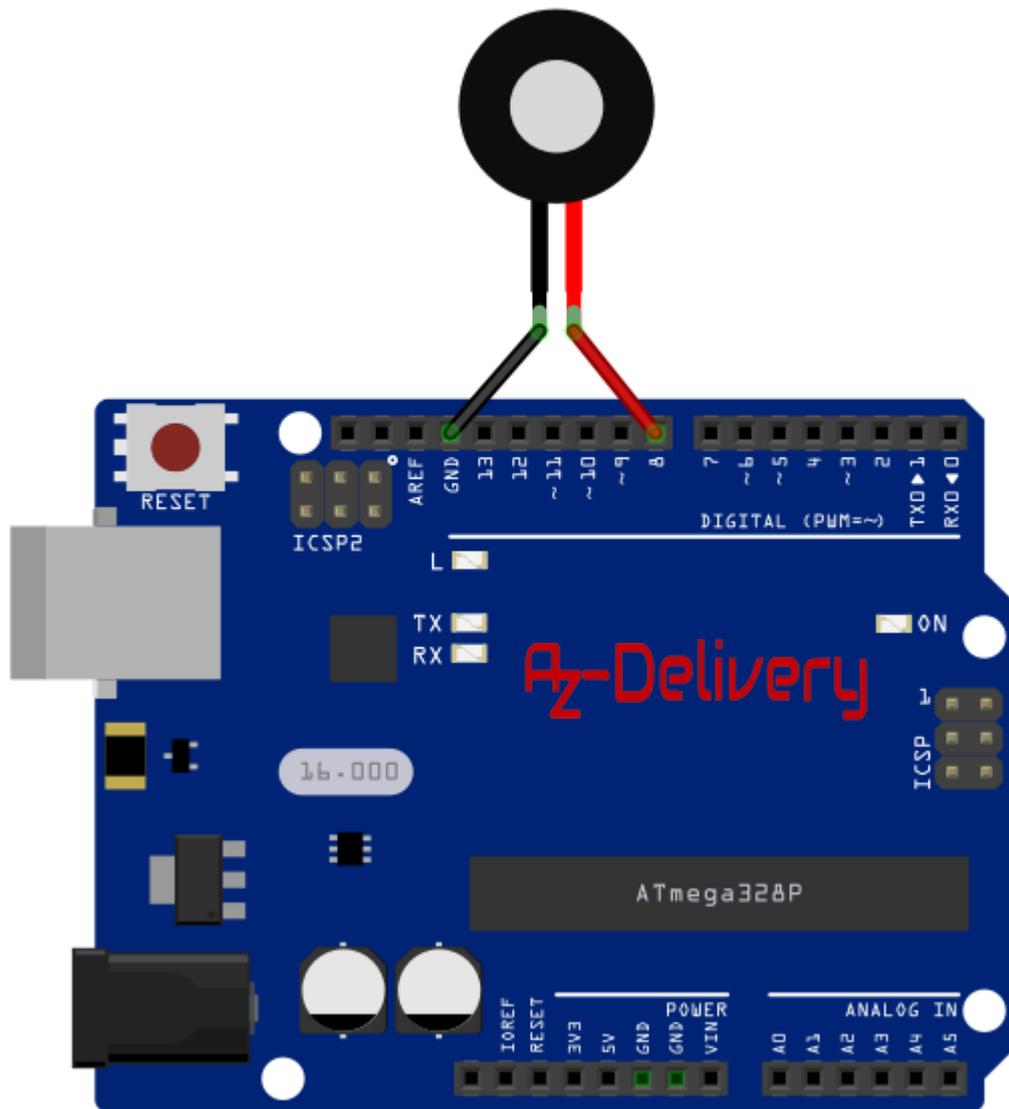
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Beispiel für eine Verdrahtung



Super Starter Kit

Beispiel-Code

```
int melody[] = {
  523, 587, 659, 698, 784, 880, 988, 1047};
int duration = 500; // 500 milliseconds

void setup() {

}

void loop() {
  for (int thisNote = 0; thisNote < 8; thisNote++) {
    // pin8 output the sound, every scale is 0.5 second
    tone(8, melody[thisNote], duration);
    delay(1000);
  }
  // restart after two seconds
  delay(2000);
}
```

Kopieren Sie diesen Code in die Arduino IDE und laden Sie ihn auf den Mikrocontroller hoch.

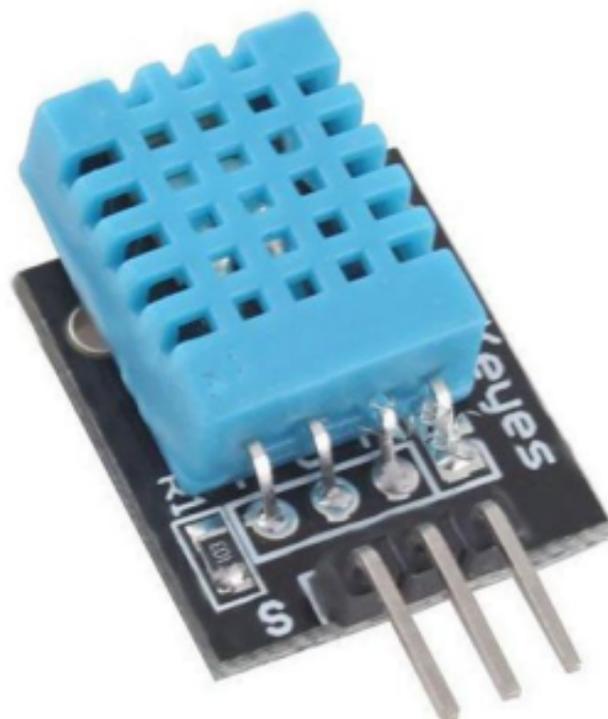
DHT11-Sensor

In diesem Beispiel lernen Sie, wie Sie einen DHT11 Temperatur- und Feuchtigkeitssensor verwenden.

Er ist genau genug für die meisten Projekte, bei denen die Luftfeuchtigkeit und die Temperatur gemessen werden müssen. Wir werden eine Bibliothek verwenden, die speziell für diese Sensoren entwickelt wurde, so dass unser Code kurz und einfach zu schreiben ist.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- DHT11-Sensormodul
- 3 Stück F-M-Drähte (weibliche zu männlichen Dupont-Drähten)



Der digitale Temperatur- und Feuchtigkeitssensor DHT11 ist ein zusammengesetzter Sensor, der einen kalibrierten digitalen Signalausgang für Temperatur und Feuchtigkeit enthält.

Super Starter Kit

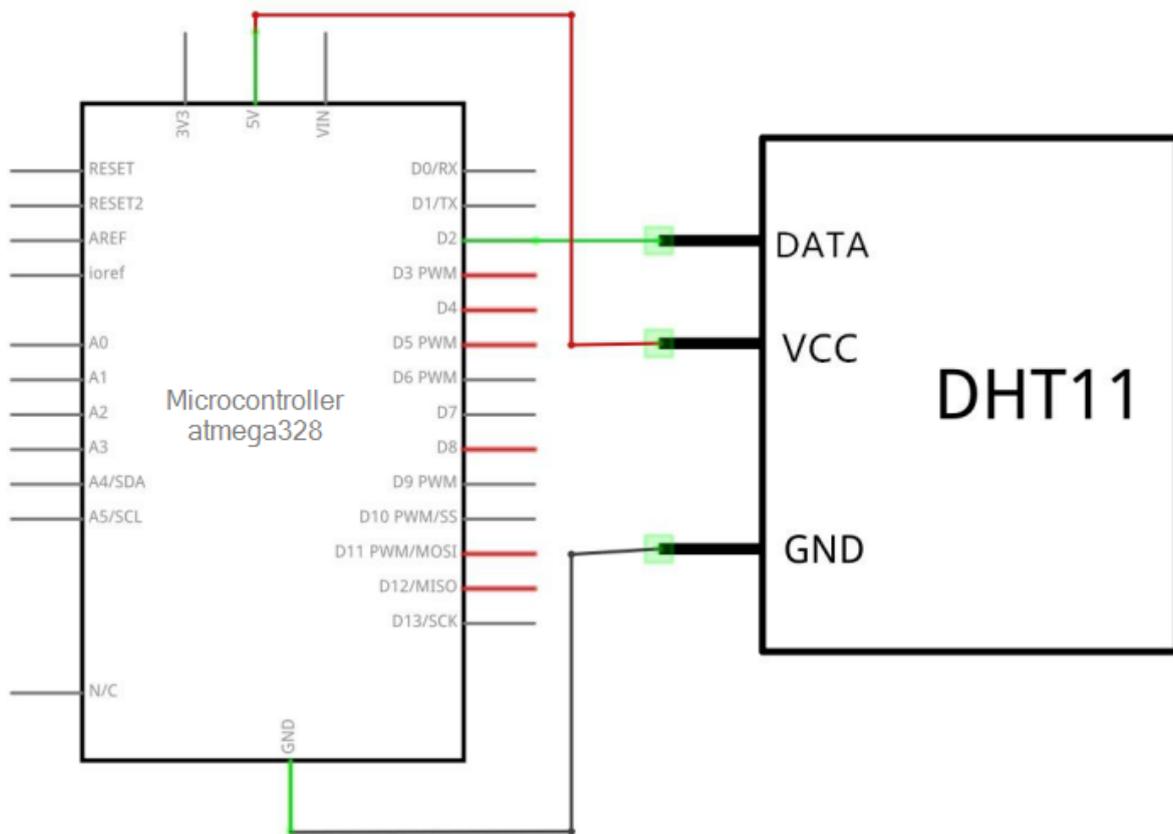
Die dedizierte digitale Modulerfassungstechnologie und die Temperatur- und Luftfeuchtigkeitssensortechnologie werden eingesetzt, um sicherzustellen, dass das Produkt eine hohe Zuverlässigkeit und ausgezeichnete Langzeitstabilität aufweist.

Der Sensor enthält einen Widerstandsfühler für feuchte Komponenten und ein NTC-Temperaturmessgerät und ist mit einem leistungsstarken 8-Bit-Mikrocontroller verbunden.

Anwendungen: HVAC, Luftentfeuchter, Prüf- und Inspektionsgeräte, Konsumgüter, Automobil, automatische Steuerung, Datenlogger, Wetterstationen, Haushaltsgeräte, Feuchtigkeitsregler, medizinische und andere Feuchtigkeitsmessung und -steuerung.

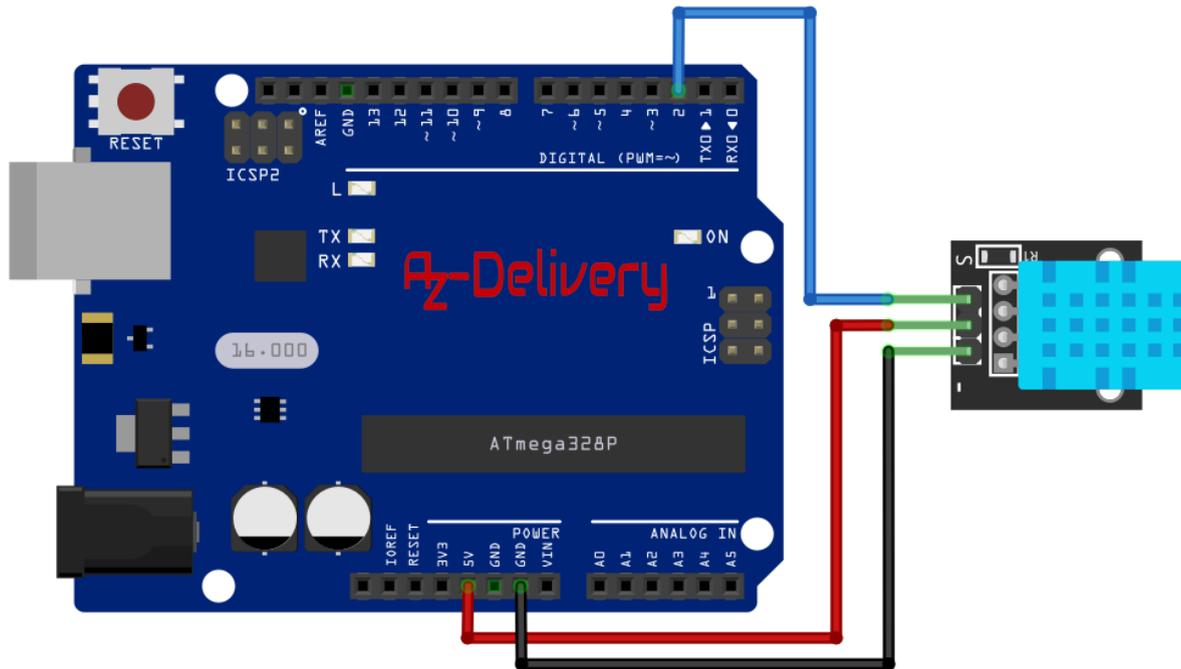
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Super Starter Kit

Beispiel-Code

Für dieses Beispiel müssen Sie die "SimpleDHT"-Bibliothek installieren. Suchen Sie im Bibliotheksmanager nach SimpleDHT und installieren Sie sie.

```

#include <SimpleDHT.h>
// for DHT11,
//     VCC: 5V or 3V
//     GND: GND
//     DATA: 2
int pinDHT11 = 2;
SimpleDHT11 dht11;

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

void loop() {
  // start working...
  Serial.println("=====");
  Serial.println("Sample DHT11...");

  // read with raw sample data.
  byte temperature = 0;
  byte humidity = 0;
  byte data[40] = {0};
  if (dht11.read(pinDHT11, &temperature, &humidity, data)) {
    Serial.print("Read DHT11 failed");
    return;
  }

  Serial.print("Sample RAW Bits: ");
  for (int i = 0; i < 40; i++) {
    Serial.print((int)data[i]);
    if (i > 0 && ((i + 1) % 4) == 0) {
      Serial.print(' ');
    }
  }
  Serial.println("");
  Serial.print("Sample OK: ");

```

Super Starter Kit

```

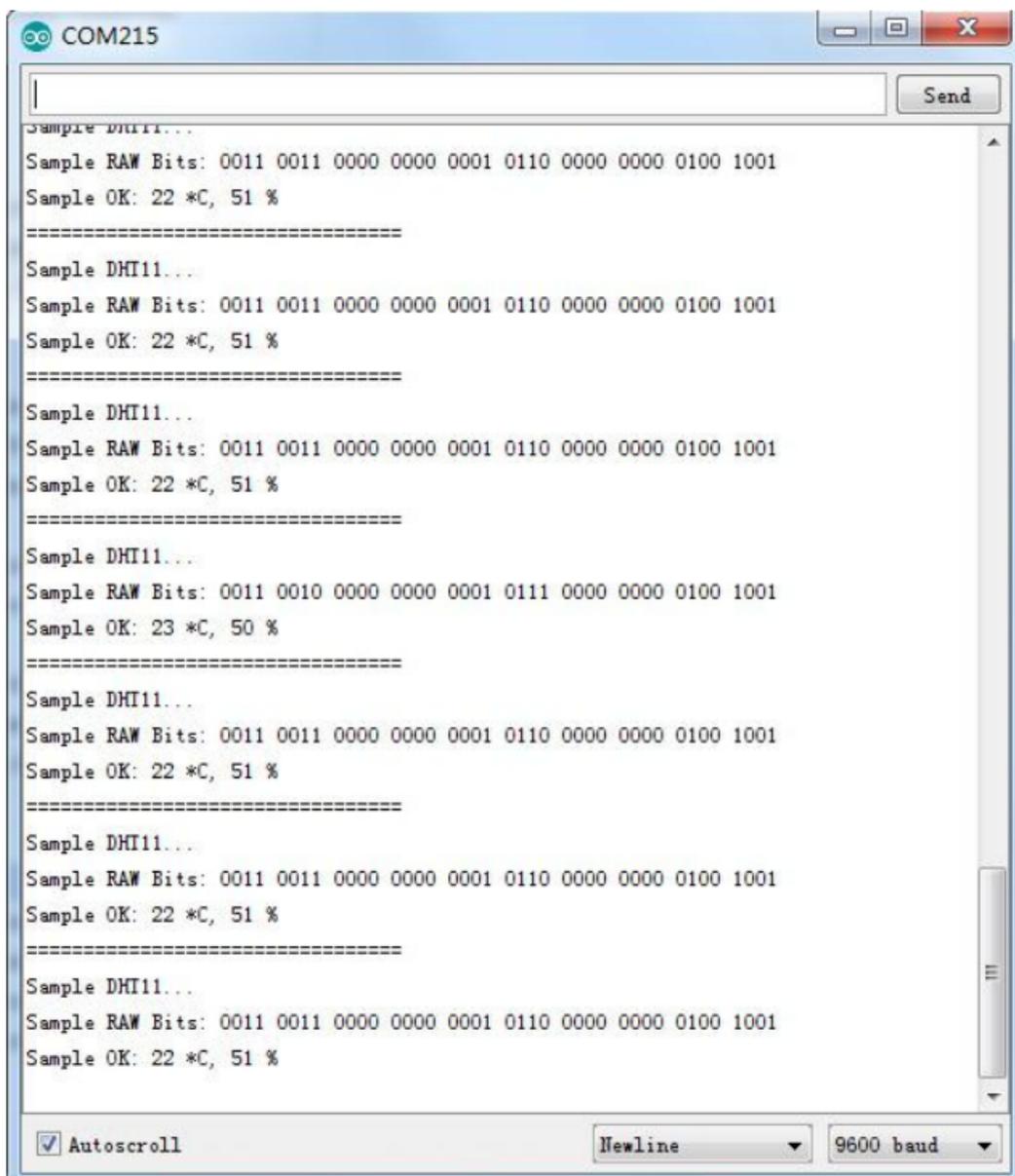
Serial.print((int)temperature); Serial.print(" *C, ");
Serial.print((int)humidity); Serial.println(" %");
delay(1000);
}

```

Kopieren Sie diesen Code in die Arduino IDE und laden Sie ihn auf den Mikrocontroller hoch.

Nachdem der Upload abgeschlossen ist, öffnen Sie den Serial Monitor, um die Ausgabe zu sehen.

Er zeigt die Temperatur der Umgebung an. Wir können sehen, dass sie hier 22 Grad beträgt.



IR-Empfängermodul

Die Verwendung einer IR-Fernbedienung ist eine hervorragende Möglichkeit, Ihr Projekt drahtlos zu steuern. Infrarot-Fernbedienungen sind einfach und leicht zu bedienen.

In diesem Beispiel werden wir den IR-Empfänger an den Mikrocontroller anschließen und dann eine Bibliothek verwenden, die für diesen speziellen Sensor entwickelt wurde.

In unserem Sketch werden wir alle hexadezimalen IR-Codes haben, die auf dieser Fernbedienung verfügbar sind. Wir werden auch erkennen, ob der Code erkannt wurde und auch, ob wir eine Taste gedrückt halten.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- IR-Empfängermodul
- IR-Fernbedienung
- 3 Stück F-M-Drähte (weibliche und männliche Dupont-Drähte)

IR-Detektoren sind kleine Mikrochips mit einer Fotozelle, die auf Infrarotlicht eingestellt sind. Sie werden fast immer für die Erkennung von Fernbedienungen verwendet - jeder Fernseher und jeder DVD-Player hat einen solchen Chip an der Vorderseite, um auf das IR-Signal des Klickers zu hören.

Im Inneren der Fernbedienung befindet sich eine passende IR-LED, die IR-Impulse aussendet, um dem Fernsehgerät mitzuteilen, dass es sich ein- oder ausschalten oder den Kanal wechseln soll. IR-Licht ist für das menschliche Auge nicht sichtbar, was bedeutet, dass es ein wenig mehr Arbeit erfordert, eine Einrichtung zu testen.

Super Starter Kit

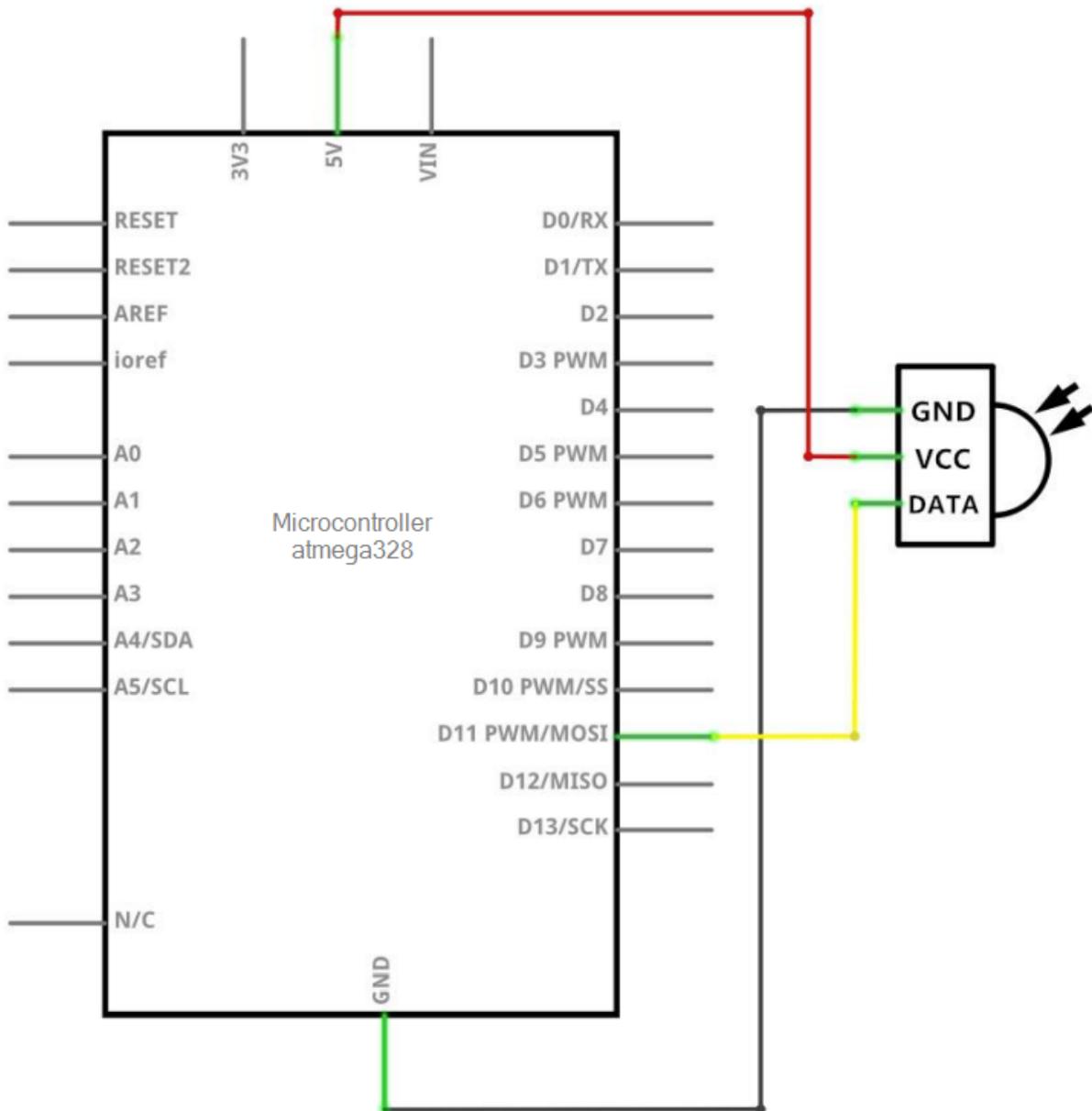
Es gibt einige Unterschiede zwischen diesen und z. B. einer CdS-Fotозelle: IR-Detektoren sind speziell für IR-Licht gefiltert, sie können sichtbares Licht nicht gut erkennen. Fotozellen hingegen sind gut im Erkennen von gelbem/grünem sichtbarem Licht und nicht gut im Erkennen von IR-Licht.

IR-Detektoren haben einen Demodulator im Inneren, der nach moduliertem IR-Licht mit 38 KHz sucht. Wenn eine IR-LED einfach nur leuchtet, wird sie nicht erkannt, sie muss mit 38 KHz PWM blinken. Fotozellen haben keine Art von Demodulator und können jede Frequenz (einschließlich Gleichstrom) innerhalb der Ansprechgeschwindigkeit der Fotozelle (etwa 1 kHz) erkennen.

IR-Detektoren sind digital - entweder sie erkennen ein 38 KHz IR-Signal und geben einen niedrigen Wert (0 V) aus oder sie erkennen kein Signal und geben einen hohen Wert (5 V) aus. Fotozellen verhalten sich wie Widerstände, der Widerstand ändert sich je nach Lichteinfall.

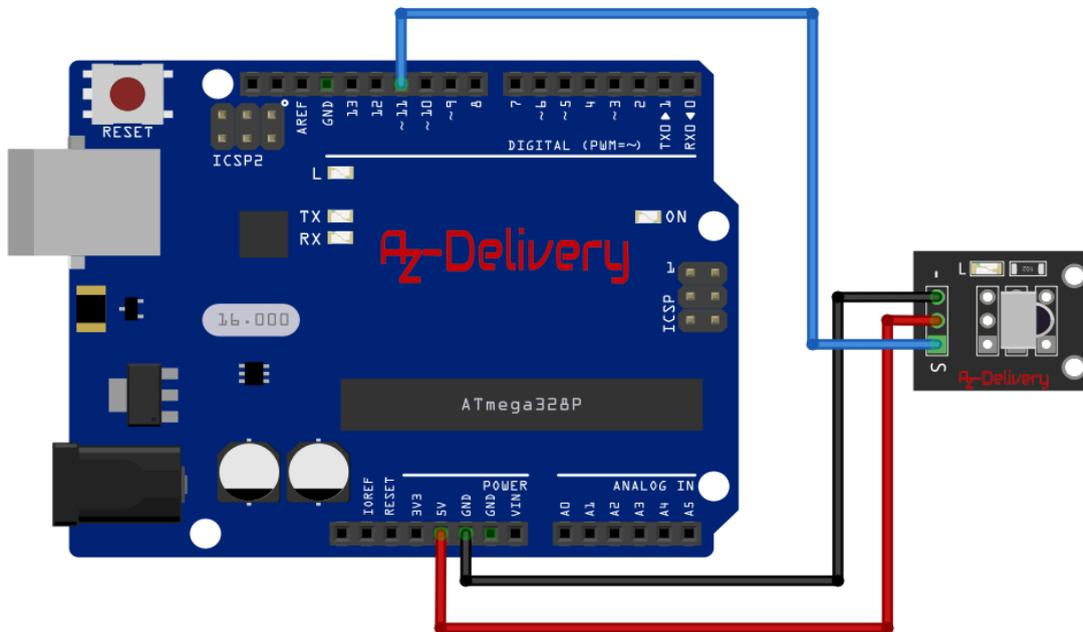
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Es gibt 3 Anschlüsse für den IR-Empfänger.

Die Anschlüsse sind: Signal, Spannung und Masse.

Das "-" ist der Erdungsstift, "S" ist der Signalstift und der mittlere Stift ist die Spannung 5V.

Beispielcode

Für dieses Beispiel müssen Sie die Bibliothek "IRremote" über den Bibliotheksmanager installieren. Nachdem die Bibliothek installiert ist, können Sie das Beispiel öffnen, indem Sie zu Datei > Beispiele > IRremote > IRrecvDemo

Laden Sie den Code in den Mikrocontroller und öffnen Sie den seriellen Monitor. Jetzt kannst du die Tasten auf deiner Fernbedienung drücken. Dieser Sketch zeigt den HEX-Code der gedrückten Taste an.

Servo-Motor

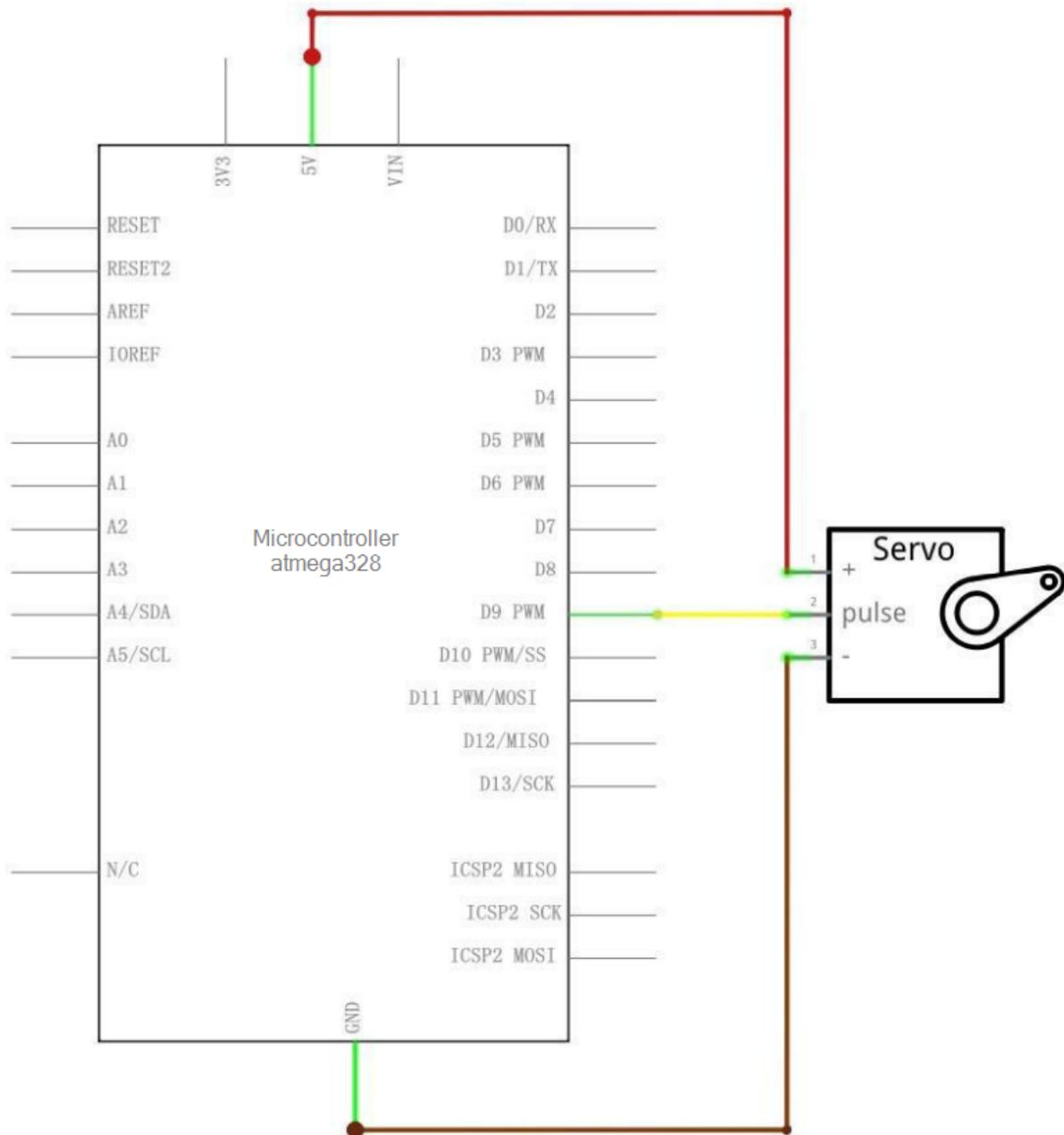
Ein Servomotor ist eine Art Getriebemotor, der sich nur um 180 Grad drehen kann. Er wird durch das Senden von elektrischen Impulsen von Ihrem Mikrocontroller-Board gesteuert. Diese Impulse sagen dem Servo, in welche Position er sich bewegen soll. Der Servo hat drei Drähte, von denen der braune der Erdungsdraht ist und mit dem GND-Anschluss verbunden werden sollte, der rote ist der Stromdraht und sollte mit dem 5-V-Anschluss verbunden werden, und der orange ist der Signaldraht und sollte mit dem D9-Anschluss verbunden werden.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- SG90-Servo
- 3 Überbrückungsdrähte M-M (Stecker auf Stecker)

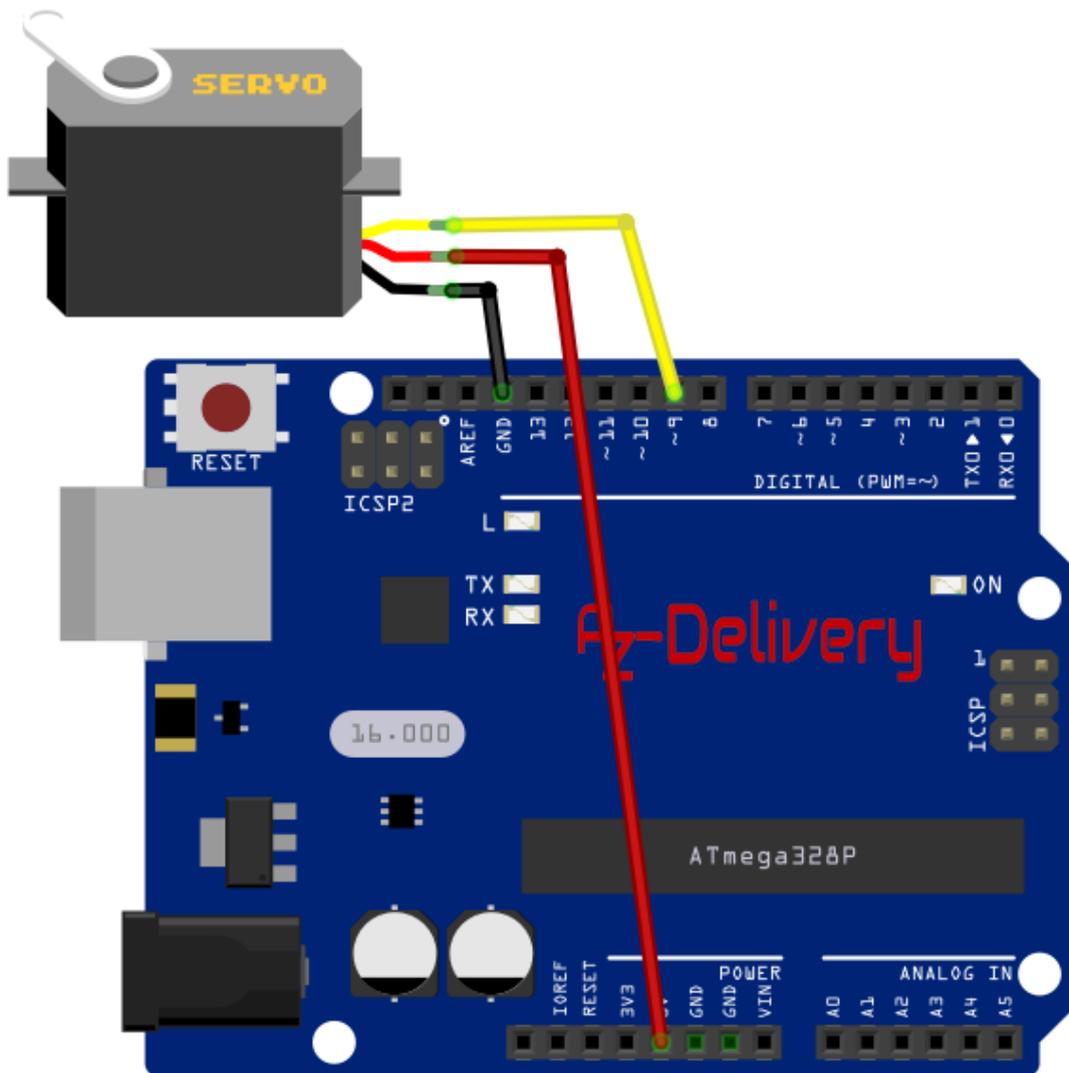
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Beispiel-Code

Zuerst müssen Sie die "Servo"-Bibliothek aus dem Bibliotheksmanager in der Arduino IDE installieren.

Um den Beispielsketch zu öffnen, gehe zu Datei > Beispiele > Servo > sweep

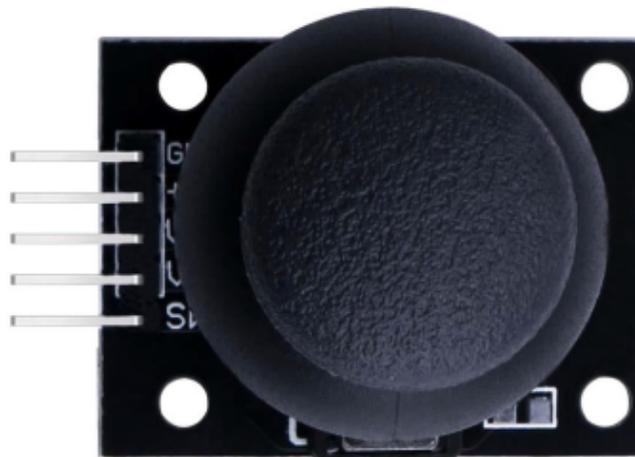
Laden Sie diesen Code auf Ihren Mikrocontroller hoch. Nach dem Hochladen sehen Sie, wie sich der Servo bewegt.

Joystick-Modul

Joysticks sind eine großartige Möglichkeit, die Steuerung in Ihren Projekten zu verbessern. In diesem Beispiel lernen Sie, wie Sie das analoge Joystick-Modul verwenden.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- Joystick-Modul
- 5 Stück F-M-Drähte (weibliche und männliche Dupont-Drähte)

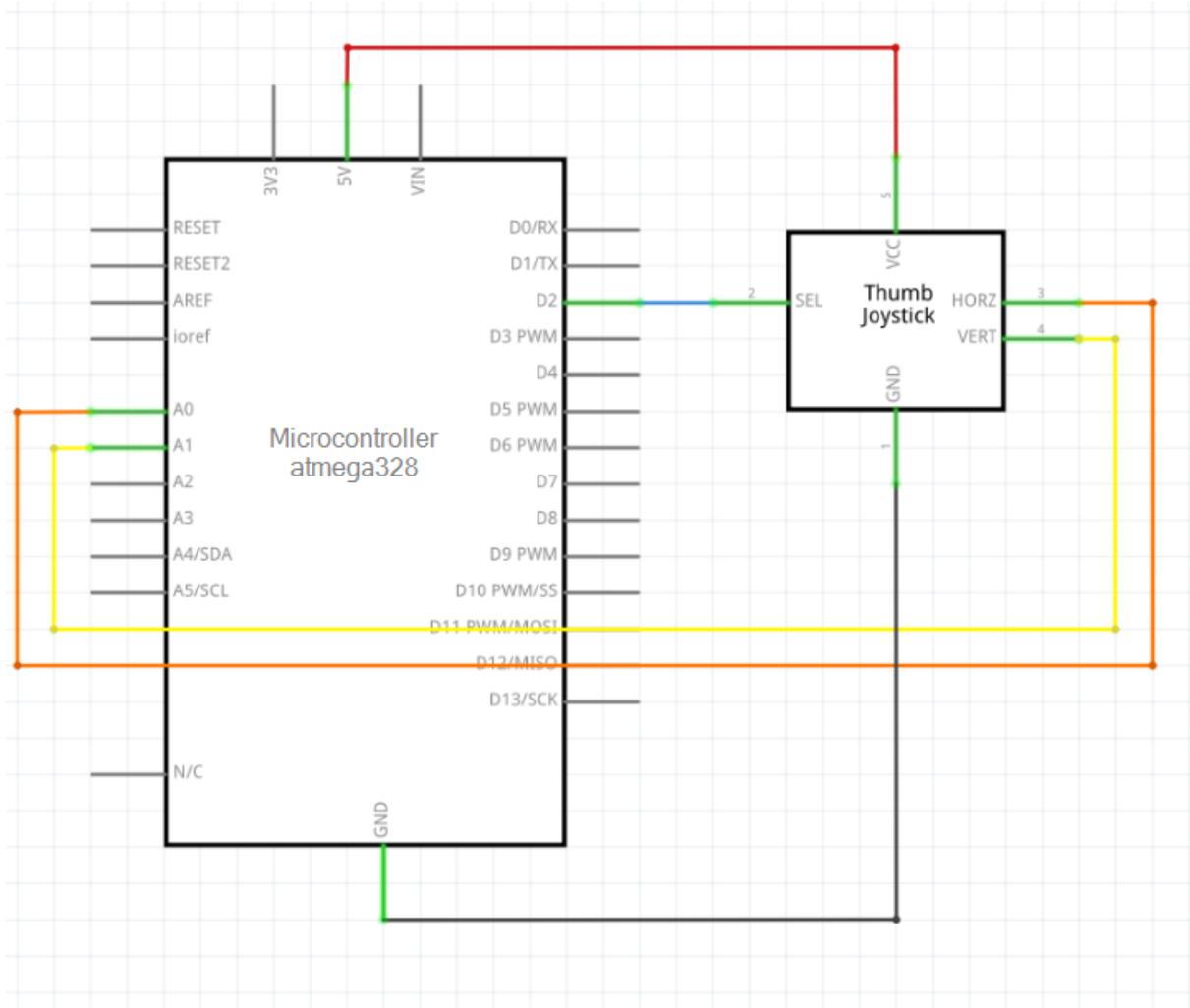


Das Modul hat 5 Stifte: Masse, VCC, X, Y, SW.

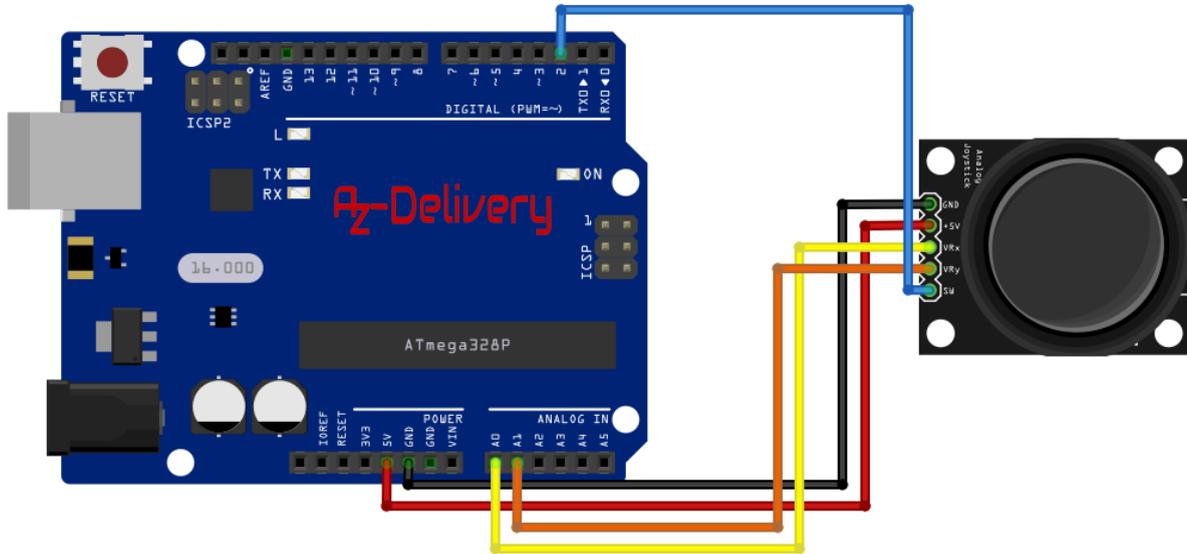
Der Daumenknüppel ist analog und sollte genauere Messwerte liefern als einfache "Richtungs"-Joysticks, die irgendeine Form von Tasten oder mechanischen Schaltern verwenden.

Außerdem können Sie den Joystick nach unten drücken, um einen Taster zu aktivieren. Wir müssen analoge Pins verwenden, um die Daten von den X/Y-Pins zu lesen, und einen digitalen Pin, um die Taste zu lesen. Der Key-Pin ist mit Masse verbunden, wenn der Joystick heruntergedrückt wird, und liegt ansonsten frei. Um stabile Messwerte vom Key/Select-Pin zu erhalten, muss er über einen Pull-up-Widerstand an VCC angeschlossen werden. Die eingebauten Widerstände an den digitalen Pins des Mikrocontrollers können verwendet werden.

Schematische Darstellung



Schaltplan



Super Starter Kit

Beispiel-Code

```
//Define pins
#define SW 2
#define X A0
#define Y A1

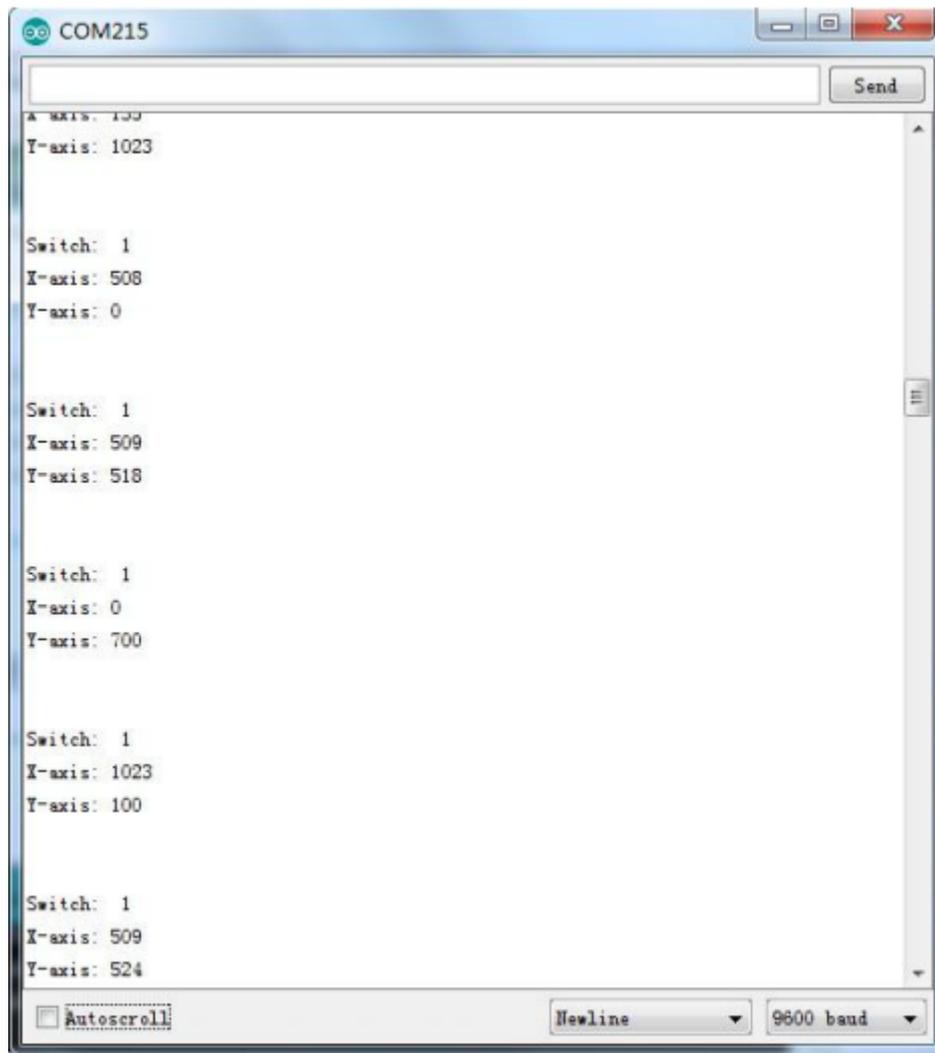
void setup() {
  pinMode(SW, INPUT_PULLUP);
  pinMode(X, INPUT);
  pinMode(Y, INPUT);
  Serial.begin(9600);
}

void loop() {
  Serial.print("Switch: ");
  Serial.print(digitalRead(SW));
  Serial.print("\n");
  Serial.print("X-axis: ");
  Serial.print(analogRead(X));
  Serial.print("\n");
  Serial.print("Y-axis: ");
  Serial.println(analogRead(Y));
  Serial.print("\n\n");
  delay(500);
}
```

Laden Sie diesen Code auf Ihren Mikrocontroller und öffnen Sie den seriellen Monitor.

Bewegen Sie den Joystick und beobachten Sie die Ausgabe auf dem seriellen Monitor.

Super Starter Kit



Analoge Joysticks sind im Grunde genommen Potentiometer, die analoge Werte liefern.

Der Wertebereich reicht von 0 bis 1024

Super Starter Kit

Modul zur Wasserstandserkennung

In diesem Beispiel lernen Sie, wie man ein Sensormodul zur Wasserstandserkennung einsetzt. Dieses Modul kann die Wassertiefe wahrnehmen, und die Kernkomponente ist eine Verstärkerschaltung, die aus einem Transistor und mehreren pektinierten Leiterplattenroutinen besteht.

Beim Eintauchen in das Wasser stellen diese Leiterbahnen einen Widerstand dar, der sich mit der Veränderung der Wassertiefe ändert. Dann wird das Signal der Wassertiefe in ein elektrisches Signal umgewandelt, und wir können die Änderung der Wassertiefe durch die ADC-Funktion Ihres Mikrocontroller-Boards erkennen.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- Wasserstandserkennungsmodul
- 3 St. F-M-Drähte (weibliche zu männlichen Dupont-Drähten)

Ein Wasser-Sensor-Brick ist für die Wassererkennung konzipiert, die weithin in der Erkennung der Niederschläge, Wasserstand, auch die Flüssigkeit Leckage verwendet werden kann. Der Brick besteht hauptsächlich aus drei Teilen: einem elektronischen Brick-Stecker, einem $1\text{ M}\Omega$ -Widerstand und mehreren Leitungen aus blanken, leitenden Drähten. Dieser Sensor arbeitet mit einer Reihe von freiliegenden Leiterbahnen, die mit der Erde verbunden sind.

Zwischen den geerdeten Leiterbahnen befinden sich die Messleitungen. Die Sensorspuren haben einen schwachen Pull-up-Widerstand von $1\text{ M}\Omega$. Der Widerstand zieht den Wert der Sensorspur hoch, bis ein Wassertropfen die Sensorspur mit der geerdeten Spur kurzschließt. Ob Sie es glauben oder nicht, diese Schaltung funktioniert mit den digitalen E/A-Pins Ihrer Mikrocontrollerplatine oder Sie können sie mit den analogen Pins verwenden, um die Menge des durch Wasser verursachten Kontakts zwischen der geerdeten und der Sensorspur zu erkennen.

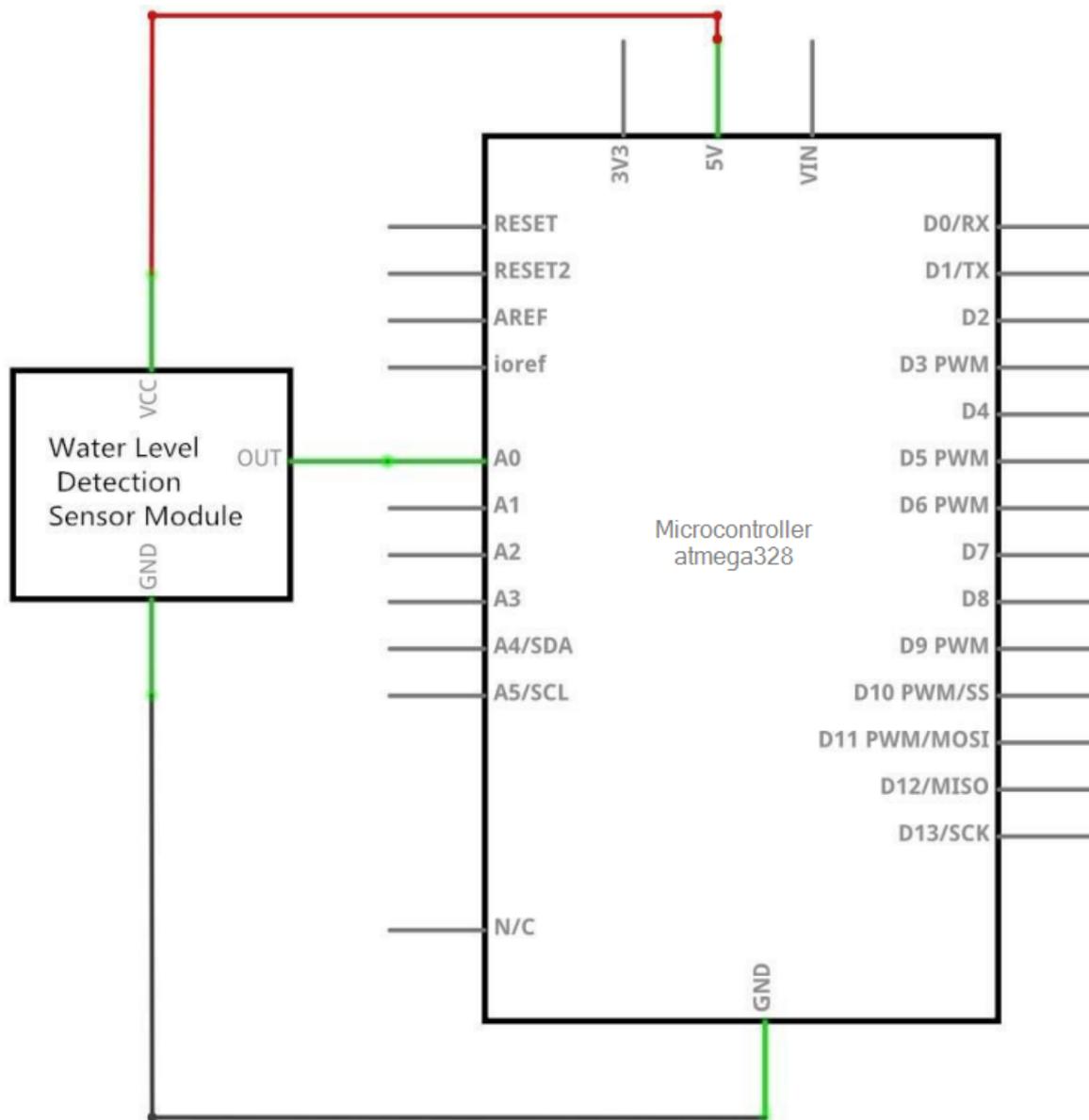
Super Starter Kit

Dieses Element kann den Wasserstand durch eine Reihe von freiliegenden parallelen Drähten beurteilen, um die Wassertropfen/Wassergröße zu messen.

Es kann leicht die Wassermenge in ein analoges Signal umwandeln, und der analoge Ausgangswert kann direkt in der Programmfunktion verwendet werden, um dann die Funktion eines Wasserstandsalarms zu erreichen.

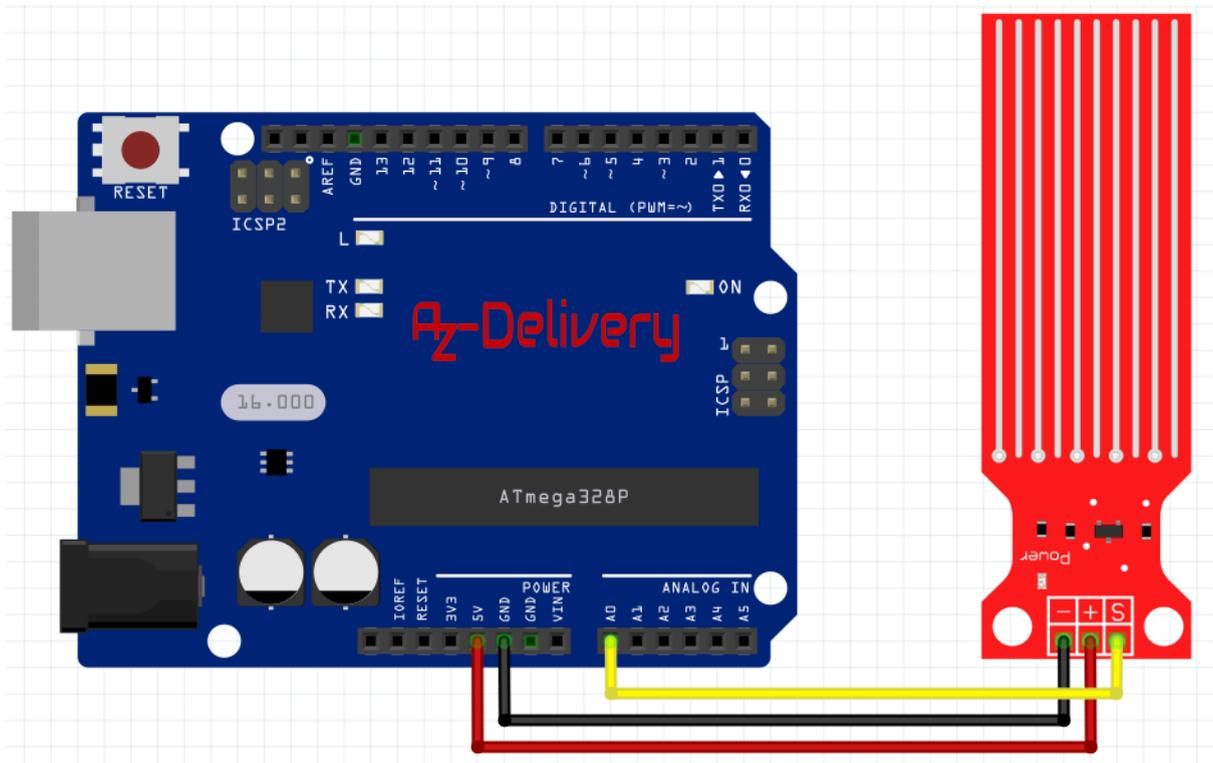
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Super Starter Kit

Beispiel-Code

```
#define in A0
void setup() {
  pinMode(in,INPUT);
  Serial.begin(9600);
}

void loop() {
  int val=analogRead(in);
  val=map(val, 0, 1024, 0, 100);
  Serial.print("Water Level: ");
  Serial.print(val);
  Serial.print("  % \n");
  delay(5000);
}
```

Laden Sie diesen Code auf Ihren Mikrocontroller und öffnen Sie den seriellen Monitor.

Dieser Code gibt alle 5 Sekunden den Wasserstand in % aus.

Ultraschall-Sensormodul

Ultraschallsensoren eignen sich hervorragend für alle Arten von Projekten, bei denen Entfernungsmessungen erforderlich sind, z. B. zur Vermeidung von Hindernissen. Der HC-SR04 ist kostengünstig und einfach zu verwenden, da wir eine speziell für diese Sensoren entwickelte Bibliothek verwenden.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- Ultraschallsensormodul
- 4 Stück F-M-Drähte (Dupont-Drähte von Buchse zu Stecker)

Das Ultraschallsensormodul HC-SR04 bietet eine berührungslose Messfunktion von 2 cm bis 400 cm, die Messgenauigkeit kann bis zu 3 mm betragen. Das Modul besteht aus einem Ultraschallsender, einem Empfänger und einem Steuerkreis. Das Grundprinzip der Arbeit:

(1) Mit IO-Trigger für mindestens 10µs High-Level-Signal,

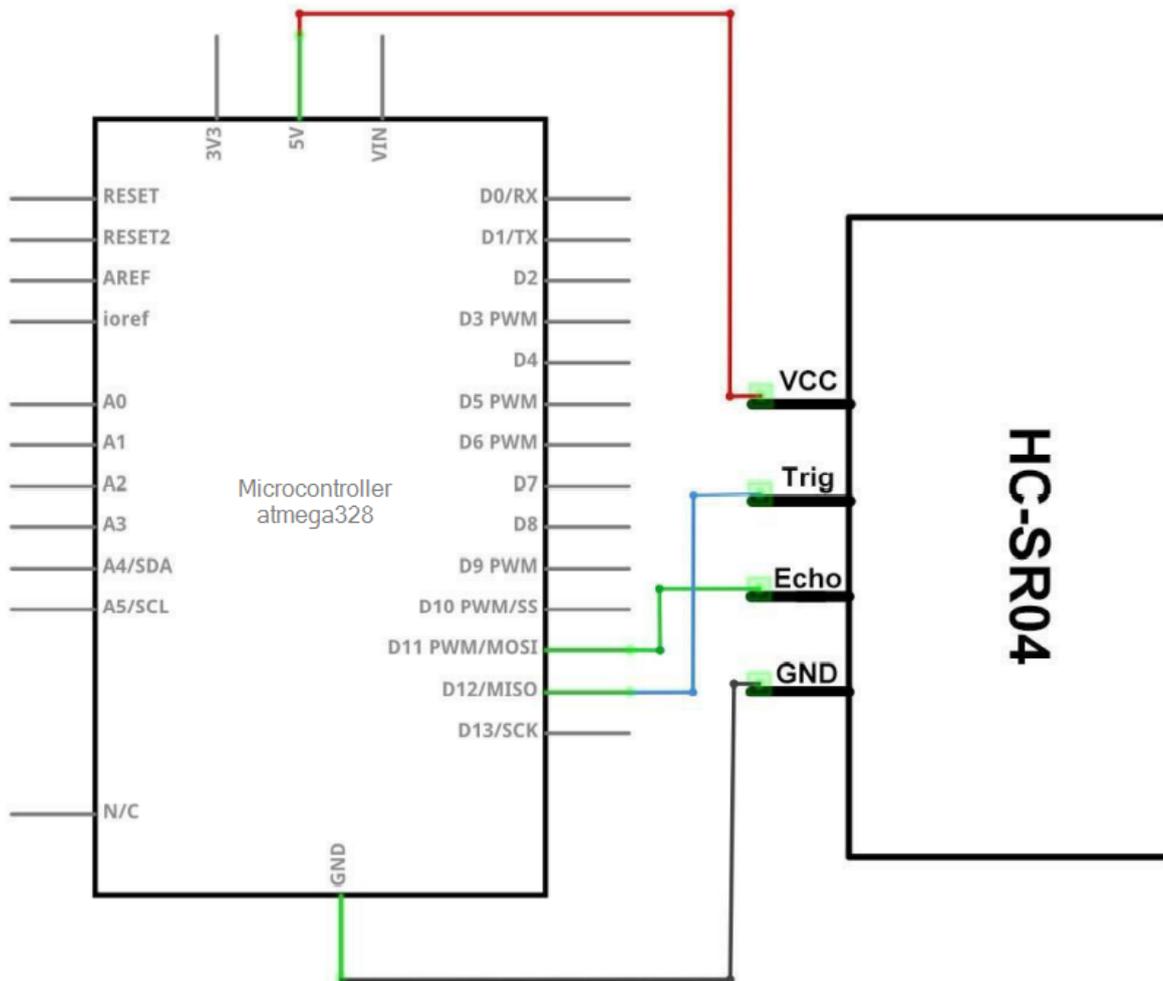
(2) Das Modul sendet automatisch acht 40 kHz und erkennt, ob es ein Puls-Signal zurück.

(3) Wenn das Signal zurück ist, durch High-Pegel, Zeit der hohen Ausgang IO Dauer ist die Zeit vom Senden von Ultraschall tore Drehen. Testabstand = (High-Level-Zeit × Schallgeschwindigkeit (340m/s) / 2 Das Zeitdiagramm ist unten dargestellt. Sie brauchen nur einen kurzen 10-µs-Impuls an den Triggereingang zu geben, um die Entfernungsmessung zu starten, und dann sendet das Modul einen 8-Zyklen-Ultraschallimpuls mit 40 kHz aus und erzeugt sein Echo. Das Echo ist ein Entfernungsobjekt, bei dem Impulsbreite und Reichweite im Verhältnis zueinander stehen.

Sie können die Reichweite anhand des Zeitintervalls zwischen dem Senden des Triggersignals und dem Empfang des Echosignals berechnen. Formel: $\mu\text{s} / 58 = \text{Zentimeter}$ oder $\mu\text{s} / 148 = \text{Zoll}$; oder: die Reichweite = High-Level-Zeit * Geschwindigkeit (340M/S) / 2; wir empfehlen, über 60ms Messzyklus zu verwenden, um Trigger-Signal, um das Echo-Signal zu verhindern.

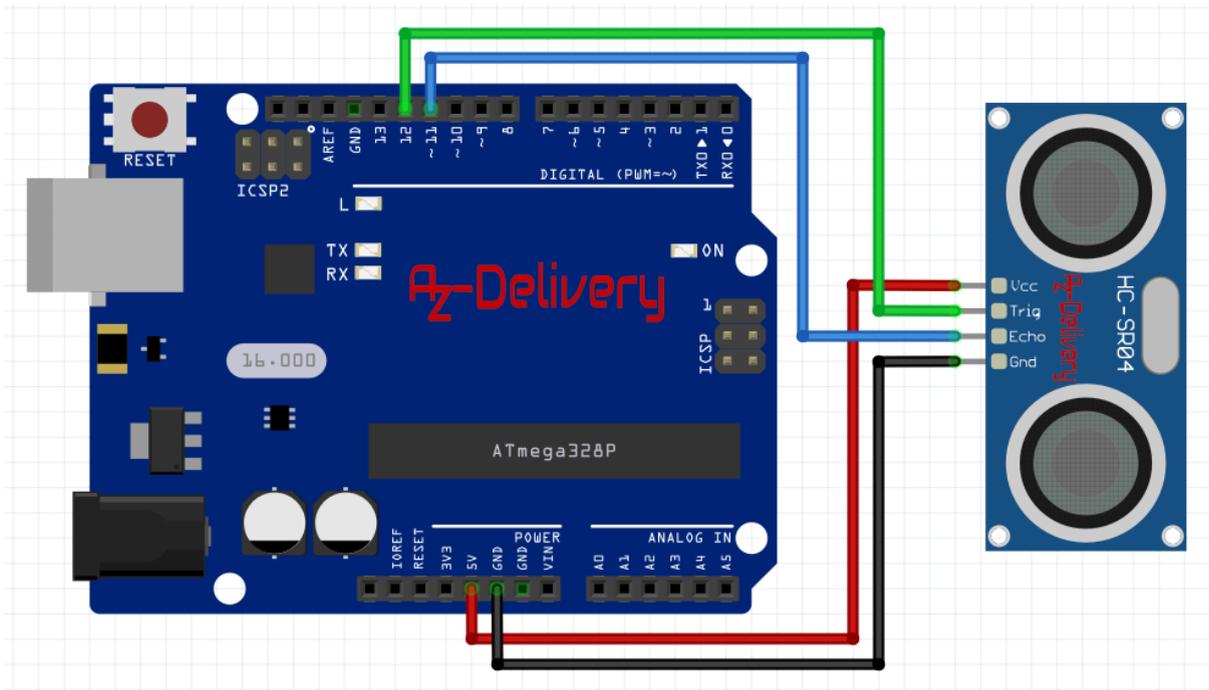
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Super Starter Kit

Beispiel-Code

Zuerst müssen Sie die "HCSR04"-Bibliothek von Martin Sosic über den Bibliotheksmanager installieren.

```
#include <HCSR04.h>
// sensor(triggerPin, echoPin);
UltraSonicDistanceSensor distanceSensor(12, 11);

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

void loop () {
    Serial.println(distanceSensor.measureDistanceCm());
    delay(500);
}
```

Laden Sie diesen Code in Ihren Mikrocontroller und öffnen Sie den Serial Monitor.

Sie erhalten nun die Entfernung in cm auf dem seriellen Monitor.

Super Starter Kit

Tonsensormodul

In diesem Beispiel lernen Sie, wie Sie ein Schallsensormodul verwenden. Dieses Modul hat zwei Ausgänge:

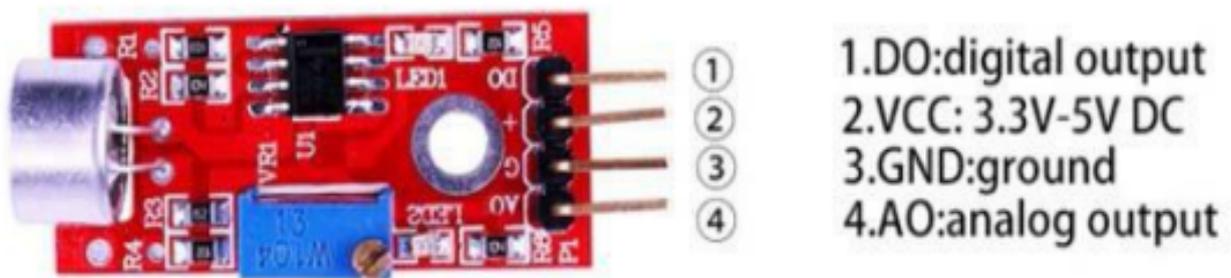
AO: Analogausgang, Echtzeit-Ausgangsspannungssignal des Mikrofons

DO: Wenn die Intensität des Schalls einen bestimmten Schwellenwert erreicht, wird ein Signal mit hohem oder niedrigem Pegel ausgegeben.

Die Schwellenempfindlichkeit kann mit dem Potentiometer eingestellt werden. Um sicherzustellen, dass das Mikrofon Geräusche normal erkennen kann, versuchen Sie bitte, die Empfindlichkeit durch Drehen des blauen Präzisionspotentiometers auf dem Modul zu ändern. Da das Potentiometer sehr genau eingestellt werden kann, dauert es mindestens 10 Umdrehungen, bis Sie eine Antwort erhalten.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- Schallsensormodul
- 4 Stück F-M-Drähte (weibliche und männliche Dupont-Drähte)

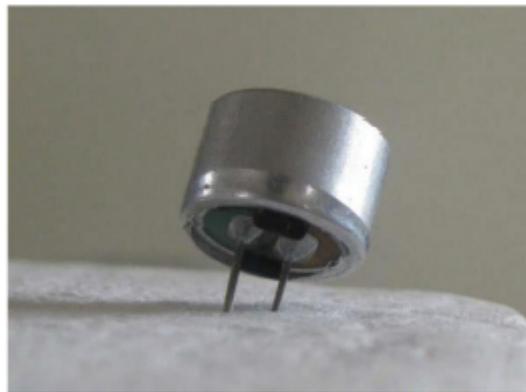


Wandler sind Geräte, die Energie von einer Form in eine andere umwandeln. Ein Mikrofon ist ein Wandler, der Schallenergie in elektrische Signale umwandelt. Es funktioniert im Gegensatz zu einem Lautsprecher. Mikrofone gibt es in verschiedenen Formen und Größen. Je nach Anwendung kann ein Mikrofon verschiedene Technologien zur Umwandlung von Schall in elektrische Signale verwenden.

Super Starter Kit

Hier werden wir das Elektret-Kondensatormikrofon besprechen, das in Mobiltelefonen, Laptops usw. weit verbreitet ist. Wie der Name schon sagt, ist das Elektret-Kondensatormikrofon ein Parallelplattenkondensator und arbeitet nach dem Prinzip der variablen Kapazität.

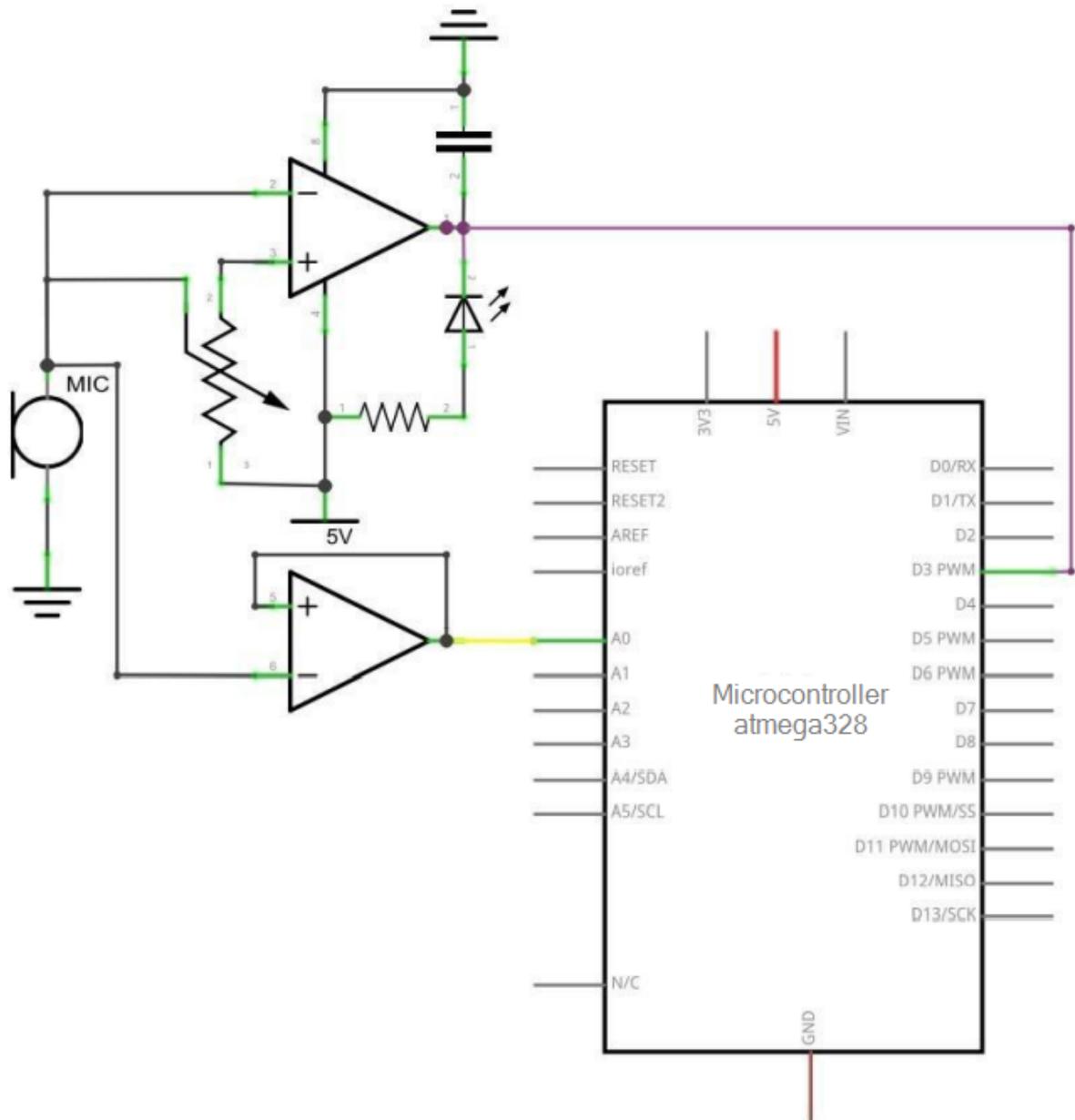
Er besteht aus zwei Platten, von denen eine fest (die so genannte Rückplatte) und die andere beweglich (die so genannte Membran) ist und zwischen denen ein kleiner Spalt besteht. Ein elektrisches Potential lädt die Platte auf. Wenn Schall auf die Membran trifft, beginnt sie sich zu bewegen, wodurch sich die Kapazität zwischen den Platten ändert, was wiederum dazu führt, dass ein variabler elektrischer Strom fließt.



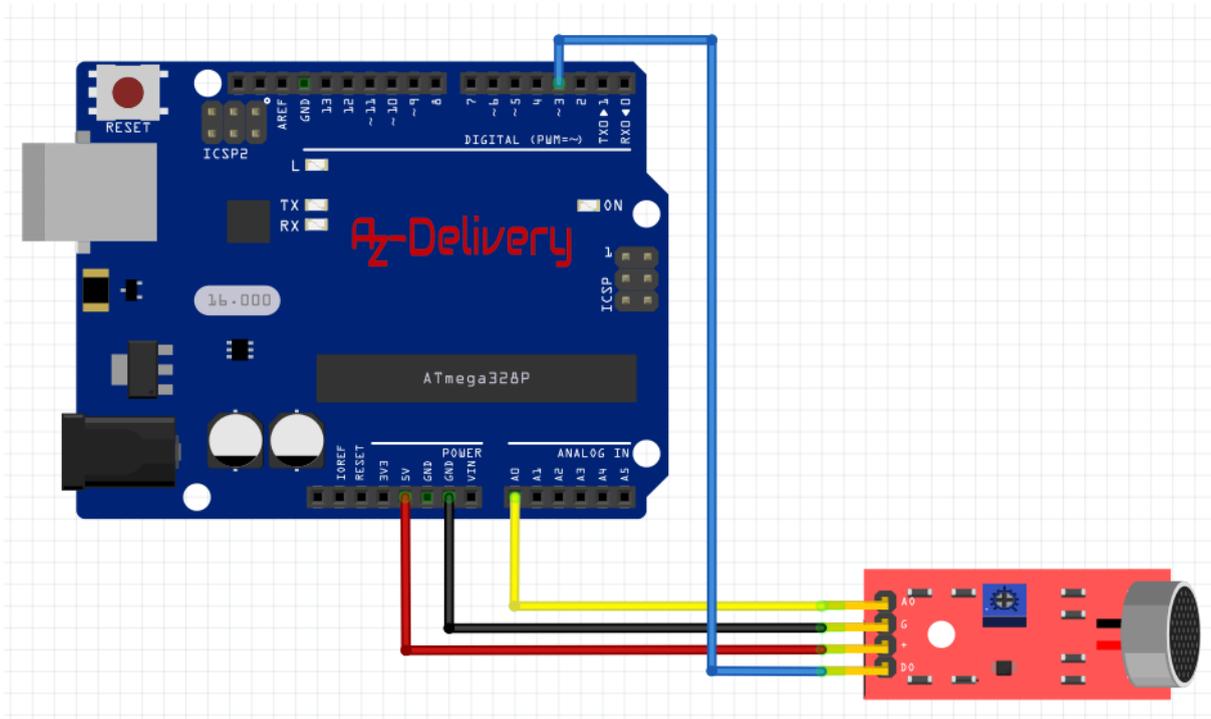
Diese Mikrofone werden häufig in elektronischen Schaltkreisen verwendet, um kleine Geräusche oder Luftschwingungen zu erfassen, die wiederum in elektrische Signale zur weiteren Verwendung umgewandelt werden. Die beiden Beine, wie in der Abbildung oben gezeigt, werden verwendet, um elektrische Verbindungen mit dem Schaltkreis herzustellen.

Super Starter Kit

Schematische Darstellung



Schaltplan



Super Starter Kit

Beispiel-Code

```
void setup() {  
  pinMode(3, INPUT);  
  pinMode(A0, INPUT);  
  pinMode(LED_BUILTIN, OUTPUT);  
  Serial.begin(9600);  
}  
  
void loop() {  
  if(digitalRead(3)==HIGH){  
    digitalWrite(LED_BUILTIN, HIGH);  
  }else digitalWrite(LED_BUILTIN, LOW);  
  
  Serial.println(analogRead(A0));  
  delay(50);  
}
```

Laden Sie diesen Code auf Ihren Mikrocontroller und öffnen Sie den seriellen Plotter, indem Sie auf Extras > Serieller Plotter gehen.

Sie sehen nun das analoge Signal. Wenn der digitale Pin HIGH ist, leuchtet die OnBoard-LED auf.

Membranschalter-Modul

In diesem Beispiel wird gezeigt, wie man eine Tastatur mit einem Mikrocontroller-Board verbindet, so dass der Mikrocontroller die vom Benutzer gedrückten Tasten lesen kann.

Für dieses Beispiel benötigen Sie:

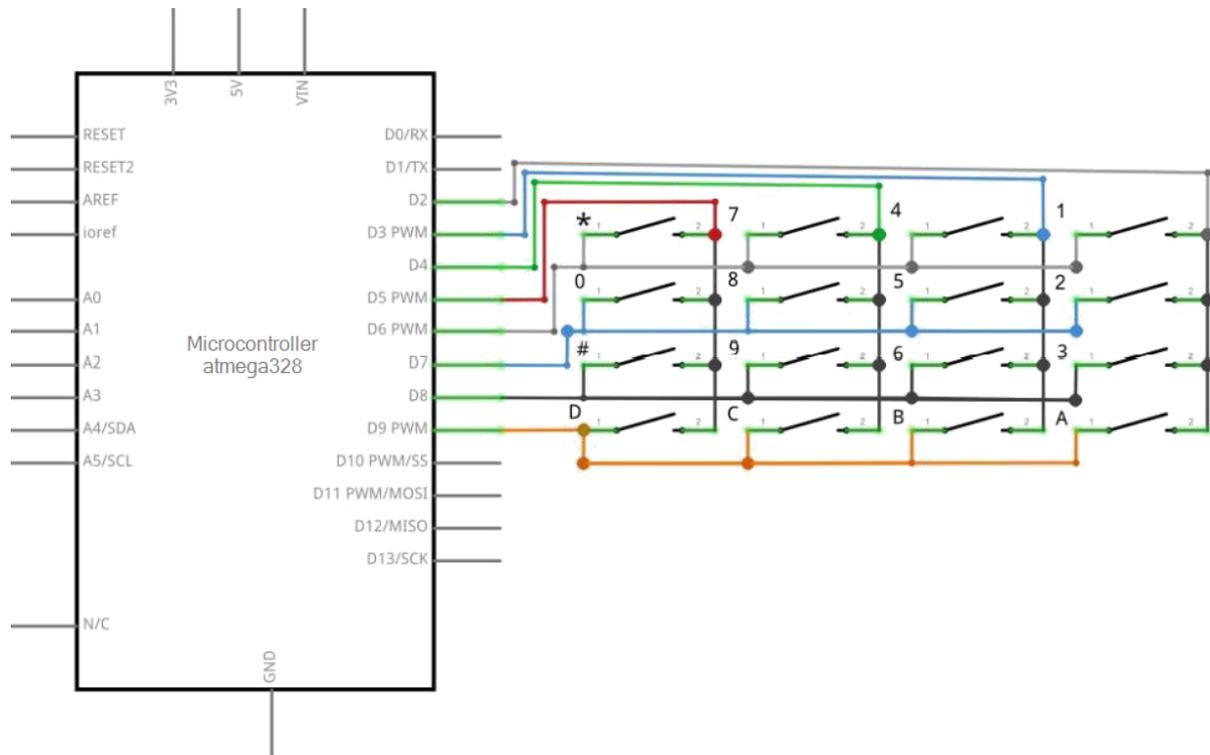
- Mikrocontroller-Platine
- Membranschaltermodul
- 8 Stück M-M-Drähte (männliche zu männliche Überbrückungsdrähte)

Tastaturen werden in allen Arten von Geräten verwendet, z. B. in Handys, Faxgeräten, Mikrowellen, Öfen, Türschlössern usw. Sie sind praktisch überall zu finden. Unzählige elektronische Geräte verwenden sie für Benutzereingaben. Zu wissen, wie man eine Tastatur an eine Mikrocontrollerplatine anschließt, ist also sehr wertvoll für den Bau vieler verschiedener Arten von kommerziellen Produkten. Am Ende, wenn alles richtig angeschlossen und programmiert ist, wird das Drücken einer Taste auf dem seriellen Monitor Ihres Computers angezeigt. In diesem Beispiel werden wir eine Matrix-Tastatur verwenden.

Dabei handelt es sich um eine Tastatur, die einem Kodierungsschema folgt, das es ermöglicht, viel weniger Ausgangspins zu haben als Tasten vorhanden sind. Die von uns verwendete Matrixtastatur hat zum Beispiel 16 Tasten (0-9, A-D, *, #), aber nur 8 Ausgangsstifte. Bei einer linearen Tastatur wären 17 Ausgangsstifte (einer für jede Taste und ein Erdungsstift) erforderlich, um zu funktionieren. Das Matrix-Codierungsschema ermöglicht weniger Ausgangsstifte und damit viel weniger Verbindungen, die für die Funktion der Tastatur hergestellt werden müssen. Auf diese Weise sind sie effizienter als lineare Tastaturen, da sie weniger Verdrahtung haben.

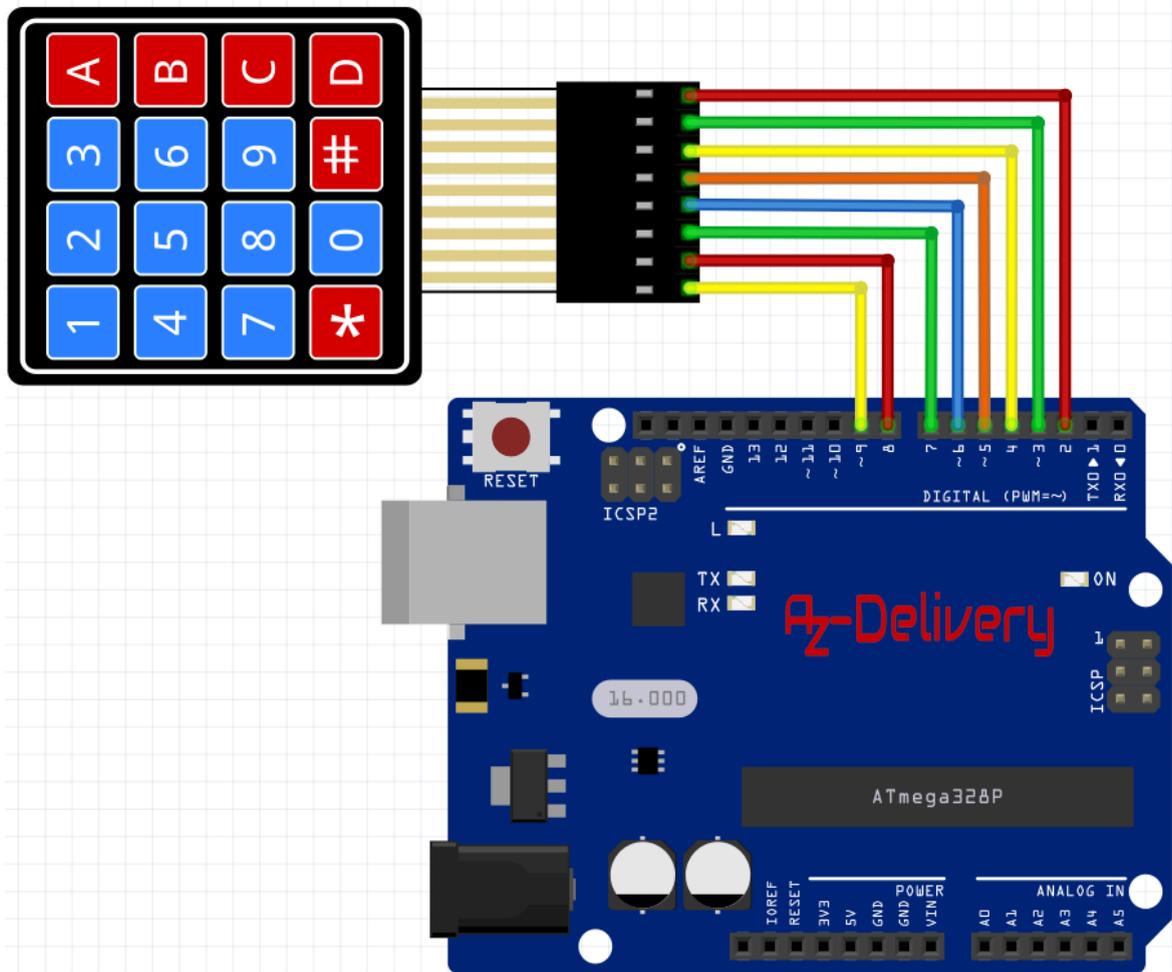
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



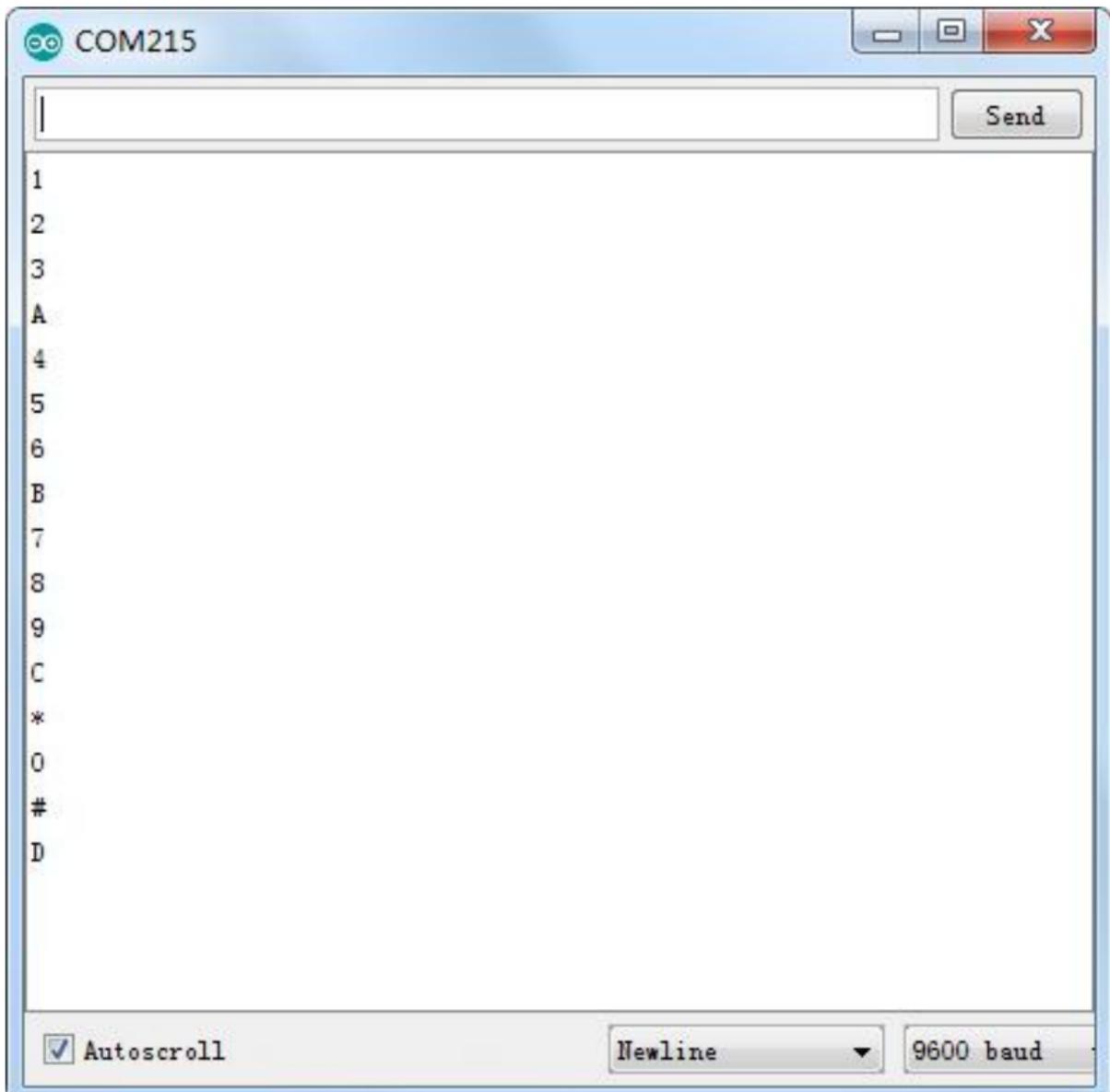
Beispiel-Code

```
/* @file CustomKeypad.pde
|| @version 1.0
|| @author Alexander Brevig
|| @contact alexanderbrevig@gmail.com
||
|| @description
|| | Demonstrates changing the keypad size and key values.
|| #
*/
#include <Keypad.h>
const byte ROWS = 4; //four rows
const byte COLS = 4; //four columns
//define the cymbols on the buttons of the keypads
char hexaKeys[ROWS][COLS] = {
  {'1','2','3','A'},
  {'4','5','6','B'},
  {'7','8','9','C'},
  {'*','0','#','D'}
};
byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of
the keypad
byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts
of the keypad
//initialize an instance of class NewKeypad
Keypad customKeypad = Keypad(makeKeymap(hexaKeys), rowPins,
colPins, ROWS, COLS);
void setup(){
  Serial.begin(9600);
}
void loop(){
  char customKey = customKeypad.getKey();
  if (customKey){
    Serial.println(customKey);
  }
}
```

Super Starter Kit

Laden Sie diesen Code in Ihren Mikrocontroller hoch und öffnen Sie den Serial Monitor.

Drücken Sie nun die Tasten auf der Tastatur. Sie werden im Serial Monitor angezeigt.



8 LEDs mit 74HC595

In diesem Beispiel lernen Sie, wie Sie acht LEDs mit einem Mikrocontroller verwenden können, ohne auf 8 Ausgangspins verzichten zu müssen!

Obwohl Sie acht LEDs mit je einem Widerstand an einen Mikrocontroller-Pin anschließen könnten, würden Ihnen schnell die Pins Ihres Mikrocontrollers ausgehen. Wenn du nicht viele Dinge an deinen Mikrocontroller angeschlossen hast, ist das in Ordnung - aber oft wollen wir Knöpfe, Sensoren, Servos usw. und bevor du es merkst, hast du keine Pins mehr übrig.

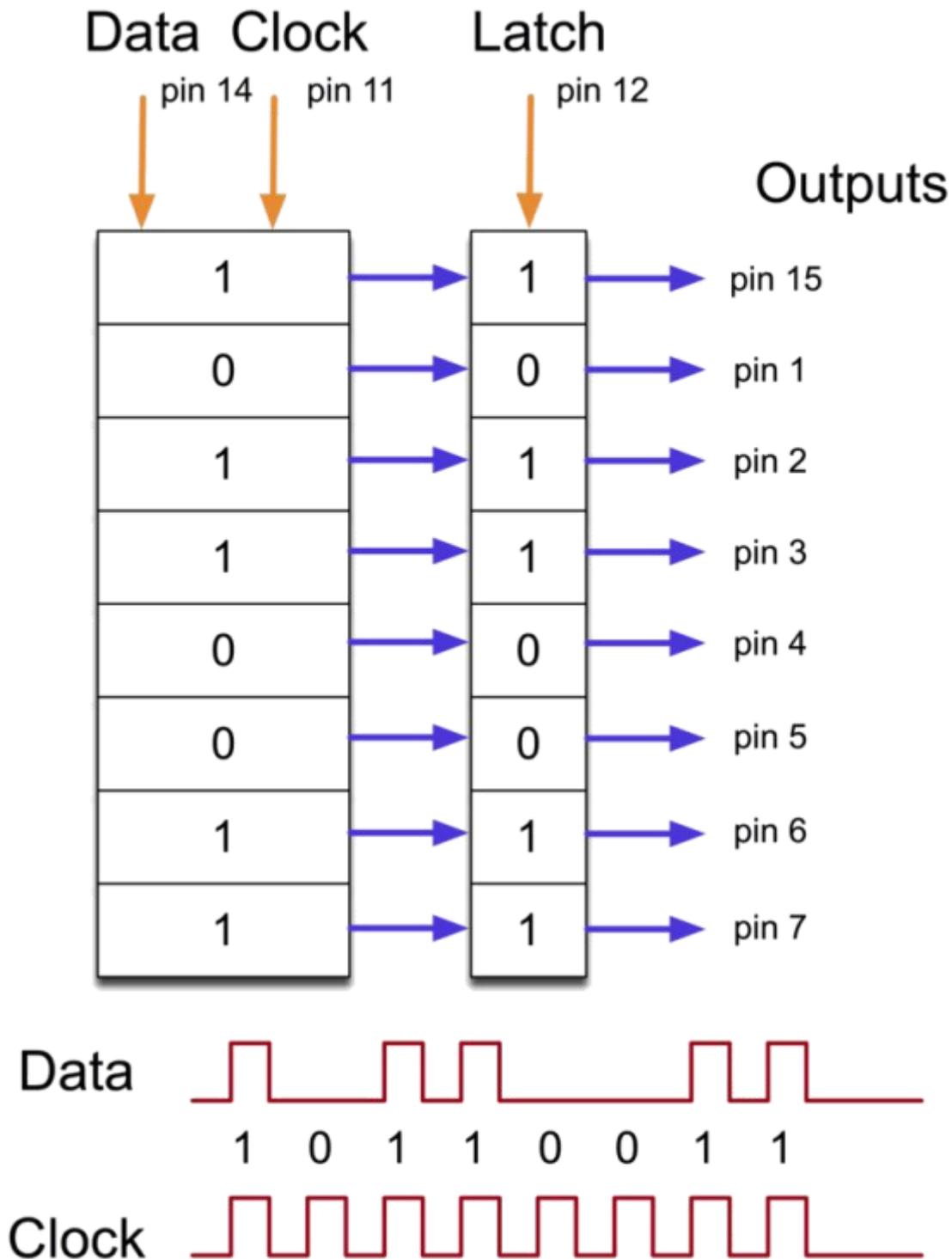
Stattdessen werden Sie einen Chip namens 74HC595 Serial to Parallel Converter verwenden. Dieser Chip hat acht Ausgänge und drei Eingänge, über die man Daten bitweise einspeisen kann. Dieser Chip macht die Ansteuerung der LEDs etwas langsamer (man kann die LEDs nur etwa 500.000 Mal pro Sekunde statt 8.000.000 Mal pro Sekunde umschalten), aber es ist immer noch sehr schnell, viel schneller als der Mensch es erkennen kann, also ist es das wert.

Für dieses Beispiel benötigst du:

- Mikrocontroller-Platine
- 830-poliges Breadboard
- 4 Stk. rote LED
- 4 Stück blaue LED
- 8 Stück 220-Ohm-Widerstände
- 74HC595 IC
- 14 St. M-M-Drähte (männliche zu männliche Überbrückungsdrähte)

Das Schieberegister ist eine Art Chip, der so etwas wie acht Speicherplätze enthält, von denen jeder entweder eine 1 oder eine 0 sein kann. Um jeden dieser Werte ein- oder auszuschalten, werden die Daten über die Pins "Data" und "Clock" des Chips eingespeist.

Super Starter Kit



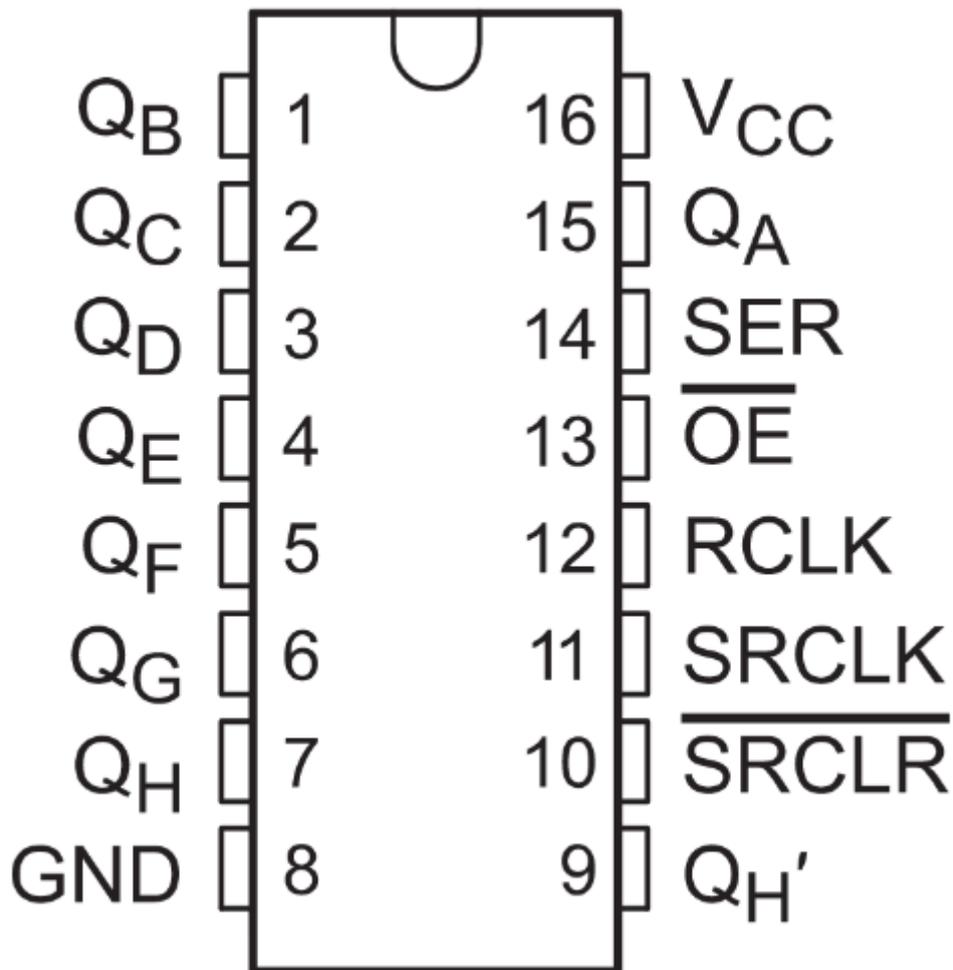
Der Clock-Pin muss acht Pulse empfangen. Bei jedem Impuls wird eine 1 in das Schieberegister geschoben, wenn der Daten-Pin high ist, andernfalls eine 0.

Wenn alle acht Impulse empfangen wurden, werden durch Aktivieren des "Latch"-Pins diese acht Werte in das Latch-Register kopiert.

Super Starter Kit

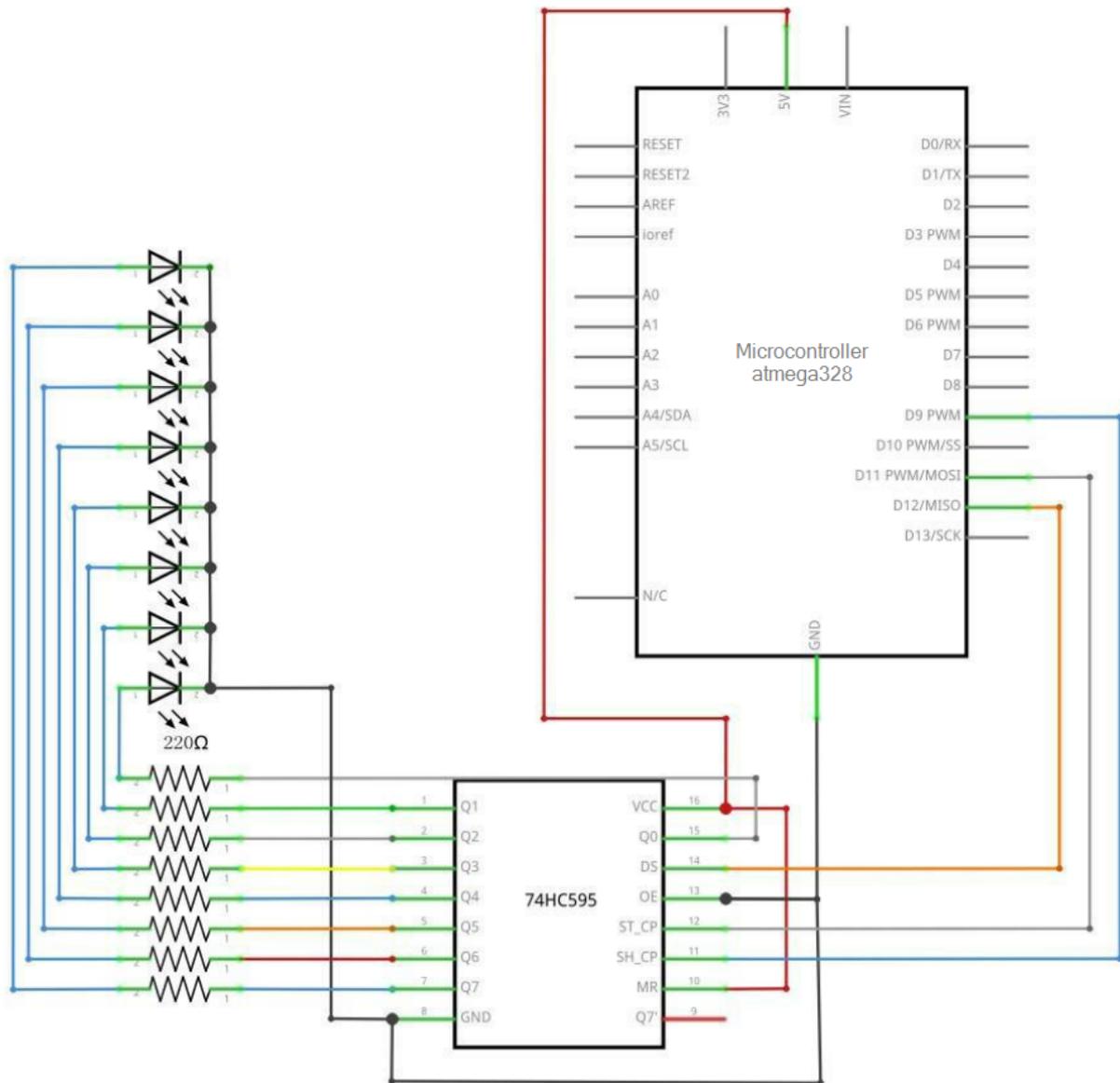
Dies ist notwendig, da sonst die falschen LEDs flackern würden, während die Daten in das Schieberegister geladen werden. Der Chip verfügt auch über einen Output-Enable-Pin (OE), mit dem die Ausgänge auf einmal aktiviert oder deaktiviert werden können. Sie könnten diesen Pin mit einem PWM-fähigen Pin verbinden und 'analogWrite' verwenden, um die Helligkeit der LEDs zu steuern. Dieser Pin ist active low, also verbinden wir ihn mit GND.

Dies ist die Pinbelegung:



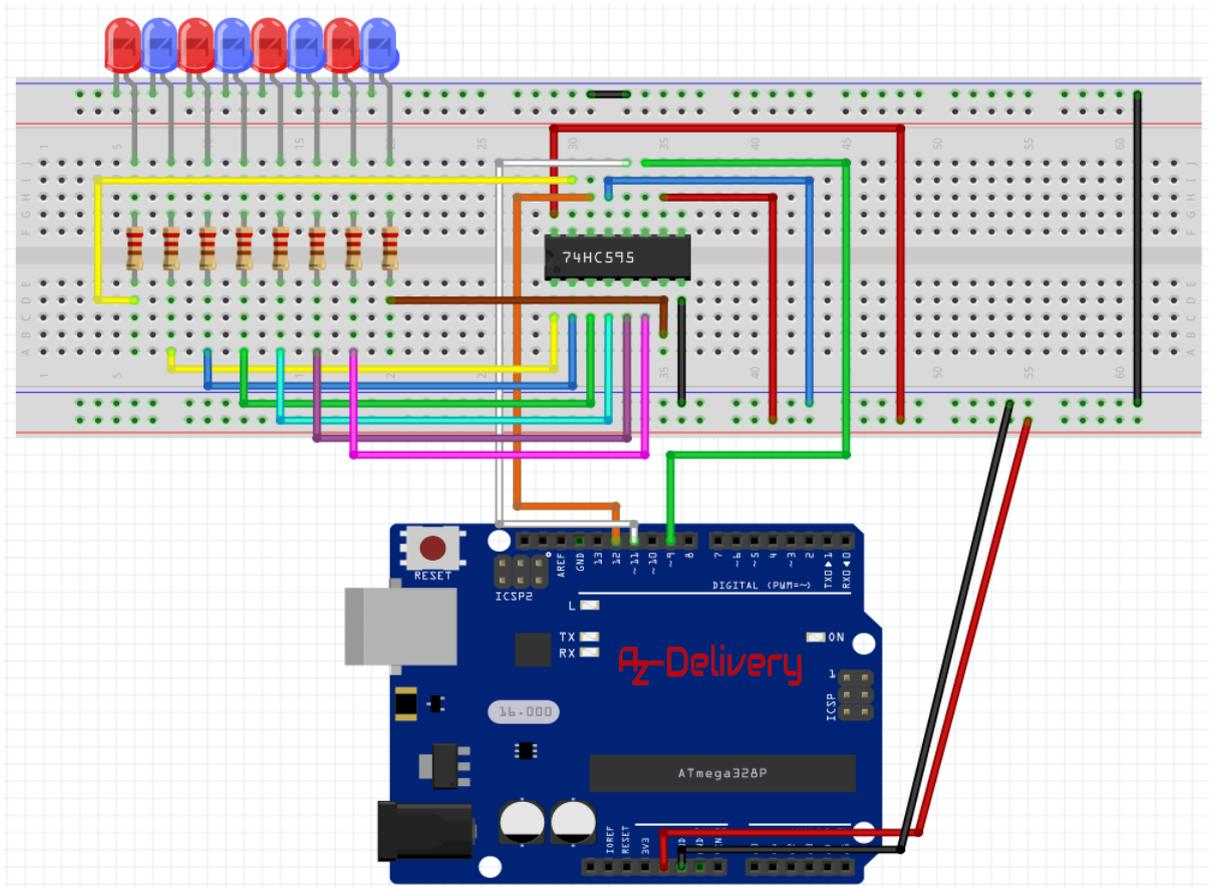
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Super Starter Kit

Beispiel-Code

```
int tDelay = 100;
int latchPin = 11;    // (11) ST_CP [RCK] on 74HC595
int clockPin = 9;    // (9) SH_CP [SCK] on 74HC595
int dataPin = 12;    // (12) DS [S1] on 74HC595

byte leds = 0;

void updateShiftRegister()
{
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, leds);
    digitalWrite(latchPin, HIGH);
}

void setup()
{
    pinMode(latchPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
}

void loop()
{
    leds = 0;
    updateShiftRegister();
    delay(tDelay);
    for (int i = 0; i < 8; i++)
    {
        bitSet(leds, i);
        updateShiftRegister();
        delay(tDelay);
    }
}
```

Laden Sie diesen Code auf Ihren Mikrocontroller. Sie sollten nun die LEDs blinken sehen. Als erstes definieren wir die drei Pins, die wir verwenden

Super Starter Kit

werden. Dies sind die digitalen Ausgänge des Mikrocontrollers, die mit den Latch-, Clock- und Daten-Pins des 74HC595 verbunden werden.

```
int latchPin = 11;  
int clockPin = 9;  
int dataPin = 12;
```

Als nächstes wird eine Variable namens "leds" definiert. Diese wird verwendet, um das Muster der LEDs zu speichern, die gerade ein- oder ausgeschaltet sind. Daten vom Typ "Byte" stellen Zahlen mit acht Bits dar. Jedes Bit kann entweder ein- oder ausgeschaltet sein, so dass diese Variable perfekt geeignet ist, um festzuhalten, welche unserer acht LEDs ein- oder ausgeschaltet sind.

Mit der Funktion "setup" werden lediglich die drei von uns verwendeten Pins als digitale Ausgänge festgelegt

Die Funktion "loop" schaltet zunächst alle LEDs aus, indem sie der Variablen "leds" den Wert 0 gibt. Dann ruft sie 'updateShiftRegister' auf, das das 'leds'-Muster an das Schieberegister sendet, so dass alle LEDs ausgeschaltet werden. Wie 'updateShiftRegister' funktioniert, werden wir später behandeln. Die Schleifenfunktion hält eine halbe Sekunde lang an und beginnt dann, mit Hilfe der 'for'-Schleife und der Variablen 'i' von 0 bis 7 zu zählen. Jedes Mal wird die Arduino-Funktion 'bitSet' verwendet, um das Bit zu setzen, das die LED in der Variablen 'leds' steuert. Anschließend wird die Funktion "updateShiftRegister" aufgerufen, damit die LEDs aktualisiert werden, um den Wert in der Variablen "leds" wiederzugeben. Es gibt dann eine halbe Sekunde Verzögerung, bevor 'i' inkrementiert wird und die nächste LED aufleuchtet.

Die Funktion 'updateShiftRegister' setzt zunächst den latchPin auf low und ruft dann die Funktion 'shiftOut' auf, bevor sie den 'latchPin' wieder auf high setzt. Diese Funktion benötigt vier Parameter, die ersten beiden sind die Pins, die für Data bzw. Clock verwendet werden sollen. Der dritte Parameter gibt an, an welchem Ende der Daten Sie beginnen wollen. Wir beginnen mit dem rechten Bit, das als 'Least Significant Bit' (LSB) bezeichnet wird. Der letzte Parameter sind die eigentlichen Daten, die in das Schieberegister geschoben werden sollen, in diesem Fall also "leds".

Der serielle Monitor

In diesem Beispiel bauen Sie auf "8 LEDs mit 74HC595" auf und fügen die Möglichkeit hinzu, die LEDs von Ihrem Computer aus mit dem Arduino Serial Monitor zu steuern. Der serielle Monitor ist das "Bindeglied" zwischen dem Computer und dem Mikrocontroller. Er ermöglicht das Senden und Empfangen von Textnachrichten, was für die Fehlersuche und die Steuerung des Mikrocontrollers über eine Tastatur sehr praktisch ist!

Zum Beispiel können Sie von Ihrem Computer aus Befehle senden, um LEDs einzuschalten. In dieser Lektion werden Sie genau die gleichen Teile und ein ähnliches Breadboard-Layout wie im vorherigen Beispiel verwenden. Wenn Sie es also noch nicht getan haben, folgen Sie jetzt dem Beispiel "8 LEDs mit 74HC595".

Beispiel-Code

```
int latchPin = 11;
int clockPin = 9;
int dataPin = 12;
byte leds = 0;
void updateShiftRegister()
{
    digitalWrite(latchPin, LOW);
    shiftOut(dataPin, clockPin, LSBFIRST, leds);
    digitalWrite(latchPin, HIGH);
}
void setup()
{
    pinMode(latchPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
    updateShiftRegister();
    Serial.begin(9600);
    while (!Serial); // Wait until Serial is ready - Leonardo
    Serial.println("Enter LED Number 0 to 7 or 'x' to clear");
```

Super Starter Kit

```
}  
void loop()  
{  
  if (Serial.available())  
  {  
    char ch = Serial.read();  
    if (ch >= '0' && ch <= '7')  
    {  
      int led = ch - '0';  
      bitSet(leds, led);  
      updateShiftRegister();  
      Serial.print("Turned on LED ");  
      Serial.println(led);  
    }  
    if (ch == 'x')  
    {  
      leds = 0;  
      updateShiftRegister();  
      Serial.println("Cleared");  
    }  
  }  
}
```

Laden Sie diesen Code in Ihren Mikrocontroller hoch und öffnen Sie den seriellen Monitor.

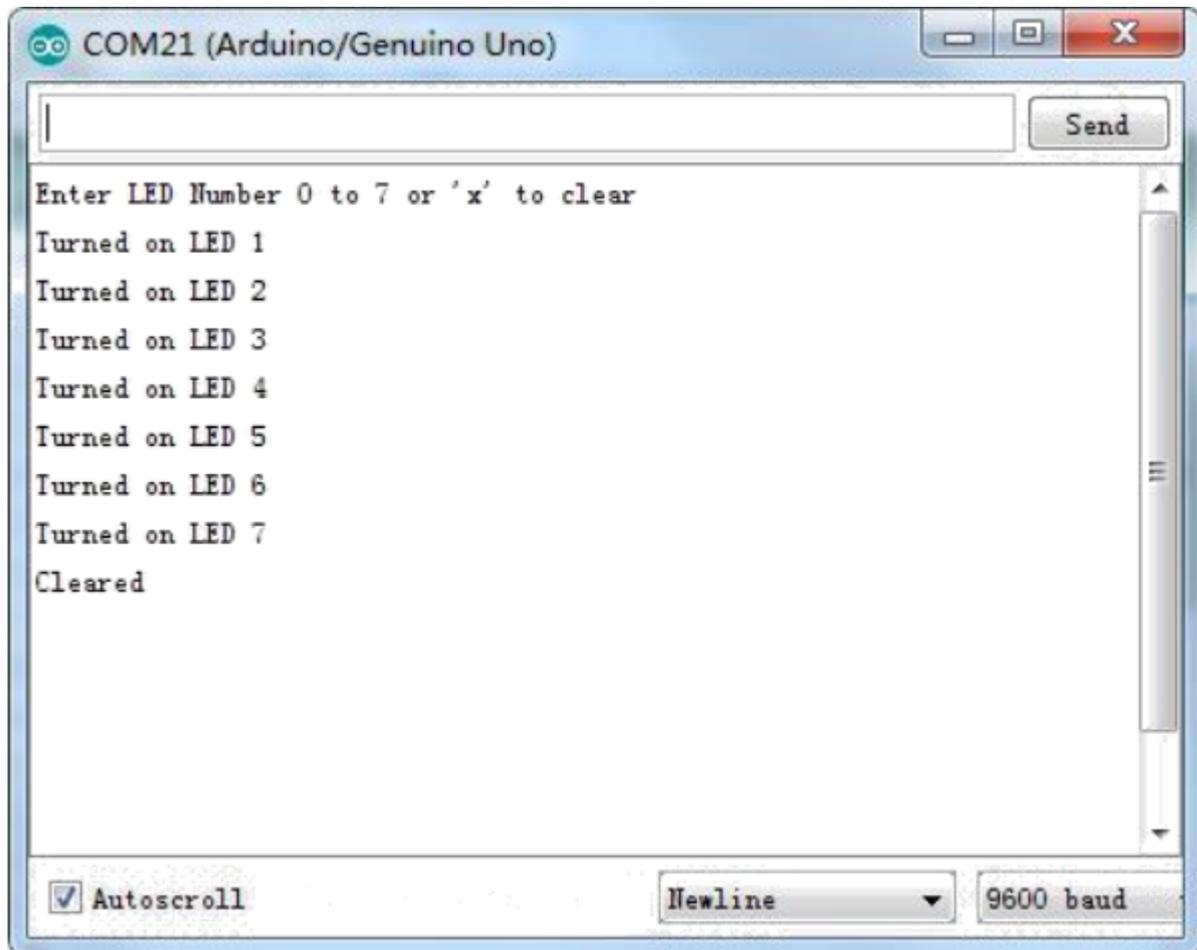
Die Meldung "Enter LED Number 0 to 7or 'x' to clear" wurde vom Mikrocontroller gesendet.

Sie teilt uns mit, welche Befehle wir an den Mikrocontroller senden können: entweder das 'x' (um alle LEDs auszuschalten) oder die Nummer der LED, die Sie einschalten möchten (wobei 0 für die erste LED steht, 1 für die nächsthöhere, bis hin zu 7 für die letzte LED).

Versuchen Sie, die folgenden Befehle in den oberen Bereich des seriellen Monitors einzugeben, der sich auf Höhe der Schaltfläche "Senden" befindet. Drücken Sie 'Send', nachdem Sie jedes dieser Zeichen eingegeben haben: x 0 3 5

Super Starter Kit

Die Eingabe von x hat keine Auswirkung, wenn die LEDs bereits alle aus sind, aber wenn Sie jede Zahl eingeben, sollte die entsprechende LED aufleuchten und Sie erhalten eine Bestätigungsmeldung vom Mikrocontroller-Board. Der serielle Monitor wird wie unten gezeigt angezeigt.



Geben Sie erneut x ein und drücken Sie "Senden", um alle LEDs auszuschalten.

Diese Skizze basiert auf der Skizze, die im vorherigen Beispiel verwendet wurde. Daher werden wir hier nur die neuen Bits behandeln. Sie werden es nützlich finden, den vollständigen Sketch in Ihrer Arduino IDE nachzuschlagen.

In der Funktion "setup" gibt es drei neue Zeilen am Ende:

```
Serial.begin(9600);  
while (! Serial); // Wait until Serial is ready - Leonardo  
Serial.println("Enter LED Number 0 to 7 or 'x' to clear");
```

Super Starter Kit

Erstens haben wir den Befehl 'Serial.begin(9600)'.

Damit wird die serielle Kommunikation gestartet, so dass der Mikrocontroller Befehle über die USB-Verbindung senden kann.

Der Wert 9600 wird als die "Baudrate" der Verbindung bezeichnet. Das ist die Geschwindigkeit, mit der die Daten gesendet werden sollen. Sie können diesen Wert auf einen höheren Wert ändern, aber Sie müssen auch den Arduino Serial Monitor auf denselben Wert einstellen. Wir werden dies später besprechen; belassen Sie es vorerst bei 9600.

Die Zeile, die mit "while" beginnt, stellt sicher, dass sich am anderen Ende der USB-Verbindung etwas befindet, mit dem der Mikrocontroller sprechen kann, bevor er mit dem Senden von Nachrichten beginnt. Andernfalls wird die Nachricht zwar gesendet, aber nicht angezeigt. Diese Zeile ist eigentlich nur notwendig, wenn Sie einen "Arduino Leonardo" verwenden, da der Mikrocontroller das Board automatisch zurücksetzt, wenn Sie den seriellen Monitor öffnen, während dies beim Leonardo nicht geschieht. Die letzte der neuen Zeilen in "setup" sendet die Nachricht aus, die wir oben im Serial Monitor sehen.

Die Funktion "loop" ist der Ort, an dem die ganze Aktion stattfindet:

Alles, was innerhalb der Schleife geschieht, ist in einer 'if'-Anweisung enthalten. Wenn also der Aufruf der eingebauten Arduino-Funktion 'Serial.available()' nicht 'true' ist, passiert nichts weiter. Serial.available() gibt 'true' zurück, wenn Daten an den Mikrocontroller gesendet wurden und dort zur Verarbeitung bereitstehen. Eingehende Nachrichten werden in einem so genannten Puffer gespeichert, und Serial.available() gibt 'true' zurück, wenn dieser Puffer nicht leer ist.

Wenn eine Nachricht empfangen wurde, geht es weiter zur nächsten Codezeile.

```
char ch = Serial.read();
```

Dies liest das nächste Zeichen aus dem Puffer und entfernt es aus dem Puffer. Außerdem wird es der Variablen 'ch' zugewiesen. Die Variable 'ch' ist vom Typ 'char', was für 'character' steht und wie der Name schon sagt, ein einzelnes Zeichen enthält.

Super Starter Kit

Wenn Sie die Anweisungen in der Eingabeaufforderung am oberen Rand des seriellen Monitors befolgt haben, wird dieses Zeichen entweder eine einstellige Zahl zwischen 0 und 7 oder der Buchstabe 'x' sein. Die 'if'-Anweisung in der nächsten Zeile überprüft, ob es sich um eine einzelne Ziffer handelt, indem sie feststellt, ob 'ch' größer oder gleich dem Zeichen '0' und kleiner oder gleich dem Zeichen '7' ist. Es sieht ein wenig seltsam aus, Zeichen auf diese Weise zu vergleichen, ist aber durchaus akzeptabel. Jedes Zeichen wird durch eine eindeutige Zahl, den sogenannten ASCII-Wert, dargestellt. Wenn wir also Zeichen mit `<=` und `>=` vergleichen, sind es tatsächlich die ASCII-Werte, die verglichen werden.

Wenn der Test erfolgreich war, kommen wir zur nächsten Zeile:

```
int led = ch - '0';
```

Jetzt führen wir arithmetische Berechnungen mit Zeichen durch! Wir subtrahieren die Ziffer '0' von der eingegebenen Ziffer. Wenn Sie also '0' getippt haben, dann ist '0' - '0' gleich 0. Wenn Sie '7' getippt haben, dann ist '7' - '0' gleich die Zahl 7, weil es tatsächlich die ASCII-Werte sind, die bei der Subtraktion verwendet werden. Da wir die Nummer der LED kennen, die wir einschalten wollen, müssen wir nur dieses Bit in der Variablen 'leds' setzen und das Schieberegister aktualisieren.

In den nächsten beiden Zeilen wird eine Bestätigungsmeldung an den seriellen Monitor zurückgeschrieben.

```
Serial.print("Turned on LED ");  
Serial.println(led);
```

In der ersten Zeile wird `Serial.print` und nicht `Serial.println` verwendet. Der Unterschied zwischen den beiden besteht darin, dass `Serial.print` nach dem Ausdruck des Parameters keine neue Zeile beginnt. Wir verwenden dies in der ersten Zeile, weil wir die Nachricht in zwei Teilen drucken. Erstens das allgemeine Bit: 'Turned on LED ' und dann die Nummer der LED. Die Nummer der LED wird in einer 'int'-Variablen gespeichert und nicht als Textstring. `Serial.print` kann entweder einen in Anführungszeichen eingeschlossenen

Super Starter Kit

Textstring oder eine 'int'-Variable oder so ziemlich jeden Variablentyp verarbeiten. Nach der 'if'-Anweisung, die den Fall behandelt, dass eine einzelne Ziffer behandelt wurde, gibt es eine zweite 'if'-Anweisung, die prüft, ob 'ch' der Buchstabe 'x' ist.

```
if (ch == 'x')
{
  leds = 0;
  updateShiftRegister();
  Serial.println("Cleared");
}
```

Ist dies der Fall, werden alle LEDs gelöscht und eine Bestätigungsmeldung gesendet.

Super Starter Kit

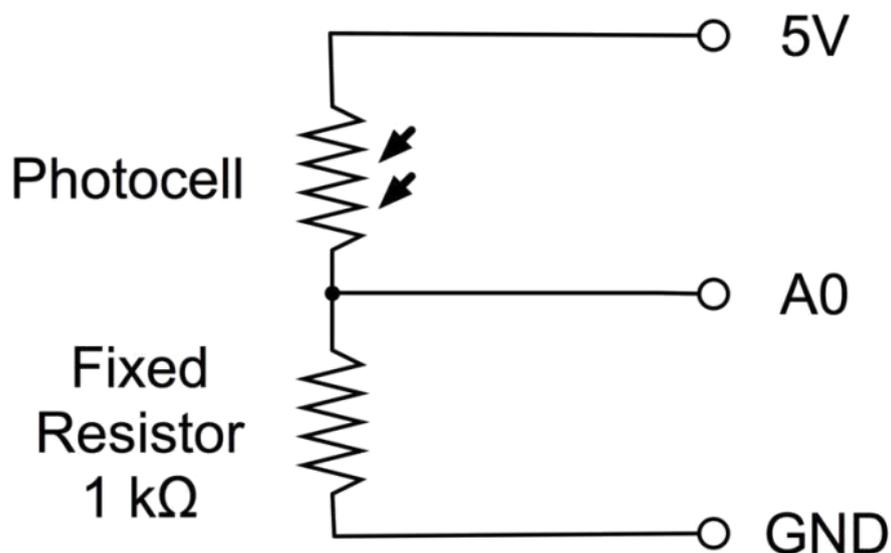
Fotowiderstand

In diesem Beispiel lernen Sie, wie Sie die Lichtintensität mit Hilfe eines Analogeingangs messen können. Sie bauen auf dem Beispiel von "8 LEDs mit 74HC595" auf und verwenden die Lichtstärke, um die Anzahl der zu beleuchtenden LEDs zu steuern.

Für dieses Beispiel benötigen Sie:

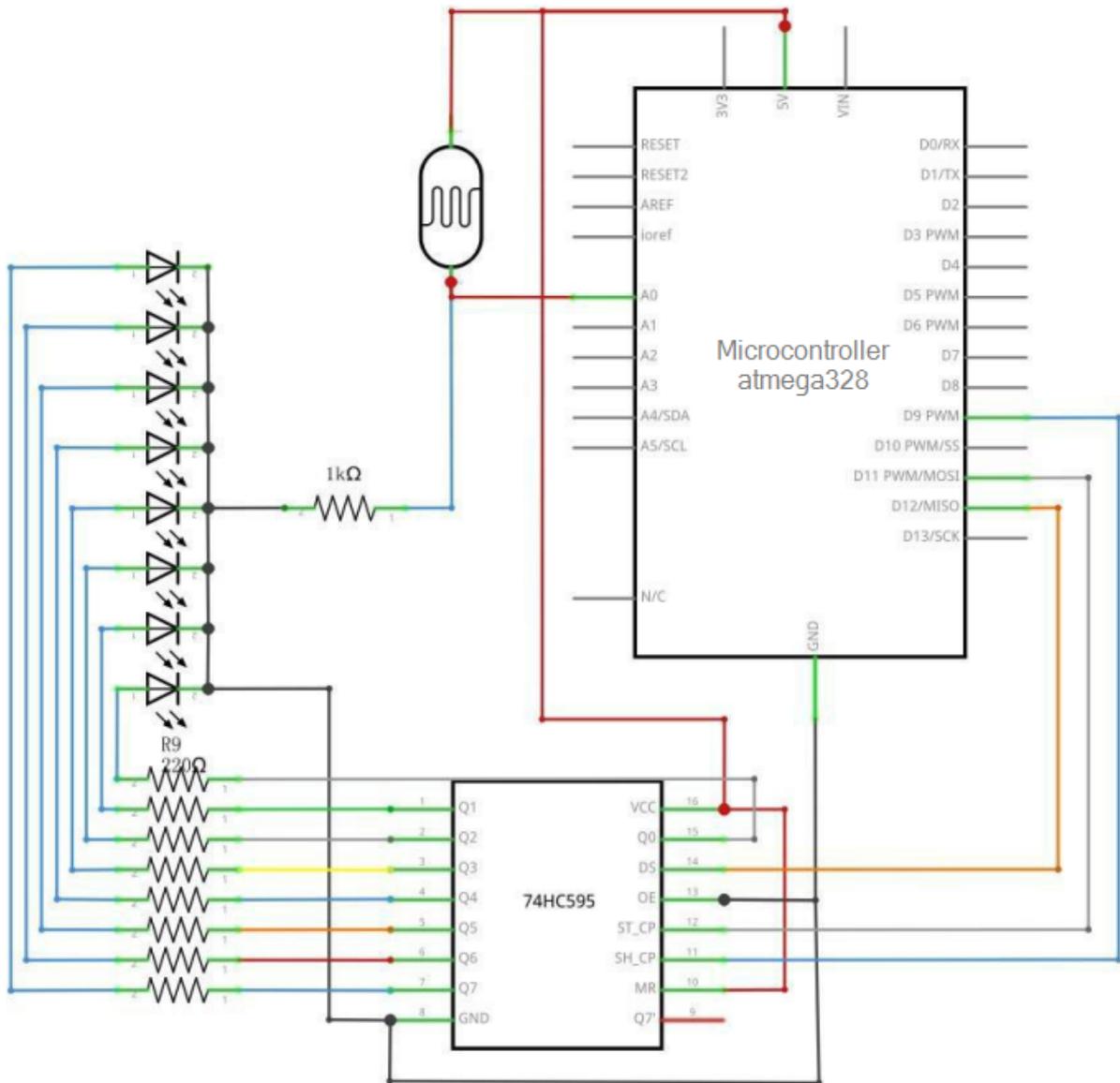
- alles aus dem Beispiel "8 LEDs mit 74HC595".
- Fotowiderstand
- 3 M-M-Drähte (Überbrückungsdrähte von Stecker zu Stecker)
- 1K-Ohm-Widerstand

Der verwendete Fotowiderstand ist ein so genannter lichtabhängiger Widerstand, manchmal auch LDR genannt. Wie der Name schon sagt, verhalten sich diese Bauteile wie ein Widerstand, mit dem Unterschied, dass sich der Widerstand in Abhängigkeit von der Lichteinstrahlung ändert. Dieses Bauteil hat einen Widerstand von etwa $50\text{ k}\Omega$ bei fast völliger Dunkelheit und $500\ \Omega$ bei hellem Licht. Um diesen veränderlichen Widerstandswert in etwas umzuwandeln, das wir am Analogeingang eines Mikrocontroller-Boards messen können, muss er in eine Spannung umgewandelt werden. Der einfachste Weg, dies zu tun, ist die Kombination mit einem Festwiderstand.



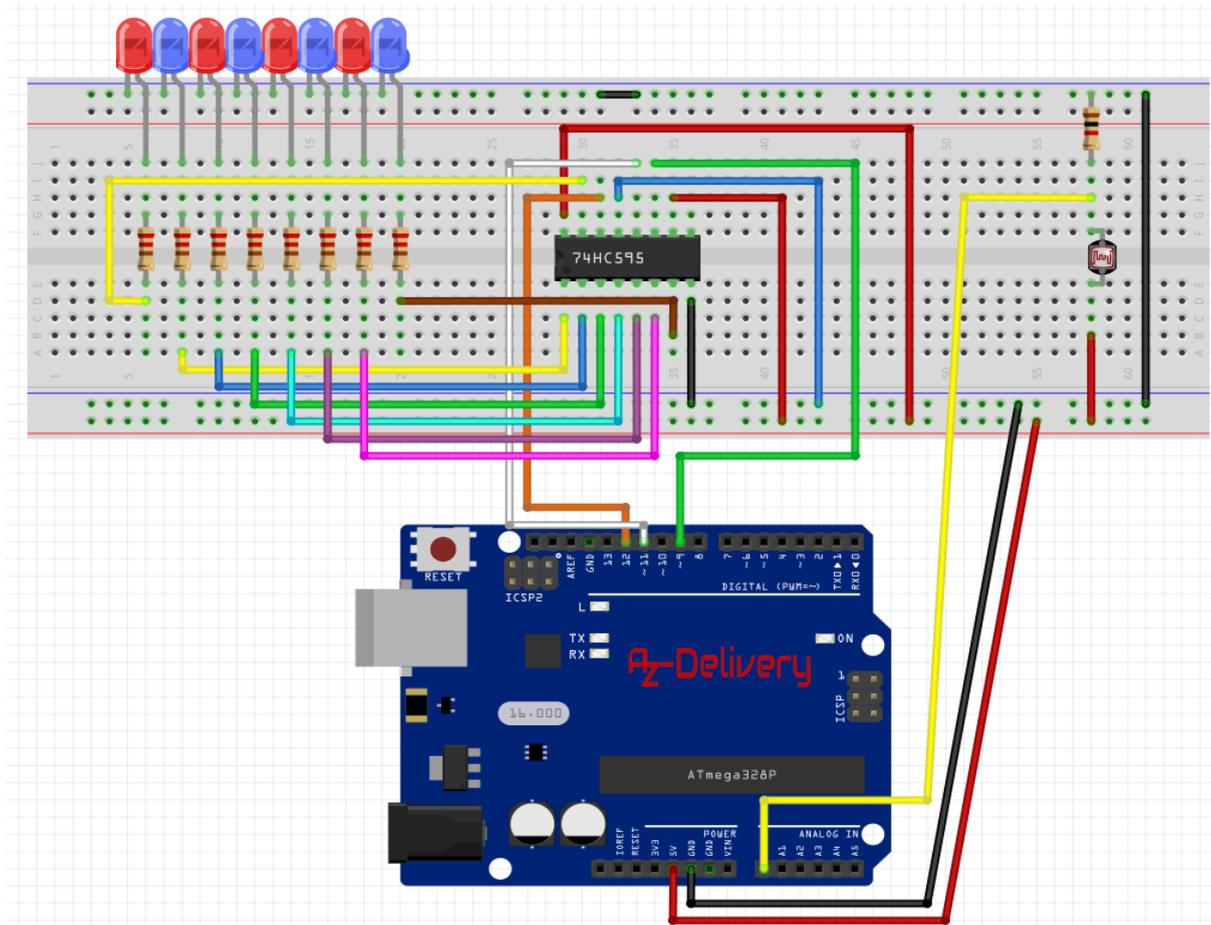
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Super Starter Kit

Beispiel-Code

```
int lightPin = 0;
int latchPin = 11;
int clockPin = 9;
int dataPin = 12;

int leds = 0;

void setup()
{
  pinMode(latchPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
}
void updateShiftRegister()
{
  digitalWrite(latchPin, LOW);
  shiftOut(dataPin, clockPin, LSBFIRST, leds);
  digitalWrite(latchPin, HIGH);
}
void loop()
{
  int reading = analogRead(lightPin);
  int numLEDSLit = reading / 57; //1023 / 9 / 2
  if (numLEDSLit > 8) numLEDSLit = 8;
  leds = 0; // no LEDs lit to start
  for (int i = 0; i < numLEDSLit; i++)
  {
    leds = leds + (1 << i); // sets the i'th bit
  }
  updateShiftRegister();
}
```

Laden Sie diesen Code auf Ihren Mikrocontroller.

LCD1602 I2C-Modul

In diesem Beispiel lernen Sie, wie Sie eine alphanumerische LCD-Anzeige verdrahten und verwenden.

Das Display hat eine LED-Hintergrundbeleuchtung und kann zwei Zeilen mit bis zu 16 Zeichen in jeder Zeile anzeigen. Sie können die Rechtecke für jedes Zeichen auf dem Display und die Pixel sehen, aus denen jedes Zeichen besteht. Das Display ist einfach weiß auf blau und ist für die Anzeige von Text gedacht.

//lib ist LiquidCrystal I2C

Für dieses Beispiel benötigst du:

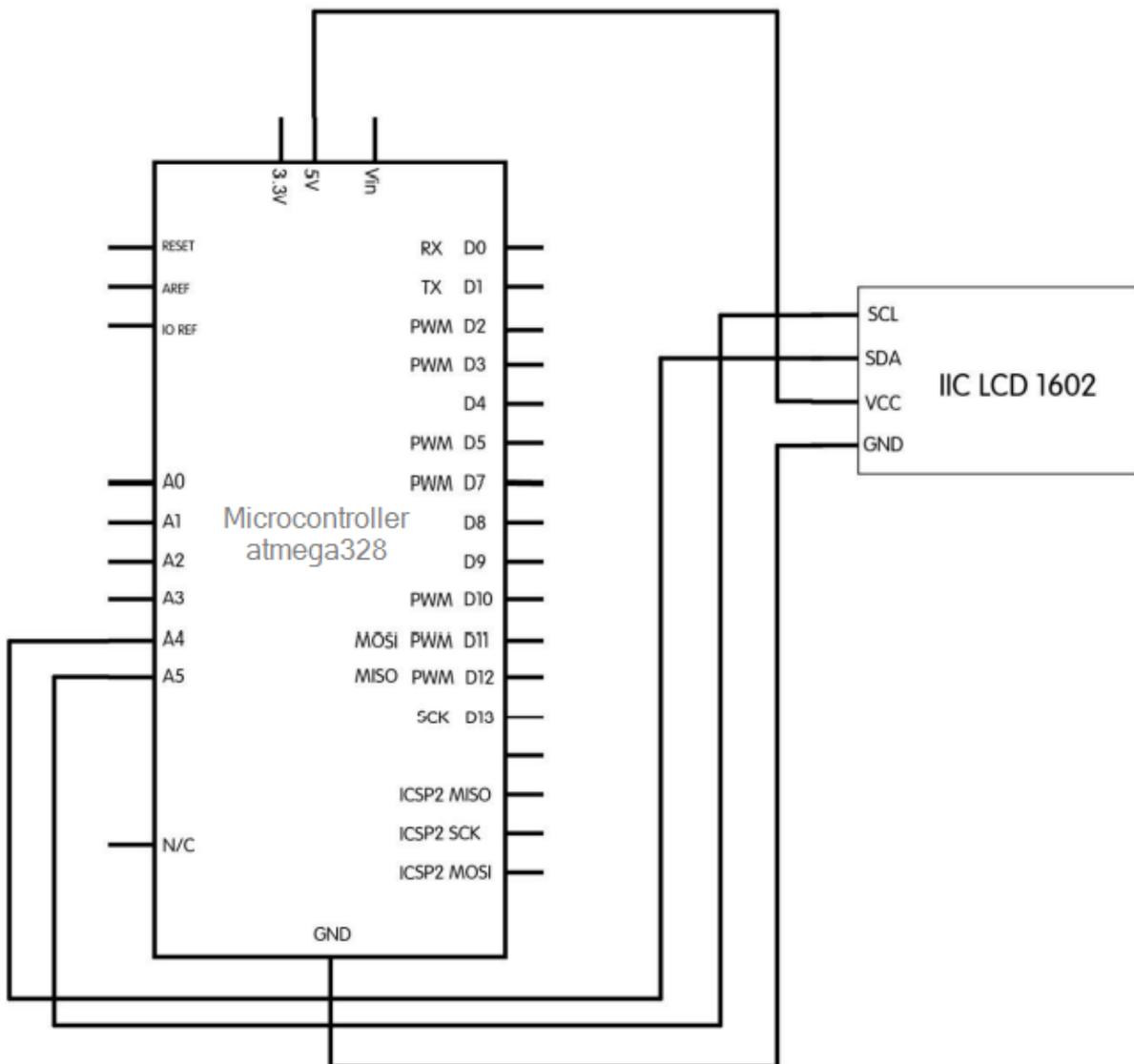
- Mikrocontroller-Platine
- LCD1602 mit I2C-Modul
- Potentiometer 10K
- 4pcs F-M Drähte (Female zu Male Dupont Drähte)

Dies ist eine großartige LCD-Anzeige, die mit den meisten Mikrocontrollern kompatibel ist. Da die Pin-Ressourcen begrenzt sind, wird Ihr Projekt mit normalen LCDs schnell an seine Grenzen stoßen. Mit diesem LCD-Modul mit I2C-Schnittstelle benötigen Sie nur 2 Zeilen, um die Informationen anzuzeigen. Wenn Sie bereits I2C-Bausteine in Ihrem Projekt haben, kostet dieses LCD-Modul eigentlich gar keine Pins mehr.

Das I2C Modul befindet sich auf der Rückseite des LCDs.

Die Standard I2C Adresse dieses LCDs ist 0x27.

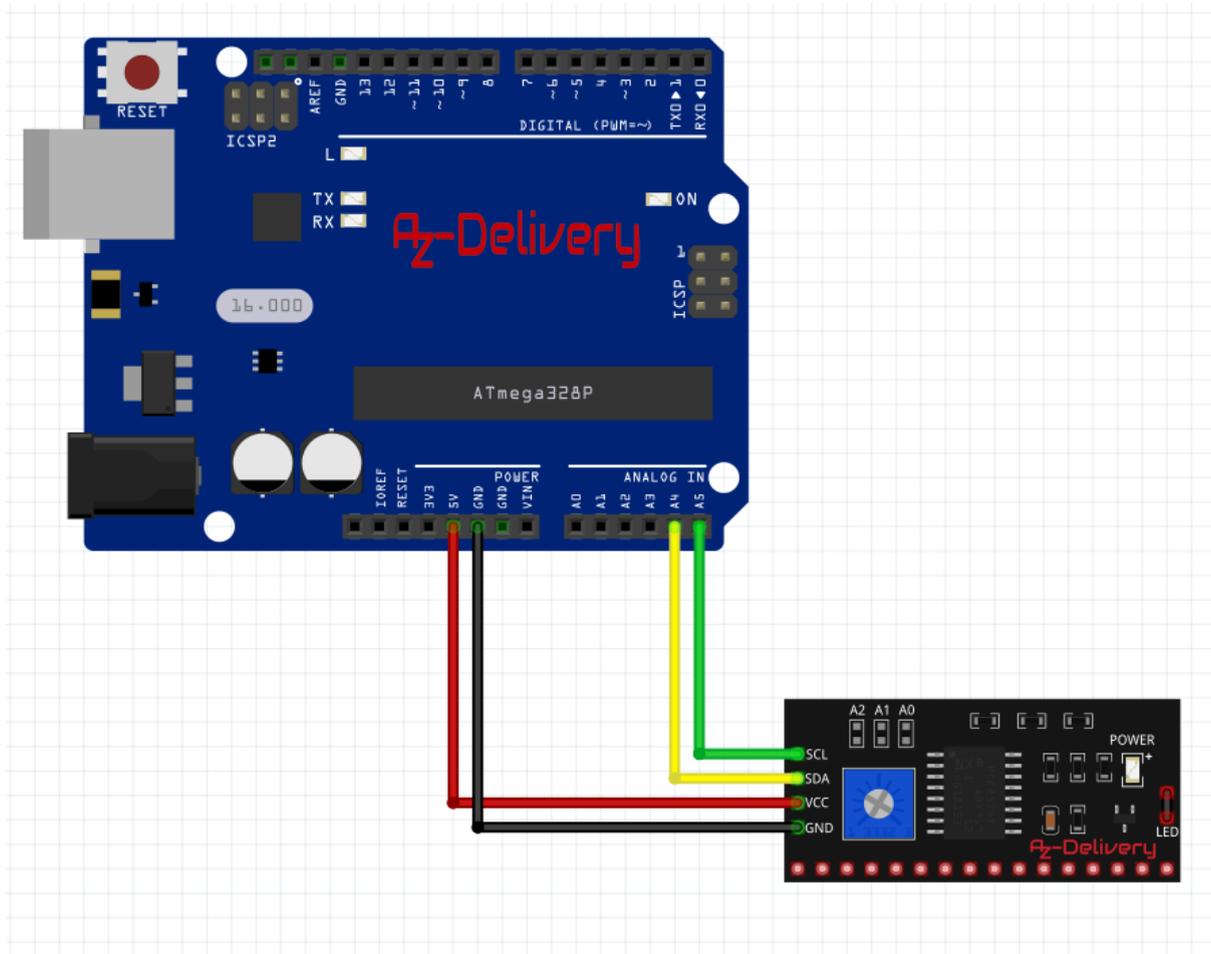
Schematische Darstellung



Super Starter Kit

Verdrahtungsschema

Drehen Sie Ihr Display so, dass das I2C-Modul nach oben zeigt. Jetzt können Sie ganz einfach Verbindungen herstellen.



Super Starter Kit

Beispiel-Code

Zuerst müssen Sie die "LiquidCrystal I2C" Library über den Library Manager installieren.

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
```

```
LiquidCrystal_I2C lcd(0x27,16,2);//set the LCD address to
0x27 for a 16 chars and 2 line display
```

```
void setup()
{
  lcd.init();// initialize the lcd
  // Print a message to the LCD.
  lcd.backlight();
  lcd.setCursor(0,0);
  lcd.print("Hello, world!");
  lcd.setCursor(0,1);
  lcd.print("LiquidCrystal");
}
```

```
void loop()
{
}
```

Laden Sie diesen Code in Ihren Mikrocontroller hoch. Sie sollten nun die Meldung "Hello world" sehen. Wenn Sie nichts auf dem Display sehen, können Sie den Kontrast durch Drehen des Potentiometers auf dem I2C-Modul einstellen.

Thermometer

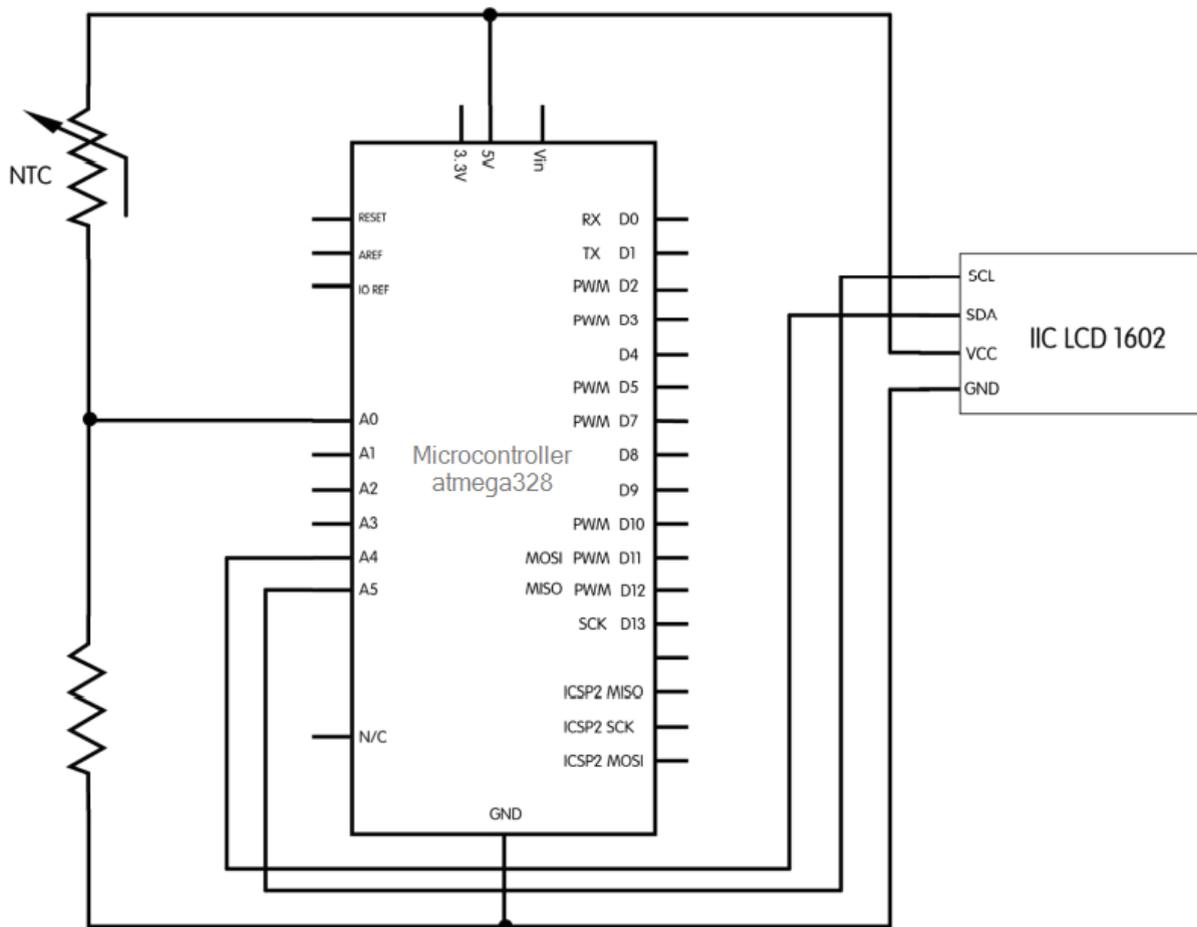
In diesem Beispiel werden Sie ein LCD-Display verwenden, um die Temperatur anzuzeigen.

Für dieses Beispiel benötigen Sie:

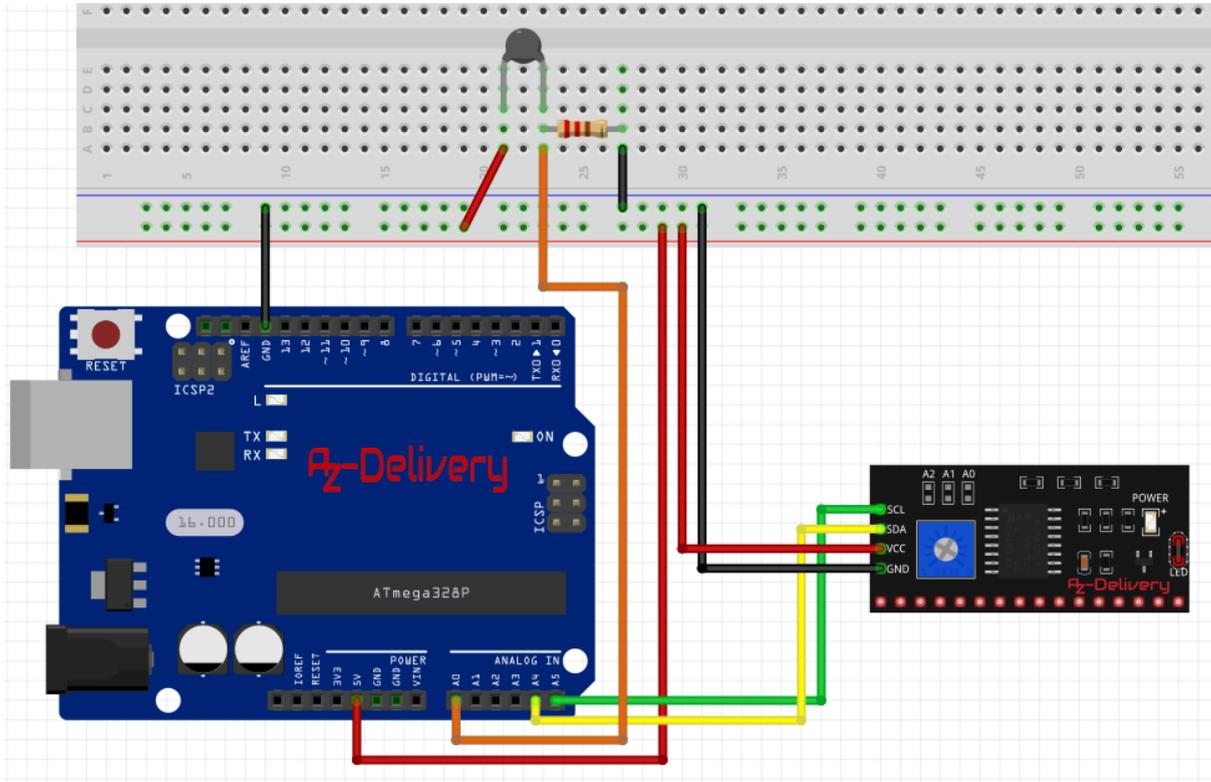
- Mikrocontroller-Platine
- LCD 1602 mit I2C-Modul
- 10-k-Ohm-Widerstand
- Thermistor
- Potentiometer
- 830-poliges Breadboard
- 4 St. F-M-Drähte (Buchse zu Stecker Dupont-Drähte)
- 5 St. M-M-Drähte (männliche zu männlichen Brückendrähnen)

Ein Thermistor ist ein Wärmewiderstand - ein Widerstand, der seinen Widerstand mit der Temperatur ändert. Technisch gesehen sind alle Widerstände Thermistoren - ihr Widerstand ändert sich geringfügig mit der Temperatur - aber die Änderung ist normalerweise sehr gering und schwer zu messen. Thermistoren sind so gebaut, dass sich der Widerstand drastisch mit der Temperatur ändert, so dass die Änderung 100 Ohm oder mehr pro Grad betragen kann! Es gibt zwei Arten von Thermistoren, NTC (negativer Temperaturkoeffizient) und PTC (positiver Temperaturkoeffizient). Im Allgemeinen werden NTC-Sensoren für Temperaturmessungen verwendet. PTCs werden oft als rücksetzbare Sicherungen verwendet - ein Temperaturanstieg erhöht den Widerstand, was bedeutet, dass sie sich erhitzen, wenn mehr Strom durch sie fließt, und den Strom "drosseln", was für den Schutz von Schaltkreisen sehr praktisch ist!

Schematische Darstellung



Schaltplan



Super Starter Kit

Beispiel-Code

```
#include <Wire.h>
#include <LiquidCrystal_I2C.h>
LiquidCrystal_I2C lcd(0x27,16,2);
int tempPin = 0;
void setup()
{
  lcd.init();
  lcd.backlight();
}
void loop()
{
  int tempReading = analogRead(tempPin);
  double tempK = log(10000.0 * ((1024.0 / tempReading - 1)));
  tempK = 1 / (0.001129148 + (0.000234125 + (0.0000000876741 *
tempK * tempK )) * tempK );//Temp Kelvin
  float tempC = tempK - 273.15;           //Convert Kelvin to
Celcius
  float tempF = (tempC * 9.0)/ 5.0 + 32.0; //Convert Celsius to
Fahrenheit
  lcd.setCursor(0, 0);
  lcd.print("Temp          C ");
  // Display Temperature in F
  //lcd.print("Temp          F ");
  lcd.setCursor(6, 0);
  // Display Temperature in C
  lcd.print(tempC);
  // Display Temperature in F
  //lcd.print(tempF);
  delay(500);
}
```

Laden Sie diesen Code auf Ihren Mikrocontroller. Sie werden nun die Temperatur auf dem LCD-Display sehen.

Super Starter Kit

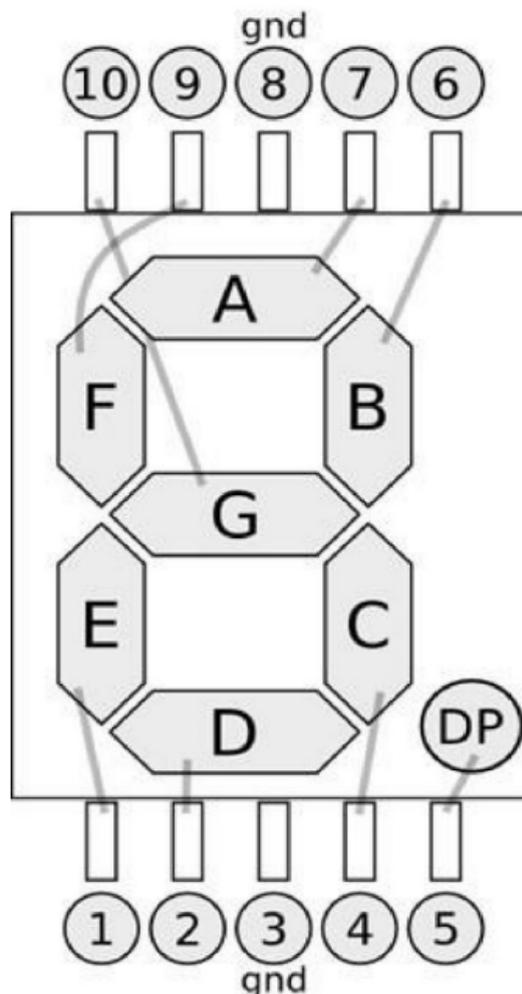
7 Segmentanzeige und 74HC595

In diesem Beispiel wird das Schieberegister 74HC595 zur Steuerung der Segmentanzeige verwendet. Die Segmentanzeige wird Zahlen von 9-0 anzeigen.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- 830-poliges Breadboard
- 74HC595 IC
- 1-stellige 7-Segment-Anzeige
- 8 Stück 220 Ohm Widerstand
- 26pcs M-M Drähte (männlich zu männlich Jumper Drähte)

Unten ist das 7-Segment-Pin-Diagramm:



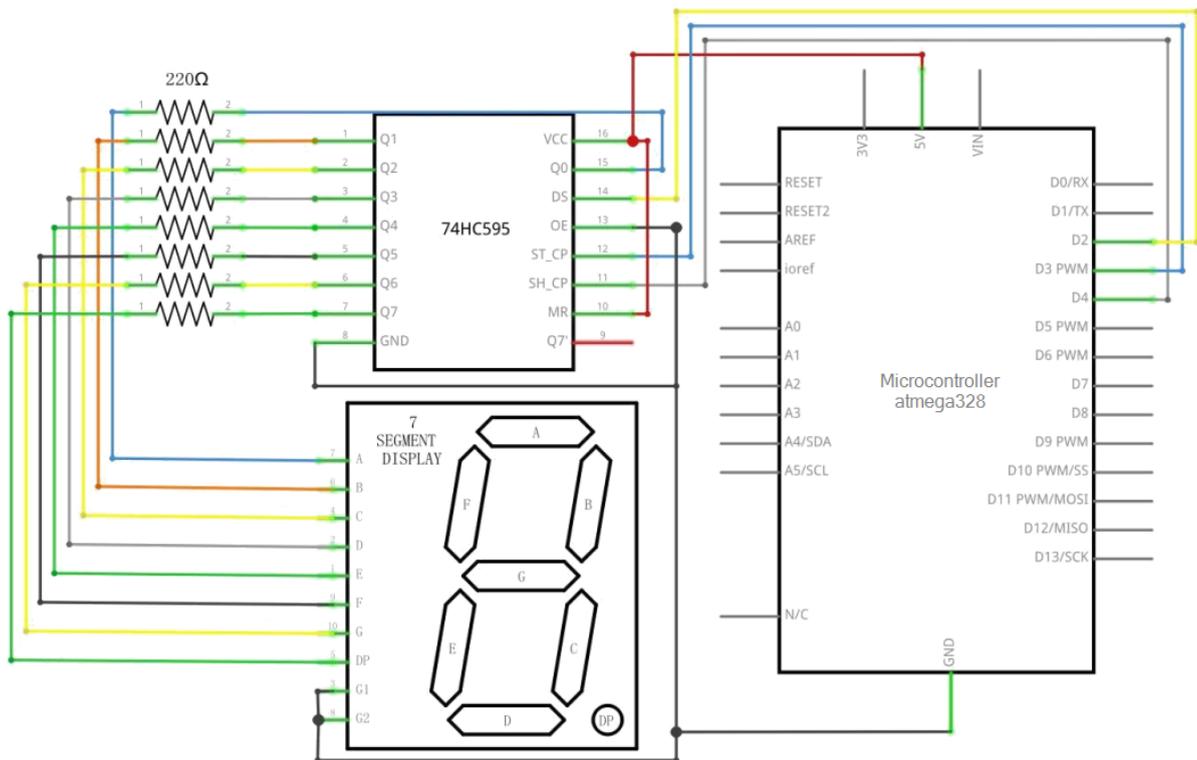
Super Starter Kit

0-9, zehn Ziffern entsprechen mit jedem Segment sind wie folgt (die folgende Tabelle gilt gemeinsame Kathode sieben Segment-Anzeigergeräte).

Nummer	DP	A	B	C	D	E	F	G
0	0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	0	0
2	0	1	1	0	1	1	0	1
3	0	1	1	1	1	0	0	1
4	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1
6	0	1	0	1	1	1	1	1
7	0	1	1	1	0	0	0	0
8	0	1	1	1	1	1	1	1
9	0	1	1	1	1	0	1	1

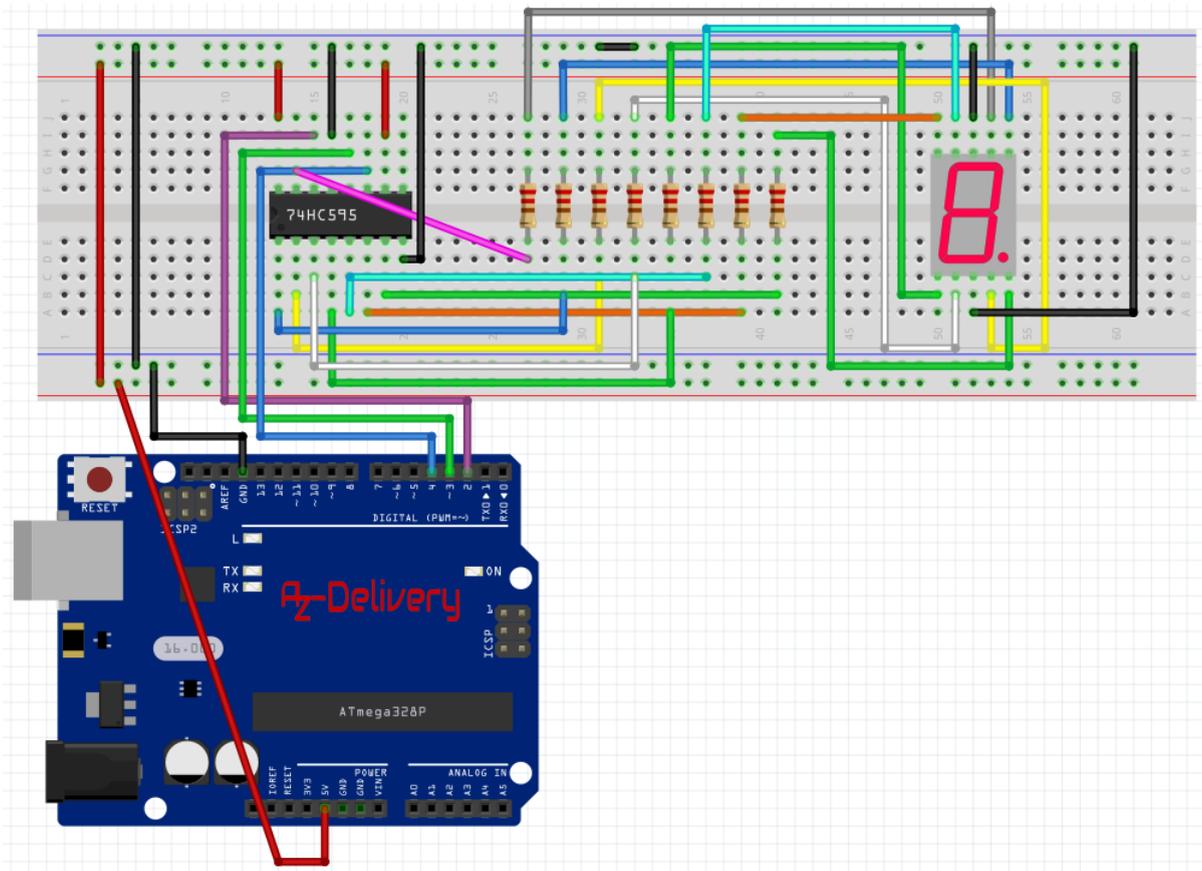
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Beispiel-Code

```
// define the LED digit patterns, from 0 - 9
// 1 = LED on, 0 = LED off, in this order:
//          74HC595 pin      Q0,Q1,Q2,Q3,Q4,Q5,Q6,Q7
//          Mapping to      a,b,c,d,e,f,g of Seven-Segment
LED
byte seven_seg_digits[10] = { B11111100, // = 0
                              B01100000, // = 1
                              B11011010, // = 2
                              B11110010, // = 3
                              B01100110, // = 4
                              B10110110, // = 5
                              B10111110, // = 6
                              B11100000, // = 7
                              B11111110, // = 8
                              B11100110, // = 9
                              B00000001  // = DP
                              };

int latchPin = 3;
int clockPin = 4;
int dataPin = 2;

void setup() {
  // Set latchPin, clockPin, dataPin as output
  pinMode(latchPin, OUTPUT);
  pinMode(clockPin, OUTPUT);
  pinMode(dataPin, OUTPUT);
}
// display a number on the digital segment display
void sevenSegWrite(byte digit) {
  // set the latchPin to low potential, before sending data
  digitalWrite(latchPin, LOW);
  // the original data (bit pattern)
  shiftOut(dataPin, clockPin, LSBFIRST, seven_seg_digits[digit]);
  // set the latchPin to high potential, after sending data
  digitalWrite(latchPin, HIGH);
}
void loop() {
  // count from 9 to 0

```

Super Starter Kit

```
for (byte digit = 11; digit > 0; --digit) {  
    delay(1000);  
    sevenSegWrite(digit - 1);  
}  
  
// suspend 3 seconds  
delay(3000);  
}
```

Laden Sie diesen Code auf Ihren Mikrocontroller. Sie werden nun sehen, dass die Anzeige von 9 bis 0 zählt.

7-Segment-Anzeige 4 Ziffern

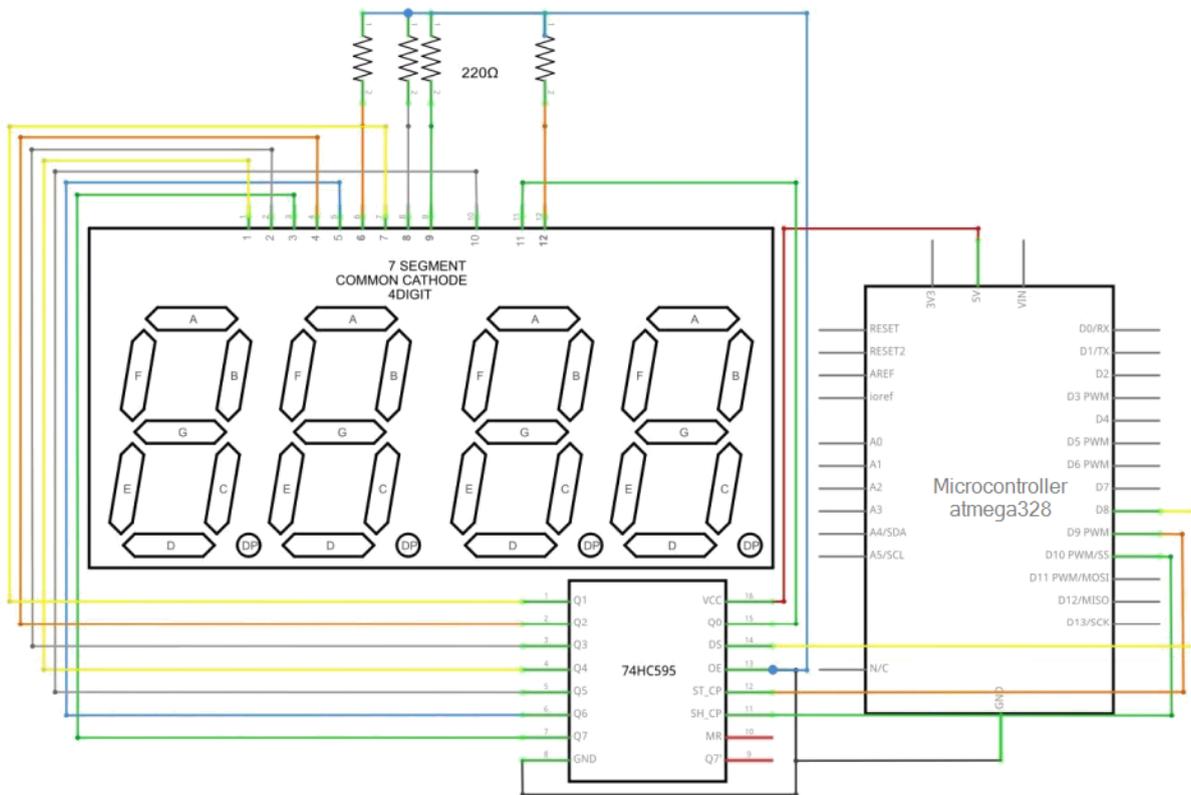
In diesem Beispiel lernen Sie, wie Sie eine 4-stellige 7-Segment-Anzeige verwenden.

Wenn Sie eine 1-stellige 7-Segment-Anzeige verwenden, beachten Sie bitte, dass bei einer gemeinsamen Anode der gemeinsame Anoden-Pin mit der Stromquelle verbunden ist; bei einer gemeinsamen Kathode ist der gemeinsame Kathoden-Pin mit dem GND verbunden. Bei Verwendung einer 4-stelligen 7-Segment-Anzeige wird über den gemeinsamen Anoden- oder den gemeinsamen Kathoden-Pin gesteuert, welche Ziffer angezeigt wird. Auch wenn nur eine Ziffer in Betrieb ist,

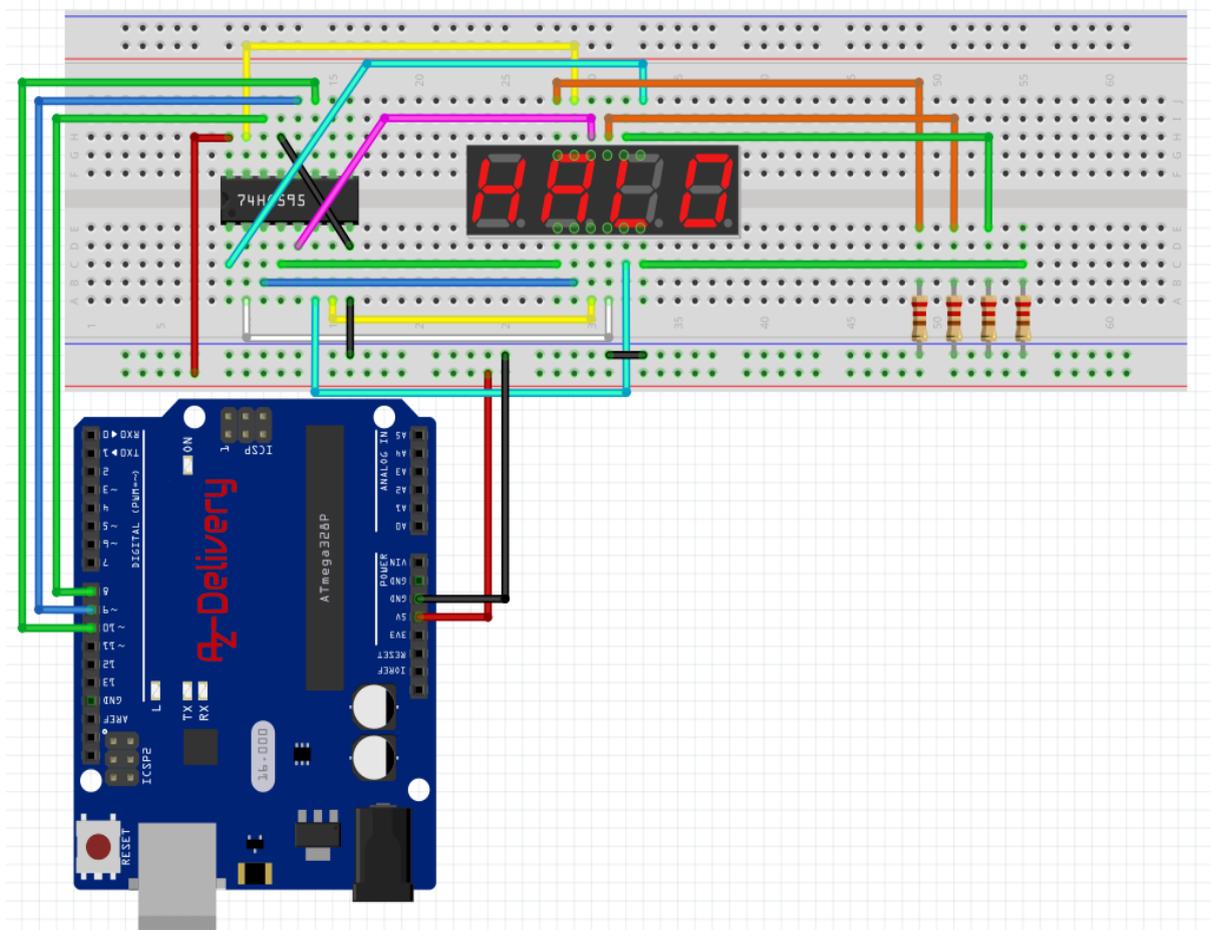
Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- 830-poliges Breadboard
- 74HC595 IC
- 4-stellige 7-Segment-Anzeige
- 4 Stück 220-Ohm-Widerstand
- 20pcs M-M Drähte (Männlich zu Männlich Überbrückungsdrähte)

Schematische Darstellung



Schaltplan



Super Starter Kit

Beispiel-Code

```
int latch = 9; //74HC595 pin 9 STCP
int clock = 10; //74HC595 pin 10 SHCP
int data = 8; //74HC595 pin 8 DS

unsigned char table[] =
{ 0x3f, 0x06, 0x5b, 0x4f, 0x66, 0x6d, 0x7d, 0x07, 0x7f,
  0x6f, 0x77, 0x7c
  , 0x39, 0x5e, 0x79, 0x71, 0x00
};

void setup() {
  pinMode(latch, OUTPUT);
  pinMode(clock, OUTPUT);
  pinMode(data, OUTPUT);
}

void Display(int num)
{
  digitalWrite(latch, LOW);
  shiftOut(data, clock, MSBFIRST, table[num]);
  digitalWrite(latch, HIGH);
}

void loop() {
  for (int i = 0; i < 17; i++) {
    Display(i);
    delay(500);
  }
}
```

Laden Sie diesen Code auf Ihren Mikrocontroller.

Gleichstrommotor

In diesem Beispiel lernen Sie, wie Sie einen kleinen Gleichstrommotor mit einem Mikrocontroller und einem Treiber-IC steuern können.

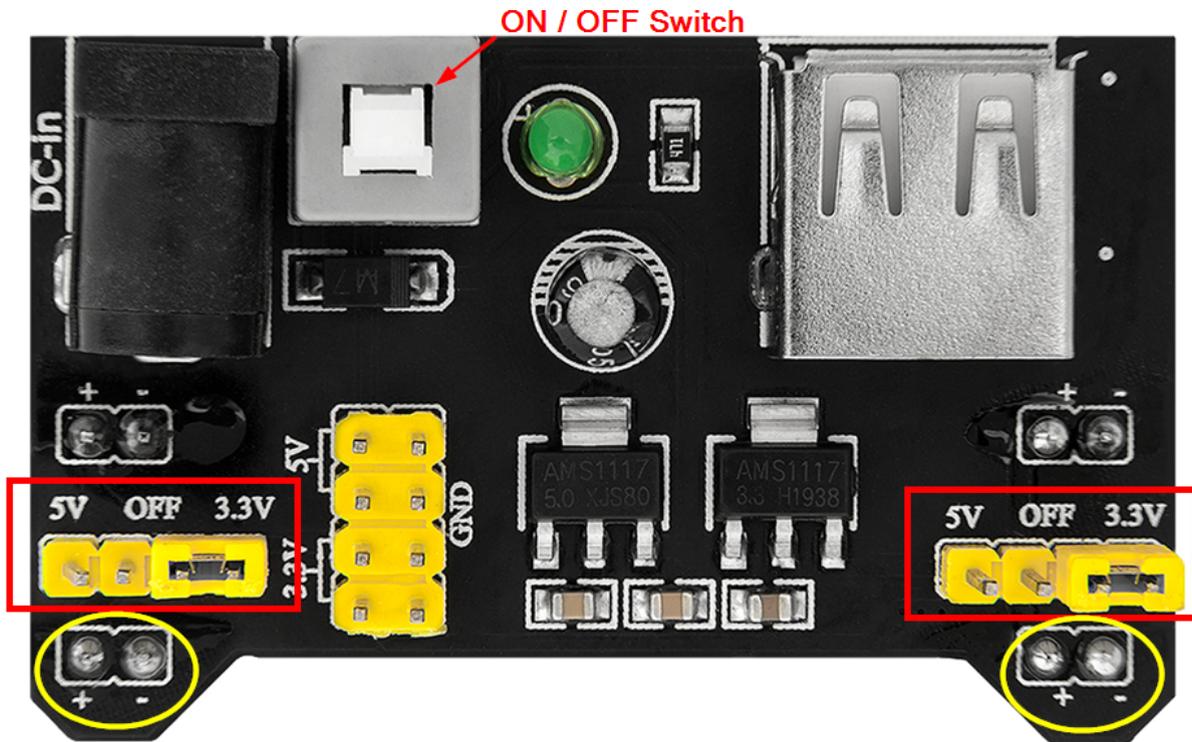
Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- 830-poliges Breadboard
- L293D IC
- Lüfterrad und Gleichstrommotor
- 5 Stück M-M Drähte (männliche zu männliche Überbrückungsdrähte)
- MB102 Breadboard Spannungsversorgung
- 9V 1A Netzadapter

Der kleine Gleichstrommotor wird wahrscheinlich mehr Strom verbrauchen, als der digitale Ausgang des Mikrocontroller-Boards direkt verarbeiten kann. Wenn wir versuchen würden, den Motor direkt an einen Pin der Mikrocontrollerplatine anzuschließen, bestünde die Gefahr, dass die Mikrocontrollerplatine beschädigt wird. Daher verwenden wir ein Stromversorgungsmodul, um den Motor mit Strom zu versorgen.

Super Starter Kit

Einstellen der Ausgangsspannung



Der linke und rechte Spannungsausgang (rotes Rechteck) können unabhängig voneinander konfiguriert werden. Um die Ausgangsspannung zu wählen, schieben Sie den Jumper auf die entsprechenden Pins.

Achten Sie darauf, dass Sie das Modul richtig auf dem Breadboard ausrichten. Der negative Pin (-) des Moduls muss mit der blauen Linie (-) auf dem Breadboard und der positive Pin (+) mit der roten Linie (+) übereinstimmen. Wenn Sie dies nicht tun, könnten Sie versehentlich die Stromversorgung Ihres Projekts umkehren.

Super Starter Kit

L293D

Der L293 und der L293D sind vierfache Hochstrom-Half-H-Treiber. Der L293 liefert bidirektionale Treiberströme von bis zu 1 A bei Spannungen von 4,5 V bis 36 V. Der L293D liefert bidirektionale Treiberströme von bis zu 600 mA bei Spannungen von 4,5 V bis 36 V. Beide Bausteine sind für die Ansteuerung von induktiven Lasten wie Relais, Solenoiden, Gleichstrom- und bipolaren Schrittmotoren sowie anderen Hochstrom-/Hochspannungslasten in Anwendungen mit positiver Versorgungsspannung ausgelegt. Alle Eingänge sind TTL-kompatibel. Jeder Ausgang ist eine vollständige Totem-Pol-Treiberschaltung mit einer Darlington-Transistorsenke und einer Pseudo-Darlington-Quelle. Die Treiber werden paarweise aktiviert, wobei die Treiber 1 und 2 durch 1,2EN und die Treiber 3 und 4 durch 3,4EN aktiviert werden. Wenn ein Enable-Eingang High ist, werden die zugehörigen Treiber aktiviert, und ihre Ausgänge sind aktiv und in Phase mit ihren Eingängen. Wenn der Freigabeeingang niedrig ist, sind diese Treiber deaktiviert, und ihre Ausgänge sind ausgeschaltet und befinden sich im Zustand hoher Impedanz. Mit den richtigen Dateneingängen bildet jedes Treiberpaar einen reversiblen Voll-H- (oder Brücken-) Antrieb, der für Magnet- oder Motoranwendungen geeignet ist.

Es sind 3 Drähte mit der Mikrocontrollerplatine, 2 Drähte mit dem Motor und 1 Draht mit einer Batterie verbunden.

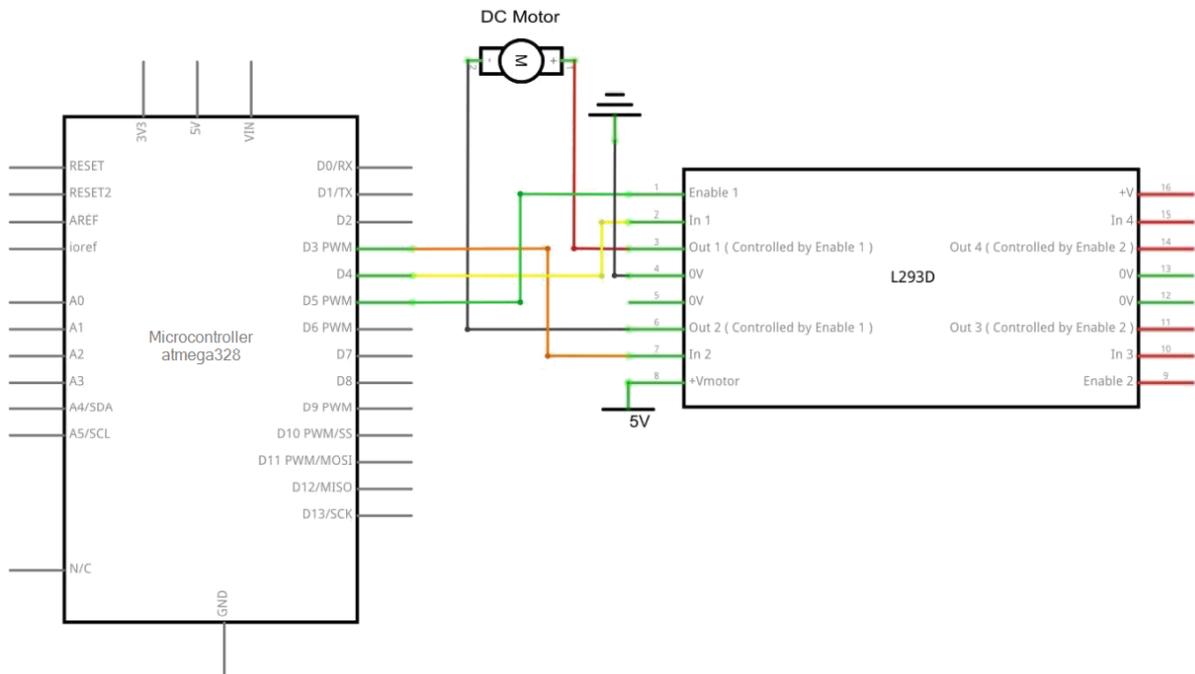
L293D			
M1 PWM	1	16	Battery +ve
M1 direction 0/1	2	15	M2 direction 0/1
M1 +ve	3	14	M2 +ve
GND	4	13	GND
GND	5	12	GND
M1 -ve	6	11	M2 -ve
M1 direction 1/0	7	10	M2 direction 1/0
Battery +ve	8	9	M2 PWM
Motor 1		Motor 2	

Super Starter Kit

Die linke Seite bezieht sich auf den ersten Motor, die rechte Seite auf einen zweiten Motor. Ja, Sie können ihn auch mit nur einem angeschlossenen Motor betreiben.

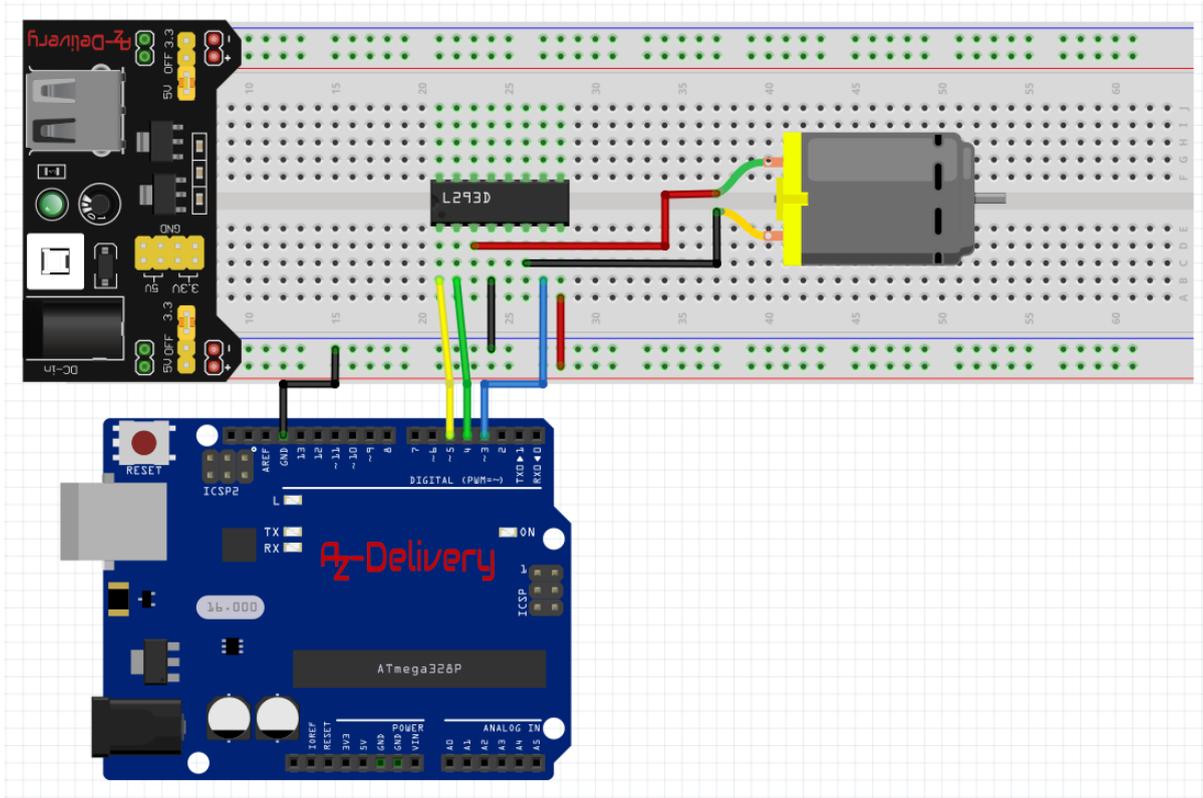
M1 PWM - verbinden Sie dies mit einem PWM-Pin auf der Mikrocontroller-Platine. Sie sind auf der Platine mit einem ~ gekennzeichnet, Pin 5 ist ein Beispiel. Geben Sie eine beliebige ganze Zahl zwischen 0 und 255 aus, wobei 0 ausgeschaltet ist, 128 die halbe Geschwindigkeit und 255 die maximale Geschwindigkeit darstellt. M1 Richtung 0/1 und M1 Richtung 1/0 - Schließen Sie diese beiden an zwei digitale MCU-Pins an. Geben Sie einen Pin als HIGH und den anderen Pin als LOW aus, und der Motor dreht sich in eine Richtung. Kehren Sie die Ausgänge auf LOW und HIGH um, so dreht sich der Motor in die andere Richtung.

Schematische Darstellung



Super Starter Kit

Schaltplan



Beispiel-Code

```
#define ENABLE 5
#define DIRA 3
#define DIRB 4

void setup() {
  //---set pin direction
  pinMode(ENABLE, OUTPUT);
  pinMode(DIRA, OUTPUT);
  pinMode(DIRB, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  //---back and forth example
  Serial.println("One way, then reverse");
  digitalWrite(ENABLE, HIGH); // enable on
  for (int i = 0; i < 5; i++) {
    digitalWrite(DIRA, HIGH); //one way
    digitalWrite(DIRB, LOW);
    delay(500);
    digitalWrite(DIRA, LOW); //reverse
    digitalWrite(DIRB, HIGH);
    delay(500);
  }
  digitalWrite(ENABLE, LOW); // disable
  delay(2000);

  Serial.println("fast Slow example");
  //---fast/slow stop example
  digitalWrite(ENABLE, HIGH); //enable on
  digitalWrite(DIRA, HIGH); //one way
  digitalWrite(DIRB, LOW);
  delay(3000);
  digitalWrite(ENABLE, LOW); //slow stop
  delay(1000);
  digitalWrite(ENABLE, HIGH); //enable on
  digitalWrite(DIRA, LOW); //one way
  digitalWrite(DIRB, HIGH);
  delay(3000);
  digitalWrite(DIRA, LOW); //fast stop
```

Super Starter Kit

```
delay(2000);

Serial.println("PWM full then slow");
//---PWM example, full speed then slow
analogWrite(ENABLE, 255); //enable on
digitalWrite(DIRA, HIGH); //one way
digitalWrite(DIRB, LOW);
delay(2000);
analogWrite(ENABLE, 180); //half speed
delay(2000);
analogWrite(ENABLE, 128); //half speed
delay(2000);
analogWrite(ENABLE, 50); //half speed
delay(2000);
analogWrite(ENABLE, 128); //half speed
delay(2000);
analogWrite(ENABLE, 180); //half speed
delay(2000);
analogWrite(ENABLE, 255); //half speed
delay(2000);
digitalWrite(ENABLE, LOW); //all done
delay(10000);
}
```

Laden Sie diesen Code auf Ihr Mikrocontroller-Board und öffnen Sie den Serial Monitor.

Relais

In diesem Beispiel lernen Sie, wie man ein Relais verwendet.

Für dieses Beispiel benötigen Sie:

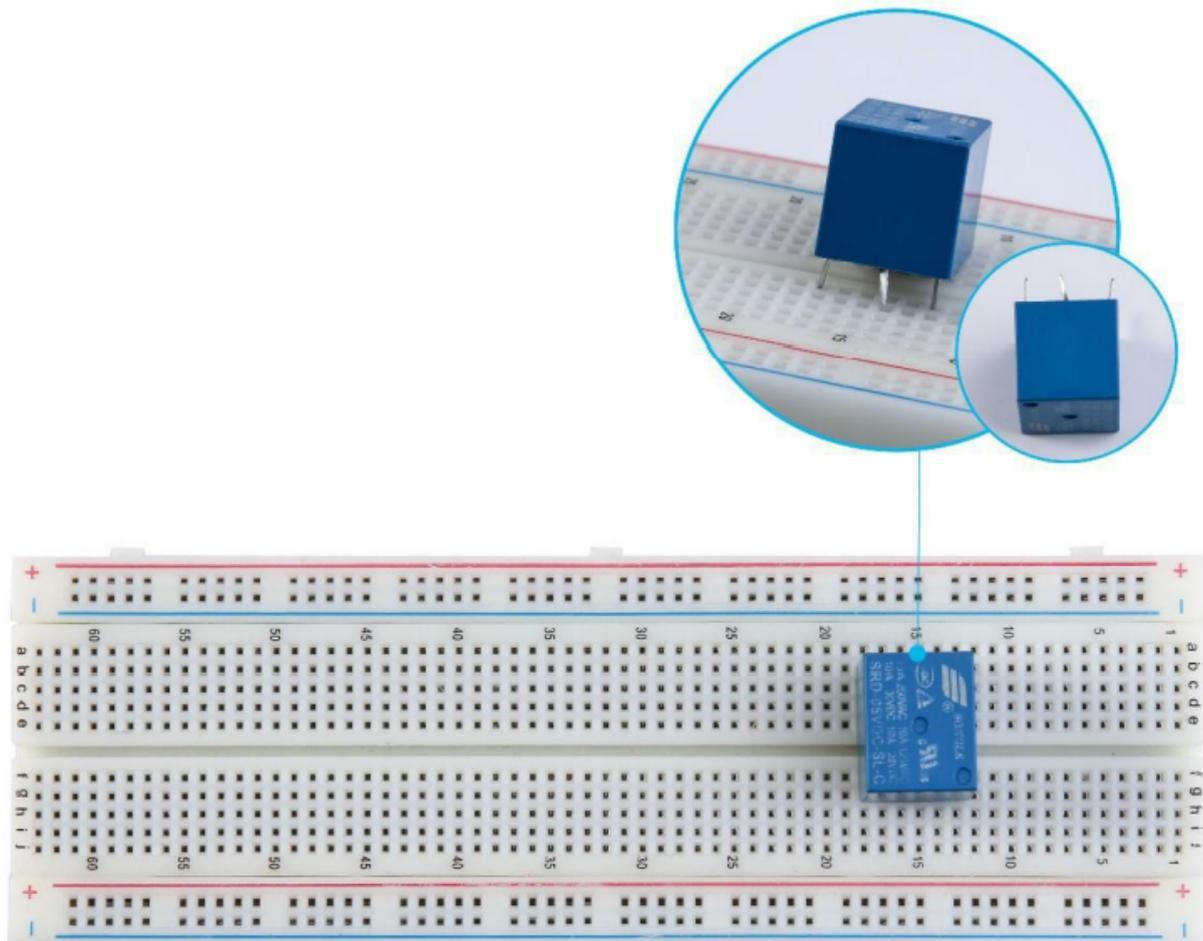
- Mikrocontroller-Platine
- 830-poliges Breadboard
- MB102 Breadboard Spannungsversorgung
- Lüfterflügel und Gleichstrommotor
- L293D IC
- Relais
- 8pcs M-M Drähte (männlich zu männlich Jumper Drähte)
- 9V 1A Netzadapter

Ein Relais ist ein elektrisch betriebener Schalter. Viele Relais verwenden einen Elektromagneten, um einen Schalter mechanisch zu betätigen, aber es werden auch andere Funktionsprinzipien wie bei Halbleiterrelais verwendet. Relais werden dort eingesetzt, wo ein Schaltkreis durch ein Signal mit geringer Leistung gesteuert werden muss (mit vollständiger elektrischer Isolierung zwischen Steuer- und gesteuertem Schaltkreis) oder wo mehrere Schaltkreise durch ein Signal gesteuert werden müssen. Die ersten Relais wurden in Langstreckentelegrafenschaltungen als Verstärker eingesetzt. Sie wiederholten das von einem Stromkreis eingehende Signal und gaben es an einen anderen Stromkreis weiter. Relais wurden häufig in Telefonzentralen und frühen Computern eingesetzt, um logische Operationen durchzuführen. Eine Art von Relais, das die hohe Leistung bewältigen kann, die zur direkten Steuerung eines Elektromotors oder anderer Lasten erforderlich ist, wird als Schütz bezeichnet. Halbleiterrelais steuern Stromkreise ohne bewegliche Teile und verwenden stattdessen einen Halbleiterbaustein, der das Schalten übernimmt. Relais mit kalibrierter Betriebscharakteristik und manchmal mehreren Betriebsspulen werden zum Schutz von Stromkreisen vor Überlast oder Fehlern eingesetzt. In modernen Stromversorgungssystemen werden diese Funktionen von digitalen Geräten, den so genannten "Schutzrelais", übernommen. Nachfolgend finden

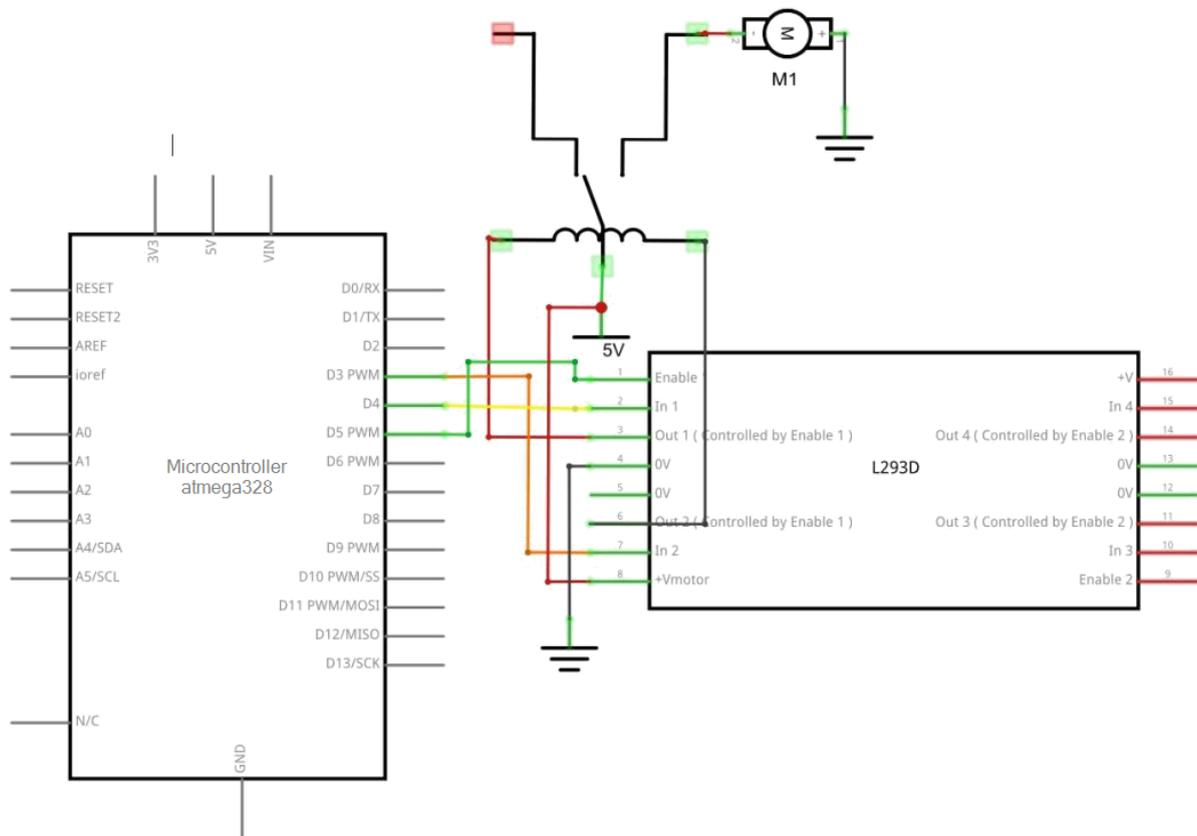
Super Starter Kit

Sie eine schematische Darstellung der Ansteuerung eines Relais mit einem Mikrocontroller.

Sie sind vielleicht verwirrt darüber, wie Sie das Relais in das Breadboard einfügen. Wie das Bild unten zeigt, müssen Sie einen der Stifte des Relais leicht biegen, dann können Sie es in das Breadboard einsetzen.



Schematische Darstellung



Beispiel-Code

```
#define ENABLE 5
#define DIRA 3
#define DIRB 4

void setup() {
  //---set pin direction
  pinMode(ENABLE, OUTPUT);
  pinMode(DIRA, OUTPUT);
  pinMode(DIRB, OUTPUT);
  Serial.begin(9600);
}

void loop() {

  //---back and forth example
  Serial.println("One way, then reverse");
  digitalWrite(ENABLE, HIGH); // enable on
  for (int i = 0; i < 5; i++) {
    digitalWrite(DIRA, HIGH); //one way
    digitalWrite(DIRB, LOW);
    delay(750);
    digitalWrite(DIRA, LOW); //reverse
    digitalWrite(DIRB, HIGH);
    delay(750);
  }
  digitalWrite(ENABLE, LOW); // disable
  delay(3000);
  for (int i = 0; i < 5; i++) {
    digitalWrite(DIRA, HIGH); //one way
    digitalWrite(DIRB, LOW);
    delay(750);
    digitalWrite(DIRA, LOW); //reverse
    digitalWrite(DIRB, HIGH);
    delay(750);
  }
  digitalWrite(ENABLE, LOW); // disable
  delay(3000);
}
```

Super Starter Kit

Laden Sie diesen Code in Ihren Mikrocontroller und öffnen Sie den seriellen Monitor.

Vergessen Sie nicht, den Schalter an der Breadboard-Stromversorgung einzuschalten

Schrittmotor

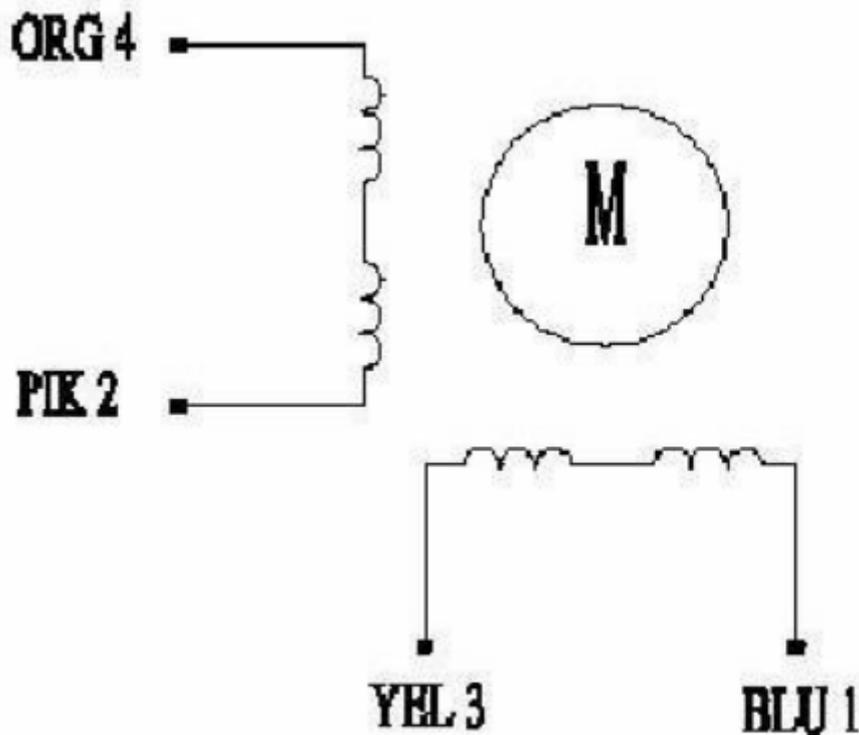
In diesem Beispiel lernst du, wie du auf einfache Weise einen Schrittmotor ansteuern kannst. Der Schrittmotor, den wir verwenden, wird mit einer eigenen Treiberplatine geliefert, die den Anschluss an das Mikrocontroller-Board erleichtert.

Für dieses Beispiel benötigst du:

- Mikrocontroller-Platine
- 830-poliges Breadboard
- ULN2003-Schrittmotor-Treiberplatine
- Schrittmotor
- MB102 Breadboard Spannungsversorgung
- 9V 1A Netzadapter
- 6pcs F-M Drähte (weiblich zu männlich Dupont Drähte)
- 1 Stück M-M-Draht (Stecker auf Stecker)

Ein Schrittmotor ist ein elektromechanisches Gerät, das elektrische Impulse in diskrete mechanische Bewegungen umwandelt. Die Welle oder Spindel eines Schrittmotors dreht sich in diskreten Schritten, wenn elektrische Steuerimpulse in der richtigen Reihenfolge an sie angelegt werden. Die Drehung des Motors steht in mehreren direkten Beziehungen zu diesen Eingangsimpulsen. Die Reihenfolge der angelegten Impulse steht in direktem Zusammenhang mit der Drehrichtung der Motorwelle. Die Drehzahl der Motorwelle steht in direktem Zusammenhang mit der Frequenz der Eingangsimpulse und die Länge der Drehung steht in direktem Zusammenhang mit der Anzahl der angelegten Eingangsimpulse. Einer der wichtigsten Vorteile eines Schrittmotors ist seine Fähigkeit, in einem offenen Regelkreis genau gesteuert zu werden. Offener Regelkreis bedeutet, dass keine Rückkopplungsinformationen über die Position benötigt werden. Bei dieser Art der Steuerung sind keine teuren Mess- und Rückmeldegeräte wie optische Encoder erforderlich. Ihre Position ist einfach durch die Verfolgung der Eingangsschrittimpulse bekannt.

Super Starter Kit



Der bipolare Schrittmotor hat in der Regel vier Drähte, die von ihm ausgehen. Im Gegensatz zu unipolaren Schrittmotoren haben bipolare Schrittmotoren keinen gemeinsamen Mittelanschluss. Stattdessen haben sie zwei unabhängige Spulensätze. Sie können sie von unipolaren Schrittmotoren unterscheiden, indem Sie den Widerstand zwischen den Drähten messen. Sie sollten zwei Adernpaare mit gleichem Widerstand vorfinden. Wenn Sie die Leitungen Ihres Messgeräts an zwei Drähte angeschlossen haben, die nicht miteinander verbunden sind (d. h. nicht an dieselbe Spule angeschlossen sind), sollten Sie einen unendlichen Widerstand (oder keinen Durchgang) feststellen.

Die einfachste Art, einen unipolaren Stepper mit einem Mikrocontroller zu verbinden, ist die Verwendung eines Breakouts für einen ULN2003A Transistor-Array-Chip. Der ULN2003A enthält sieben Darlington-Transistortreiber und ist in etwa so, als hätte man sieben TIP120-Transistoren in einem einzigen Gehäuse. Der ULN2003A kann bis zu 500 mA pro Kanal durchlassen und hat einen internen Spannungsabfall von etwa 1 V, wenn er eingeschaltet ist. Außerdem enthält er interne Klemmdioden, um Spannungsspitzen beim Ansteuern induktiver Lasten abzuleiten.

Super Starter Kit

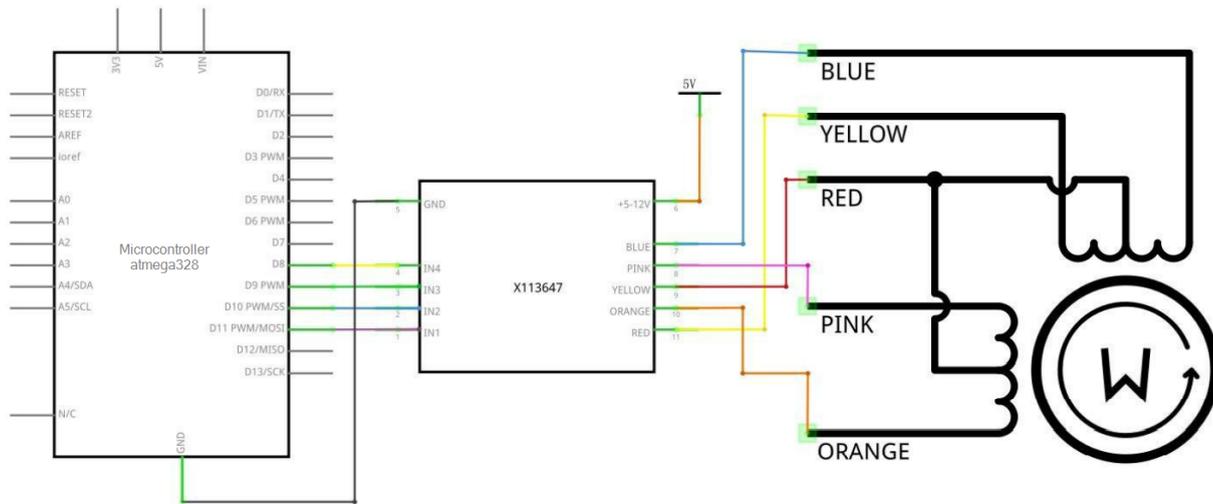
Zur Steuerung des Steppers muss an jede der Spulen in einer bestimmten Reihenfolge eine Spannung angelegt werden.

Die Sequenz würde wie folgt aussehen:

Lead Wire Color	CW Direction							
	1	2	3	4	5	6	7	8
4 ORG	x	x						
3 YEL		x	x	x				
2 PIK				x	x	x		
1 BLU						x	x	x

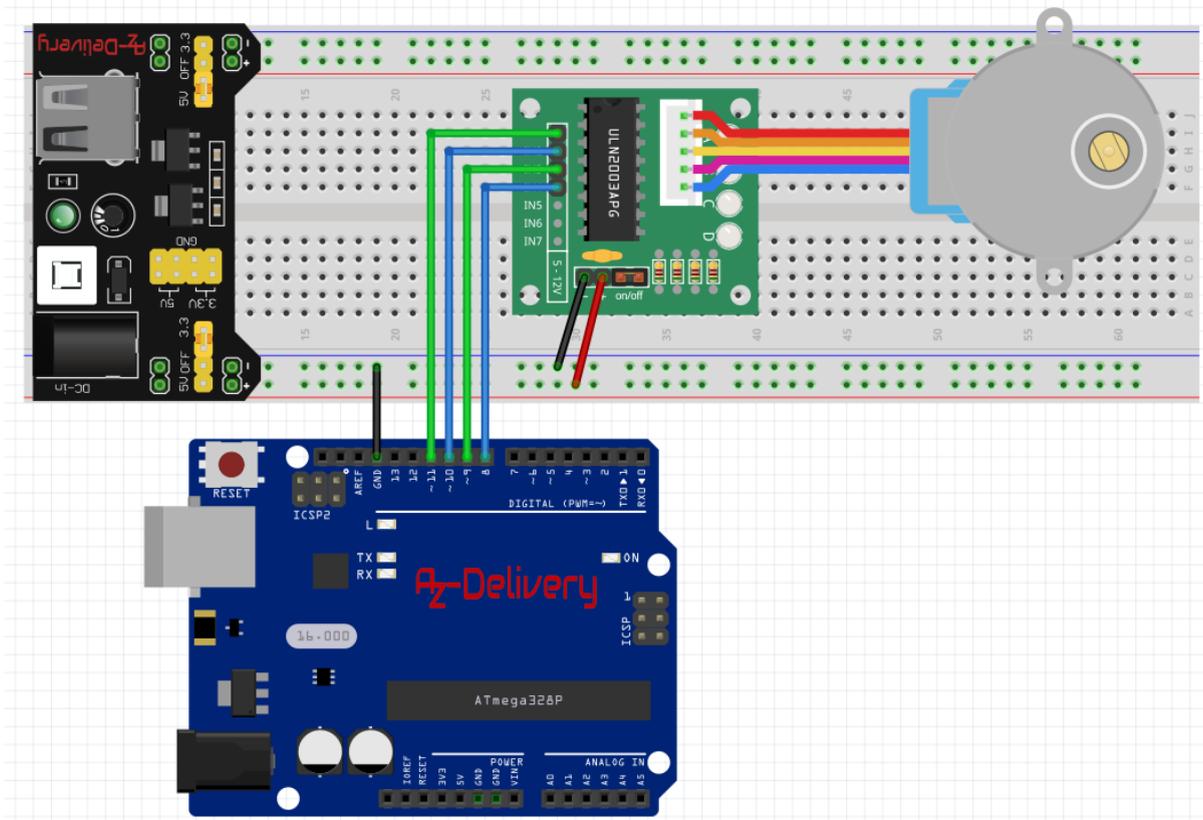
Super Starter Kit

Schematische Darstellung



Super Starter Kit

Schaltplan



Super Starter Kit

Beispiel-Code

Zuerst müssen Sie die "Stepper"-Bibliothek von Arduino über den Arduino Library Manager installieren.

```
#include <Stepper.h>

const int stepsPerRevolution = 1500; // change this to
fit the number of steps per revolution

// initialize the stepper library on pins 8 through 11:
Stepper myStepper(stepsPerRevolution, 8, 10, 9, 11);

void setup() {
  // set the speed at 20 rpm:
  myStepper.setSpeed(20);
  // initialize the serial port:
  Serial.begin(9600);
}

void loop() {
  // step one revolution in one direction:
  Serial.println("clockwise");
  myStepper.step(stepsPerRevolution);
  delay(500);

  // step one revolution in the other direction:
  Serial.println("counterclockwise");
  myStepper.step(-stepsPerRevolution);
  delay(500);
}
```

Laden Sie diesen Code auf Ihren Mikrocontroller.

Steuerung eines Schrittmotors mit einer Fernbedienung

In diesem Beispiel lernen Sie, wie Sie einen Schrittmotor mit einer IR-Fernbedienung aus der Ferne steuern können. Der Schrittmotor, den wir verwenden, wird mit einer eigenen Treiberplatine geliefert, die den Anschluss an unseren Mikrocontroller erleichtert.

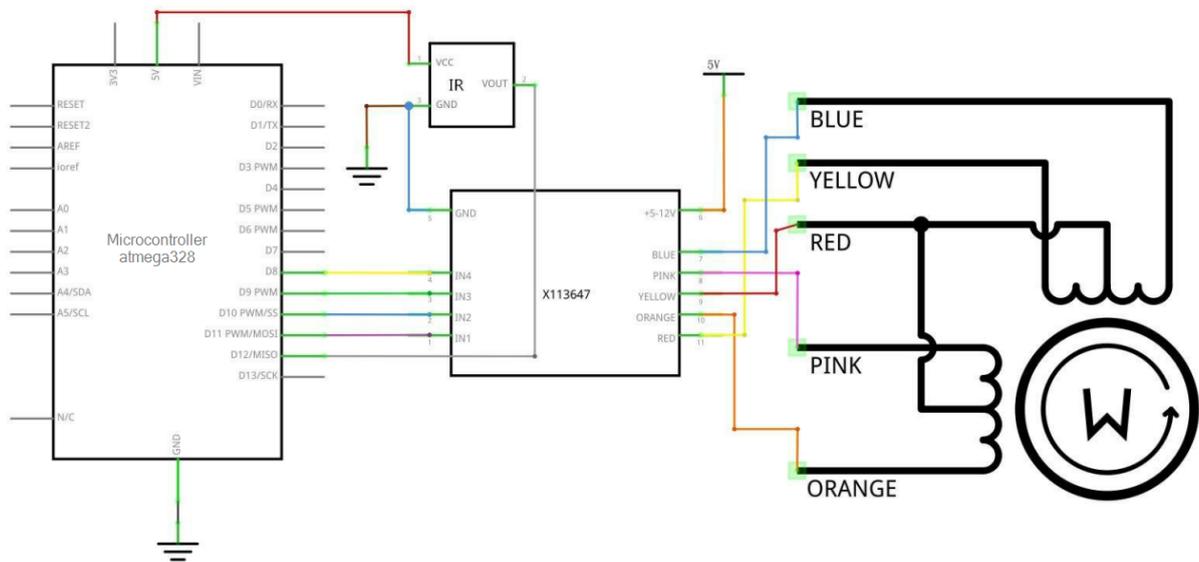
Da wir den Motor nicht direkt von der MCU aus ansteuern wollen, werden wir ein preiswertes kleines Breadboard-Netzteil verwenden, das direkt in unser Breadboard eingesteckt und mit einer 9V 1Amp-Stromversorgung betrieben wird. Der IR-Sensor wird direkt an den Mikrocontroller angeschlossen, da er fast keinen Strom verbraucht.

Für dieses Beispiel benötigen Sie:

- Mikrocontroller-Platine
- 830-polige Lochrasterplatine
- IR-Empfängermodul
- IR-Fernbedienung (Sie benötigen außerdem eine Knopfzellenbatterie)
- ULN2003-Schrittmotor-Treibermodul
- Schrittmotor
- MB102 Breadboard Spannungsversorgung
- 9V 1A Adapter
- 9 Stück F-M Drähte (weiblich zu männlich Überbrückungsdrähte)
- 1 Stck. M-M-Drähte (Stecker-zu-Stecker-Steckbrücken)

Super Starter Kit

Schematische Darstellung



Beispiel-Code

```
#include "Stepper.h"
#include "IRremote.h"

/*----- Variables, Pins -----*/
#define UP 0xFFA857 //change the hex code to match your button
#define DOWN 0xFF629D //change the hex code to match your button

#define STEPS 32 // Number of steps per revolution of Internal
shaft
int Steps2Take; // 2048 = 1 Revolution
int receiver = 12; // Signal Pin of IR receiver to Arduino Digital
Pin 6

/*-----( Declare objects )-----*/
// Setup of proper sequencing for Motor Driver Pins
// In1, In2, In3, In4 in the sequence 1-3-2-4

Stepper small_stepper(STEPS, 8, 10, 9, 11);
IRrecv irrecv(receiver); // create instance of 'irrecv'
decode_results results; // create instance of 'decode_results'

void setup()
{
  irrecv.enableIRIn(); // Start the receiver
}

void loop()
{
  if (irrecv.decode(&results)) // have we received an IR signal?
  {
    switch (results.value)
    {
      case UP: // up button pressed
        small_stepper.setSpeed(500); //Max seems to be 500
        Steps2Take = 2048; // Rotate CW
        small_stepper.step(Steps2Take);
    }
  }
}
```

Super Starter Kit

```
    delay(2000);
    break;

    case DOWN: // down button pressed
        small_stepper.setSpeed(500);
        Steps2Take = -2048; // Rotate CCW
        small_stepper.step(Steps2Take);
        delay(2000);
        break;

}

irrecv.resume(); // receive the next value
digitalWrite(8, LOW);
digitalWrite(9, LOW);
digitalWrite(10, LOW);
digitalWrite(11, LOW);
}

}/* --end main loop -- */
```

Bevor Sie diesen Code hochladen, müssen Sie auf die Zeilen 5 und 6 achten, in denen die HEX-Codes der Tasten auf der Fernbedienung definiert sind. Wenn Sie Schwierigkeiten haben, den richtigen HEX-Code zu finden, können Sie das Beispiel "IR-Empfänger-Modul" ausprobieren.