



**Das Vollständigste STARTER KIT
TUTORIAL FÜR MEGA2560**

V1.0.17.7.9

Vorwort

Unser Unternehmen

Elegoo wurde im Jahr 2011 gegründet und ist ein stark wachsendes Technologie-Unternehmen. Wir sind spezialisiert auf Open-Source-Hardware für Forschung und Entwicklung, Produktion und Marketing.

In Shenzhen, dem Silicon Valley von China, sind wir auf über 150 Mitarbeiter mit einer über 10.763 Quadratmeter großen Fabrik gewachsen.

Unsere Produktbreite reicht von DuPont Drähten, 2560 R3 Boards, mit Starter Kits für Kunden mit jedem Wissensstand um sie in die Welt der Arduinos einzuführen.

Darüber hinaus verkaufen wir auch Zubehör-Produkte für Raspberry Pi wie einen 2,8" TFT Touch und STM32. In Zukunft werden wir mehr Energie und Investitionen in 3D-Druckerprodukte einsetzen. Alle unsere Produkte entsprechen internationalen Qualitätsstandards und werden in einer Vielzahl von verschiedener Märkten auf der ganzen Welt sehr geschätzt.

Offizielle Website: <http://www.elegoo.com>

US Amazon storefront: <http://www.amazon.com/shops/A2WWHQ25ENKVJ1>

CA Amazon storefront: <http://www.amazon.ca/shops/A2WWHQ25ENKVJ1>

UK Amazon storefront: <http://www.amazon.co.uk/shops/AZF7WYXU5ZANW>

DE Amazon storefront: <http://www.amazon.de/shops/AZF7WYXU5ZANW>

FR Amazon storefront: <http://www.amazon.fr/shops/AZF7WYXU5ZANW>

ES Amazon storefront: <http://www.amazon.es/shops/AZF7WYXU5ZANW>

IT Amazon storefront: <http://www.amazon.it/shops/AZF7WYXU5ZANW>

Unser Tutorial

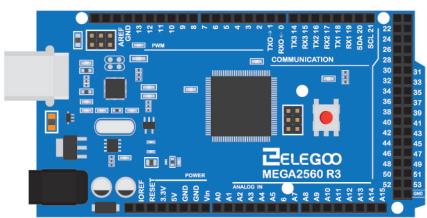
Dieses Tutorial ist für Anfänger konzipiert. Sie lernen alle grundlegenden Kenntnisse über die Verwendung des Arduino Controller Boards, Sensoren und weiteren Komponenten. Wenn Sie tiefer in die Arduinowelt einsteigen möchten, empfehlen wir Ihnen, das von Michael Margolis geschriebene Buch "Arduino Kochbuch" zu lesen. Einige Codes in diesem Tutorial wurden von Simon Monk erstellt. Simon Monk ist Autor einer Reihe von Büchern über Open Source Hardware. Sie sind in Amazon erhältlich: "Programmierung Arduino", "30 Arduino Selbstbau-Projekte" und "Programmierung des Raspberry Pi".

Kundenservice

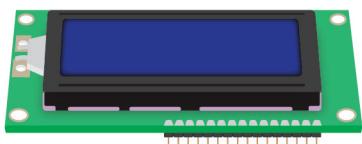
Als kontinuierliches und schnell wachsendes Technologieunternehmen streben wir unser Bestes an, Ihnen ausgezeichnete Produkte und qualitativ hochwertigen Service anzubieten, um Ihre Erwartungen zu erfüllen und Sie können uns einfach erreichen, indem Sie uns einfach unter service@elegoo.com oder EUservice@elegoo.com eine Nachricht zukommen lassen. Wir freuen uns darauf, von Ihnen zu hören und jeder Ihrer kritischen Kommentare oder Vorschläge sind für uns sehrwertvoll.

Und alle Probleme und Fragen, die Sie mit unseren Produkten haben, werden umgehend von unseren erfahrenen Ingenieuren innerhalb von 12 Stunden (24 Stunden in der Urlaubszeit) beantwortet.

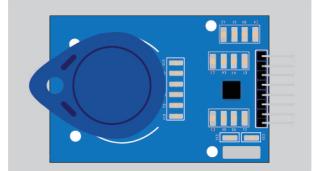
Packing list



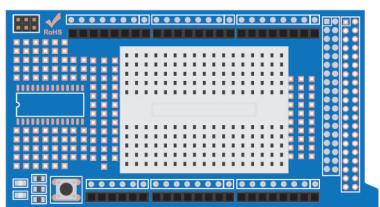
MEGA 2560 Controller Board
1PC



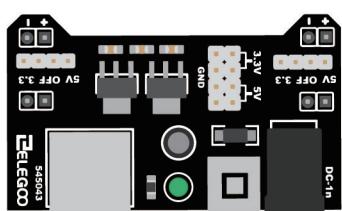
LCD1602 Module
(with pin header)
1PC



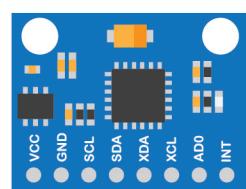
RC522 RFID Module
1PC



Prototype Expansion Module
1PC



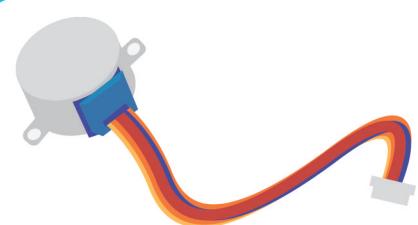
Power Supply Module
1PC



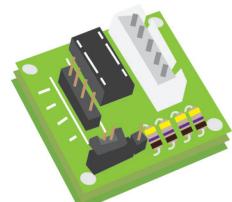
GY-521 Module
1PC



Servo Motor SG90
1PC

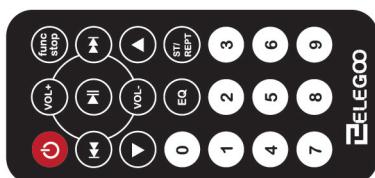


Stepper Motor
1PC

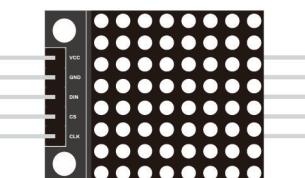


ULN2003 Stepper Motor
Driver Module
1PC

Contact us : service@elegoo.com



Remote Control
1PC



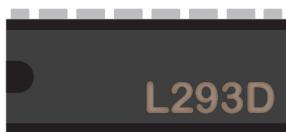
MAX7219 Module
1PC



1 Digit 7-Segment Display
1PC



4 Digit 7-Segment Display
1PC



L293D
1PC



74HC595 IC
1PC



Active Buzzer
1PC

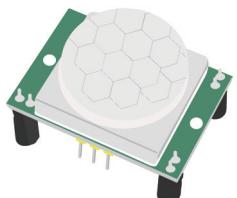


Passive Buzzer
1PC

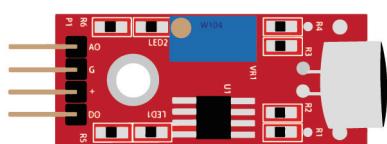


Potentiometer 10K
2PCS

Contact us : service@elegoo.com



**HC-SR501 PIR Motion
Sensor Module
1PC**



**Sound Sensor Module
1PC**



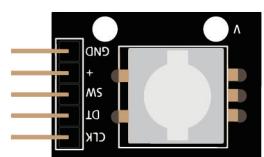
**Water Level Detection
Sensor Module
1PC**



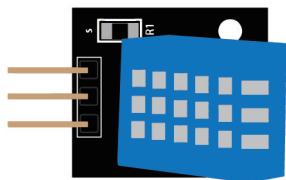
**Ultrasonic Sensor
1PC**



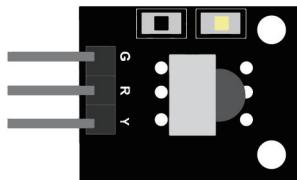
**DS1307 RTC Module
1PC**



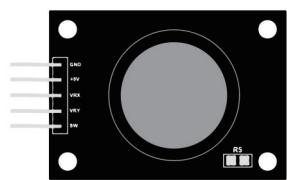
**Rotary Encoder Module
1PC**



**DHT11 Temperature and
Humidity Module
1PC**

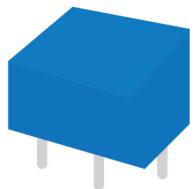


**IR Receiver Module
1PC**

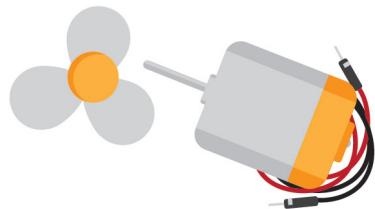


**Joystick Module
1PC**

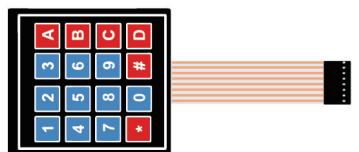
Contact us : service@elegoo.com



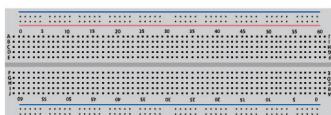
5V Relay
1PC



Fan Blade and 3-6V Motor
1PC



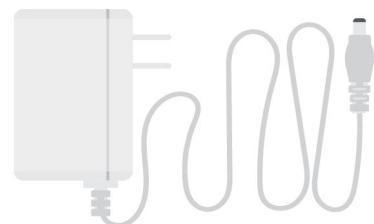
Membrane Switch Module
1PC



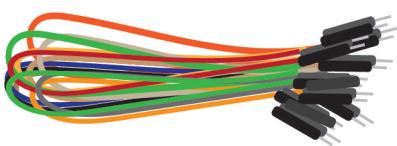
830 Tie-Points Breadboard
1PC



9V Battery with Snap-on
Connector Clip
1PC



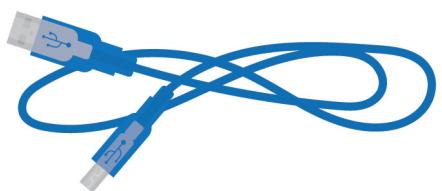
9V1A Adapter
1PC



Breadboard Jumper Wire
65PCS

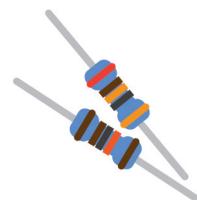


Female-to-Male Dupont Wire
20PCS



USB Cable
1PC

Contact us : service@elegoo.com



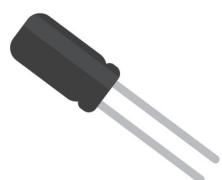
Resistor
120PCS



Thermistor
1PC



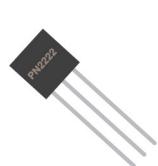
Diode Rectifier
5PCS



100uF Electrolytic Capacitor
2PCS



10uF Electrolytic Capacitor
2PCS



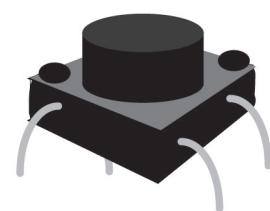
NPN Transistor PN2222
5PCS



NPN Transistor S8050
5PCS



Tilt Ball Switch
1PC



Button
5PCS

Contact us : service@elegoo.com



Red LED
5PCS



Yellow LED
5PCS



Blue LED
5PCS



Green LED
5PCS



White LED
5PCS



RGB LED
2PCS



104pF Ceramic Capacitor
5PCS



22pF Ceramic Capacitor
5PCS



Photoresistor(Photocell)
2PCS

Contact us : service@elegoo.com

Inhaltsverzeichnis

| | |
|--|-----|
| Lektion 0 Installieren der IDE..... | 12 |
| Lektion 1 Bibliotheken hinzufügen und den seriellen Monitor öffnen | 23 |
| Lektion 2 Blinken | 32 |
| Lektion 3 LED | 43 |
| Lektion 4 RGB LED | 50 |
| Lektion 5 Digitale Eingänge | 59 |
| Lektion 6 Aktiver Summer | 64 |
| Lektion 7 Passiver Summer..... | 68 |
| Lektion 8 Kippschalter | 72 |
| Lektion 9 Servo | 76 |
| Lektion 10 Ultraschallsensormodul..... | 80 |
| Lektion 11 Membranschalter-Modul | 85 |
| Lektion 12 DHT11 Temperatur- und Feuchtigkeitssensor..... | 91 |
| Lektion 13 Analoger Joystickmodul..... | 97 |
| Lektion 14 IR Empfangsmodul | 102 |
| Lektion 15 MAX7219 LED Dot Matrix Modul..... | 108 |
| Lektion 16 GY-521 Modul | 113 |
| Lektion 17 HC-SR501 PIR Sensor..... | 122 |
| Lektion 18 Wasserstandserkennungssensormodul..... | 132 |
| Lektion 19 Echtzeituhrmodul | 137 |
| Lektion 20 Schallsensor-Modul | 142 |
| Lektion 21 RC522 RFID Modul | 148 |
| Lektion 22 LCD Display | 153 |
| Lektion 23 Thermometer..... | 158 |
| Lektion 24 Acht LEDs mit 74HC595..... | 163 |
| Lektion 25 Der serielle Monitor..... | 170 |
| Lektion 26 Photozelle | 176 |
| Lektion 27 74HC595 und Segmentanzeige | 181 |

| | |
|--|-----|
| Lektion 28 Vier Sieben-Segment-Anzeigen | 187 |
| Lektion 29 DC Motor..... | 192 |
| Lektion 30 Relay..... | 202 |
| Lektion 31 Schrittmotor..... | 207 |
| Lektion 32 Steuern des Schrittmotors mit Fernbedienung | 215 |
| Lektion 33 Steuerung des Schrittmotors mit Drehgeber..... | 219 |

Lektion 0 Installieren der IDE

Einführung

Die Arduino IDE (Integrated Development Environment) ist die Software-Seite der Arduino-Plattform.

In dieser Lektion, werden Sie lernen, wie Sie Ihren Computer für den Arduino vorbereiten und für die folgenden Lektionen konfigurieren.

Die Arduino Software, die Sie verwenden, um Ihr Arduino zu programmieren, ist für Windows, Mac und Linux verfügbar. Der Installationsvorgang ist für alle drei Plattformen unterschiedlich und leider gibt es noch einige Schritte die Sie selbst tätigen müssen, um die Software zu installieren.

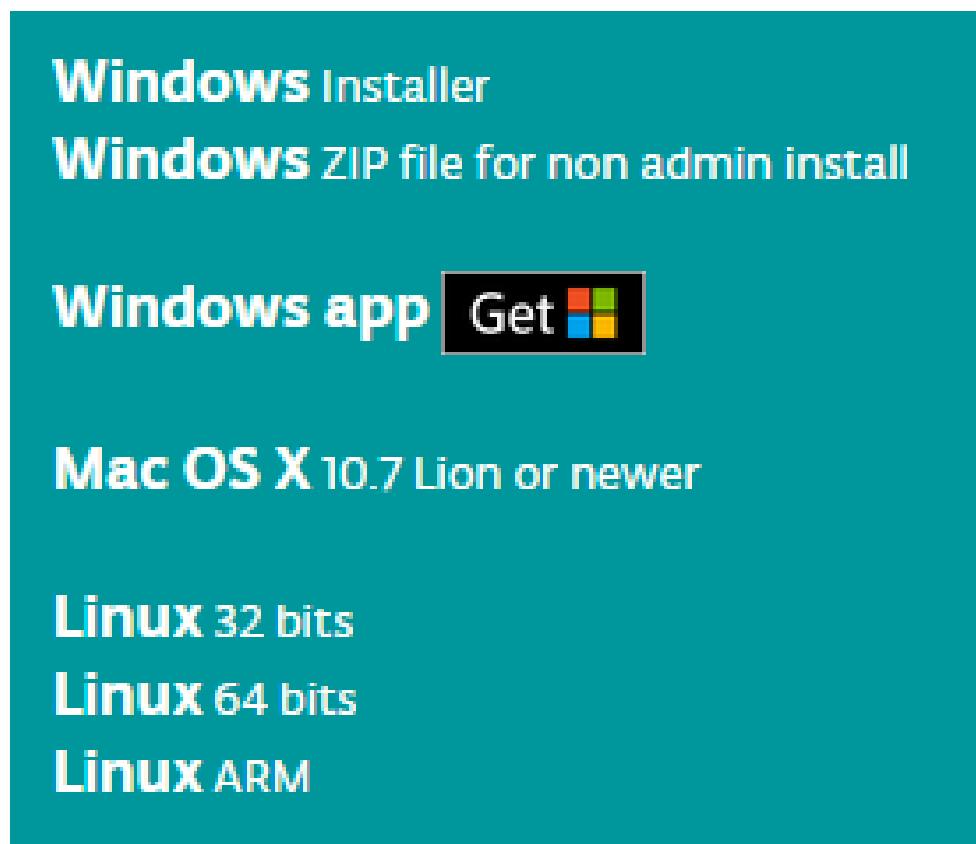
Schritt 1: Gehen Sie auf <https://www.arduino.cc/en/Main/Software> um zu folgender Seite zugelangen



The screenshot shows the Arduino Software (IDE) download page. On the left, there's a large teal circular logo with a white infinity symbol containing a minus sign on the left and a plus sign on the right. To the right of the logo, the text "ARDUINO 1.8.0" is displayed in bold capital letters. Below this, a paragraph of text describes the software: "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." On the far right, there's a vertical sidebar with links for different operating systems: "Windows Installer", "Windows ZIP file for non admin install", "Windows app" with a "Get" button, "Mac OS X 10.7 Lion or newer", "Linux 32 bits", "Linux 64 bits", "Linux ARM", "Release Notes", "Source Code", and "Checksums (sha512)".

Die auf dieser Website verfügbare Version ist in der Regel die neueste Version, und die aktuelle Version kann neuer sein als die Version in unserem Bild.

Schritt 2: Laden Sie die Entwicklungssoftware herunter, die mit dem Betriebssystem Ihres Computers kompatibel ist. Hier die Windowsversion als Beispiel.



Klicke *Windows Installer*.

Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). Learn more on how your contribution will be used.



Klicke *JUST DOWNLOAD*.

Die Version 1.8.0 ist auf der mitgelieferten CD, die wir zur Verfügung gestellt haben. Es ist die Version mit der wir dieses Tutorial erstellt haben.

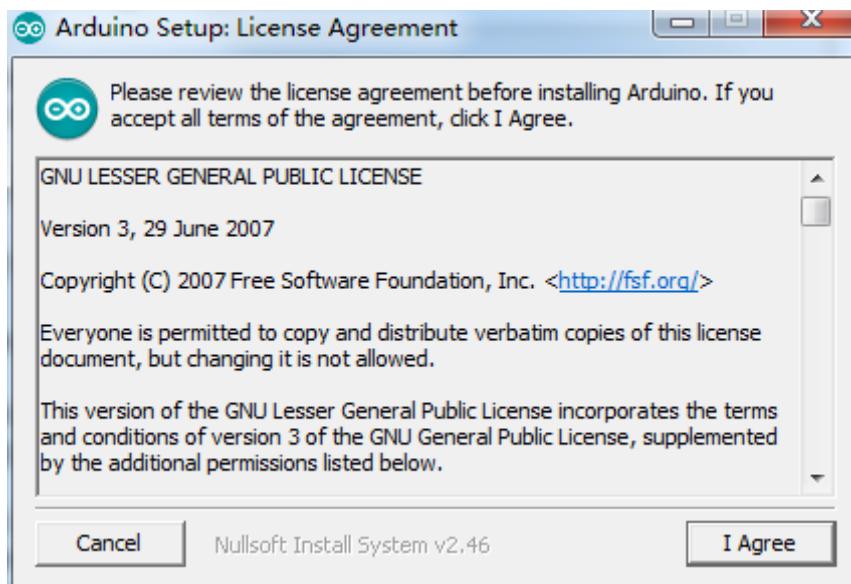
-  arduino-1.8.0-linux32.tar.xz
-  arduino-1.8.0-linux64.tar.xz
-  arduino-1.8.0-macosx.zip
-  arduino-1.8.0-windows.exe
-  arduino-1.8.0-windows.zip

Arduino installieren (Windows)

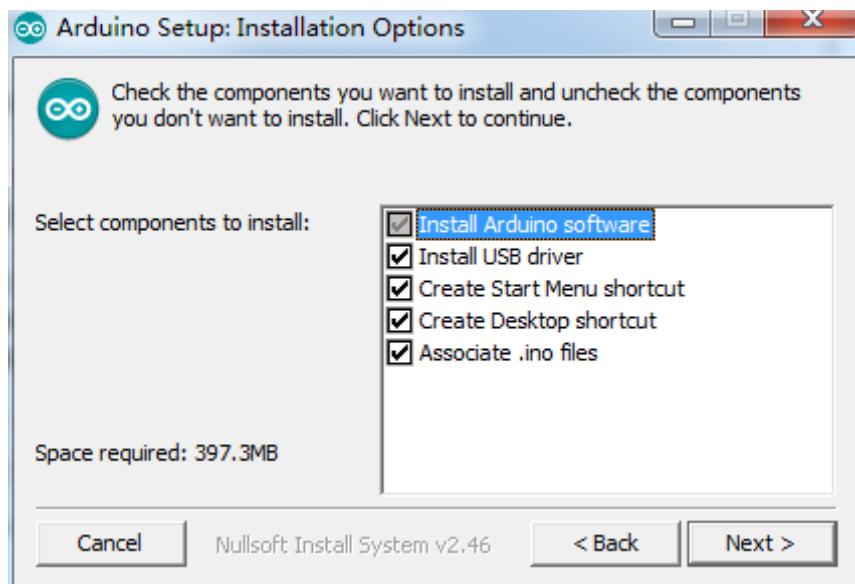
Installiere den Arduino mit der .exe Installationsdatei.



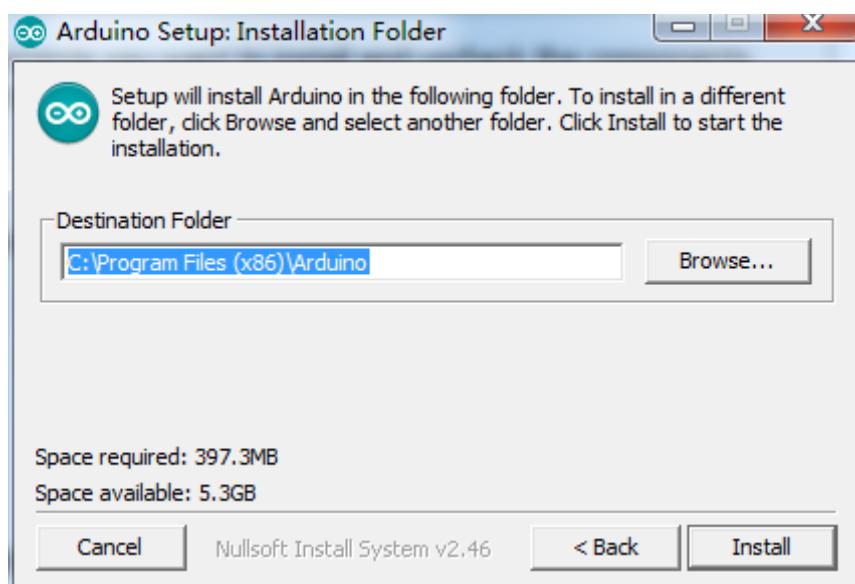
arduino-1.8.0-windows.exe



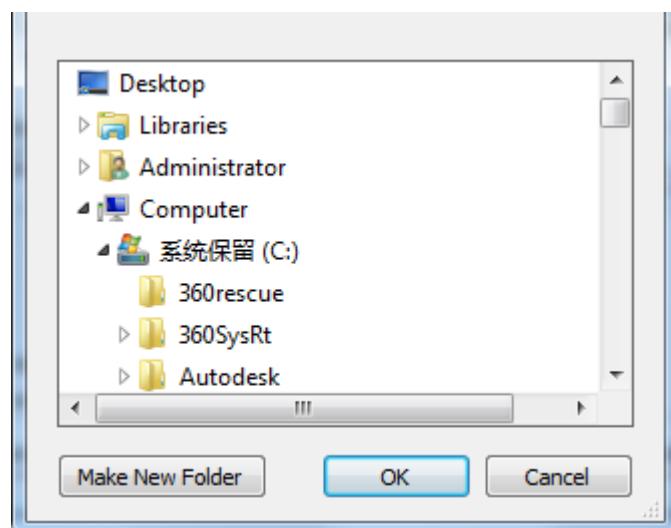
Klicke *I Agree* um die folgende Benutzeroberfläche zusehen.



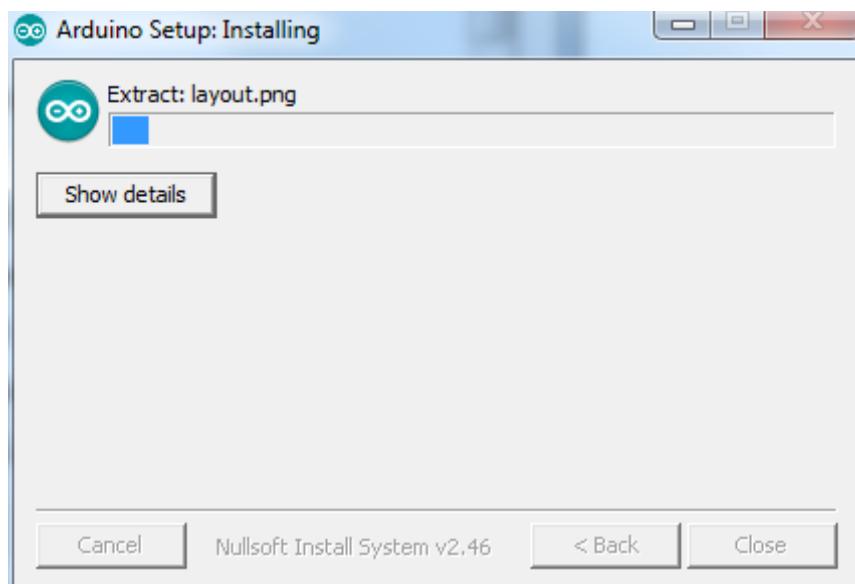
Klick Next



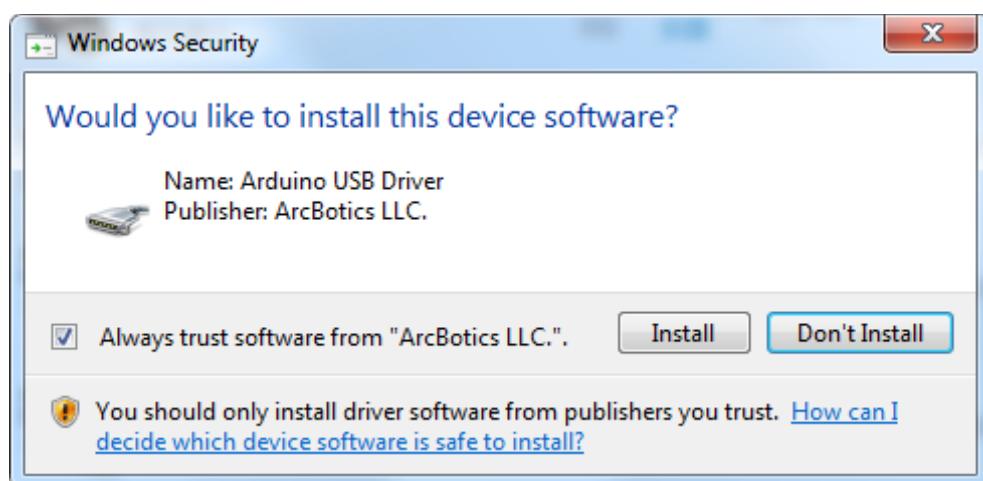
Drücken Sie "Browse" um einen von Ihnen gewählten Installationspfad auszuwählen.



Klicke *Install* um die Installation zu starten.



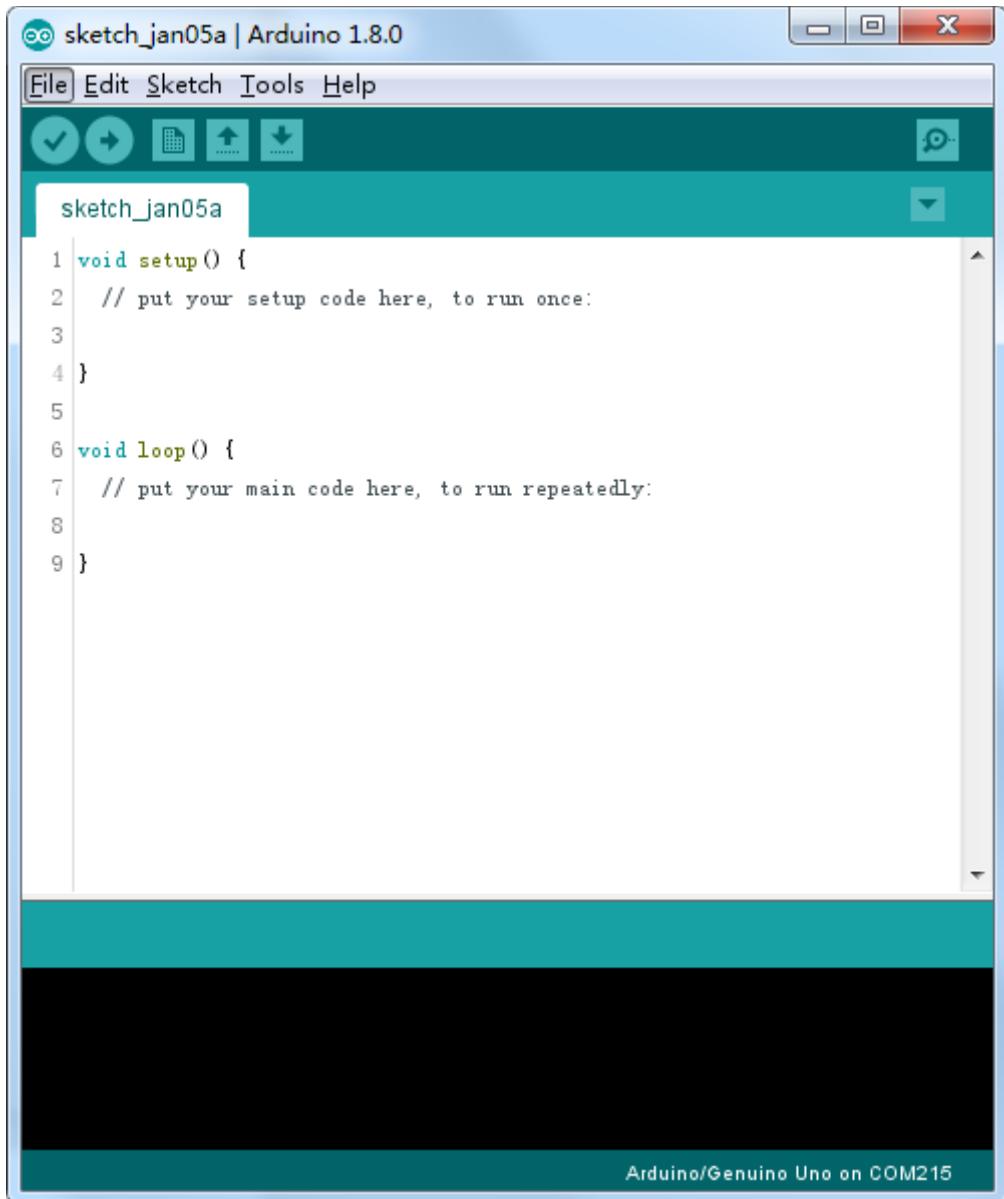
Schließlich erscheint die folgende Benutzeroberfläche, klicken Sie auf **Installieren**, um die Installation abzuschließen.



Als nächstes erscheint das folgende Symbol auf dem Desktop.



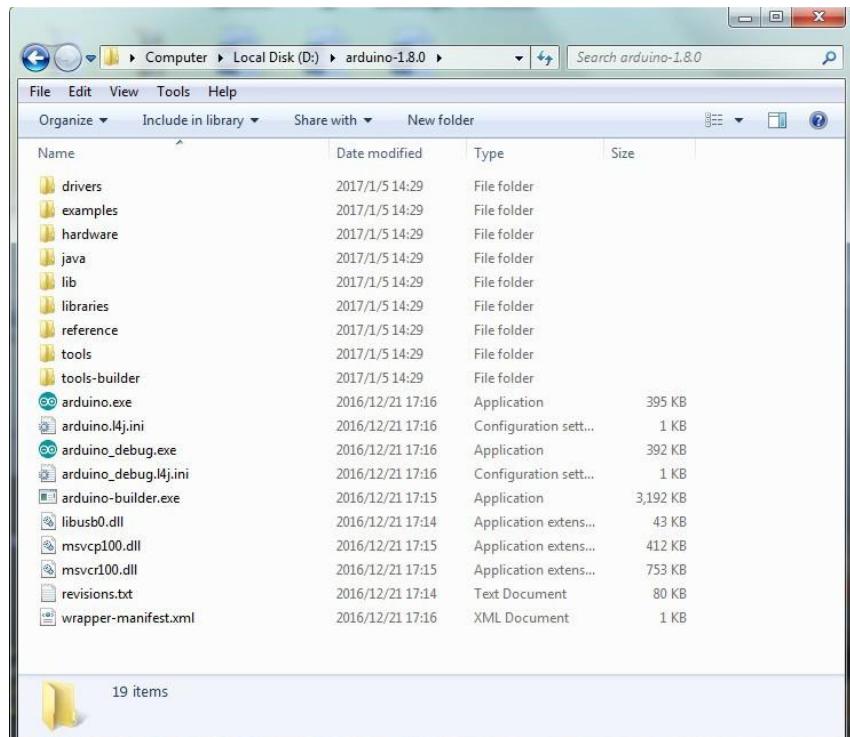
Mit einem Doppelklick öffnet sich die Entwicklungsumgebung.

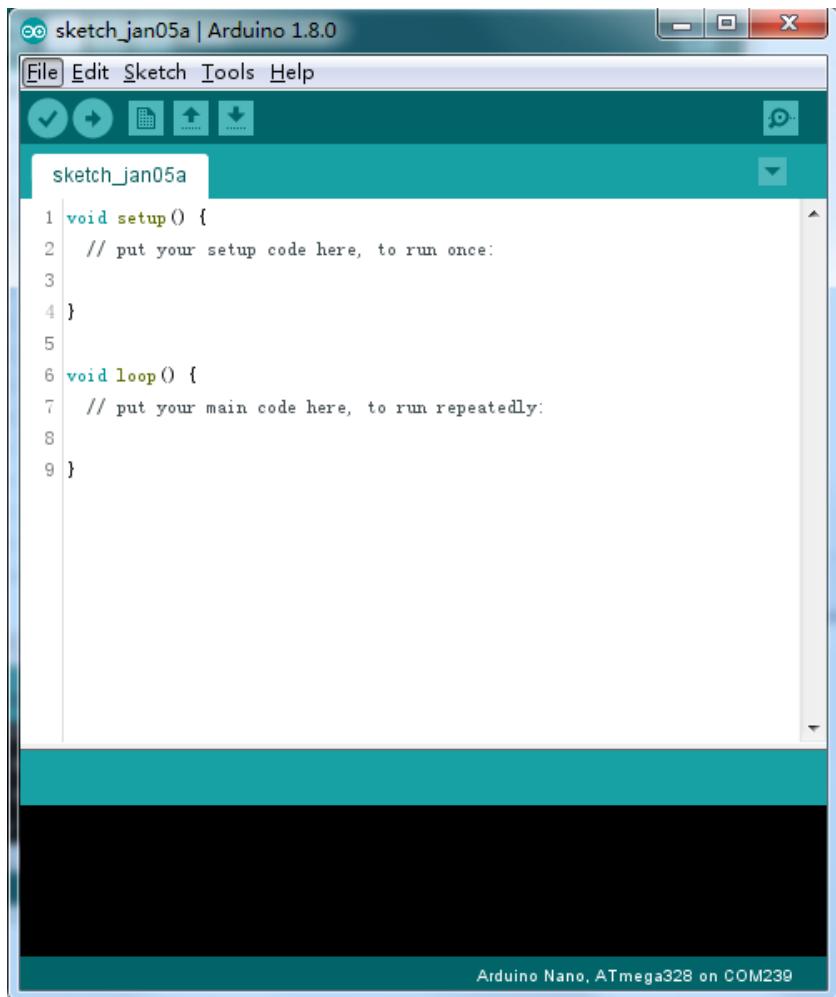


Sie können direkt das Installationspaket für die Installation auswählen und den Inhalt unten überspringen und zum nächsten Abschnitt springen. Aber wenn Sie einige Methoden mehr über das Installationspaket lernen möchten, lesen Sie bitte den Abschnitt weiter.

Entpacken Sie die heruntergeladene Zip-Datei, doppelklicken Sie, um das Programm zu öffnen und geben Sie die gewünschte Entwicklungsumgebung ein.







Diese Installationsmethode benötigt jedoch eine separate Installation des Treibers.

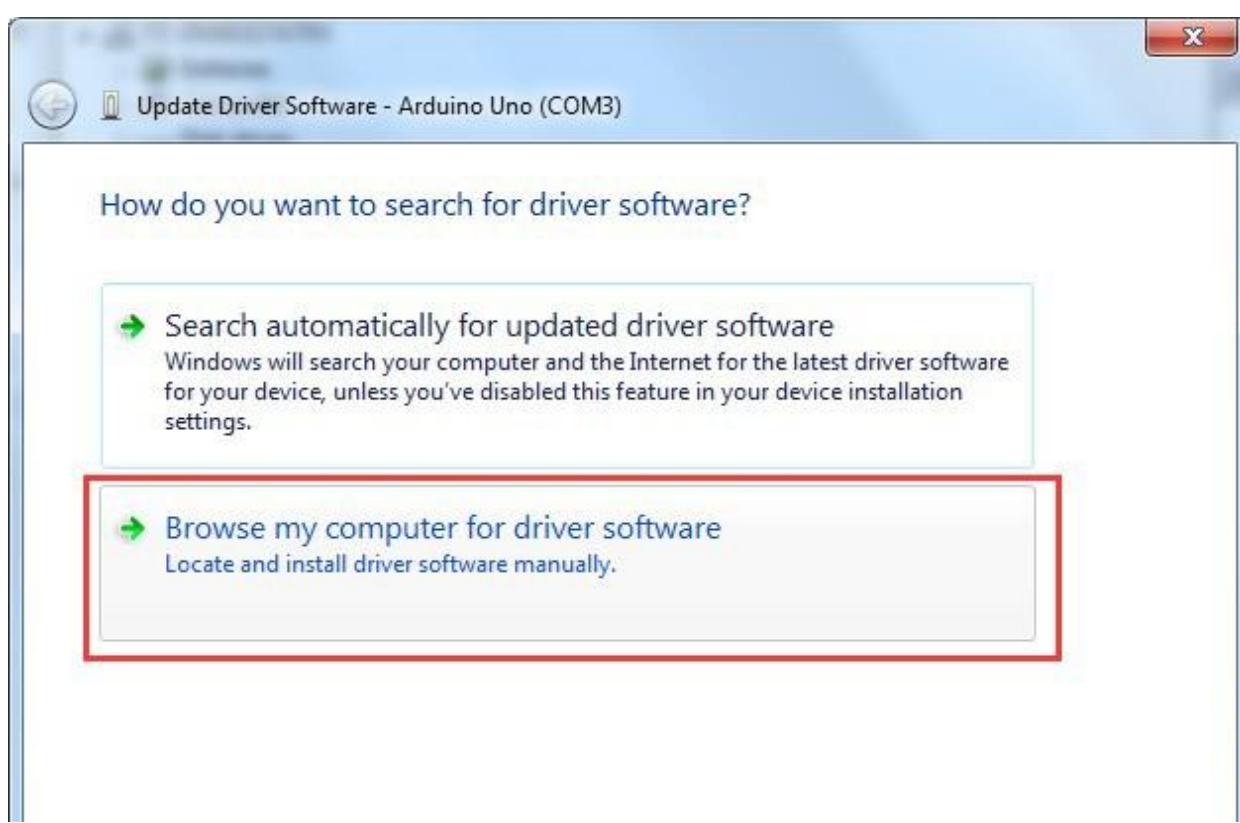
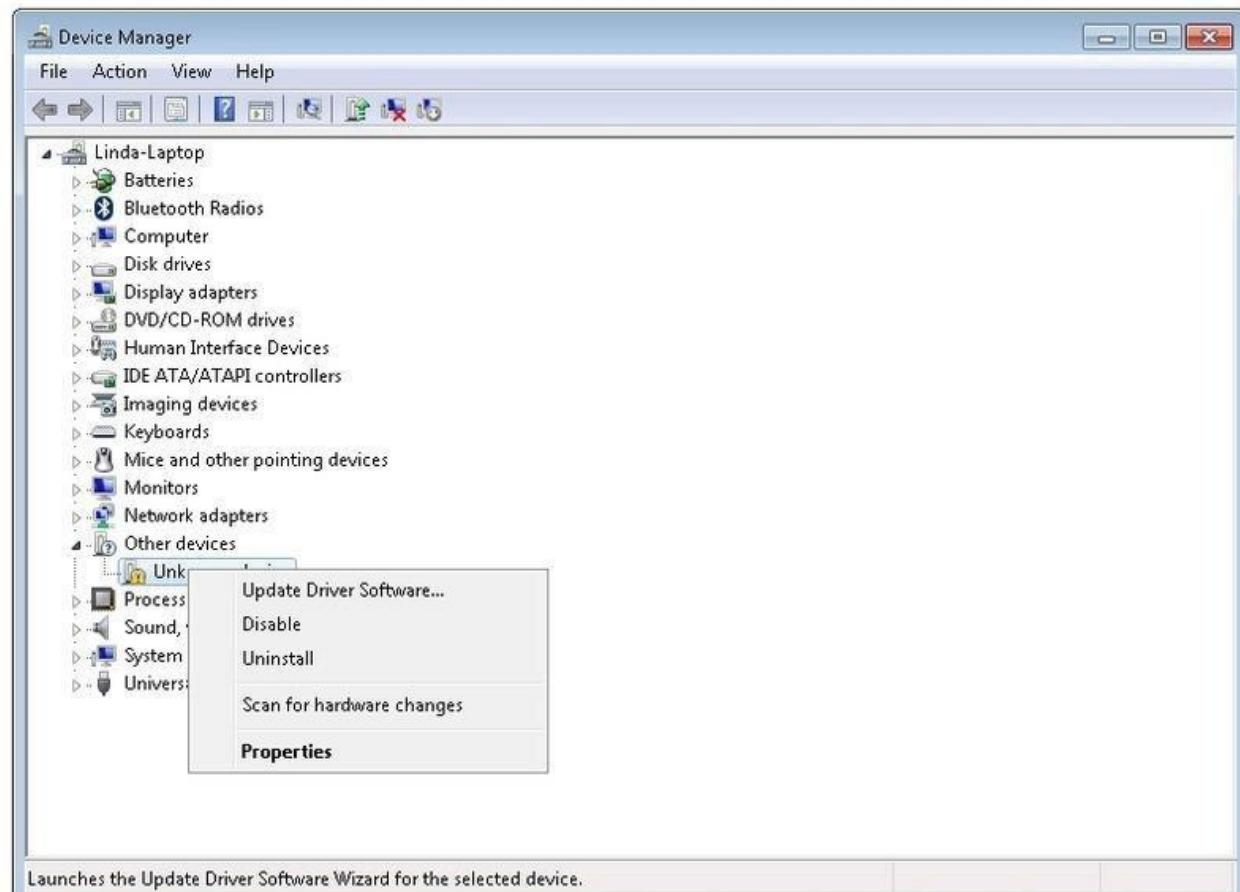
Der Arduino-Ordner enthält sowohl das Arduino-Programm selbst als auch die Treiber, die es dem Arduino ermöglichen, über ein USB-Kabel mit Ihrem Computer verbunden zu werden. Bevor wir die Arduino Software starten, werden Sie die USB Treiber installieren.

Stecken Sie ein Ende Ihres USB-Kabels in das Arduino und das andere in eine USB-Buchse an Ihrem Computer. Die Stromanzeige auf der LED leuchtet auf und Sie erhalten möglicherweise eine Nachricht 'Neue Hardware gefunden' von Windows. Ignorieren Sie diese Nachricht und beenden Sie alle Versuche, die Windows macht, um Treiber automatisch für Sie zu installieren.

Die zuverlässigste Methode zur Installation der USB-Treiber ist die Verwendung des Geräte-Managers. Dies wird je nach Ihrer Windows-Version auf unterschiedliche Weise aufgerufen. In Windows 7 müssen Sie zuerst die Systemsteuerung öffnen und dann die Option auswählen, "Icons anzeigen", und Sie sollten den Geräte-Manager

in der Liste finden.

Unter 'Andere Geräte' sollten Sie ein Symbol für 'unbekanntes Gerät' mit einem kleinen gelben Warndreieck daneben sehen. Das ist Ihr Arduino.



Klicken Sie mit der rechten Maustaste auf das Gerät und wählen Sie die obere Menüoption ("Treibersoftware aktualisieren") Sie werden dann aufgefordert, entweder "Automatisch nach aktueller Treibersoftware suchen" oder "Auf dem Computer nach Treibersoftware suchen". Wählen Sie die Option zum Durchsuchen und Navigieren zu X\arduino1.8.0\drivers.



Klicken Sie auf 'Weiter' und Sie erhalten eine Sicherheitswarnung und klicken dort auf "Ja", die Treiber werden nun installiert. Sobald die Software installiert ist, erhalten Sie eine Bestätigungs Nachricht.



Windows-Benutzer können die Installationsanleitung für Mac- und Linux-Systeme überspringen und zu Lektion 1 springen. Mac- und Linux-Benutzer können diesen Abschnitt weiterhin lesen.

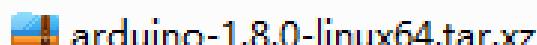
Arduino installieren (Mac OS X)

Downloaden und entpacken Sie die Zip-Datei, doppelklicken Sie auf die Arduino.app, um in die Arduino IDE zu gelangen. Das System wird Sie bitten, Java-Laufzeitbibliothek zu installieren, wenn Sie es nicht bereits auf Ihrem Mac installiert haben. Sobald die Installation abgeschlossen ist, können Sie die Arduino IDE ausführen.



Arduino installieren (Linux)

Sie müssen den Befehl *make install* verwenden. Wenn Sie das Ubuntu-System verwenden, empfiehlt es sich, Arduino IDE aus dem Software-Center von Ubuntu zu installieren.



TIPP: Wenn Sie Probleme bei der Installation der Treiber haben, wenden Sie sich bitte an die UNO R3, MEGA, NANO DRIVER FAQ.



Lektion 1 Bibliotheken hinzufügen und den seriellen Monitor

öffnen

Installieren zusätzlicher Arduino-Bibliotheken

Sobald Sie etwas mit der Arduino-Software vertraut sind und die eingebauten Funktionen nutzen können, können Sie den Funktionsumfang Ihres Arduino mit zusätzlichen Bibliotheken erweitern.

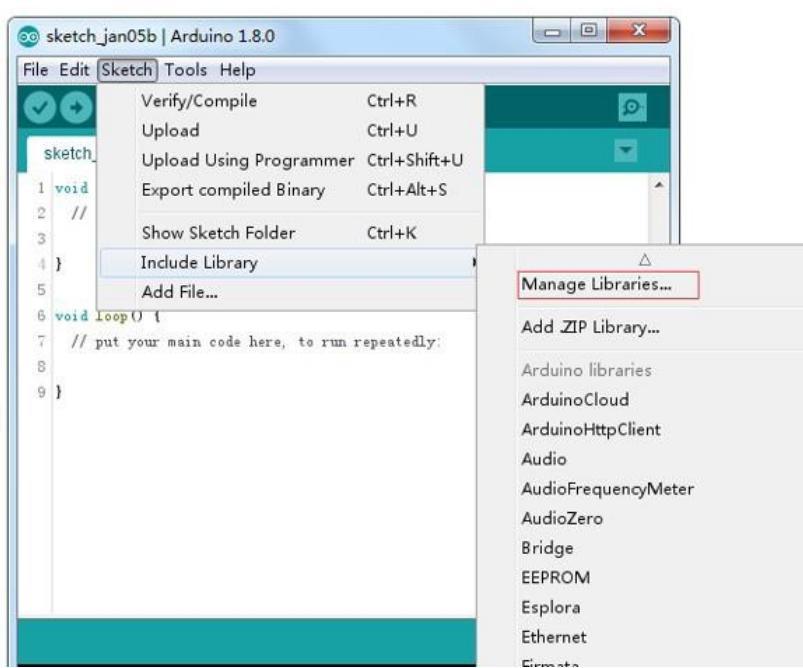
Was sind Bibliotheken?

Bibliotheken sind eine Sammlung von Code, der es Ihnen leicht macht, eine Verbindung zu einem Sensor, einer Anzeige, einem Modul usw. herzustellen. Zum Beispiel macht die integrierte "LiquidCrystal-Bibliothek" es einfach, mit LCD-Displays zu "kommunizieren". Es gibt Hunderte von zusätzlichen Bibliotheken im Internet zum Download zur Verfügung. Die eingebauten Bibliotheken und einige dieser zusätzlichen Bibliotheken sind in der Referenz aufgelistet. Um die zusätzlichen Bibliotheken zu benutzen, müssen Sie diese installieren.

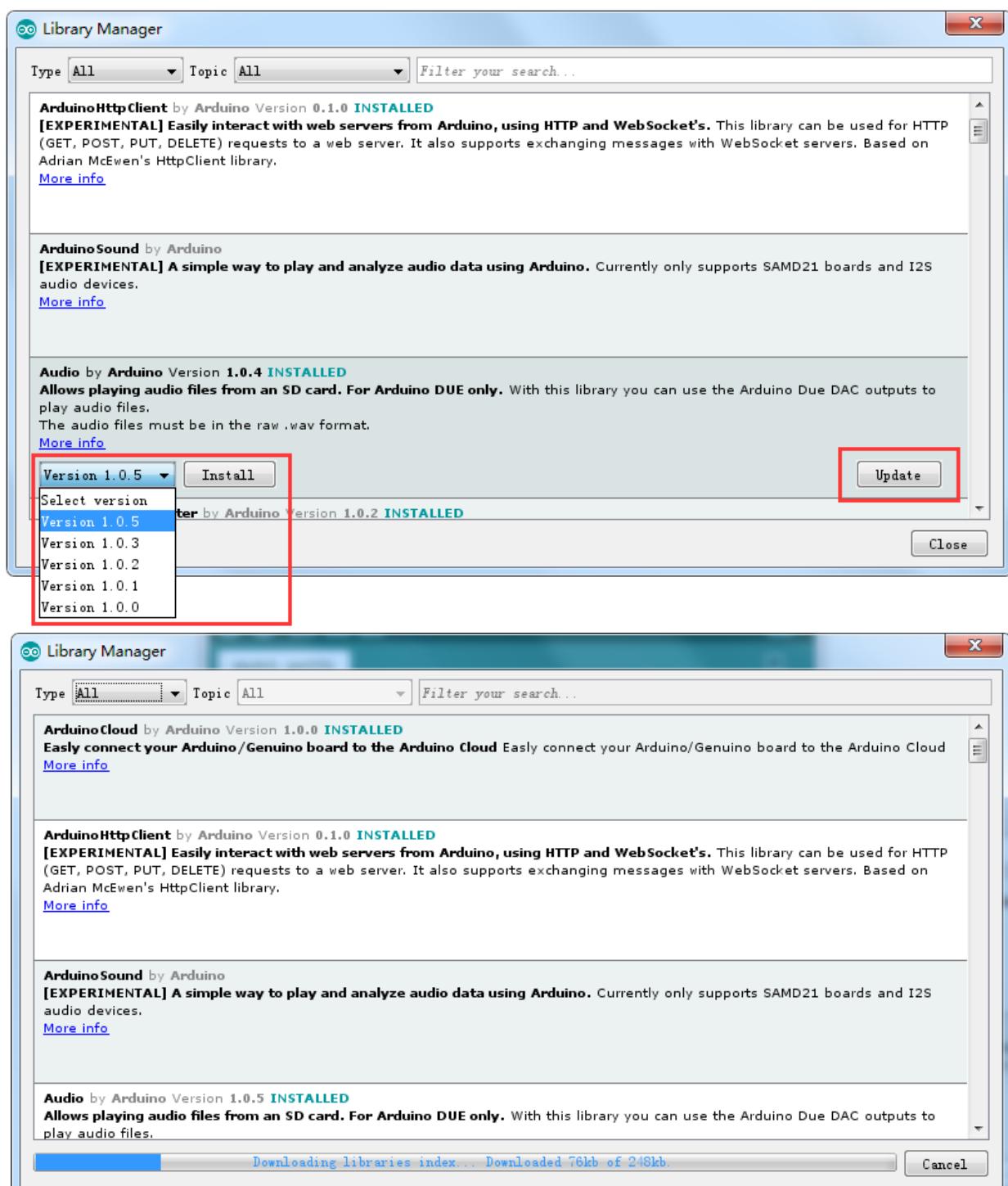
Wie installiere ich eine Bibliothek?

Durch verwenden des Bibliotheksmanagers.

Um eine neue Bibliothek in Ihre Arduino IDE zu installieren, können Sie den Bibliotheksmanager (verfügbar ab IDE Version 1.8.0) verwenden. Öffnen Sie die IDE und klicken Sie auf das "Sketch" -Menü und dann Include Library> Manage Libraries



Dann wird der Bibliotheksmanager geöffnet und Sie finden eine Liste der Bibliotheken, die bereits installiert oder bereit für die Installation sind. In diesem Beispiel installieren wir die Bridge-Bibliothek. Blättern Sie in der Liste, um sie zu finden, und wählen Sie dann die Version der Bibliothek aus, die Sie installieren möchten. Manchmal ist nur eine Version der Bibliothek verfügbar. Wenn das Versionsauswahlmenü nicht erscheint, machen Sie sich keine Sorgen, das ist normal. **Es gibt Zeiten, in denen du geduldig sein musst, genau wie in der Abbildung gezeigt.** Bitte aktualisieren Sie ("Update") und warten Sie.



Klicken Sie abschließend auf Installieren und warten Sie, bis die IDE die neue Bibliothek installiert hat. Das Herunterladen kann je nach Verbindungsgeschwindigkeit Zeit in Anspruch nehmen. Sobald es fertig ist, sollte neben der Bridge-Bibliothek ein "Installed-Tag" erscheinen. Sie können den Bibliotheksmanager nun schließen.

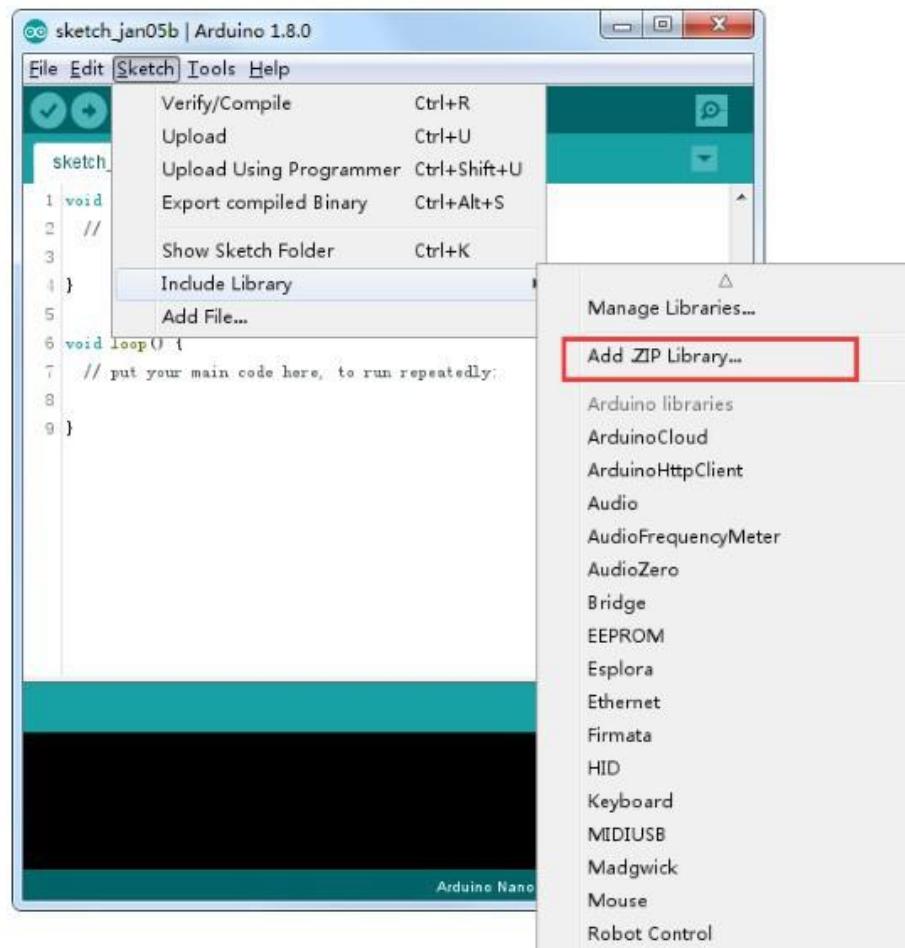


Sie finden nun die neue Bibliothek im Menü *Include Library*. Wenn Sie Ihre eigene Bibliothek hinzufügen möchten, erstellen Sie ein neues Problem auf Github.

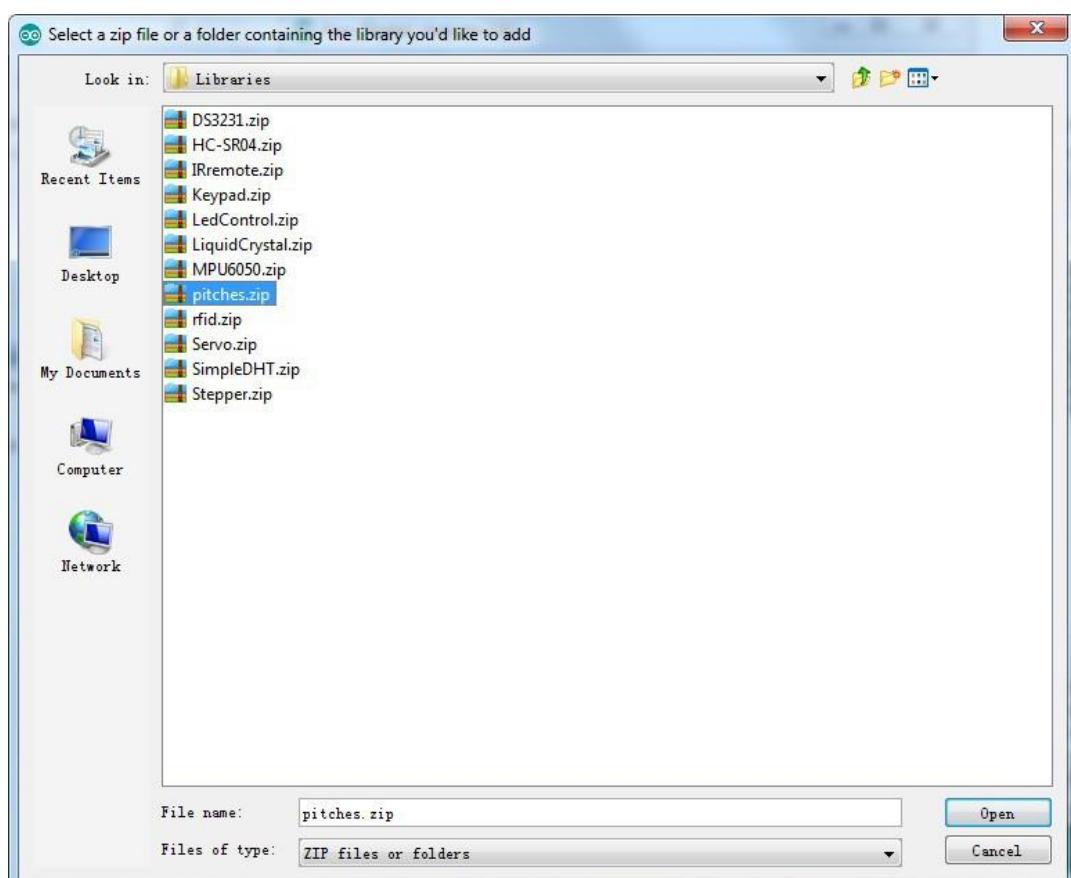
Importieren einer .zip Bibliothek

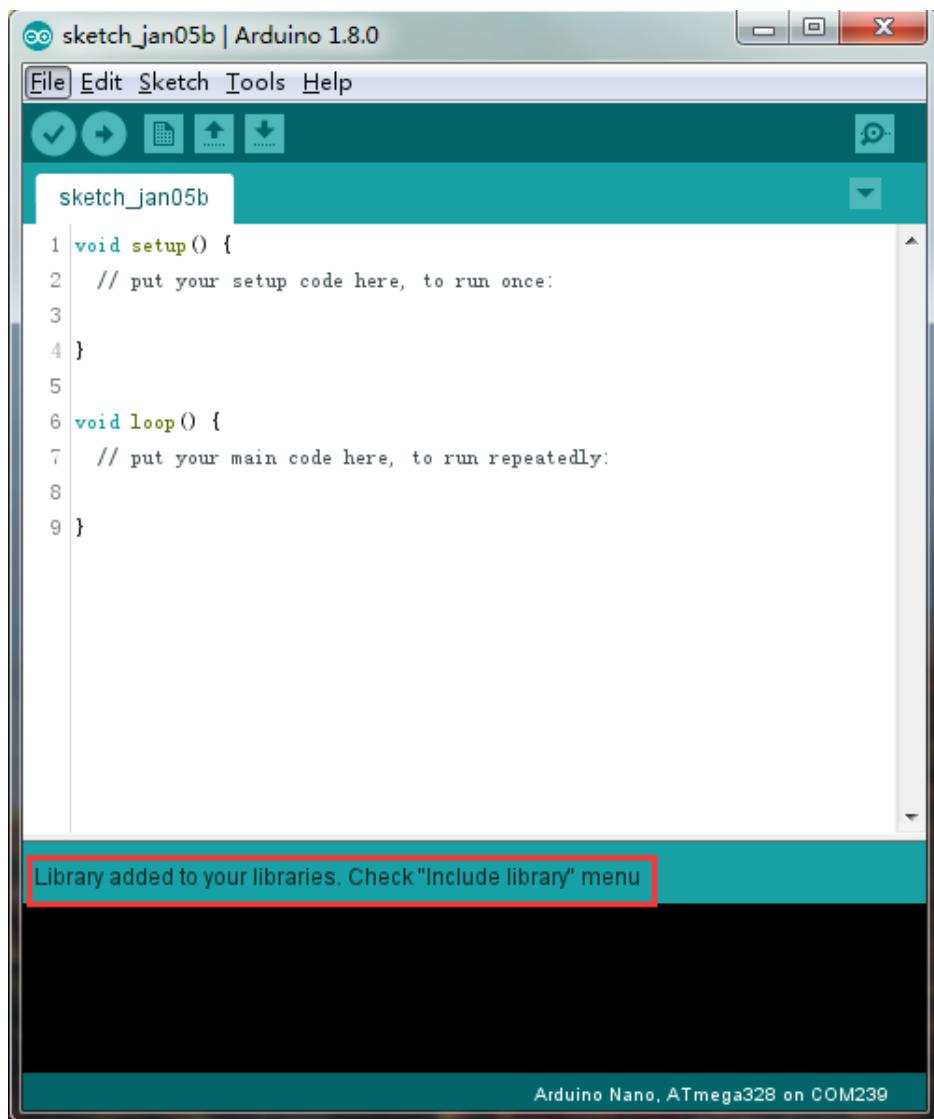
Bibliotheken werden häufig als ZIP-Datei oder Ordner verteilt. Der Name des Ordners ist der Name der Bibliothek. Innerhalb des Ordners sind eine .cpp-Datei, eine .h-Datei und oft eine keywords.txt-Datei, Beispiel-Ordner und andere Dateien, die von der Bibliothek benötigt werden. Seit Version 1.0.5 können Sie Drittanbieter-Bibliotheken in der IDE installieren. Entpacken Sie die heruntergeladene Bibliothek nicht, lassen Sie sie so wie sie ist.

In der Arduino IDE navigieren Sie zu Sketch> Include Library. Klicken Sie oben in der Dropdown-Liste auf die Option ".ZIP-Bibliothek hinzufügen".



Sie werden aufgefordert, die Bibliothek auszuwählen, die Sie hinzufügen möchten.
Navigieren Sie zum Speicherort der ZIP-Datei und öffnen Sie sie.





Gehen Sie zum Menü Sketch> Import Library zurück. Sie sollten nun die Bibliothek am unteren Rand des Dropdown-Menüs sehen. Sie steht Ihnen nun für die Programmierung zur Verfügung. Die Zip-Datei wurde im Ordner Bibliotheken im Arduino-Sketch-Verzeichnis erweitert.

NB: Die Bibliothek wird für ihre "sketche" zur Verfügung stehen, aber Beispiele für die Bibliothek werden in File>Examples erst nach dem Neustart der IDE zur Verfügung.

Diese beiden sind die häufigsten Ansätze. MAC- und Linux-Systeme können ebenfalls nach dem selben Verfahren vorgehen werden. Die manuelle Installation, die unten als Alternative aufgeführt wird, kann auch verwendet werden und Benutzer die diese automatische Funktion nutzen möchten, können es überspringen.

Manuelle installation

Um die Bibliothek zu installieren, beenden Sie zuerst die Arduino-Anwendung. Dann entpacken Sie die ZIP-Datei mit der Bibliothek. Zum Beispiel, wenn Sie eine Bibliothek namens "ArduinoParty" installieren, entpacken Sie ArduinoParty.zip. Es sollte einen Ordner namens ArduinoParty entstehen mit Dateien wie ArduinoParty.cpp und ArduinoParty.h. (Wenn die .cpp- und .h-Dateien nicht in einem Ordner sind, müssen Sie einen erstellen. In diesem Fall würden Sie einen Ordner namens "ArduinoParty" erstellen und alle Dateien, die sich im ZIP befinden, hineinkopieren, wie ArduinoParty.cpp und ArduinoParty.h.)

Ziehe den ArduinoParty-Ordner in diesen Ordner (Ihr Bibliotheksordner). Unter Windows wird es wahrscheinlich "Meine Dokumente \ Arduino \ Bibliotheken" genannt. Für Mac-Benutzer wird es wahrscheinlich "Dokumente / Arduino / Bibliotheken" genannt. Unter Linux wird es der Ordner "Bibliotheken" in Ihrem Skizzenbuch sein.

Ihr Arduino-Bibliotheksordner sollte nun so aussehen (unter Windows):

[Meine Dokumente\Arduino\libraries\ArduinoParty\ArduinoParty.cpp](#)

[Meine Dokumente \Arduino\libraries\ArduinoParty\ArduinoParty.h](#)

[Meine Dokumente\Arduino\libraries\ArduinoParty\examples](#)

or like this (on Mac and Linux):

[Dokumente/Arduino/libraries/ArduinoParty/ArduinoParty.cpp](#)

[Dokumente//Arduino/libraries/ArduinoParty/ArduinoParty.h](#)

[Dokumente//Arduino/libraries/ArduinoParty/examples](#)

....

Es gibt vielleicht noch mehr Dateien als nur die .cpp- und .h-Dateien, Seien Sie sich nur sicher, dass sie alle hinzugefügt haben. (Die Bibliothek funktioniert nicht, wenn Sie die .cpp- und .h-Dateien direkt in den Bibliotheksordner einfügen oder wenn sie in einem zusätzlichen Ordner verschachtelt sind. Beispiel:

Dokumente\Arduino\Bibliotheken\ArduinoParty.cpp

und

Dokumente\Arduino\Bibliotheken\ArduinoParty\ArduinoParty\ArduinoParty.cpp

wird nicht funktionieren.)

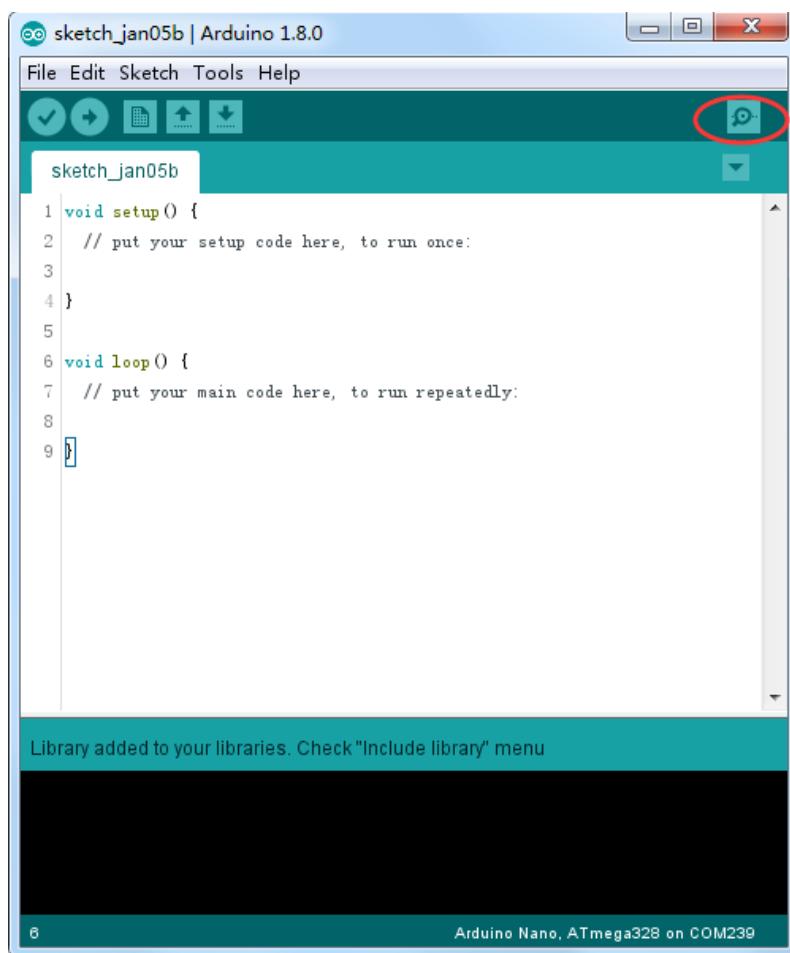
Starten Sie die Arduino-Anwendung neu. Vergewissern Sie sich, dass die neue Bibliothek im Menü Sketch-> Import Library der Software erscheint. Das ist es! Sie haben eine Bibliothek installiert!

Arduino Seriell Monitor (Windows, Mac, Linux)

Die Arduino IDE (Integrated Development Environment) ist die Software-Seite der Arduino-Plattform. Und weil die Verwendung des Terminals ein großer Teil der Arbeit mit Arduinos und anderen Mikrocontrollern ist, beschlossen die Entwickler, ein serielles Terminal mit der Software einzubinden. Innerhalb der Arduino-Umgebung heißt das Serieller Monitor.

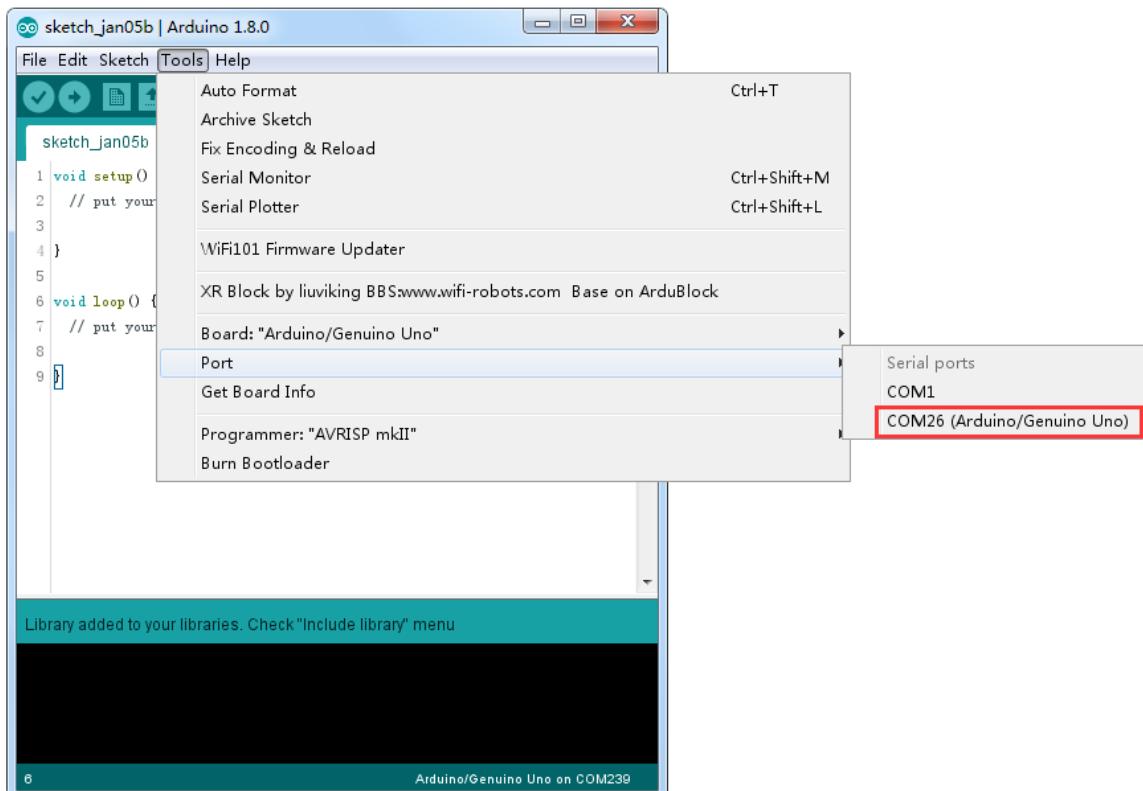
Eine Verbindung herstellen

Serieller Monitor ist Bestandteil jeder und allen Versionen der Arduino IDE. Um es zu öffnen, klicken Sie einfach auf das Symbol Serieller Monitor.

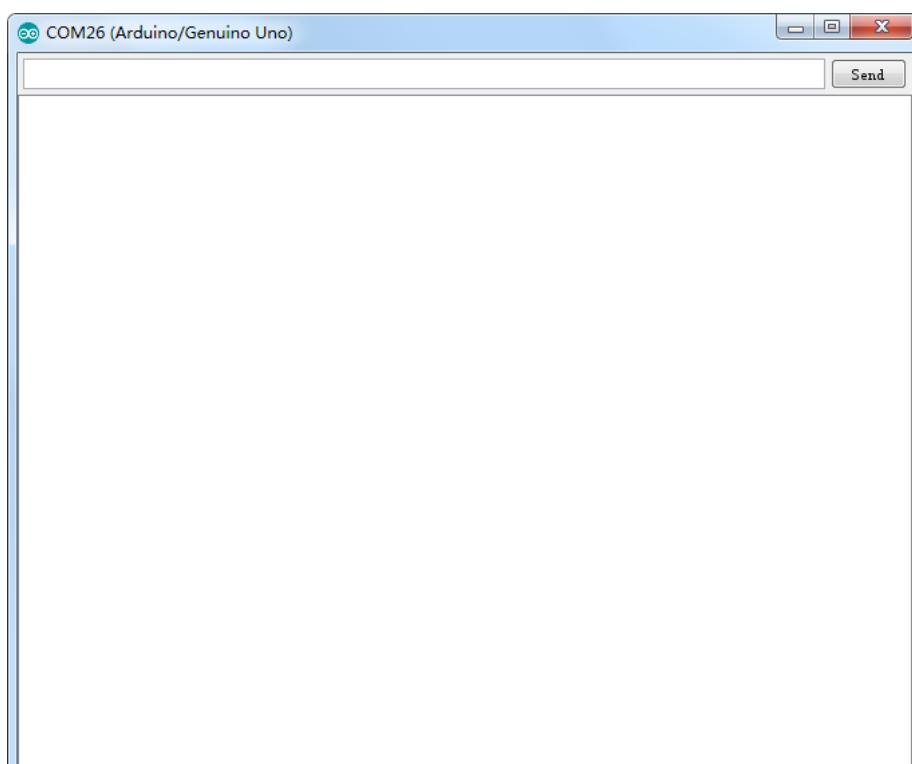


Wählen Sie aus, welcher Port im Serial Monitor geöffnet werden soll. Es ist derselbe wie bei der Auswahl eines Ports zum Hochladen des Arduino-Codes. Gehen Sie zu Extras-> Serial Port und wählen Sie den richtigen Port aus.

Tipps: Wählen Sie den gleichen COM-Port, den Sie im Geräte-Manager haben.

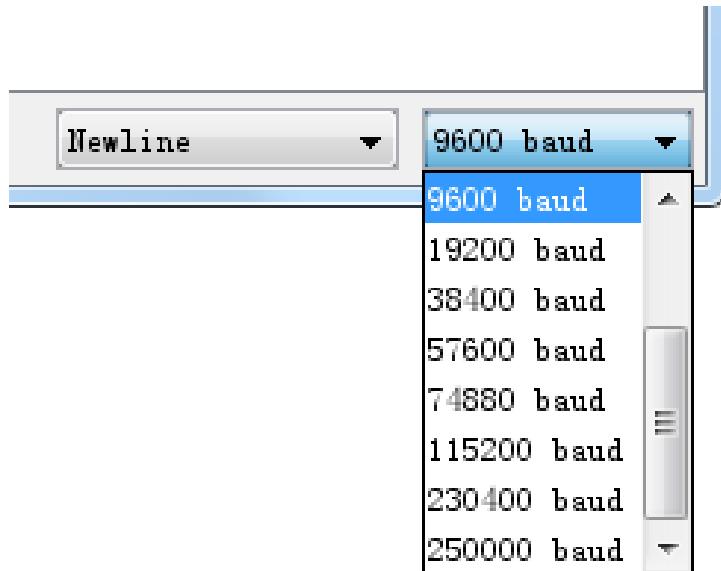


Einmal geöffnet, sollten Sie so etwas sehen:

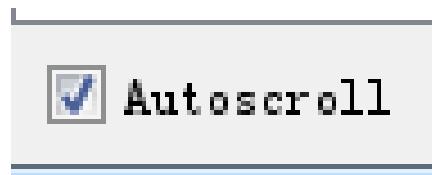


Einstellungen

Der serielle Monitor hat nur begrenzte Einstellungen, aber genug, um die meisten Ihrer seriellen Kommunikationsanforderungen zu behandeln. Die erste Einstellung, die du ändern kannst, ist die Baudrate. Klicken Sie auf das Dropdown-Menü Baudrate, um die richtige Baudrate auszuwählen. (9600 Baud)



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



Vorteile

Der serielle Monitor ist eine hervorragende, schnelle und einfache Möglichkeit eine serielle Verbindung mit Ihrem Arduino herzustellen. Wenn Sie bereits in der Arduino IDE arbeiten gibt es keine Notwendigkeit ein separates Terminal zu öffnen um Daten anzuzeigen.

Nachteile

Der Mangel an Einstellungen lässt im Serial Monitor viel zu wünschen übrig, und für fortgeschrittene serielle Kommunikation ist diese Methode vielleicht nicht gerade das Gelbe vom Ei.

Lektion 2 Blinken

Überblick

In dieser Lektion lernen Sie, wie Sie Ihre MEGA2560 R3 Controller-Karte programmieren können, um die eingebaute LED des Arduino blinken zu lassen und wie Sie Programme mit einfachen Schritten herunterladen können.

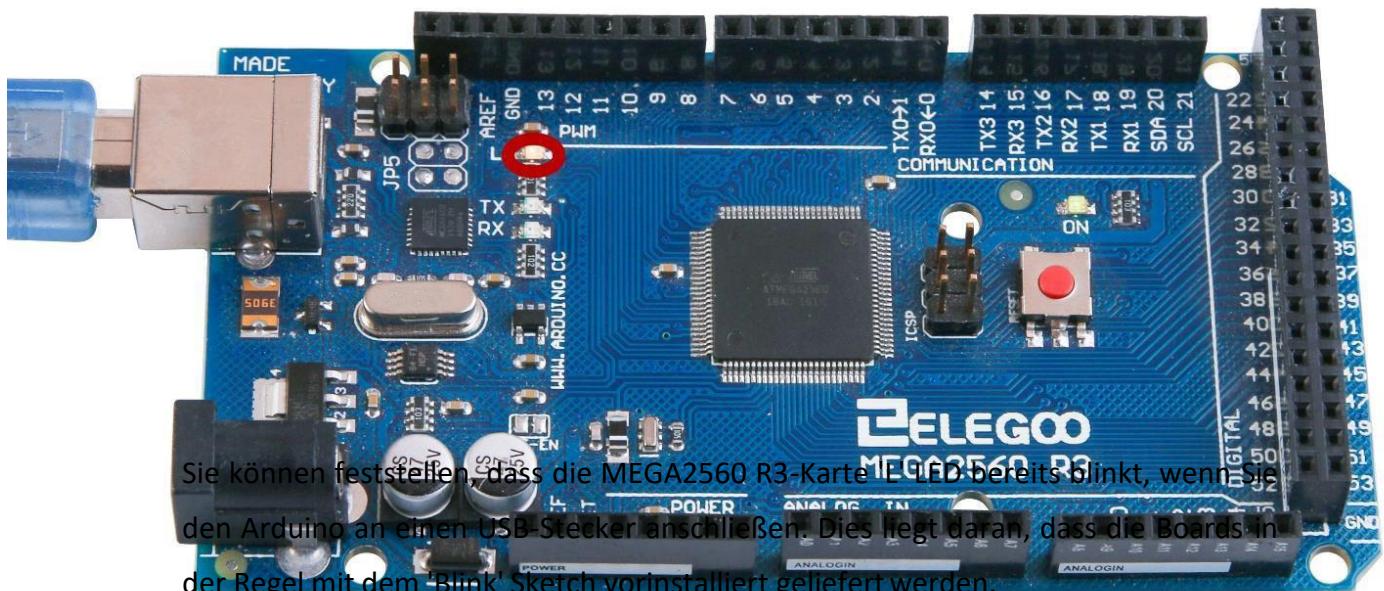
Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3

Prinzip

Die MEGA2560 R3-Platine verfügt über Reihen von Steckverbindern auf beiden Seiten, die zum Anschluss an mehrere elektronische Geräte und steckbare "Erweiterungen" (plug-in shields) verwendet werden können die ihre Leistungsfähigkeit erweitern.

Es hat auch eine einzige LED, die Sie aus Ihren Sketches ansprechen können. Diese LED ist auf der MEGA2560 R3-Platine angebracht (roter Kreis) und wird oft als 'L' LED bezeichnet, so wie sie auf der Platine beschriftet ist.



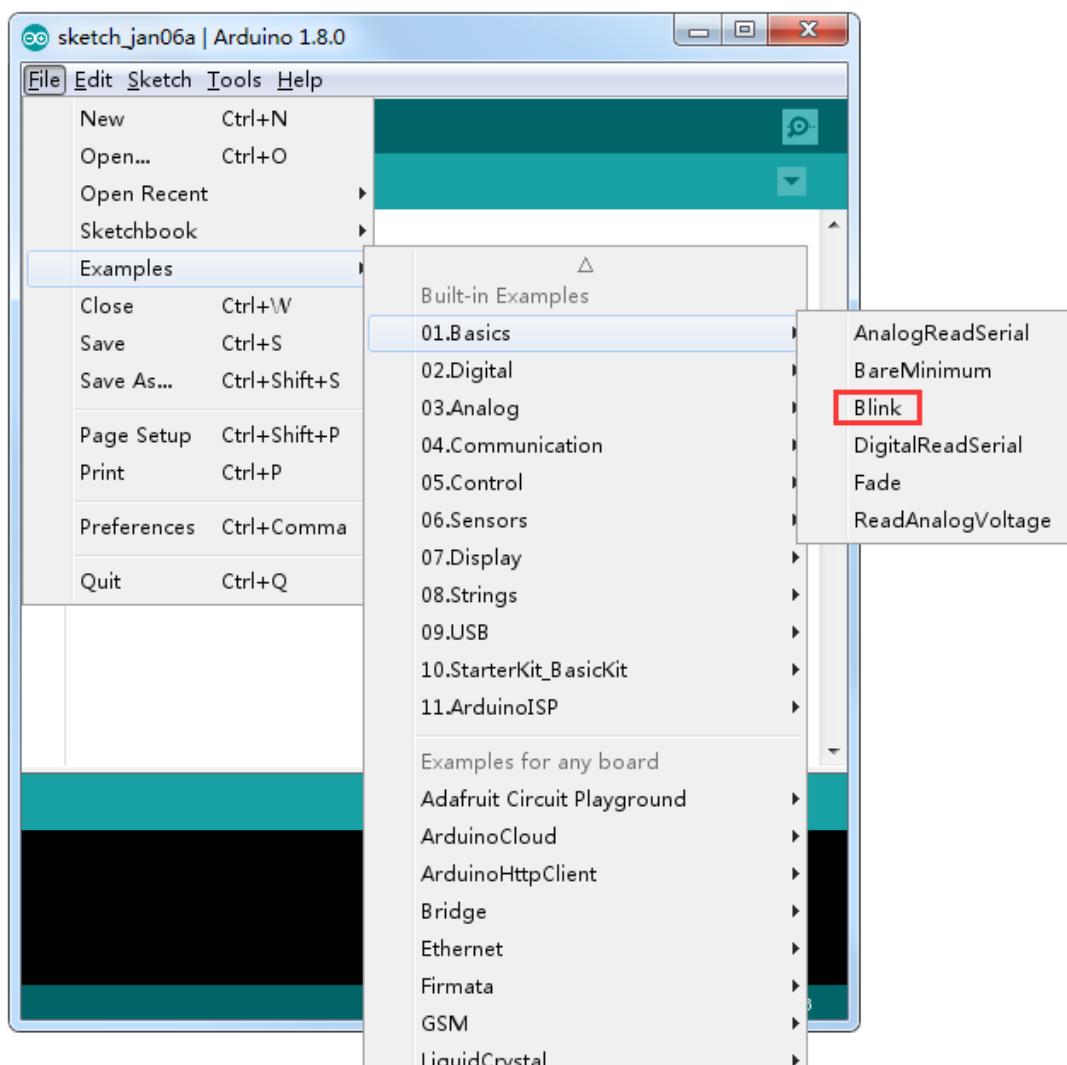
Sie können feststellen, dass die MEGA2560 R3-Karte die LED bereits blinkt, wenn Sie den Arduino an einen USB-Stecker anschließen. Dies liegt daran, dass die Boards in der Regel mit dem 'Blink' Sketch vorinstalliert geliefert werden.

In dieser Lektion werden wir das MEGA2560 R3 Board mit unserem eigenen Blink

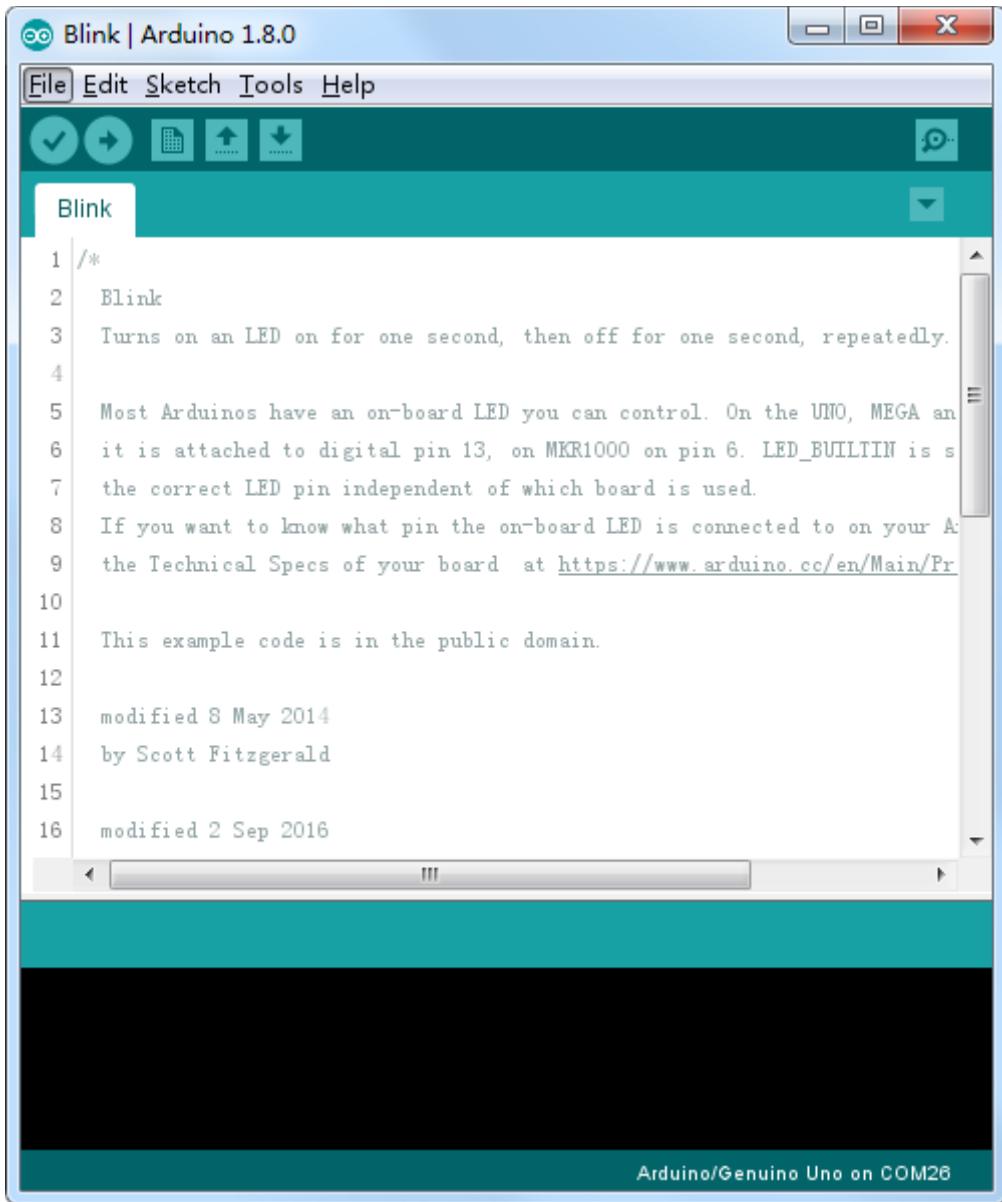
Sketch umprogrammieren und dann die Geschwindigkeit ändern, mit der esblinkt. In Lektion 0 haben Sie Ihre Arduino IDE eingerichtet und sorgten dafür, dass Sie den richtigen seriellen Port für die Verbindung zu Ihrem MEGA2560 R3 Board finden konnten. Die Zeit ist jetzt gekommen, um diese Verbindung zu testen und Ihr MEGA2560 R3 Board zu programmieren.

Die Arduino IDE enthält eine große Sammlung von Beispielsketches, die Sie laden und verwenden können. Dies beinhaltet einen Beispielsketch, um die LED 'L' blinken zu lassen.

Laden Sie den Sketch 'Blink', die Sie im Menüpunkt der IDE unter *Datei> Beispiele> 01.Basics* finden.



Wenn sich das Sketchfenster öffnet, vergrößern Sie es, damit Sie den gesamten Sketch im Fenster sehen können.

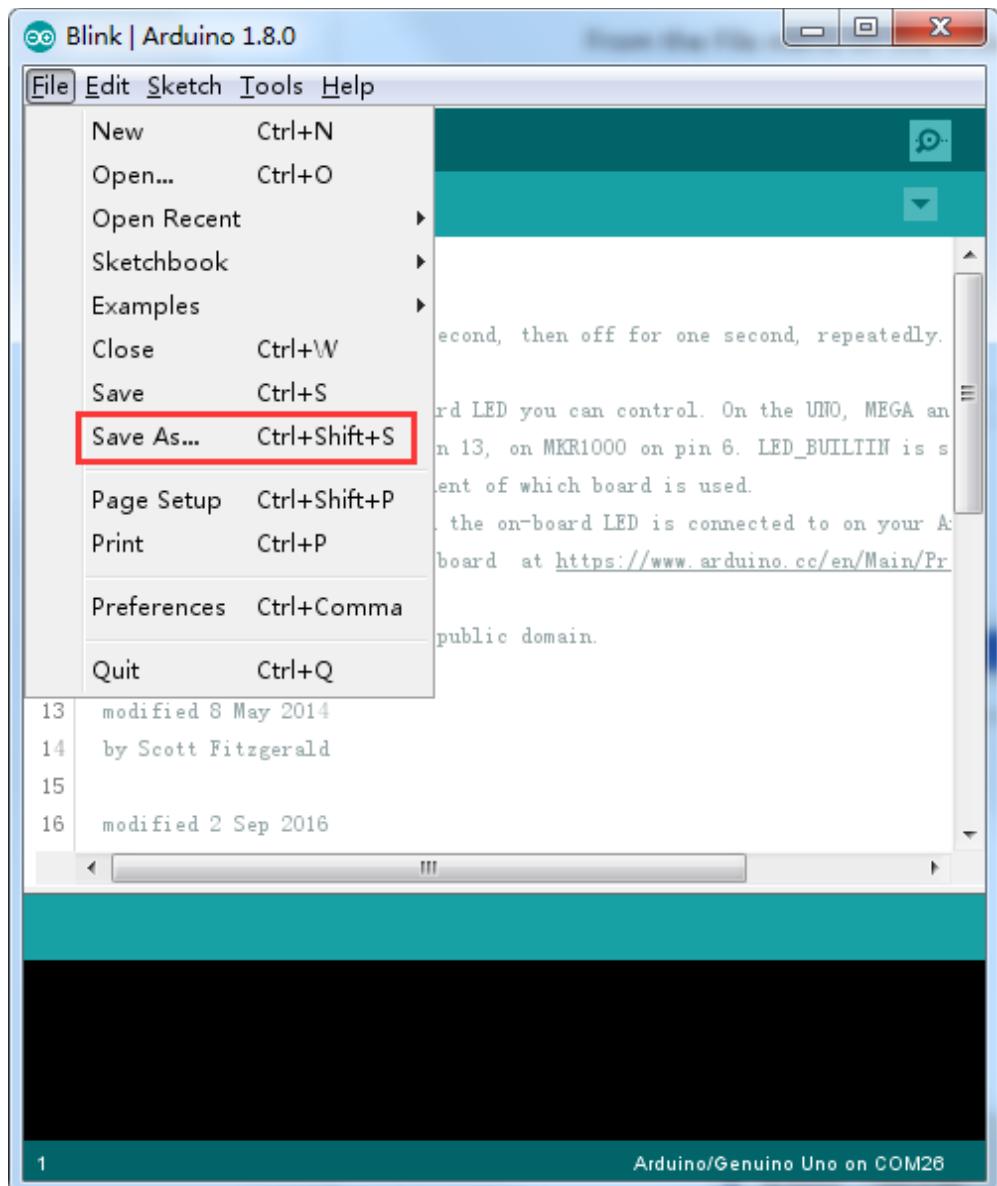


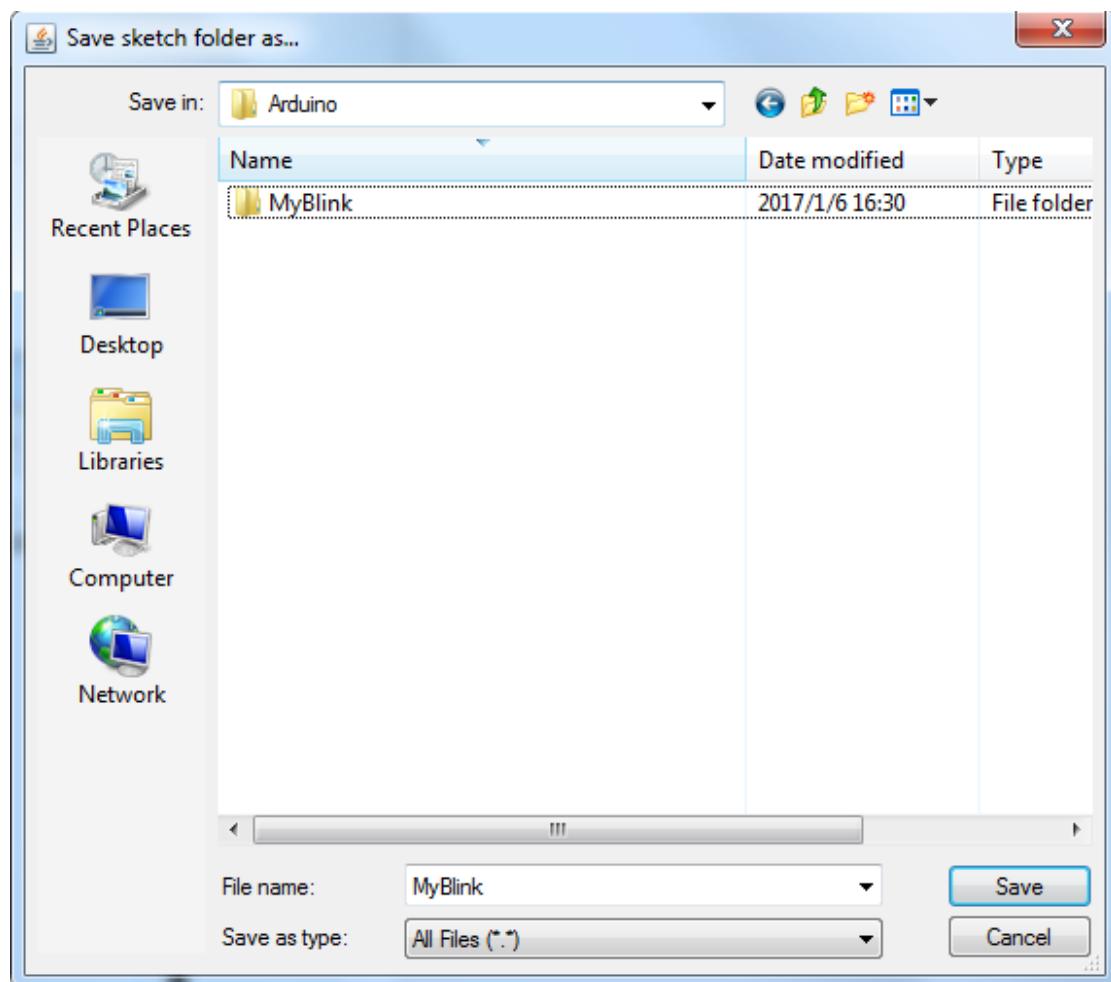
The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.8.0". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu is a toolbar with icons for save, undo, redo, and file operations. The main editor window displays the "Blink" sketch. The code is as follows:

```
1 /*
2  * Blink
3  * Turns on an LED on for one second, then off for one second, repeatedly.
4  *
5  * Most Arduinos have an on-board LED you can control. On the UNO, MEGA and MKR1000
6  * it is attached to digital pin 13, on MKR1000 on pin 6. LED_BUILTIN is set to the correct
7  * LED pin independent of which board is used.
8  * If you want to know what pin the on-board LED is connected to on your Arduino
9  * look at the Technical Specs of your board at https://www.arduino.cc/en/Main/Products
10 *
11 * This example code is in the public domain.
12 *
13 * modified 8 May 2014
14 * by Scott Fitzgerald
15 *
16 * modified 2 Sep 2016
```

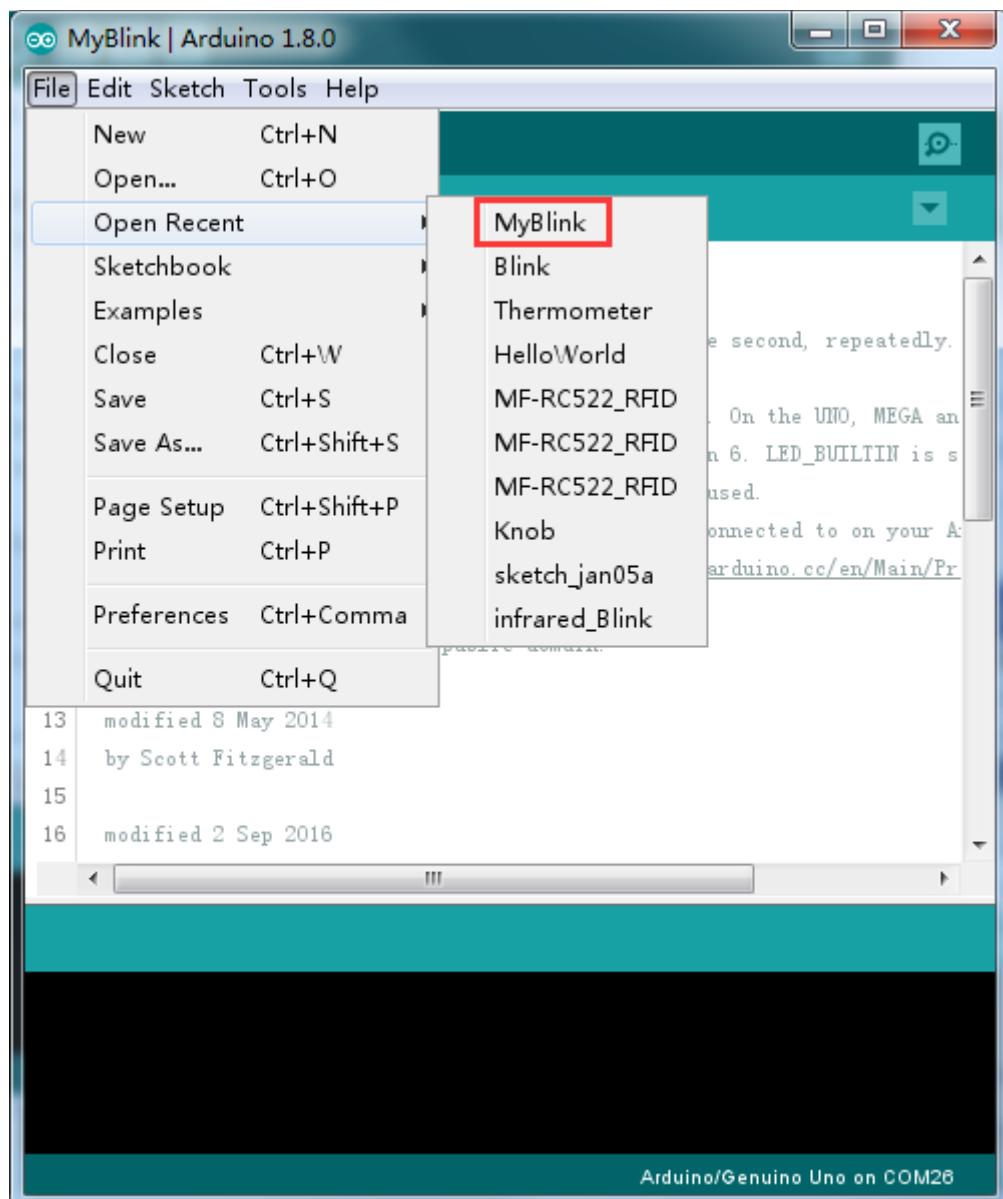
The status bar at the bottom right indicates "Arduino/Genuino Uno on COM26".

Die Beispielsketches, die in der Arduino IDE enthalten sind, sind schreibgeschützt. Das heißt, Sie können sie auf ein MEGA2560 R3 Board hochladen, aber wenn Sie sie ändern, können Sie sie nicht mit dem selben Dateinamenspeichern. Da wir diesen Sketch ändern werden, ist das erste was Sie tun müssen, diesen Sketch unter einem anderen Dateinamen zu speichern. Wählen Sie im Menü Datei auf der Arduino IDE die Option 'Speichern unter ..' und speichern Sie dann den Sketch mit dem Namen 'MyBlink'.

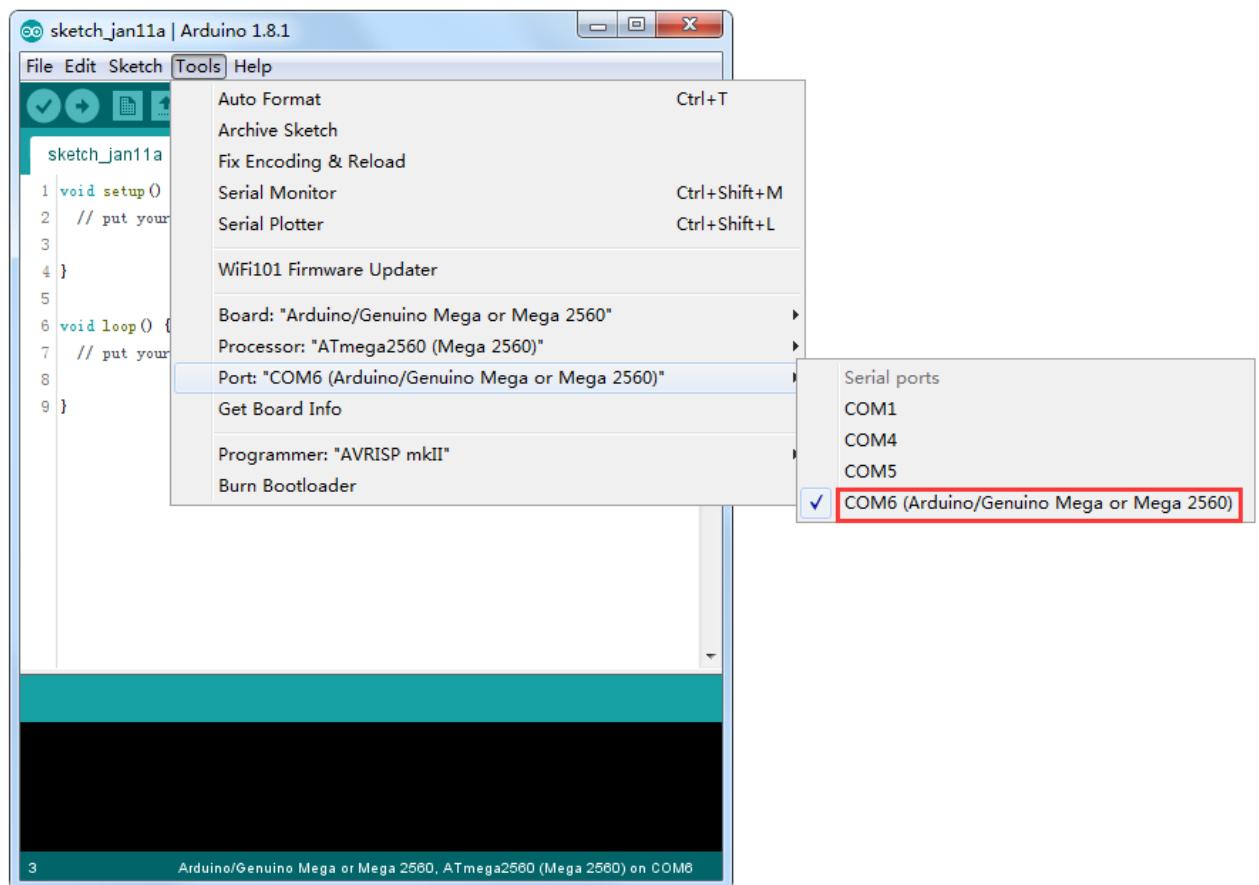
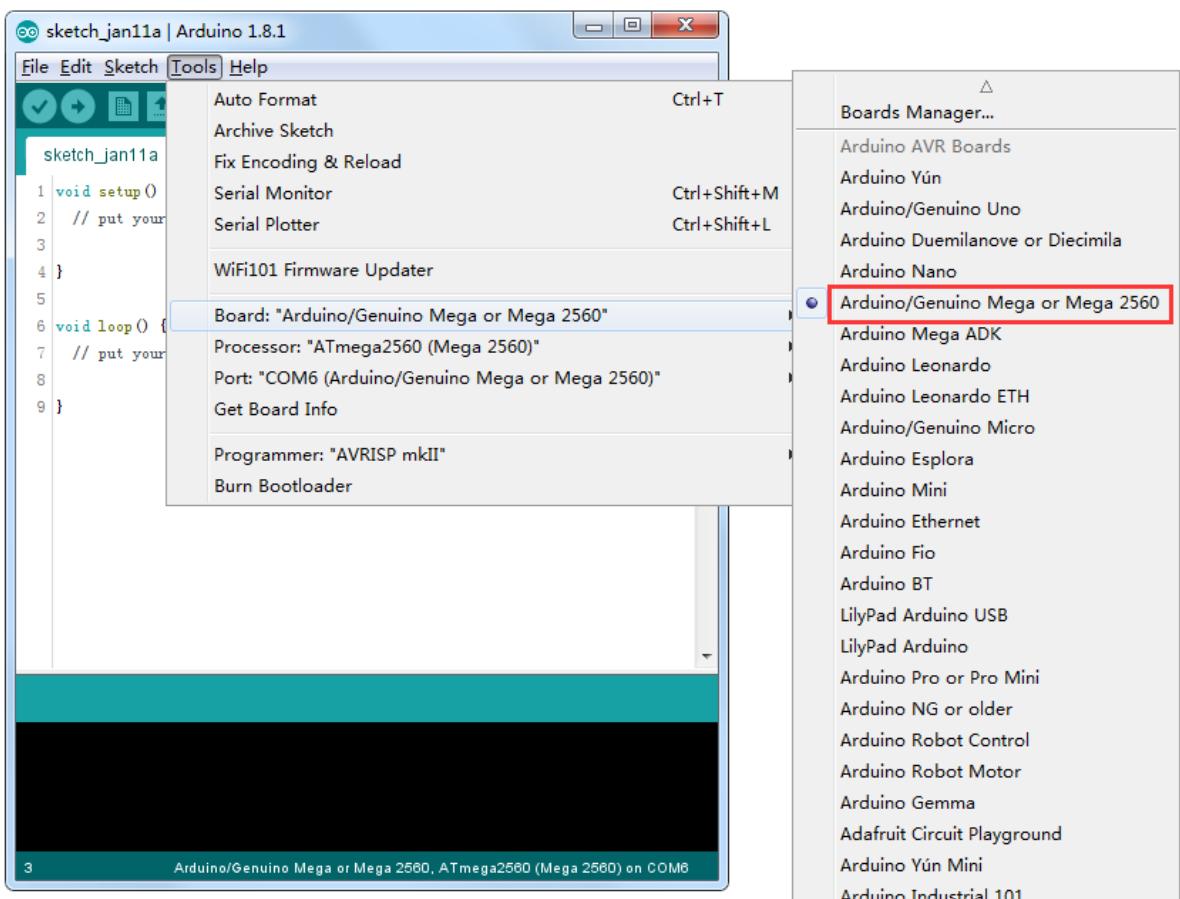




Sie haben Ihre Kopie von 'Blink' in Ihrem Sketchbuch gespeichert. Dies bedeutet, dass, wenn Sie es jemals wieder aufrufen wollen, können Sie es einfach mit dem Menü Datei> Sketchbook öffnen.



Schließen Sie Ihr Arduino-Board mit dem USB-Kabel an Ihrem Computer an und prüfen Sie, ob der 'Board Type' und 'Serial Port' richtig eingestellt sind.



Hinweis: Der Board-Typ und der Serial Port sind nicht unbedingt dieselben wie im Bild dargestellt. Wenn Sie einen Mega2560 verwenden, dann müssen Sie Mega 2560 als Board-Typ wählen, andere Optionen können in der gleichen Weise geändert werden. Und der serielle Port wird bei jedem anders angezeigt, obwohl hier COM 26 ausgewählt ist, könnte es COM3 oder COM4 auf Ihrem Computer sein. Ein richtiger COM-Port muss COMX (arduino XXX) heißen, der Zertifizierungskriterien unterliegt.

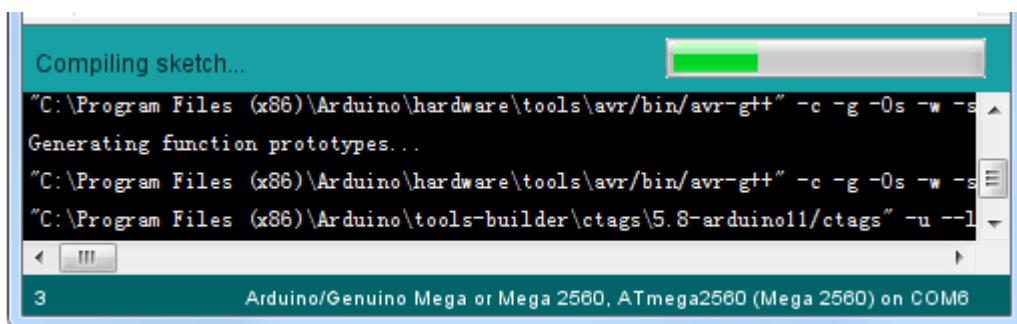
Die Arduino IDE zeigt Ihnen die aktuellen Einstellungen für das Board am unteren Rand des Fensters.



Klicken Sie auf die Schaltfläche 'Upload'. Die zweite Taste von links auf der Symbolleiste.



Wenn Sie den Statusbereich der IDE sehen, werden Sie einen Fortschrittsbalken und eine Reihe von Meldungen sehen. Zuerst wird es sagen: "Kompiliere Sketch ...". Dies verwandelt den Sketch in ein Format, das zum Hochladen auf das Board geeignet ist.



Als nächstes wechselt der Status zum 'Hochladen'. Zu diesem Zeitpunkt sollten die LEDs auf dem Arduino flackern, wenn der Sketch übertragen wird.

```
Uploading...
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-objcopy" -O ihex -R
Sketch uses 656 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 8183 bytes for loc
3 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM6
```

Schließlich wechselt der Status zu 'Done'.

```
Done uploading.
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-objcopy" -O ihex -R
Sketch uses 656 bytes (0%) of program storage space. Maximum is 253952 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 8183 bytes for loc
3 Arduino/Genuino Mega or Mega 2560, ATmega2560 (Mega 2560) on COM6
```

Die andere Nachricht sagt uns, dass der Sketch 928 Bytes der 32.256 Bytes des Arduinos belegt. Nach dem 'Kompiliere Sketch ..' können Sie die folgende Fehlermeldung erhalten:

```
Problem uploading to board. See http://www.arduino.cc/en/Troubleshooting/SerialPortProblems
Copy error messages
avrduude: stk500_recv(): programmer is not responding
avrduude: stk500_getsync() attempt 10 of 10: not in sync: resp=0x22
Problem uploading to board. See http://www.arduino.cc/en/Guide/Troubleshooting#uploading
3 Arduino/Genuino Uno on COM1
```

Dies kann bedeuten, dass Ihr Board überhaupt nicht angeschlossen ist oder die Treiber nicht installiert wurden (falls nötig) oder dass die falsche serielle Schnittstelle ausgewählt ist.

Wenn Sie diese Meldung erhalten, gehen Sie zurück zu Lektion 0 und überprüfen Sie Ihre Installation.

Sobald der Upload abgeschlossen ist, sollte das Board neu gestartet werden und beginnt zu blinken.

Öffne den Code

Beachten Sie, dass ein großer Teil dieses Sketches aus Kommentaren besteht. Dies sind keine tatsächlichen Programmanweisungen. Vielmehr erklären sie einfach, wie

das Programm funktioniert. Sie sind für SIEda.

Alles zwischen `/*` und `*/` an der Spitze des Sketches ist ein Blockkommentar; Es erklärt, was der Sketch ist.

Einzelige Kommentare beginnen mit `//` und alles bis zum Ende dieser Zeile gilt als Kommentar.

Die erste Codezeile ist:

```
int led = 13;
```

Wie der Kommentar oben im Sketch erklärt, gibt dies einen Namen an den Pin, an dem die LED angeschlossen ist. Dies ist auf den meisten Arduinos der Pin 13, einschließlich der MEGA2560 und Leonardo.

Als nächstes haben wir die 'setup' Funktion. Wieder, wie der Kommentar sagt, wird dies ausgeführt wenn die Reset-Taste gedrückt wird. Es wird auch ausgeführt, wenn die Karte aus irgendeinem Grund zurücksetzt, wie z. B. fehlender Strom, oder nachdem eine Sketch hochgeladen wurde.

```
void setup() {  
    //initialisiere den digital pin als output.  
    pinMode(led, OUTPUT);  
}
```

Jeder Arduino-Sketch muss eine "Setup" -Funktion haben, dies ist der Ort, an dem Sie selbst Anweisungen hinzufügen können und ist zwischen den beiden geschweiften Klammern `{ ... }`.

In diesem Fall gibt es dort nur einen Befehl, der, wie im Kommentar steht, dem Arduino-Board sagt, dass wir den LED-Pin als Ausgang verwenden werden.

Es ist auch zwingend erforderlich, dass ein Sketch eine 'Loop'-Funktion hat. Anders als die "Setup" -Funktion, die nur einmal läuft. Nach einem Reset wird die 'Loop'-Funktion, nach dem Ausführen der Befehle, sofort wiedergestartet.

```
void loop() {  
    digitalWrite(led, HIGH);      // LED einschalten (HIGH is the voltage level)  
    delay(1000);                // eine Sekunde lang warten  
    digitalWrite(led, LOW);       // die LED wird ausgeschaltet (Niedrige Spannung)  
    delay(1000);                // eine Sekunde lang warten  
}
```

Innerhalb der Loop-Funktion schaltet der Befehl zunächst den LED-Pin ein (HIGH), wartet dann '1000 Millisekunden (1 Sekunde), dann schaltet sich die LED-Pin aus und pausiert noch eine Sekunde.

Sie werden jetzt Ihre LED blinken sehen. Wie Sie vielleicht erraten haben, liegt der Schlüssel dazu, den Parameter in () für den Befehl 'delay' zu ändern.

```
30 // the loop function runs over and over again forever
31 void loop() {
32   digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the volt
33   delay(500)                         // wait for a second
34   digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the vo
35   delay(500)                         // wait for a second
36 }
```

Diese Verzögerungsperiode wird in Millisekunden angegeben, also wenn Sie möchten, dass die LED zweimal so schnell blinkt, ändern Sie den Wert von 1000 auf 500. Dies würde dann für eine halbe Sekunde warten anstatt einer ganzen Sekunde. Laden Sie den Sketch wieder hoch und Sie sollten sehen, dass die LED schneller blinkt.

Lektion 3 LED

Überblick

In dieser Lektion lernen Sie, wie man die Helligkeit einer LED mit unterschiedlichen Werten des Widerstands ändert.

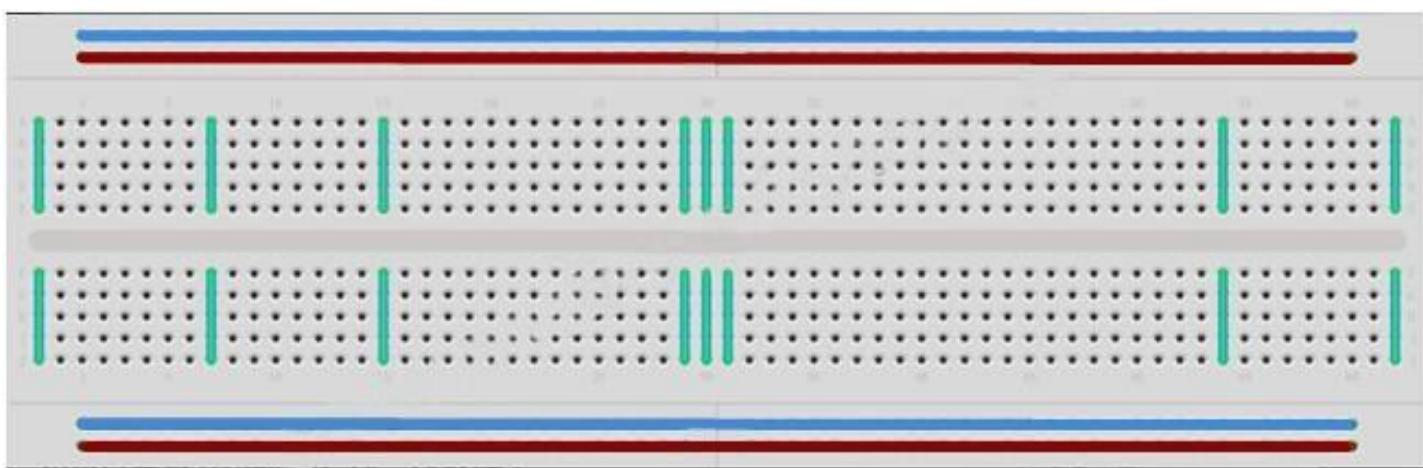
Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x 5mm rote LED
- (1) x 220 Ohm Widerstand
- (1) x 1k Ohm Widerstand
- (1) x 10k Ohm Widerstand
- (2) x M-M wires (Male to Male Kabel)

Komponenten Einführung:

BREADBOARD MB-102:

Ein Steckbrett ermöglicht es Ihnen, schnell Prototypen von Schaltungen zu erstellen, ohne die Anschlüsse löten zu müssen. Unten ist ein Beispiel.



Breadboards kommen in verschiedenen Größen und Konfigurationen. Die einfachste Art ist nur ein Gitter von Löchern in einem Plastikblock. Im Inneren befinden sich Streifen aus Metall, die eine elektrische Verbindung zwischen den

Löchern in den kürzeren Reihen herstellen. Das Schieben der Beine von zwei verschiedenen Komponenten in die gleiche Reihe verbindet sie so elektrisch zusammen. Eine tiefe Einkerbung die in der Mitte verläuft, zeigt an, dass es dort keine Verbindung gibt, was bedeutet, dass man einen Chip mit den Beinen auf beiden Seiten des Kanals verbauen kann, ohne diese miteinander zu verbinden. Einige Breadboards haben zwei Streifen von Löchern, die entlang der langen Kanten des Brettes laufen (oben rot und blau), die vom Hauptgitter getrennt sind. Diese haben farbige Streifen, die der Länge des Brettes nach hinunterlaufen und so für alle Komponenten eine gemeinsame Spannung zur Verfügung stellen kann. Sie sind in der Regel paarweise für +5 Volt und Masse. Diese Streifen werden als Reihen bezeichnet und ermöglichen es Ihnen, die Stromversorgung mit vielen Komponenten oder Punkten in der Platine zu verbinden.

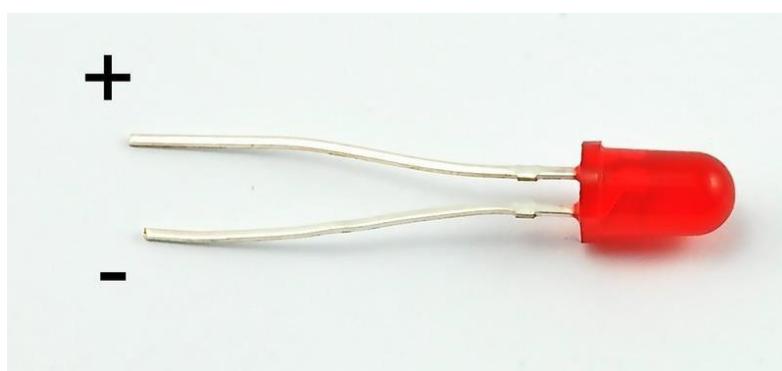
Während Breadboards ideal für das Prototyping sind, haben sie einige Einschränkungen. Weil die Anschlüsse nur hineingedrückt und lösbar sind, sind sie nicht so zuverlässig wie Lötverbindungen. Wenn Sie mit manchmal Probleme mit einer Schaltung haben (wie zum Beispiel Wackelkontakte), könnte es aufgrund einer schlechten Verbindung auf einem Steckbrett sein.

LED:

LEDs machen sich gut als Kontrollleuchten. Sie benutzen sehr wenig Elektrizität und sie halten quasi für immer und ewig.

In dieser Lektion werden Sie vielleicht die häufigste aller LEDs verwenden: eine 5mm rote LED. 5mm bezieht sich auf den Durchmesser der LED. Andere übliche Größen sind 3mm und 10mm.

Sie können eine LED nicht direkt an eine Batterie oder eine Spannungsquelle anschließen, da 1) die LED eine positive und eine negative Leitung hat und nicht leuchtet, wenn sie falsch platziert ist und 2) eine LED mit einem Widerstand zum Begrenzen der Menge des Stromes der durch sie fließt benötigen; Sonst wird sie durchbrennen!



Wenn man einer LED keinen Widerstand vorschaltet, dann kann diese fast sofort zerstört werden, da zu viel Strom durchfließen wird, erhebt und der Kontakt wo das Licht erzeugt wird zerstört wird.

Es gibt zwei Möglichkeiten zu sagen, welches ist die positive Leitung der LED ist und welches die negative.

Erstens ist die positive Leitung länger.

Zweitens, wo die negative Leitung in den Körper der LED eintritt, gibt es eine flache Kante zum Gehäuse der LED.

Wenn Sie zufällig eine LED haben, die eine flache Seite neben der längeren hat, sollten Sie davon ausgehen, dass die längere Leitung die positiveist.

Widerstände:

Wie der Name schon sagt, widerstehen Widerstände dem Stromfluss. Je höherder Wert des Widerstands ist, desto mehr behindert es den Strom am fließen und desto weniger Strom fließt durch. Wir werden das nutzen, um zu kontrollieren, wie viel Strom durch die LED fließt und deshalb, wie hell esleuchtet.

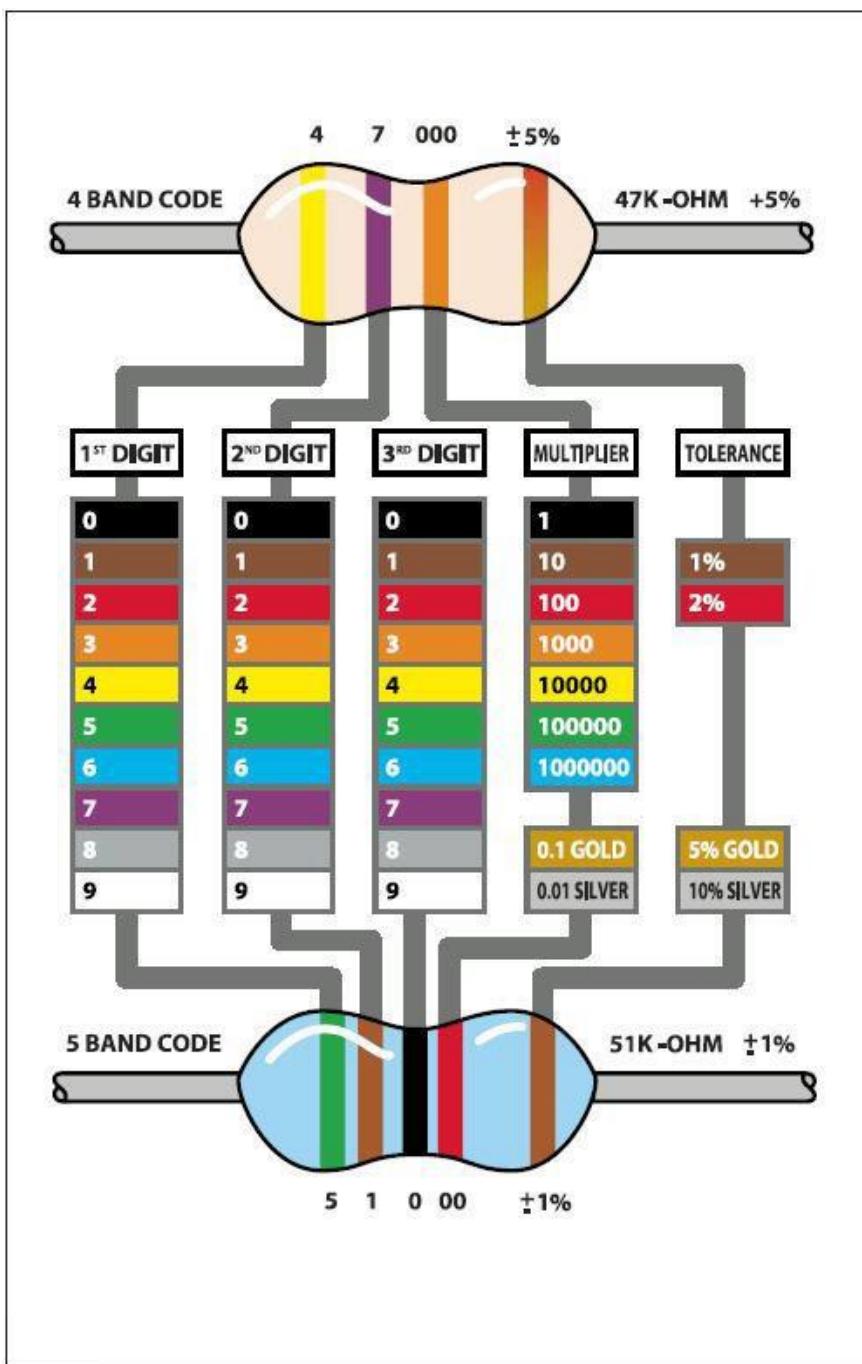


Doch zuerst, mehr über Widerstände... :-P

Die Einheit des Widerstands heißt Ohm, die gewöhnlich zu Ω , dem griechischen Buchstaben Omega verkürzt wird. Da ein Ohm ein niedriger Widerstandswert ist (es behindert den Strom quasi überhaupt nicht), bezeichnen wir auch die Werte der Widerstände in $k\Omega$ (1.000 Ω) und $M\Omega$ (1.000.000 Ω). Diese heißen Kilo-Ohm und Mega-Ohm.

In dieser Lektion werden wir drei verschiedene Werte des Widerstands verwenden: 220Ω , $1k\Omega$ und $10k\Omega$. Diese Widerstände sehen alle gleich aus, außer dass sie unterschiedliche farbige Streifen haben. Diese Streifen erzählen Ihnen den Wert des Widerstands.

Der Widerstands Farbcode hat drei farbige Streifen und dann einen Goldstreifen an einem Ende.

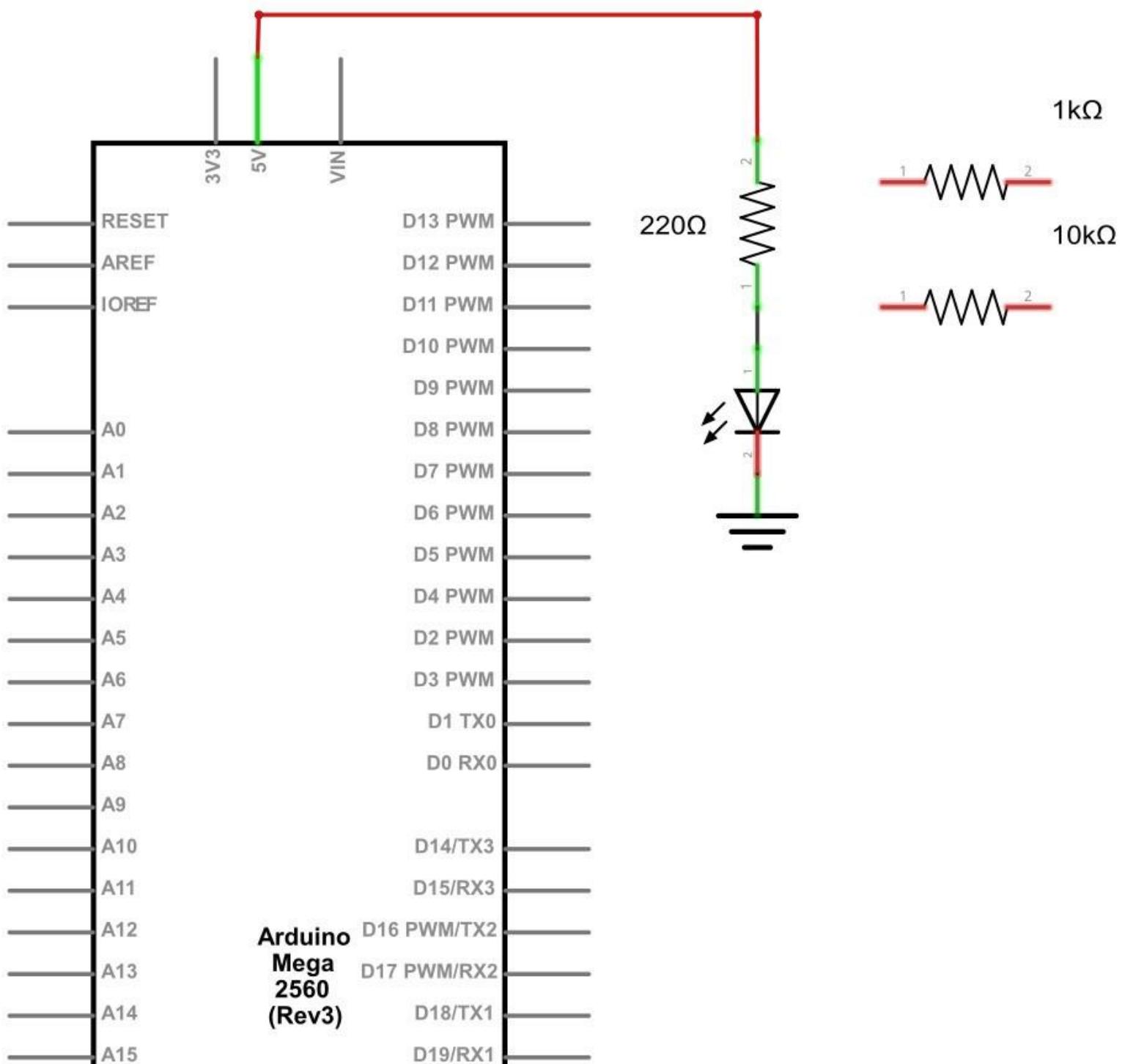


Im Gegensatz zu LEDs haben Widerstände keine positive und negative Leitung. Sie können so in beide Richtungen eingebaut werden.

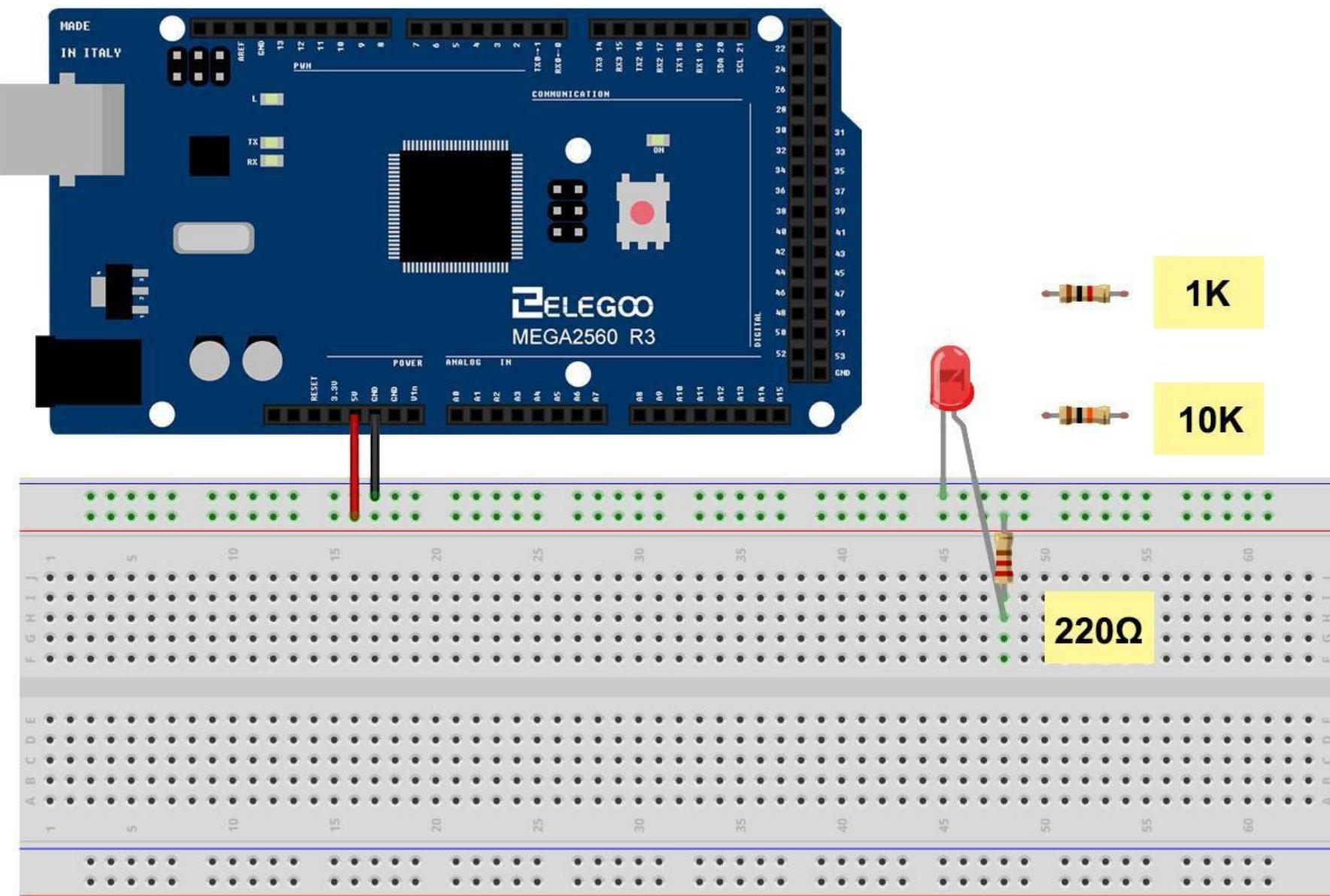
Wenn Sie diese Ansatzmethode zu kompliziert finden, können Sie den Wert der Widerstände auf dem Papier an dem diese befestigt sind direkt ablesen. Oder Sie können stattdessen ein digitales Multimeter verwenden.

Verbindung

Schema



Schaltplan:

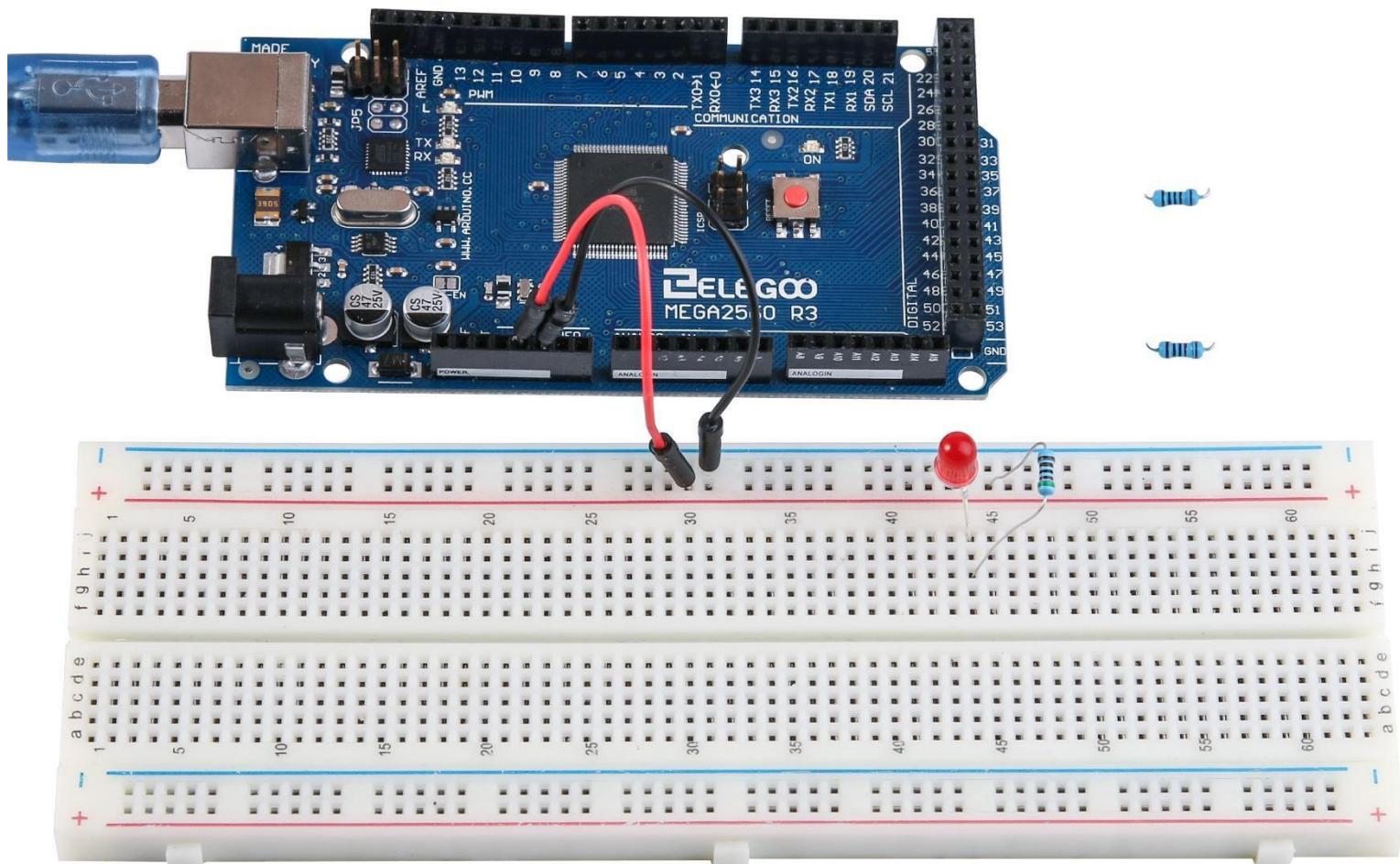


Der MEGA2560 ist eine gute Spannungquelle mit 5 Volt, die wir benötigen um die LED und den Widerstand mit Strom zu versorgen. Sie müssen nichts mit Ihrem MEGA2560 machen, außer dass Sie mit einem USB-Kabel mit Ihrem PC verbinden. Wenn der $220\ \Omega$ Widerstand angeschlossen ist, sollte die LED ganz hell sein. Wenn Sie den $220\ \Omega$ Widerstand durch den $1k\Omega$ Widerstand auswechseln, erscheint die LED ein wenig gedimmter. Schließlich wird die LED mit dem $10\ k\Omega$ Widerstand kaum noch leuchten. Ziehen Sie das rote Kabel (5V Spannungsversorgung) aus dem Steckbrett und lassen Sie es das Loch nur noch berühren, so dass es wie ein Schalter wirkt, wenn Sie es hin und her bewegen. Sie sollten den Unterschied nummerieren. Im Moment haben Sie 5V zu einem Bein des Widerstands, das andere Bein des Widerstands geht auf die positive Seite der LED und die andere Seite der LED geht zu GND (Ground). Wenn wir jedoch den Widerstand bewegen haben, so dass er hinter der LED liegt, wie unten gezeigt, leuchtet die LED immernoch.

Sie wollen nun sicherlich den 220Ω Widerstand wieder anbringen.

Es spielt keine Rolle, auf welcher Seite der LED wir den Widerstand setzen, solange es irgendwo innerhalb des Schaltkreises ist.

Beispelfoto



Lektion 4 RGB LED

Überblick

RGB-LEDs sind eine lustige und einfache Möglichkeit, Ihren Projekten etwas Farbe einzuflößen. Denn sie sind wie 3 herkömmliche LEDs in einem. Wie man sie benutzt und verbindet, ist nicht viel anders.

Sie kommen meistens in 2 Versionen: Common Anode oder Common Cathode. Common Anode verwendet 5V auf dem gemeinsamen Pin, während Common Cathode mit Masse verbunden ist.

Wie bei jeder LED, müssen wir einige Widerstände (3 insgesamt) anschließen, damit wir den Strom, der fließt, begrenzen können.

In unserer Skizze beginnen wir mit der LED im roten Farbzustand, dann verändern wir sie nach Grün, dann lassen wir sie Blau leuchten und schließlich wieder rot. Auf diese Weise werden wir nahezu alle möglichen Farben sehen.

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x 830 Tie Points Breadboard
- (4) x M-M wires (Male to Male Kabel)
- (1) x RGB LED
- (3) x 220 Ohm Widerstände

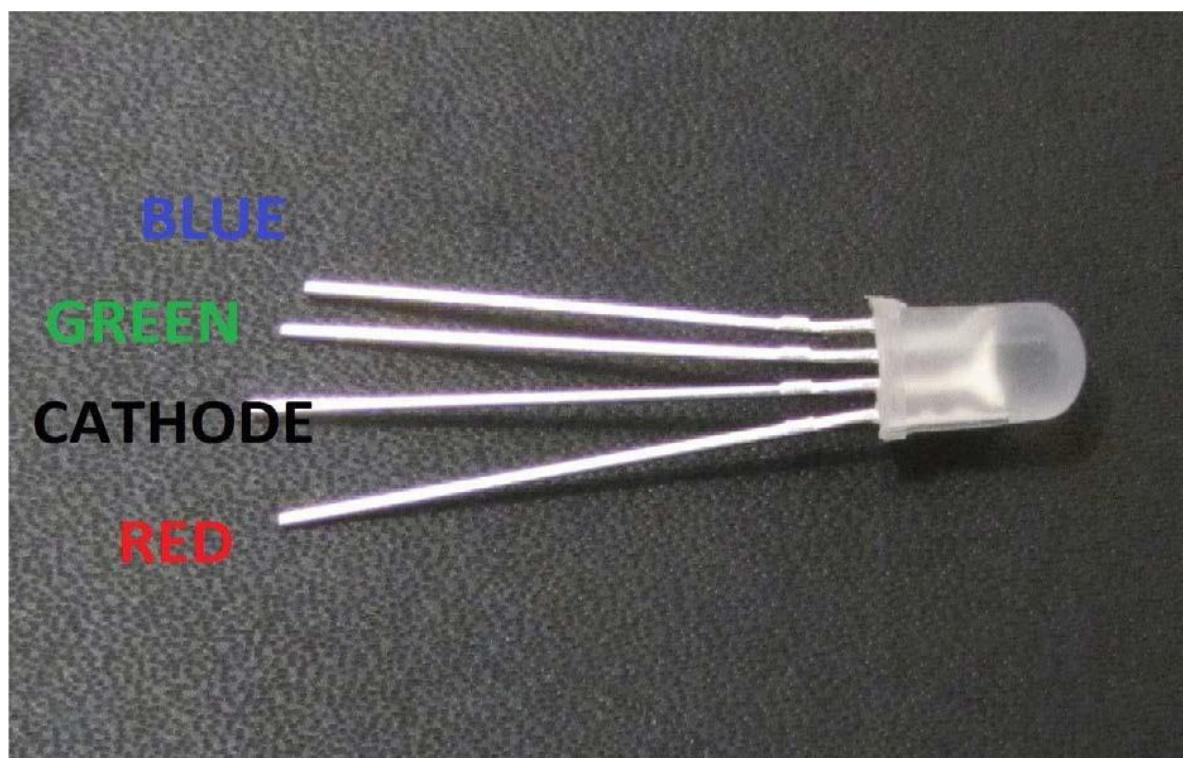
Komponenten Einführung:

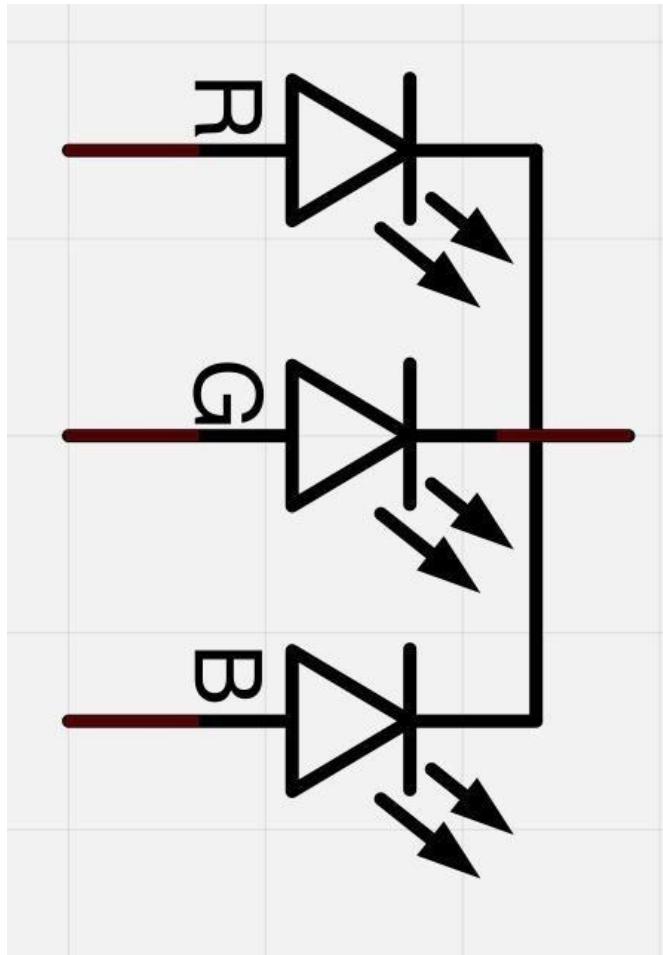
RGB:

Auf den ersten Blick sehen die LEDs RGB (rot, grün und blau) genau wie normale LEDs aus. Doch innerhalb des üblichen LED-Pakets gibt es eigentlich drei LEDs, eine rote, eine grüne und ja sogar auch eine blaue, (wer hätte das gedacht?). Durch die Kontrolle der Helligkeit der einzelnen LEDs können Sie so ziemlich jede beliebige Farbe mischen.

Wir mischen Farben auf die gleiche Weise, wie man Farbe auf einer normalen Farbpalette mischen würde - indem man die Helligkeit jeder der drei LEDs anpasst. Der "harte Weg", dies zu tun, wäre, unterschiedliche Wertwiderstände (oder variable Widerstände) zu verwenden, wie wir es in Lektion 2 gemacht haben, aber das ist eine Menge Arbeit! Zum Glück hat MEGA2560 R3 Board eine analoge Funktion, die Sie mit den Steckplätzen verwenden können, die mit einem ~ gekennzeichnet sind. So lässt sich eine variable Leistung an die entsprechenden LEDs einstellen.

Die RGB-LED hat vier Leitungen. Es gibt eine Leitung die zum positiven Anschluss jeder einzelnen LED innerhalb der LED und einer einzigen Leitung führt, die mit allen drei negativen Seiten der LEDs verbunden ist, die Kathode.





Hier auf den Fotos sehen Sie 4 Elektroden LEDs. Jeder einzelne Leitung für grüne oder blaue oder rote Farbe heißt Anode. Sie werden immer mit "+" verbunden. Die Kathode geht zu "-" (Masse). Wenn Sie es anders anschließen, leuchtet die LED nicht.

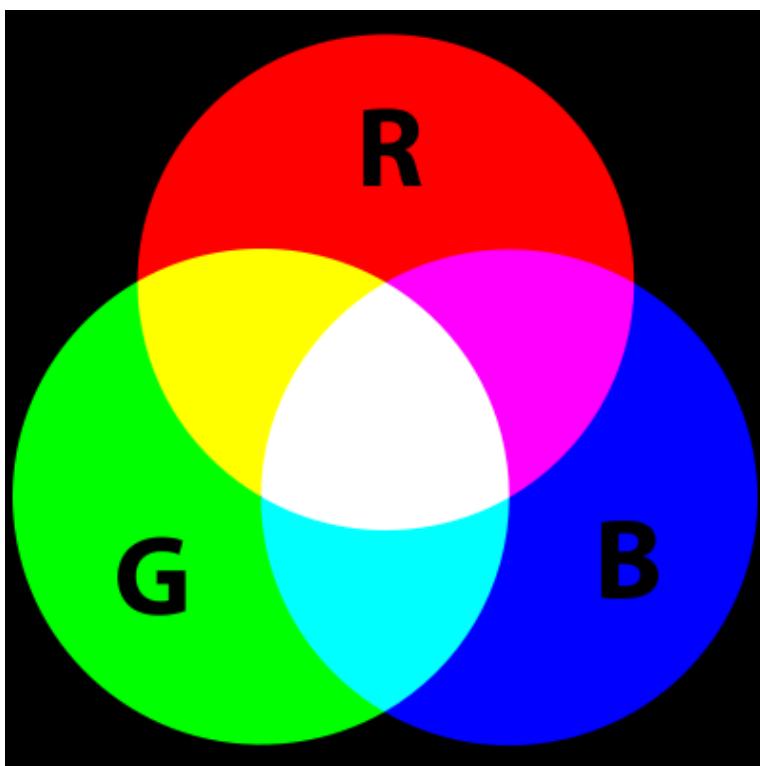
Die gemeinsame negative Leitung des LED-Pakets ist der zweite Pin von der flachen Seite. Es ist auch das längste der vier und wird mit dem Boden verbunden sein.

Jede LED im Inneren des Pakets benötigt einen eigenen 220Ω Widerstand, um zu verhindern, dass 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 Ausgangssteckplätzen des MEGA2560verbunden.

Farben:

Der Grund, dass Sie jede Farbe mischen können indem Sie die Mengen an rotem, grünem und blauem Licht variieren, ist, dass Ihr Auge drei Arten von Lichtrezeptoren hat (rot, grün und blau). Ihr Auge und Ihr Gehirn verarbeiten die Mengen von Rot, Grün und Blau und verwandeln sie in eine Farbe des für uns sichtbaren Farbpektrums.

In gewisser Weise spielen wir unserem Auge mit den drei LEDs einen Trick vor. Diese gleiche Idee wird auch in Fernsehgeräten verwendet, wo das LCD rote, grüne und blaue Farbpunkte nebeneinander hat, die zusammen jeweils einen Pixel bilden.



Wenn wir die Helligkeit aller drei LEDs gleich einstellen, dann ist die Gesamtfarbe des Lichtes weiß. Wenn wir die blaue LED ausschalten, damit nur die roten und grünen LEDs die gleiche Helligkeit haben, erscheint das Licht gelb.

Wir können die Helligkeit von jedem der roten, grünen und blauen Teile der LED separat steuern, so dass es möglich ist, jede beliebige Farbe zu mischen.

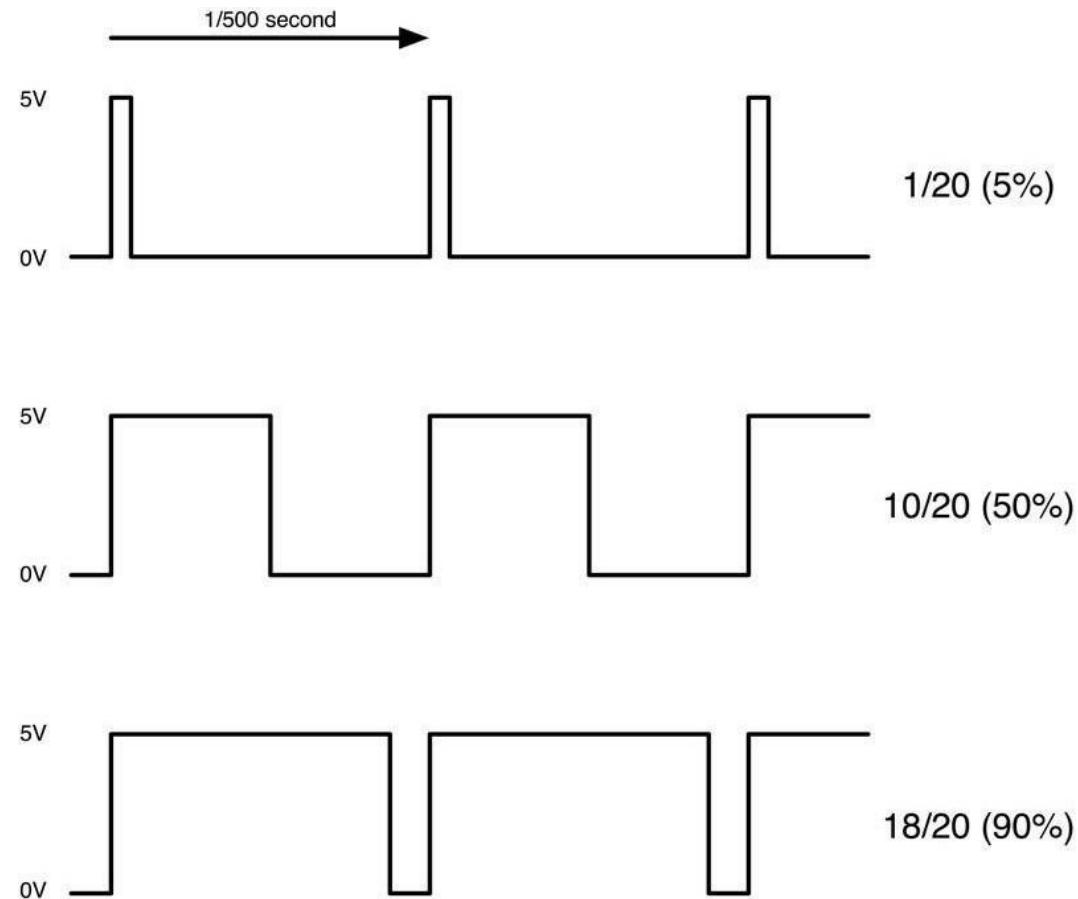
Schwarz ist im Grunde genommen so viel wie die Abwesenheit von Licht. Deshalb, kommen wir Schwarz am nächsten wenn wir alle drei Farbenausschalten.

Theorie (PWM)

Pulsbreitenmodulation (PWM) ist eine Technik zur Leistungssteuerung.

Wir verwenden es hier auch für die Helligkeit der LEDs.

Das folgende Diagramm zeigt das Signal von einem der PWM-Pins am MEGA2560.



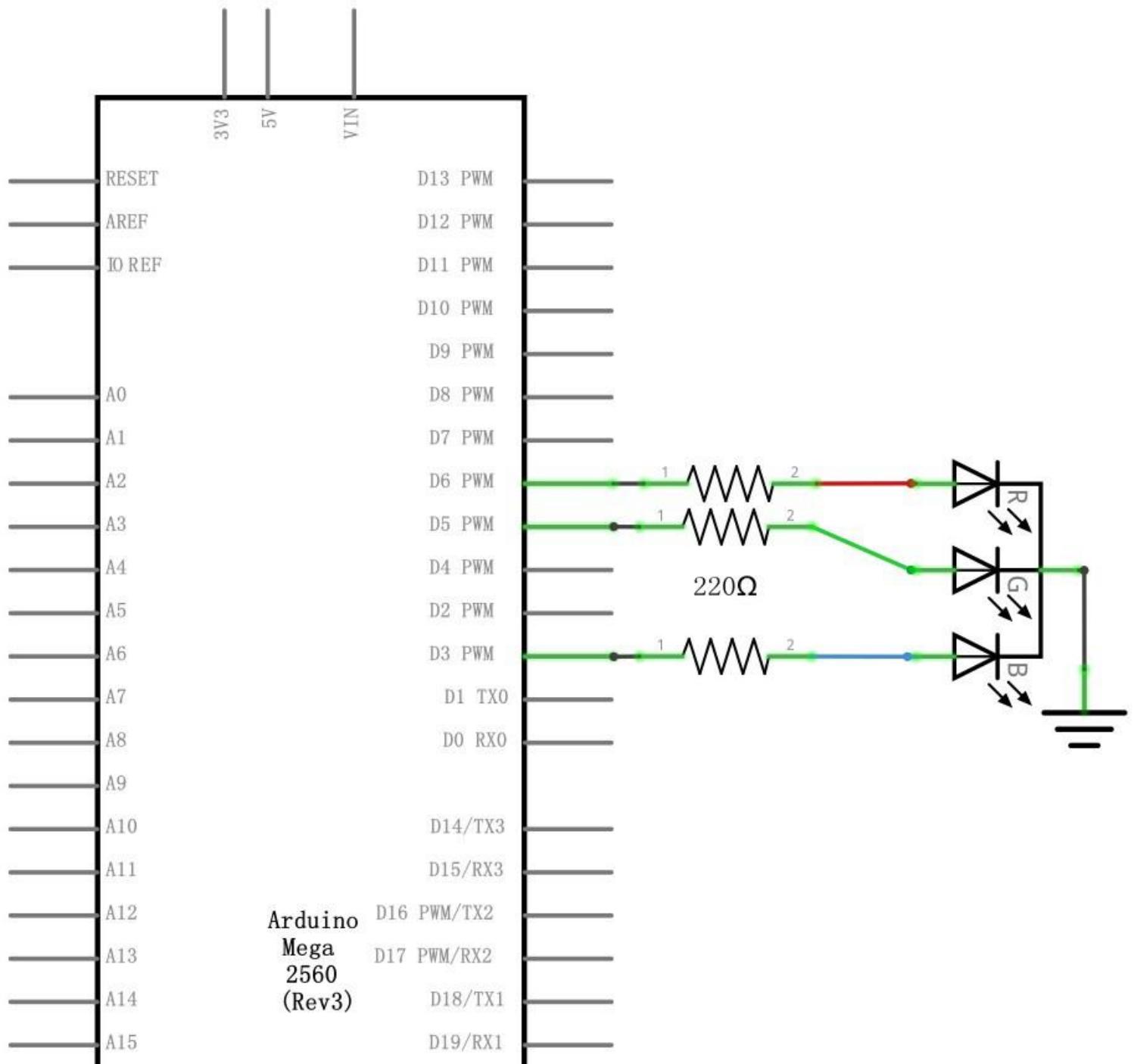
Etwa alle 1/500 Sekunden wird der PWM-Ausgang einen Puls erzeugen. Die Länge dieses Pulses wird durch die Funktion 'analogWrite' gesteuert. So wird "analogWrite (0)" überhaupt keinen Puls erzeugen und "analogWrite (255)" wird einen Puls erzeugen, der bis zum nächsten Puls dauert, so dass der Ausgang eigentlich immer eingeschaltet ist.

Wenn wir einen Wert im AnalogWrite angeben, der irgendwo zwischen 0 und 255 liegt, dann werden wir einen Puls erzeugen. Wenn der Ausgangsimpuls nur 5% der Zeit hoch ist, dann werden nur noch 5% der vollen Leistung empfangen.

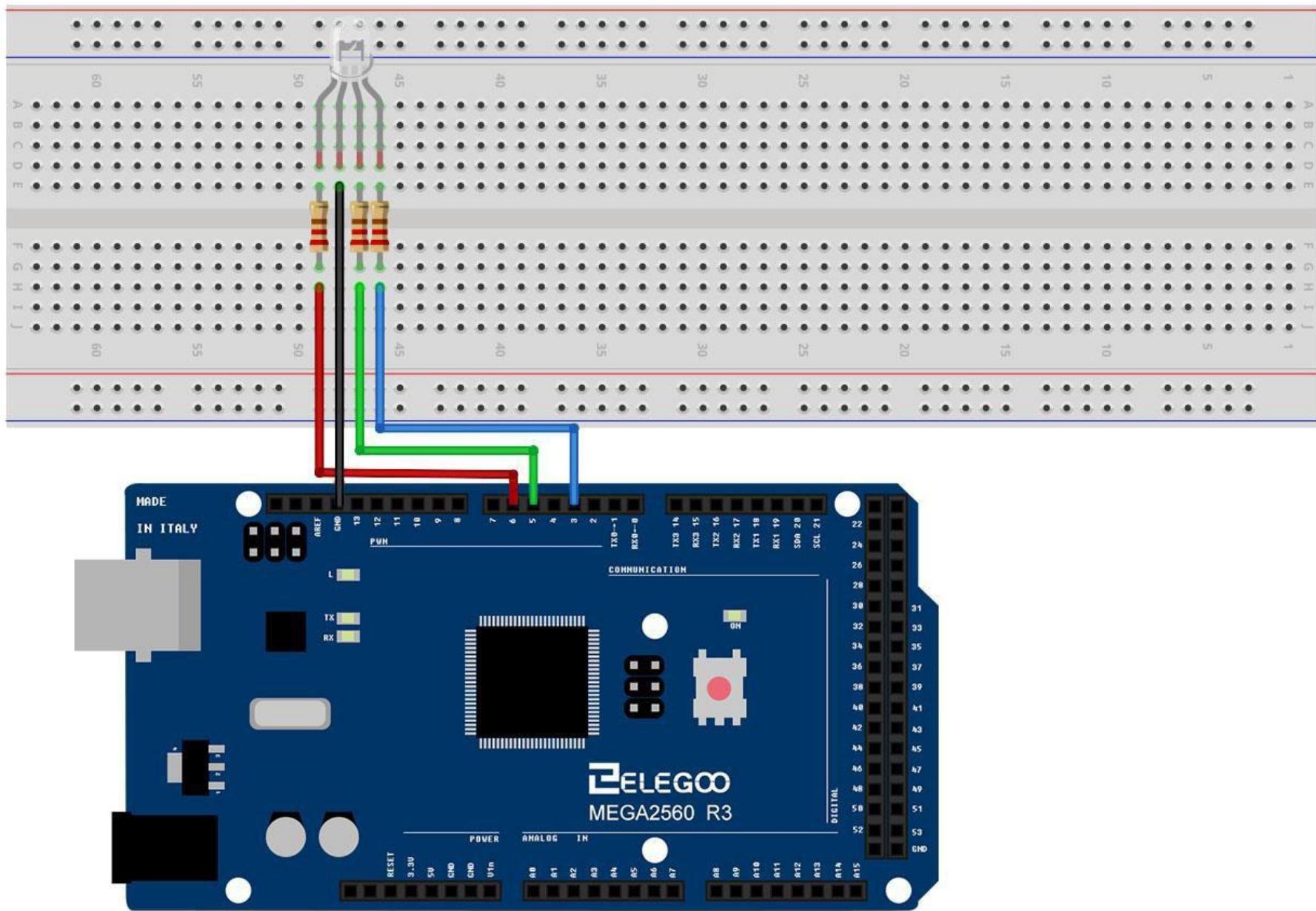
Wenn jedoch die Ausgabe von 5V für 90% der Zeit an ist, dann erhält die Last 90% der Leistung, die an sie geliefert wird. Wir können nicht sehen, dass die LEDs bei dieser Geschwindigkeit ein- und ausschalten, soder es sieht für den Menschen einfach so aus als ob sich die Helligkeit ändert.

Verbindung

Schema

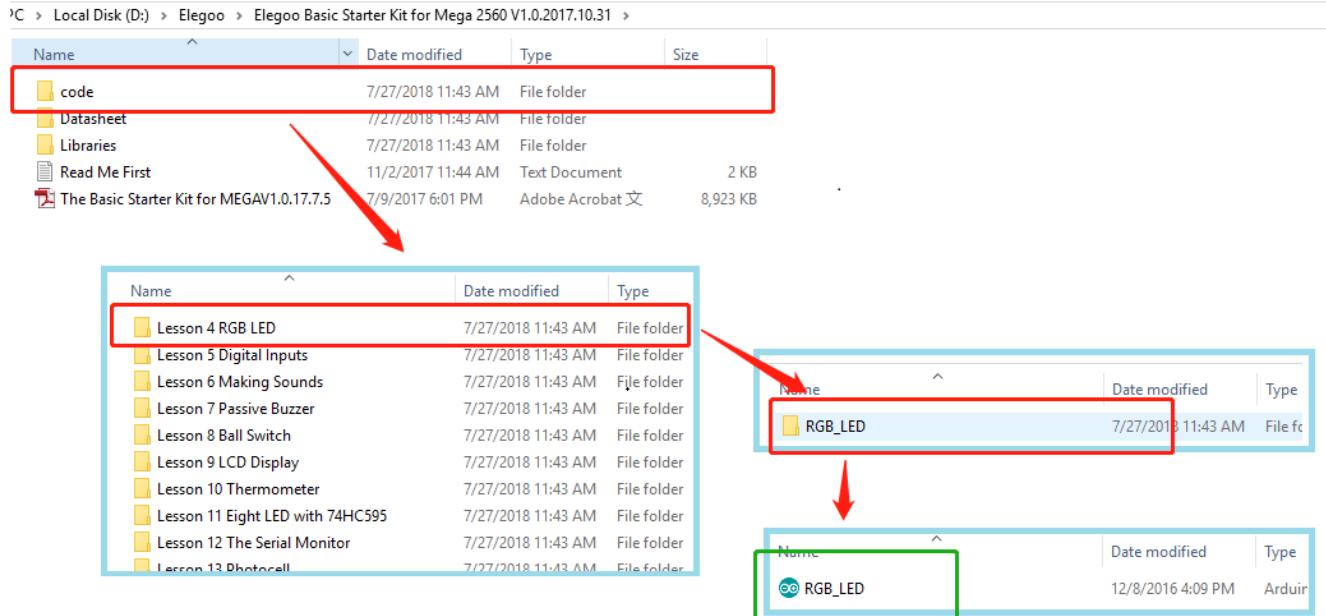


Schaltplan



Code

After wiring, please open the Sketch in folder path: Tutorial > English > code > Lesson 4 RGB LED > RGB_LED, and click UPLOAD to upload the program.



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

Der nächste Schritt ist, die 'Setup' Funktion zu schreiben. Wie wir in früheren Lektionen gelernt haben, läuft die Setup-Funktion nur einmal, und das nachdem der Arduino zurückgesetzt wurde. In diesem Fall ist alles, was es zu tun hat, die drei Steckplätze zu definieren, die wir als Ausgänge verwenden.

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

Bevor wir uns die "Loop" -Funktion anschauen, schauen wir uns die letzte Funktion in dem Sktech an.

Die definierten Variablen

57 / 224

redValue = 255; // choose a value between 1 and 255 to change the color.

```
greenValue = 0;
```

```
blueValue = 0;
```

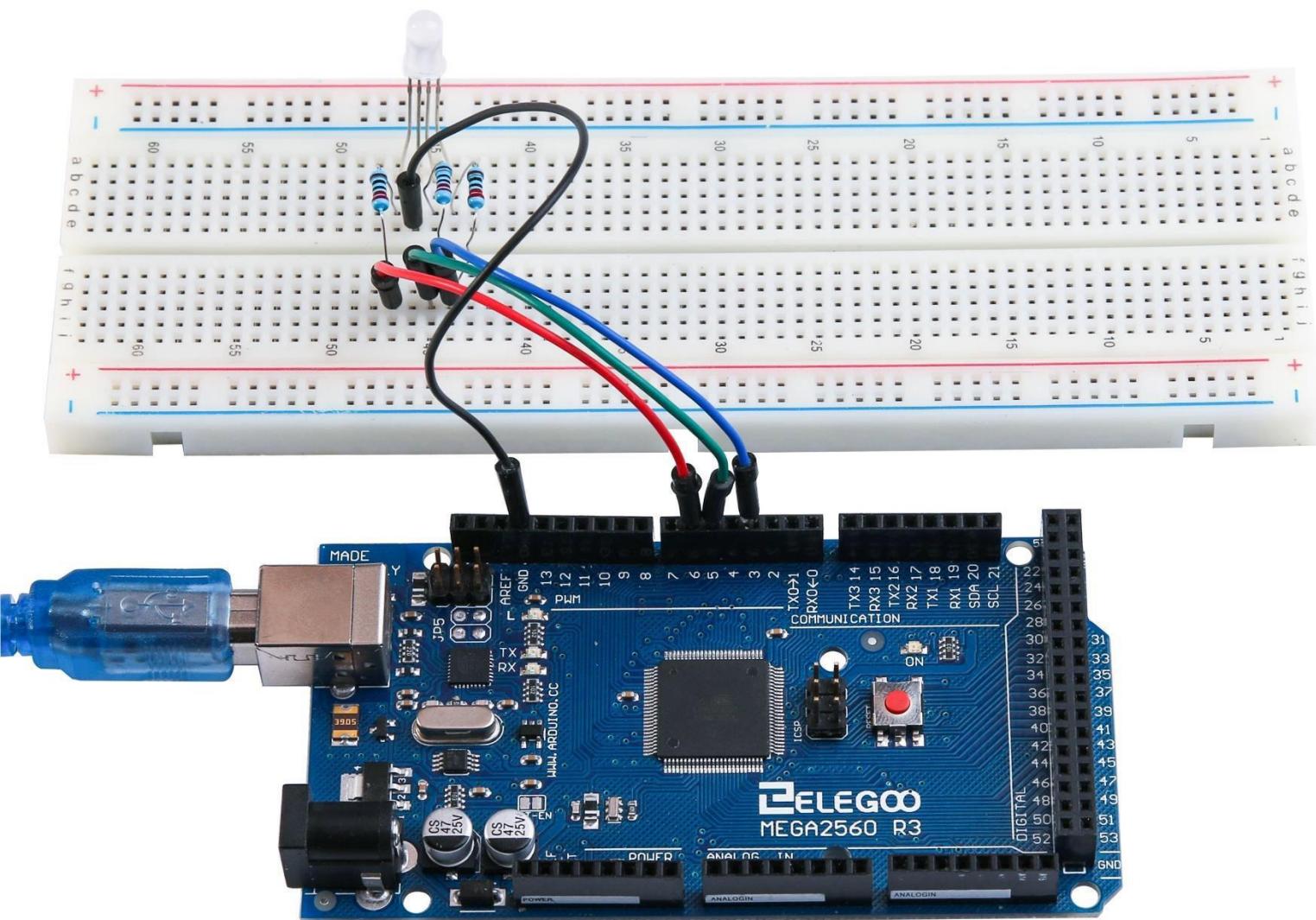
Diese Funktion beinhaltet drei Argumente. Eine für die Helligkeit der roten, grünen und blauen LEDs. In jedem Fall liegt die Zahl im Bereich von 0 bis 255, wobei 0 aus bedeutet und 255 bedeutet maximale Helligkeit. Die Funktion ruft dann "analogWrite" auf, um die Helligkeit jeder LED einzustellen.

Wenn man sich die "Loop" -Funktion anschaut, sieht man, dass wir die Menge bzw. Helligkeit an rotem, grünem und blauem Licht einstellen, dann anzeigen und dann für eine Sekunde pausieren wollen, bevor wir auf die nächste Farbe übergehen.

```
#define delayTime 10 // fading time between colors  
delay(delayTime);
```

Versuchen Sie selbst mal ein paar Farben einzustellen, um den Sketch zu verinnerlichen und beobachten Sie die Wirkung auf Ihre LED.

Beispelfoto



Lektion 5 Digitale Eingänge

Überblick

In dieser Lektion lernen Sie, mit den Digitaleingängen Drucktasten verwenden, um eine LED ein- und auszuschalten.

Durch Drücken der Taste wird die LED eingeschaltet; Durch Drücken der anderen Taste wird die LED ausgeschaltet.

Erforderliche Komponente:

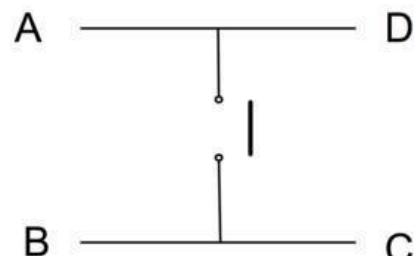
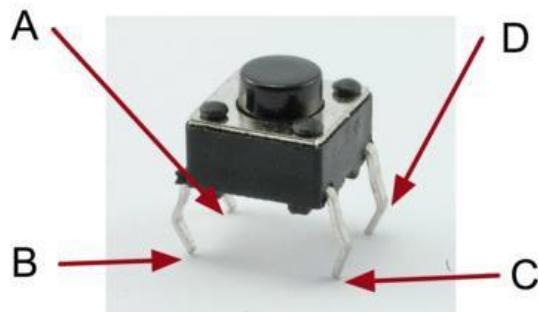
- (1) x Elegoo Mega2560 R3
- (1) x 830 Tie-points Breadboard
- (1) x 5mm rote LED
- (1) x 220 Ohm Widerstand
- (2) x Druckschalter
- (7) x M-M wires (Male to Male Kabel)

Komponenten Einführung:

Druckschalter:

Schalter sind wirklich einfache Komponenten. Wenn Sie einen Knopf drücken oder einen Hebel umdrehen, verbinden sie zwei Kontakte miteinander, so dass Strom durch sie hindurchfließen kann.

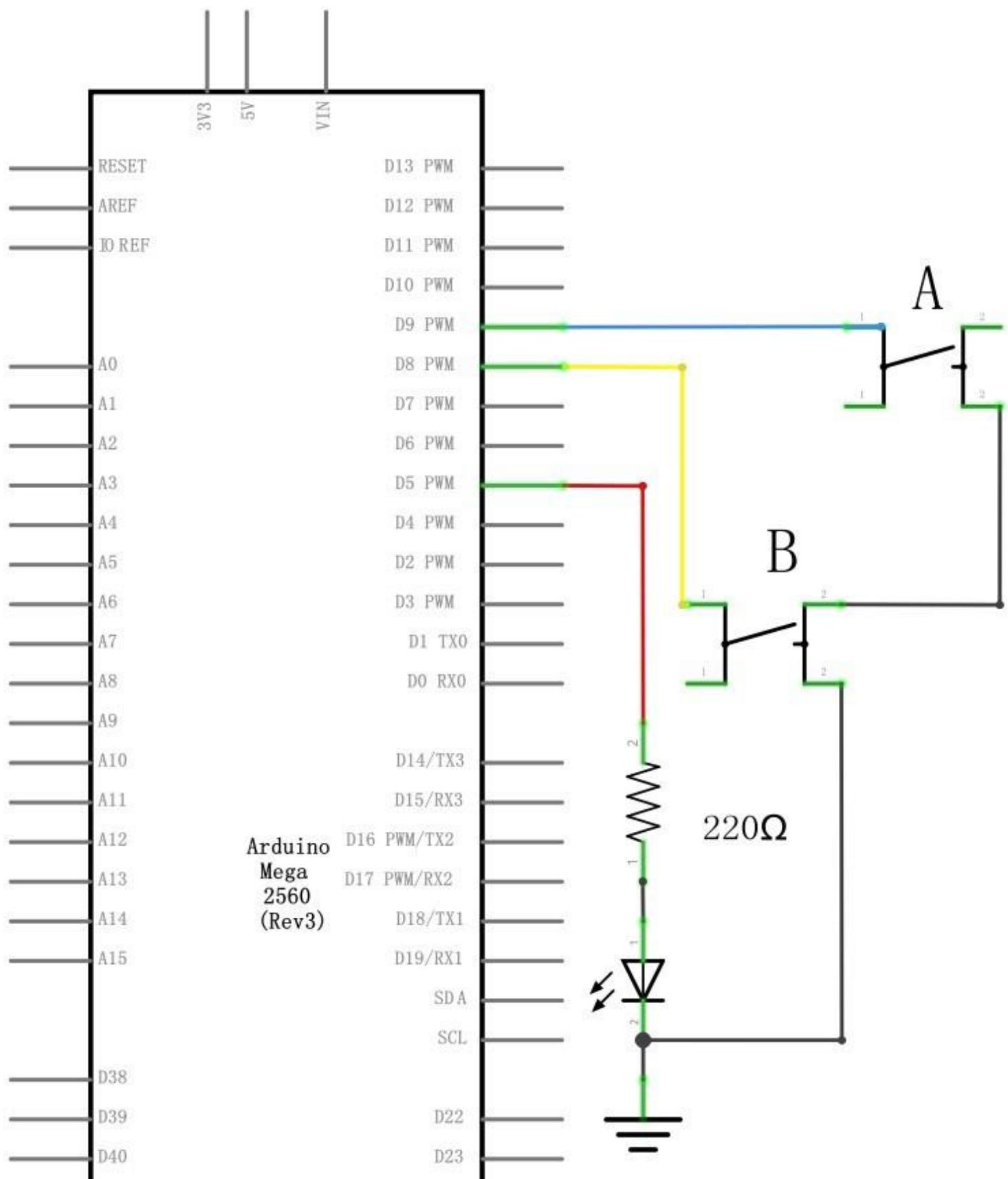
Die kleinen Schalter, die in dieser Lektion verwendet werden, haben vier Verbindungen, die ein wenig verwirrend sein können.



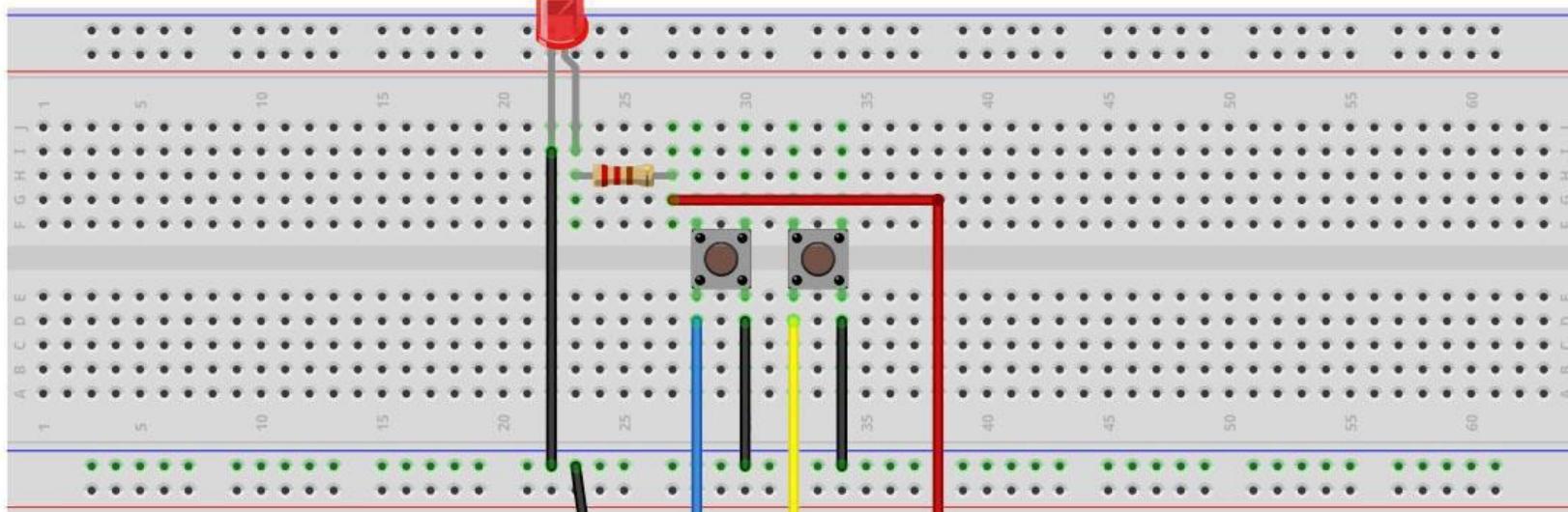
Eigentlich gibt es nur wirklich zwei elektrische Verbindungen. Innerhalb des Schalters sind die Pins B und C miteinander verbunden, ebenso wie A und D.

Verbindung

Schema



Schaltplan



Obwohl das Gehäuse des Schalters quadratisch ist, stehen die Stifte von gegenüberliegenden Seiten des Schalters vor. Das bedeutet, dass die Stifte nur weit genug auseinander liegen, wenn sie korrekt auf dem Steckbrett platziert sind.

Denken Sie daran, dass die LED die kürzere negative Führung nach rechts haben muss.

Code

Nach der Verdrahtung öffnen Sie bitte das Programm im Code-Ordner - Lektion 5 Digitale Eingänge und drücken Sie auf UPLOAD, um das Programm hochzuladen. Wenn Fehler auftauchen, siehe Lektion 2 für Details über das Tutorial zum Programm-Upload.

Laden Sie den Sketch auf Ihr MEGA2560 Board. Durch Drücken der linken Taste wird die LED eingeschaltet, während die rechte Taste ausgeschaltet wird.

Der erste Teil des Sketches definiert drei Variablen für die drei Pins, die verwendet werden sollen. Der 'ledPin' ist der Ausgangspin und 'buttonApin' bezieht sich auf den Schalter der näher an der Oberseite des Breadboards ist und 'buttonBpin' zum anderen Schalter.

Die Funktion 'Setup' definiert die LEDPin als OUTPUT wie in der vorigen Lektion, aber jetzt haben wir die beiden Eingänge zu definieren. In diesem Fall verwenden wir die Funktion den PinMode als 'INPUT_PULLUP' zu deklarieren:

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

Der Pin-Modus von INPUT_PULLUP bedeutet, dass der Pin als Eingang verwendet werden soll, aber wenn nichts anderes an den Eingang angeschlossen ist, sollte er auf HIGH (5V) gesetzt werden. Mit anderen Worten, der Standardwert für die Eingabe ist HIGH, es sei denn, es wird durch Drücken der Taste geändert.

Aus diesem Grund sind die Schalter mit GND verbunden. Wenn ein Schalter gedrückt wird, verbindet er den Eingangssteckplatz mit GND, so dass er nicht mehr HIGH ist.

Da der Eingang normalerweise HIGH ist und nur dann LOW geht, wenn die Taste gedrückt wird, ist die Logik etwas verdreht. Wir werden das in der 'Loop'-Funktion beachten.

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

```

}

if (digitalRead(buttonBPin) == LOW)
{
    digitalWrite(ledPin, LOW);
}

}

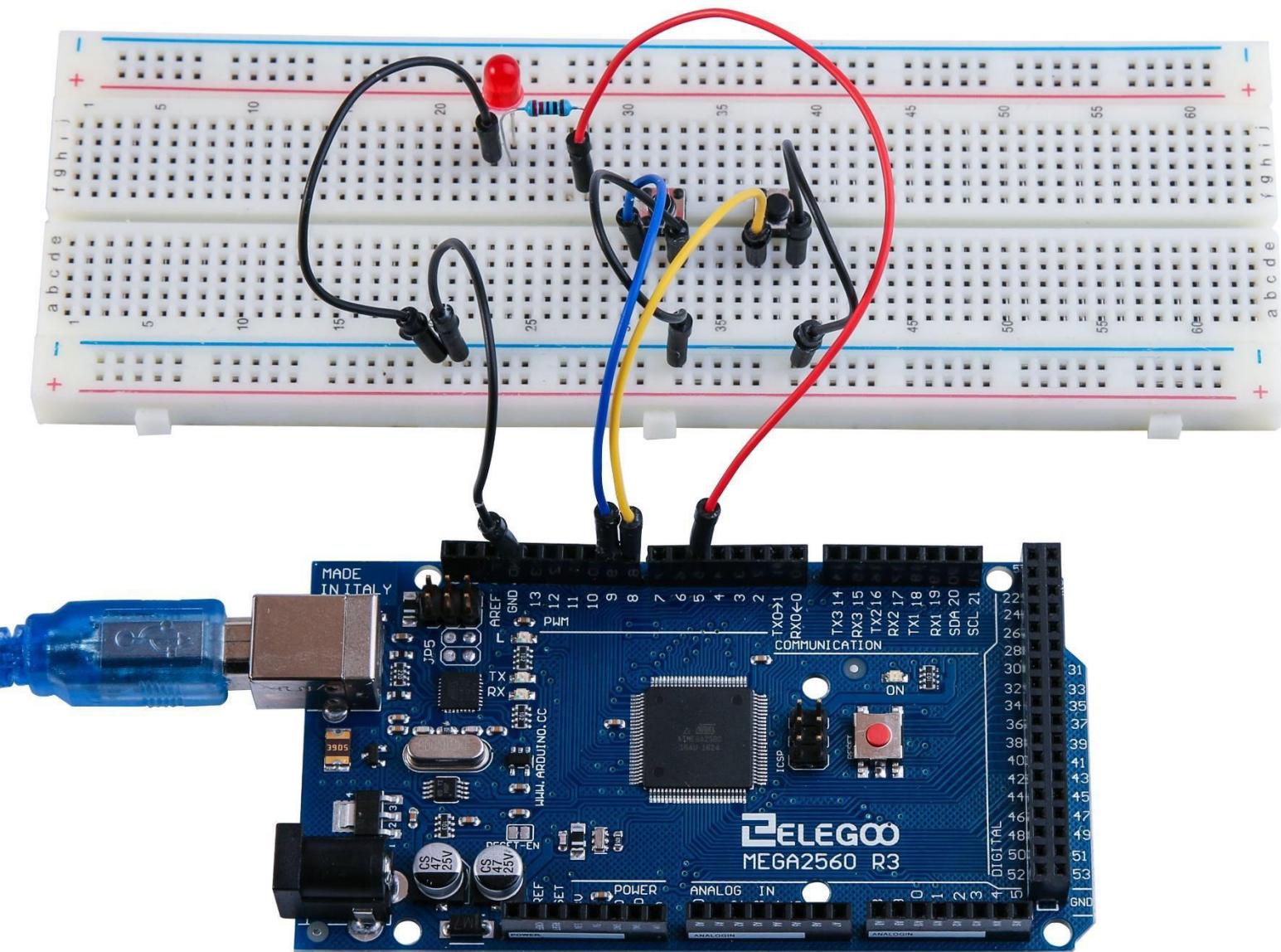
```

In der 'loop' Funktion gibt es zwei 'if' Anweisungen. Eine für jede Taste. Jede Anweisung macht ein 'digitalRead' auf dem entsprechenden Eingang.

Denken Sie daran, dass, wenn die Taste gedrückt wird, die entsprechende Eingabe LOW ist, wenn die Taste A gedrückt ist, dann schaltet ein "digitalWrite" auf dem ledPin ein.

Ähnlich, wenn Taste B gedrückt wird, wird ein LOW in das ledPingeschrieben.

Beispieldfoto



Lektion 6 Aktiver Summer

Überblick

In dieser Lektion lernen Sie, wie man einen Sound mit einem aktiven Summer erzeugt.

Erforderliche Komponente:

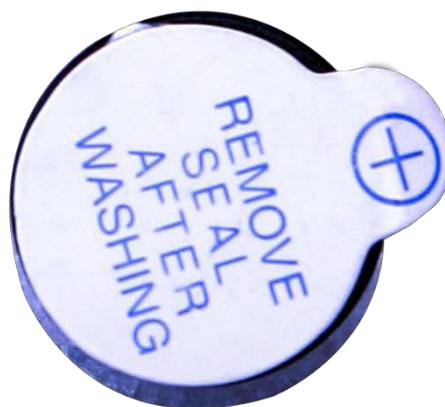
- (1) x Elegoo Mega2560 R3
- (1) x Aktiver Summer
- (2) x F-M Kabel (Female to Male Kabel)

Komponenten Einführung:

Summer:

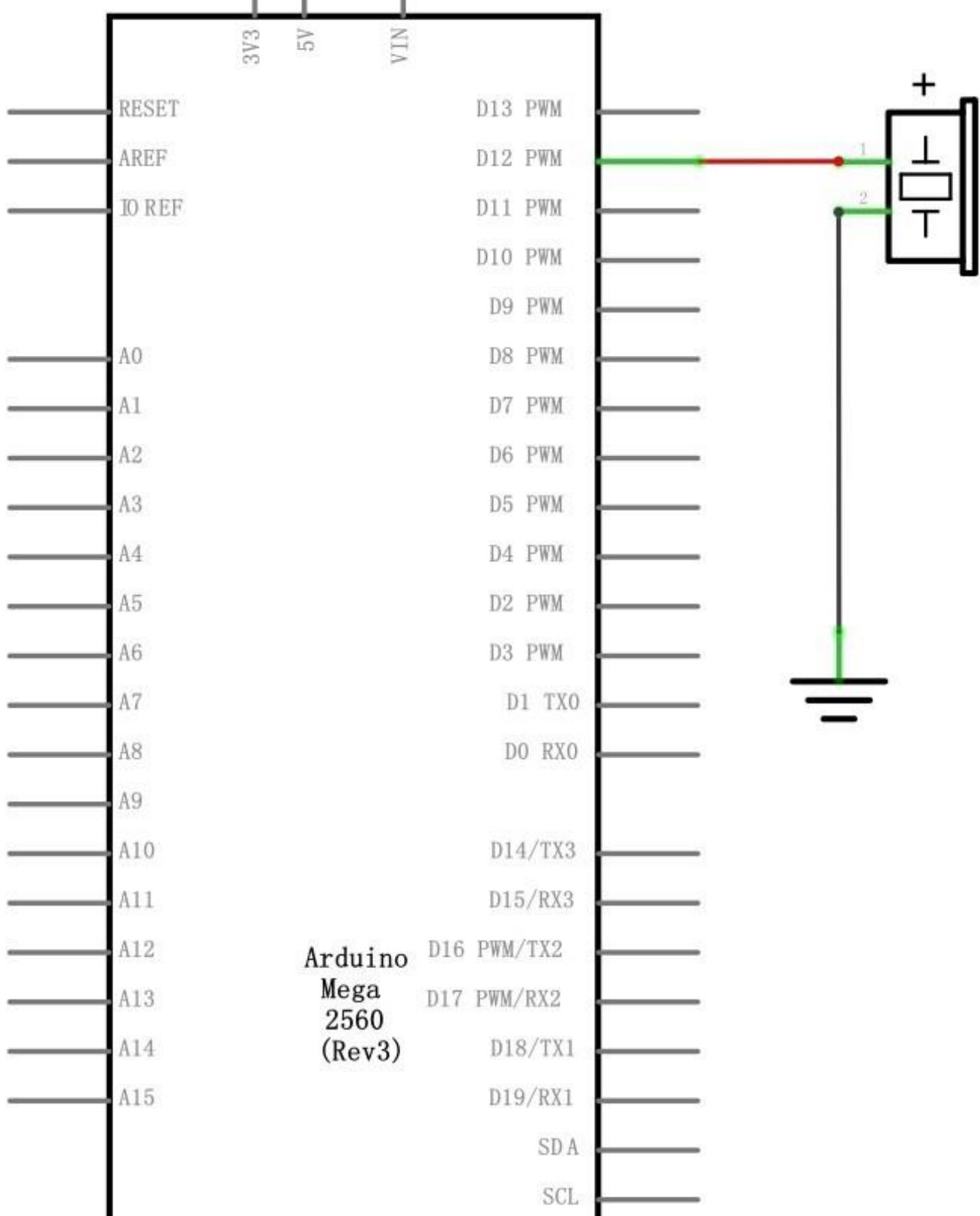
Elektronische Summer werden mit Gleichstrom betrieben und sind mit einem integrierten Schaltkreis ausgestattet. Sie sind weit verbreitet in Computern, Druckern, Fotokopierern, Alarmen, elektronischen Spielzeug, Automobil-Elektronik, Telefone, Timer und andere elektronische Produkte die für Sprachgeräte verwendet werden. Es gibt passive und active Summer. Drehen Sie die Anschlüsse von den zwei mitgelieferten Summern nach oben. Der eine mit einer grünen Platine ist ein passiver Summer, während der andere mit einem schwarzen "Klebeband" eingeschlossen ist.

Der Unterschied zwischen den beiden ist, dass ein aktiver Summer eine eingebaute oszillierende Quelle hat, so dass es einen Ton erzeugt, wenn er elektrisiert wird. Ein passiver Summer hat keine solche Quelle, also wird er kein Ton von sich geben, wenn ein DC-Signal angelegt wird. Stattdessen müssen Sie quadratische Signale (TTL) verwenden, deren Frequenz zwischen 2K und 5K liegt, damit der Summer funktioniert. Der aktive Summer ist oft teurer als der passive, da hier mehrere Schwingkreise eingebaut sind.

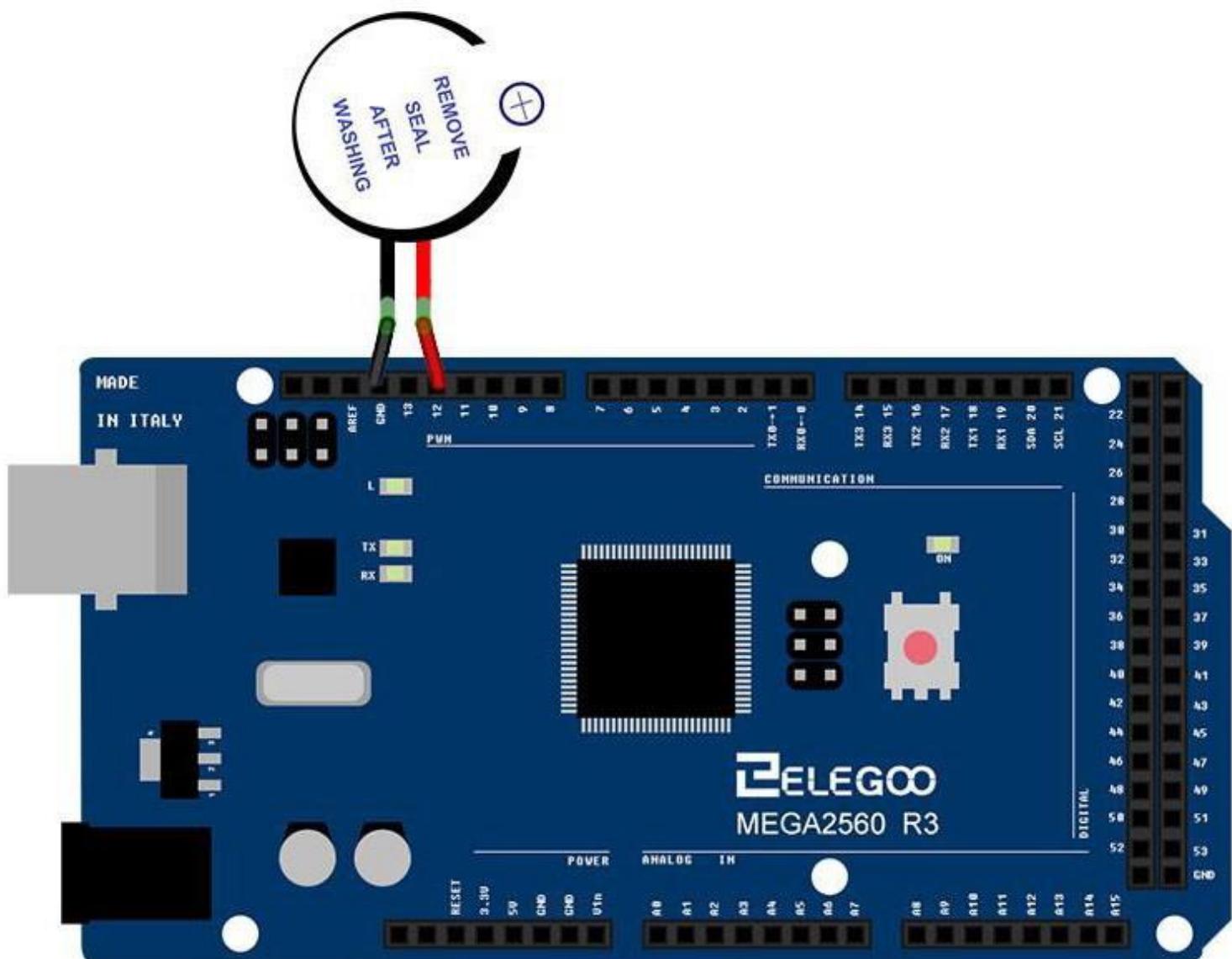


Verbindung

Schema



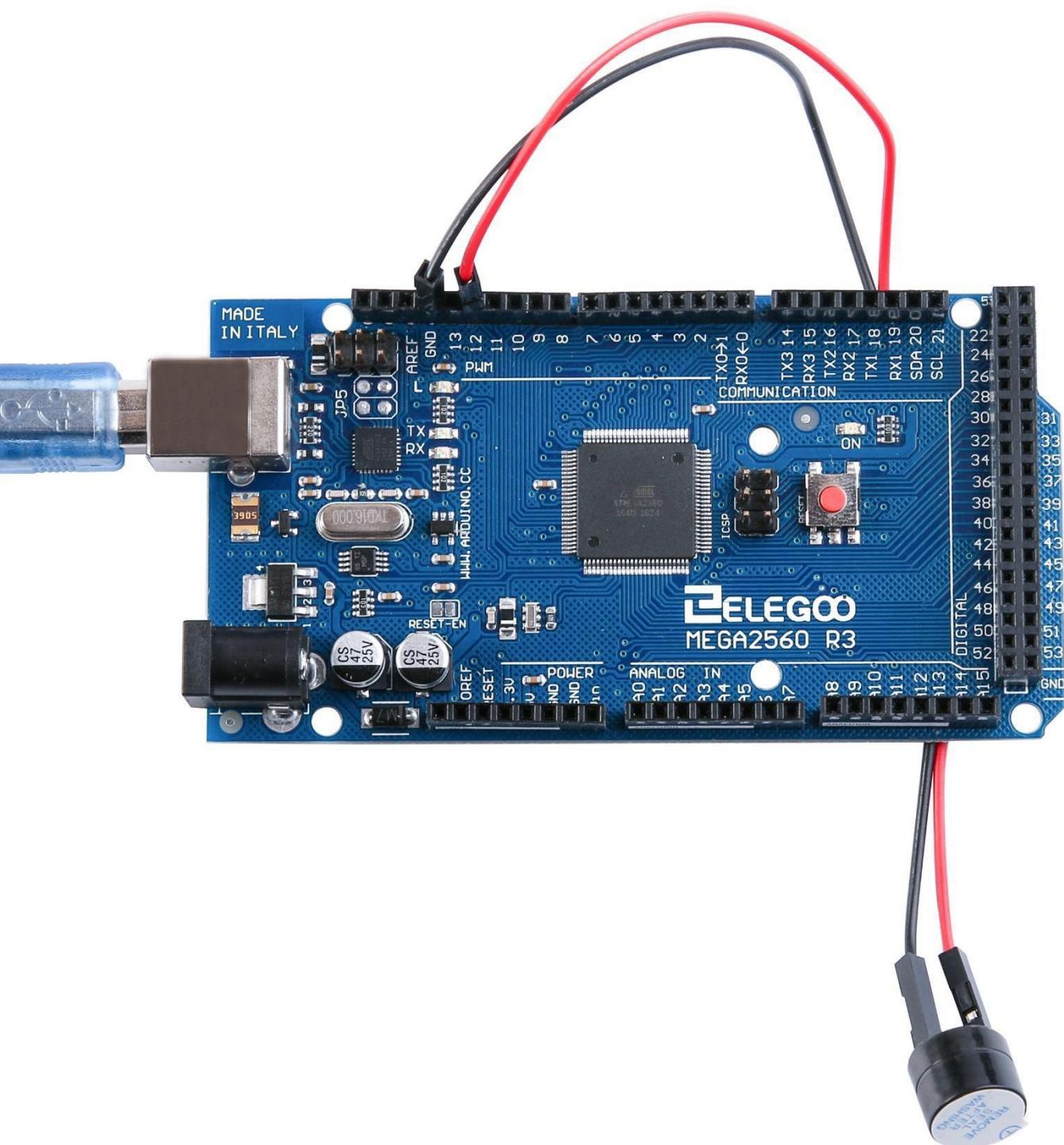
Schaltplan



Code

Nachdem sie alles korrekt angeschlossen haben, öffnen Sie bitte das Programm im Code-Ordner - Lektion 6 Making Sounds und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, falls Fehler vorliegen.

Beispielfoto



Lektion 7 Passiver Summer

Überblick

In dieser Lektion lernen Sie, wie man einen passiven Summer benutzt.

Der Zweck des Experiments besteht darin, acht verschiedene Klänge zu erzeugen, wobei jeder Klang 0,5 Sekunden dauert: von Alto Do (523 Hz), Re (587 Hz), Mi (659 Hz), Fa (698 Hz), So (784 Hz), La (880 Hz) Si (988Hz) bis Treble Do (1047Hz).

Erforderliche Komponente:

- (1) x Elegoo Mega2560 R3
- (1) x Passiver Summer
- (2) x F-M Kabel (Female to Male Kabel)

Komponenten Einführung:

Passiver Summer:

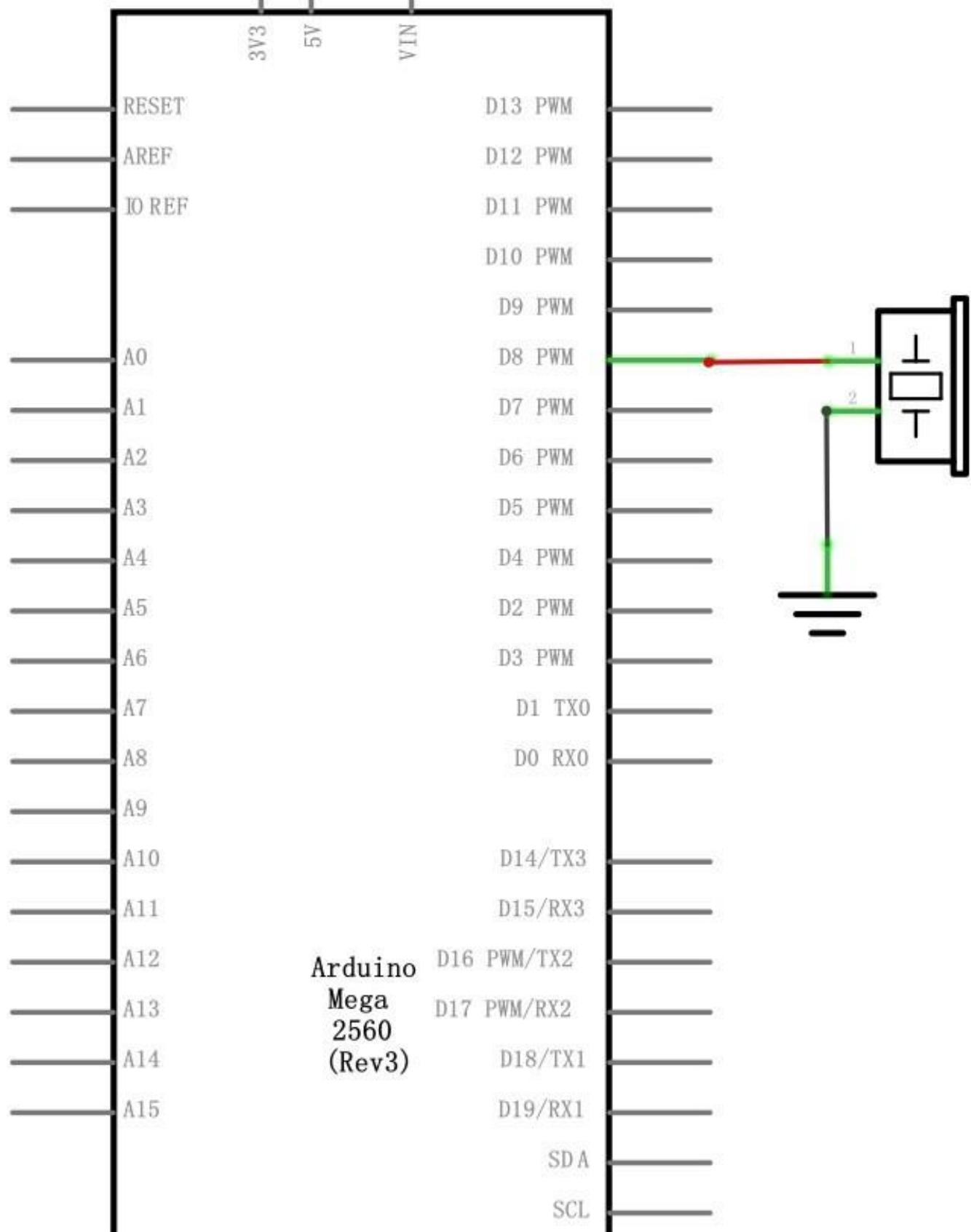
Das Funktionsprinzip des passiven Summers ist das verwenden von PWM, welches die Töne erzeugt, indem es die Luft zum "vibrieren" bringt. Wenn man die Schwingungsfrequenz richtig verändert, kann man so verschiedene Töne erzeugen. Zum Beispiel, durch senden eines Pulses von 523Hz kann man ein Alto Do erzeugen, mit einer Pulsfrequenz von 587Hz kann man Mittelton Re erzeugen, eine Pulsfrequenz von 659Hz, produziert ein Mi. Durch solch eine Pulsfrequenzveränderung können Sie mit dem Summer ein Lied spielen.

Wir sollten darauf achten dass wir die MEGA2560 R3-Platine analog Write () nicht verwenden, um unseren Impuls an den Summer zu erzeugen, da der Impulsausgang des Analog Write () konstant ist (500Hz).

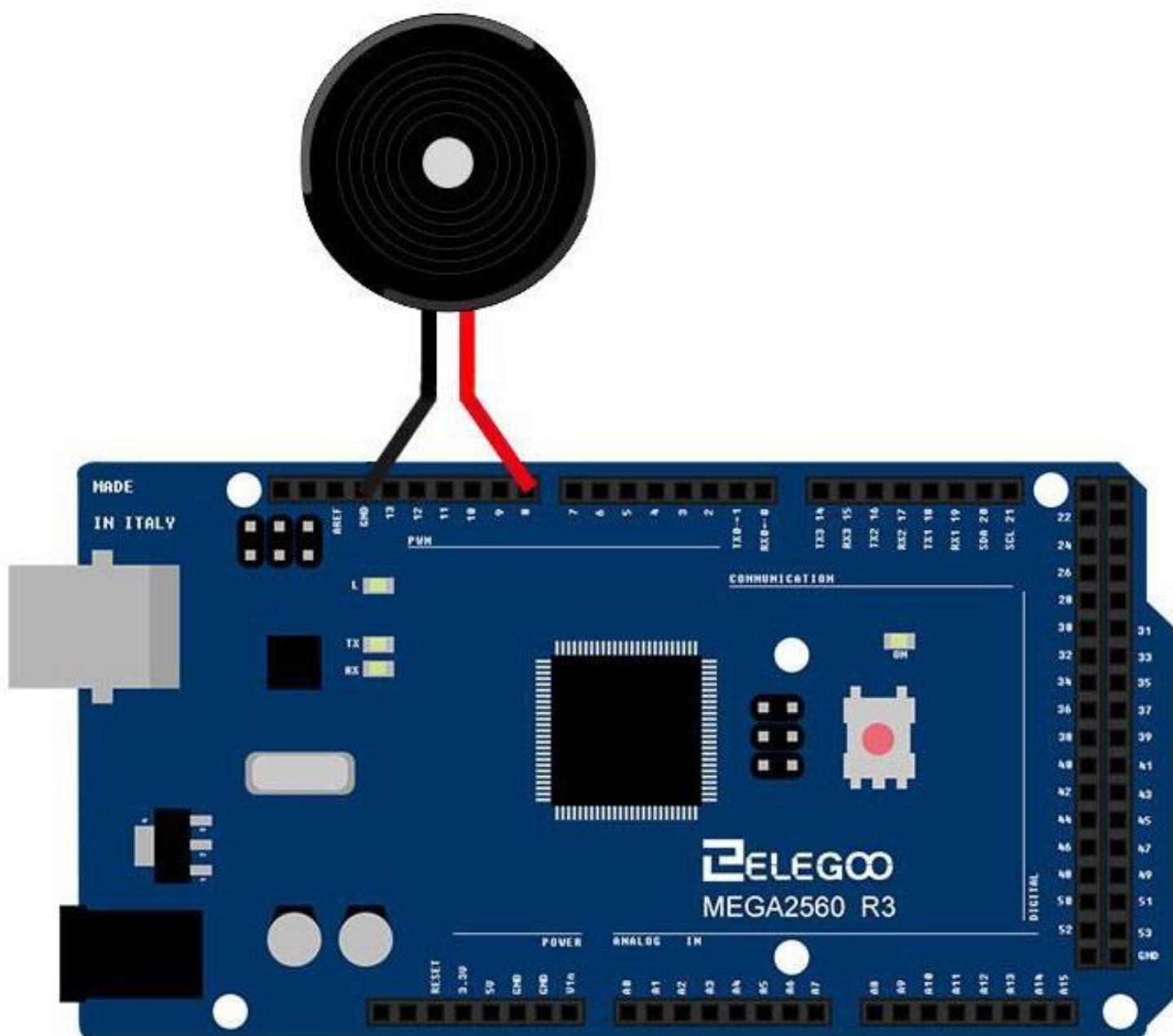


Verbindung

Schema



Schaltplan



Schließen Sie den Summer mit dem roten Kabel (positiv) an Pin8 der MEGA2560 R3-Platine, das schwarze Kabel (negativ) zum GND(Masse).

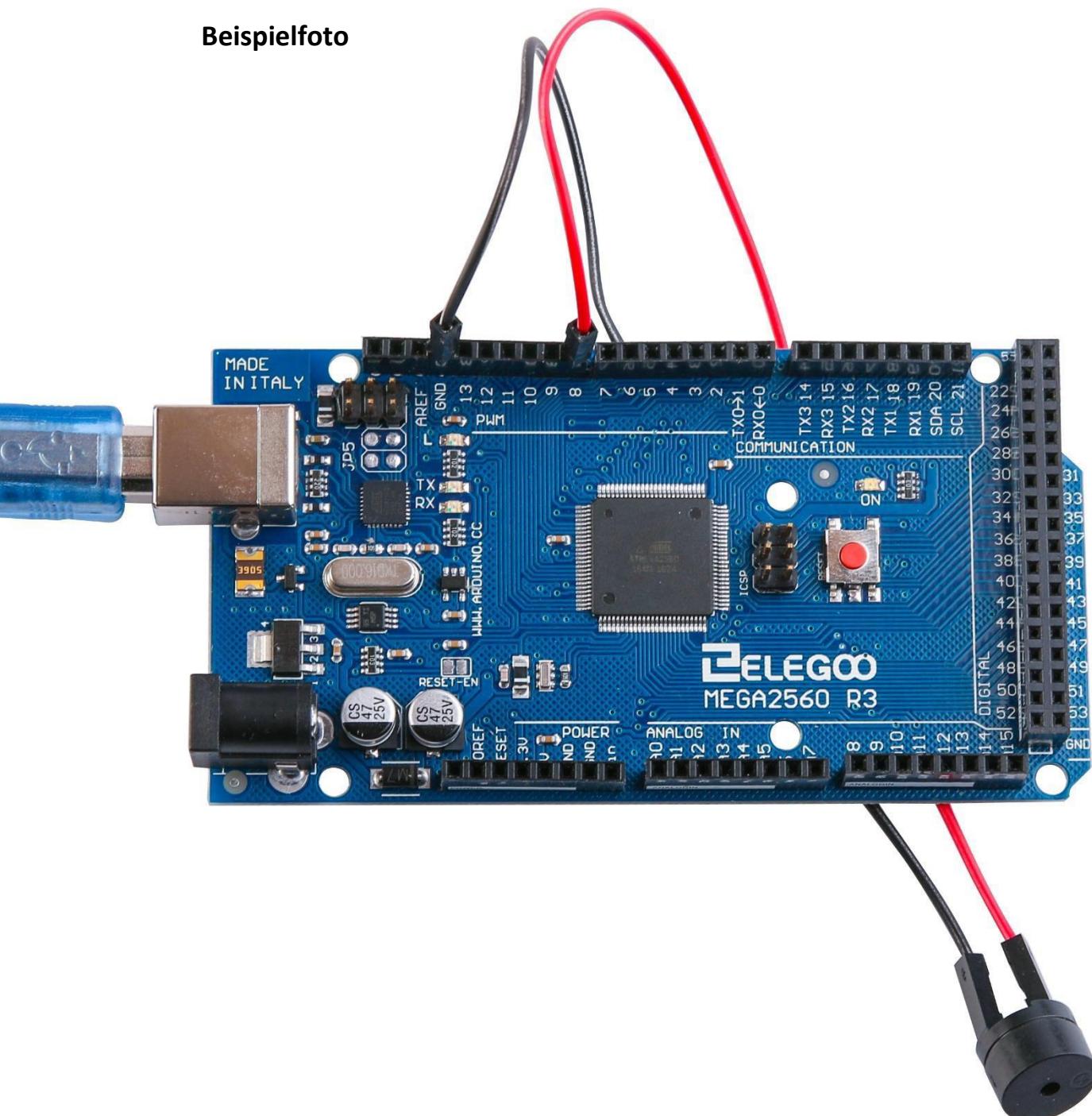
Code

Nach dem Verschalten öffnen Sie bitte das Programm im Codeordner - Lektion 7 Passiver Summer und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die <pitches>-Bibliothek installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion 1.

Beispieldfoto



Lektion 8 Kippschalter

Überblick

In dieser Lektion lernen Sie, wie man einen Neigungskugelschalter benutzt, um einen kleinen Neigungswinkel zu erkennen.

Erforderliche Komponente:

- (1) x Elegoo Mega2560 R3
- (1) x Kippschalter
- (2) x F-M Kabel (Female to Male DuPont Kabel)



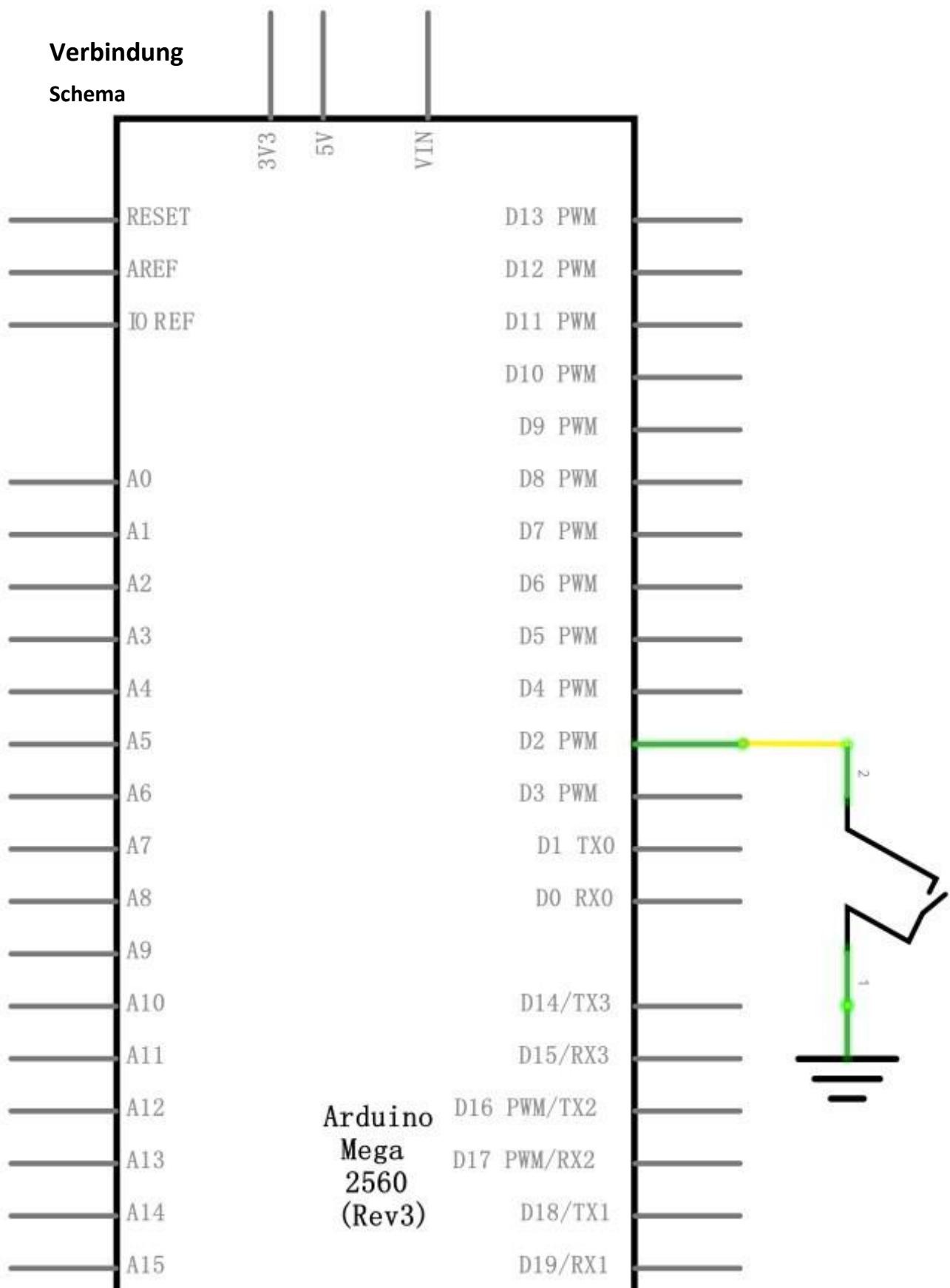
Komponenten Einführung:

Kippschalter:

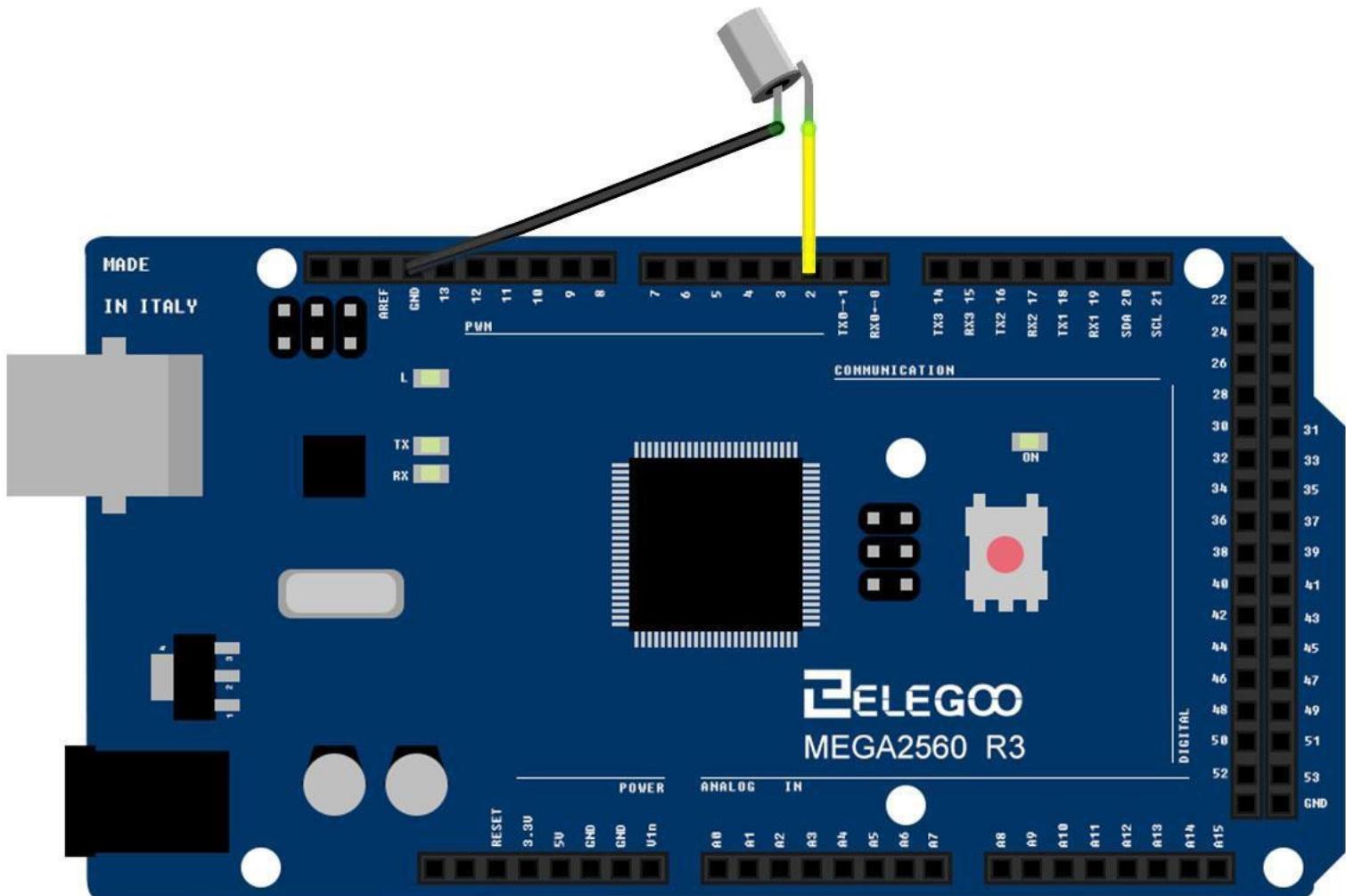
Neigungssensoren (Neigungskugelschalter) können die Orientierung bzw. Neigung erkennen. Sie sind klein, preiswert, leistungsstark und einfach zu bedienen. Wenn sie richtig verwendet werden, werden sie nicht verschleien. Ihre Einfachheit macht sie beliebt für Spielzeug, Gadgets und Geräte. Manchmal werden sie auch als "Quecksilberschalter", "Kippschalter" oder "Rollkugel-Sensoren" bezeichnet.

Sie bestehen gewöhnlich aus einem zylindrischem Hohlraum (zylindrisch ist beliebt, wenn auch nicht immer) mit einer leitfähigen freien Masse im Inneren, wie zum Beispiel ein Tropfen von Quecksilber oder eine Rollkugel. Ein Ende des Hohlraums hat zwei leitfähige Elemente (Pole). Wenn der Sensor so orientiert ist, dass dieses Ende nach unten gerichtet ist, rollt die Masse auf die Pole, schließt sie und wirkt so als Schalter.

Während dieser Sensor nicht so präzise oder flexibel ist wie ein voller Beschleunigungsmesser, können Kippschalter einfach und kostengünstig Bewegungen oder Orientierungen erkennen. Ein weiterer Vorteil ist, dass die größere Schalter ihre eigenen Strom schalten können. Beschleunigungsmesser hingegen, geben digitale oder analoge Spannung aus, die dann mit Hilfe von zusätzlichen Schaltungen analysiert werden müssen.



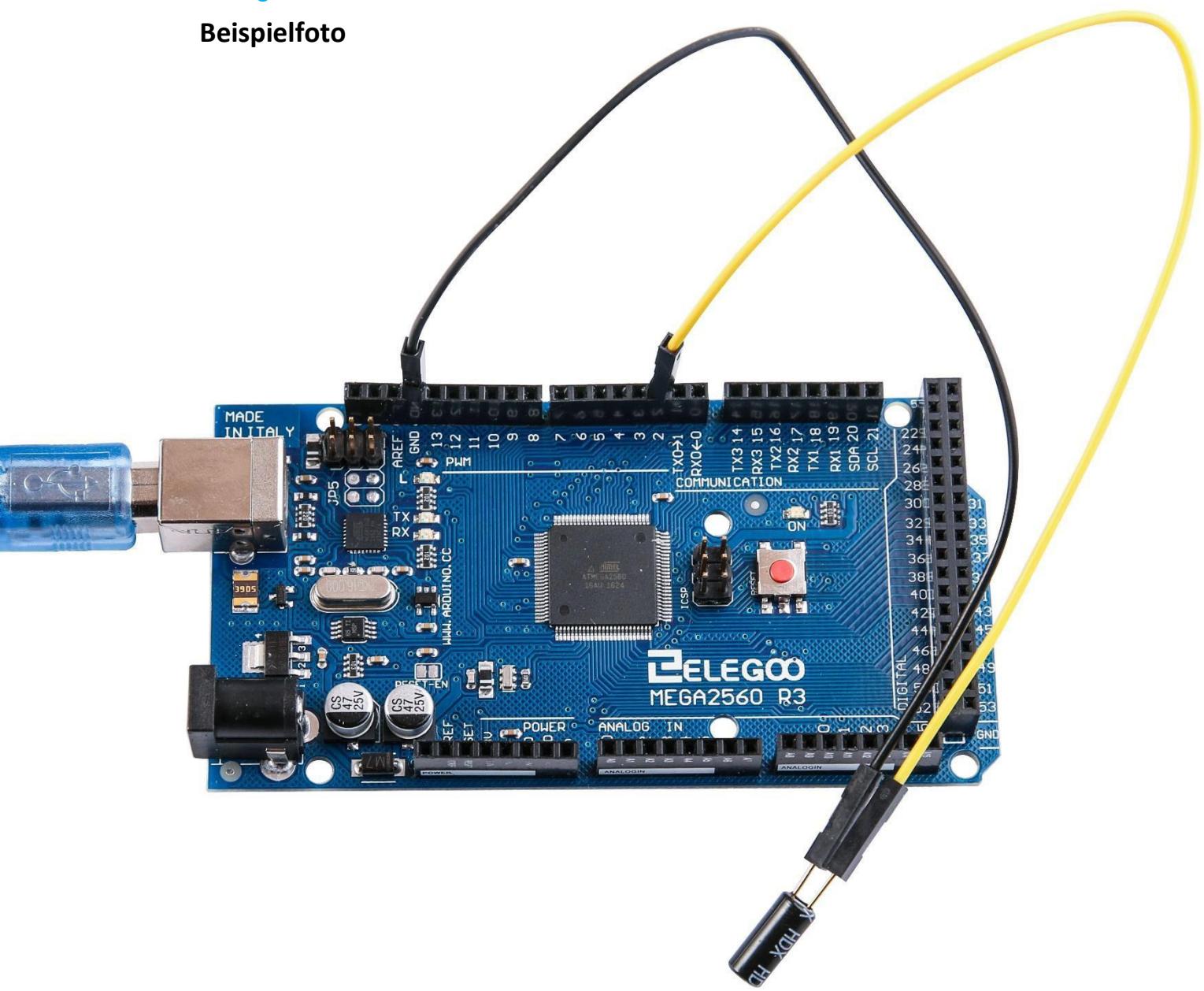
Schaltplan



Code

Nach dem Verschalten öffnen Sie bitte das Programm im Code-Ordner - Lektion 8 Kippschalter und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Beispielfoto



Lektion 9 Servo

Überblick

Servos sind eine Art von Getriebemotor, die sich nur um 180 Grad drehen können. Sie werden durch das Senden elektrischer Impulse von der Mega R3 Platine gesteuert. Diese Impulse sagen dem Servo auf welche Position es sich bewegen soll. Der Servo hat drei Drähte von denen der braune der Erdleiter (Masse) ist und sollte an dem GND-Port der Platine angeschlossen werden, der rote ist das Stromkabel und sollte an den 5V-Port angeschlossen werden, und der orange ist der Signalleiter Und sollte mit dem Digitalen #9 Port verbunden werden.

Benötigte Komponenten

- (1) x Elegoo Mega2560 R3
- (1) x Servo (SG90)
- (3) x M-M Kabel (Male to Male Kabel)

Komponenten Einführung

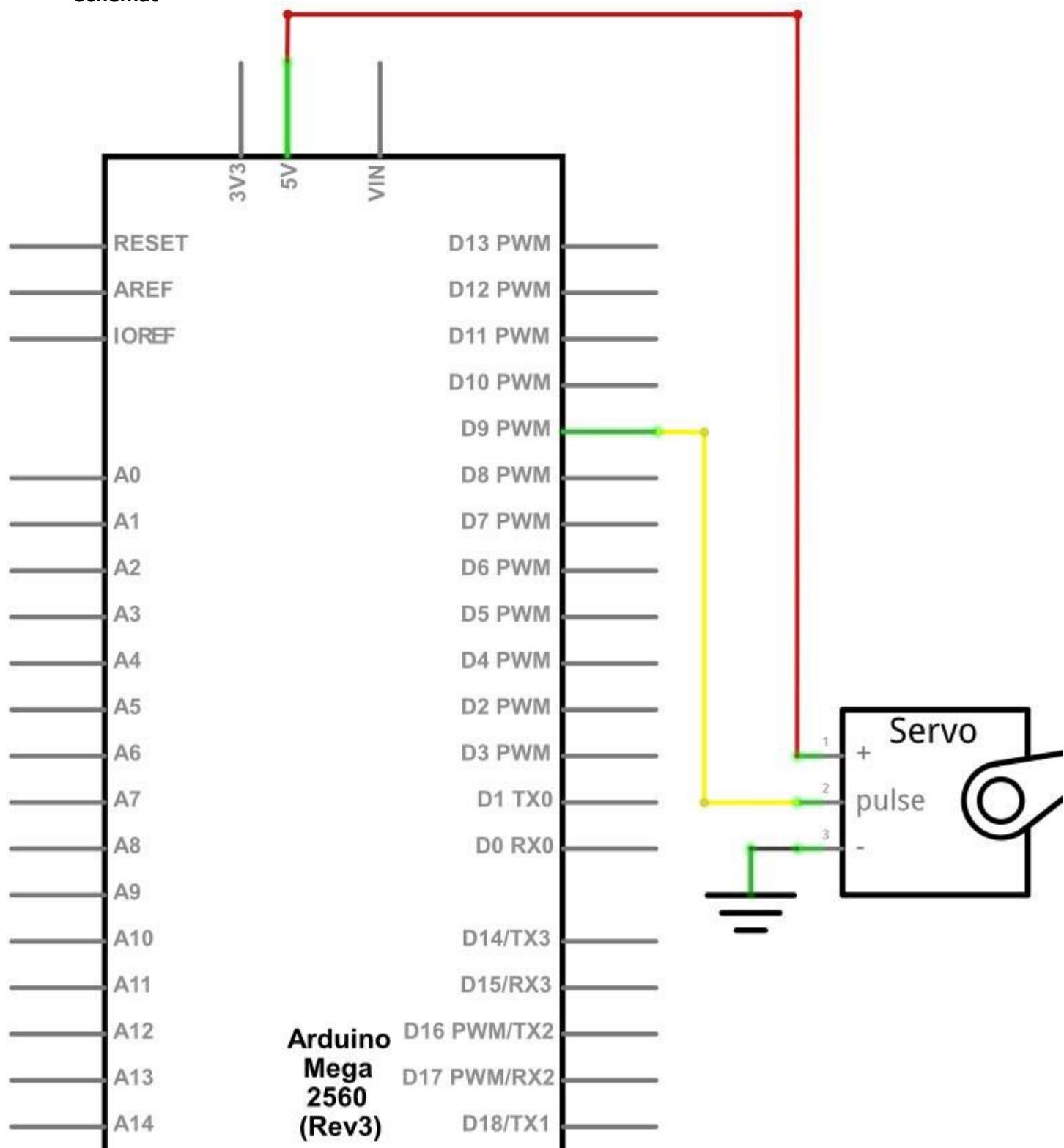
SG90

- Universell für JR und FP Stecker
- Kabellänge : 25cm
- Keine Last; Stelleschwindigkeit: 0.12 sek / 60 Grad (4.8V), 0.10 sek / 60 Grad (6.0V)
- Stellkraft (4.8V): 1.6kg/cm
- Temperatur : -30~60'C
- Bandbreite: 5us
- Betriebsspannungen: 3.5~6V
- Abmaße : 1.26 in x 1.18 in x 0.47 in (3.2 cm x 3 cm x 1.2 cm)
- Gewicht : 4.73 oz (134 g)

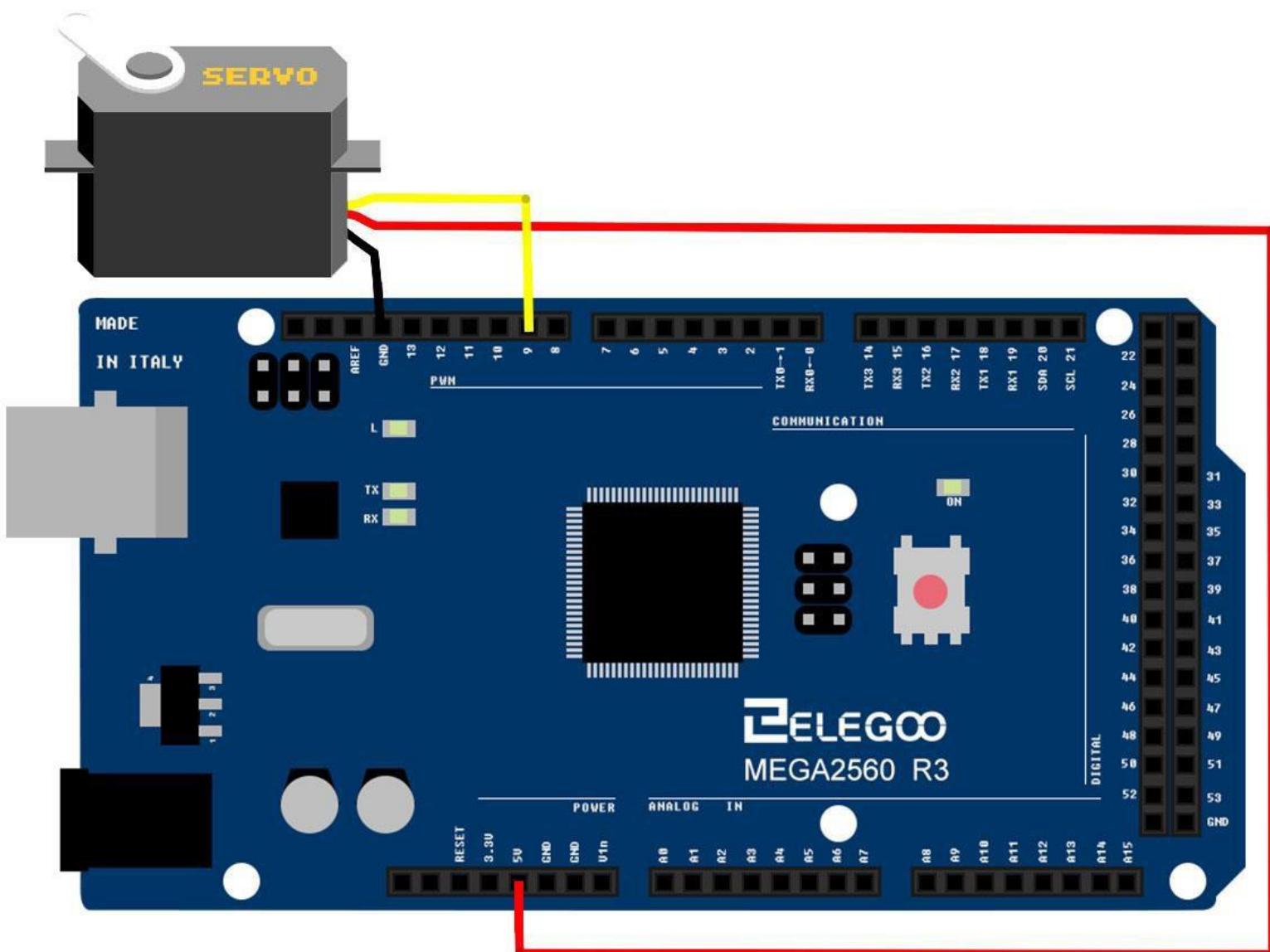


Verbindung

Schemat



Schaltplan



Code

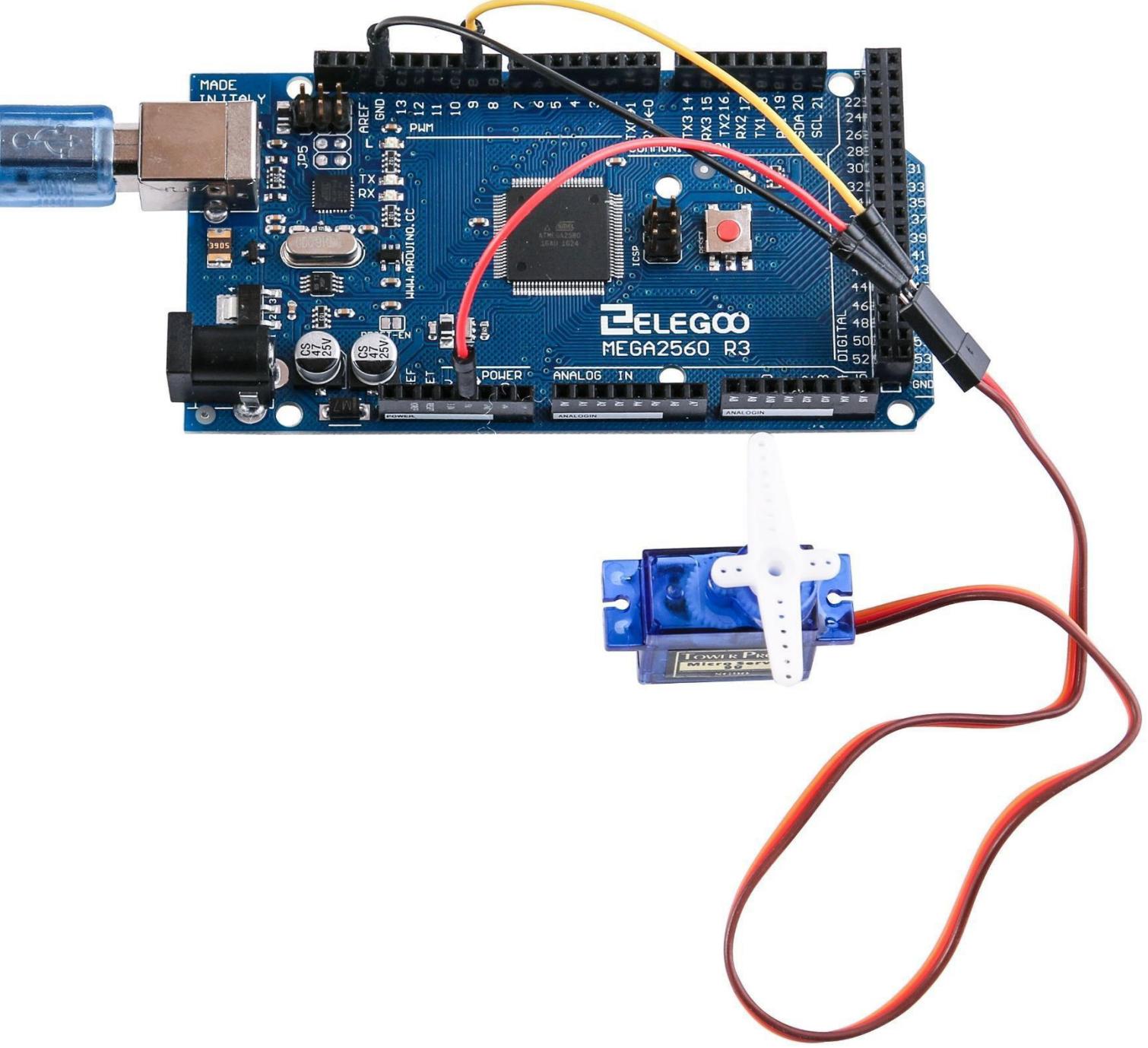
Nach der Verschaltung öffnen Sie bitte das Programm im Code-Ordner - Lektion 9 Servo und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die <Servo> -Bibliothek installiert oder ggf. neu installiert haben. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion 1.

Beispelfoto

Im Bild wird der braune Draht des Servos über die schwarzen M-M-Drähte angeschlossen, der rote über die roten M-M-Drähte angeschlossen und der orangefarbene über die gelben M-M-Drähte angeschlossen.



Lektion 10 Ultraschallsensormodul

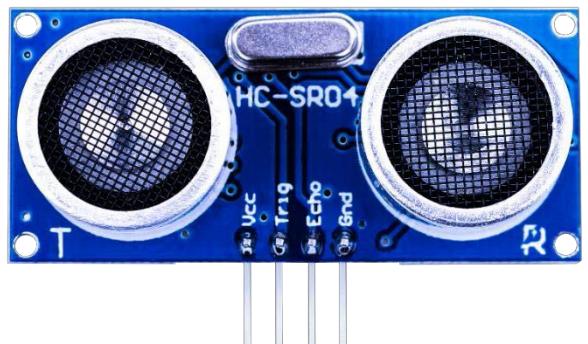
Überblick

Ultraschallsensoren sind ideal für alle Arten von Projekten, die Distanzmessungen benötigen und zum Beispiel Hindernisse erkennen sollen.

Der HC-SR04 ist preiswert und einfach zu bedienen, da wir eine speziell für diesen Sensor entwickelte Bibliothek verwenden werden.

Erforderliche Komponente:

- (1) x Elegoo Mega2560 R3
- (1) x Ultraschallsensor
- (4) x F-M Kabel (Female to Male DuPont Kabel)



Komponenten Einführung

Ultraschallsensor

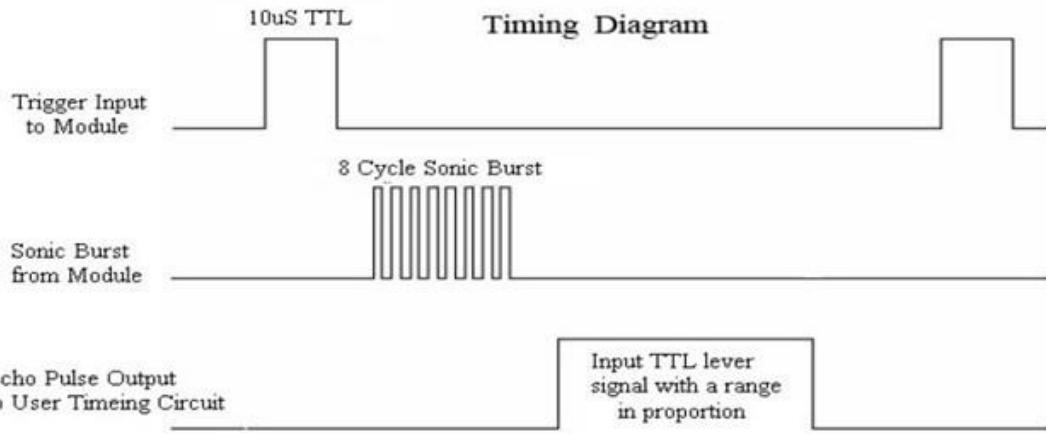
Das Ultraschallsensormodul HC-SR04 kann Reichweiten von 2cm-400cm berührungslose messen, die Reichweitengenauigkeit kann bis zu 3mm erreichen. Die Module umfassen einen Ultraschallsender, einen Empfänger und ein Steuercircus.

Das Grundprinzip des Sensors:

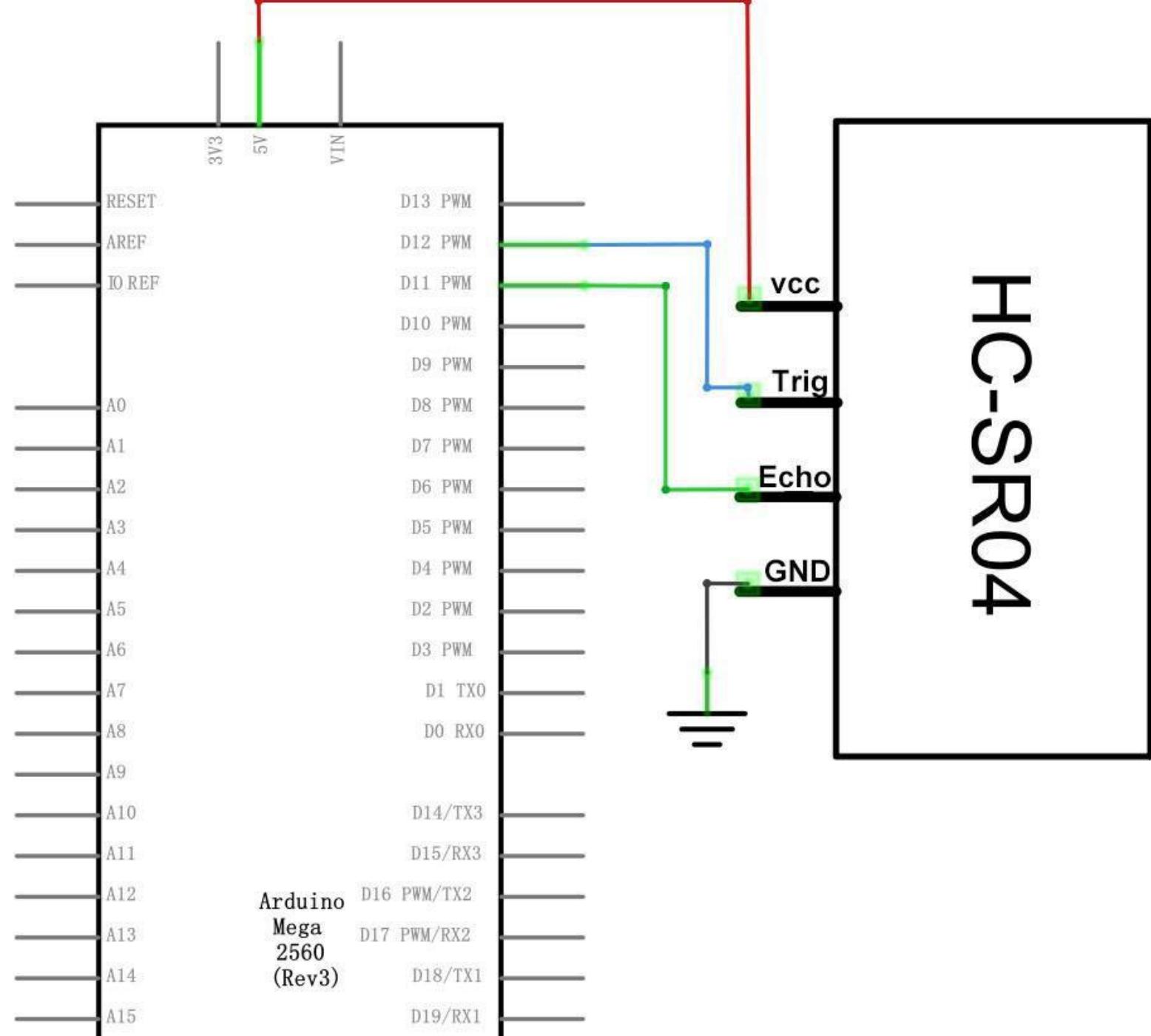
- (1) Mit einem IO Trigger liegt für mindestens $10\mu s$ ein High-Level-Signal an,
- (2) Das Modul sendet automatisch acht 40 kHz Signale und erkennt dann, ob ein Impulssignal zurück kommt.
- (3) WENN das Signal zurückkommt, gibt der Sensor aus, wie lange das Signal gebraucht hat vom aussenden, bis es wieder empfangen wurde.

$$\text{Testabstand} = (\text{Hochpegelzeit} \times \text{Schallgeschwindigkeit (340 m / s)} / 2$$

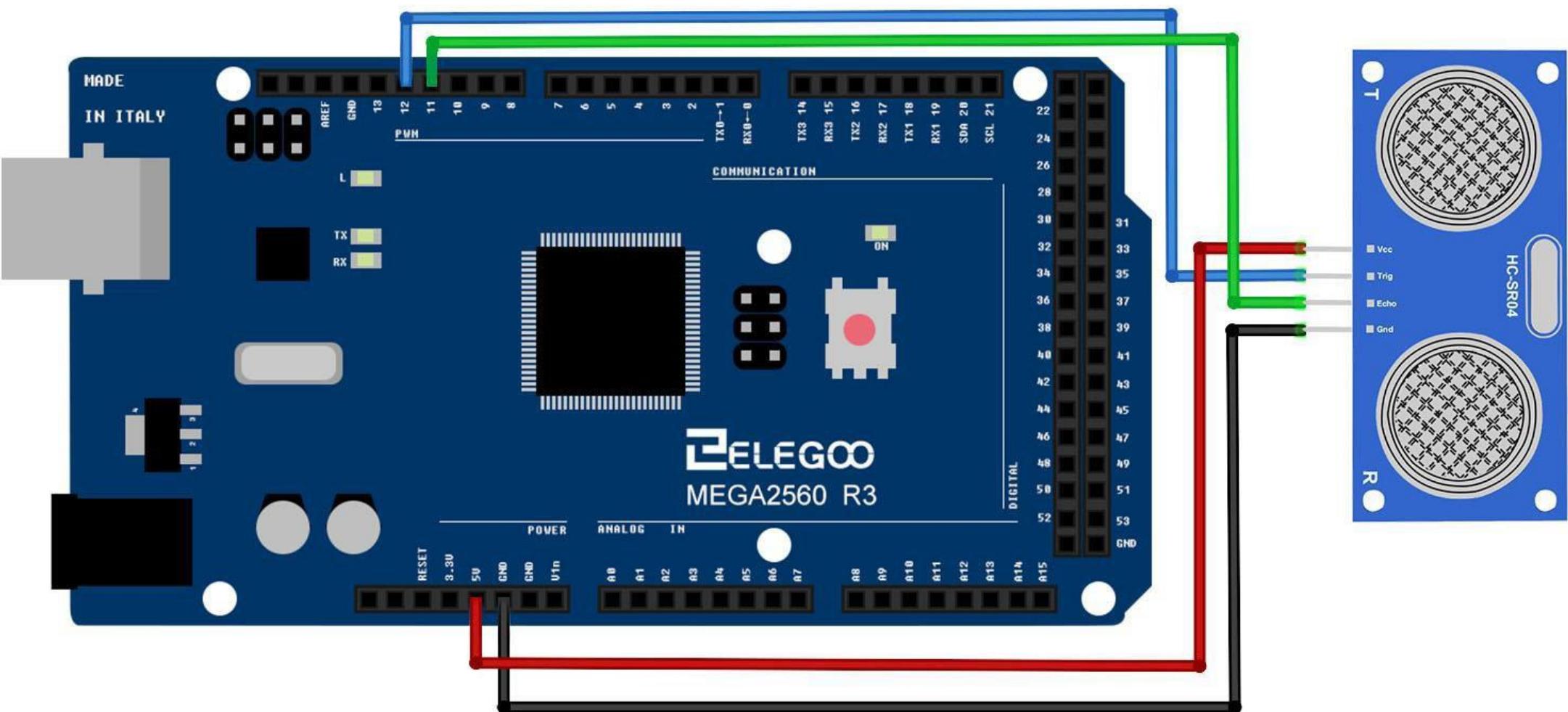
Das Zeitdiagramm ist unten dargestellt. Sie müssen nur einen kurzen Impuls von $10\mu s$ an den Triggereingang liefern, um den Bereich zu starten, und dann sendet das Modul einen 8-Zyklus-Ultraschall-Burst bei 40 kHz aus und hebt dessen Echo an. Das Echo ist ein Distanzobjekt, welches die Pulsbreite im Verhältnis des Bereichs ist. Sie können den Bereich über das Zeitintervall zwischen dem Senden des Triggersignals und dem Empfang des Echosignals berechnen. Formel: $\mu s / 58 = \text{Zentimeter}$ oder $\mu s / 147 = \text{Zoll}$; Oder: der Bereich = hohe Pegelzeit * Geschwindigkeit (340M / S) / 2; Wir empfehlen einen 60ms langen Messzyklus zu verwenden, um ein Triggersignal zum Echosignal zu verhindern.



Schema



Schaltplan



Code

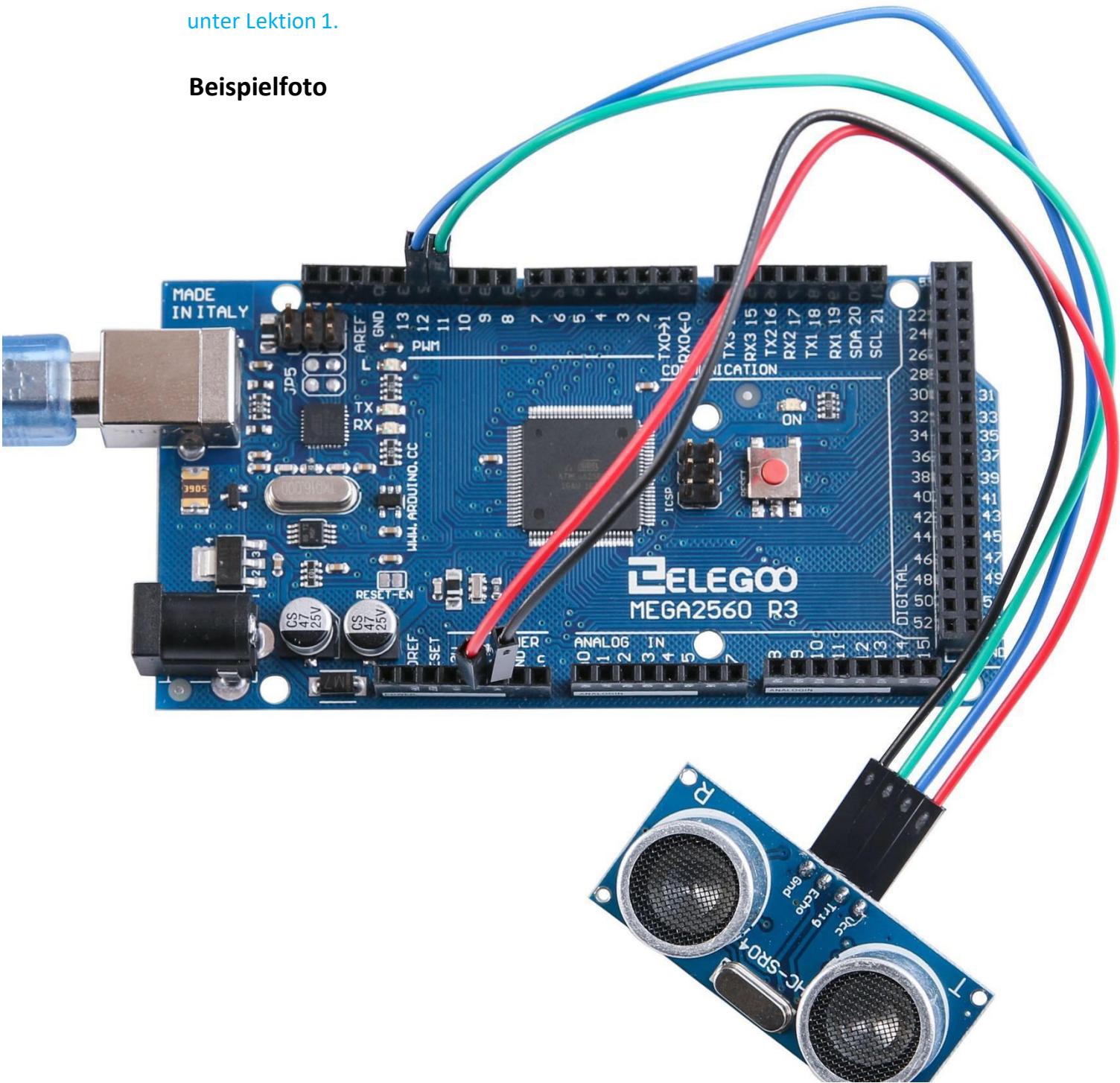
Mit einer für diese Sensoren entwickelten Bibliothek wird unser Code kurz und einfach.

Wir schließen die Bibliothek am Anfang unseres Codes ein, und dann können wir mit einfachen Befehlen das Verhalten des Sensors steuern.

Nach dem Verschalten öffnen Sie bitte das Programm im Codeordner - Lektion 10 Ultraschallsensormodul und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die Bibliothek <HC-SR04> installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren. Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion 1.

Beispielfoto



Öffnen Sie den Seriellen Monitor, dann können Sie die Daten wie unten sehen:

Klicken Sie auf die Schaltfläche Serielle Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors werden in Lektion 1 ausführlich vorgestellt.

```
4cm
124cm
125cm
126cm
125cm
125cm
126cm
127cm
125cm
125cm
125cm
126cm
125cm
126cm
125cm
126cm
12cm
125cm
125cm
8cm
126cm
8cm
10cm
11cm
0cm
5cm
0cm
5cm
178cm
126
```

Autoscroll Newline 9600 baud

Lektion 11 Membranschalter-Modul

Überblick

In diesem Projekt werden wir zeigen, wie man eine Tastatur mit einer MEGA2560 R3-Platine verbindet, damit der MEGA2560 R3 die Tasteneingabe von einem Benutzer erkennt und bearbeiten kann.

Keypads werden in allen Arten von Geräten, einschließlich Handys, Faxgeräte, Mikrowellen, Öfen, Türschlösser, etc. verwendet. Sie sind praktisch überall. Millionen von elektronischen Geräten verwenden sie für die Benutzereingabe.

Das Wissen, wie man ein Tastenfeld an einen Mikrocontroller wie ein MEGA2560 R3 Board anschließen kann ist sehr wertvoll für den Bau von vielen verschiedenen Arten von kommerziellen Produkten.

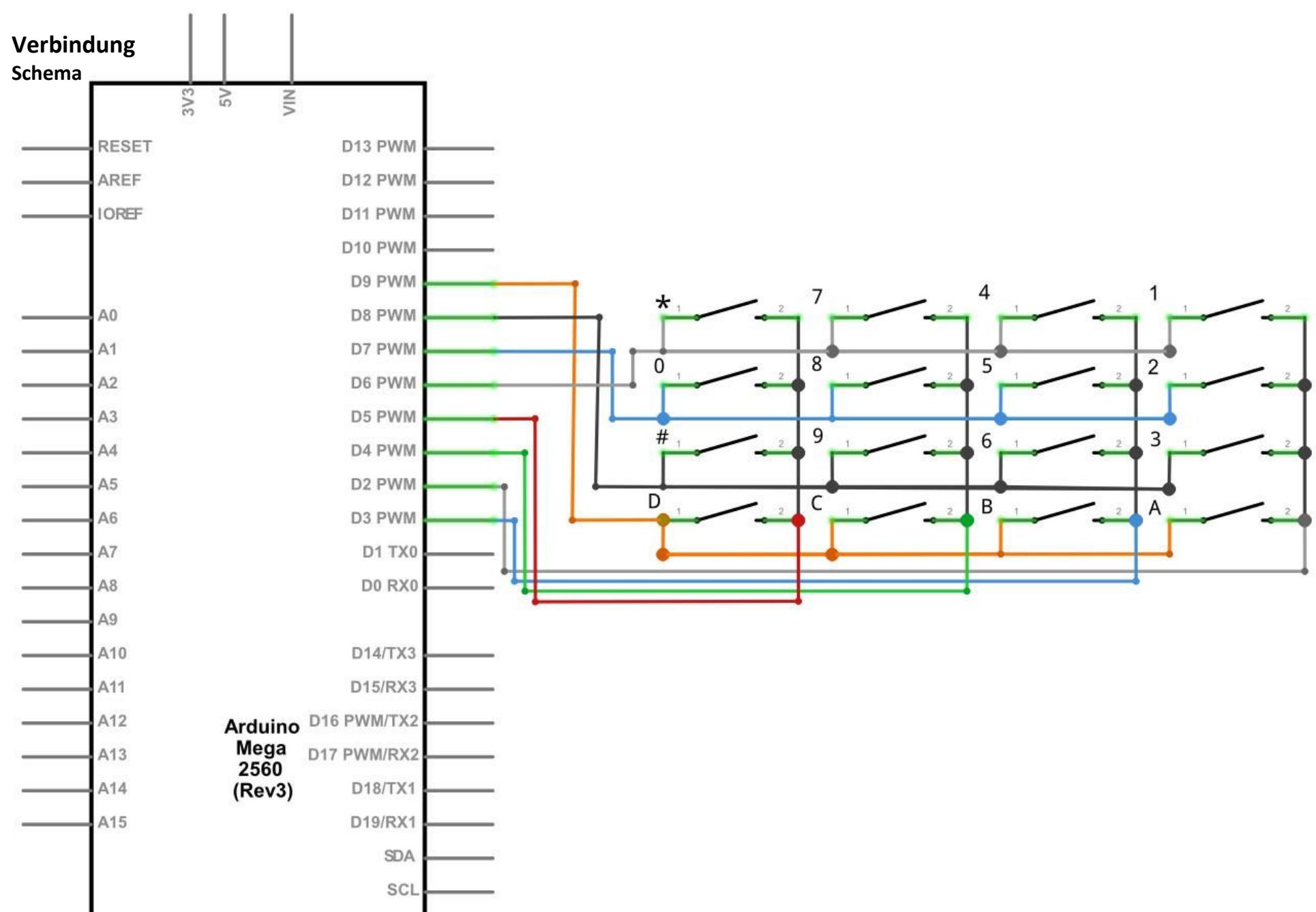
Wenn am Ende alles ordnungsgemäß angeschlossen und programmiert ist und eine Taste gedrückt wird, erscheint es am seriellen Monitor auf Ihrem Computer. Wenn Sie eine Taste drücken, erscheint sie auf dem seriellen Monitor. Aus Gründen der Einfachheit beginnen wir mit der einfachen Betätigung der Taste auf dem Computer. Für dieses Projekt verwenden wir eine Matrix-Tastatur. Dies ist ein Tastenfeld, das einem Codierungsschema folgt. Das erlaubt es, viel weniger Ausgangspins zu haben, als es Schlüssel gibt. Zum Beispiel hat die Matrix-Tastatur, die wir verwenden, 16 Tasten (0-9, A-D, *, #), aber nur 8 Ausgangspins. Mit einer linearen Tastatur müsste es 17 Ausgangspins geben (eine für jede Taste und einen Massepin), um zu funktionieren. Das Matrix-Codierungsschema ermöglicht weniger Ausgabe-Pins und damit viel weniger Verbindungen, die für das Tastenfeld zu arbeiten haben müssen. Auf diese Weise sind sie effizienter als lineare Tastaturen, da sie weniger Verdrahtung haben.

Component Required:

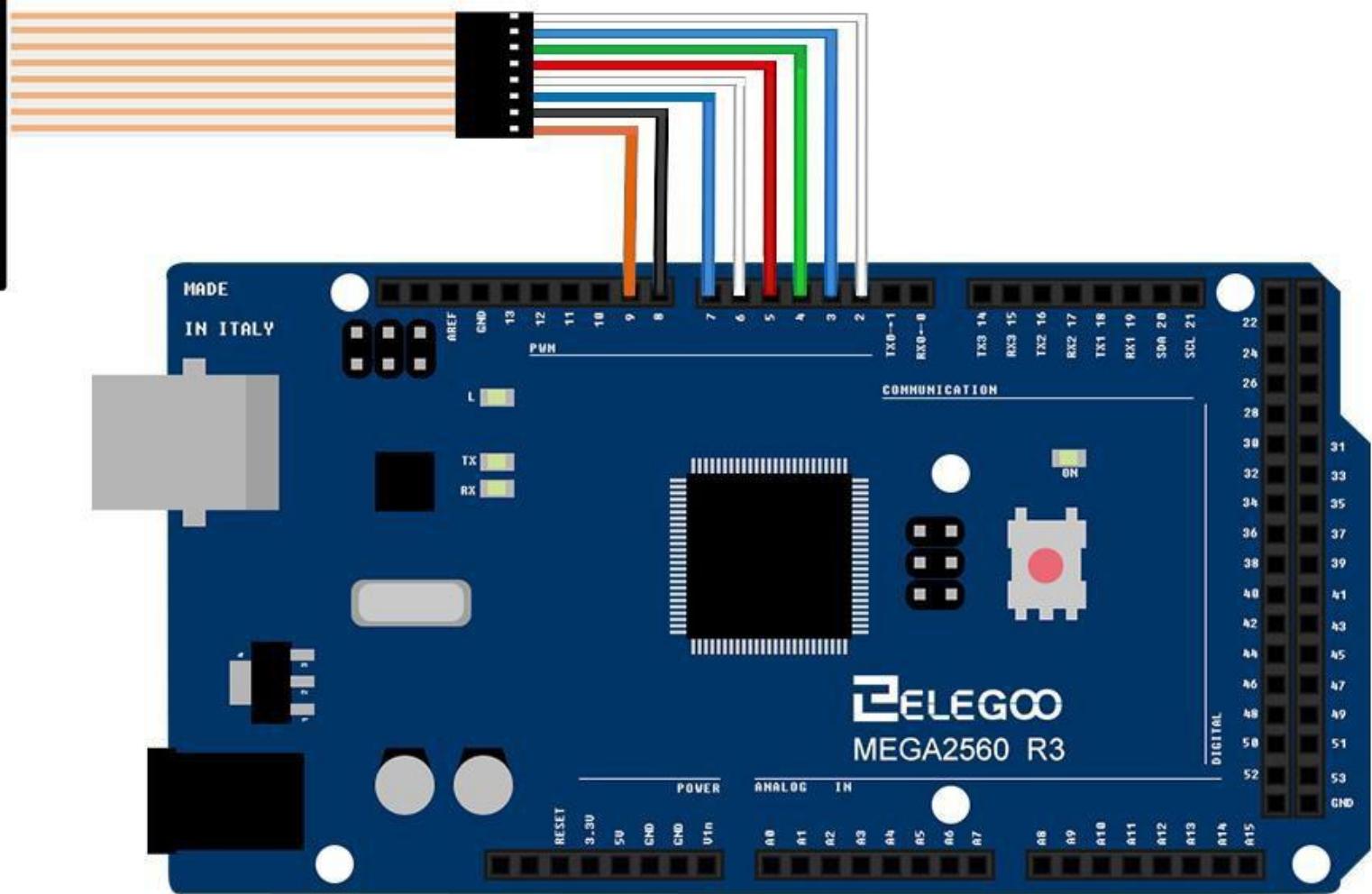
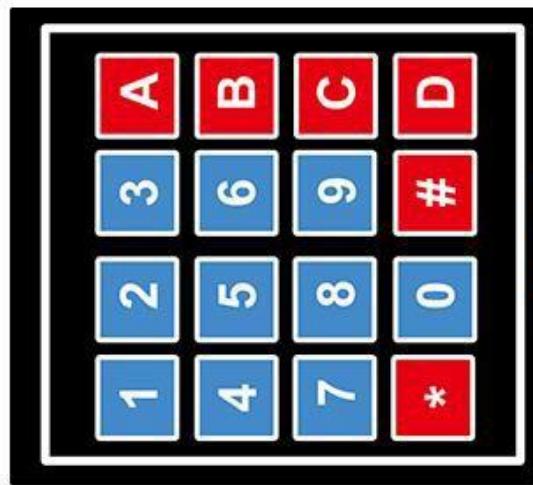
- (1) x Elegoo Mega2560 R3
- (1) x Membran Schaltermodul
- (8) x M-M Kabel (Male to Male Kabel)



Verbindung Schema



Schaltplan



Beim Anschluss der Stifte an die MEGA2560 R3-Platine verbinden wir sie mit den digitalen Ausgangsstiften D9-D2. Wir verbinden den ersten Pin der Tastatur mit D9, den zweiten Pin auf D8, den dritten Pin auf D7, den vierten Pin auf D6, den fünften Pin auf D5, den sechsten Pin auf D4, den siebten Pin auf D3 und den achten Pin zu D2.

Dies sind die Verbindungen in einer Tabelle:

| Keypad Pin | Connects to Arduino Pin... |
|------------|----------------------------|
| 1 | D9 |
| 2 | D8 |
| 3 | D7 |
| 4 | D6 |
| 5 | D5 |
| 6 | D4 |
| 7 | D3 |
| 8 | D2 |

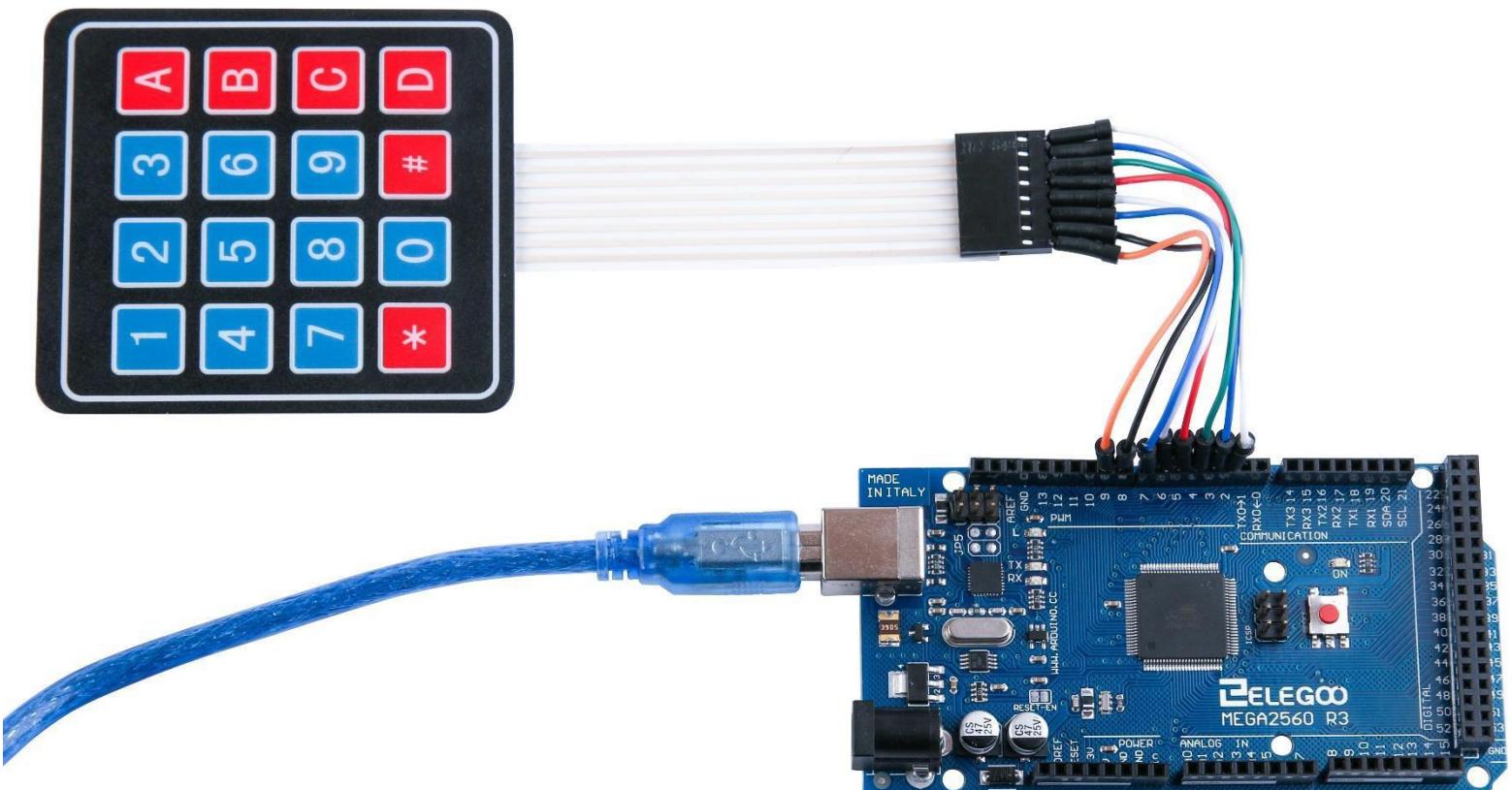
Code

Nach der Verschalten öffnen Sie bitte das Programm im Code-Ordner - Lektion 11 Membranschalter-Modul und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die <Keypad>-Bibliothek installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

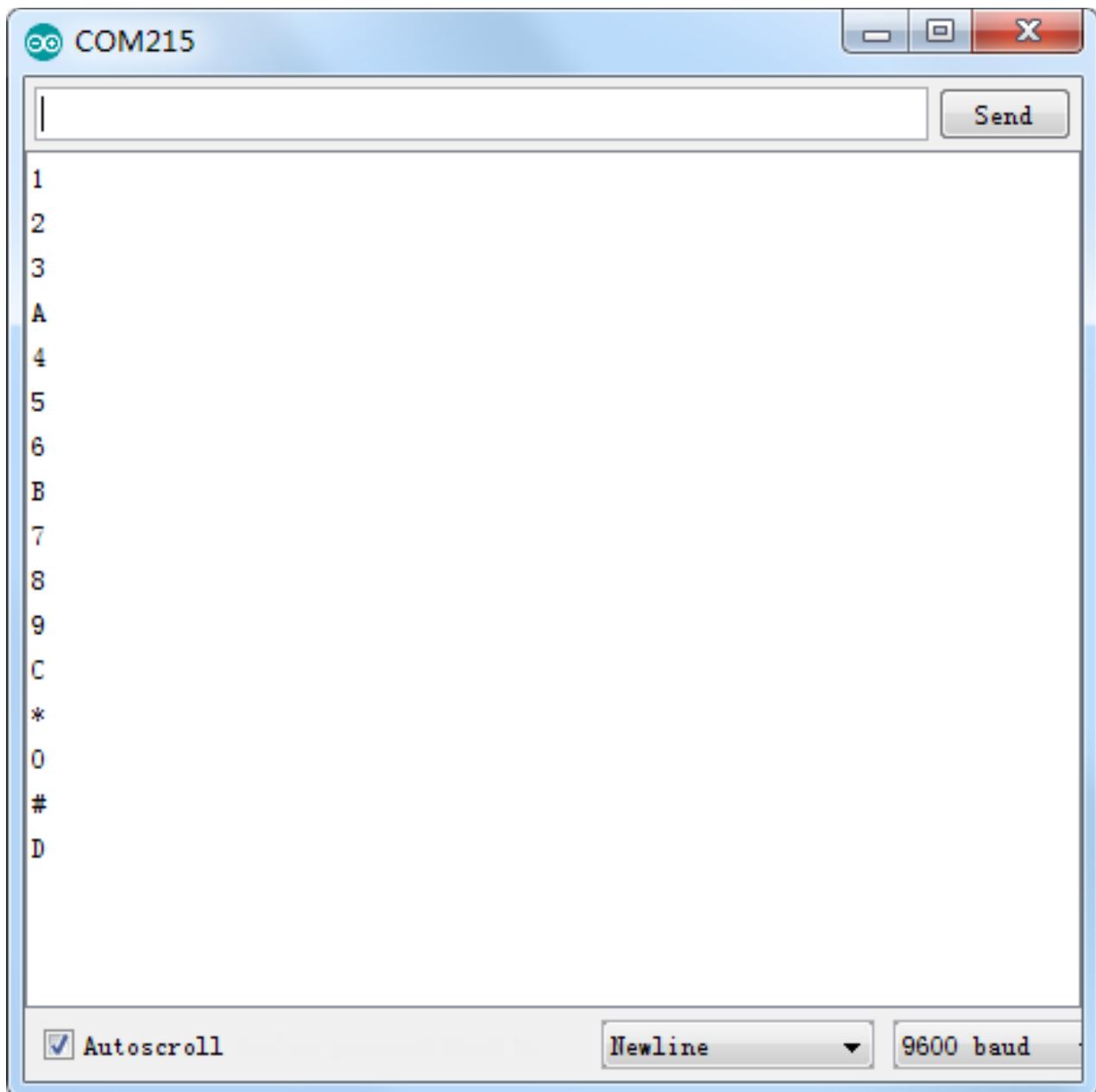
Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion 1.

Beispielfoto



Mit diesem Code bezwecken wir, dass sobald wir eine Taste auf dem Tastenfeld drücken, diese auf dem seriellen Monitor der Arduino-Software erscheinen, sobald der Code kompiliert und auf die MEGA2560 R3-Karte hochgeladen wurde.

Klicken Sie auf die Schaltfläche Serielle Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors werden in Lektion 1 ausführlich vorgestellt.



Lektion 12 DHT11 Temperatur- und Feuchtigkeitssensor

Überblick

In diesem Tutorial lernen wir, wie man einen DHT11, einem Temperatur- und Feuchtigkeitssensor, benutzt.

Es ist genau genug für die meisten Projekte, die einen Überblick über Feuchtigkeits- und Temperatormessungen haben müssen.

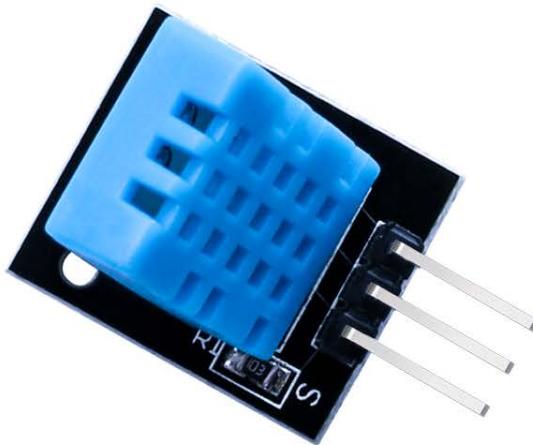
Wieder werden wir eine Bibliothek verwenden, die speziell für diese Sensoren entwickelt wurde, um unseren Code kurz und einfach zuschreiben.

Erforderliche Komponente:

- (1) x Elegoo Mega2560 R3
- (1) x DHT11 Temperatur- und Feuchtigkeitssensor
- (3) x F-M Kabel (Female to Male DuPont Kabel)

Komponenten Einführung

Temperatur- und Feuchtigkeitssensor:



Der DHT11, digitaler Temperatur- und Feuchtigkeitssensor, ist ein zusammengesetzter Sensor der einen kalibrierten digitalen Signalausgang über die Temperatur und der Feuchtigkeit ausgibt. Die dedizierte digitale "Module Collection-Technologie" und die Temperatur- und Feuchtigkeitsmesstechnik

werden angewendet, um sicherzustellen, dass das Produkt eine hohe Zuverlässigkeit und eine hervorragende Langzeitstabilität aufweist. Der Sensor verfügt über einen resistiven Sensor für nasse Bauteile und einen NTC-Temperaturmessgerät und verbindet diese mit einem leistungsstarken 8-Bit-Mikrocontroller.

Anwendungen: HVAC, Entfeuchter, Prüf- und Inspektionsausrüstung, Konsumgüter, Automotive, automatische Steuerung, Datenlogger, Wetterstationen, Haushaltsgeräte, Feuchtigkeitsregler, medizinische und andere Feuchtigkeitsmessung und -kontrolle.

Produktparameter

Relative Feuchtigkeit:

Auflösung: 16Bit

Reproduzierbarkeit: $\pm 1\%$ RH

Genauigkeit: bei 25°C $\pm 5\%$ RH

Austauschbarkeit: Voll austauschbar

Reaktionszeit: $1/e$ (63%) bei 25°C 6s

1m/s Luft 6s

Hysteresis: $<\pm 0.3\%$ RH

Langzeitstabilität: $<\pm 0.5\%$ RH / yr in

Temperatur:

Auflösung: 16Bit

Reproduzierbarkeit: $\pm 0.2^\circ\text{C}$

Genauigkeit: bei 25°C $\pm 2^\circ\text{C}$

Reaktionszeit: $1/e$ (63%) 10S

Elektrische Eigenschaften

Energieversorgung: DC $3.5 \sim 5.5$ V

Versorgungsstrom: Messung 0.3mA standby $60\mu\text{A}$

Messdauer: mehr als 2 Sekunden

Pin Beschreibung

1, Die VDD-Stromversorgung 3.5 ~ 5.5 V DC

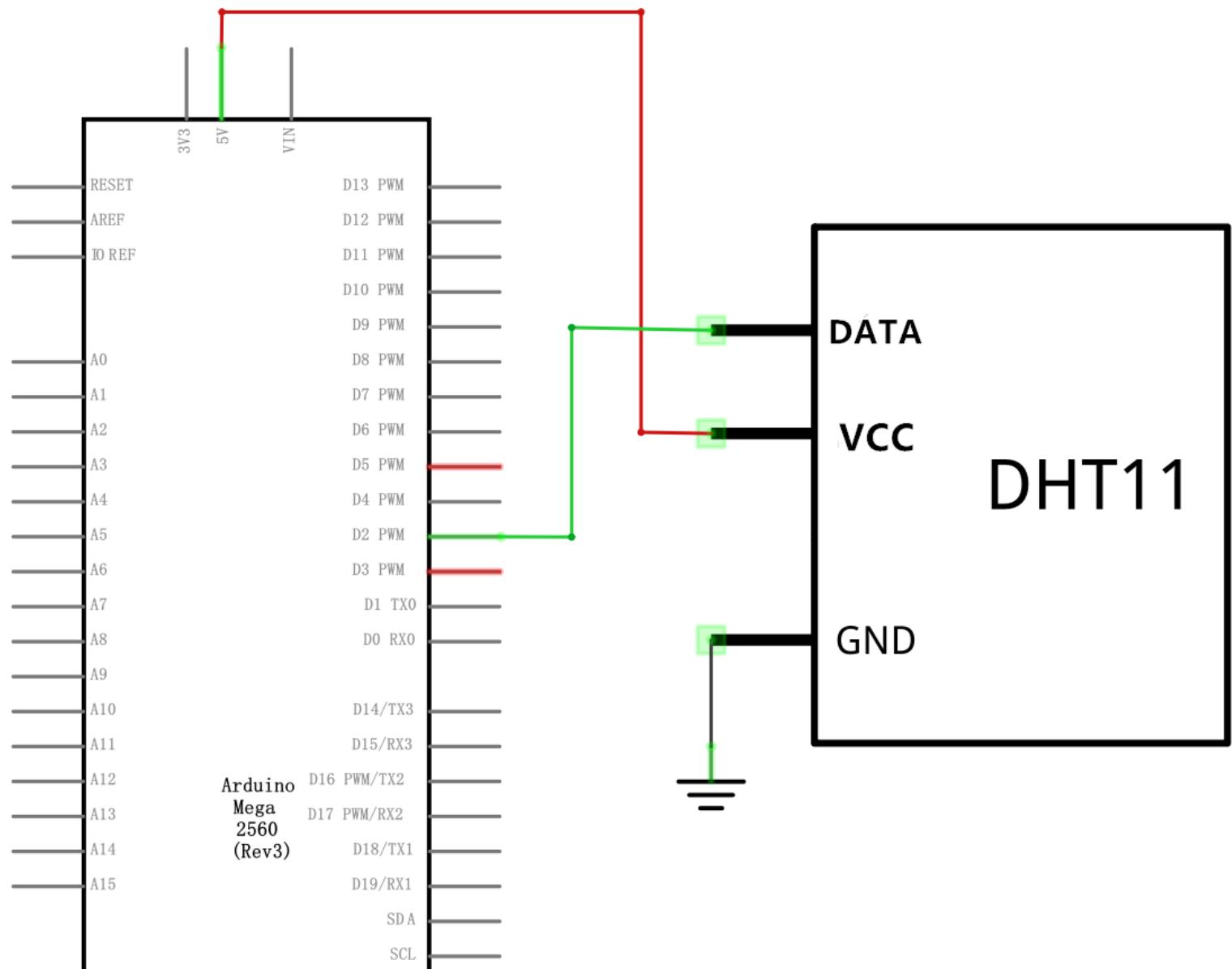
2 DATA serielle daten, ein einzelner bus

3, NC, nicht belegt

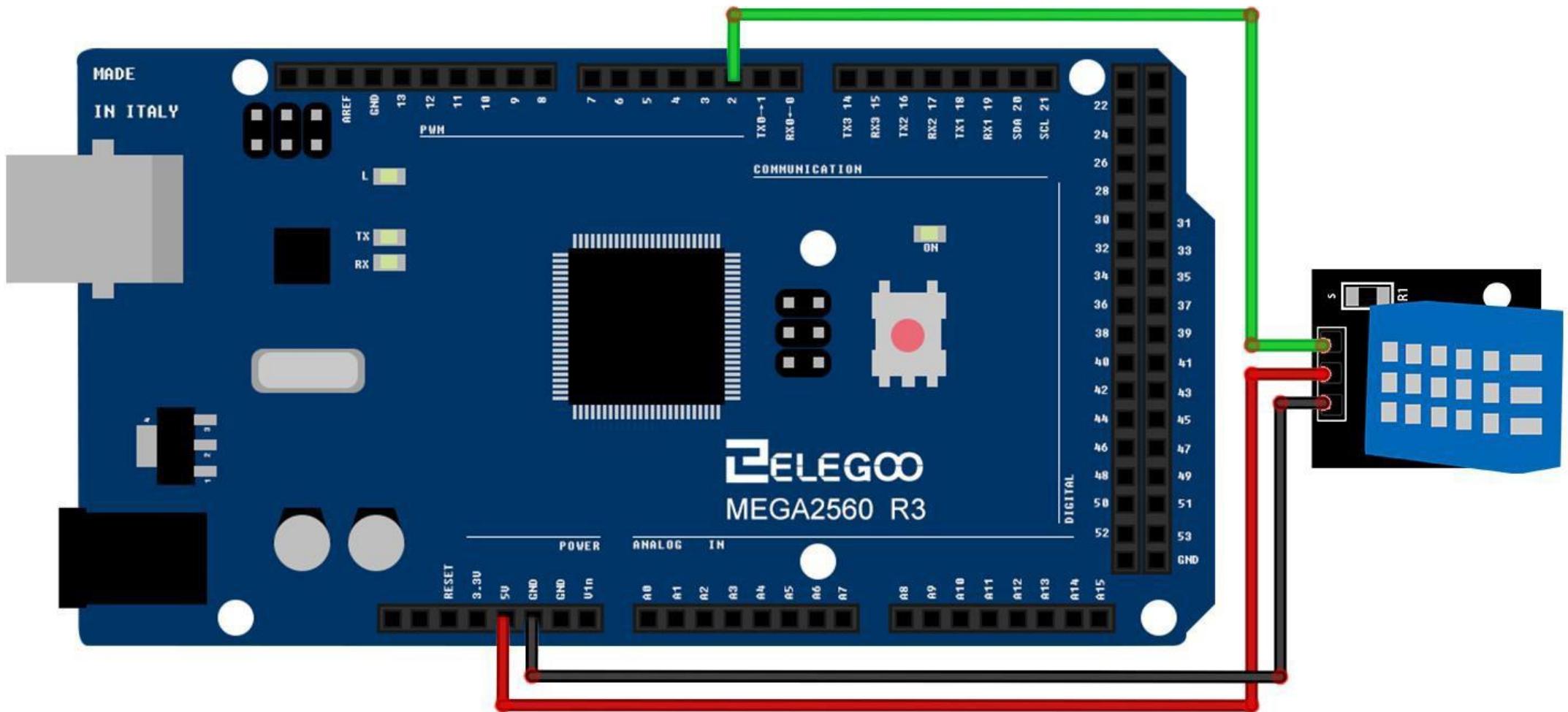
4, GND Masse, der "negative" Anschluss

Verbindung

Schema



Schaltplan



Wie Sie sehen können, brauchen wir nur 3 Verbindungen zum Sensor, da einer der Pins nicht verwendet wird.

Die Anschlüsse sind: Spannung, Masse und Signal, die an jeden Pin an unserem MEGA2560 angeschlossen werden müssen.

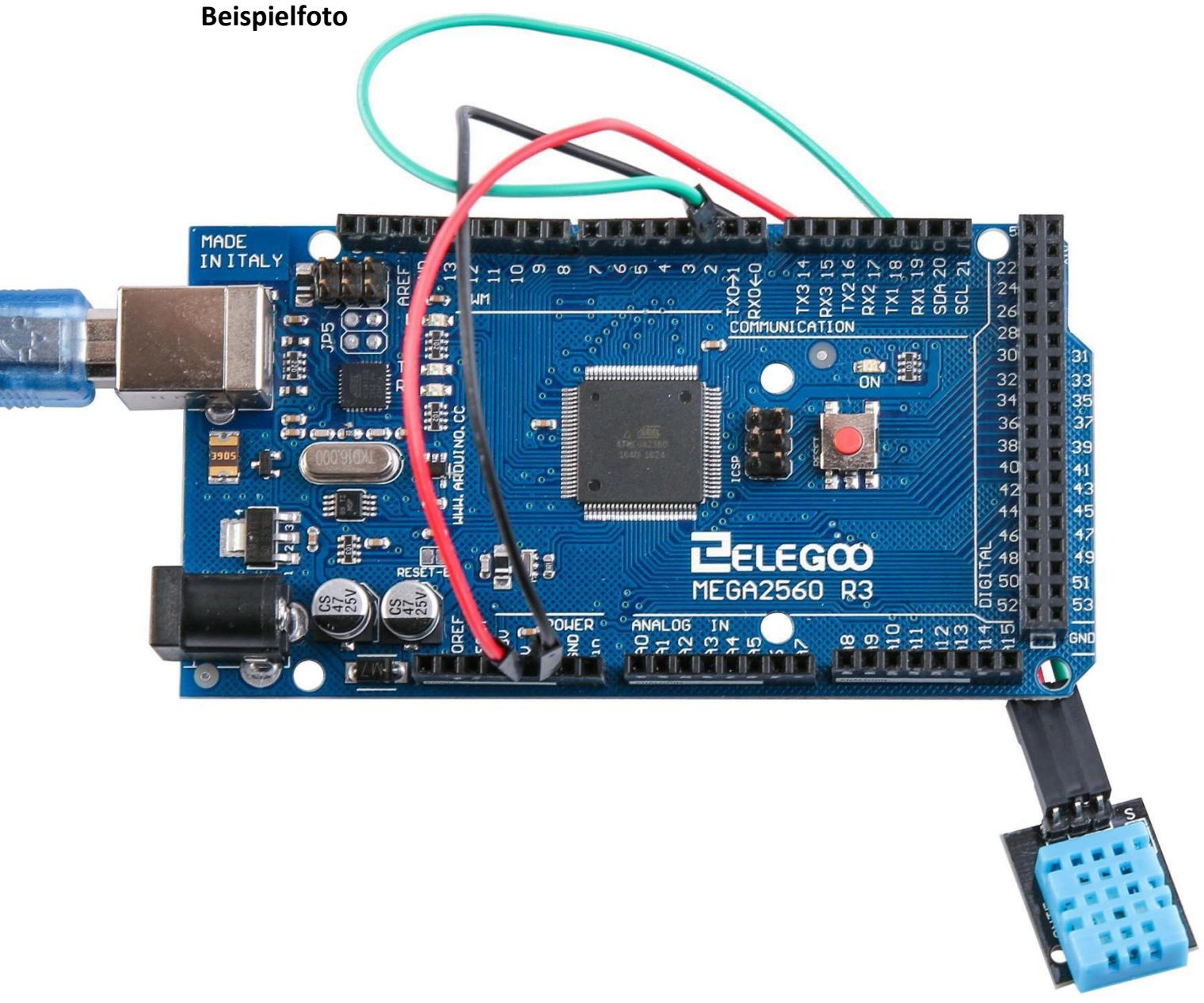
Code

Nach der Verschaltung öffnen Sie bitte das Programm im Code-Ordner - Lektion 12 DHT11 Temperatur- und Feuchtigkeitssensor und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die <SimpleDHT> - Bibliothek installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

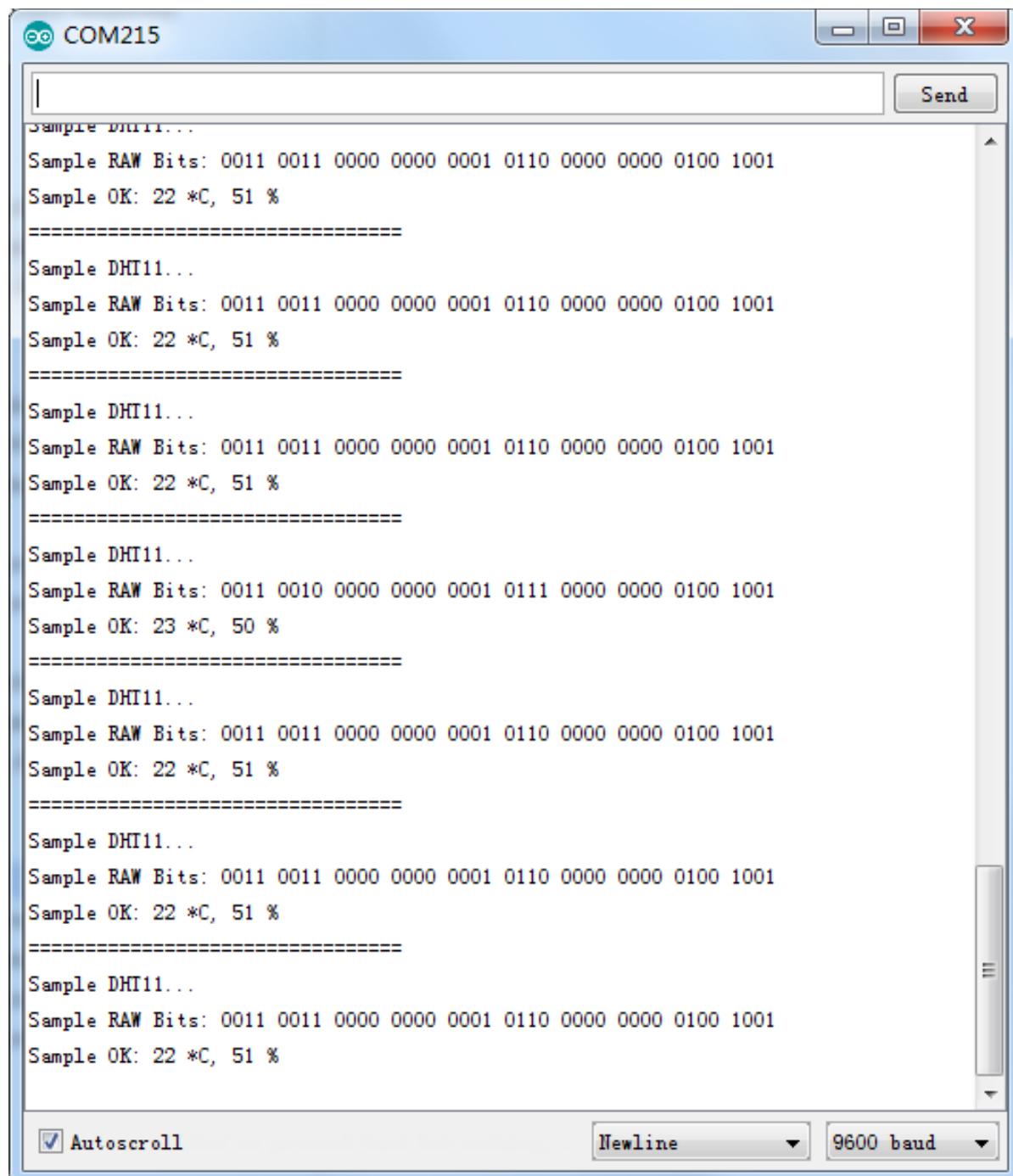
Weitere Informationen zum Tutorial zum Laden der Bibliotheksdatei finden Sie unter Lektion 1.

Beispieldfoto



Laden Sie das Programm, dann öffnen Sie den seriellen Monitor und jetzt können wir die Daten wie unten sehen: (Es zeigt die Temperatur der Umgebung, wir können sehen, es sind 22 Grad, wuhuu kuschelig)

Klicken Sie auf die Schaltfläche Serielle Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors werden in Lektion 1 ausführlich vorgestellt.



The screenshot shows a Windows-style application window titled "COM215". The main window area displays a series of temperature readings from a DHT11 sensor. Each reading consists of a header, raw binary data, and a human-readable format. The headers are "Sample DHT11...", followed by "Sample RAW Bits:" and then "Sample OK: 22 *C, 51 %". The raw binary data is a sequence of 16 bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001. This pattern repeats eight times. At the bottom of the window, there are several control buttons: "Autoscroll" (checked), "Newline" (dropdown menu), and "9600 baud" (dropdown menu).

```
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
Sample DHT11...
Sample RAW Bits: 0011 0011 0000 0000 0001 0110 0000 0000 0100 1001
Sample OK: 22 *C, 51 %
=====
```

Lektion 13 Analoger Joystickmodul

Überblick

Analoge Joysticks sind eine großartige Möglichkeit, um Ihren Projekten ein wenig Kontrolle hinzuzufügen.

In diesem Tutorial lernen wir, wie man das analoge Joystick-Modul benutzt.

Erforderliche Komponente:

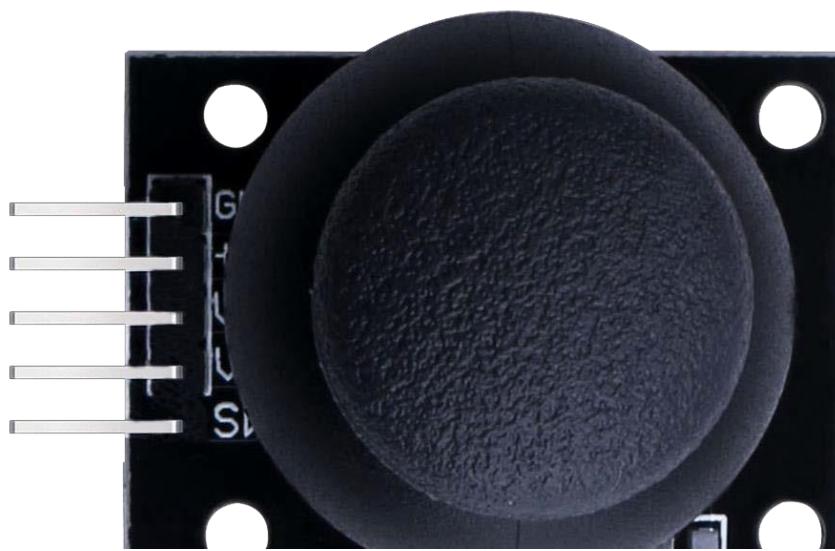
- (1) x Elegoo Mega2560 R3
- (1) x Joystickmodul
- (5) x F-M Kabel (Female to Male DuPont Kabel)

Komponenten Einführung

Joystick

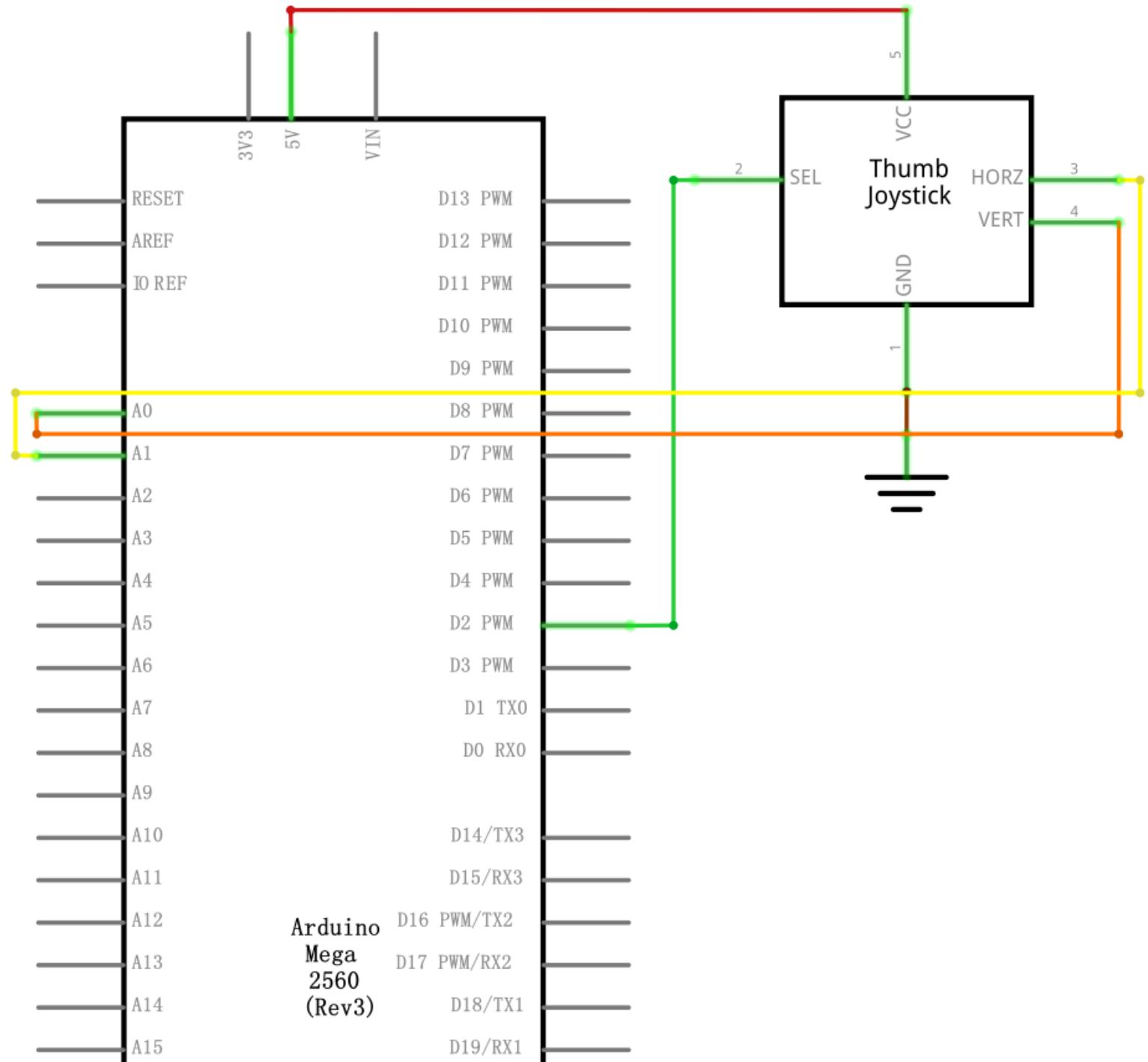
Das Modul hat 5 Pins: VCC, Ground, X, Y, Key. Beachten Sie, dass die Etiketten auf Ihrem kann etwas anders sein, je nachdem, wo Sie das Modul gekauft haben. Der Joystick ist analog und sollte genauere Messwerte liefern als einfache "direkte" Joysticks welche andere Arten von Tasten oder mechanische Schalter nutzen. Zusätzlich kannst du den Joystick nach unten drücken (wie bei den herkömmlichen Spielekonsolen PS2/3/4, Xbox und so weiter), um einen 'press to select' Druckknopf zu benutzen.

Wir müssen analoge Arduino-Pins verwenden, um die Daten aus den X / Y-Pins zu lesen, und einen digitalen Pin, um den Button zu lesen. Der Keypin ist mit Masse verbunden wenn der Joystick heruntergedrückt wird. Um stabile Messwerte aus dem Key / Select Pin zu erhalten, muss er über einen Pull-up-Widerstand an VCC angeschlossen werden. Die eingebauten Widerstände auf den Arduino Digital Pins können verwendet werden. Ein Tutorial für die Aktivierung der Pull-up-Widerstände für Arduino-Pins ist, dass sie als Eingänge konfiguriert werden.

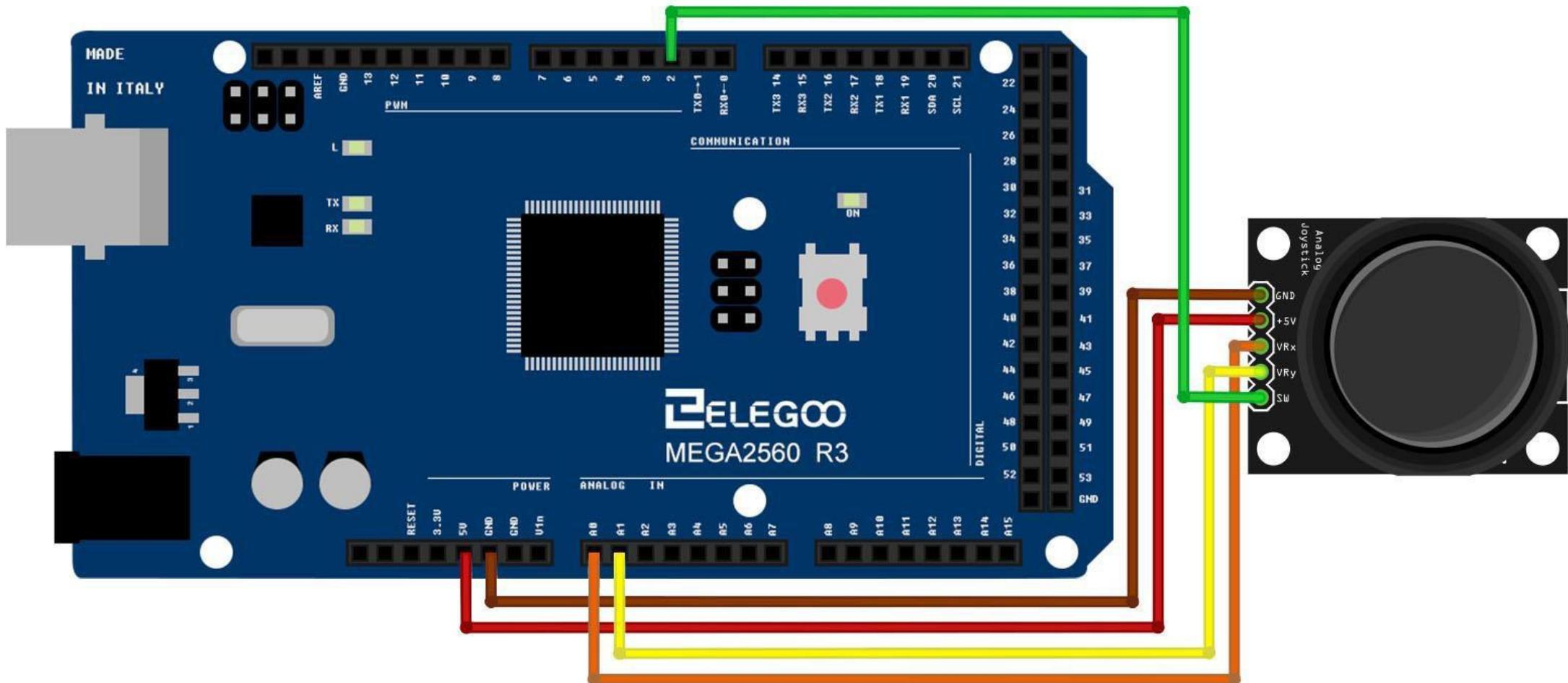


Verbindung

Schema



Schaltplan



Wir benötigen 5 Verbindungen zum Joystick.

Die Anschlüsse sind: Key, Y, X, Spannung und Masse.

"Y und X" sind analog und "Key" ist Digital. Wenn Sie den Schalter nicht benötigen, brauchen Sie nur 4 der insgesamt 5 Pins.

Code

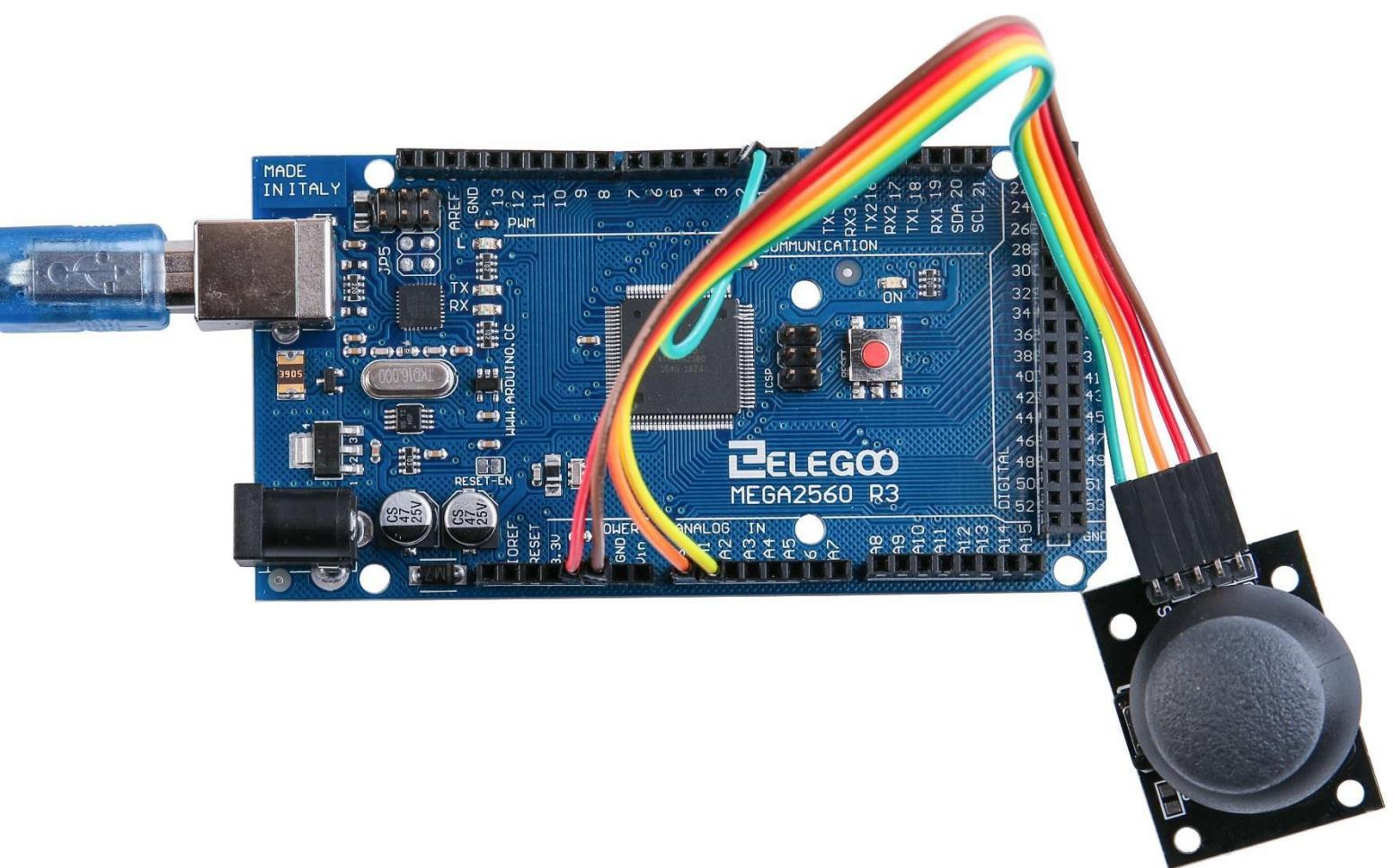
Nach der Verschaltung öffnen Sie bitte das Programm im Codeordner - Lektion 13 Analog Joystick Module und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Analoge Joysticks sind grundsätzlich aufgebaut wie Potentiometer, so dass sie analoge Werte zurückgeben.

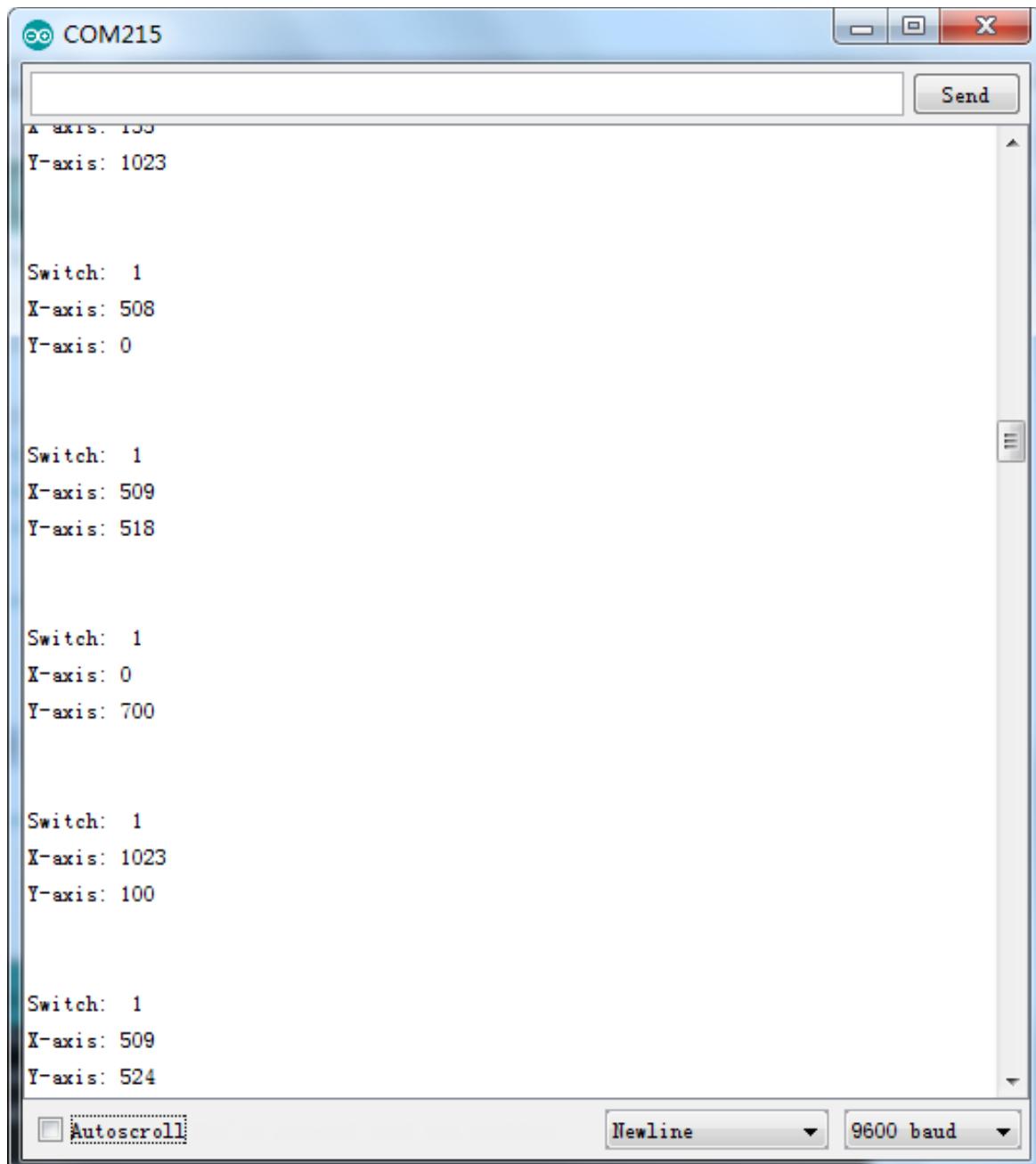
Wenn der Joystick in der Ruheposition oder Mitte ist, sollte er einen Wert von etwa 512 zurückgeben.

Der Wertebereich reicht von 0 bis 1024.

Beispielfoto



Öffnen Sie den seriellen Monitor, dann können Sie die unten gezeigten Daten sehen:
Klicken Sie auf die Schaltfläche Serielle Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors werden in Lektion 1 ausführlich vorgestellt.



The screenshot shows a Windows-style application window titled "COM215". The window contains a text area displaying several lines of sensor data. The data consists of three parameters: "Switch", "X-axis", and "Y-axis", each followed by a numerical value. The values fluctuate across the different lines. At the bottom of the window, there are three control buttons: "Autoscroll" (unchecked), "Newline" (set to "▼"), and "9600 baud" (set to "▼").

```
Axes. 100
Y-axis: 1023

Switch: 1
X-axis: 508
Y-axis: 0

Switch: 1
X-axis: 509
Y-axis: 518

Switch: 1
X-axis: 0
Y-axis: 700

Switch: 1
X-axis: 1023
Y-axis: 100

Switch: 1
X-axis: 509
Y-axis: 524
```

Lektion 14 IR Empfangsmodul

Überblick

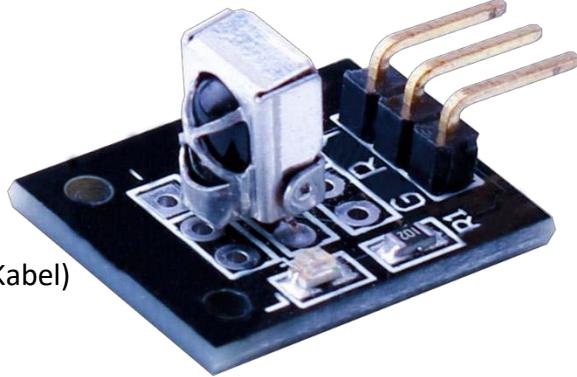
Mit einer IR-Fernbedienung ist eine gute Möglichkeit, drahtlose Kontrolle über Ihr Projekt zu haben.

Infrarot-Fernbedienungen sind einfach zu programmieren und einfach zu bedienen. In diesem Tutorial werden wir den IR-Empfänger an den MEGA2560 anschließen und dann eine Bibliothek verwenden, die für diesen speziellen Sensor entworfen wurde.

In unserem Sketch haben wir alle IR Hexadezimal-Codes, die auf dieser Fernbedienung verfügbar sind, wir werden auch erkennen, ob der Code erkannt wurde.

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x IR Empfängermodul
- (1) x IR Fernbedienung
- (3) x F-M Kabel (Female to Male DuPont Kabel)



Komponenten Einführung

IR EMPFÄNGERSENSOR:

IR-Detektoren sind kleine Mikrochips mit einer Fotozelle, die auf das Infrarotlicht abgestimmt sind. Sie werden fast immer für die Fernbedienung verwendet - jeder TV- und DVD-Player hat eine davon in der Front, um das IR-Signal von der Fernbedienung zu empfangen. Im Inneren der Fernbedienung befindet sich eine passende IR-LED, die IR-Impulse aussendet, um dem Fernsehgerät zu signalisieren auszuschalten oder Kanäle zu wechseln. IR-Licht ist für das menschliche Auge nicht sichtbar, was bedeutet, dass es ein wenig mehr Arbeit braucht, um ein Setup zu testen.

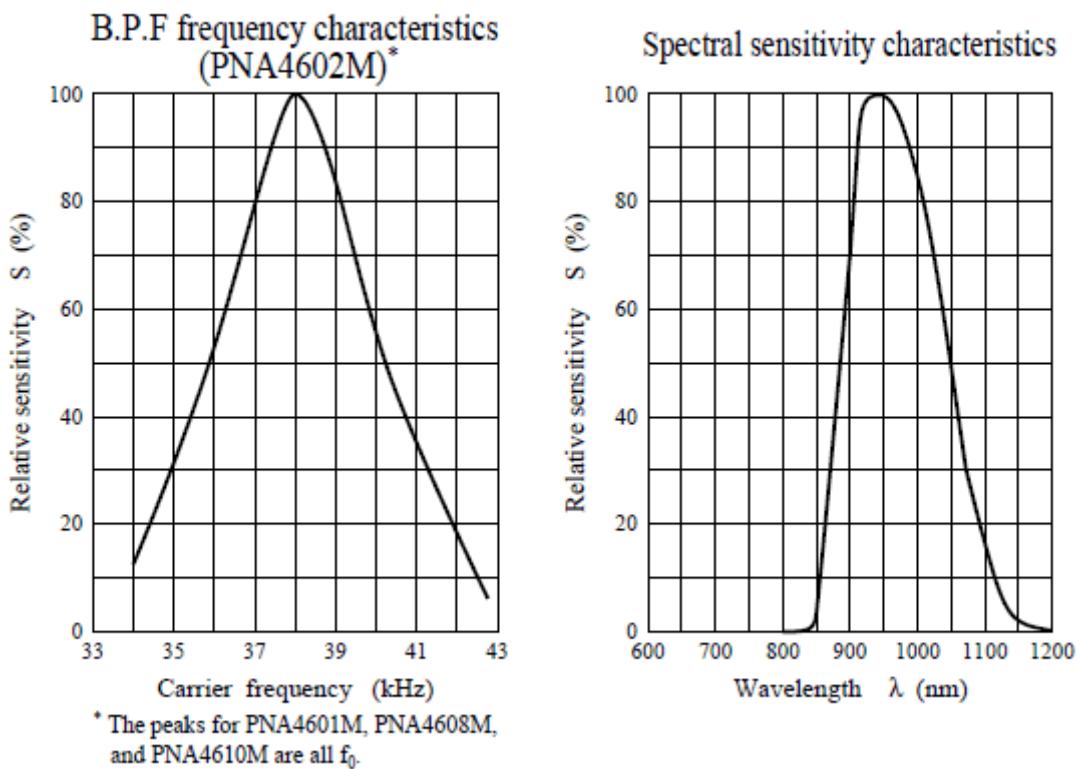
Es gibt ein paar Unterschiede zwischen diesen und zum Beispiel LDR Photowiderständen:

IR-Detektoren sind speziell für IR-Licht gefiltert, sie sind nicht gut bei der Erkennung von sichtbarem Licht. Auf der anderen Seite sind die LDR gut bei der Erkennung von gelb / grün sichtbarem Licht und sind bei IR-Licht nichtgut.

IR-Detektoren haben einen Demodulator im Inneren, der nach moduliertem infrarotem Licht bei 38 kHz sucht. Ein dauerleuchtet einer IR-LED kann nicht erkannt werden, es muss PWM blinken bei 38KHz sein. LDR haben keine Art Demodulator und können jede Frequenz (einschließlich DC) innerhalb der Ansprechgeschwindigkeit der Lichtschranke (die etwa 1 kHz beträgt) erkennen,

IR-Detektoren sind digital aus - entweder ermitteln sie 38KHz IR-Signal und geben niedrig (0V) aus oder sie ermitteln keine und geben Sie hoch (5V). Lichtschranken wirken wie Widerstände, der Widerstand ändert sich je nachdem, wie viel Licht sie ausgesetzt sind.

Was Sie messen können:

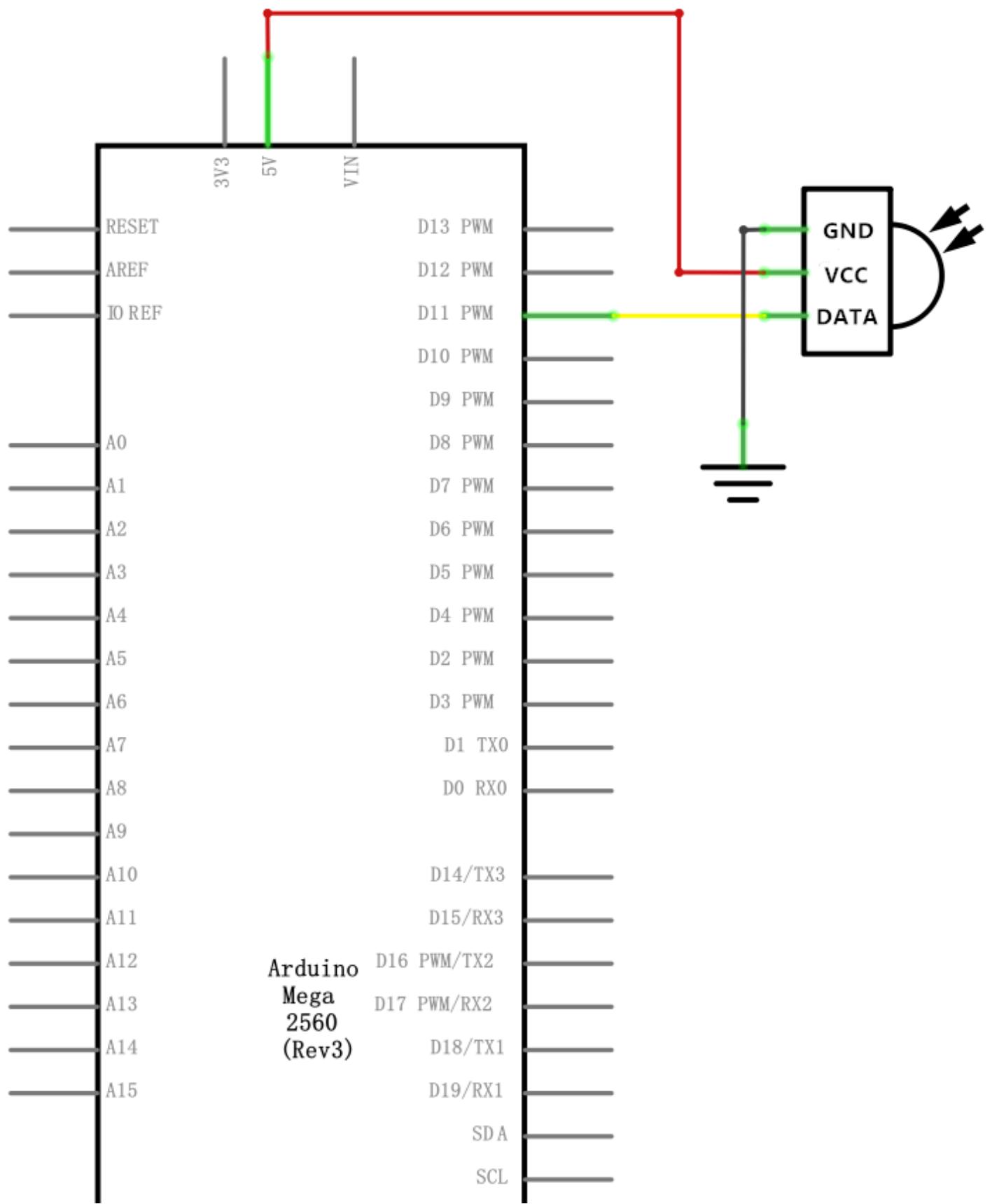


Wie Sie aus diesem Datenblatt-Graphen entnehmen können, liegt die Peak-Frequenz-Detektion bei 38 kHz und die Peak-LED-Farbe beträgt 940 nm. Sie können den Sensor bei einer Frequenz von ca. 35 kHz bis 41 kHz verwenden, die Empfindlichkeit nimmt allerdings ab, so dass es sein kann, dass Ihre Eingaben über längere Distanzen gar nicht erkannt werden. Ebenso können Sie LEDs mit einer emittierten Wellenlänge von 850 bis 1100 nm verwenden, aber sie funktionieren nicht so gut wie 900 bis 1000nm stellen Sie also sicher, dass Sie passende LEDs zu benutzen! Überprüfen Sie das Datenblatt für Ihre IR-LED, um die Wellenlänge zu überprüfen.

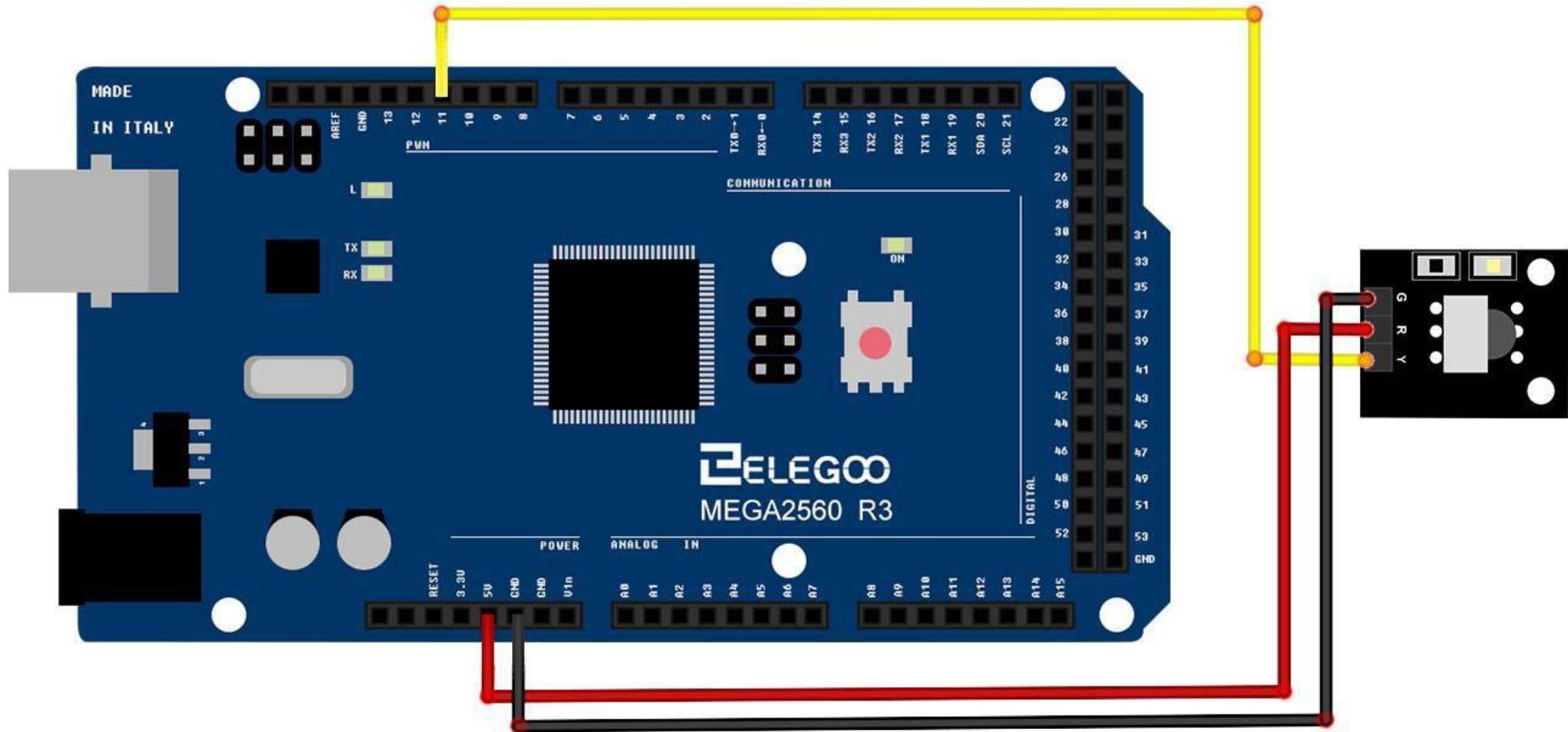
Versuchen Sie, eine LED mit einer 940nm Wellenlänge zu bekommen - denken Sie daran, dass 940nm nicht sichtbares Licht ist!

Verbindung

Schema



Schaltplan



Es gibt 3 Verbindungen zum IR-Empfänger.

Die Anschlüsse sind: Signal, Spannung und Masse.

Das "-" ist das Ground, "S" ist Signal und der mittlere Pin ist Spannung 5V.

Code

Nach der Verschaltung öffnen Sie bitte das Programm im Code-Ordner - Lektion 14 IR-Empfänger-Modul und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

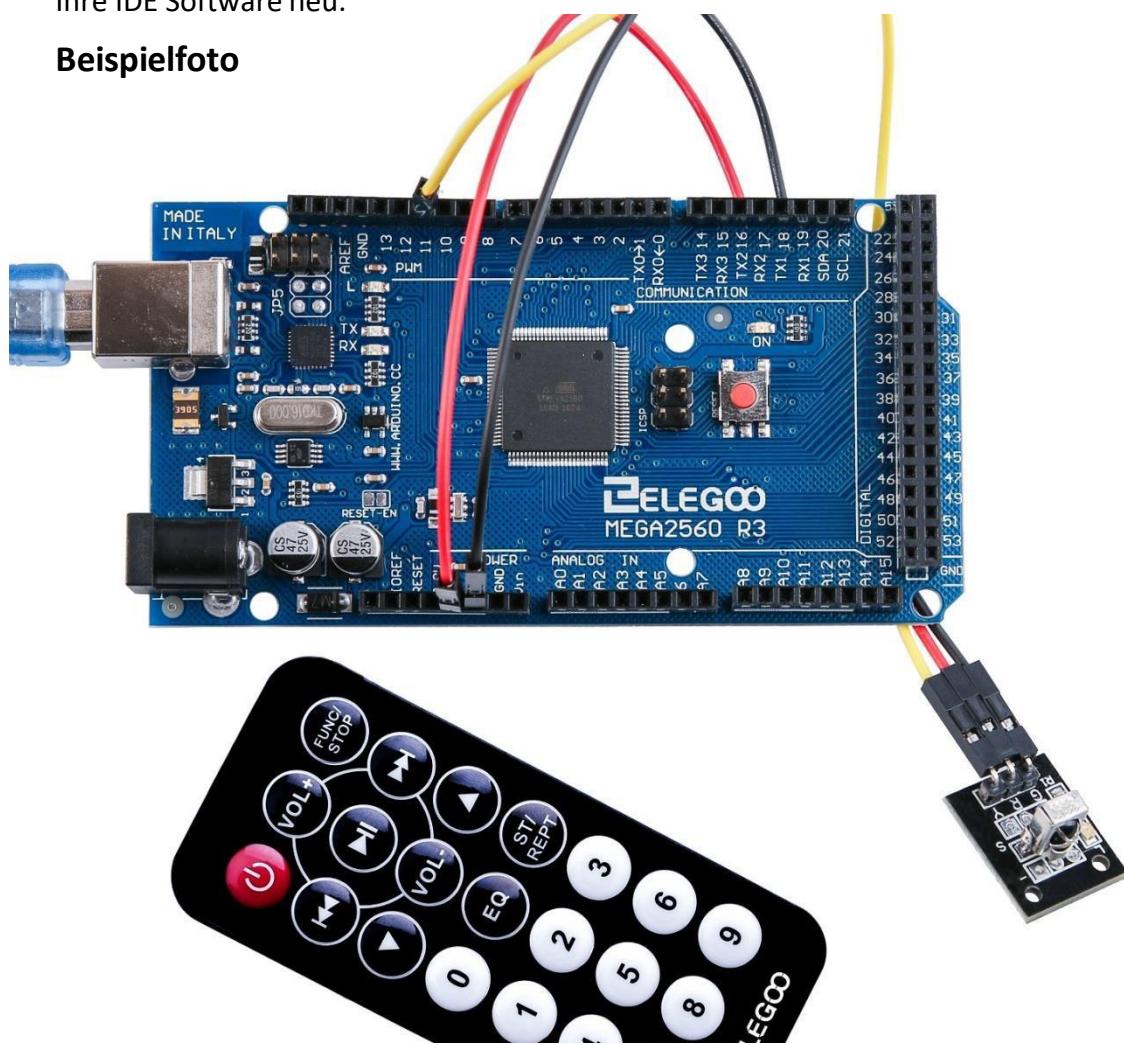
Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die `<IRremote>` - Bibliothek installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion 1.

Als nächstes werden wir die `<RobotIRremote>` aus dem Bibliotheksordner verschieben, wir tun dies, weil diese Bibliothek die selbe ist, mit der wir arbeiten. Sie können es einfach zurück in den Bibliotheksordner ziehen, sobald Sie fertig sind Ihren Mikrocontroller zu programmieren.

Sobald Sie die Bibliothek installiert haben, gehen Sie einfach weiter und starten Sie Ihre IDE Software neu.

Beispelfoto



Öffnen Sie den seriellen Monitor, dann können Sie die unten angezeigten Daten sehen:
Klicken Sie auf die Schaltfläche Serielle Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors werden in Lektion 1 ausführlich vorgestellt.

```
IR Receiver Button Decode
POWER
VOL+
FUNC/STOP
FAST BACK
PAUSE
FAST FORWARD
DOWN
VOL-
UP
0
EQ
ST/REPT
1
2
3
4
5
6
7
8
9
```

Autoscroll Newline 9600 baud

Lektion 15 MAX7219 LED Dot Matrix Modul

Überblick

In diesem Tutorial werden wir einen MAX7219 anschließen und den Text überrollen. Da diese Module den MAX7219 LED-Treiberchip verwenden, können wir die 64 LEDs jedes Moduls ein- und ausschalten, wobei nur 3 Pins auf unserem MEGA2560 verwendet werden.

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x Max7219 Modul
- (5) x F-M Kabel (Female to Male DuPont Kabel)

Komponenten Einführung

MAX7219 LED Dot Matrix Modul

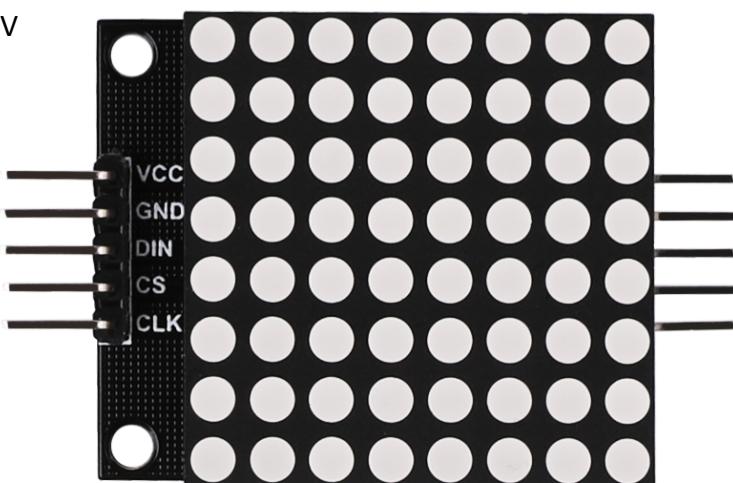
Unser Projekt ist im grunde genommen ein Arduino der über eine serielle Schnittstelle einen MAX7219 IC betreibt, welcher wiederum eine 8x8 LED Matrix betreibt. Der MAX7219 IC ist ein serieller Eingang/ Ausgang Common-Cathode-Display-Treiber, der Mikroprozessor ist die Schnittstelle zu einer 7-Segment numerischen LED-Anzeigen von bis zu 8 Ziffern, Grafischen Anzeigen oder 64 einzelnen LEDs. Aus Gründen der Bequemlichkeit wird hier eine 8×8 LED-Matrix verwendet, die mit einem MAX7219 IC-Setup integriert ist und als vorverdrahtetes Modul erhältlich ist. Typische Spezifikation dieses LED Matrix Moduls ist nachfolgend dargestellt:

Betriebsspannung: DC 4.7V - 5.3V

Typische Spannung: 5V

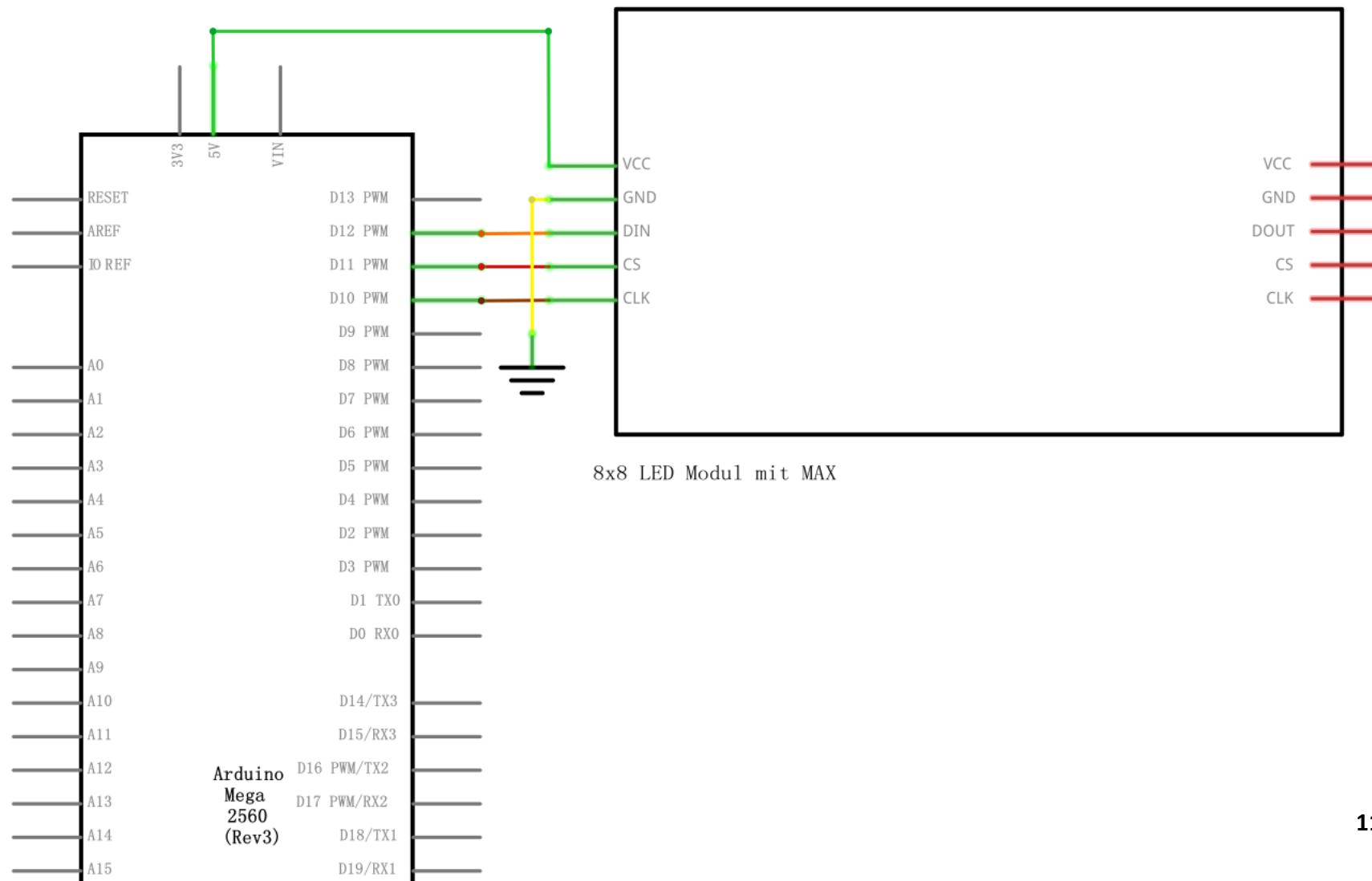
Betriebsstrom: 320mA

Max Betriebsstrom: 2A

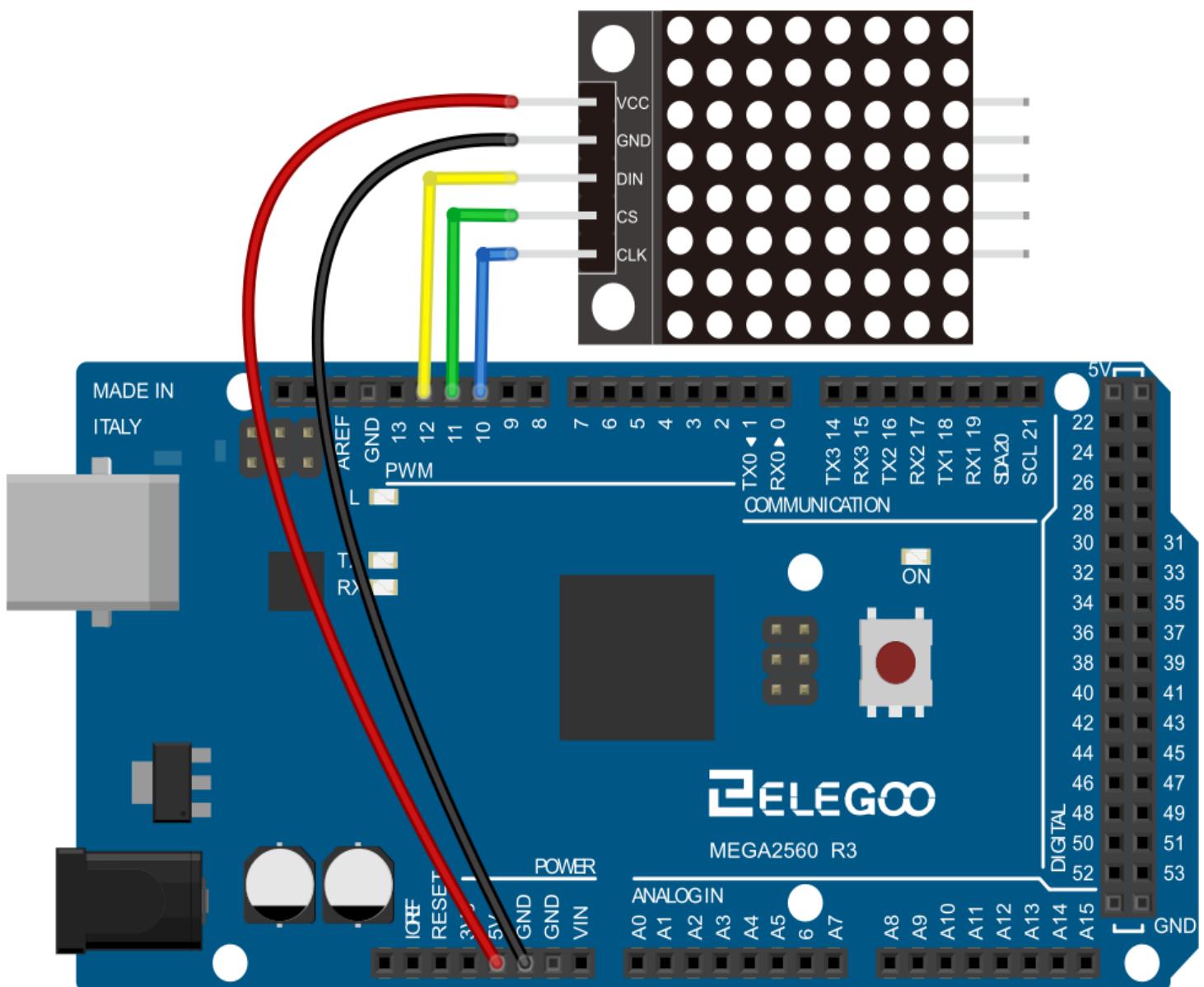


Verbindung

Schema



Schaltplan



VCC und Ground sind mit dem Arduino verbunden.

Pin 12 ist mit DIN verbunden, Pin 11 ist an CS angeschlossen und Pin 10 ist an CLK angeschlossen.

Code

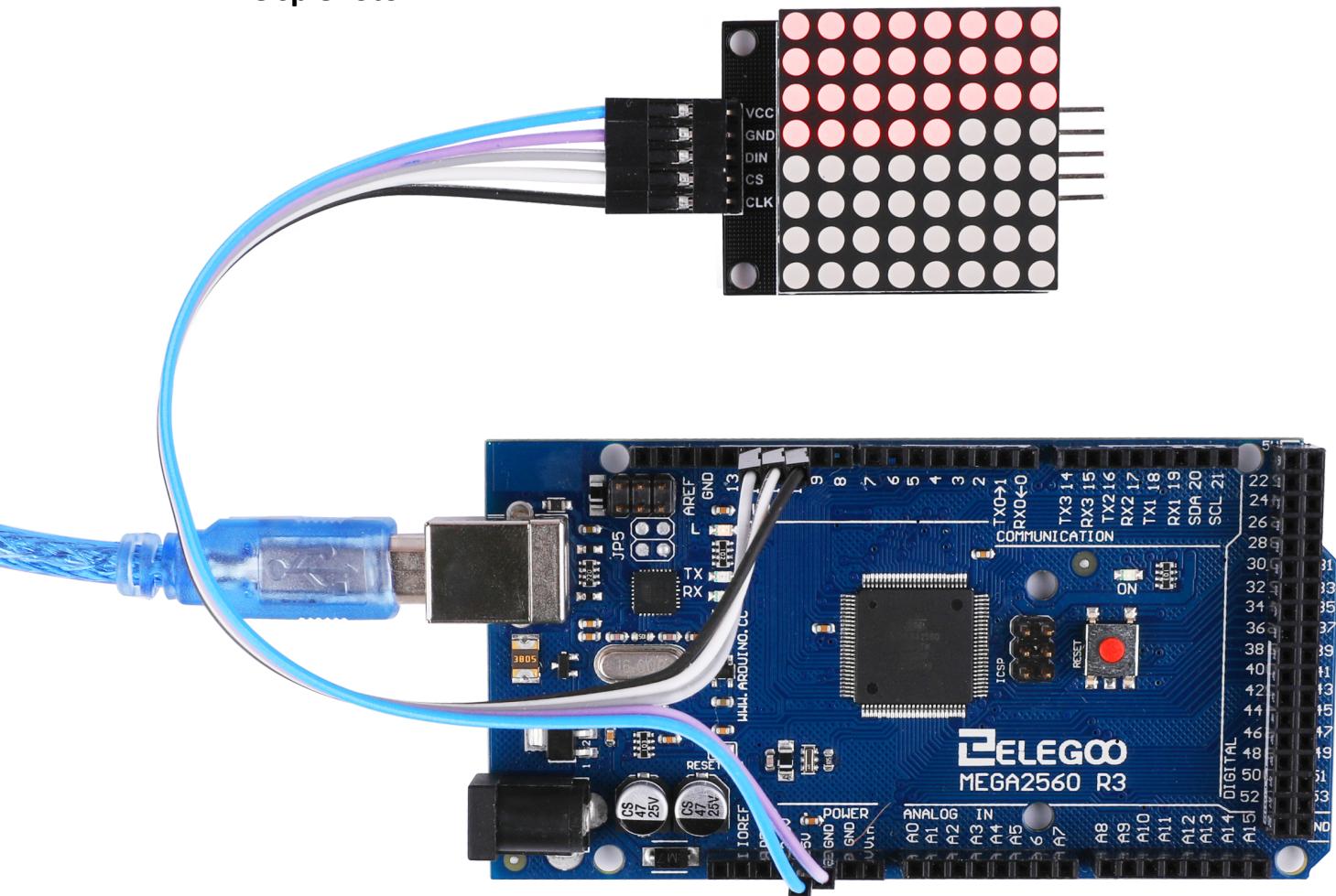
Unser Sketch wird von der "Maxmatrix" -Bibliothek Gebrauch machen, um mit dem MAX7219 Modul zu kommunizieren.

Nach der Verdrahtung öffnen Sie bitte das Programm im Codeordner - Lektion 15 MAX7219 LED Dot Matrix Modul und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die Bibliothek <LedControl> installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion 1

Beispelfoto



Lektion 16 (GY-521/QMI8568) Modul

Überblick

In dieser Lektion lernen wir, wie man das GY-521-Modul einsetzt, das zu den besten IMU-Sensoren (Inertia Measurement Unit) gehört, die mit dem Arduino kompatibel sind. IMU-Sensoren wie die GY-521 werden in selbstbalancierenden Robotern, UAVs, Smartphones usw. eingesetzt.

Erforderliche Komponente:

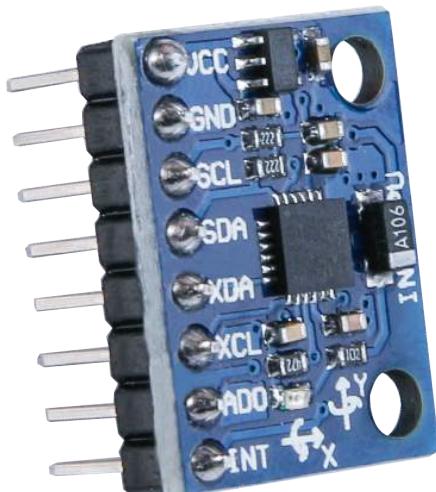
- (1) x Elegoo Mega2560 R3
- (1) x GY-521 Modul
- (4) x F-M Kabel

Komponenten Einführung

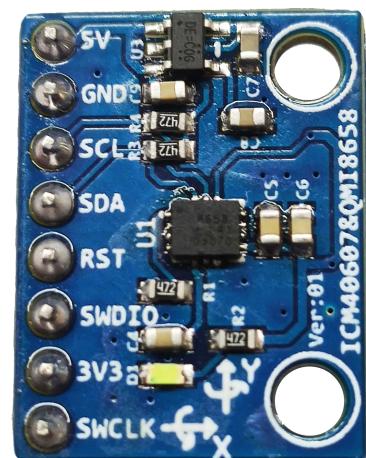
GY-521 SENSOR

Der InvenSense GY-521 Sensor enthält einen MEMS-Beschleunigungsmesser und einen MEMS-Gyro in einem einzigen Chip. Er ist sehr genau, da er 16-Bit-Analog-Digital-Konvertierungs-Hardware für jeden Kanal enthält. Deshalb erfasst er gleichzeitig den x-, y- und z-Kanal. Der Sensor nutzt den I2C-Bus zur Anbindung an den Arduino.

Die GY-521 Sensoren sind nicht teuer, vor allem angesichts der Tatsache, dass sie sowohl einen Beschleunigungsmesser als auch einen Gyroskop kombiniert.



GY-521



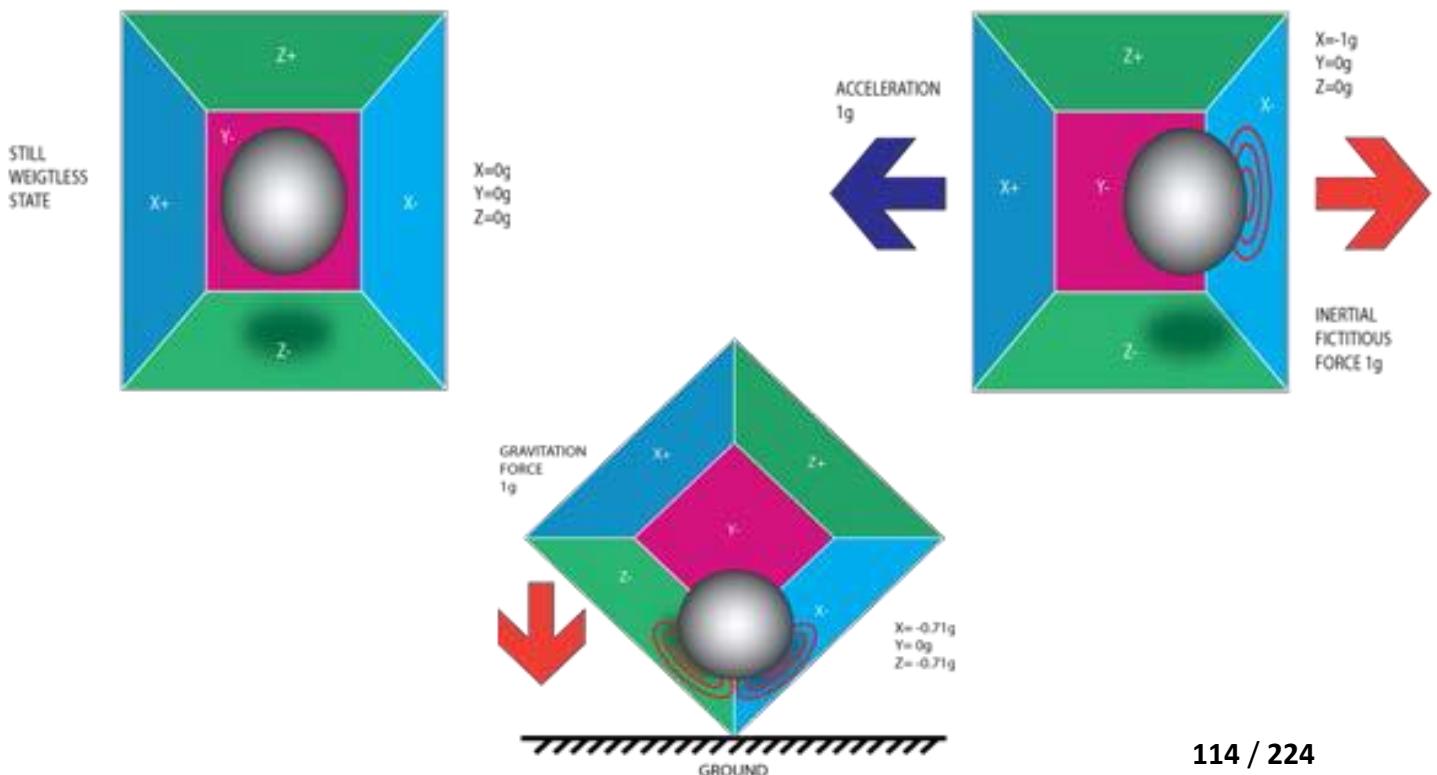
QMI-8568

IMU-Sensoren sind eine der am meisten verbauten Arten von Sensoren, die heute in allen Arten von elektronischen Gadgets verwendet werden. Sie sind in Smartphones, Wearables, Game-Controller, etc. zu finden. IMU-Sensoren helfen uns dabei, die Haltung eines Objekts zu messen, das an den Sensor im dreidimensionalen Raum verbunden ist. Diese gemessenen Werte sind in der Regel Winkel und helfen uns so, seine Haltung zu bestimmen. So werden sie in Smartphones eingesetzt um ihre Orientierung zu erkennen. Und auch in tragbaren Gadgets wie das Nike-Fuelband (Fitnessarmband) oder fitbit Aktivitätentracker, haben IMU-Sensoren verbaut um die Bewegung zuverfolgen.

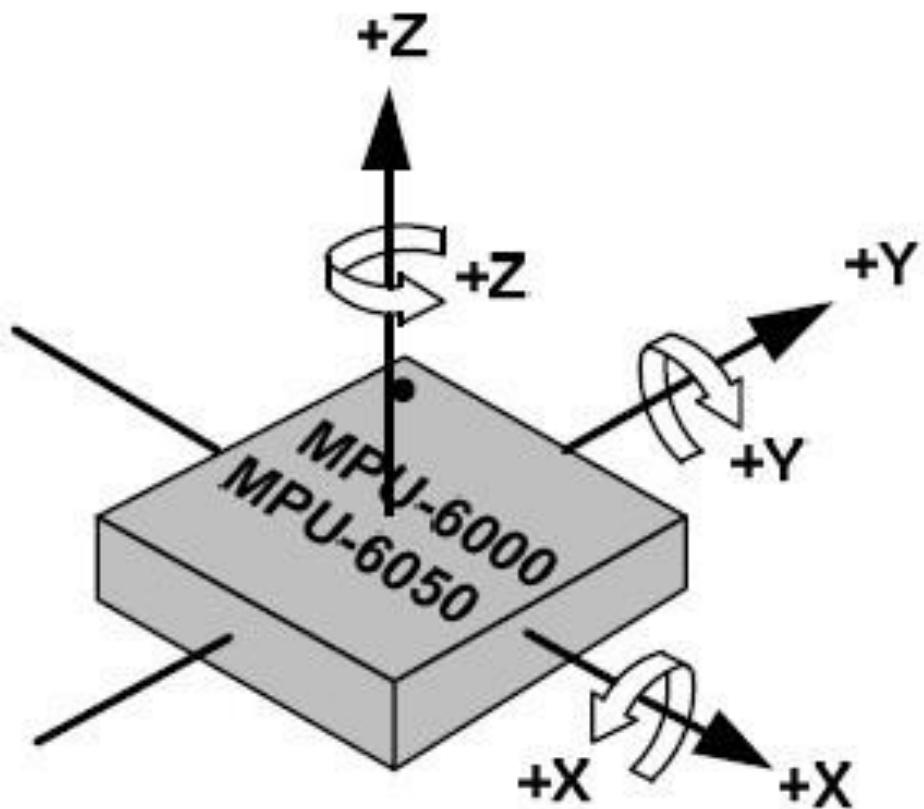
Wie funktioniert es?

IMU-Sensoren bestehen in der Regel aus zwei oder mehr Teilen. Aufgeführt nach ihrer Priorität, sind es: Beschleunigungsmesser, Gyroskop, Magnetometer und Höhenmesser. Die GY-521 ist ein 6 DOF (Freiheitsgrade) oder ein sechsachsiger IMU-Sensor, was bedeutet, dass es sechs Werte als Ausgang gibt. Drei Werte aus dem Beschleunigungsmesser und drei vom Gyroskop. Der GY-521 ist ein Sensor, der auf MEMS (Micro Electro Mechanical Systems) Technologie basiert. Sowohl der Beschleunigungsmesser als auch das Gyroskop sind in einem einzigen Chip eingebettet. Dieser Chip verwendet das I2C (Inter Integrated Circuit) Protokoll für die Kommunikation

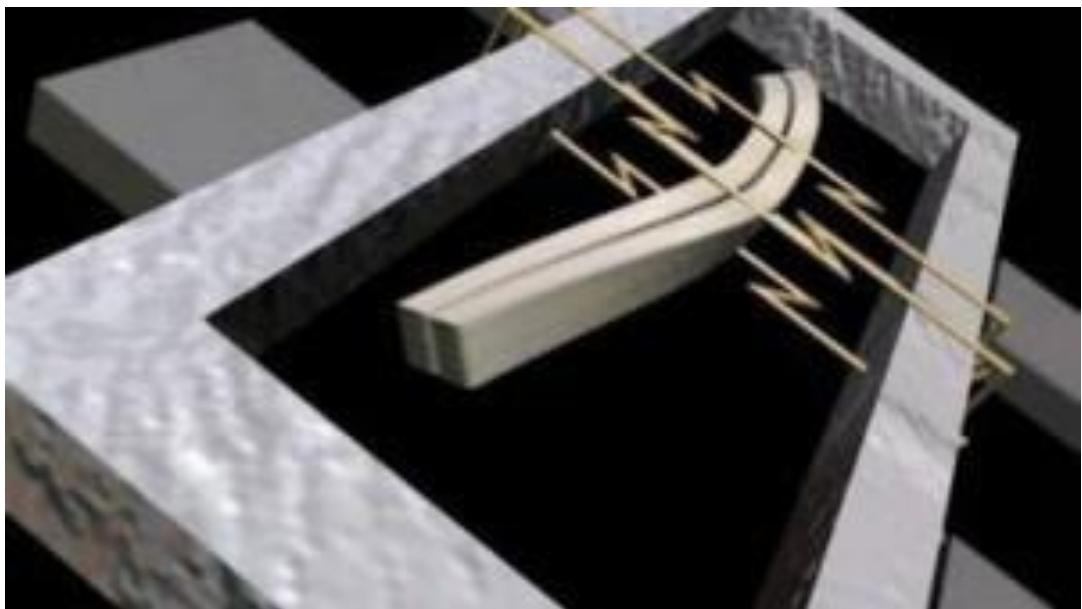
Wie funktioniert ein Beschleunigungsmesser?



Ein Beschleunigungsmesser arbeitet nach dem Prinzip der piezoelektrischen Wirkung. Hier stellen Sie sich eine quaderförmige Schachtel vor, mit einer kleinen Kugel drin, wie im Bild oben. Die Wände dieser Box sind mit piezoelektrischen Kristallen versehen. Immer wenn du den Kasten kippst, ist der Ball gezwungen, sich in Richtung der Neigung zu bewegen, wegen der Schwerkraft. Die Wand, mit der der Ball kollidiert, erzeugt winzige piezoelektrische Ströme. Es gibt drei Paare gegenüberliegender Wände in einem Quader. Jedes Paar entspricht einer Achse im 3D-Raum: X-, Y- und Z-Achse. Je nach dem Strom aus den piezoelektrischen Wänden können wir die Richtung der Neigung und deren Größe bestimmen. Weitere Informationen finden Sie hier.



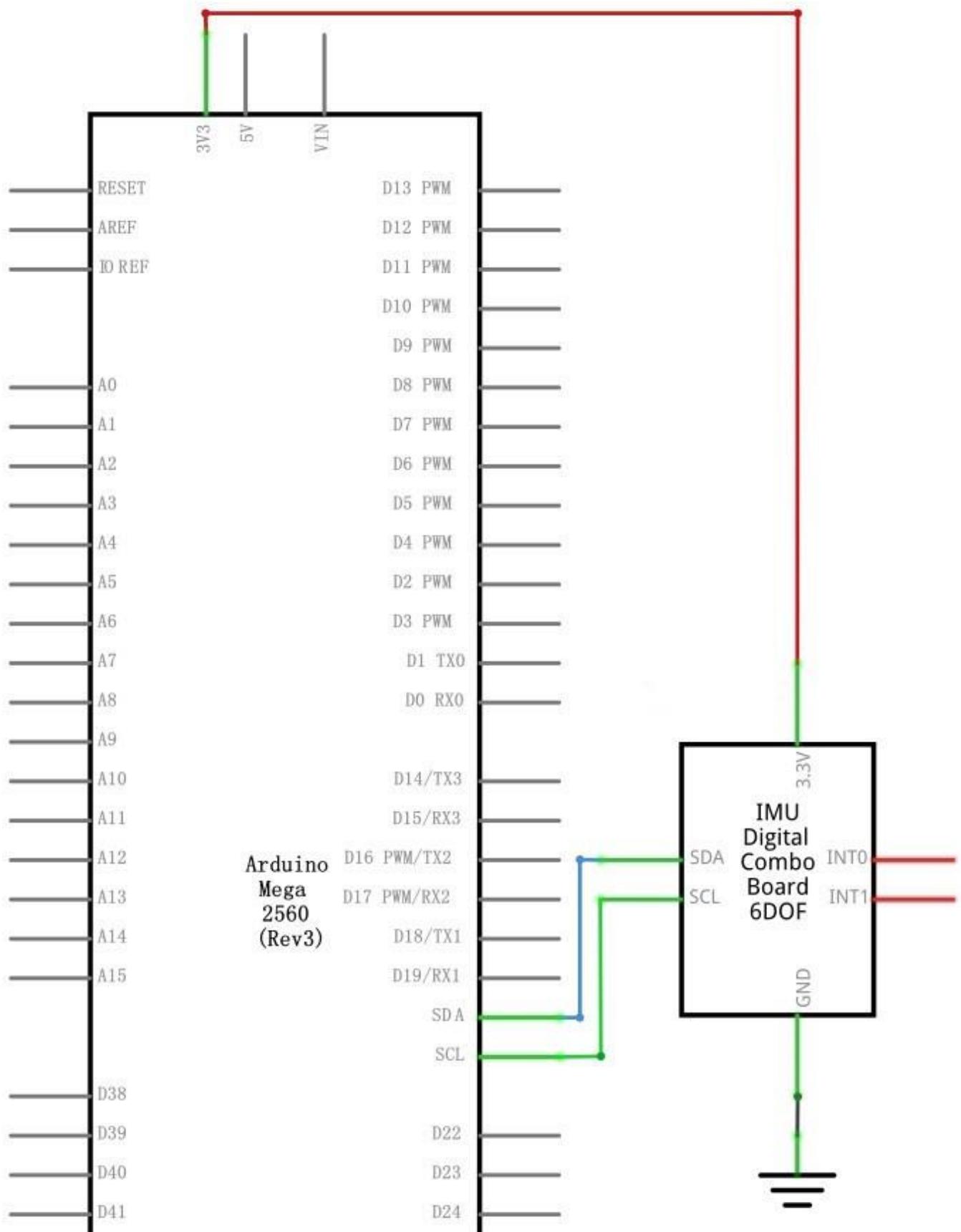
Wie funktioniert ein Gyroskop?



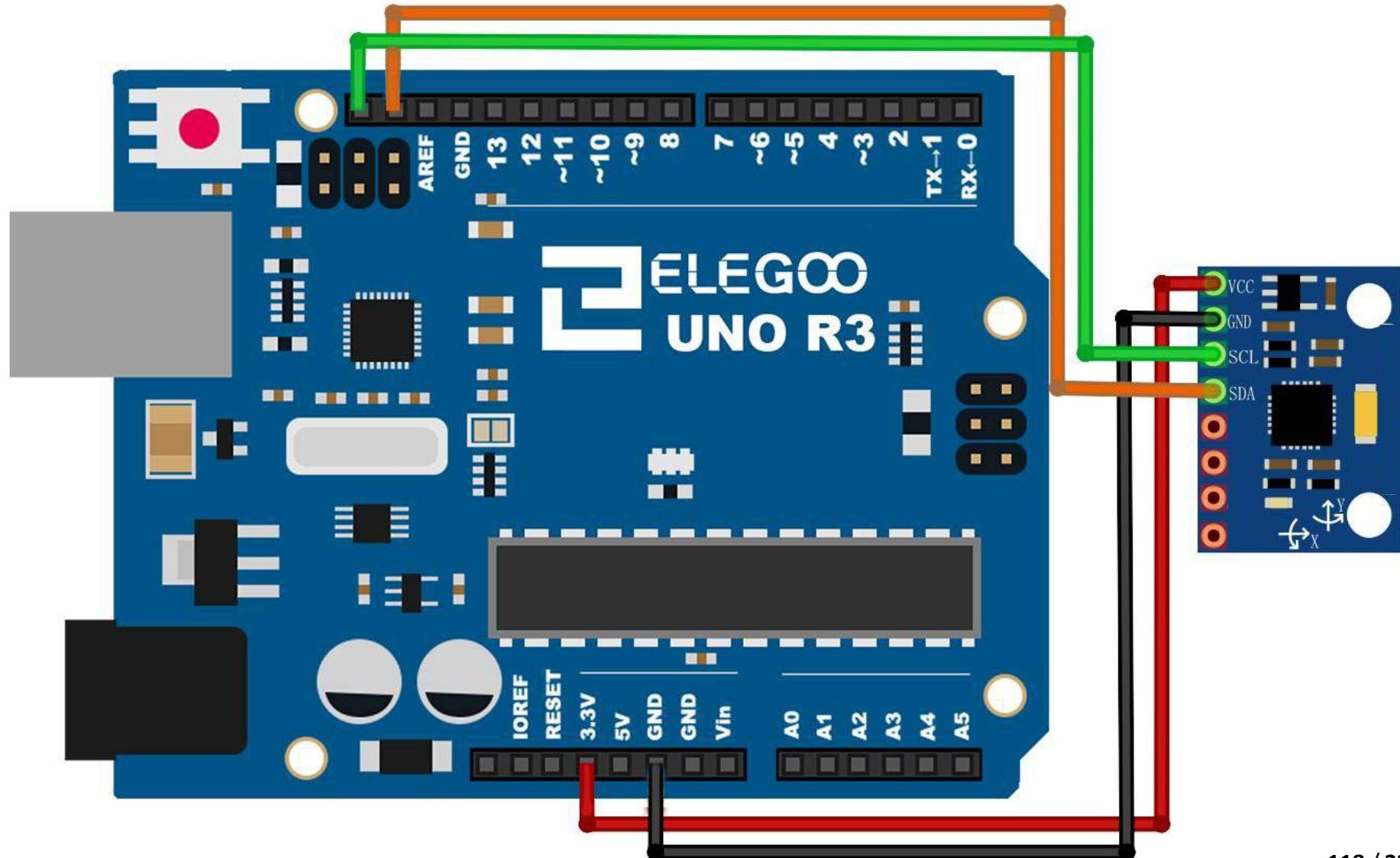
Gyroskope arbeiten nach dem Prinzip der Coriolis-Beschleunigung. Stellen Sie sich vor, dass es eine gabelartige Struktur gibt, die in ständiger Hin- und Herbewegung steht. Es wird mit piezoelektrischen Kristallen gehalten. Wann immer Sie versuchen, diese Anordnung zu kippen, erleben die Kristalle eine Kraft in Richtung der Neigung. Dies wird durch die Trägheit der Gabelung verursacht. Die Kristalle erzeugen also einen Strom im Konsens mit dem piezoelektrischen Effekt, und dieser Strom wird verstärkt. Die Werte werden dann vom Host-Mikrocontroller verfeinert.

Verbindung

Schema



Schaltplan



Als nächstes müssen wir die I2C-Leitungen einrichten. Dazu verbinden Sie den Pin SDA auf der GY-521 mit dem Arduino Analog Pin 4 (SDA). Und der Pin als SCL auf der GY-521 zum Arduino's Analog Pin 5 (SCL) markiert. Und das ist es, Sie haben den Arduino GY-521 verkabelt.

Benötigte Bibliotheken

MPU-6050

Der Code

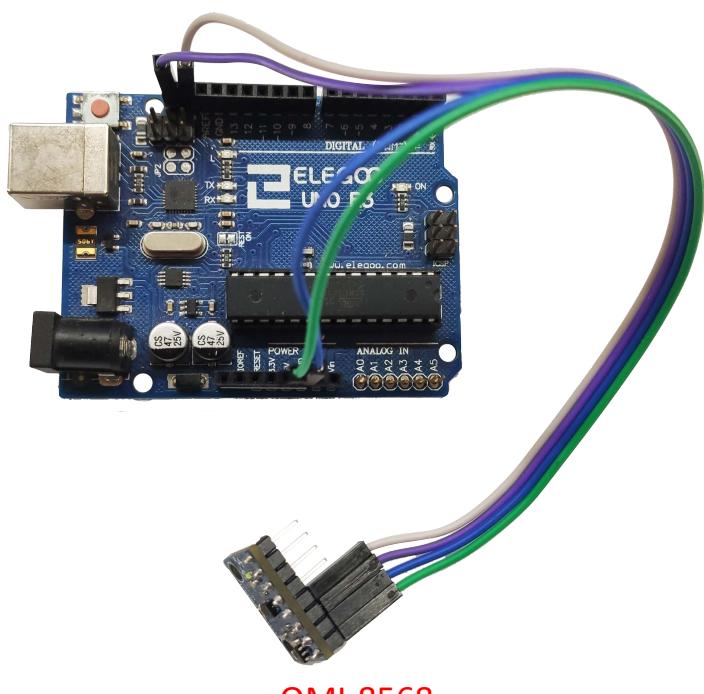
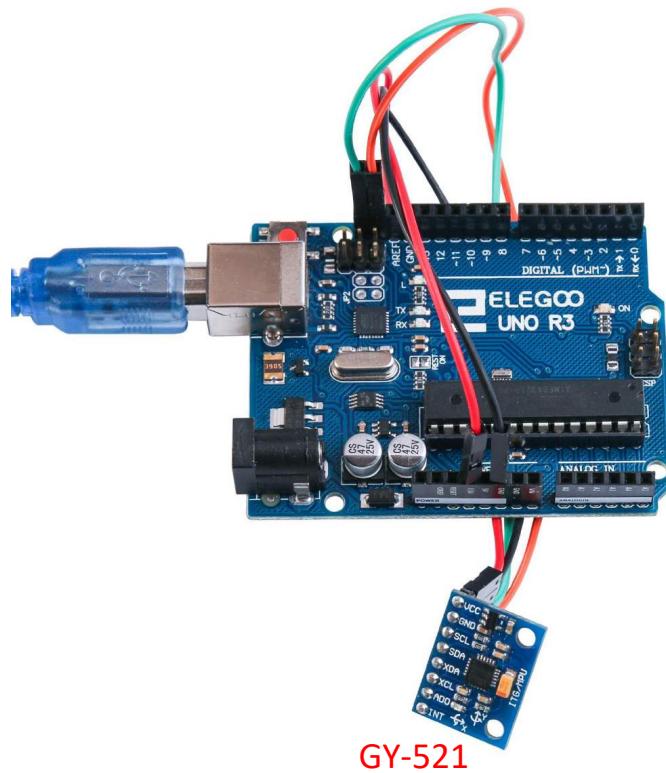
Der kurze Beispielsketch ist ein sehr kurzer Sketch und zeigt alle Rohwerte (Beschleunigungsmesser, Kreisel und Temperatur). Es sollte mit dem Arduino MEGA2560, Nano, Leonardo und auch Due funktionieren.

Nach der Verschalten öffnen Sie bitte das Programm im Codeordner - Lektion 16 GY-521 Modul und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die Bibliothek <GY-521> installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion1.

Beispiefoto



Öffnen Sie den seriellen Monitor, dann können Sie die Daten wie unten sehen:

Klicken Sie auf die Schaltfläche Serielle Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors werden in Lektion 1 ausführlich vorgestellt.

```
AcX = 15976 | AcY = -4280 | AcZ = -596 | Imp = 24.62 | GyX = 230 | GyY = -26 | GyZ = -1231
AcX = 15940 | AcY = -4408 | AcZ = -648 | Imp = 24.53 | GyX = -357 | GyY = -590 | GyZ = -519
AcX = 15852 | AcY = -4328 | AcZ = -668 | Imp = 24.62 | GyX = -189 | GyY = -79 | GyZ = -198
AcX = 15844 | AcY = -3972 | AcZ = -708 | Imp = 24.62 | GyX = -254 | GyY = -216 | GyZ = -44
AcX = 15740 | AcY = -4232 | AcZ = -968 | Imp = 24.53 | GyX = -319 | GyY = -141 | GyZ = -227
AcX = 15900 | AcY = -4936 | AcZ = -1008 | Imp = 24.62 | GyX = 126 | GyY = 111 | GyZ = 3870
AcX = 15356 | AcY = -5080 | AcZ = -1192 | Imp = 24.58 | GyX = -1670 | GyY = -1741 | GyZ = -2571
AcX = 14592 | AcY = -6504 | AcZ = -5700 | Imp = 24.53 | GyX = 662 | GyY = 264 | GyZ = 3219
AcX = 13740 | AcY = -7020 | AcZ = -2744 | Imp = 24.58 | GyX = 8265 | GyY = 4962 | GyZ = 8163
AcX = 3600 | AcY = -16556 | AcZ = 4244 | Imp = 24.58 | GyX = -17048 | GyY = -12197 | GyZ = 3845
AcX = 12248 | AcY = -12292 | AcZ = 7256 | Imp = 24.62 | GyX = 12046 | GyY = 24428 | GyZ = -5483
AcX = 588 | AcY = -3832 | AcZ = 19208 | Imp = 24.53 | GyX = 9258 | GyY = -4420 | GyZ = -4557
AcX = 1896 | AcY = -3784 | AcZ = 6320 | Imp = 24.62 | GyX = -7486 | GyY = -32768 | GyZ = 2677
AcX = 32767 | AcY = -19068 | AcZ = -1920 | Imp = 24.58 | GyX = -9262 | GyY = -19403 | GyZ = 25320
AcX = -19160 | AcY = 12004 | AcZ = -2452 | Imp = 24.58 | GyX = -32768 | GyY = -32768 | GyZ = -4809
AcX = -25124 | AcY = 1616 | AcZ = 32767 | Imp = 24.62 | GyX = 7628 | GyY = 7064 | GyZ = 6299
AcX = 11976 | AcY = -8432 | AcZ = -32600 | Imp = 24.53 | GyX = 29381 | GyY = 32767 | GyZ = -19841
AcX = 972 | AcY = -22992 | AcZ = -12480 | Imp = 24.62 | GyX = -31051 | GyY = -32768 | GyZ = 32767
AcX = -27260 | AcY = 16868 | AcZ = 10704 | Imp = 24.62 | GyX = 32767 | GyY = 28603 | GyZ = -20636
AcX = 32268 | AcY = -32468 | AcZ = -21952 | Imp = 24.58 | GyX = -27684 | GyY = -32768 | GyZ = 32767
AcX = -22476 | AcY = -8436 | AcZ = -3976 | Imp = 24.58 | GyX = 32156 | GyY = 32767 | GyZ = 25696
AcX = -3836 | AcY = -13428 | AcZ = -9628 | Imp = 24.58 | GyX = -30925 | GyY = -32768 | GyZ = 32767
AcX = 3164 | AcY = -5392 | AcZ = -19464 | Imp = 24.48 | GyX = -30769 | GyY = -17986 | GyZ = 17236
AcX = 3408 | AcY = -3584 | AcZ = -13752 | Imp = 24.58 | GyX = 1820 | GyY = -2660 | GyZ = -186
AcX = 4404 | AcY = -5552 | AcZ = -15216 | Imp = 24.58 | GyX = -578 | GyY = -234 | GyZ = -425
AcX = 4160 | AcY = -5456 | AcZ = -15304 | Imp = 24.53 | GyX = -445 | GyY = -154 | GyZ = -277
AcX = 4152 | AcY = -5192 | AcZ = -15300 | Imp = 24.53 | GyX = -404 | GyY = -114 | GyZ = -262
```

Lektion 17 HC-SR501 PIR Sensor

Überblick

In dieser Lektion lernen Sie, wie man einen PIR Bewegungsmelder mit einem MEGA2560 benutzt.

Der MEGA2560 ist das Herzstück dieses Projektes. Es "hört" die ganze Zeit dem PIR-Sensor zu und wenn eine Bewegung erkannt wird, weist er die LED an, an- oder abzuschalten.

Erforderliche Komponente:

- (1) x Elegoo Mega2560 R3
- (1) x HC-SR501 PIR Bewegungssensor
- (3) x F-M Kabel (Female to Male DuPont Kabel)

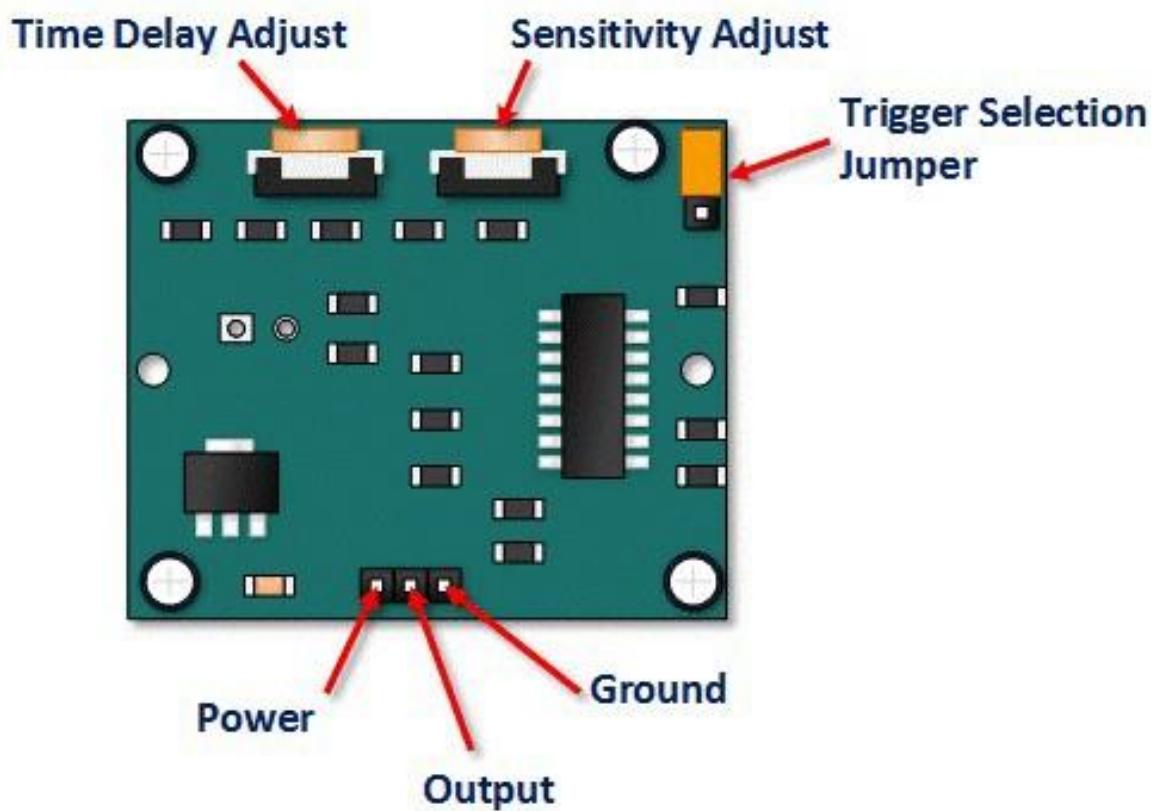
Komponenten Einführung

PIR SENSOR:

PIR-Sensoren sind komplizierter als viele der anderen in diesem Tutorial erläuterten Sensoren (wie LDR, FSRs und Kippschalter), da es mehrere Variablen gibt, die die Ein- und Ausgänge der Sensoren beeinflussen.

Der PIR-Sensor selbst hat zwei Steckplätze. Jeder Schlitz besteht aus einem speziellen Material, das empfindlich auf IR ist. Das hier verwendete Objektiv macht nicht wirklich viel und so sehen wir, dass die beiden Slots in einiger Entfernung (im Grunde die Empfindlichkeit des Sensors) sehen können. Wenn der Sensor im Leerlauf ist, erkennen beide Schlitze die gleiche Menge an IR, das ist die Umgebungsmenge, die von dem Raum oder den Wänden oder im Freien abgestrahlt wird. Wenn ein warmer Körper wie ein Mensch oder ein Tier vorbeikommt, fängt er zuerst die Hälfte des PIR-Sensors ab, was einen positiven Differenzunterschied zwischen den beiden Hälften bewirkt. Wenn der warme Körper den Erfassungsbereich verlässt, passiert das Gegenteil, wobei der Sensor eine negative Differentialänderung erzeugt. Diese Änderungsimpulse sind das, was erkannt wird.





| Pin or Control | Funktion |
|--------------------------|--|
| Time Delay Adjust | Sets how long the output remains high after detecting motion.... Anywhere from 5 seconds to 5 minutes. |
| Sensitivity Adjust | Sets the detection range.... from 3 meters to 7 meters |
| Trigger Selection Jumper | Set for single or repeatable triggers. |
| Ground pin | Ground input |
| Output Pin | Low when no motion is detected.. High when motion is detected. High is 3.3V |
| Power Pin | 5 to 20 VDC Supply input |

HC SR501 PIR Funktionsbeschreibung

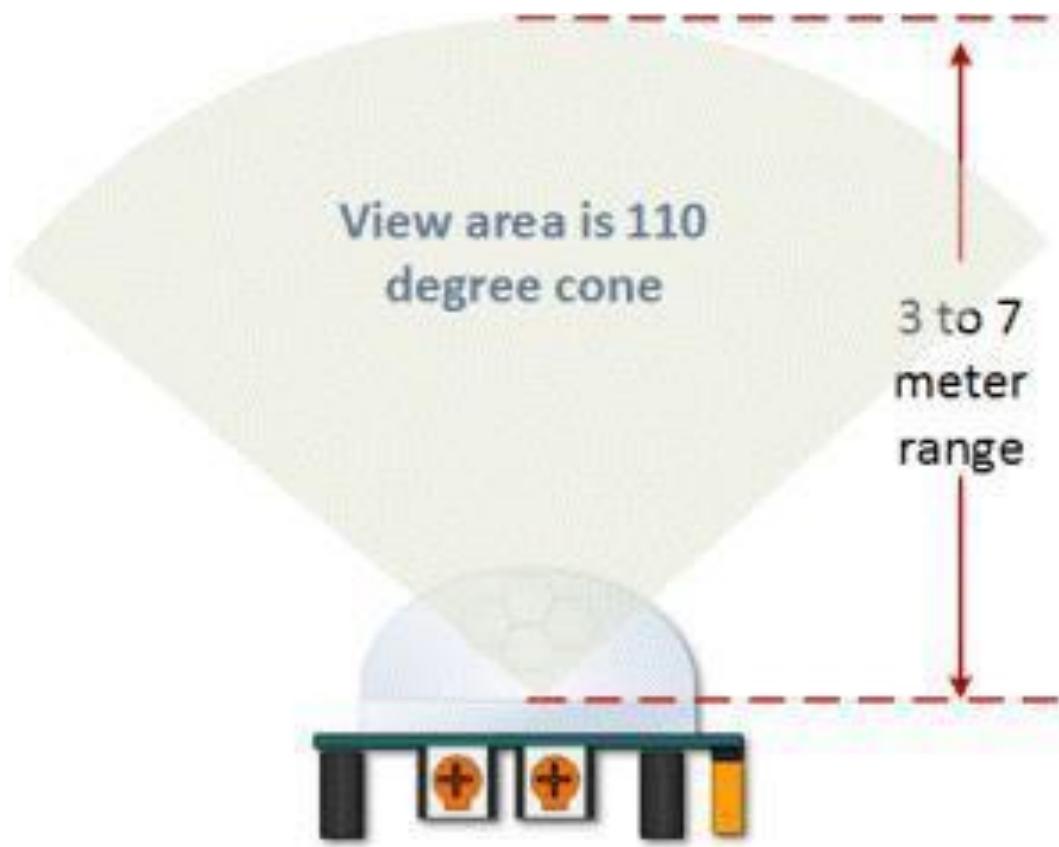
Der SR501 erkennt Infrarotänderungen und wenn er als Bewegung interpretiert wird, wird der Ausgang "niedrig" eingestellt. Was als Bewegung oder als keine Bewegung interpretiert wird, ist weitgehend abhängig von Benutzereinstellungen und weiteren Anpassungen.

Geräteinitialisierung

Das Gerät benötigt fast eine Minute, um zu initialisieren. Während dieser Zeitspanne kann es und wird oft falsche Erkennungssignale ausgeben. Schaltkreise- oder Reglerlogiken müssen diese Initialisierungsperiode berücksichtigen.

Erkennungsbereich des Sensors

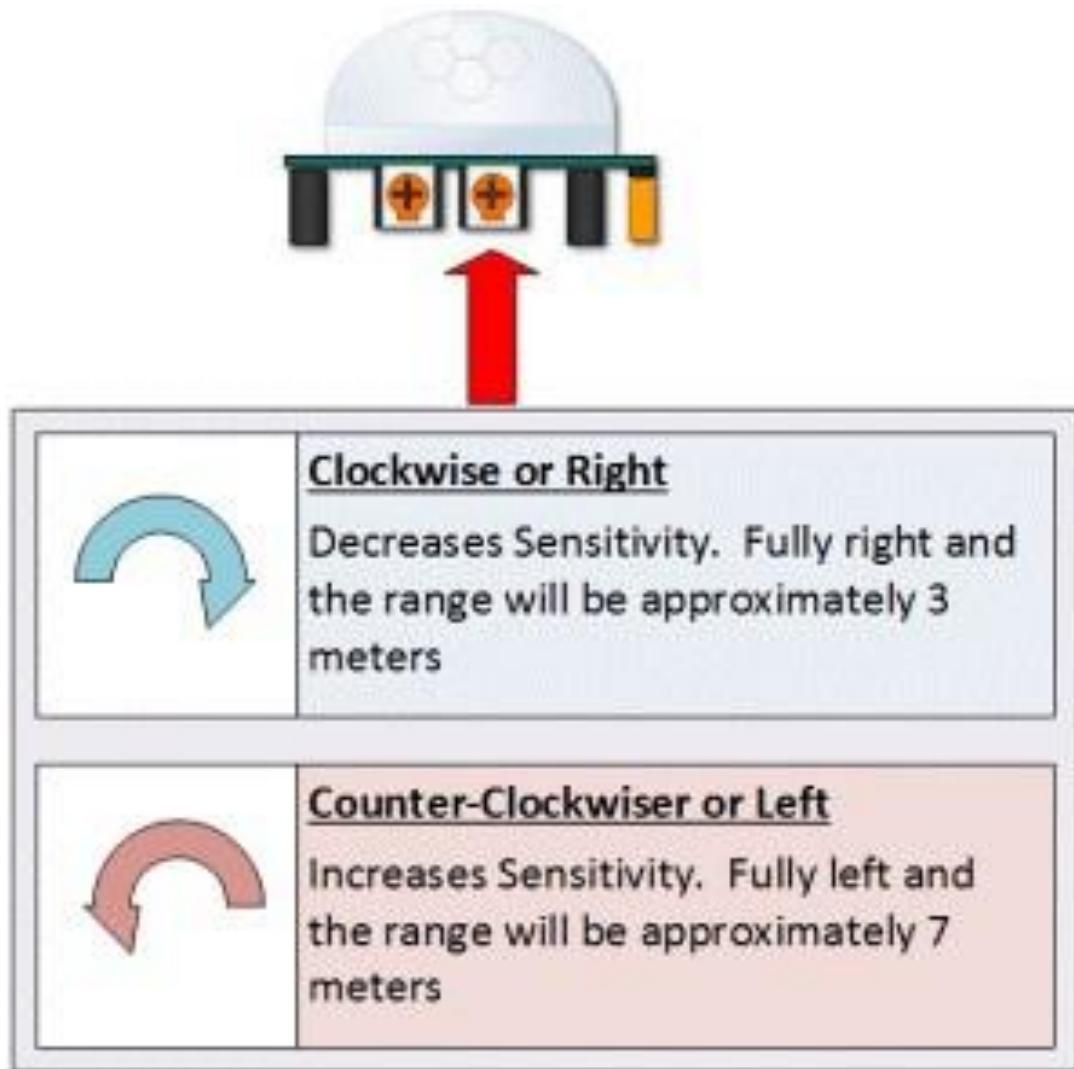
Das Gerät erkennt die Bewegung innerhalb eines 110-Grad-Kegels mit einer Reichweite von 3 bis 7 Metern.



HC SR501 Sichtbereich

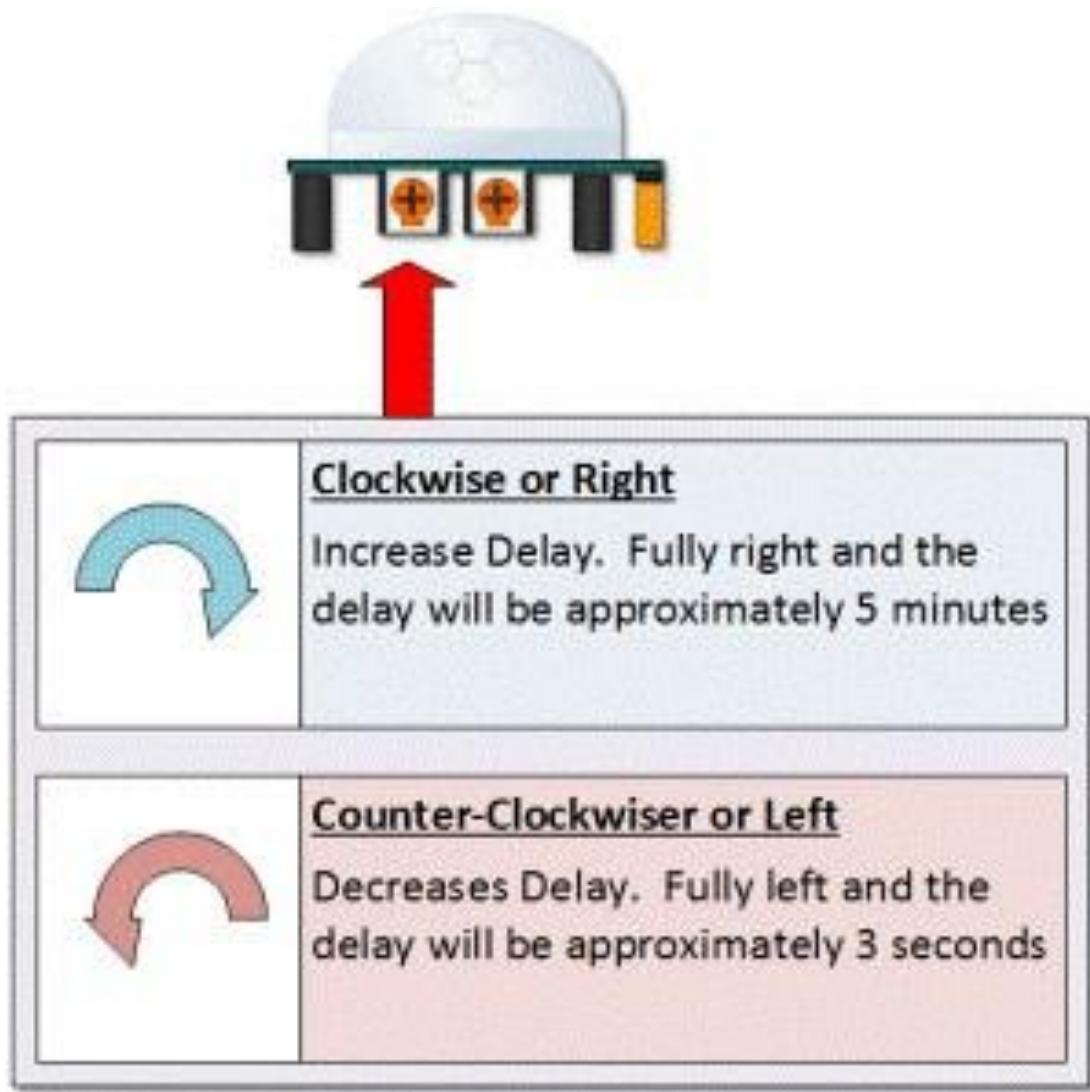
PIR-Bereich (Empfindlichkeit) Einstellung

Wie bereits erwähnt, beträgt der einstellbare Bereich ca. 3 bis 7 Meter. Die Abbildung unten zeigt diese Einstellung.



HC SR501 Empfindlichkeit Einstellung der Zeitverzögerung

Die Zeitverzögerungseinstellung bestimmt, wie lange der Ausgang des PIR-Sensormoduls nach der Erkennungsbewegung "HIGH" bleibt. Die Dauer kann von ca. 3 Sekunden bis 5 Minuten betragen.



HC SR501 Zeitverzögerung

WICHTIG: 3 Sekunden ausgeschaltet lassen, nachdem die Zeitverzögerung abgeschlossen ist!

Der Ausgang dieses Gerätes geht ca. 3 Sekunden lang auf LOW (oder Aus). Nach Ablauf dieser Zeitverzögerung wird die Zeitverzögerung abgeschlossen. Mit anderen Worten, ALLE Bewegungserkennungen während dieser drei Sekunden Periode werden blockiert.

Zum Beispiel:

Stellen Sie sich vor, Sie sind im Single-Trigger-Modus und Ihre Zeitverzögerung ist auf 5 Sekunden eingestellt.

Die PIR erkennt die Bewegung und setzt sie 5 Sekunden lang "HIGH".

Nach fünf Sekunden setzt der PIR seinen Ausgang für ca. 3 Sekunden "LOW".

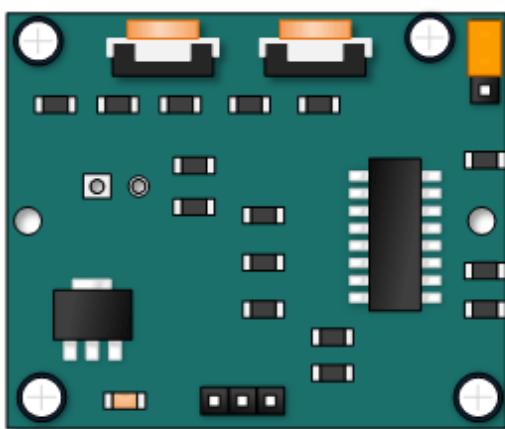
Während der drei Sekunden erkennt die PIR keine Bewegung.

Nach drei Sekunden erkennt die PIR die Bewegung wieder und die erkannte Bewegung setzt den Ausgang wieder hoch.

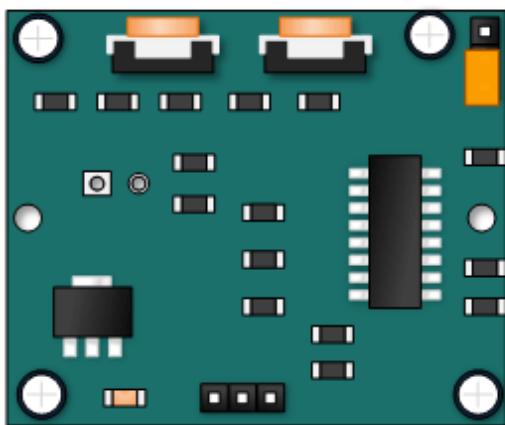
Trigger Auswahl- Jumper

Mit dem Trigger-Modus-Auswahl-Jumper können Sie zwischen einzelnen und wiederholbaren Triggern auswählen. Für Jumper-Einstellung ist zu bestimmen, wann die Zeitverzögerung beginnt, damit dies ordentlich funktioniert.

- **SINGLE TRIGGER** – Die Zeitverzögerung beginnt sofort, wenn die Bewegung zuerst erkannt wird.
- **REPEATABLE TRIGGER** – Jede erkannte Bewegung setzt die Zeitverzögerung zurück. So beginnt die Zeitverzögerung mit der zuletzt erfassten Bewegung.



Single Trigger Mode – Time Delay is started immediately upon detecting motion. Continued detection is blocked



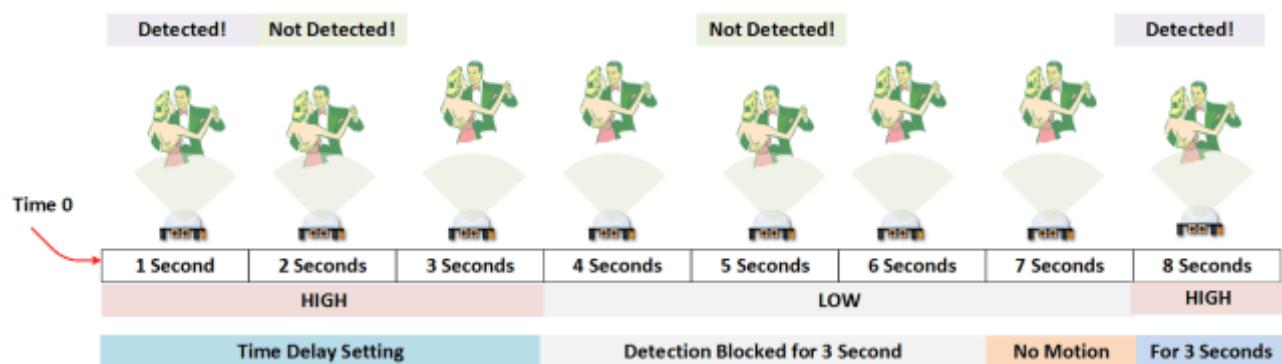
Repeatable Trigger Mode – Time Delay is re-started every time motion is detected.

HC-SR501 Tanzflächen Anwendungsbeispiele

Stellen Sie sich vor, Sie wollen die Beleuchtung auf einer Tanzfläche steuern, basierend darauf, wo die Tänzer tanzen. Verstehen, wie die Zeitverzögerung und der Trigger-Modus miteinander interagieren müssen, um die Beleuchtung in der Art und Weise, die Sie wollen.

Beispiel Eins

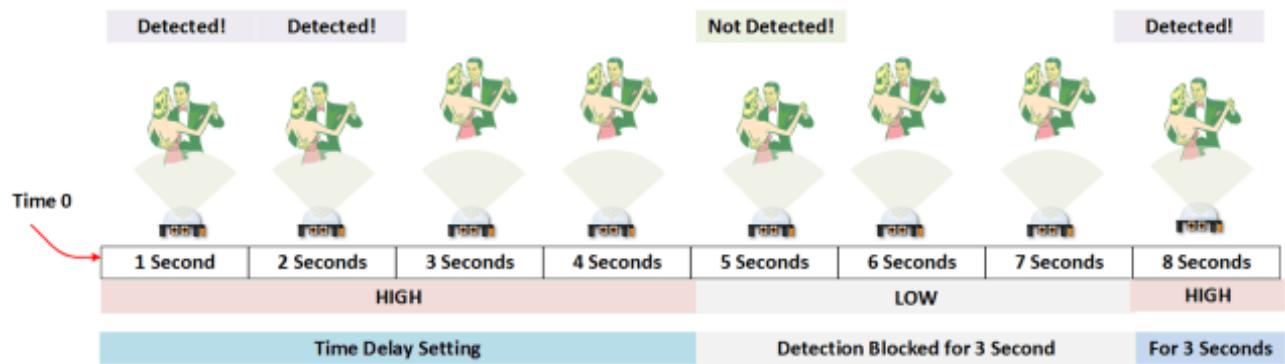
In diesem ersten Beispiel wird die Zeitverzögerung auf drei Sekunden gesetzt und der Trigger-Modus wird auf Single gesetzt. Wie Sie in der folgenden Abbildung sehen können, wird die Bewegung nicht immer erkannt. In der Tat gibt es einen Zeitraum von etwa sechs Sekunden, wo Bewegungen nicht erkannt werden können. Fühlen Sie sich frei, auf das Bild zu klicken, um zu vergrößern.



Beispiel Zwei

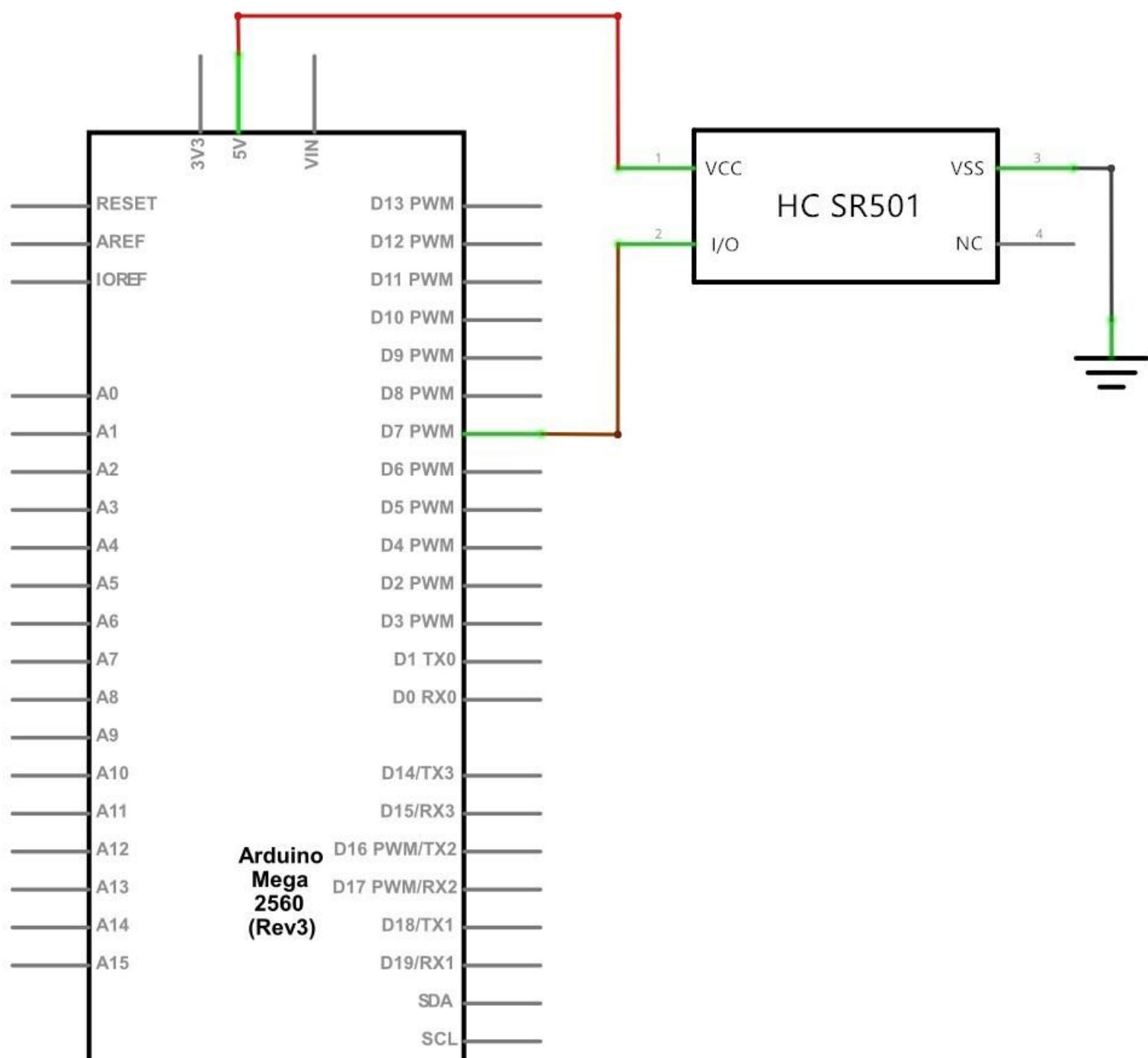
Im nächsten Beispiel ist die Zeitverzögerung immer noch auf drei Sekunden eingestellt und der Trigger ist auf wiederholbar eingestellt. In der folgenden Abbildung sehen Sie, dass die Zeitverzögerungsperiode neu gestartet wird. Nach drei Sekunden wird die Erkennung jedoch noch drei Sekunden lang gesperrt.

Wie ich bereits erwähnt habe, könnten Sie die 3-Sekunden-Sperrperiode miteinem kreativen Code überschreiben, aber lassen wir das vorerst einmal nur eine Überlegung sein. Einige der Elektronik, die Sie verwenden, ist nicht auf ein plötzliches ein und ausschalten ausgelegt. Die drei Sekunden erlauben diesen Bauteilen ein wenig Ruhe vor dem Starten.

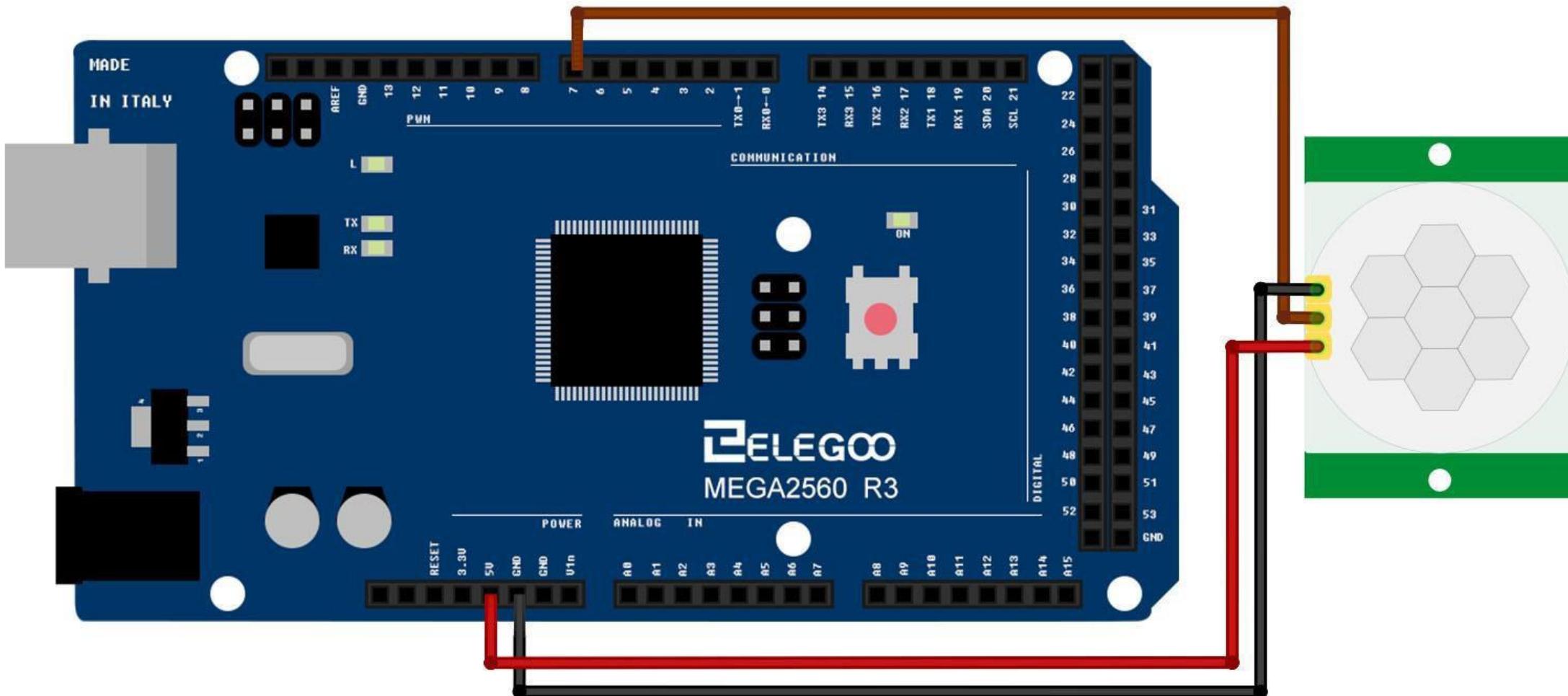


Verbindung

Schema



Schaltplan



Das Anschließen von PIR-Sensoren an einen Mikrocontroller ist wirklich einfach. Die PIR fungiert als Digitalausgang, so dass alles, was Sie tun müssen, ist zu "hören", ob der Pin "HIGH" ist (erkannt) oder "LOW" (nicht erkannt).

Es ist wahrscheinlich, dass Sie den Sensor retriggern wollen. Seien Sie sich also sicher, dass Sie den Jumper in die H-Position gesteckt haben!

Schließen Sie den PIR mit 5V und Masse an Masse an. Verbinden Sie dann den Ausgang mit einem digitalen Pin. In diesem Beispiel verwenden wir Pin 7.

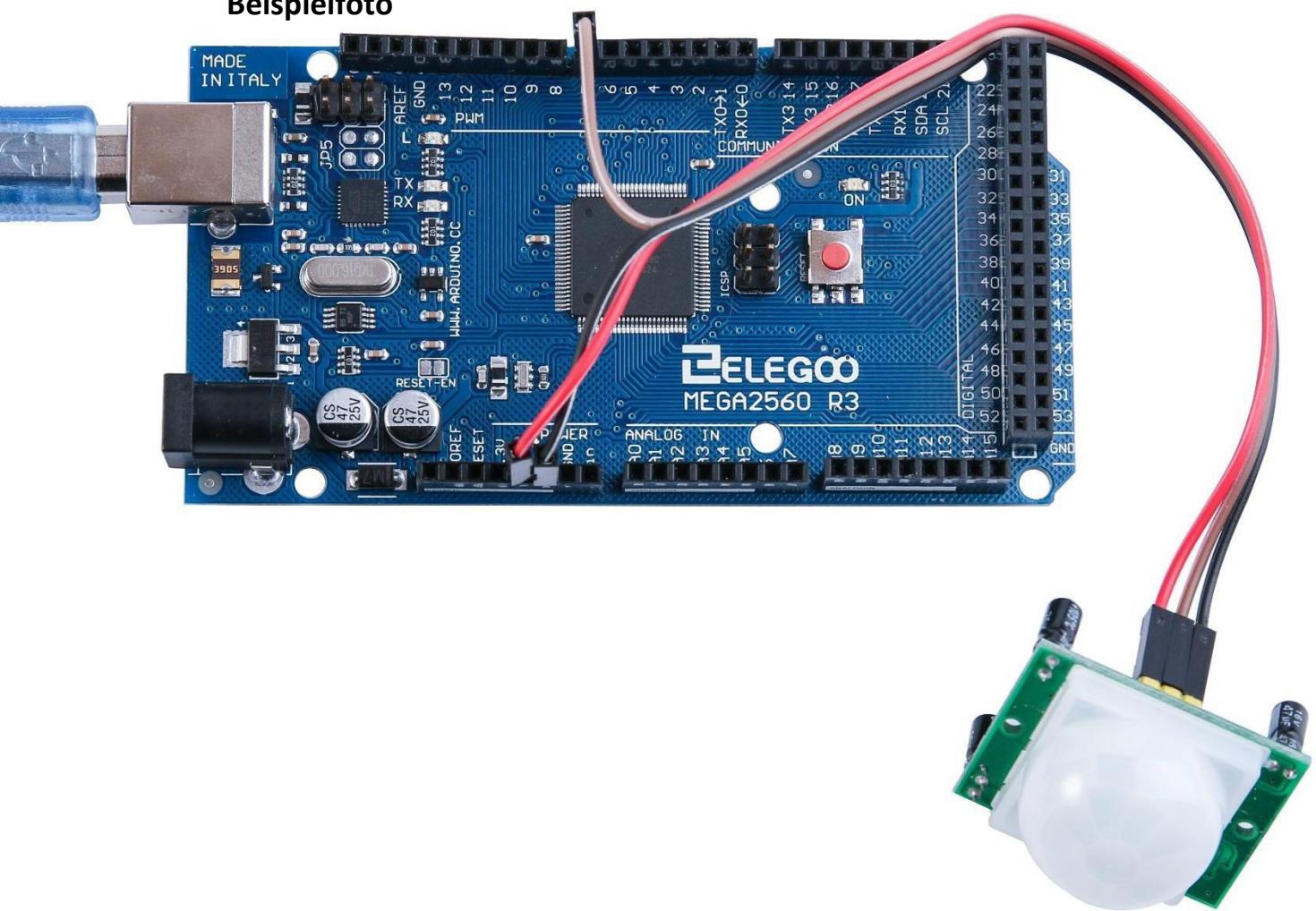
Code

Nach der Verschaltung öffnen Sie bitte das Programm im Codeordner - Lektion 17 HC-SR501 PIR Sensor und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Der Sketch schaltet einfach Ihre Arduino-LED ein, die an Pin 13 angeschlossen ist, wenn eine Bewegung erkannt wird.

Beachten Sie die 1 Minute andauernde Initialisierung in jeder Anwendung die Sie entwickeln.

Beispielfoto



Lektion 18 Wasserstandserkennungssensormodul

Überblick

In dieser Lektion lernen Sie, wie man ein Wasserstandserkennungssensormodul verwendet.

Dieses Modul kann die Tiefe von Wasser wahrnehmen und die Kernkomponente ist eine Verstärkerschaltung, die aus einem Transistor und mehreren Pectin-PCB-Routinen besteht. Wenn sie in das Wasser gelegt werden, werden diese Leitungen einen Widerstand darstellen, der sich mit dem Wechsel der Wassertiefe ändern kann. Dann wird das Signal der Wassertiefe in ein elektrisches Signal umgewandelt, und wir können die Veränderung der Wassertiefe durch die ADC-Funktion des MEGA2560 R3 erkennen.

Erforderliche Komponente:

- (1) x Elegoo Mega2560 R3
- (3) x F-M Kabel (Female to Male DuPont Kabel)
- (1) x Wasserstandserkennungssensormodul

Komponenten Einführung

Wassersensor:



Wassersensor Module sind für die Wasser-Erkennung weit verbreitet und finden zum Beispiel in der Erfassung von Niederschlag, Wasserständen ihre Verwendung. Der Sensor besteht hauptsächlich aus drei Teilen: einem elektronischen Sensorstecker, einem $1\text{ M}\Omega$ Widerstand und mehreren Leitungen aus blanken Drähten.

Dieser Sensor arbeitet mit einer Reihe von exponierten Strängen, die mit der Masse verbunden sind. Zwischen den geerdeten Strängen liegen die Sensorstränge.

Die Sensorstränge haben einen schwachen Pull-up-Widerstand von $1\text{ M}\Omega$. Der Widerstand zieht den Sensor-Trace-Wert hoch, bis ein Wassertropfen den Sensor auf den geerdeten Strängen umschaltet. Ob Sie es glauben oder nicht, diese Schaltung funktioniert mit den digitalen I/O Pins des Mega2560 R-3. Sie können sie aber auch mit den analogen Pins verwenden um die Menge des wasserinduzierten Kontakts zwischen den geerdeten und den Sensorspuren zu erkennen.

Dieser Sensor kann den Wasserstand durch eine Reihe von exponierten, parallelen

Drähten und Maschen einen Wassertropfen und auch eine Wassermengemessen.
Der Sesnor kann leicht eingestellt werden, sodass die Wassermenge als Analogsignal ausgegeben wird und der Ausgangs Analogwert kann direkt in der Programmfunction verwendet werden, um dann Beispielsweise des Wasserstandsalarm auszugeben.

Der Sensor hat einen sehr geringen Stromverbrauch und eine hohe Empfindlichkeit.

Eigenschaften:

1、Betriebsspannung: 5V

2、Betriebsstrom: <20ma 3、

Schnittstelle: Analog 4、

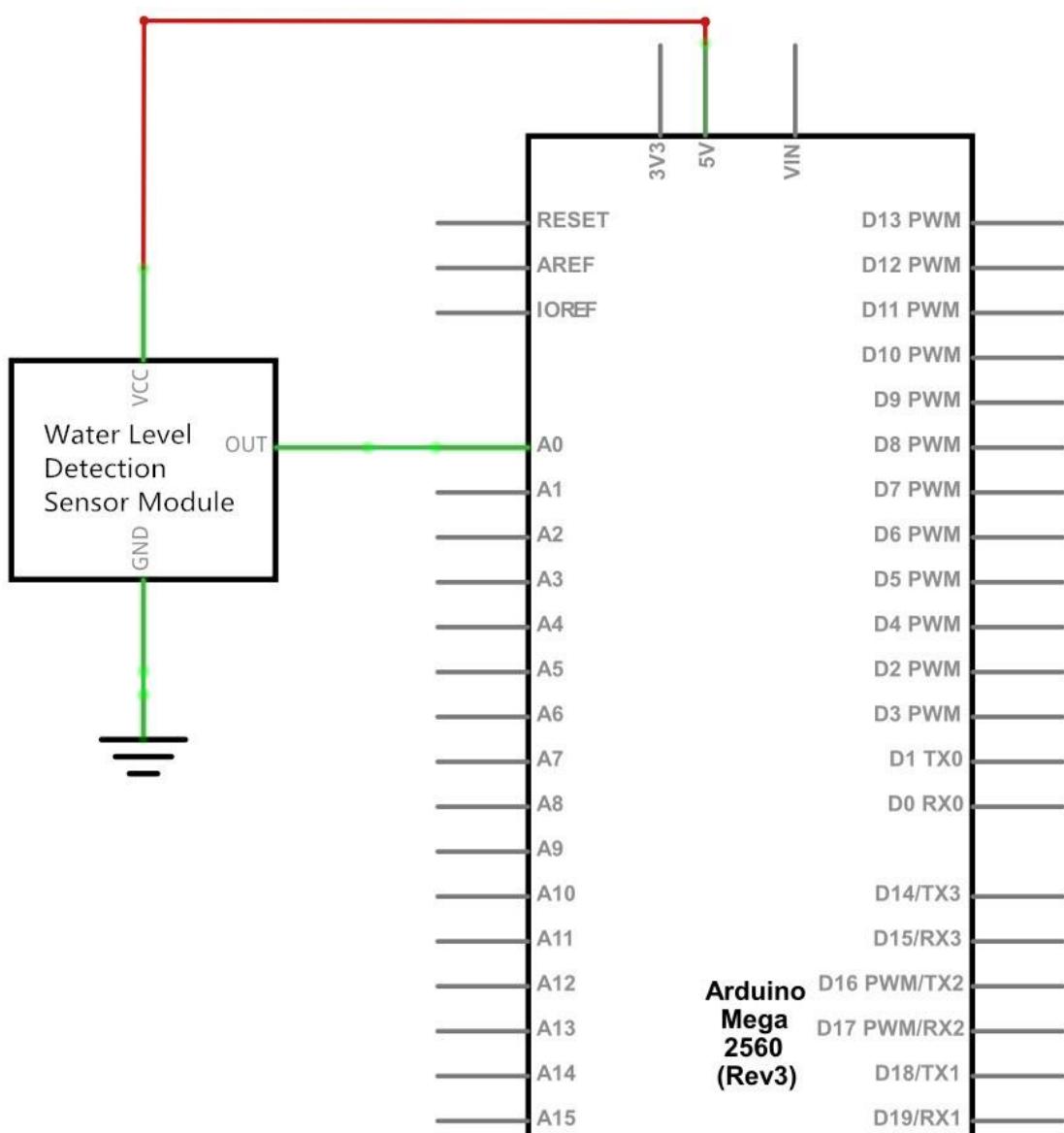
Erkennungsbereich: 40mm×16mm 5、

Arbeitstemperatur: 10°C~30°C

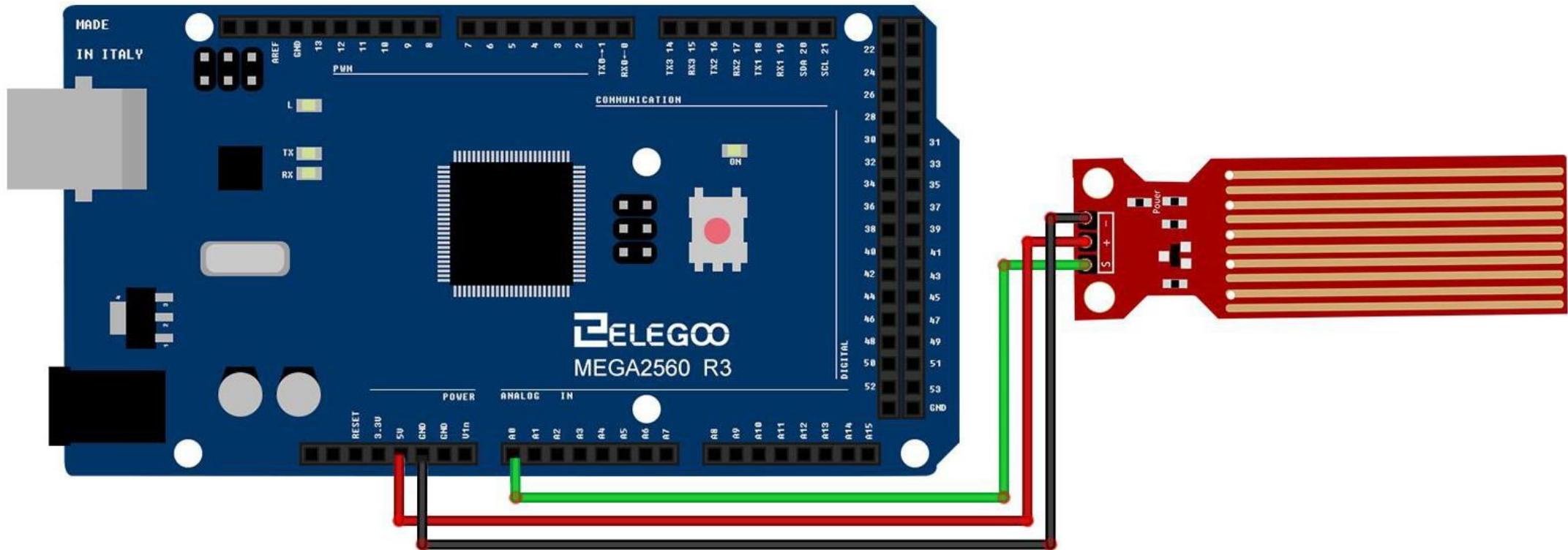
6、Ausgangsspannungssignal: 0~4.2V

Verbindung

Schema



Schaltplan

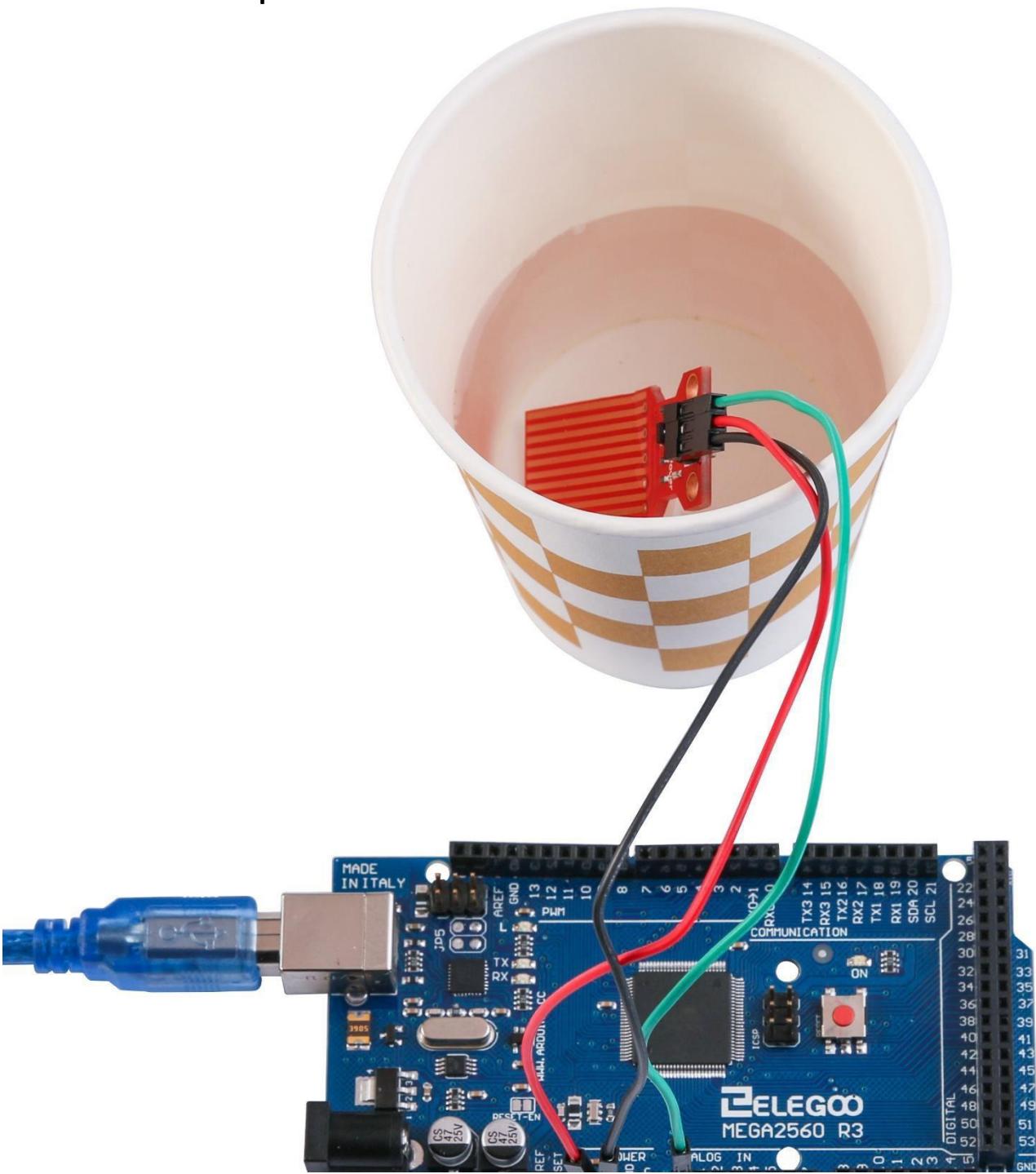


Vorschaltungstipps: Die Stromversorgung (+) ist an 5V der MEGA2560 R3 Platine angeschlossen, die Masseelektrode (-) ist an GND angeschlossen. Der Signalausgang (S) ist mit einem der Ports (A0-A5) verbunden werden, da diese die Funktion haben das analoge Signal im MEGA2560 R3-Platine zu empfangen und zu verarbeiten, es ist egal welches dieser Analog Ports Sie nutzen, aber Sie sollten den gleichen Port dann in dem Demo-Code definieren, sonst funktioniert dies nicht.

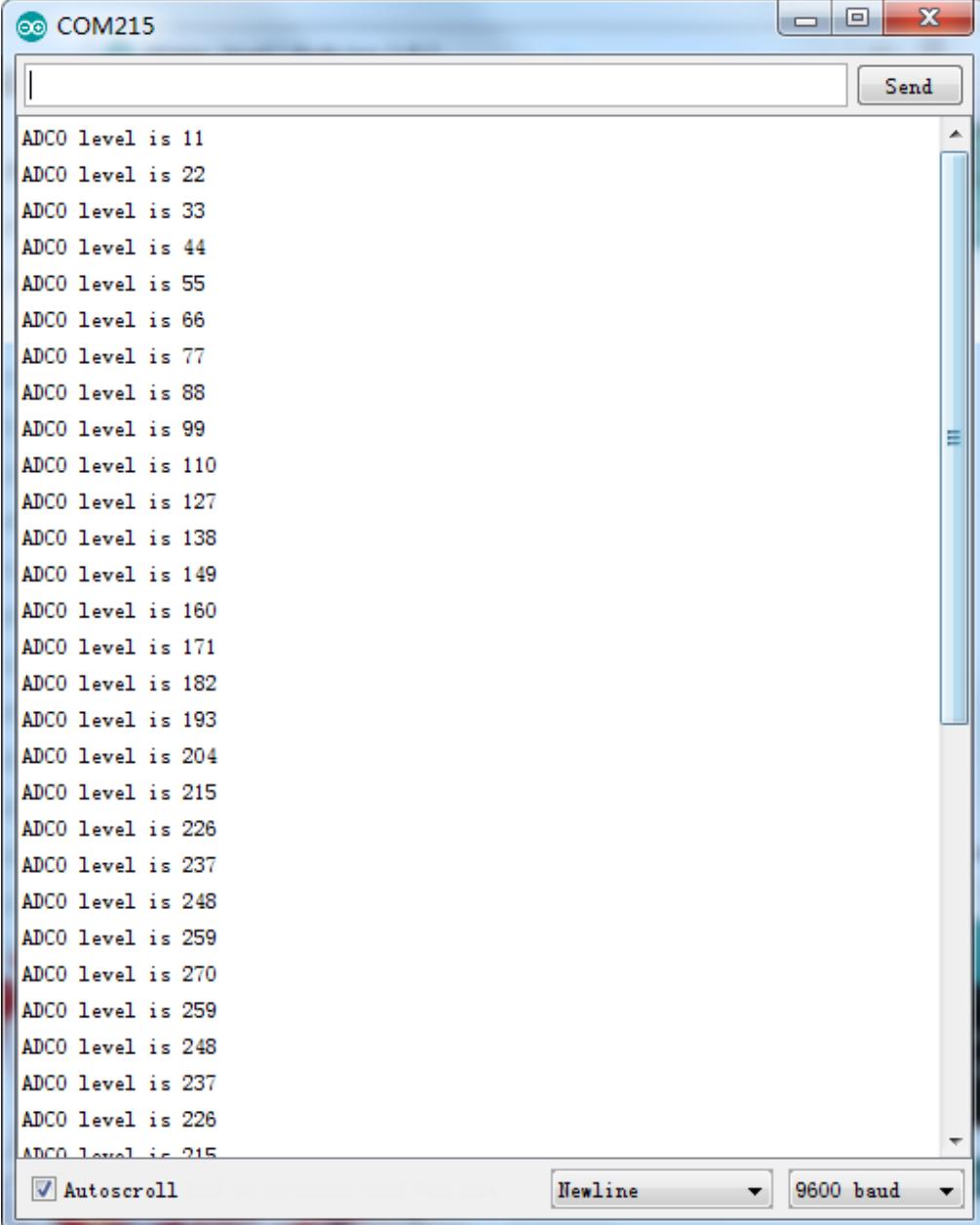
Code

Nach der Verschalten öffnen Sie bitte das Programm im Codeordner - Lektion 18 Wasserstandserkennungssensormodul und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Beispelfoto



Öffnen Sie den seriellen Monitor, dann können Sie die Daten wie unten sehen:
Klicken Sie auf die Schaltfläche Serial Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors werden in Lektion 1 ausführlich vorgestellt.



The screenshot shows a Windows-style application window titled "COM215". The main area is a scrollable text box displaying a series of lines of text. Each line represents a reading from an ADC0 pin, starting at level 11 and increasing in increments of 1 up to 259, then decreasing back down to 215. The text box has a "Send" button in the top right corner. At the bottom of the window, there are three control buttons: "Autoscroll" (checked), "Newline" (set to "Newline"), and "9600 baud" (set to "9600 baud").

```
ADC0 level is 11
ADC0 level is 22
ADC0 level is 33
ADC0 level is 44
ADC0 level is 55
ADC0 level is 66
ADC0 level is 77
ADC0 level is 88
ADC0 level is 99
ADC0 level is 110
ADC0 level is 127
ADC0 level is 138
ADC0 level is 149
ADC0 level is 160
ADC0 level is 171
ADC0 level is 182
ADC0 level is 193
ADC0 level is 204
ADC0 level is 215
ADC0 level is 226
ADC0 level is 237
ADC0 level is 248
ADC0 level is 259
ADC0 level is 270
ADC0 level is 259
ADC0 level is 248
ADC0 level is 237
ADC0 level is 226
ADC0 level is 215
```

Lektion 19 Echtzeituhrmodul

Überblick

In dieser Lektion lernen Sie, wie man mit dem DS3231 Uhrmodul, das Jahr, Monat, Tag, Stunde, Minute, Sekunde und Woche anzeigt. Man kann dieses Modul über ein Backup-Akku verwenden, solange es mit nur drei Datenkabeln am MEGA2560 angeschlossen ist.

Erforderliche Komponenten:

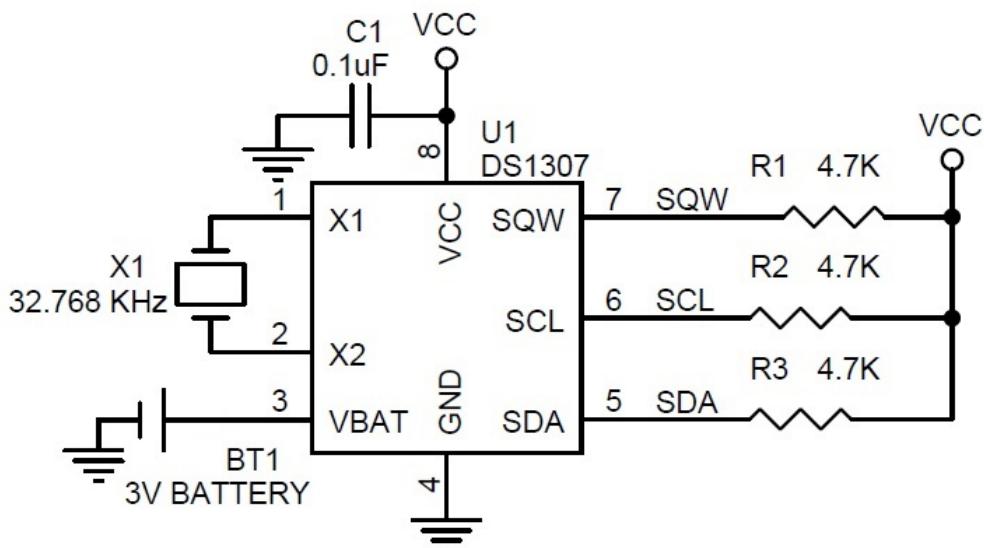
- (1) x Elegoo Mega2560 R3
 - (1) x DS3231 RTC Moduk
 - (4) x F-M Kabel (Female to Male DuPont Kabel)



Komponenten Einführung

DS3231

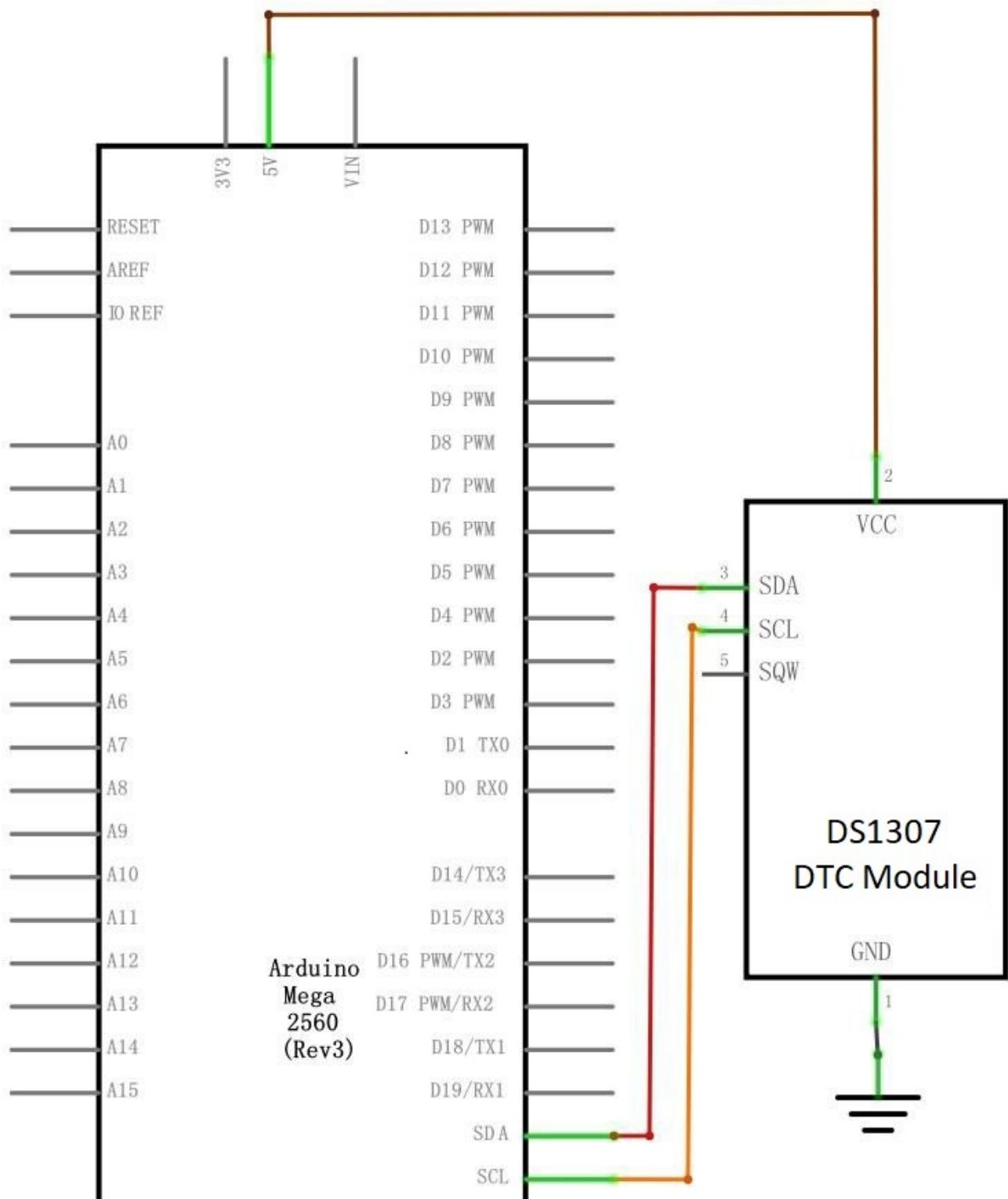
Der DS3231 ist ein einfacher Time-Keeping-Chip. Es hat eine integrierte Batterie, so dass die Uhr die Zeit nimmt, auch wenn er nicht angeschlossen ist.

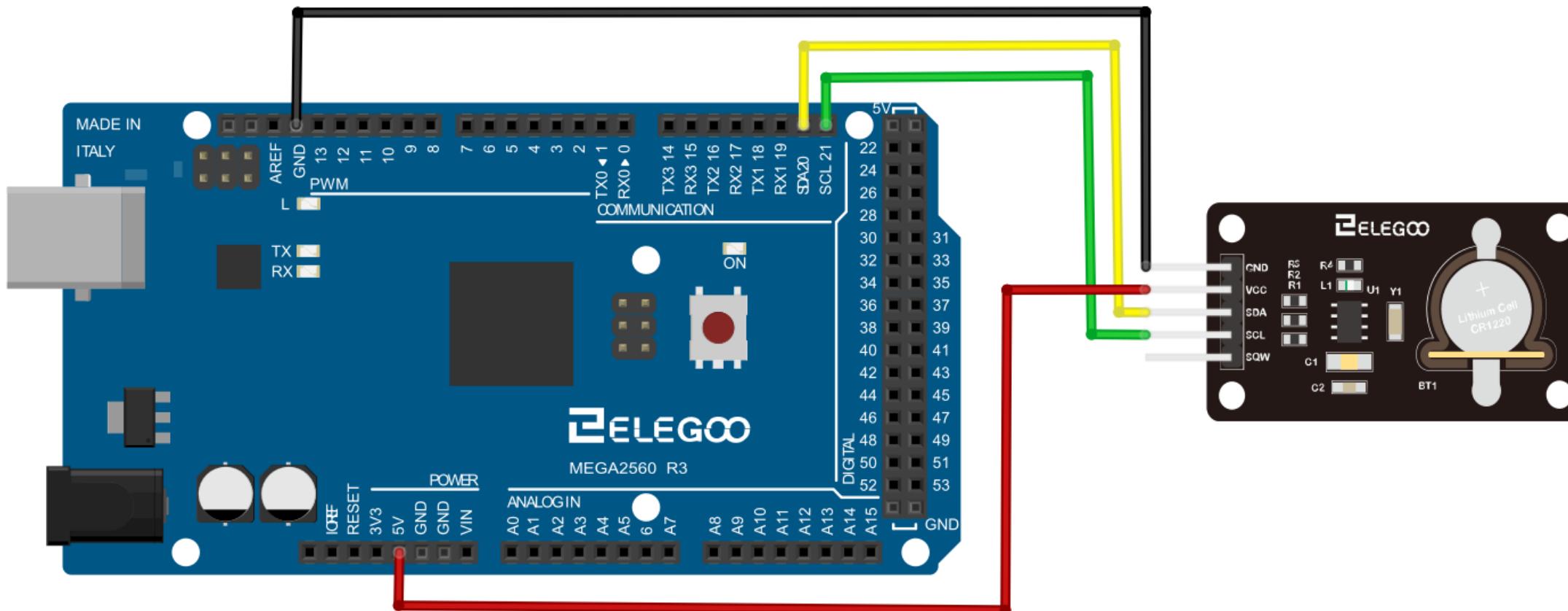


NOTICE: If there is the left version of module in the Kit, don't worry, its function and pin names are the same as the new version. Just follow the wiring diagrams and sketch in the tutorial below to get it working.

Verbindung

Schema





Schließen Sie es nach folgendem Bild an.

Ignorieren Sie die 32K- und SQW-Pins; Sie brauchen sie nicht. Stecken Sie den SCL-Pin in den SCL-Anschluss des MEGA2560 R3 und den SDA-Pin in den SDA-Anschluss.

Der VCC-Pin steckt in den 5V-Port und der GND-Stecker in den GND-Port.

Code

Nach der Verschalten öffnen Sie bitte das Programm im Codeordner - Lektion 19 Echtzeituhrmodul und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

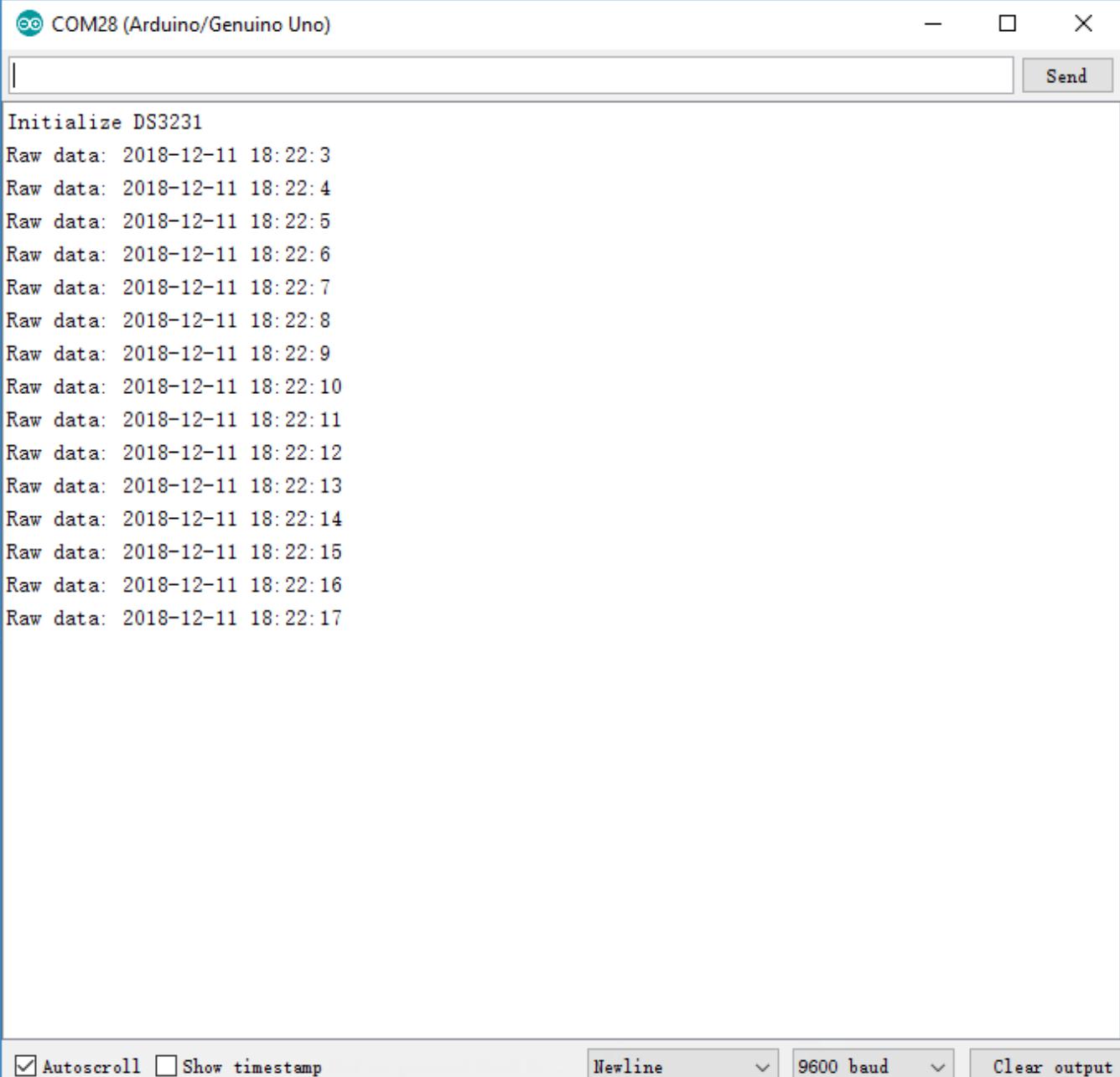
Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die Bibliothek <DS3231> installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion1.

Beispelfoto

Öffnen Sie den seriellen Monitor, dann können Sie sehen, das Modul kann die Zeit, wie unten zu sehen, anzeigen:

Klicken Sie auf die Schaltfläche Serielle Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors werden in Einzelheiten in Lektion 1 eingefügt.



The screenshot shows the Arduino Serial Monitor window titled "COM28 (Arduino/Genuino Uno)". The window displays a series of timestamp outputs from the DS3231 module. The output starts with "Initialize DS3231" followed by 16 lines of "Raw data: 2018-12-11 18:22:XX" where XX ranges from 3 to 17. At the bottom of the monitor, there are three configuration buttons: "Autoscroll" (checked), "Show timestamp" (unchecked), "Newline" (dropdown menu), "9600 baud" (dropdown menu), and "Clear output".

```
Initialize DS3231
Raw data: 2018-12-11 18:22:3
Raw data: 2018-12-11 18:22:4
Raw data: 2018-12-11 18:22:5
Raw data: 2018-12-11 18:22:6
Raw data: 2018-12-11 18:22:7
Raw data: 2018-12-11 18:22:8
Raw data: 2018-12-11 18:22:9
Raw data: 2018-12-11 18:22:10
Raw data: 2018-12-11 18:22:11
Raw data: 2018-12-11 18:22:12
Raw data: 2018-12-11 18:22:13
Raw data: 2018-12-11 18:22:14
Raw data: 2018-12-11 18:22:15
Raw data: 2018-12-11 18:22:16
Raw data: 2018-12-11 18:22:17
```

Lektion 20 Schallsensor-Modul

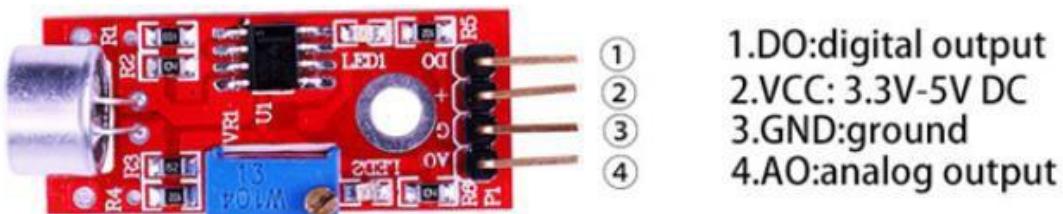
Überblick

In dieser Lektion lernen Sie, wie man ein Schallsensormodul benutzt. Dieses Modul hat zwei Ausgänge:

AO: Analogausgang, Echtzeit-Ausgangsspannungssignal des Mikrofons

DO: Wenn die Intensität des Tons eine bestimmte Schwelle erreicht, ist der Ausgang mit einem Signal mit „HIGH“ oder „LOW“ Pegel gesetzt. Die Schwellenempfindlichkeit kann durch Einstellen des Potentiometers erreicht werden.

Um sicherzustellen, dass das Mikrofon Ihre Stimme normal erkennen kann, versuchen Sie bitte seine Empfindlichkeit zu ändern, indem Sie das blaue präzise Potentiometer auf dem Modul drehen. Angesichts seiner Präzision, dauert es mindestens 10 Zyklen, um eine Reaktion zu sehen.



Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x Schallsensormodul
- (4) x F-M Kabel (Female to Male DuPont Kabel)

Komponenten Einführung

Mikrofon

Transducer (oder auch Kontakttonabnehmer) sind Geräte, die Energie von einer Form in eine andere umwandeln. Ein Mikrofon ist ein Wandler, der Schallenergie in elektrische Signale umwandelt. Es funktioniert genau andersherum als ein Lautsprecher. Mikrofone sind in verschiedenen Formen und Größen erhältlich. Abhängig von der Anwendung kann ein Mikrofon verschiedene Technologien verwenden um Ton in elektrische Signale umzuwandeln. Hier werden wir über das Elektrische-Kondensator-Mikrofon diskutieren, das in Mobiltelefonen, Laptops usw.

weit verbreitet ist.

Wie der Name schon sagt, ist das Elektrische-Kondensatormikrofon ein Parallelplatten-Kondensator und arbeitet nach dem Prinzip einer variablen Kapazität. Es besteht aus zwei Platten, eine feste (genannt die hintere Platte) und die andere ist beweglich (auch Zwerchfell genannt) und es ist eine kleine Lücke zwischen ihnen. Ein elektrisches Potential lädt die Platten auf. Wenn der Schall auf die Membran auftrifft, beginnt sie an, sich zu bewegen, wodurch die Kapazität zwischen den Platten verändert wird, was wiederum zu einem variablen elektrischen Strom führt der fließt.



Diese Mikrofone sind in elektronischen Schaltkreisen weit verbreitet, um kleinere Töne oder Luftschnüsse zu detektieren, die wiederum zur weiteren Verwendung in elektrische Signale umgewandelt werden. Die beiden „Beine“, wie in der Abbildung oben gezeigt, werden verwendet um eine elektrische Verbindung mit der Schaltung herzustellen.

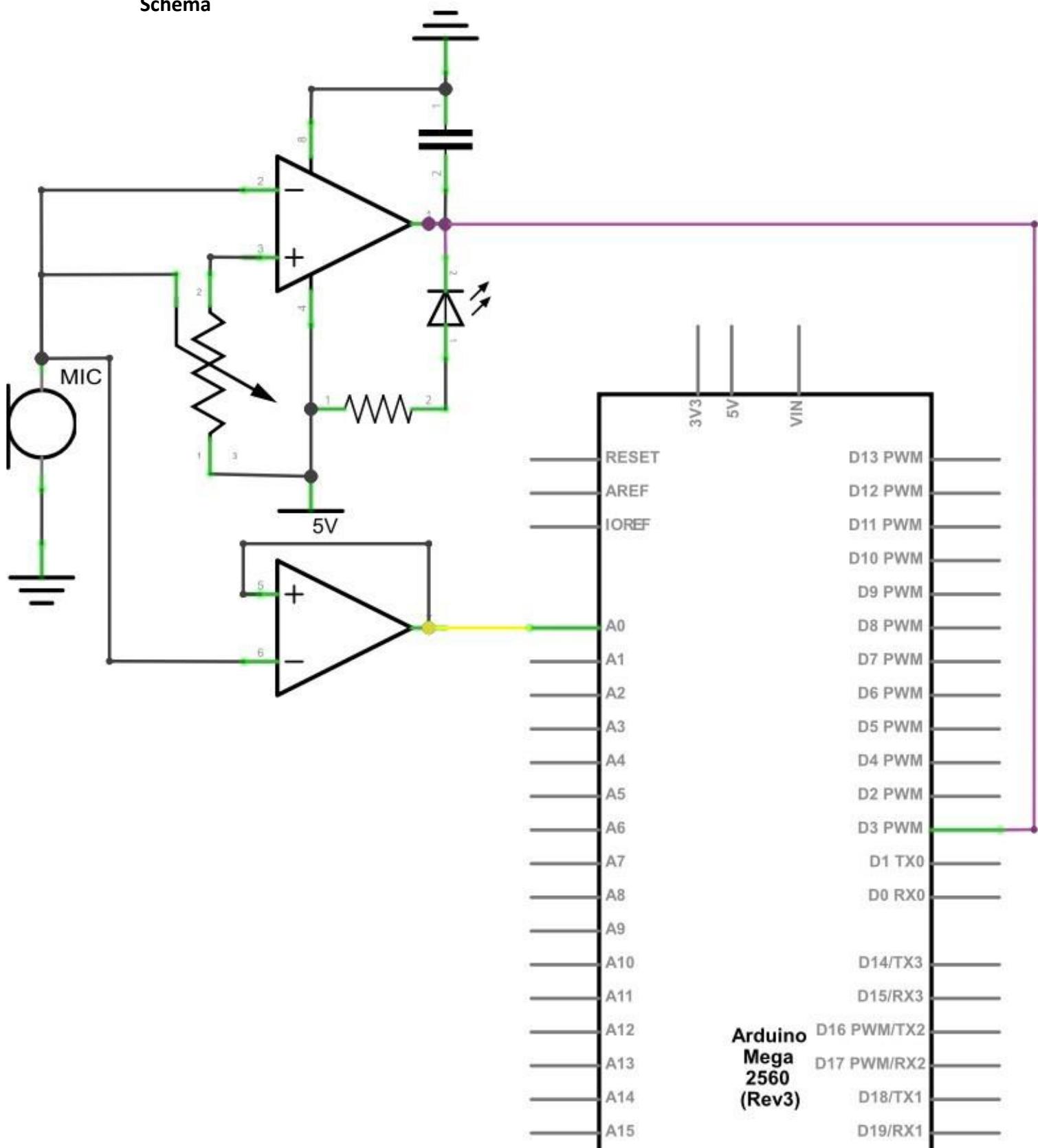


Ein fester leitfähiger Metallkörper verkapselt die verschiedenen Teile des Mikrofons. Die Oberseite ist mit einem porösen Material mit Hilfe von Leim bedeckt. Es wirkt als

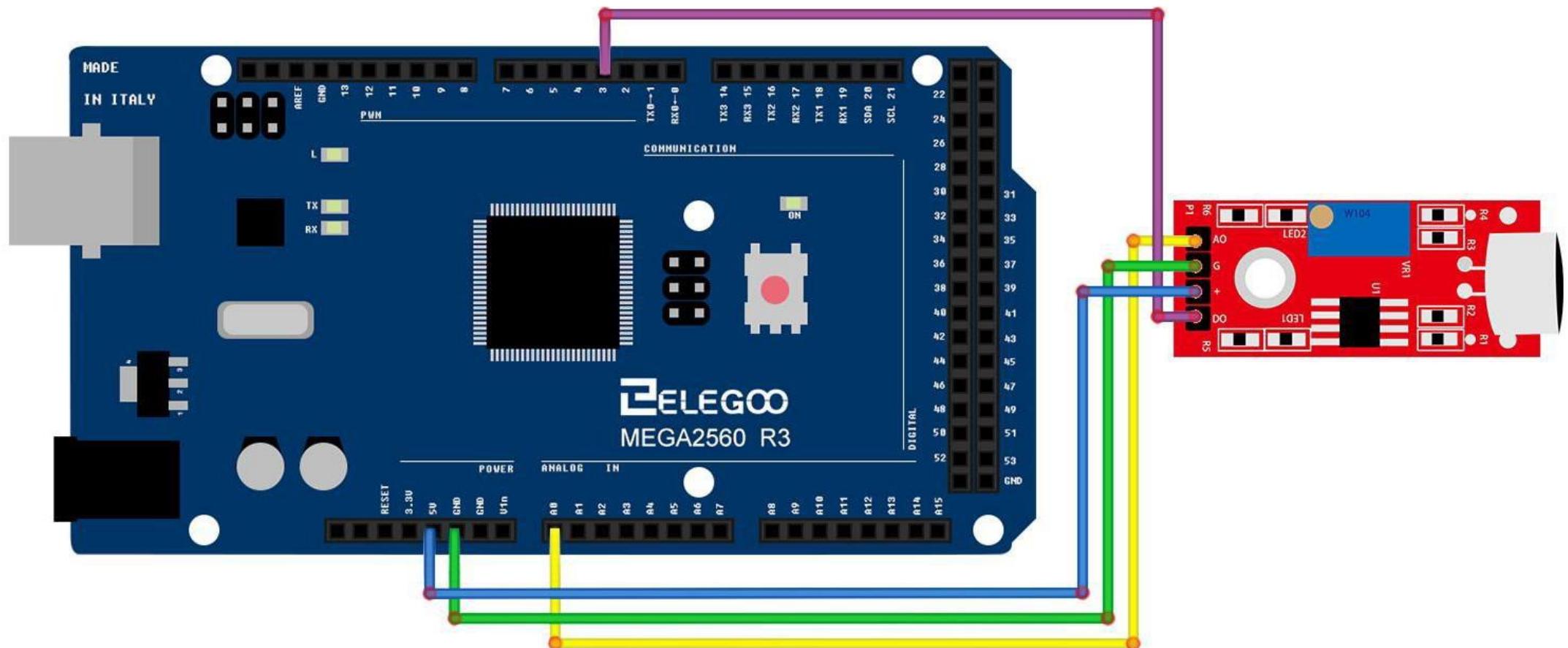
Filter für Staubpartikel. Die Schallsignale / Luftschwingungen gelangen durch das poröse Material und fallen dann auf das „Zwerchfell“ durch das im obigen Bild gezeigte Loch.

Verbindung

Schema



Schaltplan

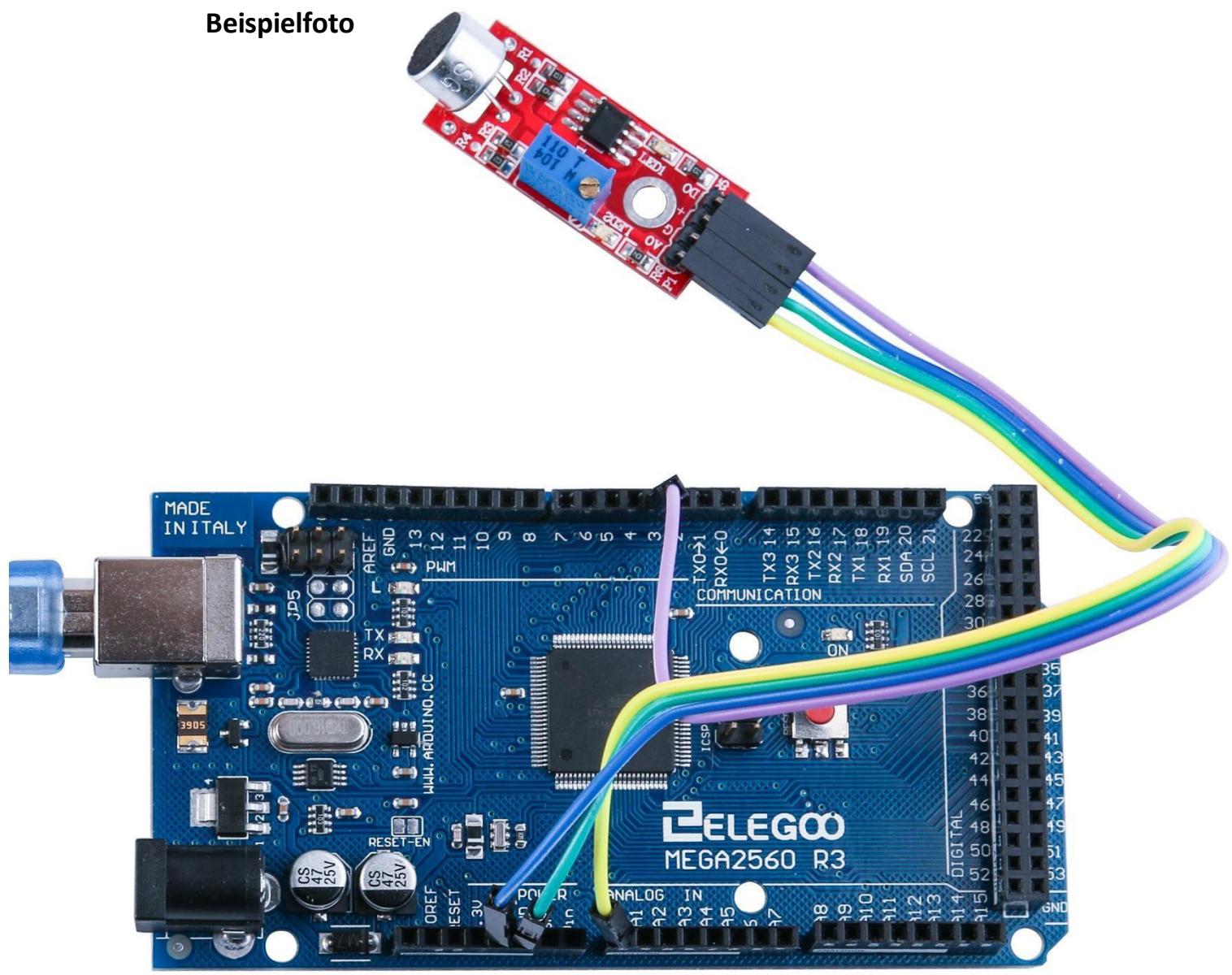


Der code

Nach der Verschaltung öffnen Sie bitte das Programm im Codeordner - Lektion 20 Schallsensormodul und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Dieses Modul bietet zwei Signalausgabemodi, für die wir zwei Codes geschrieben haben: `digital_signal_output` und `analog_signal_output`. Der Code von `digital_signal_output` funktioniert folgendermaßen: wenn die Stimme oder ein Geräusch einen bestimmten Wert erreicht, wird es ein digitales Signal auslösen und der dig # 11 Pin auf Arduino wird einen „HIGH“ Pegel ausgeben und die Anzeige L (haben wir in der Lektion „blitzen“ kennengelernt) wird gleichzeitig leuchten. Dieser Auslösewert kann entsprechend der oben erwähnten Empfindlichkeitseinstellungsmethode geändert werden. Der Code von `analog_signal_output` liest den analogen Wert des Moduls und zeigt ihn direkt auf dem seriellen Monitor an, ebenso kann dieser Wert auch entsprechend der oben erwähnten Empfindlichkeitseinstellung geändert werden.

Beispelfoto



Öffnen Sie den Monitor, dann können Sie die Daten wie unten sehen:

Klicken Sie auf die Schaltfläche Serial Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors werden in Lektion 1 ausführlich vorgestellt.

```
52
51
52
52
51
52
52
50
48
48
50
51
53
124
50
54
52
46
49
51
52
53
52
52
52
54
```

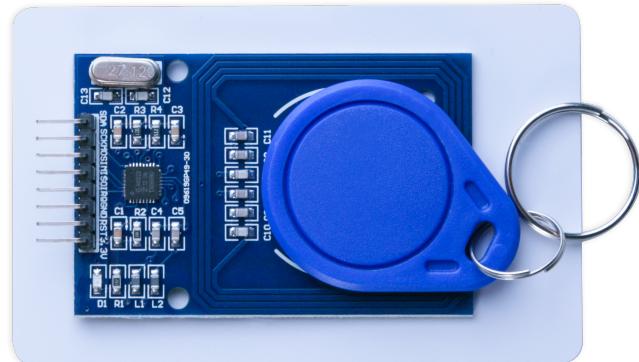
Lektion 21 RC522 RFID Modul

Überblick

In dieser Lektion erfahren Sie, wie Sie das RC522 RFID Lesermodul auf MEGA2560 R3 anwenden können. Dieses Modul verwendet den Serial Peripheral Interface (SPI) Bus, um mit Controllern wie Arduino, Raspberry Pi, Beagle Board etc. zu kommunizieren.

Component Required:

- (1) x Elegoo Mega2560 R3
- (1) x RC522 RFID Modul
- (7) x F-M Kabel (Female to Male DuPont Kabel)



Komponenten Einführung

RC522

Der MFRC522 ist ein hochintegrierter Leser/Schreiber für berührungslose Kommunikation bei 13,56 MHz. Der MFRC522 Leser unterstützt den ISO 14443A / MIFARE® Modus.

Das interne Senderteil des MFRC522 ist in der Lage, eine Leser / Schreiber-Antenne zu betreiben, die für die Kommunikation mit ISO / IEC 14443A / MIFARE® Karten und Transpondern ohne zusätzliche aktive Schaltungen ausgelegt ist. Der Empfängerteil sorgt für eine robuste und effiziente Implementierung einer Demodulations- und Decodierschaltung für Signale von ISO / IEC 14443A / MIFARE® kompatiblen Karten und Transpondern. Der digitale Teil übernimmt die komplette ISO / IEC 14443A Rahmen- und Fehlererkennung (Parity & CRC). Der MFRC522 unterstützt MIFARE® Classic (z. B. MIFARE® Standard) Produkte. Der MFRC522 unterstützt die kontaktlose Kommunikation mit MIFARE® höheren Übertragungsgeschwindigkeiten bis zu 848 kbit / s in beide Richtungen.

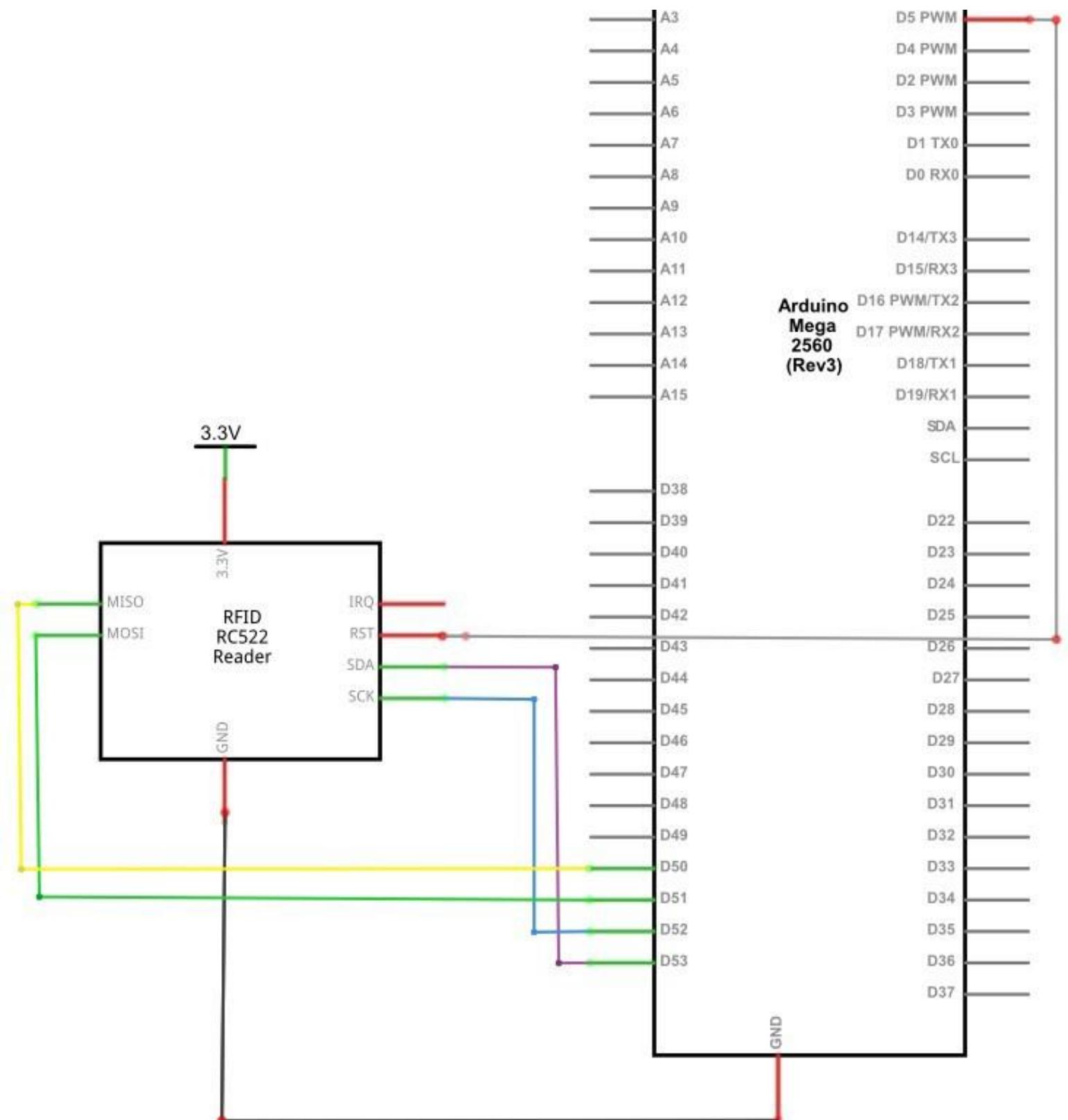
Verschiedene Host-Schnittstellen sind implementiert:

- SPI Schnittstelle
- Serial UART (ähnlich wie RS232 mit Spannungsniveaus bei Pad-Spannungsversorgung)
- I2C Schnittstelle

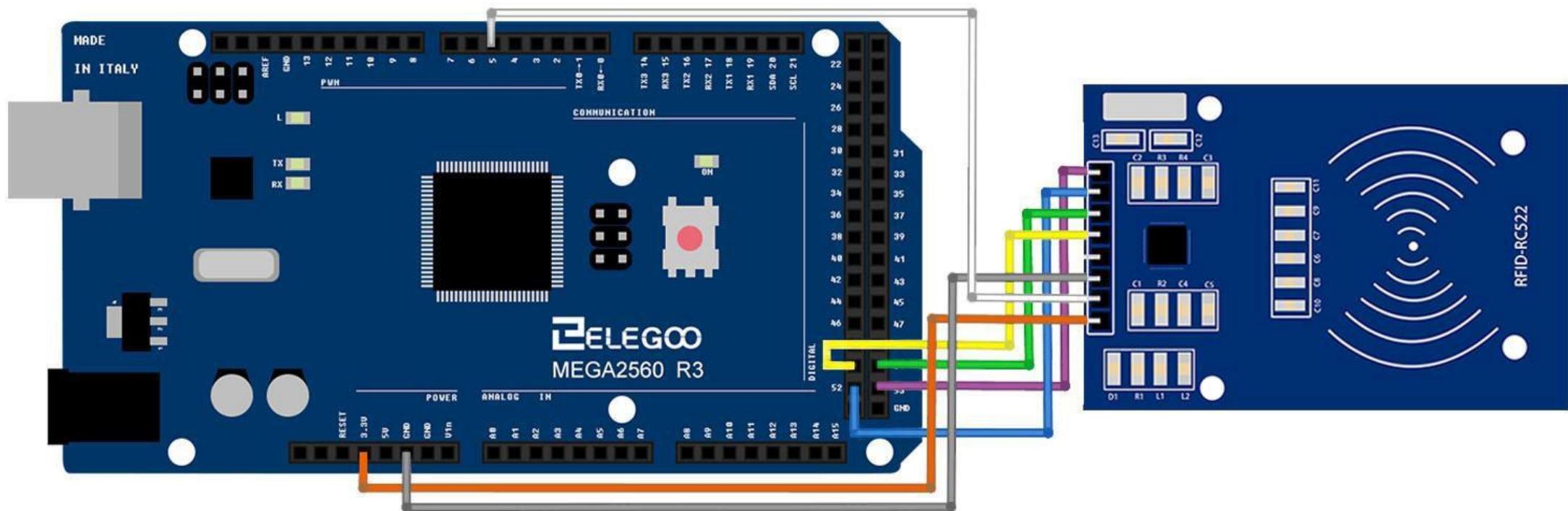
Die folgende Abbildung zeigt ein typisches Schaltbild unter Verwendung einer komplementären Antennenverbindung zum MFRC522.

Verbindung

Schema



Schaltplan



Code

Nach der Verschaltungung öffnen Sie bitte das Programm im Code-Ordner - Lektion 21 RC522 RFID-Modul und drücken Sie UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die <rfid> -Bibliothek installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion1.

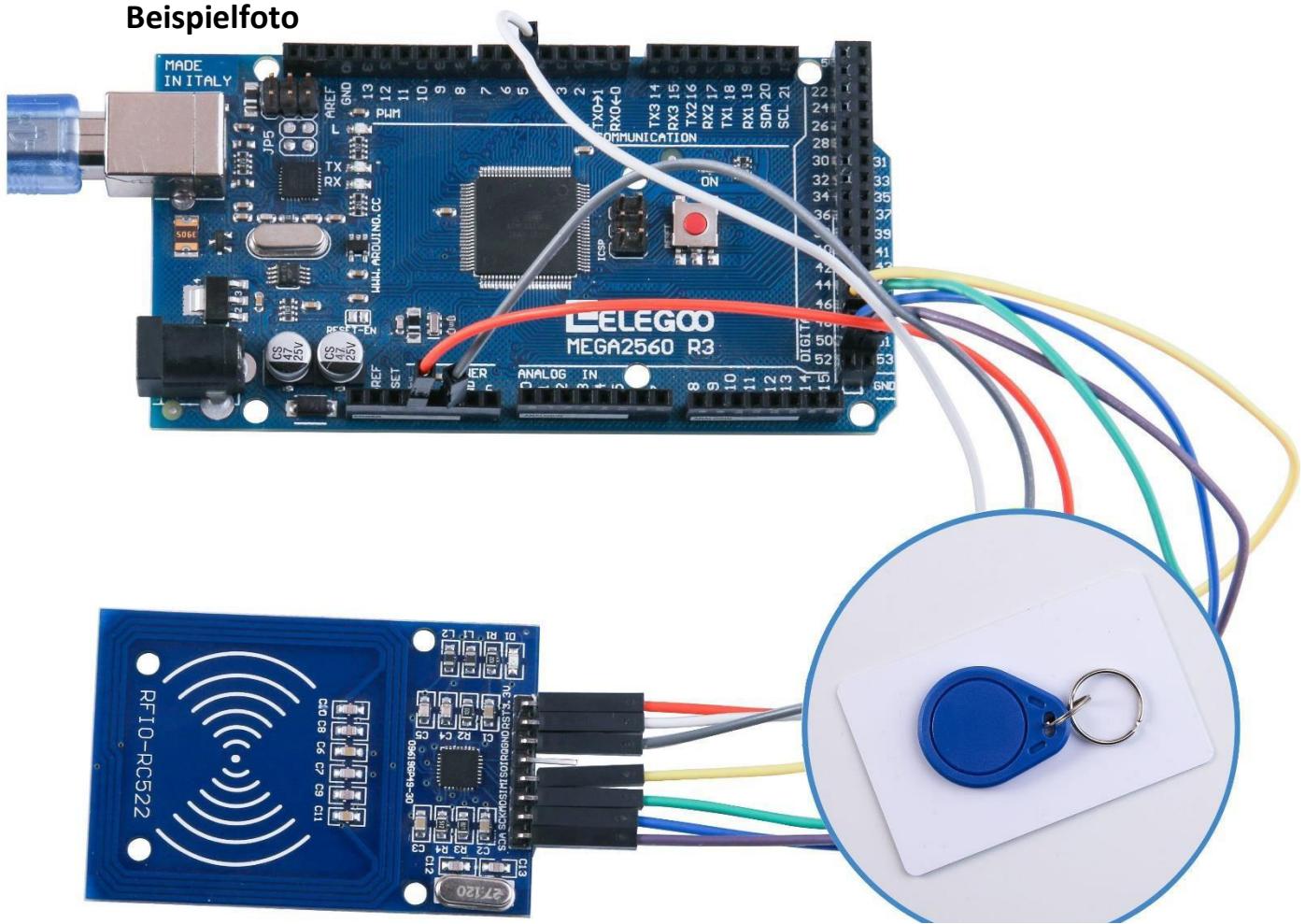
| Signal | MFRC522 Reader/PCD Pin | Arduino Uno Pin | Arduino Mega Pin | Arduino Nano v3 Pin | Arduino Leonardo/Micro Pin | Arduino Pro Micro Pin |
|-----------|------------------------|-----------------|------------------|---------------------|----------------------------|-----------------------|
| RST/Reset | RST | 9 | 5 | D9 | RESET/ICSP-5 | RST |
| SPI SS | SDA (SS) | 10 | 53 | D10 | 10 | 10 |
| SPI MOSI | MOSI | 11 / ICSP-4 | 51 | D11 | ICSP-4 | 16 |
| SPI MISO | MISO | 12 / ICSP-1 | 50 | D12 | ICSP-1 | 14 |
| SPI SCK | SCK | 13 / ICSP-3 | 52 | D13 | ICSP-3 | 15 |

```
#define RST_PIN      5      // Konfigurierbar, siehe typisches Pin-Layout oben
```

```
#define SS_PIN       53     // Konfigurierbar, siehe typisches Pin-Layout oben
```

Die Orte der SPI-Pin variieren bei verschiedenen Chips, weshalb Sie manchmal eine kleine Änderung des Codes vornehmen müssen.

Beispielfoto



Öffnen Sie den Monitor, dann können Sie die Daten wie unten sehen:

Klicken Sie auf die Schaltfläche Serial Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors sind in Lektion 1 ausführlich dargestellt.

The screenshot shows the Arduino Serial Monitor window titled "COM6 (Arduino/Genuino Mega or Mega 2560)". The window displays the following text output:

```
Warning: this example overwrites the UID of your UID changeable card, use with care
Card UID: 66 E3 F4 74
Card did not respond to 0x40 after HALT command. Are you sure it is a UID changeable?
Error name: Timeout in communication.
Activating the UID backdoor failed.
New UID and contents:
Card UID: 66 E3 F4 74
Card SAK: 08
PICC type: MIFARE 1KB
Sector Block  0  1  2  3  4  5  6  7  8  9  10 11  12 13 14 15  AccessBits
  15    63  00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
                62  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
                61  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
                60  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
  14    59  00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
                58  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
                57  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
                56  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
  13    55  00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
                54  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
                53  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
                52  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
  12    51  00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
                50  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
                49  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
                48  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
  11    47  00 00 00 00  00 00 FF 07  80 69 FF FF  FF FF FF FF  [ 0 0 1 ]
                46  00 00 00 00  00 00 00 00  00 00 00 00  00 00 00 00  [ 0 0 0 ]
```

At the bottom of the window, there are three buttons: "Autoscroll" (checked), "Newline", and "9600 baud".

Lektion 22 LCD Display

Überblick

In dieser Lektion lernen Sie, wie man ein alphanumerisches LCD-Display benutzt.

Das Display verfügt über eine LED-Hintergrundbeleuchtung und kann in jeder seiner zwei Zeilen bis zu 16 Zeichen darstellen. Sie können die Rechtecke für jedes Zeichen auf dem Display und die Pixel sehen, die jedes Zeichen bilden. Die Anzeige ist nur weiß auf blau und dient zum Anzeigen von Text.

In dieser Lektion werden wir das Arduino-Beispielprogramm für die LCD-Bibliothek ausführen, aber in der nächsten Lektion benutzen wir unser Display, um die Temperatur mit Sensoren anzuzeigen.

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x LCD1602 Modul
- (1) x Potentiometer (10k)
- (1) x 830 tie-points Breadboard
- (16) x M-M Kabel (Male to Male jumper Kabel)



Komponenten Einführung

LCD1602

Einführung in die Pins des LCD1602:

VSS: Ein Stift, der mit der Masse verbunden ist

VDD: Ein Pin, der mit einer + 5V Stromversorgung verbunden ist

VO: Ein Pin, der den Kontrast von LCD1602 anpasst

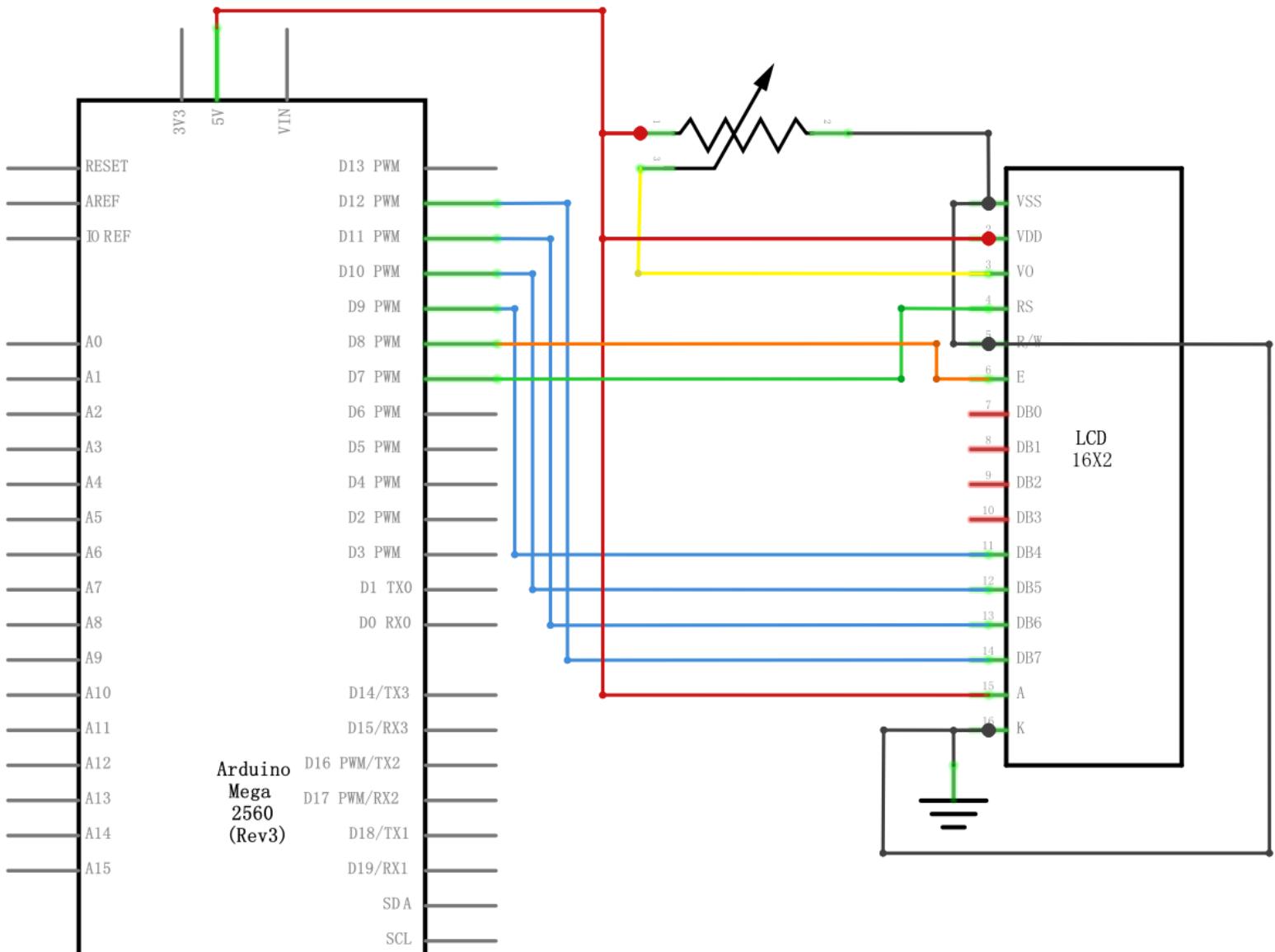
RS: Ein Registerauswahlstift, der steuert, wo Sie im LCD-Speicher Sie Daten schreiben. Sie können entweder das Datenregister auswählen, dies „speichert“, was auf dem Bildschirm angezeigt wird, oder ein Befehlsregister, wo der LCD-Controller nach Anweisungen sucht, was als nächstes zu tun ist.

R/W: Ein Lese- / Schreib-Pin, der den Lesemodus oder den Schreibmodus auswählt **E:** Ein Pin, der, wenn er mit Low-Level-Energie versorgt wird, das LCD-Modul dazu veranlasst, entsprechende Anweisungen auszuführen.

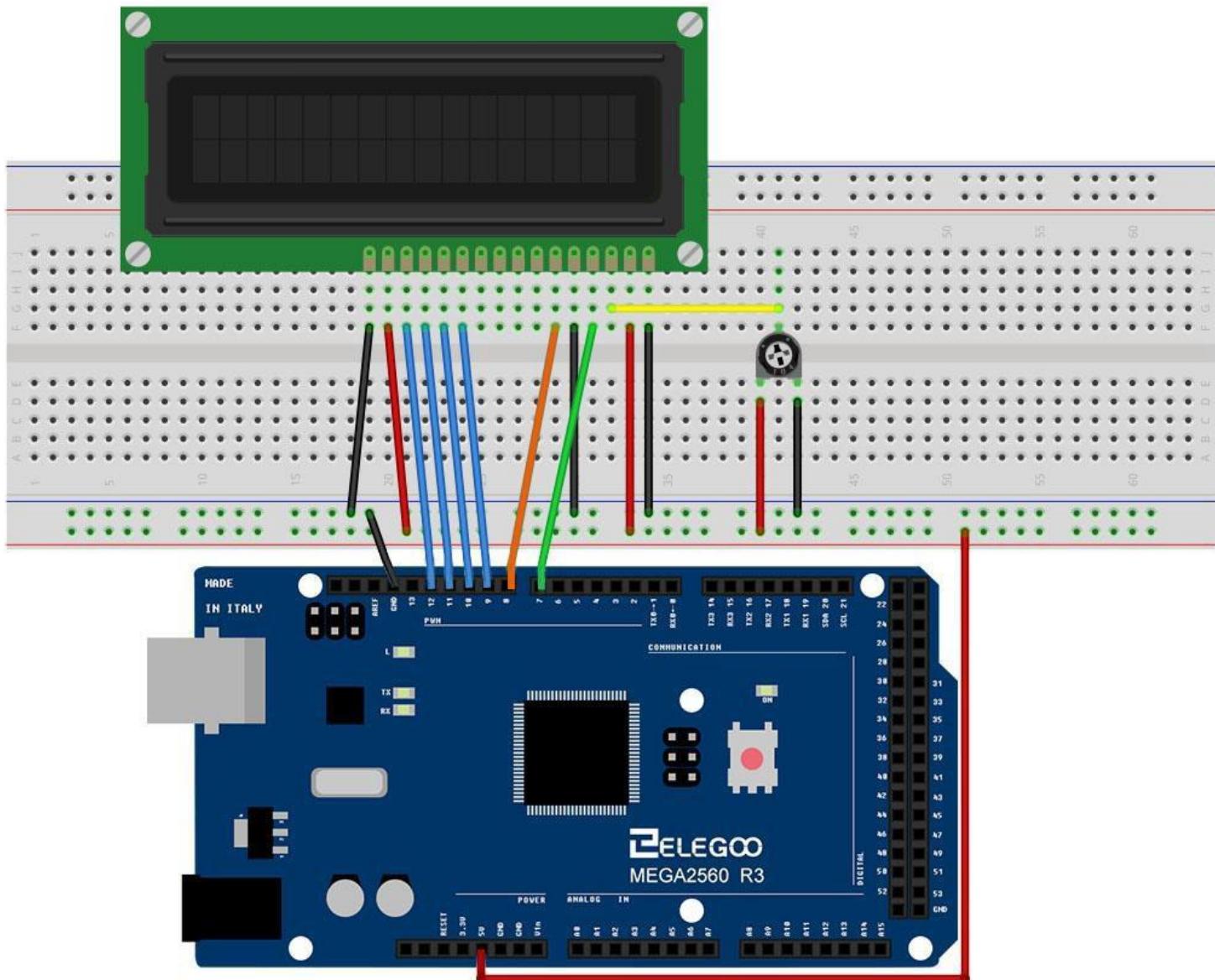
D0-D7: Pins, die Daten lesen und schreiben

A und K: Pins, die die LED-Hintergrundbeleuchtung steuern

Verbindung Schema



Schaltplan



Die LCD-Anzeige benötigt sechs Arduino-Pins, die alle digitale Ausgänge sind. Man braucht auch 5V und GND Verbindungen.

Man muss eine ganze Reihe an Verbindungen herstellen. Das Aufstecken der LCD-Anzeige mit der Oberseite des Steckbrettes hilft, seine Pins ohne zu viel Zählen zu identifizieren, besonders wenn das Breadboard seine Reihen mit Reihe 1 als die obere Reihe des Brettes nummeriert hat. Vergessen Sie nicht, das lange gelbe Blei, das den Schieber des Topfes mit dem Pin 3 des Displays verbindet. Mit dem 'Poti' wird der Kontrast des Displays gesteuert.

Vielleicht jönnen Sie feststellen, dass Ihre Anzeige ohne Stiftleiste geliefert wird.

Wenn dem so ist, folgen Sie bitte den Anweisungen im nächsten Abschnitt.

Code

Nach der Verschaltung öffnen Sie bitte das Programm im Codeordner - Lektion 22 LCD Display und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die Bibliothek <LiquidCrystal> installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion 1.

Laden Sie den Code auf Ihren Arduino-Board und Sie sollten die Nachricht 'hallo, Welt' angezeigt sehen, gefolgt von einer Zahl, die von Null aufwärtszählt.

Die erste Zeile in dem Sketcg ist die folgende:

`#include <LiquidCrystal.h>`

Dieses sagt dem Arduino, dass wir die Liquid Crystal Bibliothek benutzen wollen.

Als nächstes haben wir die Zeile, die wir ändern müssen/können. Diese legt fest, welche Pins des Arduino mit welchen Pins des Displays verbunden werden sollen.

`LiquidCrystal lcd(7, 8, 9, 10, 11, 12);`

Nachdem Sie diesen Code hochgeladen haben, stellen Sie sicher, dass die Hintergrundbeleuchtung leuchtet und stellen Sie das Potentiometer so weit ein, bis Sie die Textnachricht sehen können.

In der Funktion 'Setup' haben wir zwei Befehle:

`lcd.begin(16, 2);
lcd.print("Hello, World!");`

Der erste Zeile sagt der Liquid Crystal Bibliothek, wie viele Spalten und Zeilen das Display hat. Die zweite Zeile zeigt die Nachricht an, die wir in der ersten Zeile des

Bildschirms sehen.

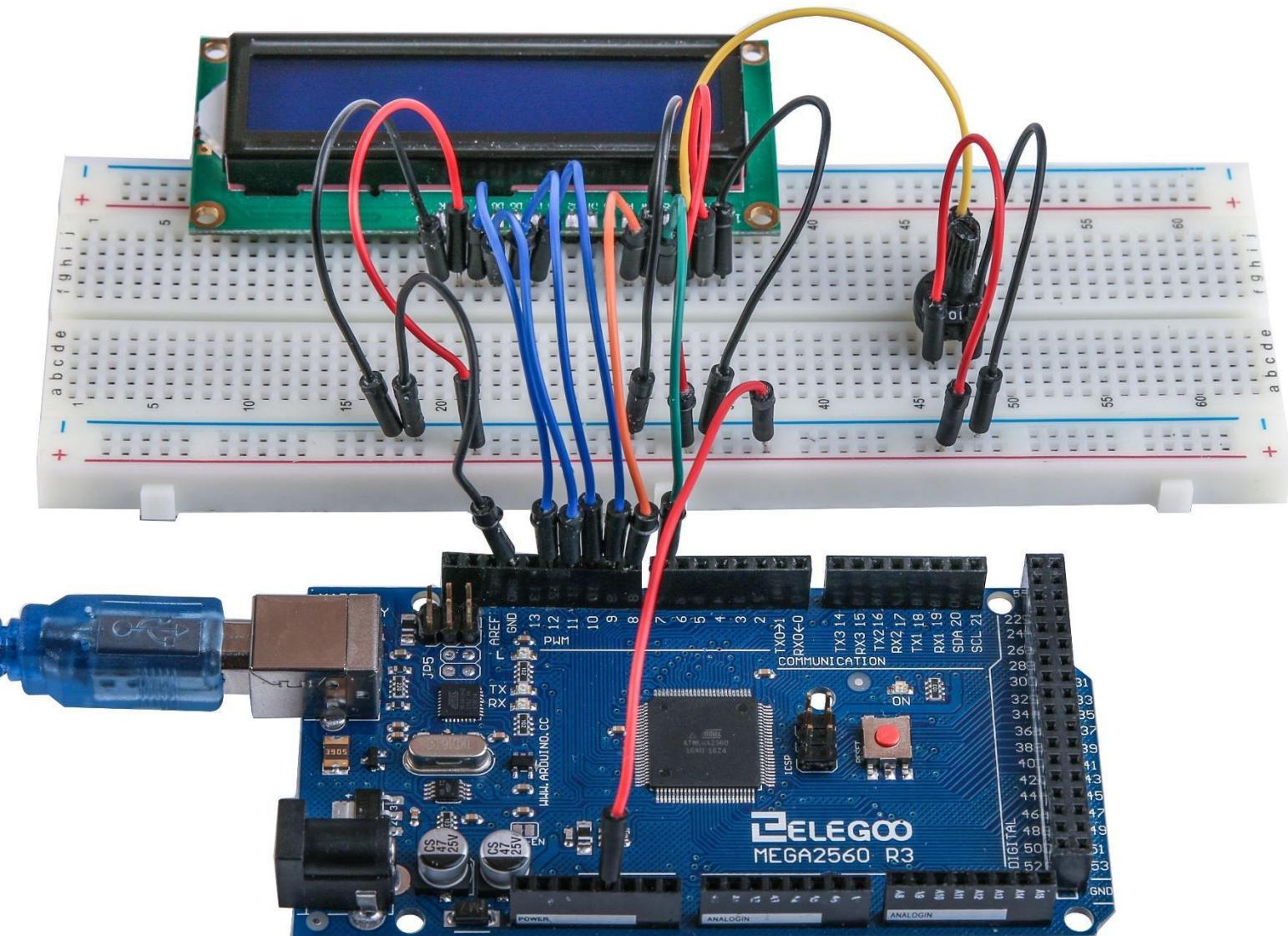
In der 'loop' Funktion haben wir auch zwei Befehle:

```
lcd.setCursor(0, 1);  
lcd.print(millis()/1000);
```

Der erste Zeile setzt die Cursorposition (wo der nächste Text erscheint) in Spalte 0 und Zeile 1. Sowohl Spalten- als auch Zeilenummern beginnen bei 0 anstatt bei 1.

Die zweite Zeile zeigt die Anzahl der Millisekunden an, seit der Arduino zurückgesetzt wurde.

Beispelfoto



Lektion 23 Thermometer

Überblick

In dieser Lektion wird ein LCD-Display zur Anzeige der Temperatur verwendet.

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x LCD1602 Modul
- (1) x 10k ohm Widerstand
- (1) x Thermistor
- (1) x Potentiometer
- (1) x 830 tie-points Breadboard
- (18) x M-M Kabel (Male to Male jumper Kabel)

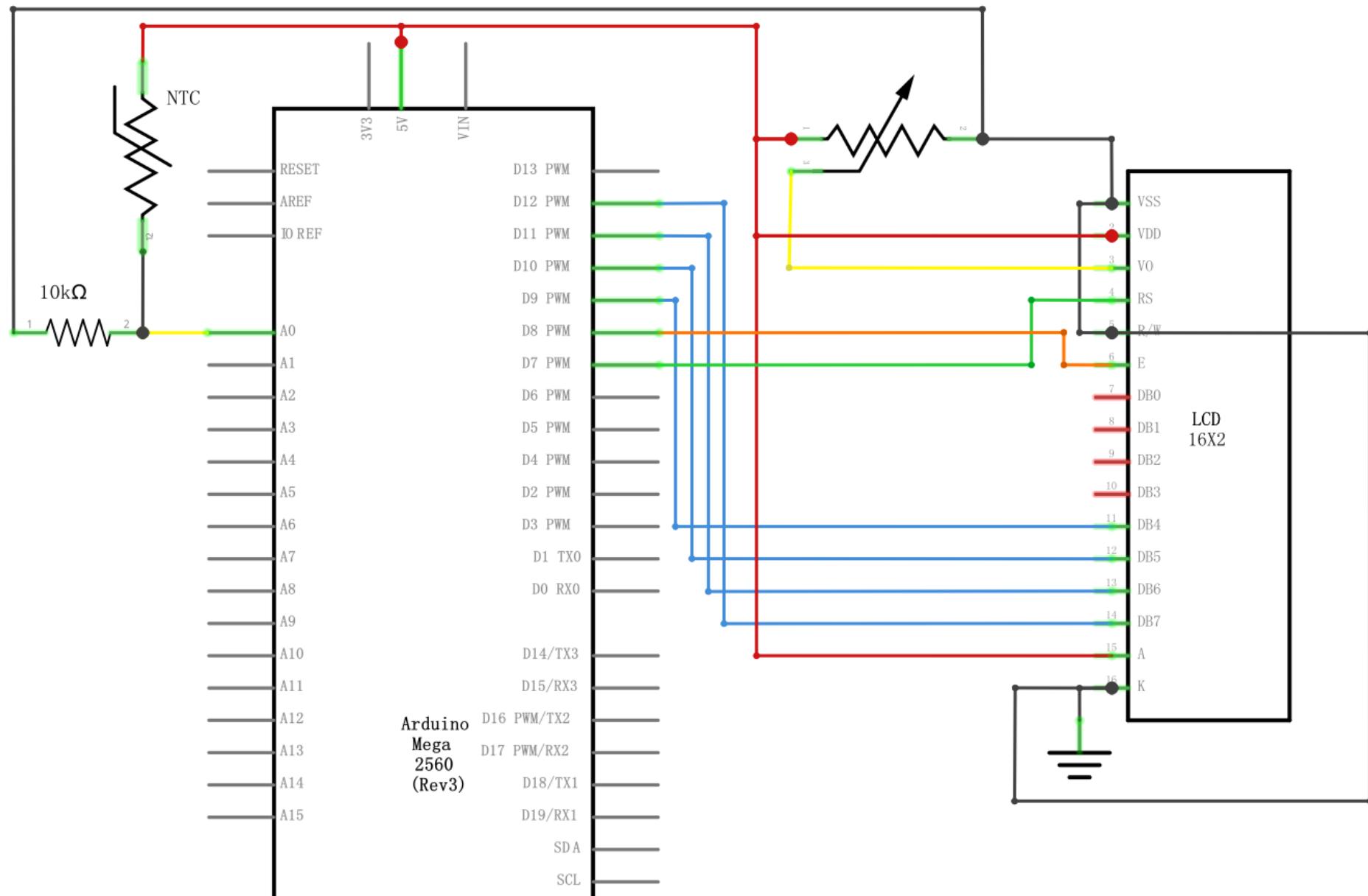
Komponenten Einführung

Thermistor

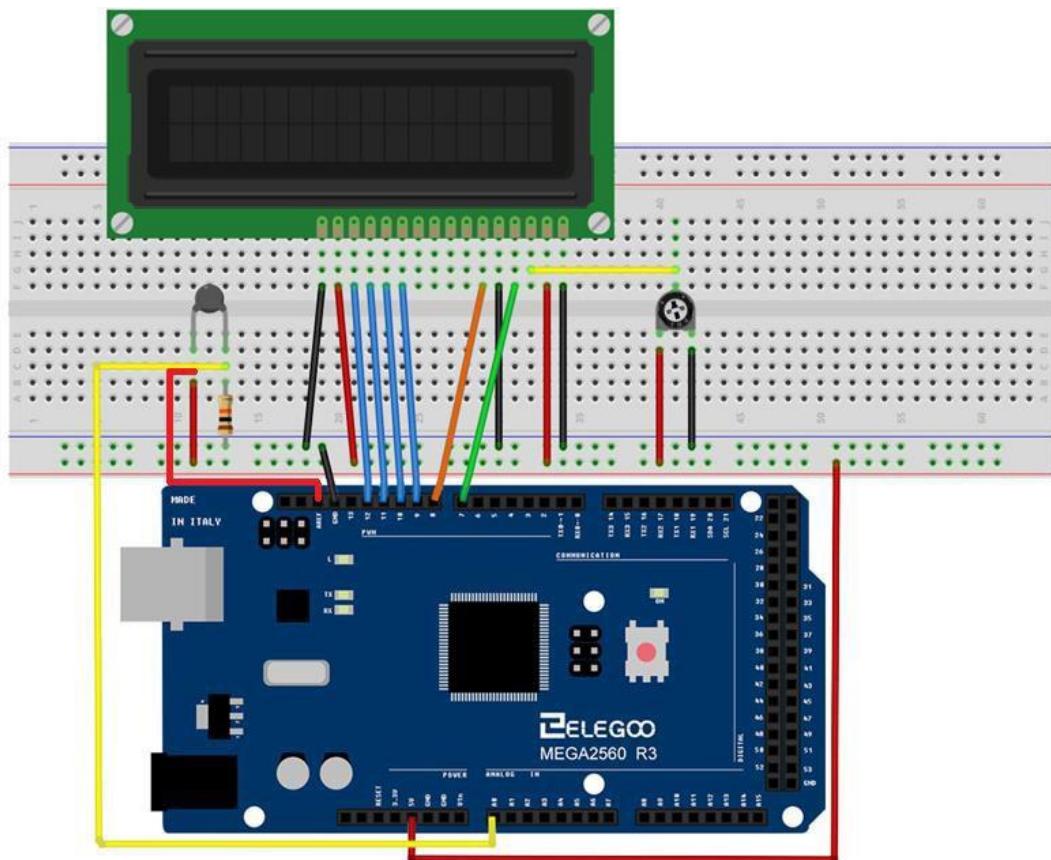
Ein Thermistor sind ein thermischer Widerstand - ein Widerstand, der seinen Widerstand mit der Temperatur ändert. Technisch sind alle Widerstände Thermistoren - ihr Widerstand ändert sich leicht mit der Temperatur - aber die Veränderung ist meist sehr klein und schwer zu messen. Thermistoren werden so hergestellt, dass sich der Widerstand drastisch mit der Temperatur ändert, so dass es 100 Ohm oder mehr Veränderung pro Grad sein können!

Es gibt zwei Arten von Thermoelementen, NTC (negativer Temperaturkoeffizient) und PTC (positiver Temperaturkoeffizient). Im Allgemeinen nutzen Sie NTC-Sensoren für die Temperaturmessung. PTC's werden oft als rücksetzbare Sicherungen verwendet. Sie ändern Ihren Widerstandswert in der Nähe einer kritischen Temperatur drastisch.

Verbindung Schema



Schaltdiagramm



Das Breadboard-Layout basiert auf dem Layout von Lektion 22, also ist das ganze noch etwas einfacher, wenn man das noch auf dem Steckbretthat.

Es gibt ein paar Jumper Drähte in der Nähe des Poti, die auf unserem Layout etwas verschoben wurden.

Der 10 kΩ Widerstand und der Thermistor sind alle Neuzugänge der Platine.

Code

Nach der Verschaltung öffnen Sie bitte das Programm im Codeordner - Lektion 23 Thermometer und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die Bibliothek <LiquidCrystal> installiert haben oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion 1.

Der Sketch dafür basiert auf der Lektion 22. Laden Sie es auf Ihr Arduino und Sie sollten feststellen, dass Sie die Erwärmung des Sensors, beispielsweise durch berühren mit der Hand, ablesen können.

Ich finde es sinnvoll, eine Kommentarzeile über dem Befehl 'lcd' zusetzen.

```
// BS  E  D4 D5  D6 D7
```

```
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Dies erleichter es Ihnen, wenn Sie sich entscheiden welche Pins Sie verwenden.

In der 'Loop'-Funktion gibt es jetzt zwei interessante Dinge zu beachten. Zuerst müssen wir das Analog Signal vom Temperatursensor in eine aktuelle Temperatur umwandeln und zweitens müssen wir herausfinden, wie man sie anzeigt.

Das Ändern von Messwerten auf einem LCD-Display kann schwierig sein. Das Hauptproblem ist, dass die Messwerte nicht immer die gleiche Anzahl von Ziffern hat. Wie zum Beispiel wenn sich die Temperatur von 101,50 auf 99,00 ändert, dann besteht die Gefahr, dass die alte „übrige“ Zahl einfach stehen bleibt.

Um dies zu vermeiden, schreibe die ganze Zeile des LCDs jedes Mal in der Schleife neu.

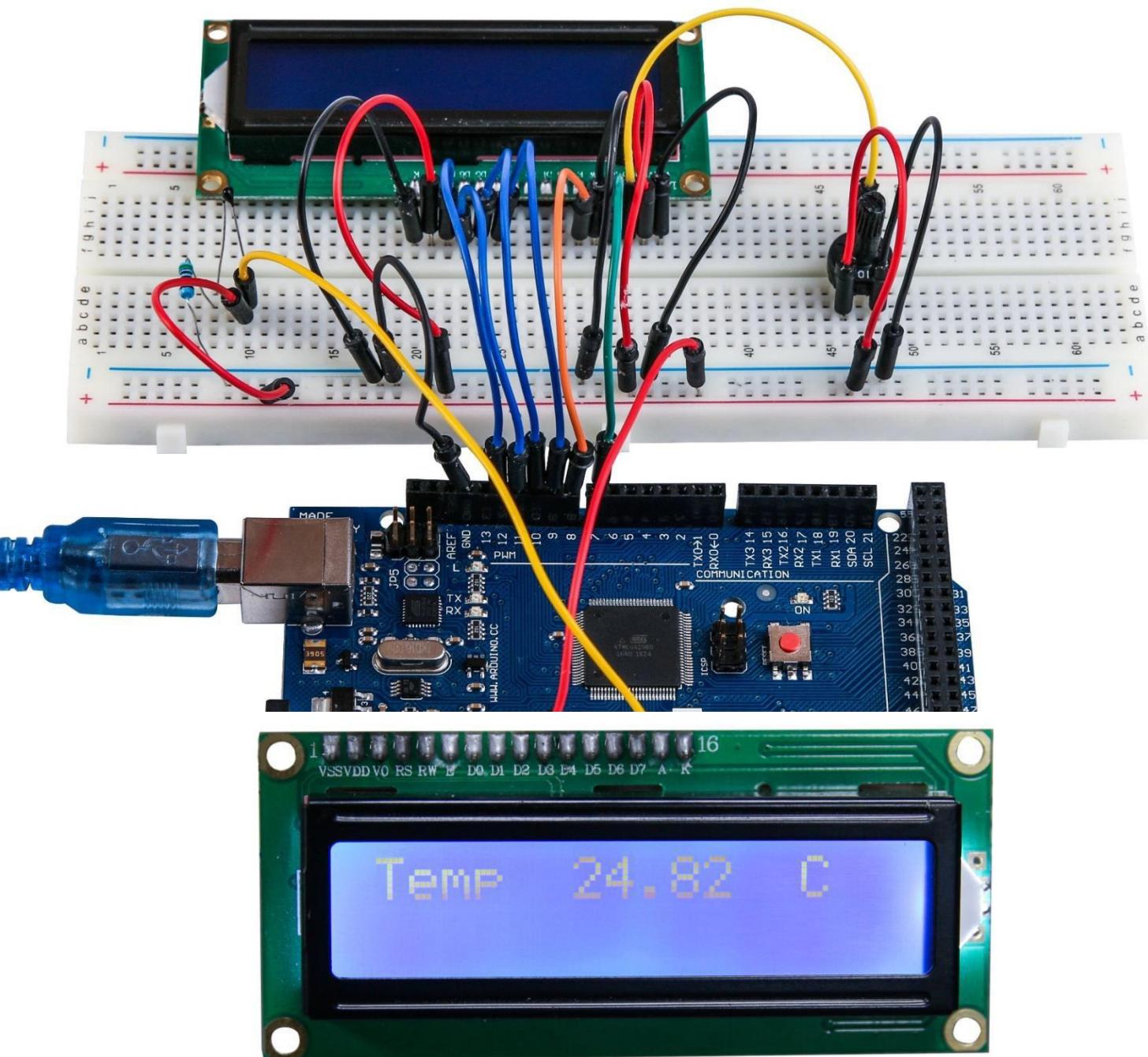
```
lcd.setCursor(0, 0);
lcd.print("Temp      C  ");
lcd.setCursor(6, 0);
lcd.print(temperatura);
```

Der seltsame Befehl dient dazu, Sie an die 16 Spalten des Displays zu erinnern. Sie

können dann einen String dieser Länge mit Leerzeichen anzeigen, in denen das eigentliche auslesen des Sensors stattfindet.

Um die Leerzeichen auszufüllen, stellen Sie die Cursorposition ein wo die ausgelesenen Daten erscheinen sollen.

Beispiefoto



Lektion 24 Acht LEDs mit 74HC595

Überblick

In dieser Lektion lernen Sie, wie man acht große rote LEDs mit einem MEGA2560 benutzt, ohne alle 8 Ausgabepins zu benutzen!

Obwohl man acht LEDs mit jeweils einem Widerstand mit einem MEGA2560 Pin verdrahten könnte, würde man schnell merken, dass man nicht mehr viele Pins für weitere Bauteile zur Verfügung hätte. Wenn Sie nicht viel Sachen mit Ihrem MEGA2560 verbinden möchten ist es OK, diese auf herkömmlich weise anzuschließen - aber oftmals wollen wir weitere Knöpfe, Sensoren, Servos, etc. anschließen und ehe man sich versieht, hat man keine Pins mehr. Also, anstatt das zu tun, werden wir einen Chip namens 74HC595 Serial to Parallel Converter verwenden. Dieser Chip hat acht Ausgänge (perfekt) und drei Eingänge die man bei Zeiten mit Daten füttern kann.

Dieser Chip die Schaltung insgesamt etwas langsamer, die LEDs werden allerdings immer noch schnell genug angesteuert (man kann die LEDs nur etwa 500.000 Mal pro Sekunde anstelle von 8.000.000 pro Sekunde ändern), schneller als der Mensch erkennen kann ob wir die LED aus oder anmachen 😊😊. Also lohnt es sich!

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x 830 tie-points breadboard
- (8) x LEDs
- (8) x 220 ohm Widerstände
- (1) x 74hc595 IC
- (14) x M-M Kabel (Male to Male jumper Kabel)



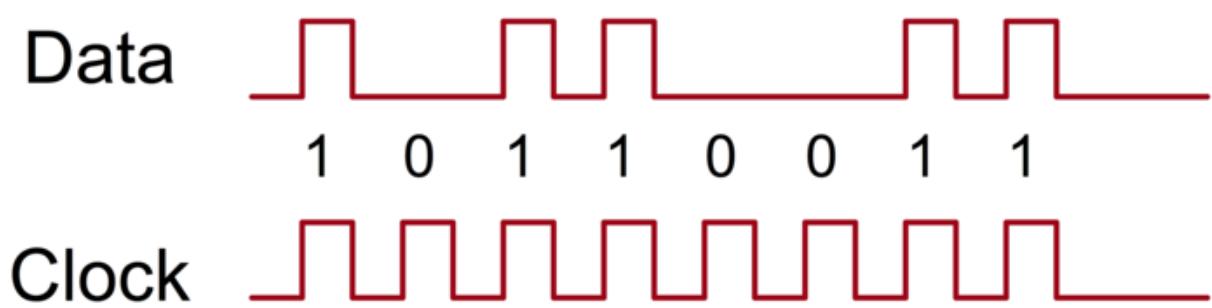
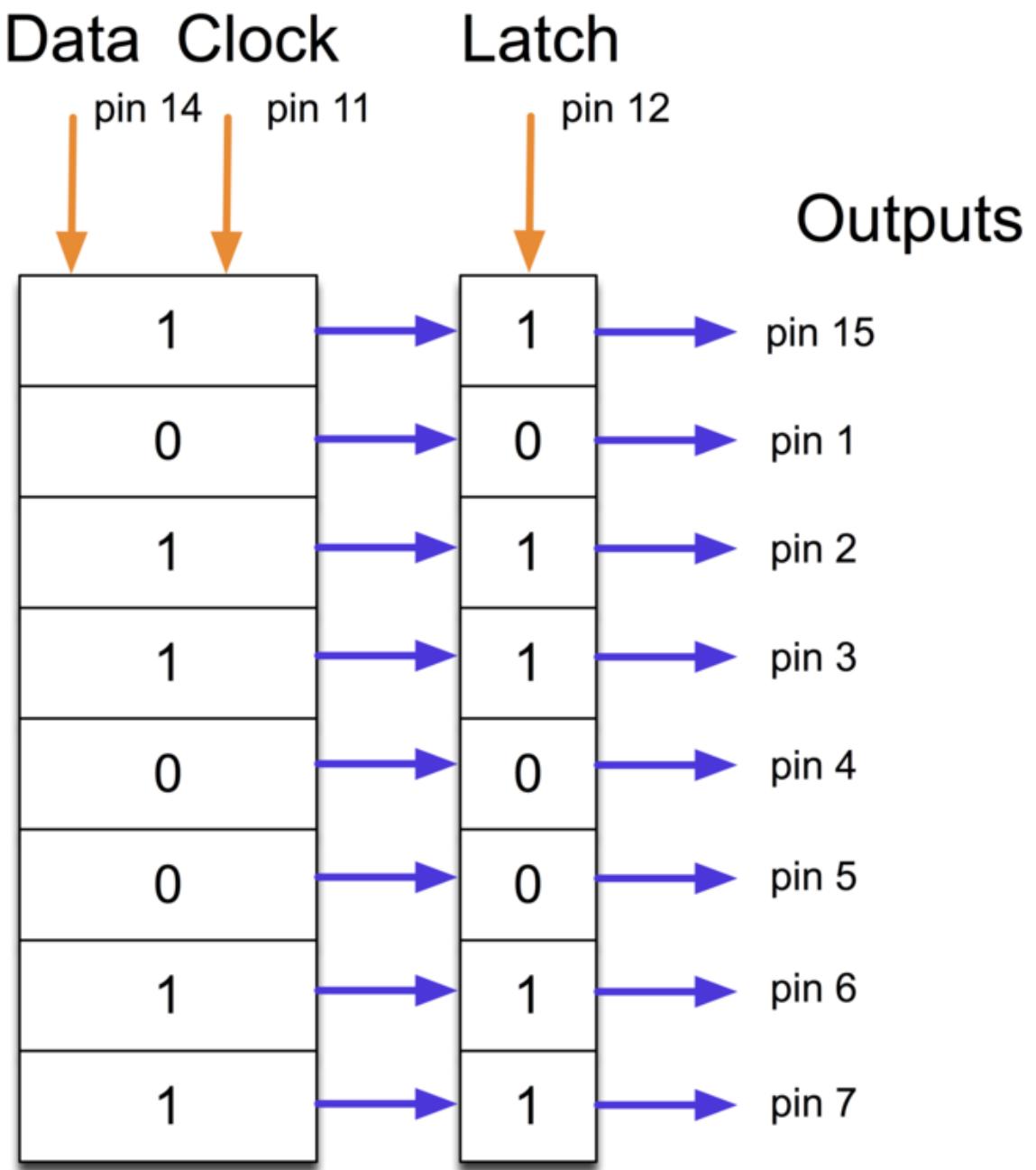
Komponenten Einführung

74HC595 Schieberegister:

Ein Schieberegister ist ein logisches Schaltwerk. Mehrere in Reihe geschaltete Flipflops schieben ihren Speicherinhalt (je 1 Bit) bei jedem Arbeitstakt um ein Flipflop weiter – anschaulich gesehen ähnlich einer Eimerkette. Die Anzahl der im Register vorhandenen Speicherplätze ist konstant.

Das Schieberegister ist eine Art Chip, das quasi aus acht Speicherplätzen besteht,

von denen jeder entweder ein 1 oder ein 0 sein kann. Um jeden dieser Werte einzulesen oder auszuschalten, geben wir die Daten mit den 'Data' und 'Clock' Pins in dem Chip ein.

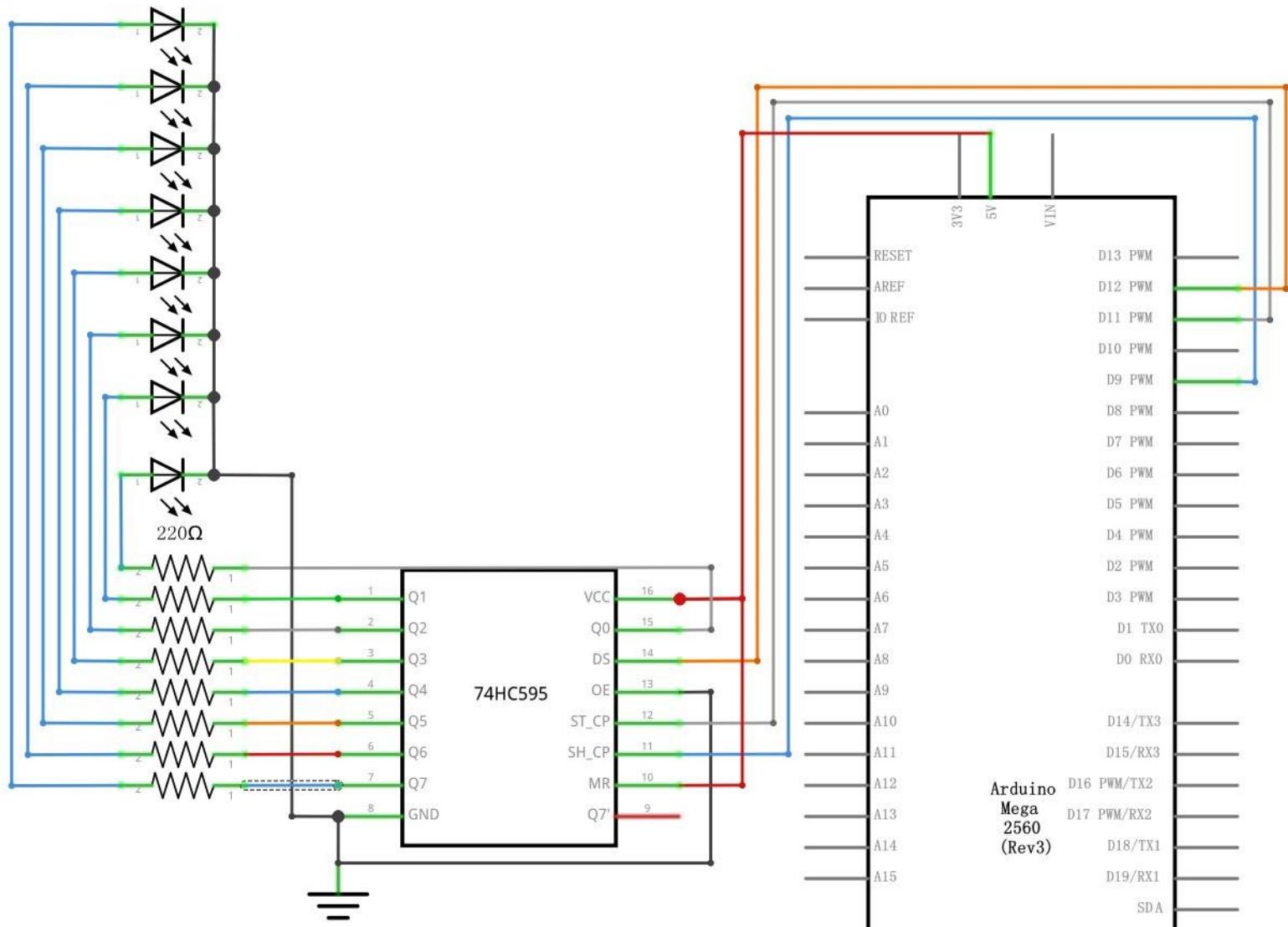


Die Takt-Pin muss acht Impulse empfangen. Bei jedem Puls, wenn der Datenpin „HIGH“ ist, wird eine 1 in das Schieberegister geschrieben. Andernfalls eine 0. Wenn alle acht Impulse empfangen worden sind, kopiert der 'Latch' Pin diese acht Werte in das Latchregister. Das ist notwendig! Andernfalls würden die falschen LEDs flackern, wenn die Daten in das Schieberegister geladen werden.

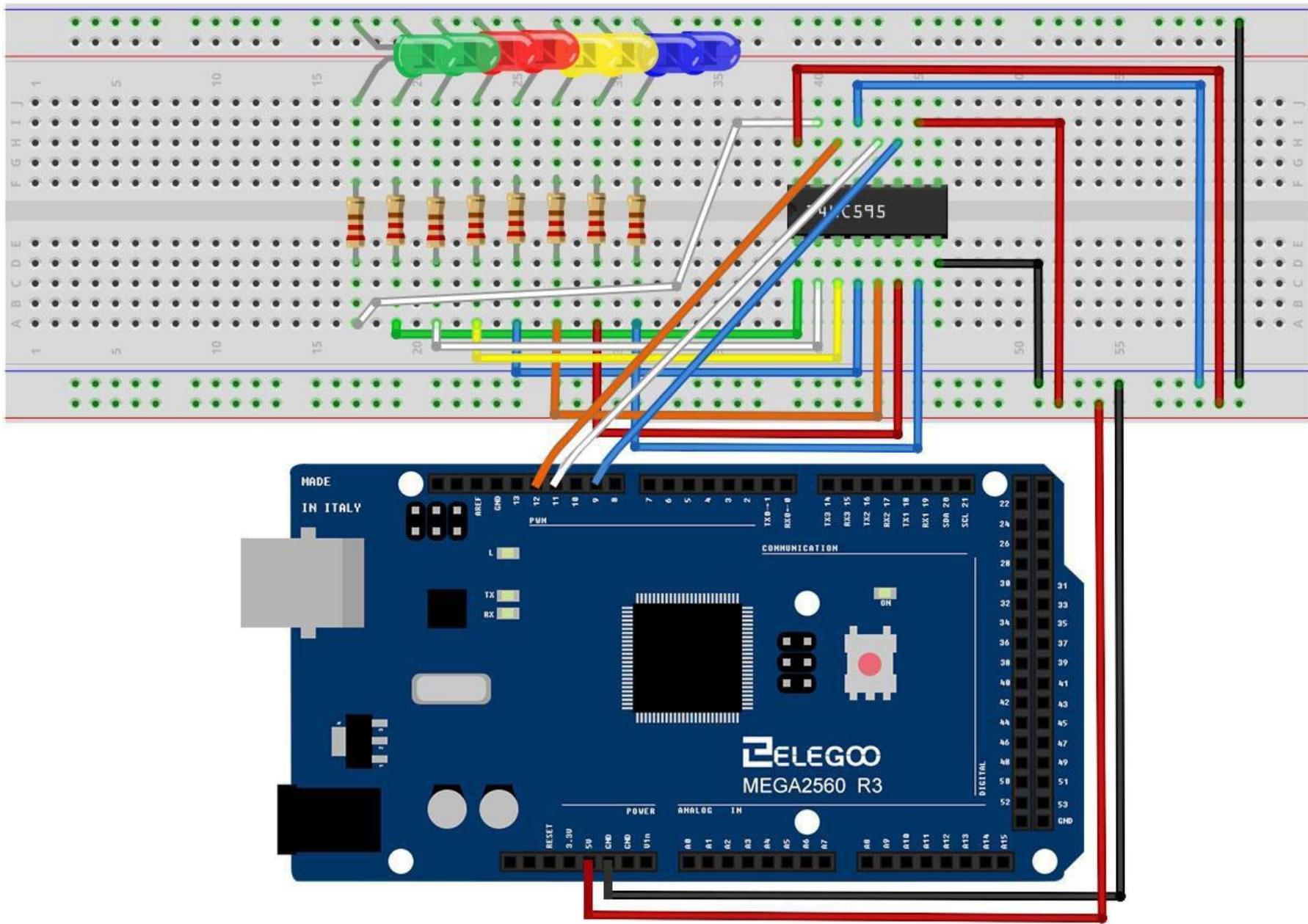
Der Chip verfügt auch über einen Output-Freigabe-Pin (OE), mit dem alle Ausgänge auf einmal aktiviert oder deaktiviert werden können. Sie können diese an einen PWM-fähigen MEGA2560 Pin anschließen und den Befehl 'analogWrite' verwenden um die Helligkeit der LEDs zu steuern. Dieser Pin ist standardmäßig „LOW“, so dass wir ihn an GND anschließen.

Verbindung

Schema



Schaltplan



Da wir acht LEDs und acht Widerstände zu Verbinden haben, gibt es eigentlich noch ein paar Verbindungen die herzustellen sind.

Es ist wahrscheinlich am einfachsten, den 74HC595 Chip zuerst auf dem Breadboard anzubringen, da so ziemlich alles andere mit ihm verbunden wird. Setzen Sie es so, dass die kleine U-förmige Kerbe in Richtung der Spitze des Breadboard ist. Pin 1 des Chips befindet sich links von dieser Kerbe.

Digital 12 vom MEGA2560 geht an Pin # 14 des Schieberegisters

Digital 11 vom MEGA2560 geht an Pin # 12 des Schieberegisters

Digital 9 vom MEGA2560 geht an Pin # 11 des Schieberegisters

Alle, bis auf einen der Ausgänge vom IC befinden sich auf der linken Seite des Chips.

Für die einfache Verbindung ist das auch dort, wo die LEDssind.

Nach dem Chip, setzen Sie die Widerstände auf dem Breadboard ein. Sie müssen vorsichtig sein, dass sich keiner der Leitungen der Widerstände gegenseitig berühren. Sie sollten dies noch einmal überprüfen, bevor Sie die Stromversorgung des MEGA2560 anschließen. Wenn es schwierig ist, die Widerstände ohne dass sie sich Berührung zu ordnen, dann hilft es die Leitungen zu verkürzen, so dass sie näher an der Oberfläche des Steckbretts liegen.

Als nächstes legen Sie die LEDs auf das Steckbrett. Die längeren positiven LED-Leads müssen alle zum Chip gehen, je nachdem, auf welcher Seite des Steckbrettes sie sind.

Bringen Sie die Jumper-Anschlüsse wie oben gezeigt an. Vergessen Sie nicht die Anschlüsse, die von Pin 8 des IC auf die GND Spalte des Steckbretts gehen.

Laden Sie den Sketch ein hoch und probieren Sie es aus. Jede LED sollte so lang leuchten, bis alle LEDs eingeschaltet sind, und dann gehen sie alle aus und der Zyklus wiederholt sich.

Code

[Nach der Verschaltung öffnen Sie bitte das Programm im Code-Ordner - Lektion 24](#)

[Acht LED mit 74HC595 und klicken Sie auf UPLOAD, um das Programm hochzuladen.](#)

[Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.](#)

Das erste, was wir tun, ist die drei Pins zu definieren, die wir verwenden werden. Dies sind die digitalen Ausgänge MEGA2560, die an die Latch-, Takt- und Datenpins des 74HC595 angeschlossen werden.

```
int latchPin = 11;
```

```
int clockPin = 9;
```

```
int dataPin = 12;
```

Als nächstes wird eine Variable namens 'leds' definiert. Diese wird verwendet, um das Muster zu erhalten, welche LEDs gerade ein- oder ausgeschaltet sind. Daten des Typs 'Byte' stellen Zahlen mit acht Bits dar. Jedes Bit kann entweder ein oder aus, so ist dies perfekt zum Nachvollziehen, welche unserer acht LEDs an oder aus sind.

```
byte leds = 0;
```

Die "Setup" -Funktion setzt einfach die drei Pins, die wir verwenden, auf digitale Ausgänge.

```
void setup()
```

```
{
```

```
    pinMode(latchPin, OUTPUT);
    pinMode(dataPin, OUTPUT);
    pinMode(clockPin, OUTPUT);
```

```
}
```

Die "Loop" -Funktion schaltet zunächst alle LEDs aus, indem sie der Variable 'leds' den Wert 0 gibt. Dann ruft sie 'updateShiftRegister' auf, der das 'leds'-Muster an das Schieberegister sendet, damit alle LEDs ausschalten. Wir werden uns damit, wie 'updateShiftRegister' funktioniert, später beschäftigen.

Die Loop-Funktion pausiert für eine halbe Sekunde und beginnt dann, mit Hilfe der 'for' Schleife und der Variablen 'i', von 0 bis 7 zu zählen. Es verwendet jedes Mal die Arduino-Funktion 'bitSet', um das Bit zu setzen, das diese LED in der Variablen 'leds' steuert. Dann rufen Sie auch den Befehl 'updateShiftRegister' auf, so dass die LEDs aktualisiert werden und geschaut wird, was in der Variablen 'leds' steht.

Es gibt dann eine halbe Sekunde Verzögerung, bevor 'i' inkrementiert wird und die nächste LED leuchtet.

```
void loop()
```

```
{
```

```
    leds = 0;
    updateShiftRegister();
    delay(500);
    for (int i = 0; i < 8; i++)
    {
        bitSet(leds, i);
        updateShiftRegister(); delay(500);
```

}

Die Funktion 'updateShiftRegister' setzt zunächst den LatchPin auf Low und ruft dann die MEGA2560 Funktion 'shiftOut' auf, bevor man den 'latchPin' wieder High setzt. Dies dauert vier Parameter, die ersten beiden sind die Pins für Data und Clock die angesteuert werden.

Der dritte Parameter gibt an, mit welchem Ende der Daten Sie starten möchten. Wir werden meistens mit dem rechten Ende anfangen, was als "Least Significant Bit" (LSB) bezeichnet wird.

Der letzte Parameter ist die in das Schieberegister zu verschiebende Ist-Daten, die in diesem Fall 'led' ist.

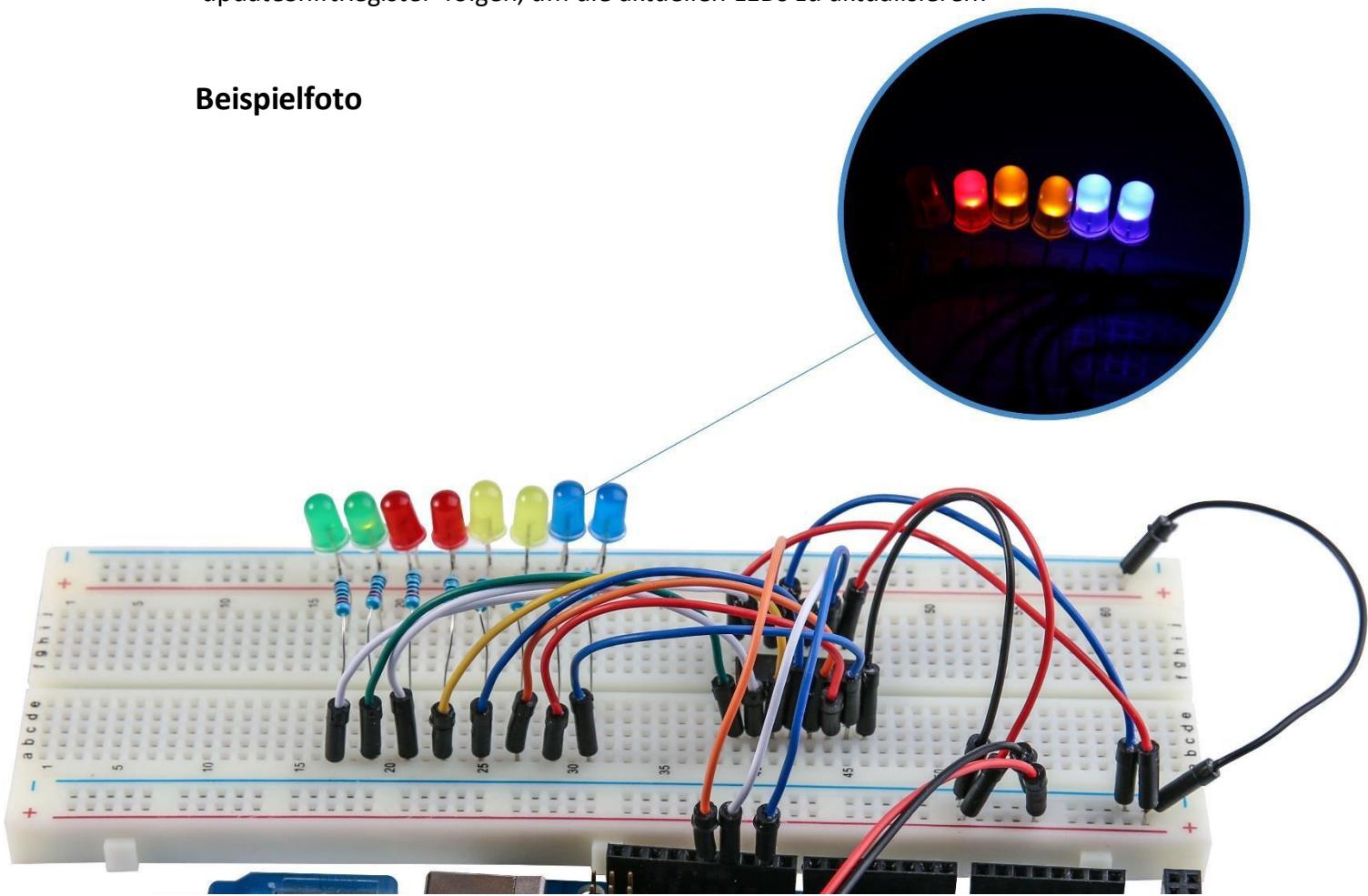
void updateShiftRegister()

{

```
digitalWrite(latchPin, LOW);
shiftOut(dataPin, clockPin, LSBFIRST, leds);
digitalWrite(latchPin, HIGH);
```

Wenn Sie eine der LEDs ausschalten möchten, dann nehmen Sie eine ähnliche Arduino-Funktion (`bitClear`) mit der Variable 'leds' auf. Dies wird das Bit von 'leds' auf 0 setzen und Sie müssen dann nur noch mit einem Aufruf von '`updateShiftRegister`' folgen, um die aktuellen LEDs zu aktualisieren.

Beispielfoto



Lektion 25 Der serielle Monitor

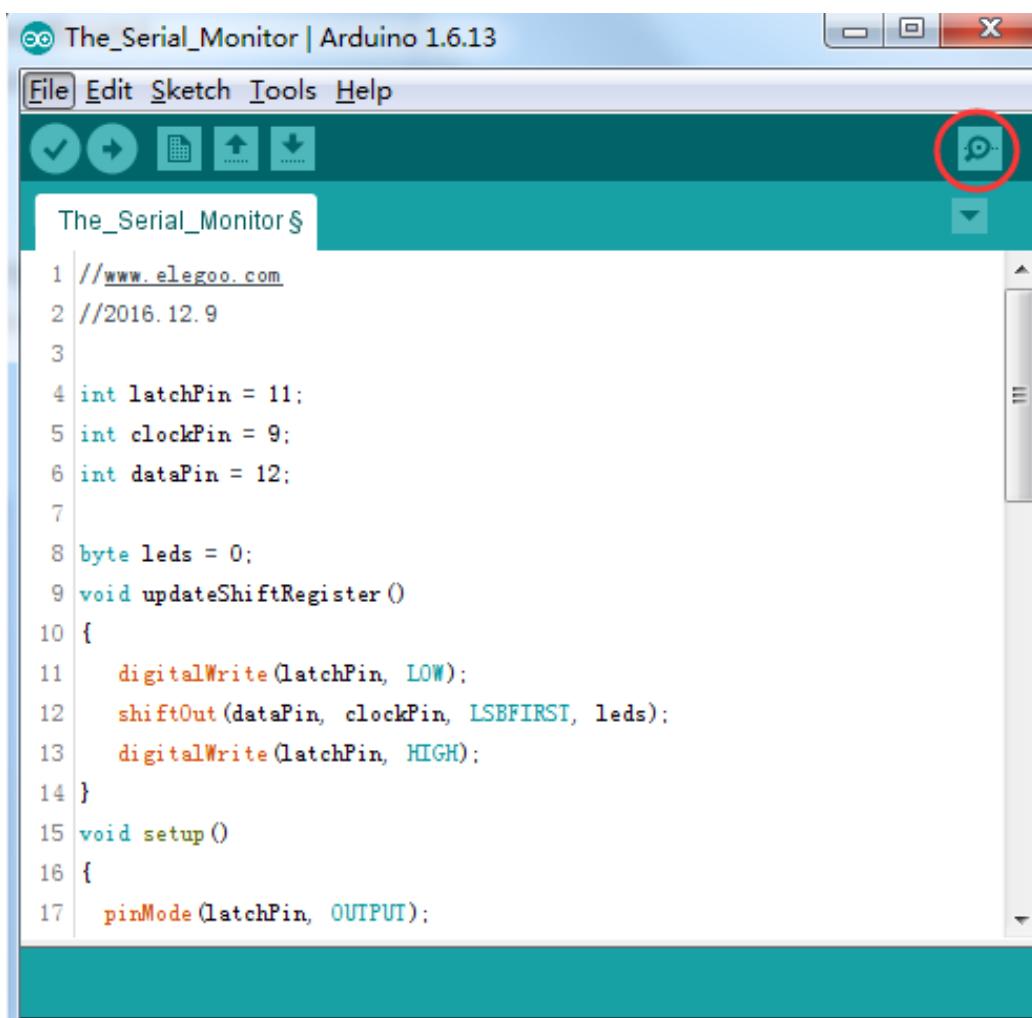
Überblick

In dieser Lektion bauen Sie Lektion 24 auf und fügen die Möglichkeit hinzu, die LEDs von Ihrem Computer mit dem Arduino Serial Monitor zu steuern. Der serielle Monitor ist die Verbindung zwischen dem Computer und Ihrem MEGA2560. Es kann für Sie Textnachrichten senden und empfangen, dies ist praktisch zum Debuggen und auch das Steuern des MEGA2560 mit einer Tastatur! Zum Beispiel können Sie Befehle von Ihrem Computer senden, um LEDse einzuschalten.

In dieser Lektion werden Sie genau die gleichen Teile und ein ähnliches Bread-Layout wie in Lektion 24 verwenden. Wenn Sie dies noch nicht getan haben, folgen Sie Lektion 24 jetzt.

Schritte erledigt

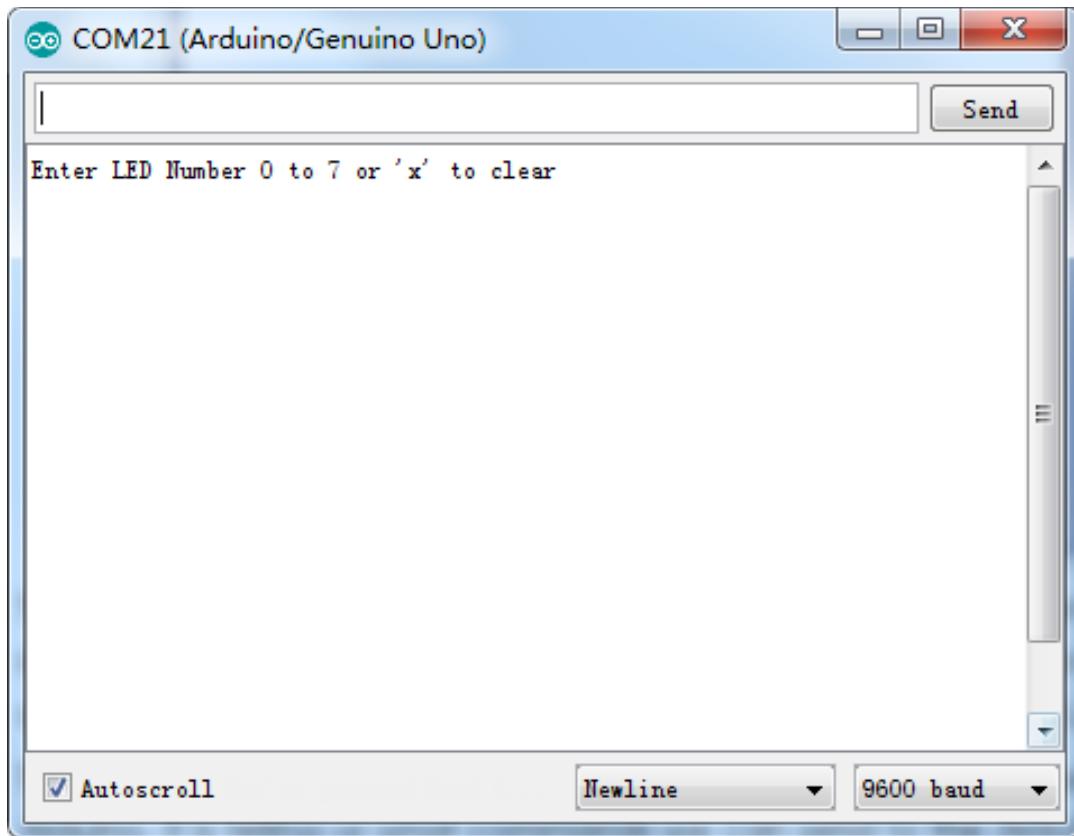
Nachdem Sie diesen Sketch auf Ihren MEGA2560 hochgeladen haben, klicken Sie auf die rechte Schaltfläche auf der Symbolleiste in der Arduino IDE. Die Schaltfläche ist unten eingekreist.



```
1 //www.elegoo.com
2 //2016.12.9
3
4 int latchPin = 11;
5 int clockPin = 9;
6 int dataPin = 12;
7
8 byte leds = 0;
9 void updateShiftRegister()
10 {
11     digitalWrite(latchPin, LOW);
12     shiftOut(dataPin, clockPin, LSBFIRST, leds);
13     digitalWrite(latchPin, HIGH);
14 }
15 void setup()
16 {
17     pinMode(latchPin, OUTPUT);
```

Das folgende Fenster wird geöffnet.

Klicken Sie auf die Schaltfläche Serial Monitor, um den seriellen Monitor einzuschalten. Die Grundlagen des seriellen Monitors werden in Lektion 1 ausführlich vorgestellt.



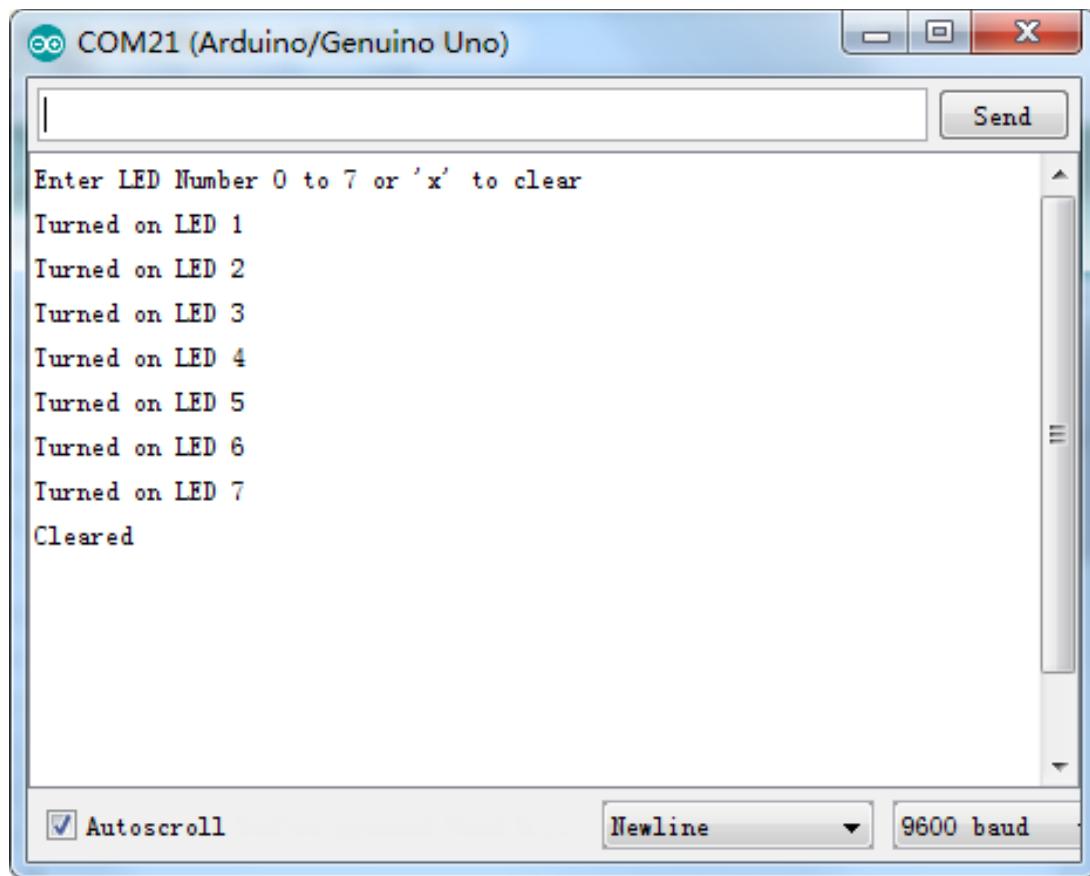
Dieses Fenster heißt Serialer Monitor und gehört zur Arduino IDE Software. Sein Job ist es, Ihnen zu ermöglichen, Nachrichten von Ihrem Computer zu einem MEGA2560 Board (über USB) zu senden und auch Nachrichten vom MEGA2560 zu empfangen.

Die Nachricht "Enter LED Number 0 bis 7 or 'x' to clear" wurde vom Arduino gesendet. Es sagt uns, welche Befehle wir an den Arduino senden können: entweder die 'x' (um alle LEDs auszuschalten) oder die Nummer der LED, die er einschalten soll (wo 0 die untere LED ist, 1 ist die nächste bis hin zu 7 für die obere LED).

Versuchen Sie, die folgenden Befehle in den oberen Bereich neben der Schaltfläche „Send“ des seriellen Monitors einzugeben. Drücken Sie 'Send', nachdem Sie jedes dieser Zeichen eingegeben haben: x 0 3 5

Das Eingeben von x hat keine Auswirkung, wenn die LEDs schon alle ausgeschaltet sind, aber wenn Sie jede einzelne Zahl eingeben, sollte die entsprechende LED leuchten und Sie erhalten eine Bestätigungsmeldung von dem MEGA2560-Board.

Der serielle Monitor erscheint wie unten gezeigt.



Geben Sie x erneut ein und drücken Sie 'Senden', um alle LEDsauszuschalten.

Code

Nach der Verdrahtung öffnen Sie bitte das Programm im Codeordner - Lektion 25
Der serielle Monitor und klicken Sie auf UPLOAD, um das Programm hochzuladen.
Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Wie Sie vielleicht erwarten, basiert der Sketch auf dem Sketch, der in Lektion 24 verwendet wird. So werden wir hier nur die neuen Bits abdecken.

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

```
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");
}

```

Erstens haben wir den Befehl 'Serial.begin (9600)'. Dies startet die serielle Kommunikation, so dass der MEGA2560 Befehle über die USB-Verbindung aussenden kann. Der Wert 9600 deklariert die 'Baudrate' der Verbindung. So schnell werden die Daten gesendet. Sie können dies auf einen höheren Wert ändern, aber Sie müssen auch den Arduino Serial Monitor auf den gleichen Wert ändern. Wir werden dies später besprechen. Für jetzt, lassen Sie es bei 9600.

Die Zeile, die mit 'while' beginnt, sorgt dafür, dass am anderen Ende des USB-Anschlusses etwas für das Arduino vorhanden ist, bevor es mit dem Senden von Nachrichten beginnt. Andernfalls kann die Nachricht gesendet, aber nicht angezeigt werden. Diese Zeile ist eigentlich nur notwendig, wenn man einen Arduino Leonardo benutzt, weil der Arduino MEGA2560 automatisch den Arduino Board zurücksetzt, wenn man den Serial Monitor öffnet, während dies bei Leonardo nicht passiert.

Die letzte der neuen Zeilen in 'setup' sendet die Nachricht, die wir am oberen Rand des seriellen Monitors sehen.

Die "Loop" -Funktion ist, wo die ganze Aktion passiert:

```

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");
}
}

```

Alles, was in der Schleife passiert, ist in einer "if" -Anweisung enthalten. Also, es sei denn, der Aufruf zur eingebauten Arduino-Funktion 'Serial.available ()' ist 'true', dann wird nichts anderes passieren.

Serial.available () gibt 'true' zurück, wenn Daten an den MEGA2560 gesendet wurden und dort fertig sind. Eingehende Nachrichten werden im sogenannten Puffer gespeichert und Serial.available () gibt true zurück, wenn dieser Puffer nicht leer ist.

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

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

Dies liest das nächste Zeichen aus dem Puffer und entfernt es aus dem Puffer. Sie weist sie auch der Variablen 'ch' zu. Die Variable 'ch' ist vom Typ 'char', die für 'character' steht und wie der Name schon sagt, ein einziges Zeichenenthält.

Wenn Sie die Anweisungen in der Eingabeaufforderung am oberen Rand des seriellen Monitors befolgt haben, dann ist dieses Zeichen entweder eine einstellige Zahl zwischen 0 und 7 oder der Buchstabe 'x'.

Die "if" -Anweisung in der nächsten Zeile prüft, ob es eine einzelne Ziffer ist, indem sie sieht, 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 vollkommen akzeptabel.

Jedes Zeichen wird durch eine eindeutige Nummer, ASCII-Wert genannt, dargestellt.

Dies bedeutet, dass, wenn wir Zeichen mit <= und > = vergleichen sind es tatsächlich die ASCII-Werte, die verglichen wurden.

Wenn der vergleich fertig ist, kommen wir zur nächsten Zeile:

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

Jetzt rechnen wir mit Zeichen! Wir subtrahieren die Ziffer '0' von der beliebigen Ziffer. Wenn du also '0' eingegeben hast, dann ist '0' - '0' gleich 0. Wenn du '7' eingegeben hast, dann '7' - '0' entspricht der Nummer 7, weil es eigentlich die ASCII-

Werte sind, die benutzt werden in der Subtraktion.

Da wir die Nummer der LED kennen, die wir einschalten wollen, müssen wir nur das Bit in die Variable 'leds' setzen und das Schieberegister aktualisieren.

```
bitSet(leds, led);  
updateShiftRegister();
```

In den nächsten beiden Zeilen schreiben eine Bestätigungs Nachricht an den seriellen Monitor zurück.

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

Die erste Zeile verwendet Serial.print anstatt Serial.println. Der Unterschied zwischen den beiden ist, dass Serial.print nicht sofort eine neue Zeile startet nach dem anzeigen. Wir verwenden das in der ersten Zeile, weil wir die Nachricht in zwei Teile drucken. Zuerst das allgemeine Bit: 'LED eingeschaltet' und dann die Nummer der LED.

Die Nummer der LED wird in einer Variable "int" gehalten, anstatt in einer Textzeichenfolge. Serial.print kann entweder einen Text-String in Anführungszeichen, oder ein 'int' oder für dieses Thema so ziemlich jede Art von Variable aufnehmen.

Nach der "if" -Anweisung, die den Fall behandelt, wenn eine einzelne Ziffer behandelt wurde, gibt es eine zweite "if" -Anweisung, die überprüft, ob die Variable 'ch' der Buchstabe 'x' ist.

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

Wenn dem so ist, dann löscht es alle LEDs und sendet eine Bestätigungs Nachricht.

Lektion 26 Photozelle

Überblick

In dieser Lektion lernen Sie, wie man die Lichtintensität mit einem analogen Eingang mißt. Sie wird auf Lektion 25 aufbauen und die Lichtstärke verwenden, um die Anzahl der zu beleuchtenden LEDs zu kontrollieren.

Die Lichtschranke befindet sich am Boden des Breadboards.

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x 830 tie-points breadboard
- (8) x LEDs
- (8) x 220 ohm Widerstände
- (1) x 1k ohm Widerstand
- (1) x 74hc595 IC
- (1) x Photowiderstand (Photozelle)
- (16) x M-M Kabel (Male to Male jumper Kabel)



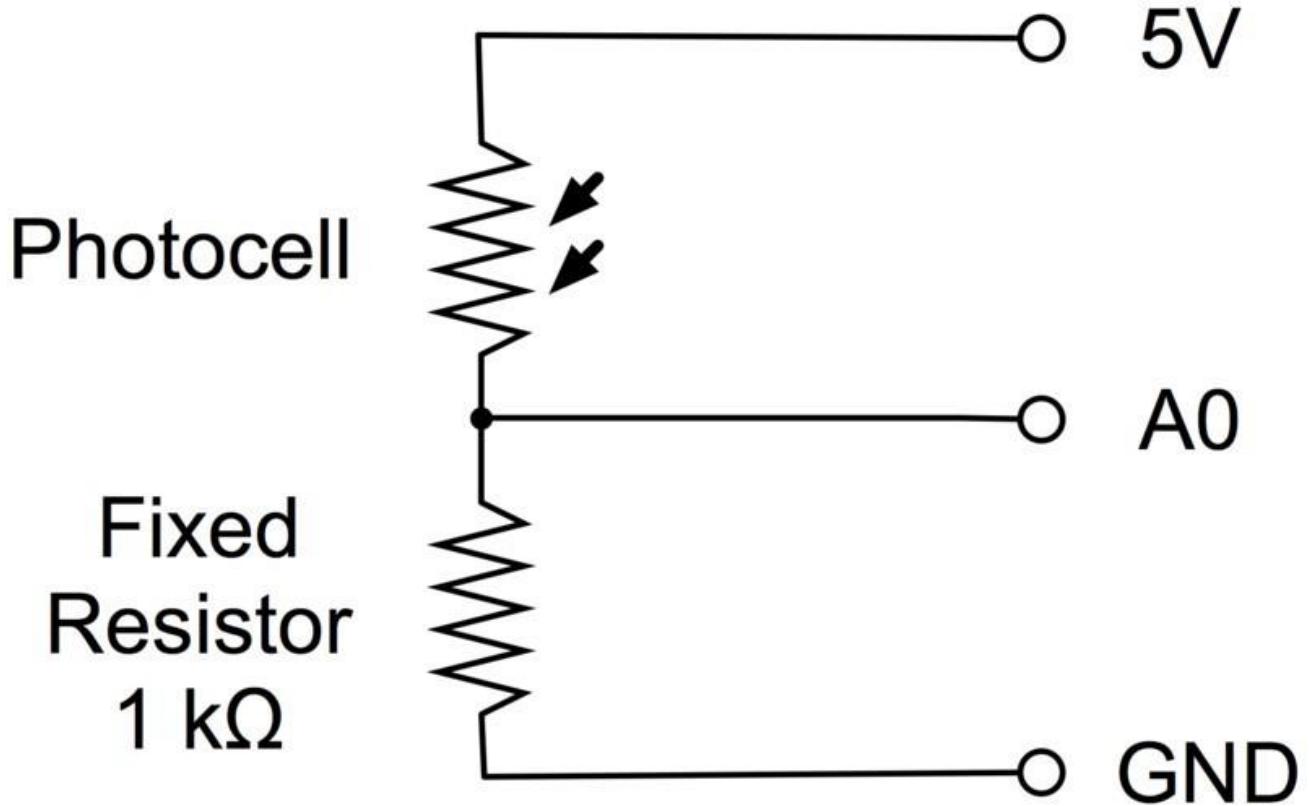
Komponenten Einführung

PHOTOZELLE:

Die verwendete Photozelle ist von einem Typ, der als lichtabhängiger Widerstand bezeichnet wird, der manchmal als LDR bezeichnet wird. Wie der Name schon sagt, wirken diese Komponenten genau wie ein Widerstand, außer dass sich der Widerstand in Reaktion darauf ändert, wie viel Licht auf sie fällt.

Dieser hat einen Widerstand von etwa $50\text{ k}\Omega$ bei Nahdunkelheit und $500\text{ }\Omega$ bei hellem Licht. Um diesen variierenden Wert des Widerstands in etwas umzuwandeln, das wir an einem analogen Eingang der MEGA2560 R3-Platine messen können, muss er in eine Spannung umgewandelt werden.

Der einfachste Weg, dies zu tun, ist, es mit einem festen Widerstand zu kombinieren.

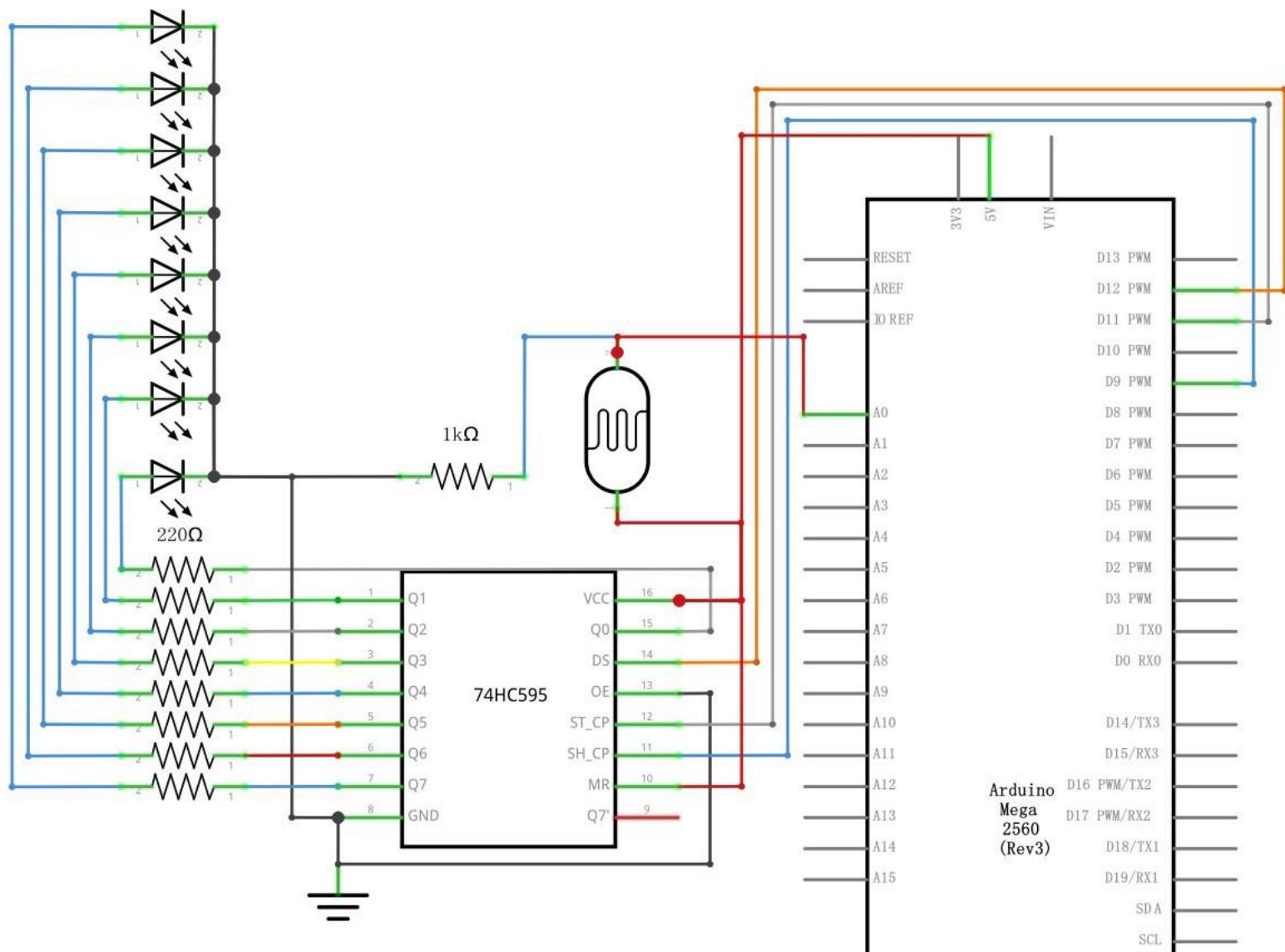


Der Widerstand und die Photozelle verhalten sich wie ein Poti. Wenn das Licht sehr hell ist, dann ist der Widerstand der Photozelle sehr niedrig im Vergleich zu dem Festwert-Widerstand, und so ist es, als ob der Poti auf Maximum gedreht wurde.

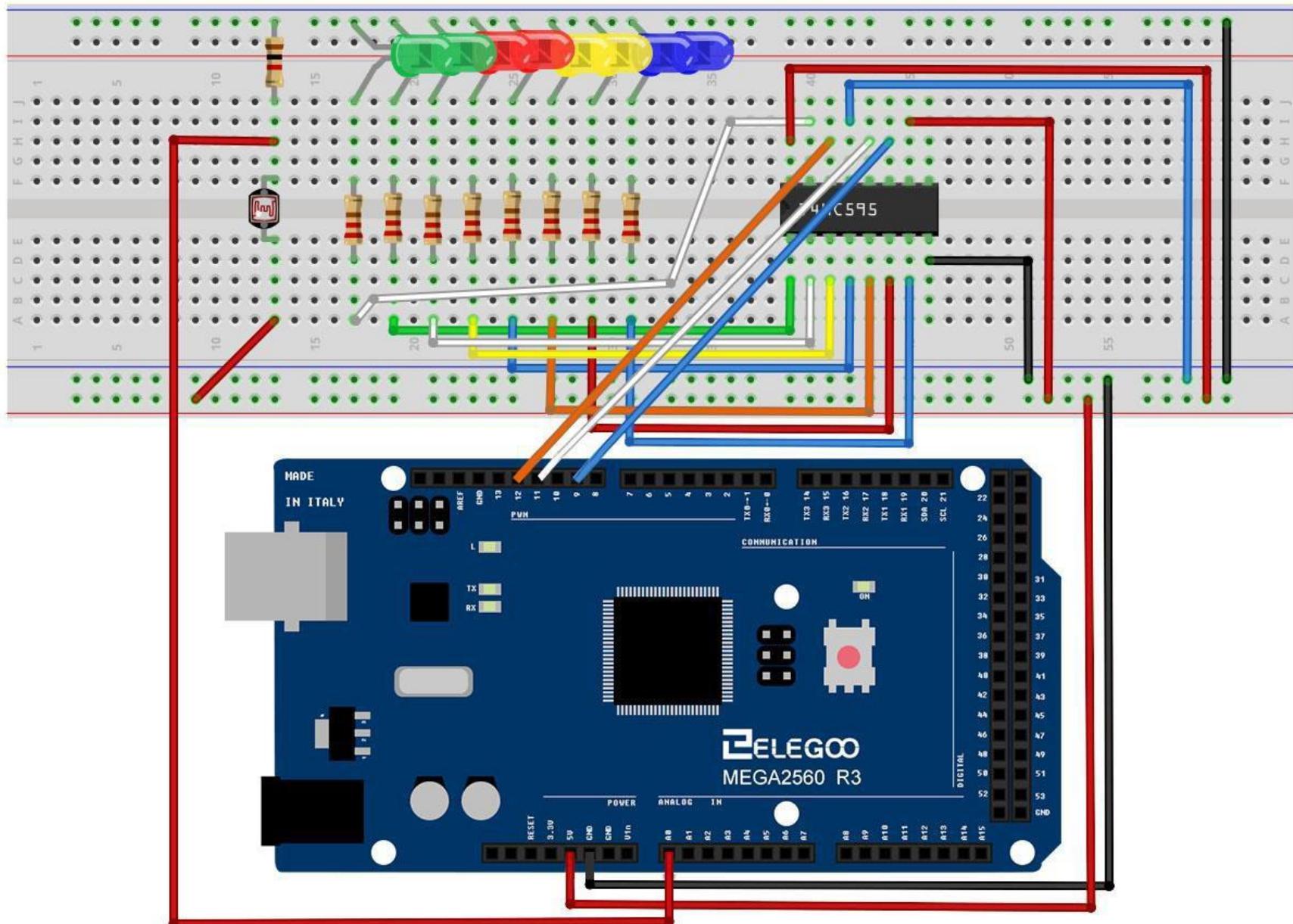
Wenn die Photozelle in gedimmten Licht ist, wird der Widerstand größer als der feste $1\text{ k}\Omega$ Widerstand und es ist, als ob der Poti in Richtung GND gedreht wurde.

Laden Sie den Sketch, der im nächsten Abschnitt gegeben wird, und versuchen Sie, die Fotozelle mit dem Finger zu bedecken, und halten Sie sie dann in der Nähe einer Lichtquelle.

Verbindung Schema



Schaltplan



Code

Nach der Verschaltung öffnen Sie bitte das Programm im Codeordner - Lektion 26 Lichtschranke und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

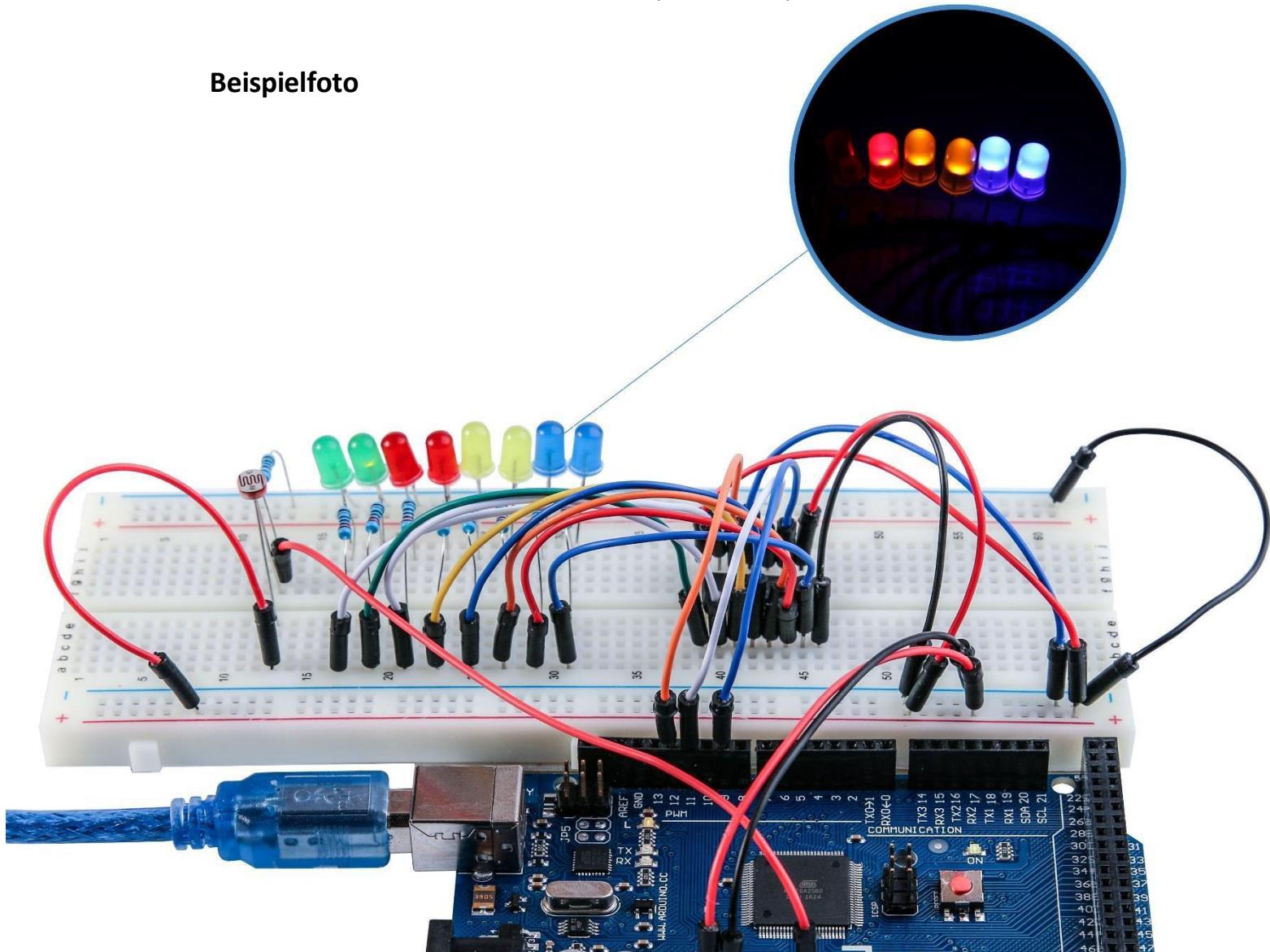
Das erste, was zu beachten ist, dass wir den Namen des Analog-Pins geändert haben, um 'lightPin' zu sein, anstatt 'potPin', da wir keinen Poti mehrhaben.

Die einzige andere wesentliche Änderung des Sketches ist die Zeile, in der berechnet wird, wie viele der LEDs leuchten sollen:

```
int numLEDSLit = reading / 57; // all LEDs lit at 1k
```

Dieses Mal teilen wir die ausgelesenen Werte mit 57 anstatt 114. Mit anderen Worten, wir teilen es um die Hälfte, so viel wie mit dem Poti, um es in neun Zonen zu teilen, von keine der LEDs leuchtet auf alle acht leuchten. Dieser zusätzliche Faktor ist für den festen 1 kΩ Widerstand verantwortlich. Dies bedeutet, dass, wenn die Lichtschranke einen Widerstand von 1 kΩ hat (der gleiche wie der feste Widerstand), der rohe Messwert $1023/2 = 511$ ist. Dies entspricht allen LEDs, die beleuchtet werden, und dann wird der Bit (numLEDSLit) 8 sein

Beispielfoto



Lektion 27 74HC595 und Segmentanzeige

Überblick

Nach den Lektionen 24,25 und Lektion 26 verwenden wir das Schieberegister 74HC595, um die Segmentanzeige zu steuern. Die Segmentanzeige zeigt die Nummer von 9-0 an.

Erforderliche Komponenten:

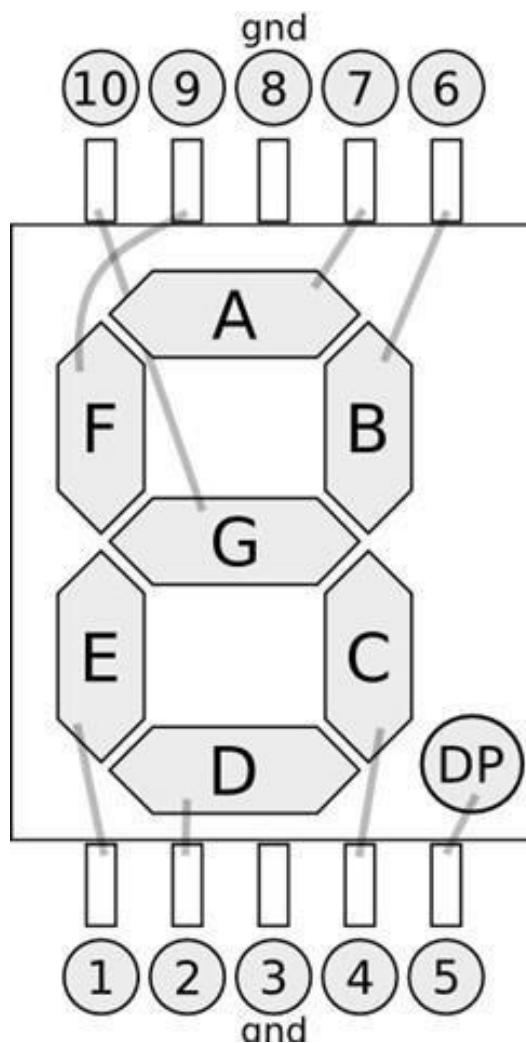
- (1) x Elegoo Mega2560 R3
- (1) x 830 tie-points breadboard
- (1) x 74HC595 IC
- (1) x 1 Digit 7-Segment Display
- (8) x 220 ohm Widerstand
- (26) x M-M Kabel (Male to Male jumper Kabel)



Komponenten Einführung

Sieben Segment Anzeige

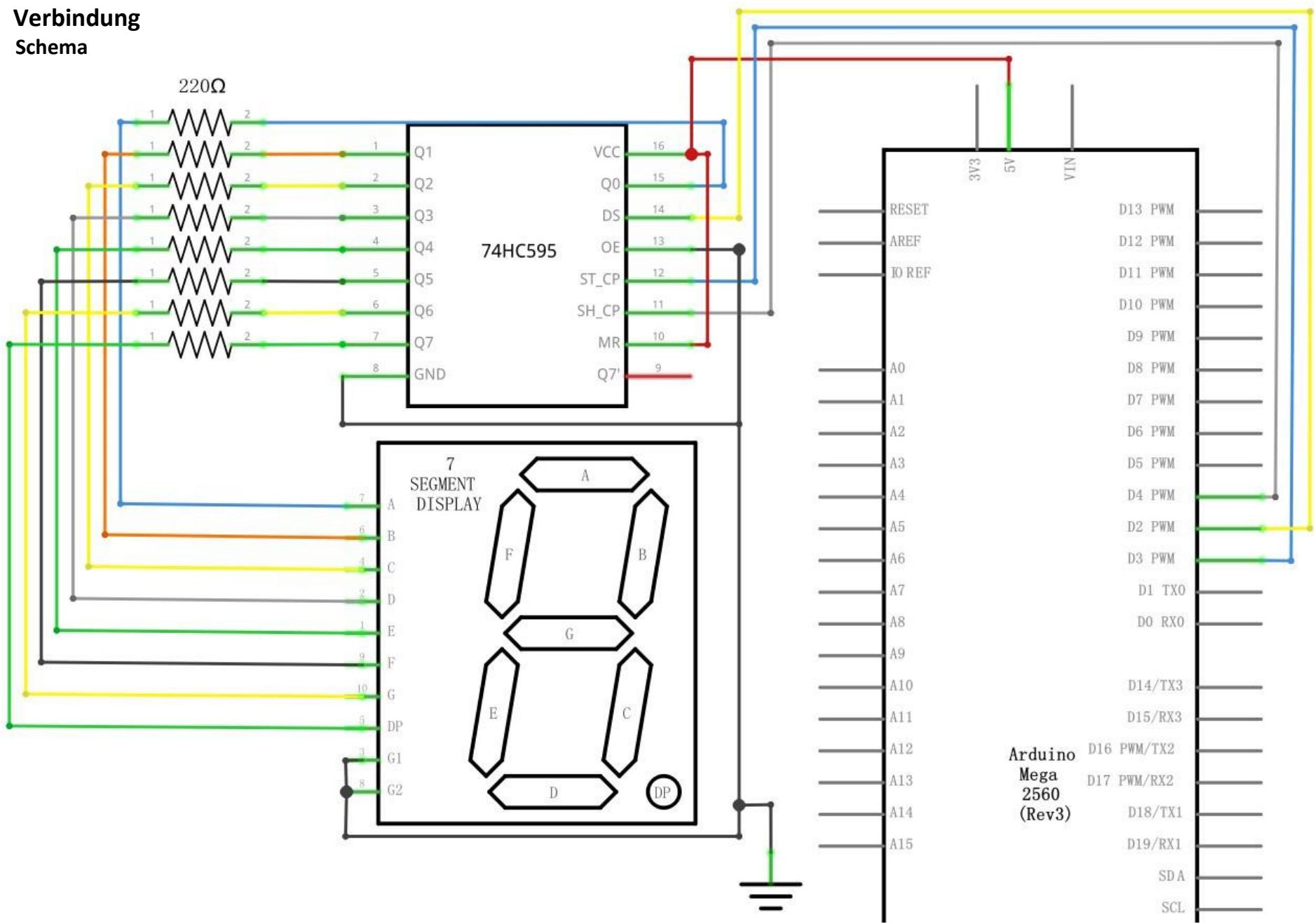
Unten ist das sieben-Segment-Pin-Diagramm.

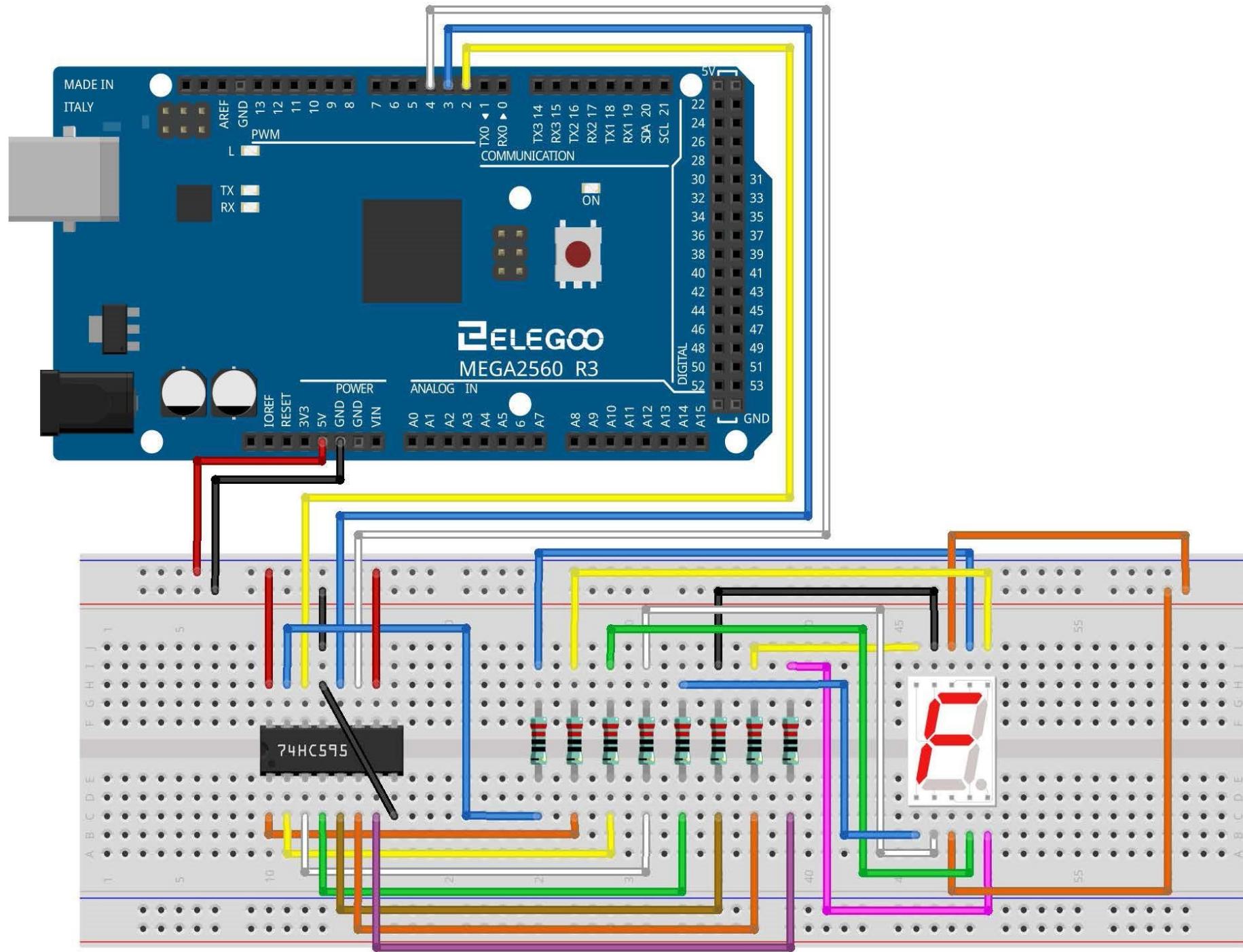


0-9 zehn Ziffern entsprechen jedem Segment sind wie folgt (die folgende Tabelle gilt gemeinsame Kathode sieben Segment Display-Gerät, wenn Sie eine gemeinsame Anode verwenden, sollte die Tabelle ersetzt werden alle 1 0 0 sollte alle ersetzt durch 1):

| Digital Display | 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 |

Verbindung Schema





Die folgende Tabelle zeigt die 7-Segment-Anzeige 74HC595 Pin-Korrespondenztabelle:

| 74HC595 pin | Segmentanzeigen-Pinbelegung |
|-------------|-----------------------------|
| Q0 | 7 (A) |
| Q1 | 6 (B) |
| Q2 | 4 (C) |
| Q3 | 2 (D) |
| Q4 | 1 (E) |
| Q5 | 9 (F) |
| Q6 | 10 (G) |
| Q7 | 5 (DP) |

Schritt eins: Verbinden Sie 74HC595

Zuerst wird die Schaltung an Strom und Masseangeschlossen:

VCC (pin 16) und **MR** (pin 10) an 5Vangeschlossen

GND (pin 8) und **OE** (pin 13) an Masse

Verbinde **DS**, **ST_CP** und **SH_CP** pin:

DS (pin 14) verbunden mit MEGA2560 R3 board pin 2

ST_CP (pin 12, latch pin) angeschlossen an MEGA2560 R3 board pin 3 (blaue linie unten)

SH_CP (pin 11, clock pin) angeschlossen an MEGA2560 R3 board pin 4

Schritt 2: Verbinden Sie die siebenSegmentanzeige.

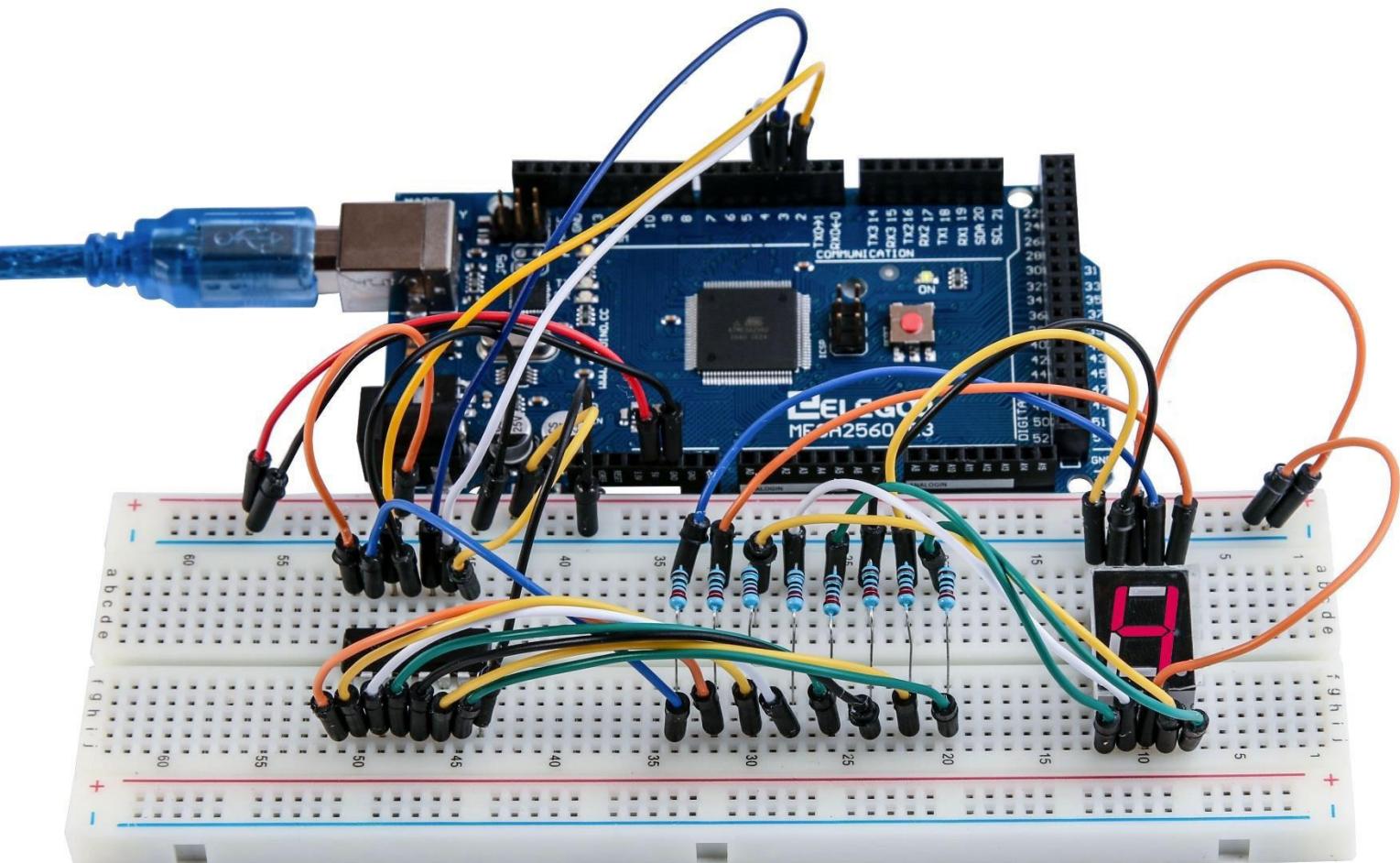
Die 7-Segment-Anzeige 3, 8-polig auf MEGA2560 R3-Karte GND (Dieses Beispiel verwendet die gemeinsame Kathode, wenn Sie die gemeinsame Anode verwenden, verbinden Sie bitte die 3, 8-polige MEGA2560 R3-Karte +5V)

Entsprechend der obigen Tabelle verbinden Sie den 74HC595 Q0 ~ Q7 mit dem Sechs-Segment-Display Pin (A ~ G und DP) und dann jeder Pin mit einem 220 Ohm Widerstand in Serie.

Code

Nach der Verschaltung öffnen Sie bitte das Programm im Codeordner - Lektion 27 74HC595 und Segmentanzeige und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Beispielfoto



Lektion 28 Vier Sieben-Segment-Anzeigen

Überblick

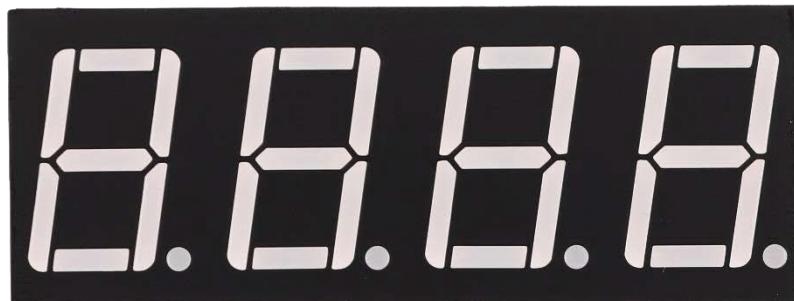
In dieser Lektion lernen Sie, wie Sie eine 4-stellige 7-Segment-Anzeige verwenden können.

Bei der Verwendung einer 1-stelligen 7-Segment-Anzeige beachten Sie bitte, dass bei einer gemeinsamen Anode der gemeinsame Anodenstift mit der Stromquelle verbunden ist. Wenn es eine gemeinsame Kathode ist, verbindet sich der gemeinsame Kathodenstift mit dem GND.

Bei Verwendung einer 4-stelligen 7-Segment-Anzeige wird die gemeinsame Anode oder der gemeinsame Kathodenstift verwendet um zu steuern, welche Ziffer angezeigt wird. Obwohl es nur eine Ziffernarbeit gibt, ermöglicht das Prinzip der Persistenz der Vision (kleiner Trick um unser "dummes" Auge auszutricksen), alle Zahlen zu sehen, da die Scan-Geschwindigkeit so schnell ist, dass man die Intervalle kaum bemerkt.

Erforderliche Komponenten:

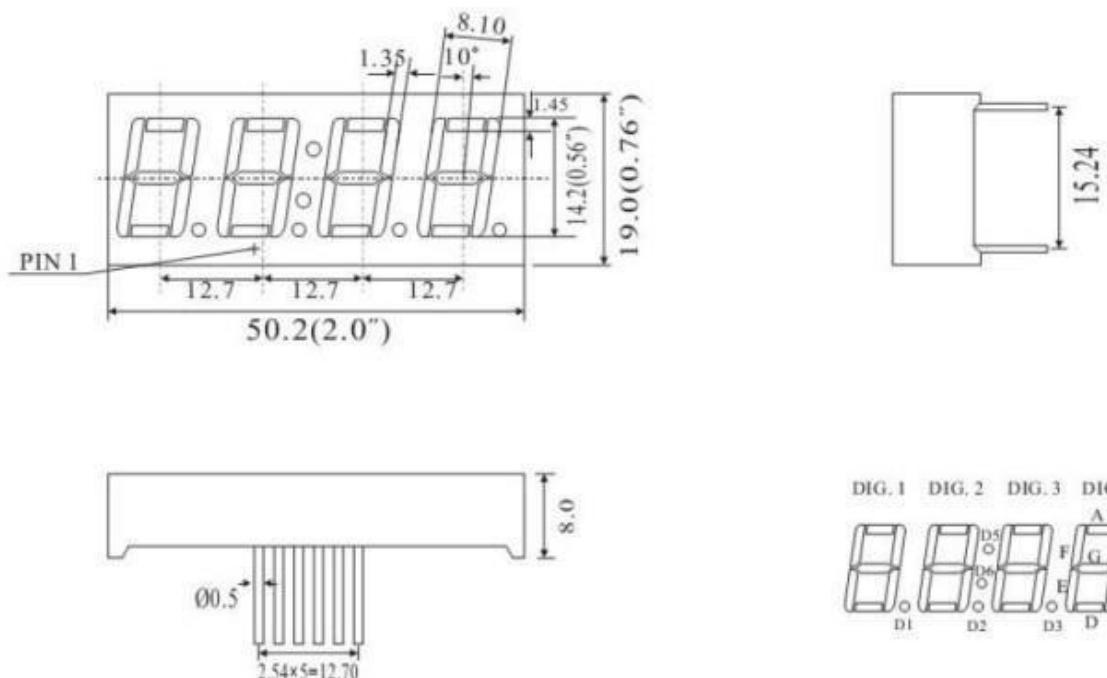
- (1) x Elegoo Mega2560 R3
- (1) x 830 tie-points breadboard
- (1) x 74HC595 IC
- (1) x 4 Digit 7-Segment Displays
- (4) x 220 ohm Widerstände
- (23) x M-M Kabel (Male to Male jumper Kabel)



Bauteil Einführung
Vier Digitale Sieben-Segment-Anzeigen

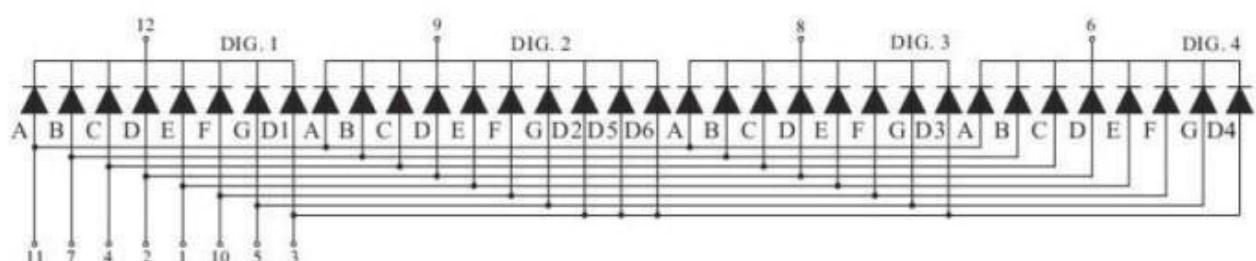
Package Dimensions

CPS05643AB

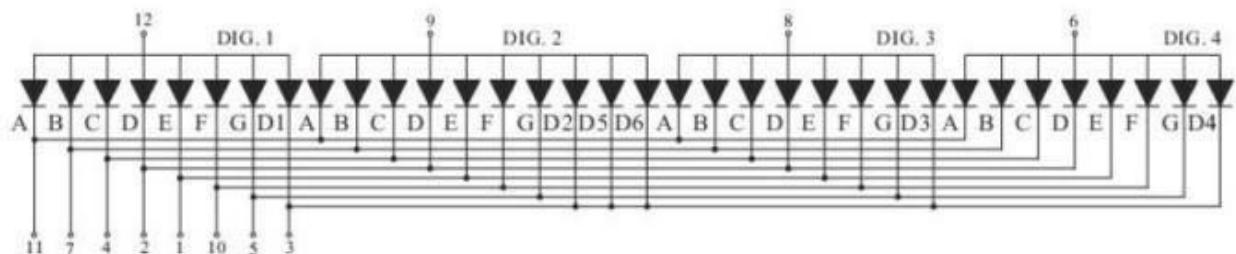


UNIT: MM(INCH) TOLERANCE: $\pm 0.25(0.01")$

Internal Circuit Diagram



5643A

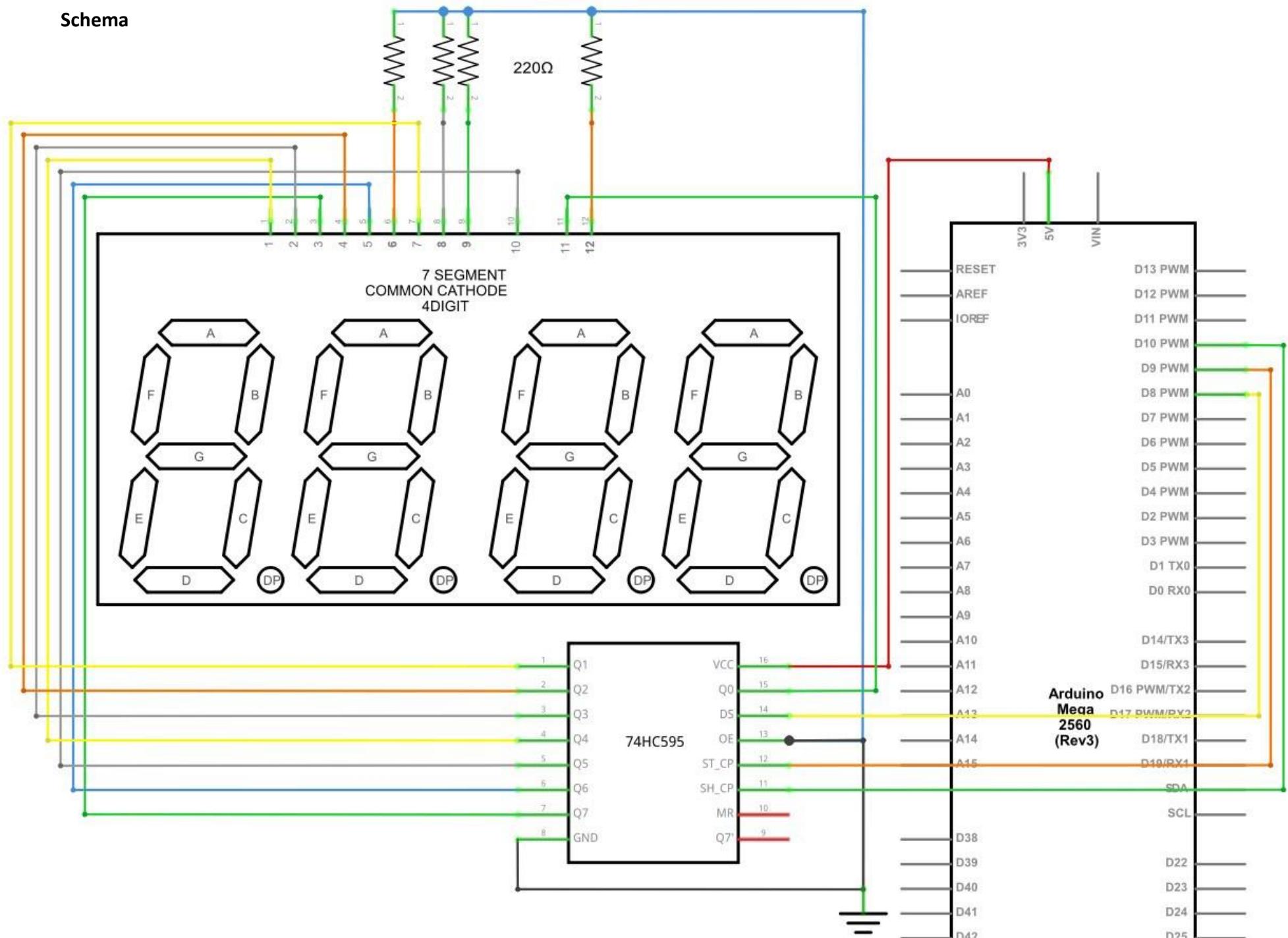


5643B

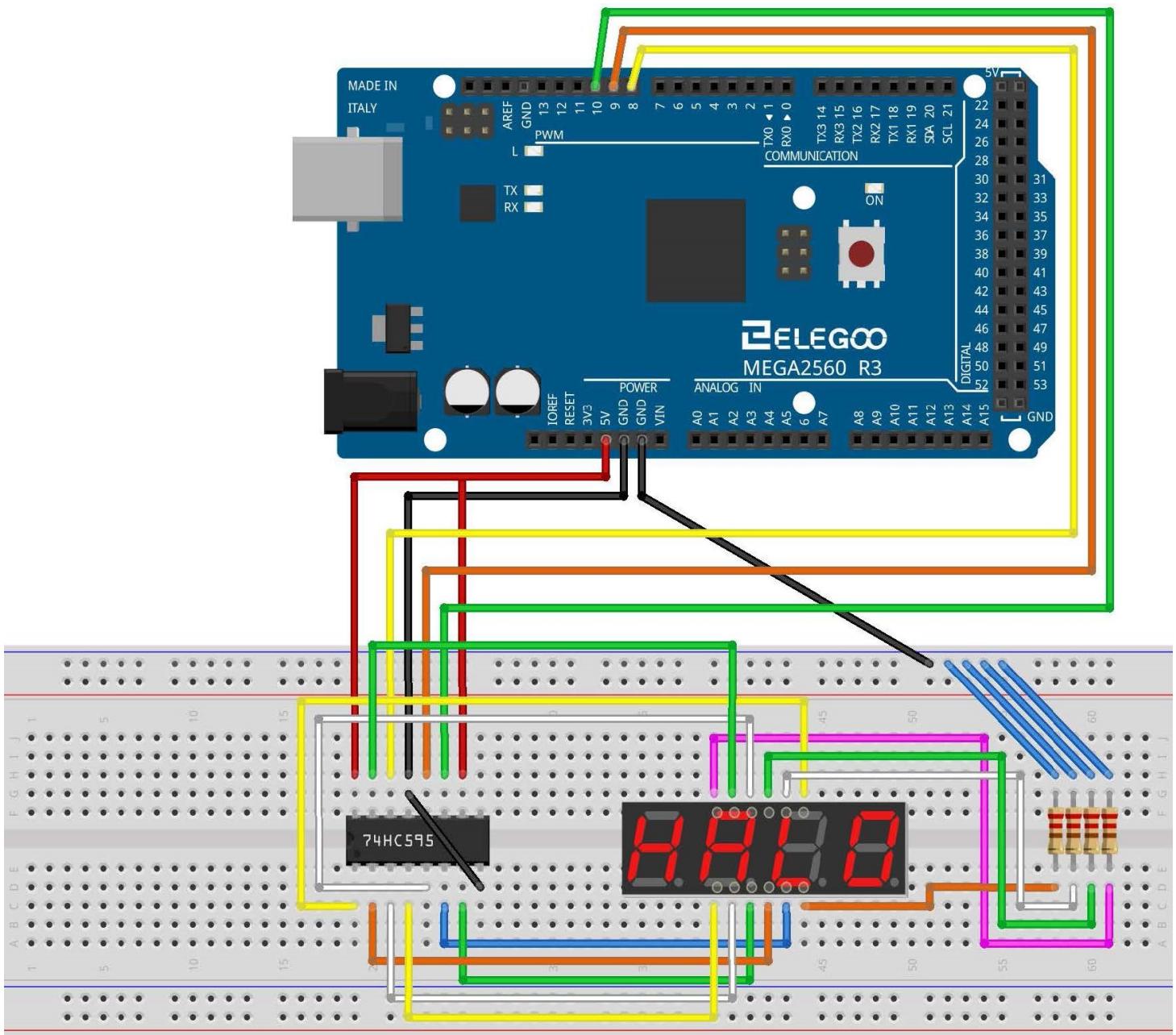
Four Digits Displays Series

Verbindung

Schema



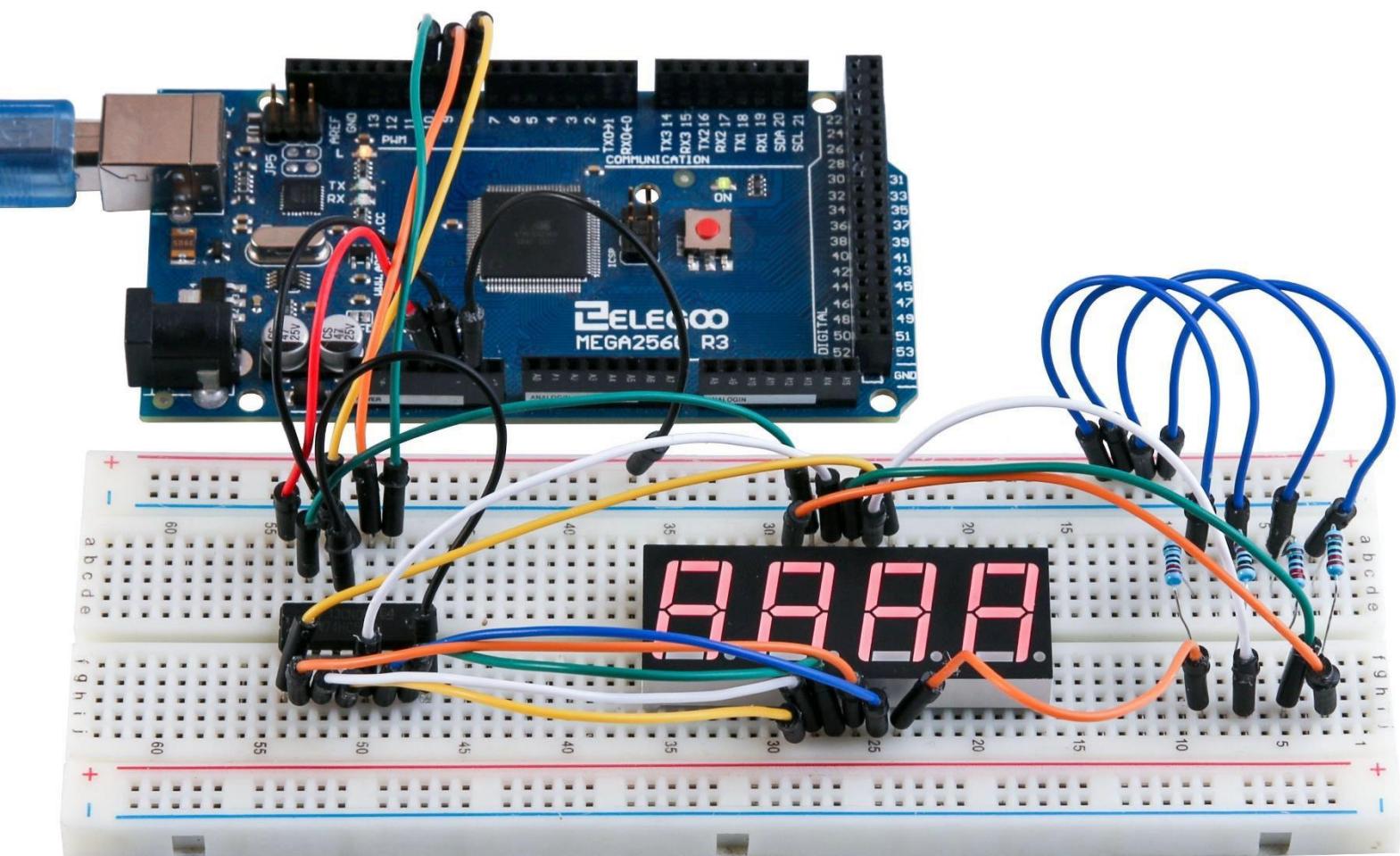
Schaltplan



Code

Nach der Verschaltung öffnen Sie bitte das Programm im Codeordner - Lektion 28 Vier Digitale Sieben Segment Anzeigen und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehlervorliegen.

Beispielfoto



Lektion 29 DC Motor

Überblick

In dieser Lektion lernen Sie, wie man einen kleinen DC-Motor mit einem MEGA2560 R3 mit einem Transistor steuert.

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x 830 tie-points breadboard
- (1) x L293D IC
- (1) x Propeller und 3-6v Motor
- (5) x M-M Kabel (Male to Male jumper Kabel)
- (1) x Netzteilmodul
- (1) x 9V1A Adapter

Komponenten Einführung

Breadboard Energieversorgung

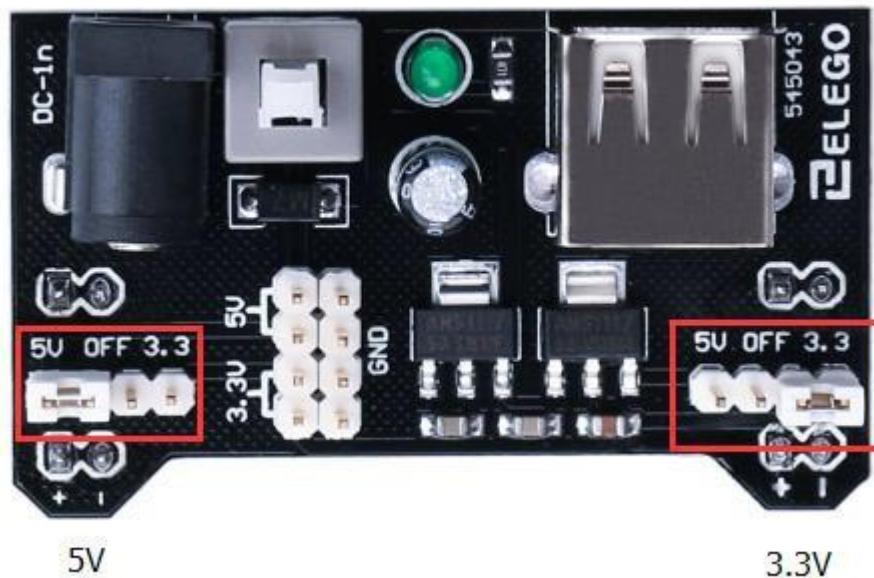
Der kleine DC-Motor wird wahrscheinlich mehr Leistung verwenden, als ein MEGA2560 R3-Board direct ausgeben kann. Wenn wir versuchen würden den Motor direkt an einen MEGA2560 R3 Board Pin anzuschließen, ist die wahrscheinlichkeit groß, dass es das MEGA2560 R3 Board beschädigen könnte. So verwenden wir ein Netzteil-Modul zur Stromversorgung



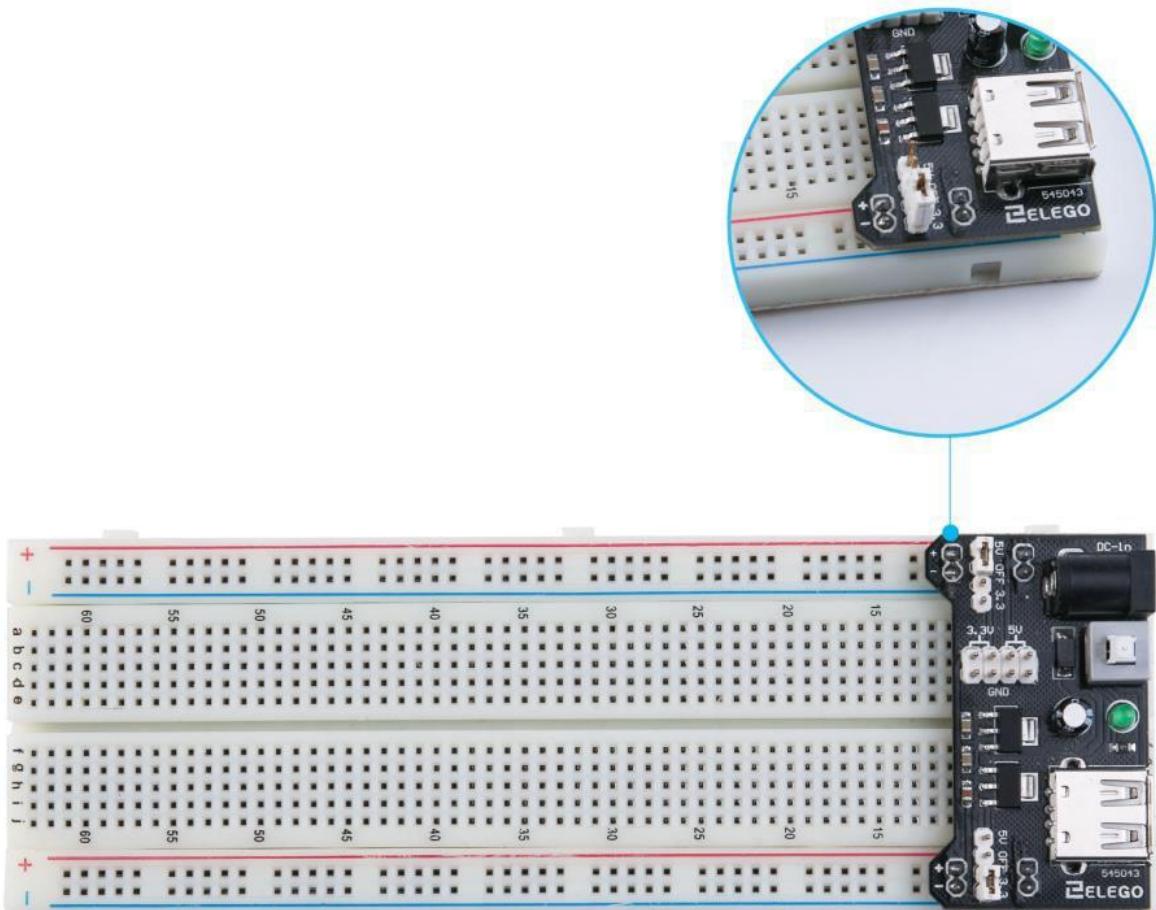
Produktspezifikationen:

- Ein- / Ausschalter
- LED-Betriebsanzeige
- Eingangsspannung: 6.5-9V (DC) über 5.5mm x 2.1mm Stecker
- Ausgangsspannung: 3,3V / 5v
- Maximaler Ausgangsstrom: 700 mA
- Unabhängiger Steuerausgang. 0v, 3.3v, 5v zum breadboard
- Ausgangspins für bequemen externen Gebrauch
- Größe: 2.1 in x 1.4 in
- USB-Geräte-Anschluss auf dem Bord zum Anschließen externer Geräte

Einstellen der Ausgangsspannung:



Der linke und rechte Spannungsausgang können unabhängig voneinander konfiguriert werden. Um die Ausgangsspannung auszuwählen, bewegen Sie den Jumper auf die entsprechenden Pins. Hinweis: Die Betriebsanzeige-LED und die Stromversorgung werden nicht eingeschaltet, wenn sich beide Jumper in der Position "OFF" befinden.



Wichtig:

Vergewissern Sie sich, dass Sie das Modul korrekt auf dem Steckbrett ausrichten. Der Negativstift (-) am Modul stimmt mit der blauen Linie (-) auf dem Steckbrett und der Pluspol (+) mit der roten Linie (+) überein. Wenn Sie dies nicht tun, kann es zu einem unbeabsichtigten vertauschen der Leistung kommen

L293D

Das ist ein sehr nützlicher Chip. Er kann tatsächlich zwei Motoren unabhängig steuern. Wir verwenden nur den halben Chip in dieser Lektion, die meisten Pins auf der rechten Seite des Chips sind für die Steuerung eines zweiten Motors.



Produktspezifikationen:

- Mit Unitrode L293 und L293D Produkte jetzt von Texas Instruments
- Breiter Versorgungsspannungsbereich: 4,5 V bis 36 V
- Separate Input-Logic Versorgung
- Interner ESD-Schutz
- Thermisches Abschalten
- High-Noise-Immunity Eingänge
- Funktionell ähnlich wie SGS L293 und SGS L293D
- Ausgangsstrom 1 A pro Kanal (600 mA für L293D)
- Spitzenausgangsstrom 2 A pro Kanal (1,2 A für L293D)
- Ausgang Klemmdioden für induktive Transient Unterdrückung (L293D)

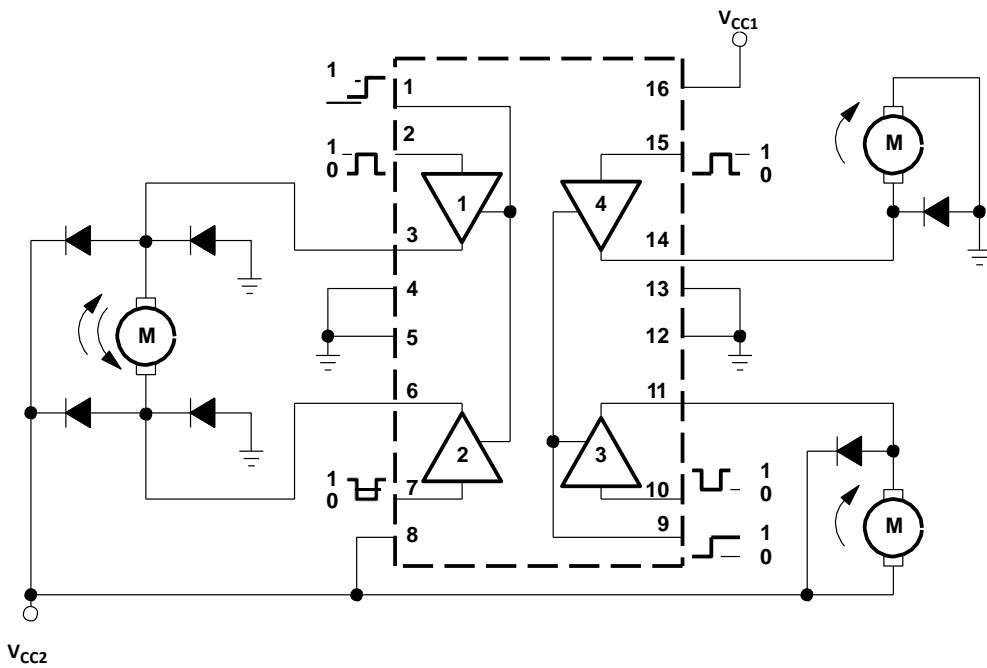


Beschreibung / Bestellinformationen

Die L293 und L293D sind vierfache Hochstrom-Halb-H-Treiber. Der L293 ist so konstruiert, dass er bidirektionale Antriebsströme von bis zu 1 A bei Spannungen von 4,5 V bis 36 V liefert. Der L293D ist so ausgelegt, dass er bidirektionale Antriebsströme von bis zu 600 mA bei Spannungen von 4,5 V bis 36 V liefert. Beide Geräte sind Entworfen um induktive Lasten wie Relais, Zylinderspulen, DC und Bipolar-Schrittmotoren sowie andere Hochstrom- / Hochspannungslasten in Anwendungen mit positiver Versorgung anzutreiben.

Alle Eingänge sind TTL-kompatibel. Jeder Ausgang ist eine komplette Totem-Pole-Schaltung, mit einer Darlington-Transistor-Senke und einer Pseudo-Darlington-Quelle. Die Treiber sind paarweise aktiviert, wobei die Treiber 1 und 2 mit 1,2EN und den Treibern 3 und 4 aktiviert wurden, die durch 3.4EN aktiviert wurden. Wenn ein Freigabeeingang hoch ist, werden die zugehörigen Treiber freigegeben und ihre Ausgänge sind aktiv und in Phase mit ihren Eingängen. Wenn der Freigabeeingang niedrig ist, sind diese Treiber deaktiviert und ihre Ausgänge sind aus und im hochohmigen Zustand. Mit den richtigen Dateneingängen bildet jedes Paar von Treibern einen Full-H (oder Brücken) reversiblen Antrieb, der für Zylinderspulen oder Motoranwendungen geeignet ist.

Blockdiagramm



Ich habe mich mit unentzifferbaren Pinbelegungsdiagrammen in Datenblättern verärgert, also habe ich mein eigenes entworfen, wo ich denke, dass es mehr relevante Informationen enthält.

Es sind 3 Drähte angeschlossen an dem Arduino, 2 Drähte an den Motor angeschlossen, und 1 Draht an eine Batterie angeschlossen.

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

Um diese Pinbelegung zu verwenden:

Die linke Seite befasst sich mit dem ersten Motor, die rechte Seite beschäftigt sich mit einem zweiten Motor.

Ja, du kannst es auch mit nur einem Motor laufenlassen.

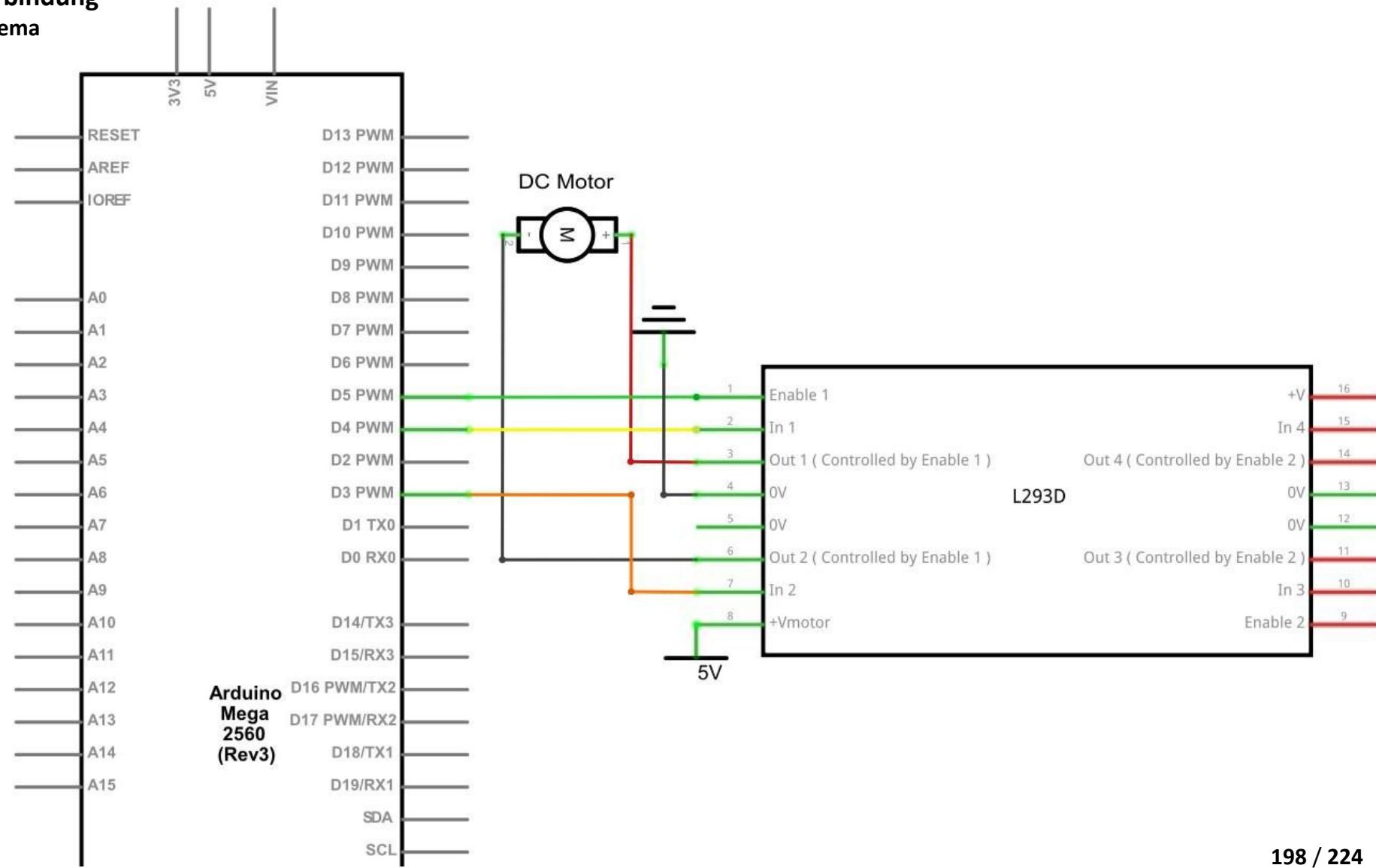
Arduino Verbindung

M1 PWM - Verbinden Sie diese mit einem PWM-Pin am Arduino. Sie sind auf dem 2560 beschriftet, Pin 5 zum Beispiel. Geben Sie eine ganze Zahl zwischen 0 und 255 aus, wobei 0 ausgeschaltet ist, 128 ist die halbe Geschwindigkeit und 255 ist die Höchstgeschwindigkeit.

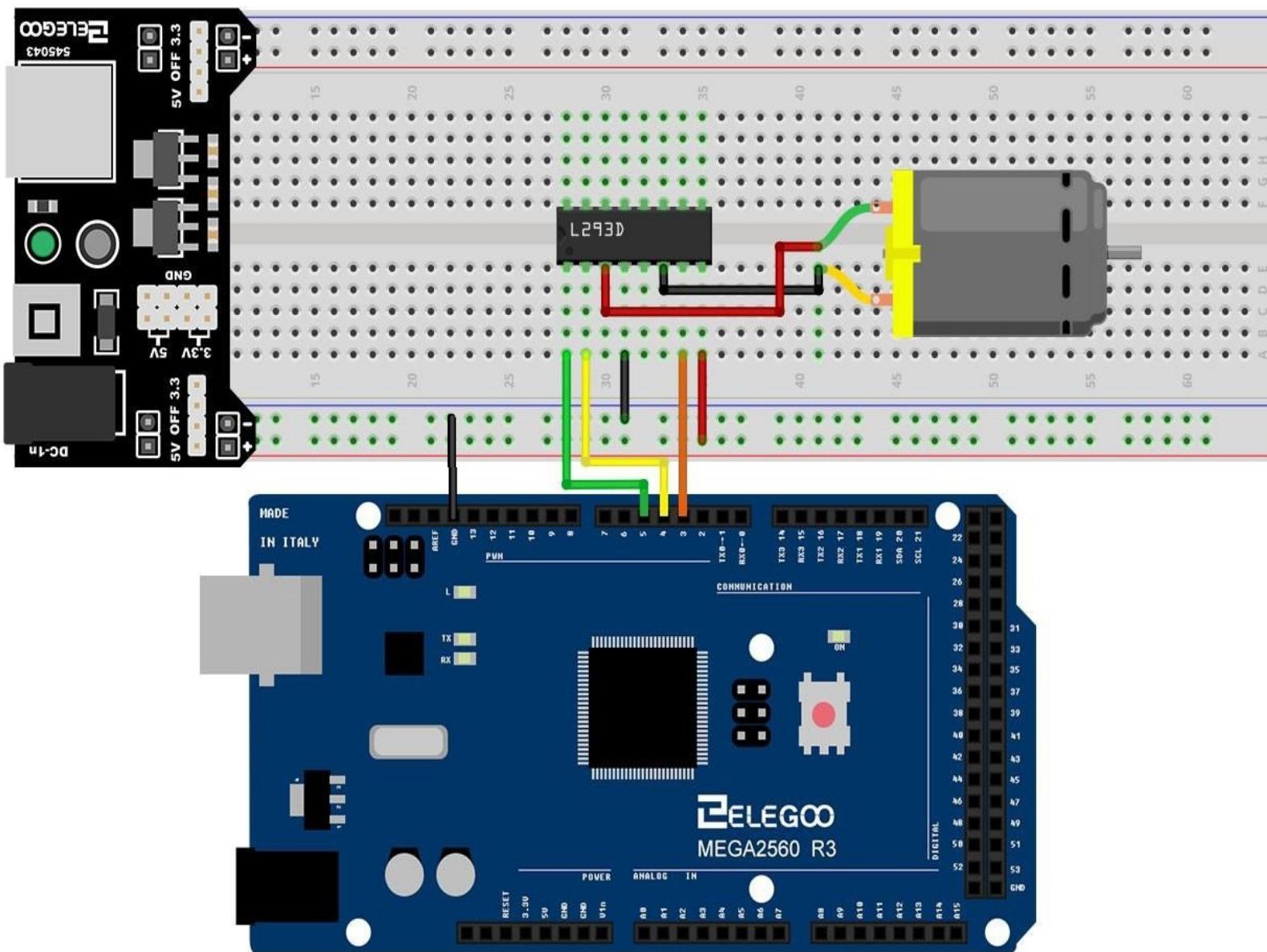
M1 Richtung 0/1 und M2 Richtung 1/0 - Verbinden Sie diese beiden mit zwei digitalen Arduino Pins. Geben Sie einen Pin als HIGH und den anderen Pin als LOW, und der Motor dreht sich in eine Richtung.

Umgekehrt wenn Sie die Ausgänge auf LOW und HIGH, wird der Motor sich in die andere Richtung drehen.

Verbindung Schema

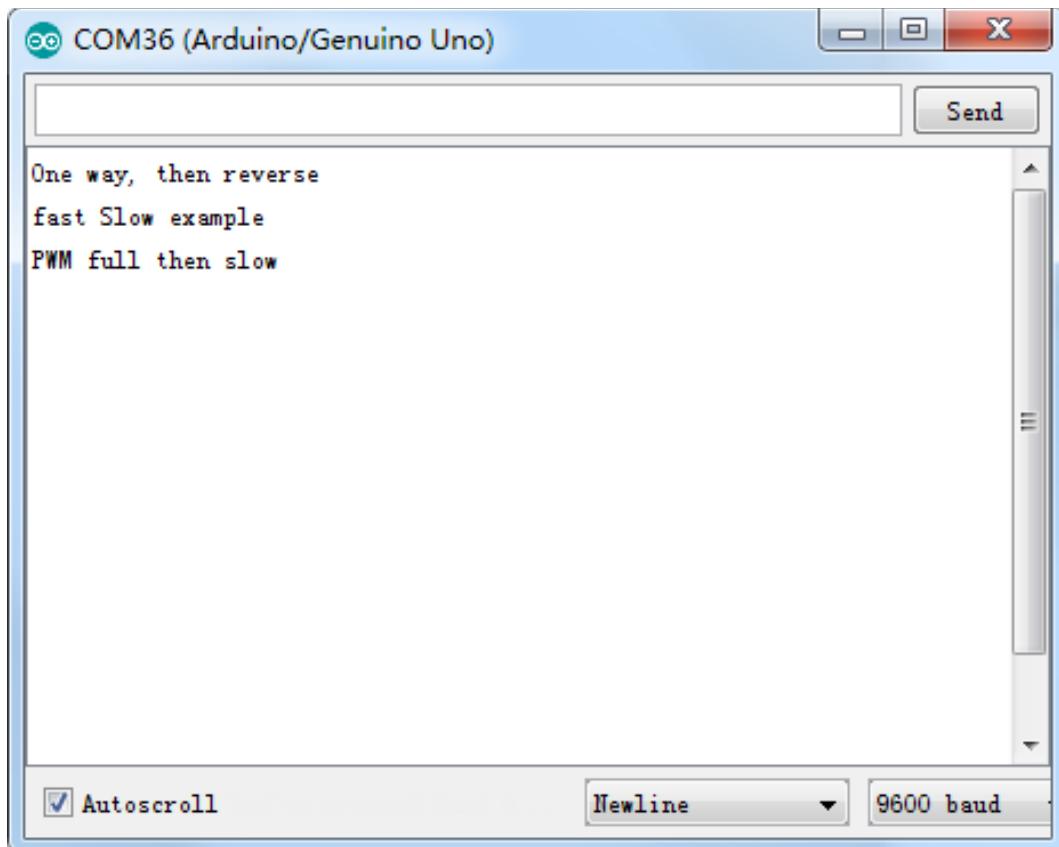


Schaltplan



Der untenstehende Code verwendet keine separate Stromversorgung (dh eine Batterie), sondern verwendet stattdessen die 5V Strom aus dem Arduino. Beachten Sie, dass dies riskant ist, ohne dass der L293D eskontrolliert!

Sie sollten einen Motor NIEMALS direkt an den Arduino anschließen, denn wenn man einen Motor ausschaltet, bekommt man ein elektrisches Feedback. Bereits mit einem kleinen Motor, wird dies Ihr Arduino beschädigen, und mit einem großen Motor, können Sie interessante Flammen und Funken sehen. LASS DEN SCHEIß!!



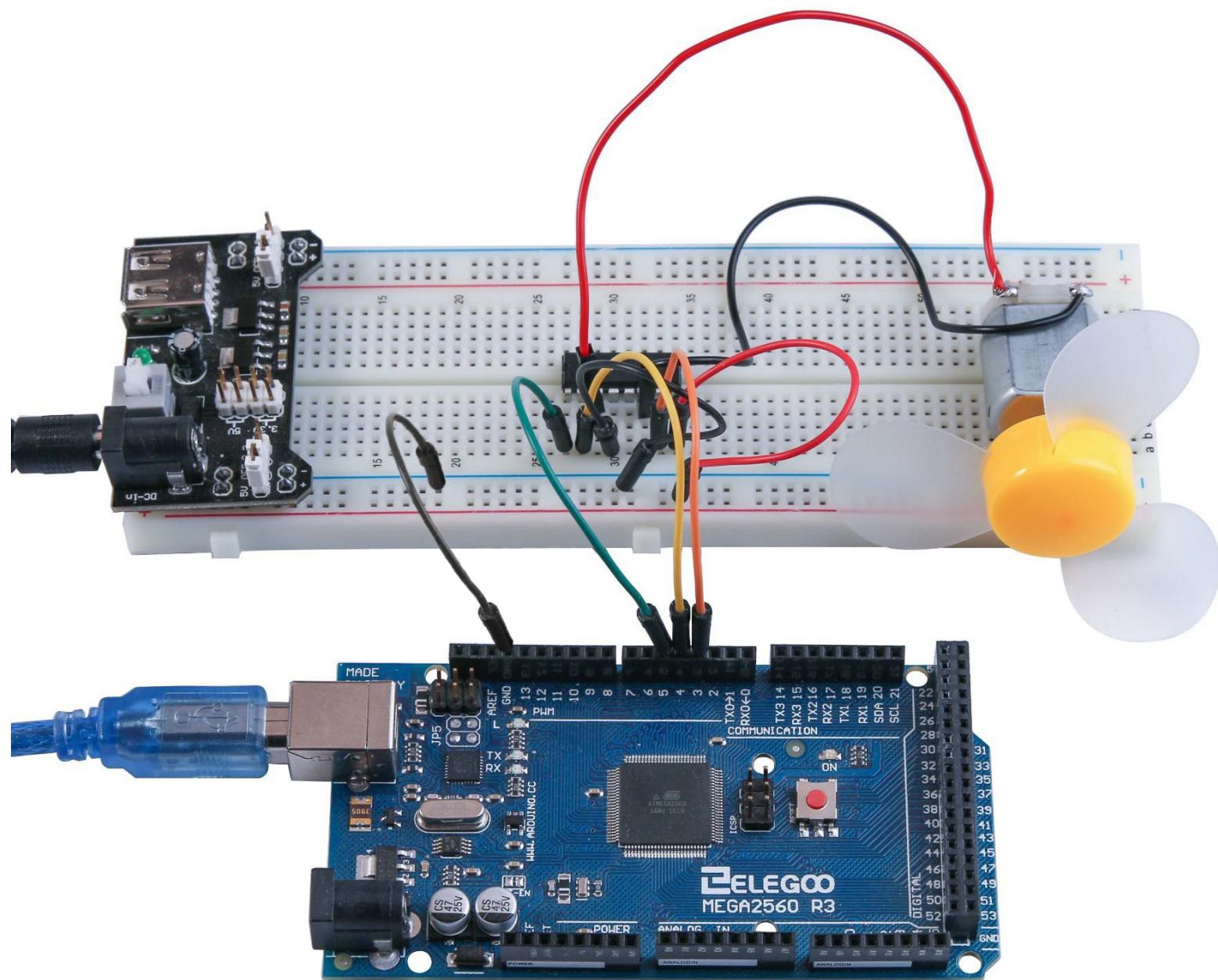
Code

Nach der Verschalten öffnen Sie bitte das Programm im Code-Ordner - Lektion 29 DC-Motoren und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Nach dem Laden des Programms schalten Sie alle Netzschalter ein. Der Motor dreht sich im Uhrzeigersinn und gegenseitig gegen den Uhrzeigersinn. Dann wird es weiter schneller im Uhrzeigersinn drehen. Nach einer kurzen Pause wird es schnell gegen den Uhrzeigersinn gedreht. Dann sendet die Anschaltbaugruppe PWM-Signale um den Motor anzutreiben, der Motor wird langsam seine maximale Drehzahl auf das Minimum reduziert und auf das Maximum wieder erhöht. Schließlich kommt es für

10s zu einem Stillstand, bis der nächste Zyklus beginnt.

Beispielfoto



Lektion 30 Relay

Überblick

In dieser Lektion lernen Sie, wie man ein Relais benutzt.

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x 830 tie-points breadboard
- (1) x Propeller und 3-6v Motor
- (1) x L293D IC
- (1) x 5v Relay
- (1) x Netzteilmodul
- (1) x 9V1A Adapter
- (8) x M-M Kabel (Male to Male jumper Kabel)



Komponenten Einführung

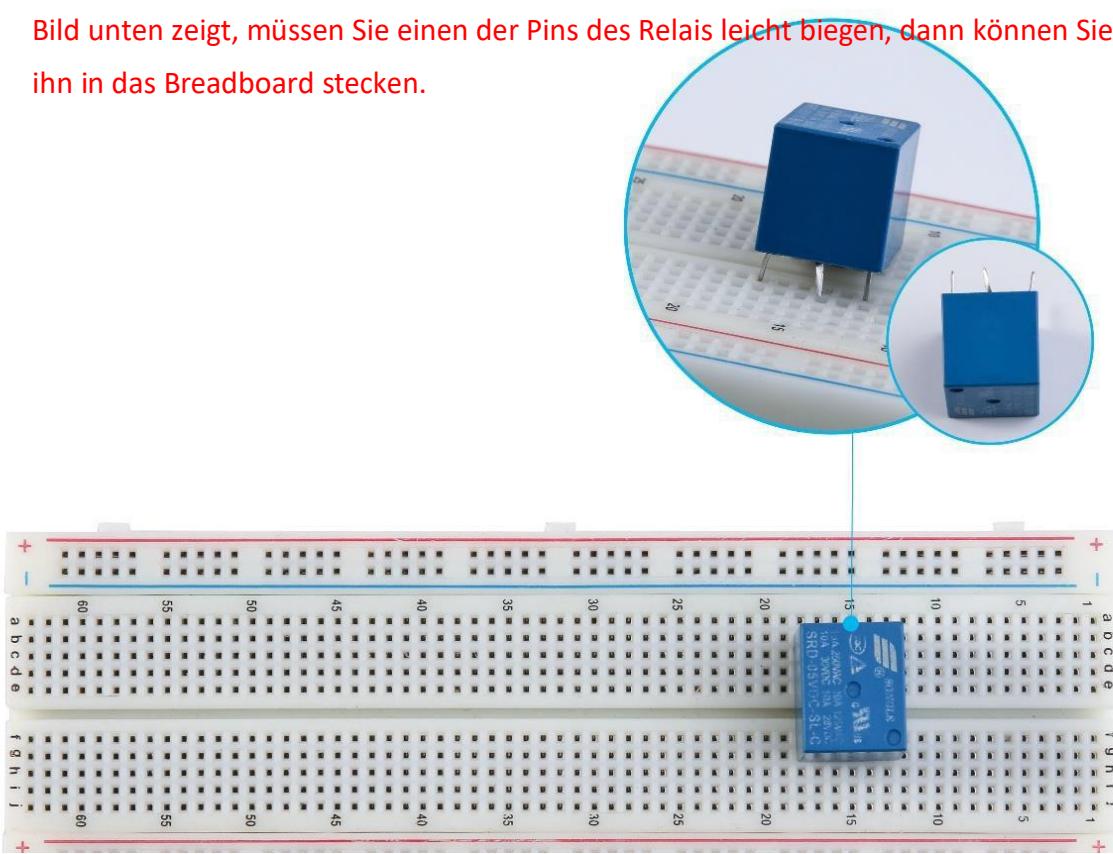
Relay:

Ein Relais ist ein elektrisch betriebener Schalter. Viele Relais verwenden einen Elektromagneten, um einen Schalter mechanisch zu betätigen, aber andere Bedienprinzipien werden auch in Festkörperrelais verwendet. Es werden Relais verwendet, bei denen eine Schaltung durch ein Low-Power-Signal (mit vollständiger galvanischer Trennung zwischen Steuer- und Regelkreisen) gesteuert werden muss oder mehrere Schaltkreise von einem Signal gesteuert werden müssen. Die ersten Relais wurden in Langstrecken-Telegraphenschaltungen als Verstärker verwendet. Sie wiederholten das Signal, das von einem Kreislauf hereinkam und es auf einer anderen Schaltung wiederübertragen. Relais wurden weitgehend in Telefonvermittlungen und frühen Computern verwendet, um logische Operationen durchzuführen.

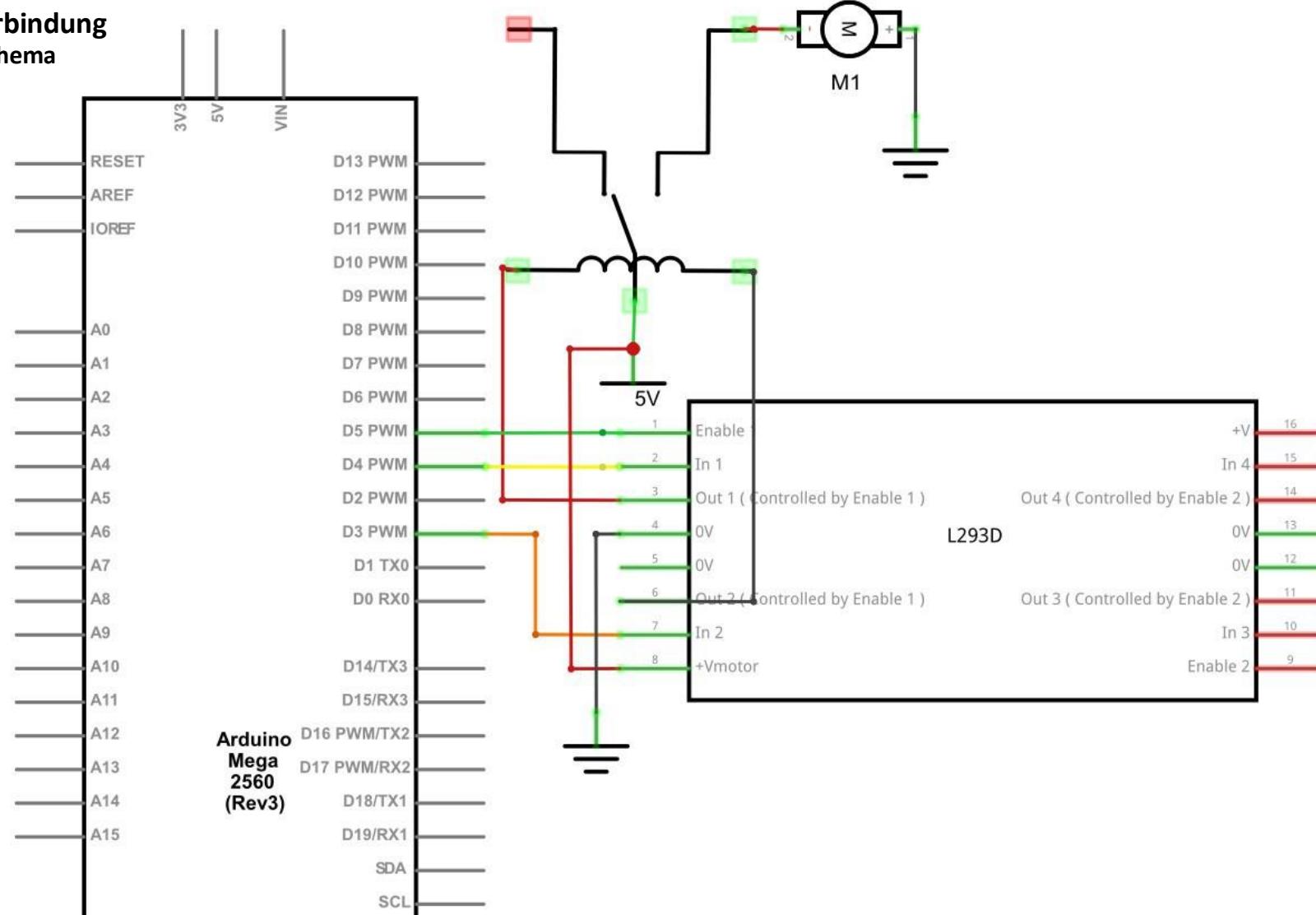
Eine Art von Relais, das die hohe Leistung verarbeiten kann, die erforderlich ist, um einen Elektromotor oder andere Lasten direkt zu steuern, wird als Schaltvorrichtung oder Schütz bezeichnet. Halbleiterrelais steuern Leistungsschaltkreise ohne bewegliche Teile, stattdessen mit einem Halbleiterbauelement, um die Umschaltung durchzuführen. Relais mit kalibrierten Betriebsmerkmalen und manchmal mehrere Betriebsspulen dienen zum Schutz von Stromkreisen vor Überlast oder Störungen. In modernen elektrischen Energiesystemen werden diese Funktionen von digitalen Instrumenten mit dem Namen "Schutzrelais" durchgeführt.

Unten ist das Schema, wie man Relais mit einem Arduino betreibt

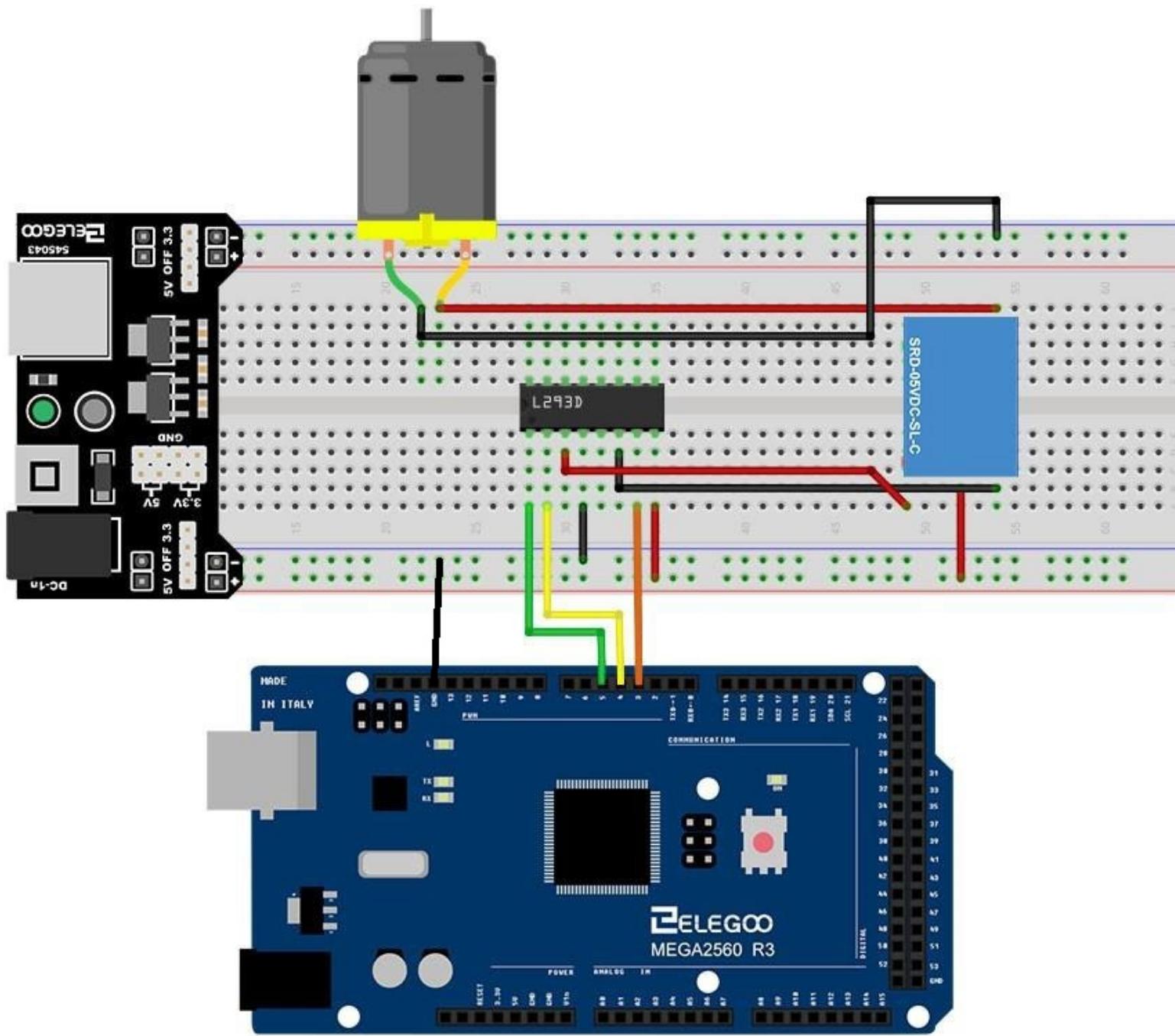
Sie können verwirrt sein, wie man das Relais in das Breadboard einsetzt. Wie das Bild unten zeigt, müssen Sie einen der Pins des Relais leicht biegen, dann können Sie ihn in das Breadboard stecken.



Verbindung Schema



Schaltprinzip

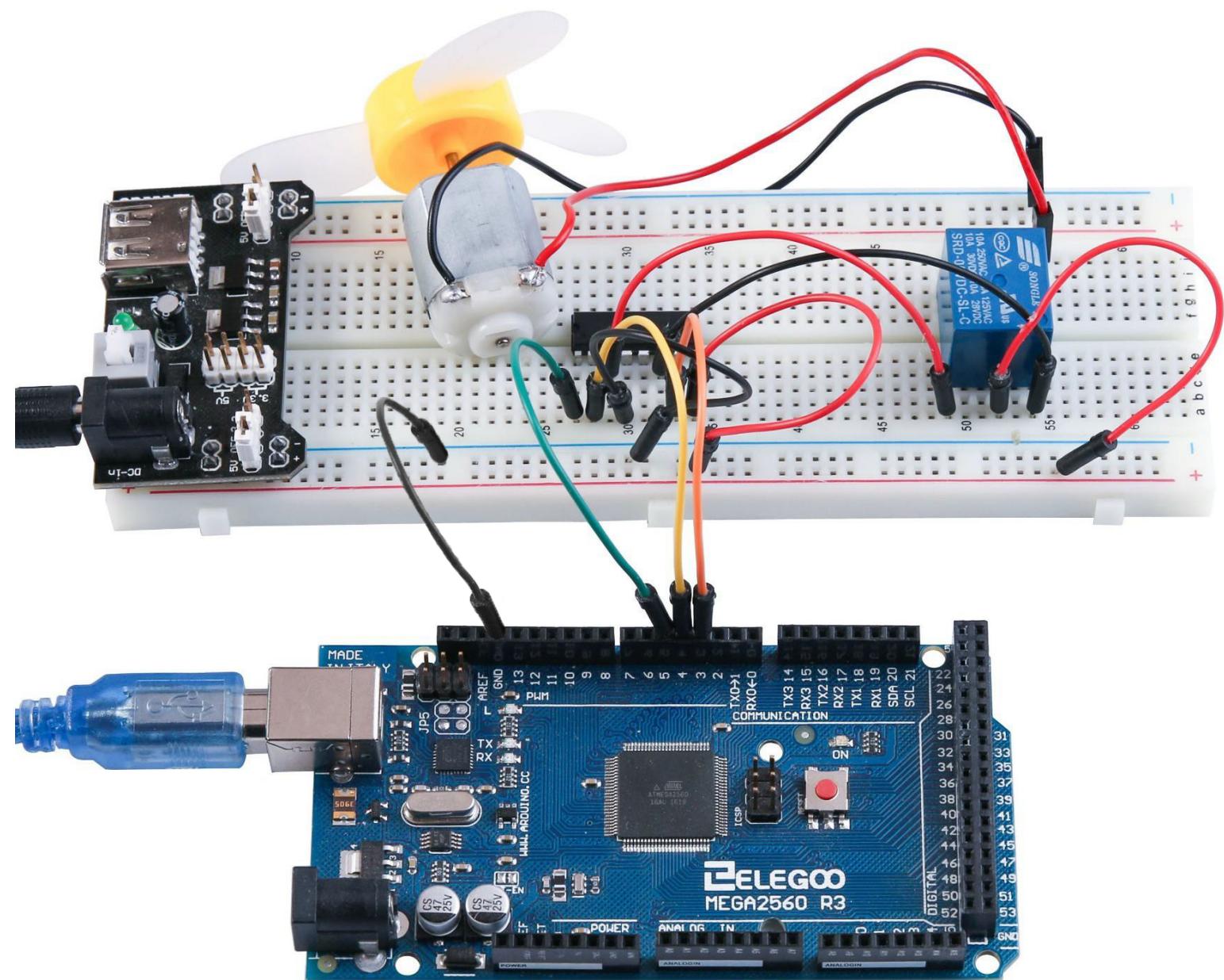


Code

Nach der Verschaltung öffnen Sie bitte das Programm im Codeordner - Lektion 30 Relais und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Nach dem Laden des Programms schalten Sie alle Netzschalter ein. Das Relais wird mit einem Klingelton starten. Dann dreht sich der Motor. Nach einer gewissen Zeit wird das Relais freigegeben und der Motor bleibt stehen.

Beispielfoto



Lektion 31 Schrittmotor

Überblick

In dieser Lektion lernen Sie eine lustige und einfache Art, einen Schrittmotor zu betreiben.

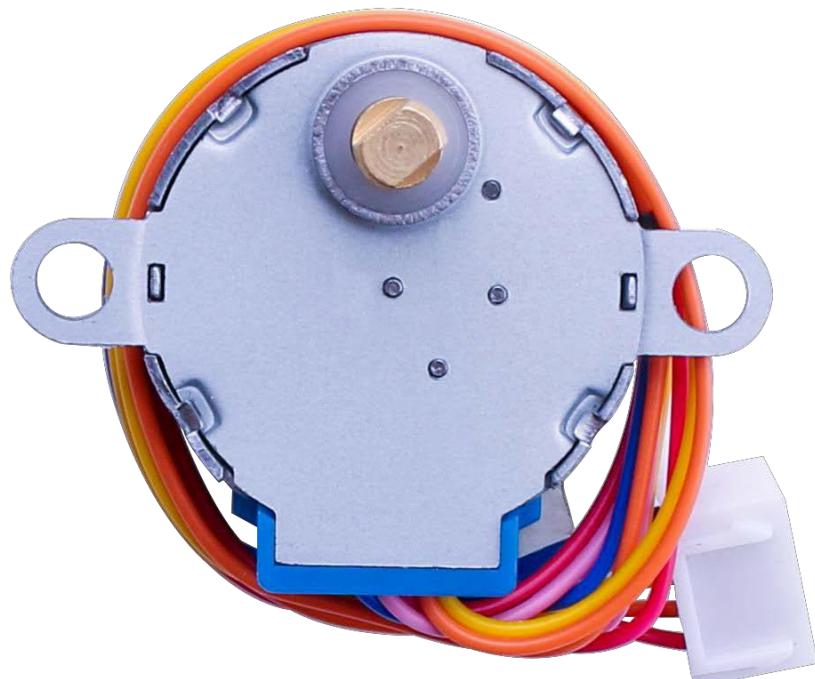
Der Stepper, den wir benutzen, kommt mit einem eigenen Treiberboard, der es einfach macht, mit unserem MEGA2560 zu verbinden.

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x 830 tie-points breadboard
- (1) x ULN2003 Schrittmotor-Treiber-Modul
- (1) x Schrittmotor
- (1) x 9V1A Adapter
- (1) x Netzteilmodul
- (6) x F-M Kabel (Female to Male DuPont Kabel)
- (1) x M-M Kabel (Male to Male jumper Kabel)

Bauteil Einführung

Schrittmotor



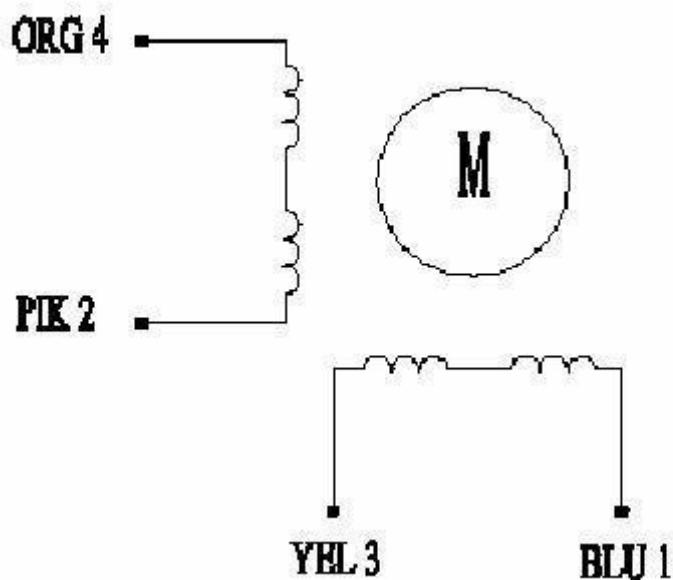
Ein Schrittmotor ist ein elektromechanisches Gerät, das elektrische Impulse in mechanische Bewegungen umwandelt. Die Welle oder Spindel eines Schrittmotors dreht sich in diskreten Schrittschritten wenn elektrische Befehlsimpulse in der richtigen Reihenfolge an sie angelegt werden. Die Rotation der Motoren hat mehrere direkte Beziehungen zu diesen angelegten Eingangsimpulsen. Die Reihenfolge der angelegten Impulse steht in direktem Zusammenhang mit der Drehrichtung der Motorwelle. Die Drehzahl der Motorwellen dreht sich direkt auf die Frequenz der Eingangsimpulse und die Rotationslänge steht in direktem Zusammenhang mit der Anzahl der angelegten Eingangsimpulse. Einer der bedeutendsten Vorteile eines Schrittmotors ist seine Fähigkeit, in einem offenen Loop-System genau gesteuert zu werden. Open-Loop-Steuerung bedeutet, dass keine Rückmeldung über Position benötigt wird. Diese Art von Steuerung eliminiert die Notwendigkeit von teuren Sensor- und Rückkopplungsvorrichtungen, wie z. B. optischen Encodern. Ihre Position ist einfach bekannt, indem man die Eingangsstufenimpulseverfolgt.

Schrittmotor 28BYJ-48 Parameter

- Modell: 28BYJ-48
- Nennspannung: 5VDC
- Anzahl der Phasen: 4
- Geschwindigkeitsänderungsverhältnis: 1/64
- Schrittinkel: 5.625 ° / 64
- Frequenz: 100Hz
- DC-Widerstand: $50 \Omega \pm 7\%$ (25 °C)
- Leerlauf-Intraktionsfrequenz:> 600Hz
- Leerlauf-Auszugsfrequenz:> 1000Hz
- Intraktion Drehmoment> 34,3mN.m (120Hz)
- Selbstpositionierendes Drehmoment> 34.3mN.m
- Reibmoment: 600-1200 gf.cm
- Drehmoment ziehen: 300 gf.cm
- Isolierter Widerstand> 10MΩ (500V)
- Isolierte Stromleistung: 600VAC / 1mA / 1s
- Isolationsgrad: A
- Temperaturanstieg <40K (120Hz)
- Geräusche <35dB (120Hz, keine Last, 10cm)

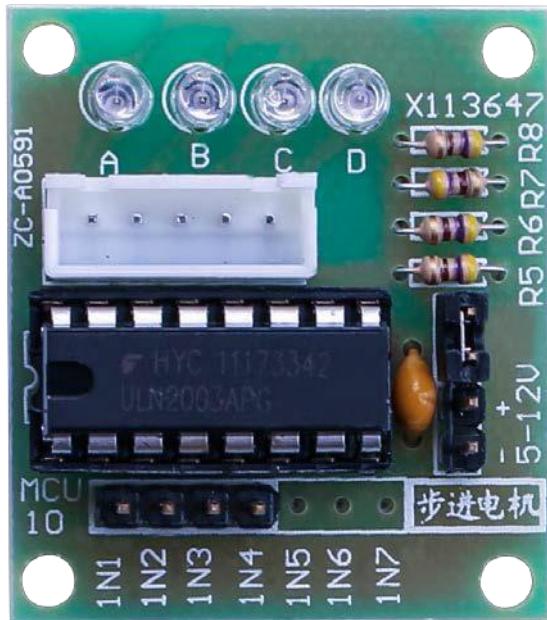
Schnittstellen Schaltkreise

WIRING DIAGRAM



Der bipolare Schrittmotor hat normalerweise vier Drähte, die aus ihm herauskommen. Im Gegensatz zu unipolaren Steppern haben bipolare Stepper keine gemeinsame Mittenverbindung. Sie haben stattdessen zwei unabhängige Sätze von Spulen. Sie können sie von unipolaren Steppern unterscheiden, indem Sie den Widerstand zwischen den Drähten messen. Sie sollten zwei Paare von Drähten mit gleichem Widerstand finden. Wenn Sie die Leads Ihres Messgerätes an zwei Drähten angeschlossen haben, die nicht angeschlossen sind (d. H. Nicht an dieselbe Spule angeschlossen), sollten Sie einen unendlichen Widerstand (oder keine Kontinuität) sehen.

ULN2003 Treiber Board



Produktbeschreibung

- Größe: 42mmx30mm
- ULN2003 Treiberchip verwenden, 500mA
- A. B. C. D LED, die den Vierphasen-Schrittmotor-Betriebszustand anzeigen.
- Weiße Buchse ist die Vier-Phasen-Schrittmotor-Standard-Buchse.
- Netzpins sind getrennt
- Wir haben die restlichen Pins des ULN2003-Chips für Ihr weiteres Prototyping freigehalten.

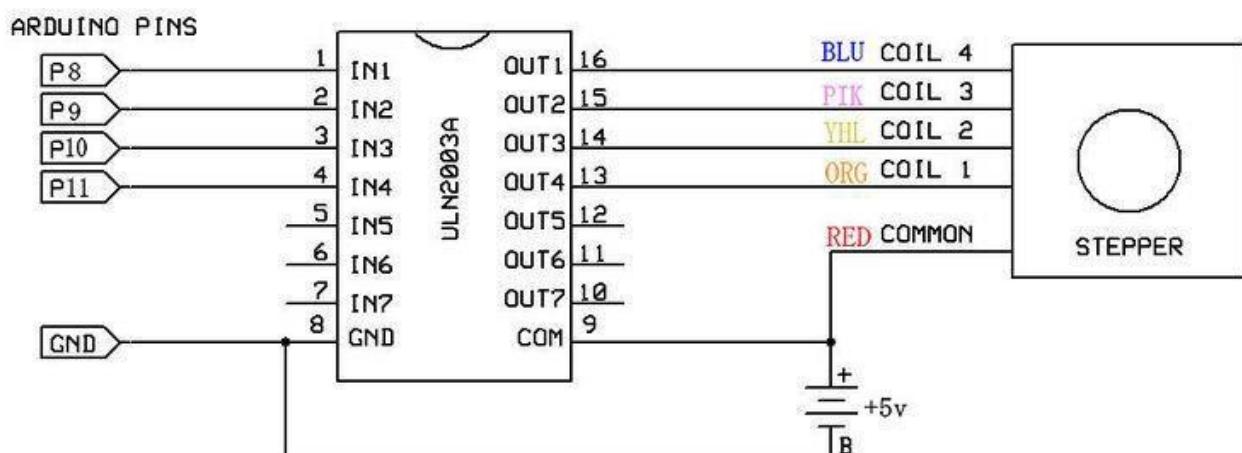
Der einfachste Weg, einen unipolaren Stepper am Arduino anzuschließen, besteht darin, einen Ausbruch für den ULN2003A-Transistor-Array-Chip zu verwenden. Die ULN2003A enthält sieben Darlington Transistor Treiber und ist etwa wie mitsieben TIP120 Transistoren alle in einem Paket. Der ULN2003A kann bis zu 500 mA pro Kanal schalten und hat einen internen Spannungsabfall von ca. 1V, wenn er

eingeschaltet ist. Er enthält auch interne Klemmdiode, um Spannungsspitzen zu zerstreuen wenn induktive Lasten angetrieben werden. Um den Stepper zu steuern, wenden Sie die Spannung an jede der Spulen in einer bestimmten Reihenfolge an.

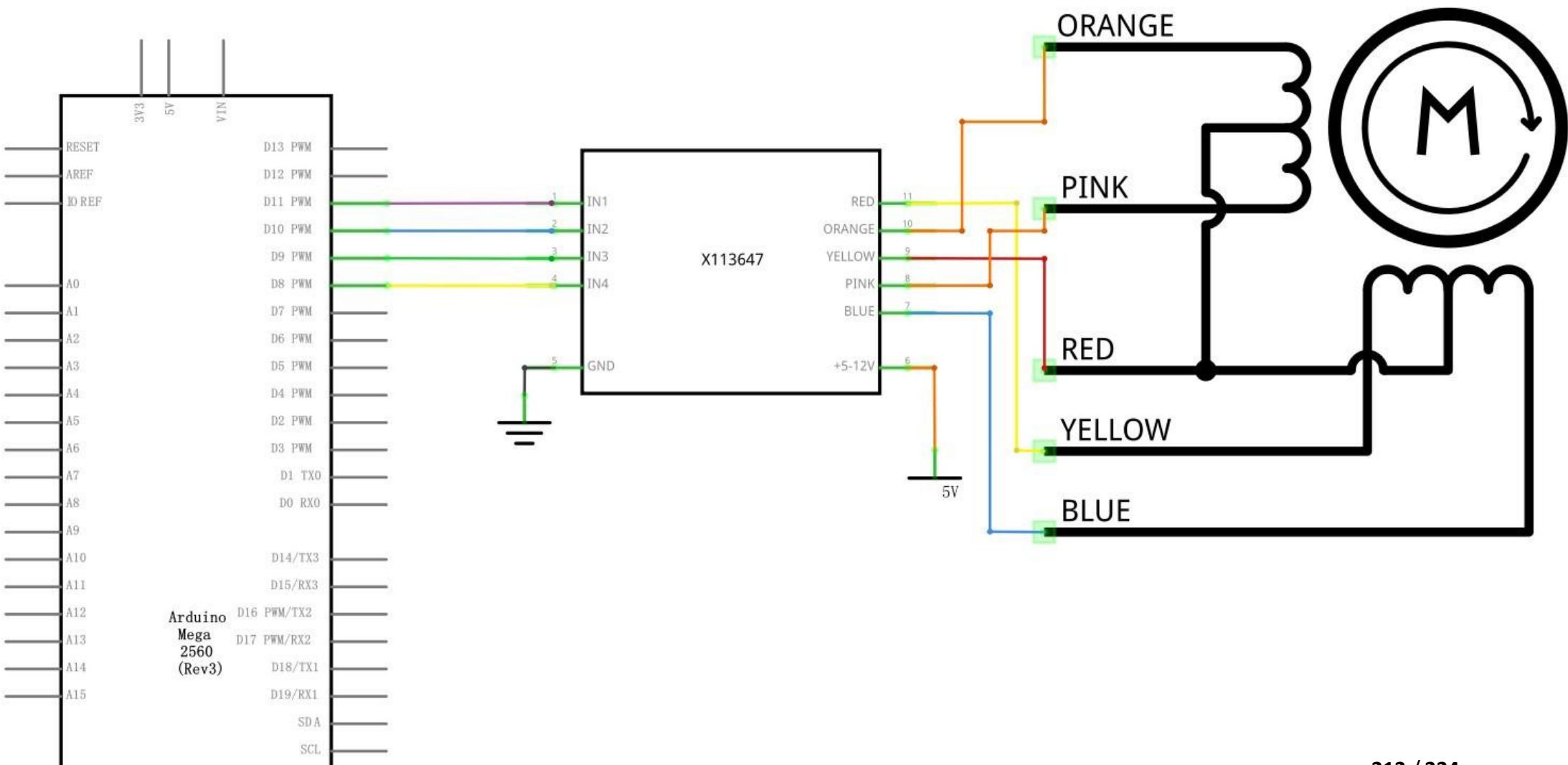
Die Sequenz würde so funktionieren:

| Lead Wire Color | ---> CW Direction (1-2 Phase) | | | | | | | |
|--------------------|-------------------------------|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 4 ORG | - | - | | | | | | - |
| 3 YEL | | - | - | - | | | | |
| 2 PIK | | | | - | - | - | | |
| 1 BLU | | | | | | - | - | - |

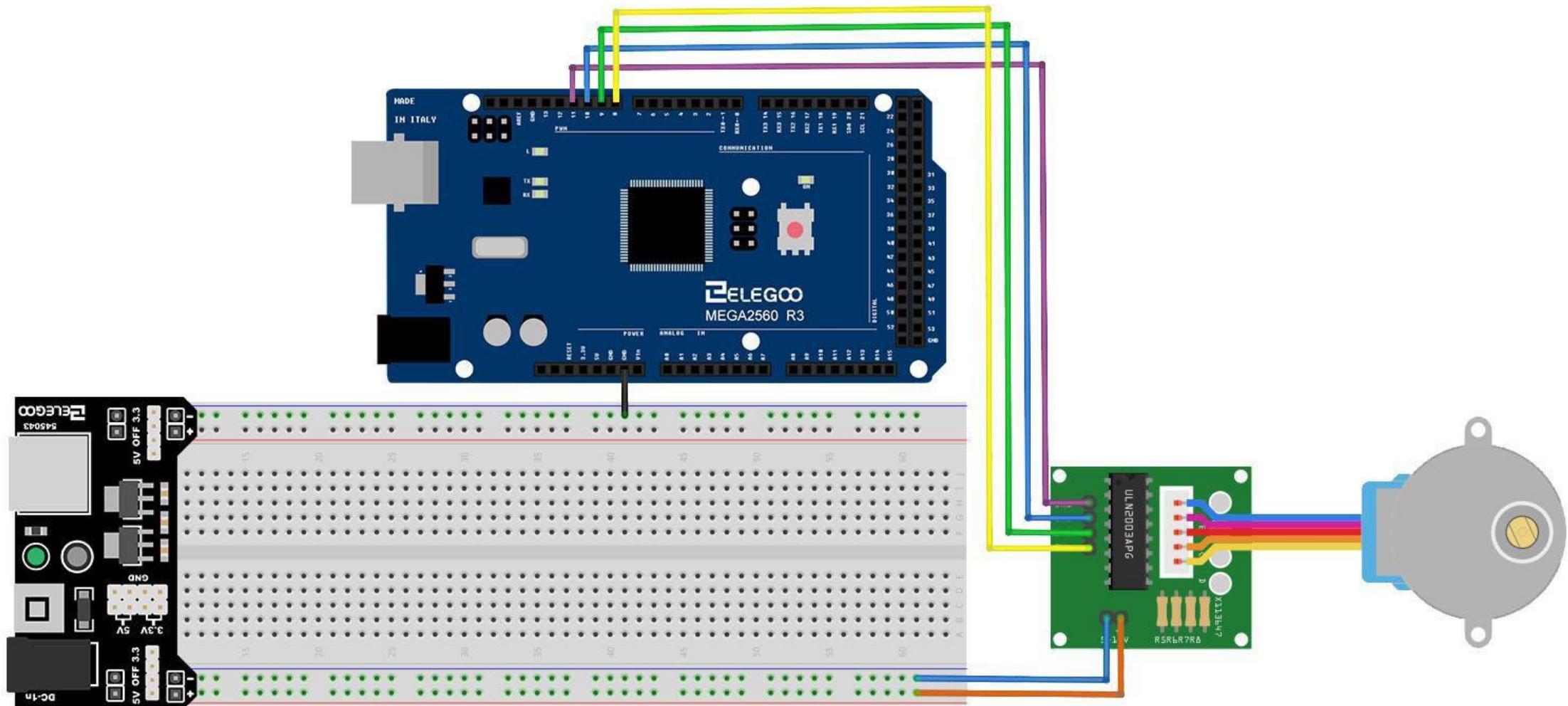
Hier sind Schemata, die zeigen, wie man einen unipolaren Schrittmotor zu vier Steuerpultnetzen unter Verwendung eines ULN2003A anschließt und zeigt, wie man mit vier com verbindet



Verbindun Schema



Schaltplan



Wir benutzen 4 Pins, um den Stepper zukontrollieren.

Pin 8-11 steuern den Schrittmotor.

Wir verbinden die Masse von MEGA2560 zum Schrittmotor.

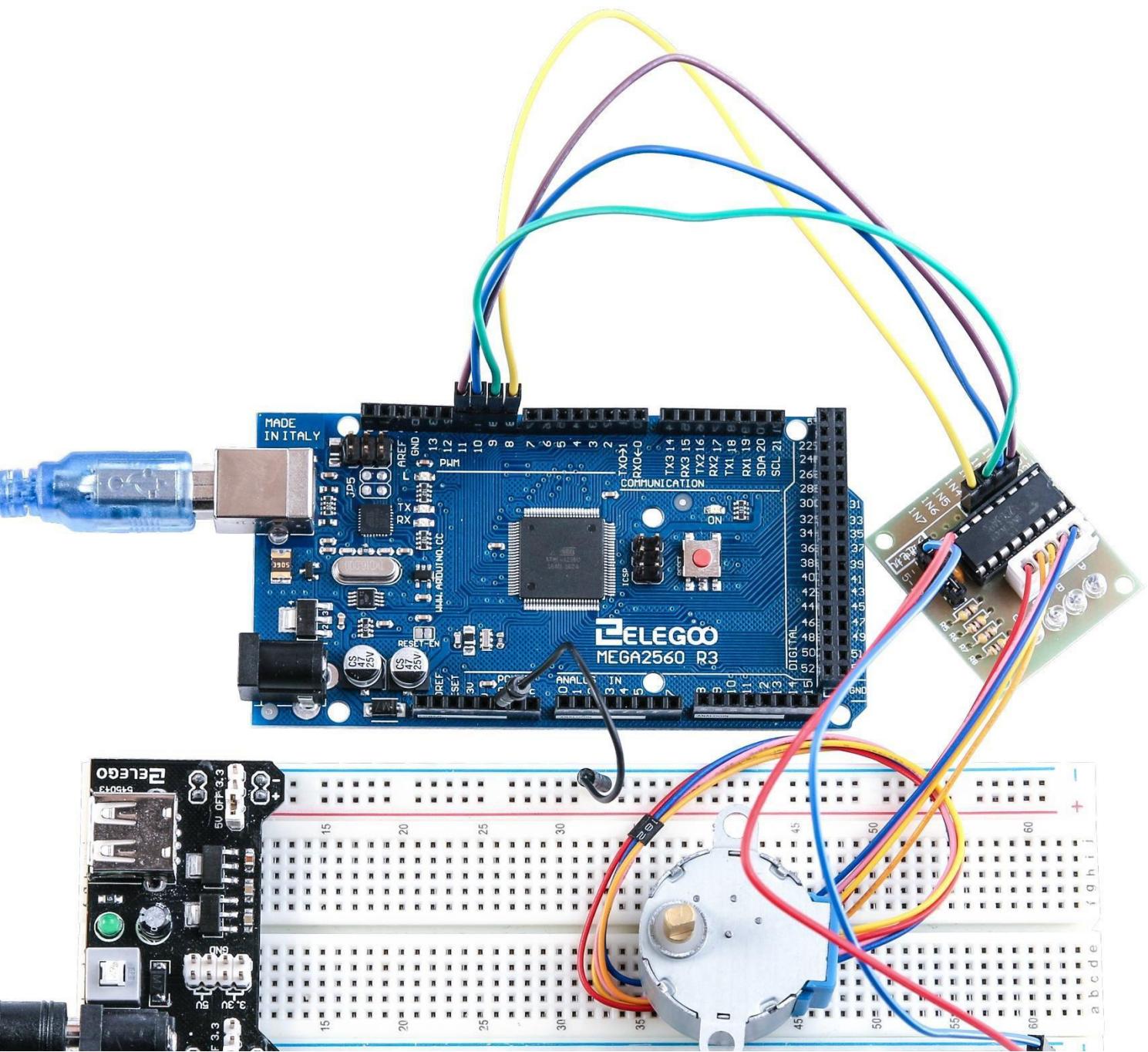
Code

Nach der Verdrahtung öffnen Sie bitte das Programm im Code-Ordner - Lektion 31 Schrittmotor und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die <Stepper> - Bibliothek installiert oder ggf. neu installiert haben. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion1.

Beispielfoto



Lektion 32 Steuern des Schrittmotors mit Fernbedienung

Überblick

In dieser Lektion lernen Sie eine lustige und einfache Möglichkeit, einen Schrittmotor aus der Ferne mit einer IR-Fernbedienung zusteuern.

Der Stepper, den wir benutzen, kommt mit einem eigenen Treibermodul, der es einfach macht, mit unserem MEGA2560 zu verbinden.

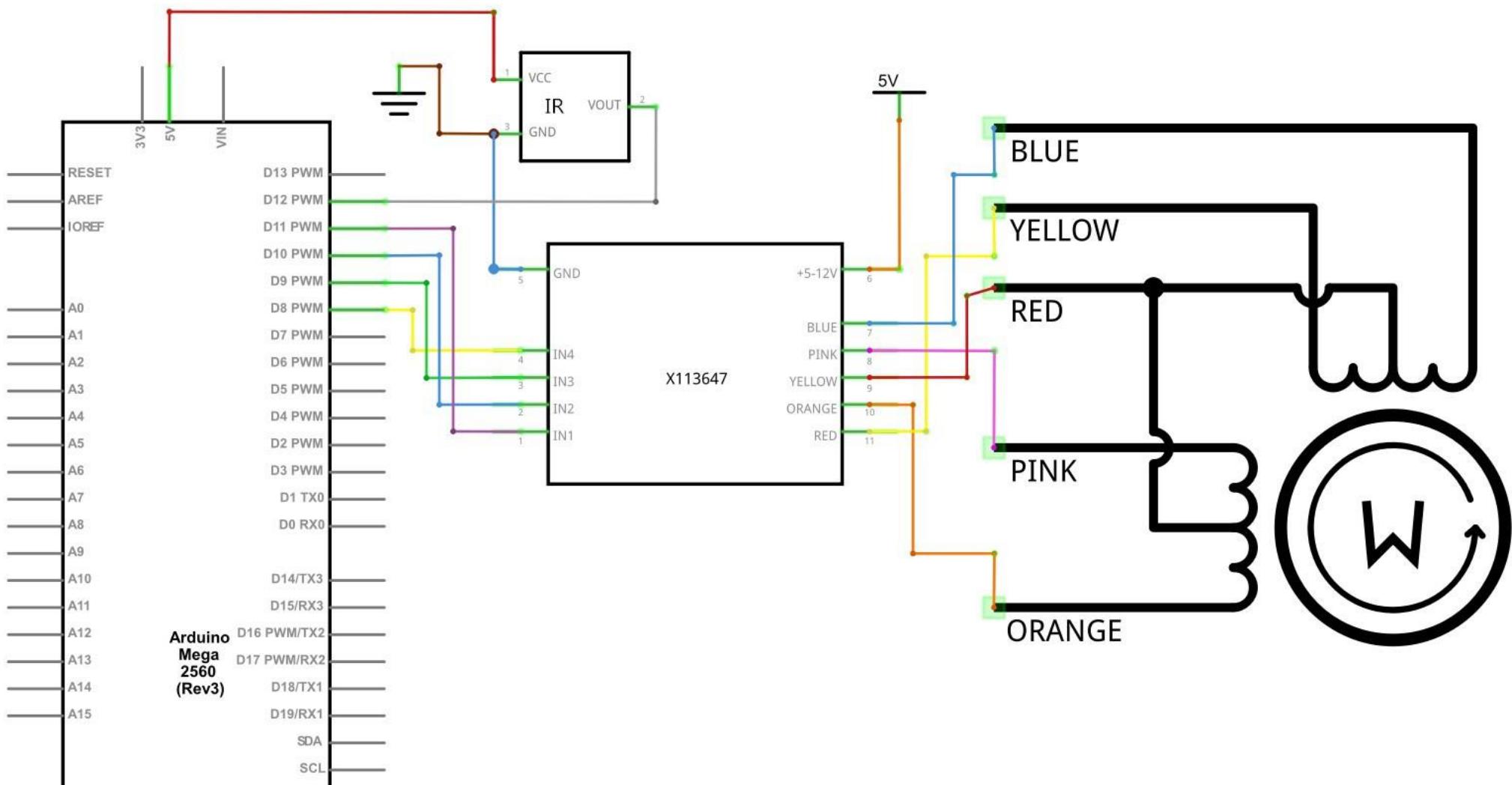
Da wir den Motor nicht direkt vom MEGA2560 betreiben wollen, werden wir eine preiswerte kleine "Steckdose" verwenden, die direkt in unserer Stromversorgung steckt und mit einem 9V 1Amp Netzteil versorgt wird.

Der IR-Sensor ist direkt an dem MEGA2560 angeschlossen, da er fast keine Stromversorgung benötigt.

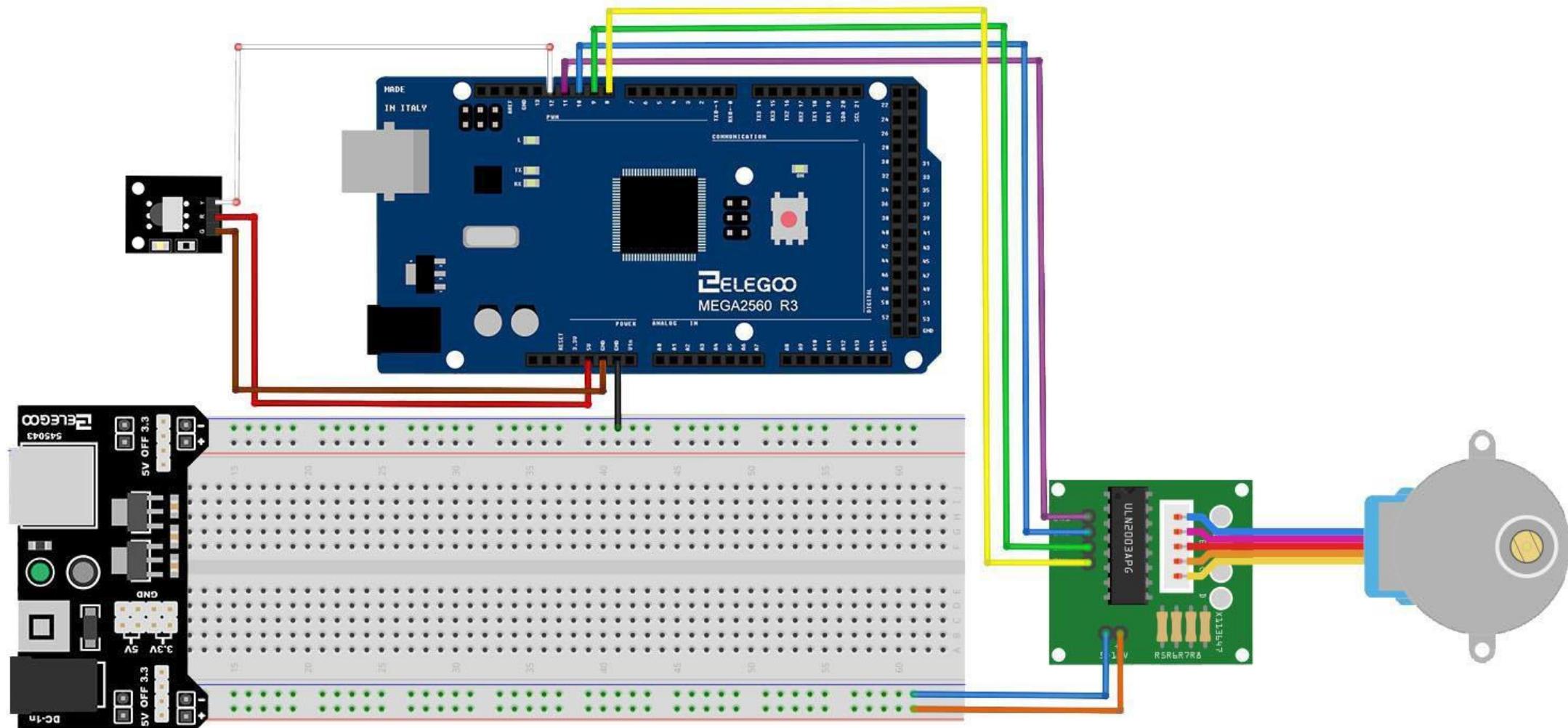
Erforderliche Komponente:

- (1) x Elegoo Mega2560 R3
- (1) x 830 tie-points breadboard
- (1) x IR Empfängermodul
- (1) x IR Fernbedienung
- (1) x ULN2003 Schrittmotor-Treiber-Modul
- (1) x Schrittmotor
- (1) x Netzteilmodul
- (1) x 9V1A Adapter
- (9) x F-M Kabel (Female to Male DuPont Kabel)
- (1) x M-M Kabel (Male to Male jumper Kabel)

Verbindung Schema



Schaltplan



Wir verwenden 4 Pins um den Schrittmotor und 1 Pin für den IR-Sensor zu steuern. Die Pins 8-11 steuern den Schrittmotor und der Stift 12 empfängt die IR-Information. Wir verbinden das 5V und Ground vom MEGA2560 zum Sensor. Als Vorsichtsmaßnahme verwenden Sie eine externe Stromversorgung, um den Schrittmotor zu versorgen, da er mehr Leistung nutzen kann und wir die Stromversorgung des MEGA2560 nicht beschädigen wollen.

Code

Nach der Verdrahtung öffnen Sie bitte das Programm im Code-Ordner - Lektion 32 Steuern des Schrittmotors mit Fernbedienung und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die <IRremote> installiert haben

<Stepper> Bibliothek oder ggf. neu installieren. Andernfalls wird Ihr Code nicht funktionieren.

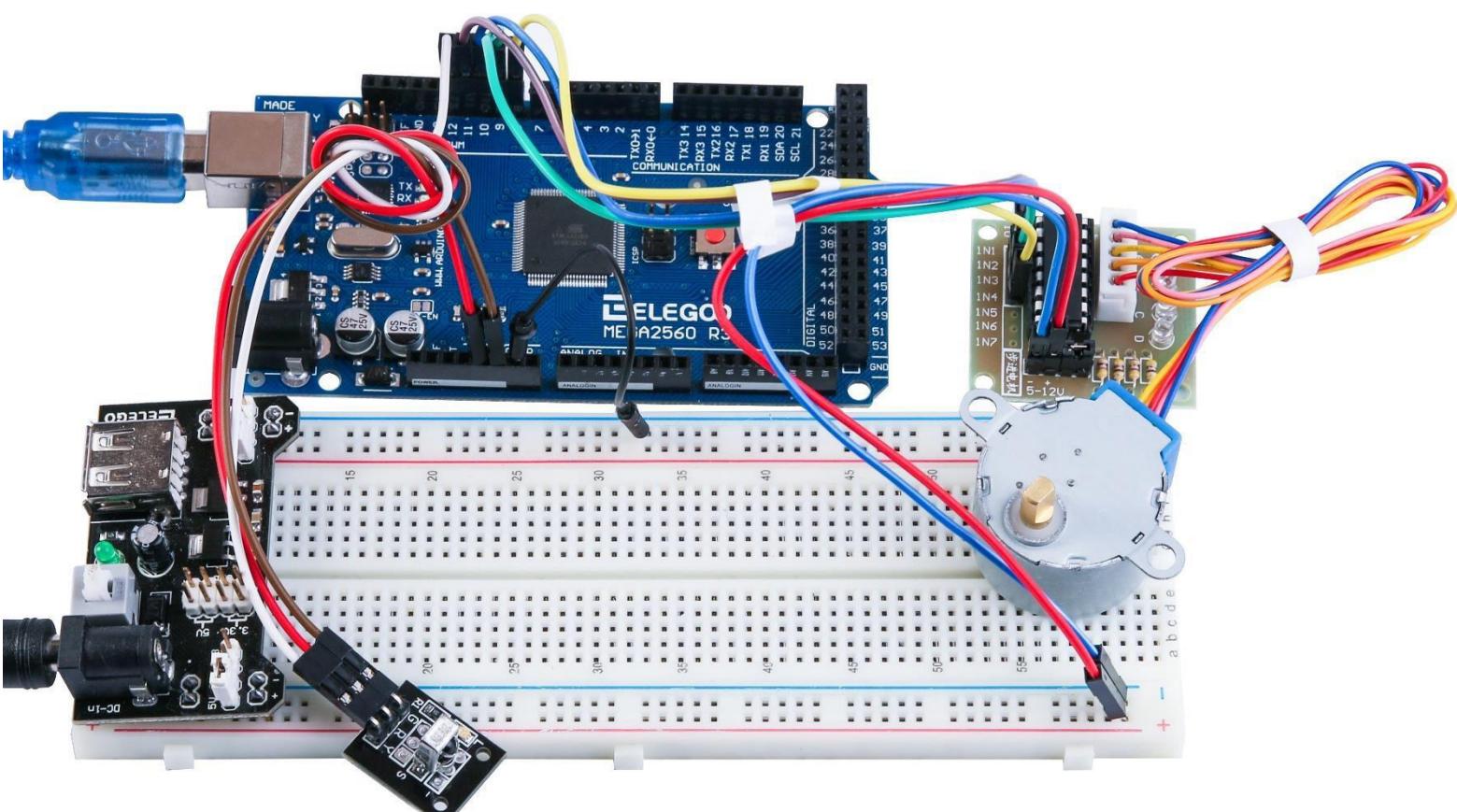
Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter Lektion 1.

Der Code erkennt nur 2 Werte von der IR-Fernbedienung: VOL + und VOL-.

Wenn VOL + auf die Fernbedienung gedrückt wird, macht der Motor eine volle Drehung im Uhrzeigersinn.

VOL- wird eine volle Drehung gegen den Uhrzeigersinn machen.

Beispelfoto



Lektion 33 Steuerung des Schrittmotors mit Drehgeber

Überblick

In dieser Lektion lernen Sie, wie man Schrittmotoren mit einem Drehgebersteuert.

Wir verwenden den preiswerten und beliebten Schrittmotor, der mit einer eigenen Steuerplatine auskommt: der 28BYJ-48 Schrittmotor mit der ULN2003Platte.

Der 28BYJ-48-Motor ist nicht sehr schnell oder sehr stark, aber er ist ideal für Anfänger, die beginnen mit der Steuerung eines Schrittmotors mit einem Arduino zu experimentieren.

Wir schreiben einen Code, um den Motor in die Richtung zu bewegen, in die wir den Drehgeber drehen, und wird auch verfolgen, wie viele Schritte wir gemacht haben, damit wir den Motor wieder in die Ausgangsposition bringen können, indem wir ihnden Drehgeberschalter nach unten drücken.

Erforderliche Komponenten:

- (1) x Elegoo Mega2560 R3
- (1) x 830 tie-points breadboard
- (1) x Drehgebermodul
- (1) x ULN2003 Schrittmotor-Treiber-Modul
- (1) x Schrittmotor
- (1) x Netzteilmodul
- (1) x 9V1A Adapter
- (9) x F-M Kabel (Female to Male DuPont Kabel)
- (1) x M-M Kabel (Male to Male jumper Kabel)

Komponenten Einführung

Drehcodierer

Ein Drehgeber, der auch als shaft-encoder bezeichnet wird, ist eine elektromechanische Vorrichtung, die die Winkelposition oder Bewegung einer Welle oder Achse in einen analogen oder digitalen Code umwandelt.

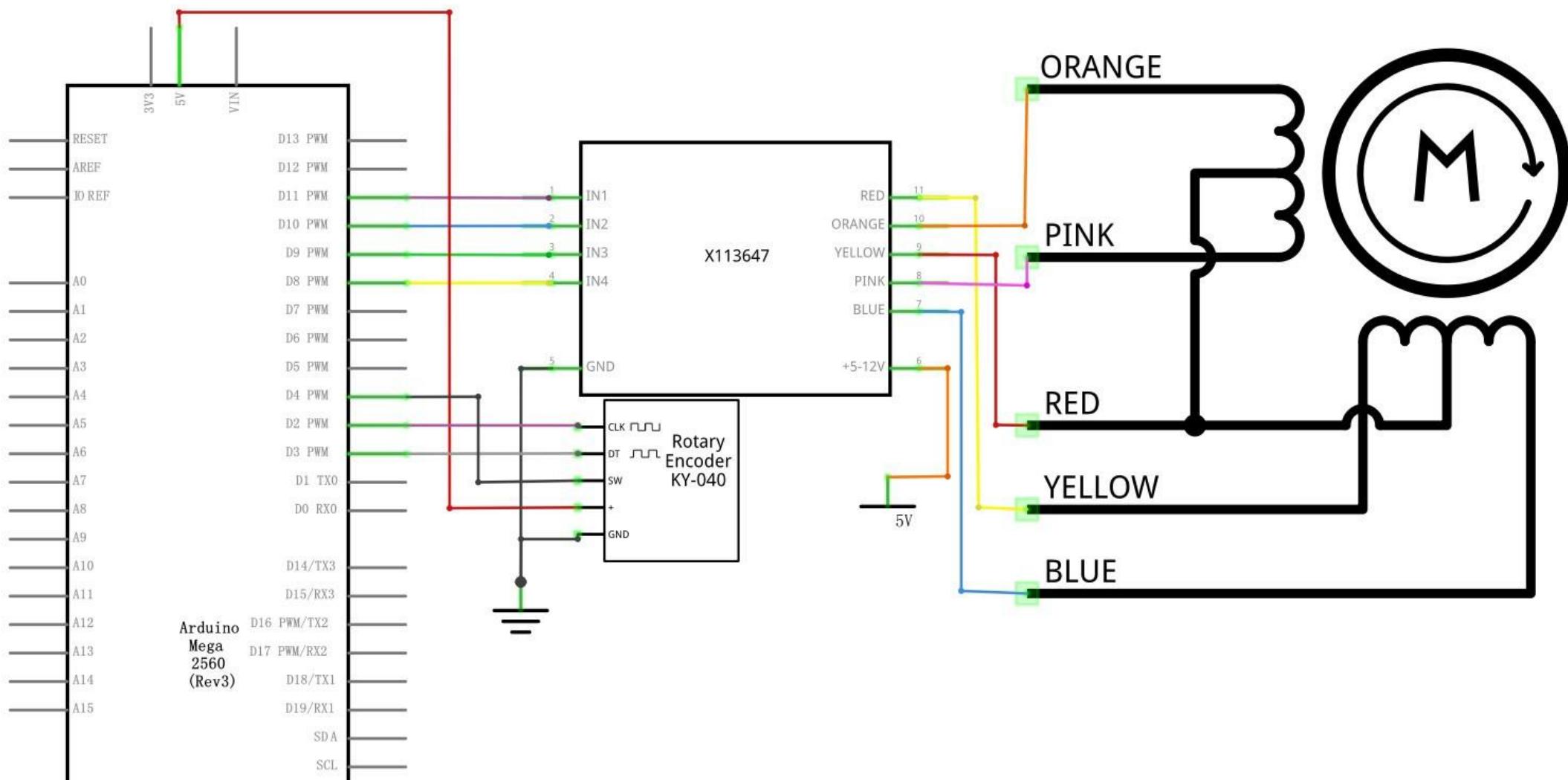
Es gibt zwei Haupttypen: absolut und inkrementell (relativ). Der Ausgang der Absolutwertgeber zeigt die aktuelle Position der Welle an, so dass sie Winkelgeber sind. Die Ausgabe von Inkrementalgebern gibt Auskunft über die Bewegung der Welle, die typischerweise anderweitig zu Informationen wie Geschwindigkeit, Distanz und Position weiterverarbeitet wird.

Drehgeber werden in vielen Anwendungen eingesetzt, die eine präzise, unbegrenzte Drehung benötigen - einschließlich industrieller Steuerungen, Robotik, fotografische Objektive, [1] Computer-Eingabegeräte (wie optomechanische Mäuse und Trackballs), kontrollierte Stress-Rheometer und rotierende Radarplattformen.

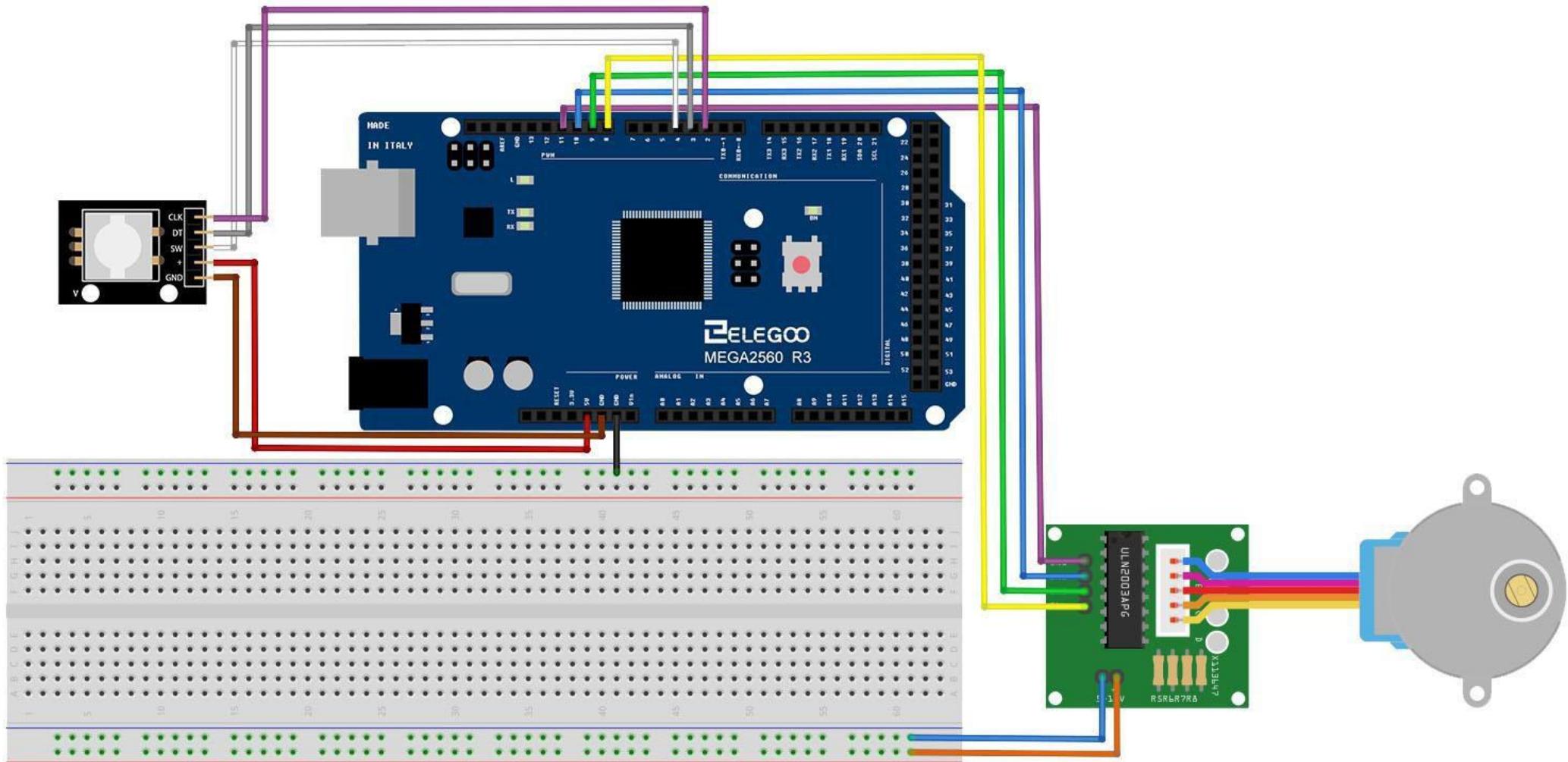


Verbindungen

Schema



Schaltplan



Wir verwenden 4 Pins, um den Stepper und 3 Pins um das Drehgebermodul zu steuern.

Die Stifte 8-11 steuern den Schrittmotor und die Stifte 2-4 empfangen Informationen vom Drehgeber.

Wir verbinden die 5V und Ground mit MEGA2560 an den Drehgeber und Vorsichtsmaßnahme für unseren MEGA2560 nutzen wir eine externe Stromversorgung des Schrittmotors, da der Motor mehr Leistung nutzen kann, als die die der MEGA2560 bieten kann.

Wir verbinden auch den MEGA2560 Ground mit dem Breadboard, um als Referenz zu dienen.

Code

Nach der Verdrahtung öffnen Sie bitte das Programm im Code-Ordner - Lektion 33 Steuern des Schrittmotors mit Rotary Encoder und klicken Sie auf UPLOAD, um das Programm hochzuladen. Weitere Informationen zum Programm-Upload finden Sie unter Lektion 2, wenn Fehler vorliegen.

Bevor Sie dies ausführen können, stellen Sie sicher, dass Sie die <Stepper> - Bibliothek installiert oder ggf. neu installiert haben. Andernfalls wird Ihr Code nicht funktionieren.

Weitere Informationen zum Laden der Bibliotheksdatei finden Sie unter [Lektion 1](#).

Wir verwenden einige Variablen, um die aktuelle Position zu speichern, da wir die Position des Schrittmotors verfolgen wollen, damit wir es wieder in die Ausgangsposition bringen können.

Wir haben auch einen Fehlercode eingegeben, um sicherzustellen, dass der Drehgeber keine Schritte übersieht oder vergisst, da dies unsere Motorposition ungenau machen würde.

Beispiefoto

