



**DAS VOLLSTÄNDIGSTE STARTER-KIT-TUTORIAL
FÜR UNO**

V1.0.17.1.11

Vorwort

Unsere Gesellschaft

Die im Jahr 2011 gegründete Elegoo GmbH ist ein professioneller Hersteller und Exporteur im Bereich Design, Entwicklung, Herstellung und Vermarktung von Arduino, 3D-Drucker, Raspberry Pi und STM32. Wir befinden uns in der Stadt Shenzhen, die als das chinesische Silicon Valley bekannt ist. Alle unsere Produkte erfüllen internationalen Qualitätsnormen, und werden in zahlreichen Märkten der Welt hoch geschätzt.

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

DE Amazon Storefront: <http://www.amazon.de/shops/A1780XYQ9DFQM6>

Unser Tutorial

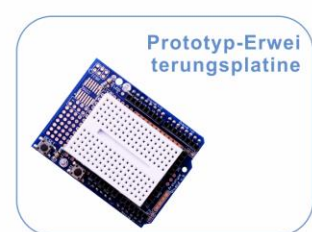
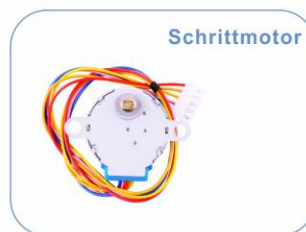
Dieses Tutorial richtet sich an Anfänger. Sie werden alle grundlegenden Informationen darüber erfahren, wie Controller-Bard, Sensoren und Komponenten von Arduino verwendet werden. Wenn Sie mehr über Arduino erfahren möchten, empfehlen wir Ihnen, das von Michael Margolis geschriebene *Arduino Cookbook* zu lesen.

Unser Service

Falls Sie Fragen, Kommentare oder Vorschläge zu unserer Gesellschaft, unserem Produkt oder dem Tutorial haben, senden Sie bitte eine E-Mail unverzüglich an EUservice@elegoo.com. Wir sind zur kontinuierlichen Verbesserung verpflichtet, so ist Ihr Feedback für uns unschätzbar.

Komponentenliste

www.elegoo.com



Kontaktiere Uns: EUserice@elegoo.com

www.elegoo.com



Kontaktiere Uns: EUservice@elegoo.com

 www.elegoo.com



Kontaktiere Uns: EUservice@elegoo.com

 www.elegoo.com



Kontaktiere Uns: EUservice@elegoo.com

www.elegoo.com



Kontaktiere Uns: EUService@elegoo.com

 www.elegoo.com



Kontaktiere Uns: EUservice@elegoo.com

Inhaltsverzeichnis

Lektion 0: IDE installieren.....	11
Lektion 1: Bibliotheken hinzufügen.....	22
Lektion 2: Blinken.....	32
Lektion 3: LED.....	41
Lektion 4: RGB LED.....	48
Lektion 5: Digitale Eingänge.....	55
Lektion 6: Töne erzeugen.....	60
Lektion 7: Passive Summer.....	64
Lektion 8: Kugelschalter.....	68
Lektion 9: Servomechanismus.....	72
Lektion 10: Ultraschall-Sensor-Modul.....	76
Lektion 11 Tastenmodul.....	81
Lektion 12: DHT11 Temperatur- und Feuchtesensor.....	87
Lektion 13: Analoges Joystick-Modul.....	92
Lektion 14: IR-Empfängermodul.....	98
Lektion 15: MAX7219 LED Dot-Matrix-Modul.....	104
Lektion 16 MPU-6050-Modul.....	108
Lektion 17: HC-SR501 PIR Sensor.....	116
Lektion 18: Wasstand-Erkennung-Sensormodul.....	124
Lektion 19: Echtzeituhren-Modul.....	129
Lektion 20: Schallsensor-Modul.....	134
Lektion 21: RC522 RFID-Modul.....	140
Lektion 22: LCD-Anzeige.....	146
Lektion 23: Thermometer.....	151
Lektion 24: Acht LEDs mit 74HC595.....	155
Lektion 25: Der Serielle Monitor.....	164
Lektion 26: Fozelle.....	170
Lektion 27: 74HC595 und Segmentanzeige.....	175

Lektion 28: 4-Stellige 7-Segmentanzeige	181
Lektion 29: DC Motoren	186
Lektion 30: Relais	196
Lektion 31: Schrittmotor	201
Lektion 32: Steuerung des Schrittmotors mit Fernbedienung	208
Lektion 33: Steuerung des Schrittmotors mit Encoder	212

Lektion 0: IDE installieren

Einführung

In dieser Lektion werden Sie erfahren, wie Sie Ihren Computer zur Verwendung von Arduino einrichten, und wie die folgenden Lektionen angeordnet werden.

Die Arduino-Software, mit der Sie Arduino programmieren werden, ist für Windows, Mac und Linux verfügbar. Der Installationsprozess unterscheidet sich für alle drei Plattformen, und leider müssen Sie für die Installation der Software noch eine bestimmte Menge an manuellen Arbeiten durchführen. Es gibt kein Installationsprogramm; stattdessen müssen Sie eine Datei, die Ihnen einen Arduino-Ordner, in dem das Arduino-Programm und einige andere Einträge enthalten werden, gibt, entpacken.

In einem getrennten Schritt müssen Sie dann die USB-Treiber installieren.

SCHRITT 1: Gehen Sie zu <https://www.arduino.cc/en/Main/Software> und Suchen Sie über Seite.

ARDUINO 1.8.0
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.

Windows Installer
Windows ZIP file for non admin install
Windows app

Mac OS X 10.7 Lion or newer


Linux 32 bits
Linux 64 bits
Linux ARM

[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

Die Version auf dieser Website ist in der Regel die neueste Version und die aktuelle Version neuer als die Version auf dem Bild werden kann.

STEP2: Laden Sie die **Entwicklungs software** herunter, die mit dem **Betriebssystem** Ihres **Computers** kompatibel ist. **Windows hier als Beispiel zu nehmen.**

Windows Installer
Windows ZIP file for non admin install

Windows app 

Mac OS X 10.7 Lion or newer

Linux 32 bits
Linux 64 bits
Linux ARM

Klicken Sie auf 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.



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **8,808,272** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

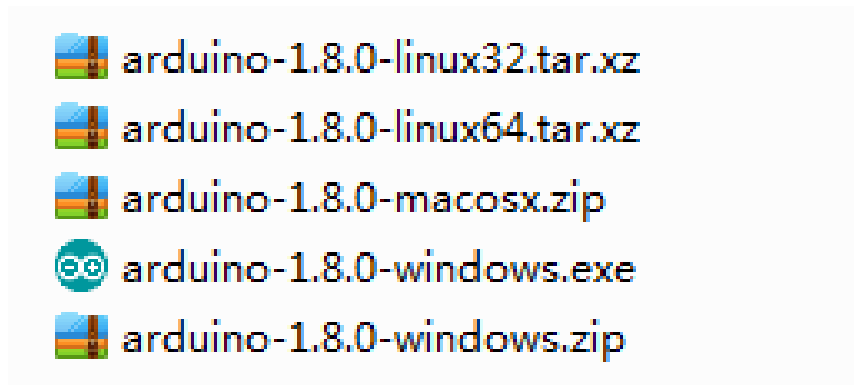
\$3 **\$5** **\$10** **\$25** **\$50** **OTHER**

JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

Klicken Sie auf JUST DOWNLOAD.

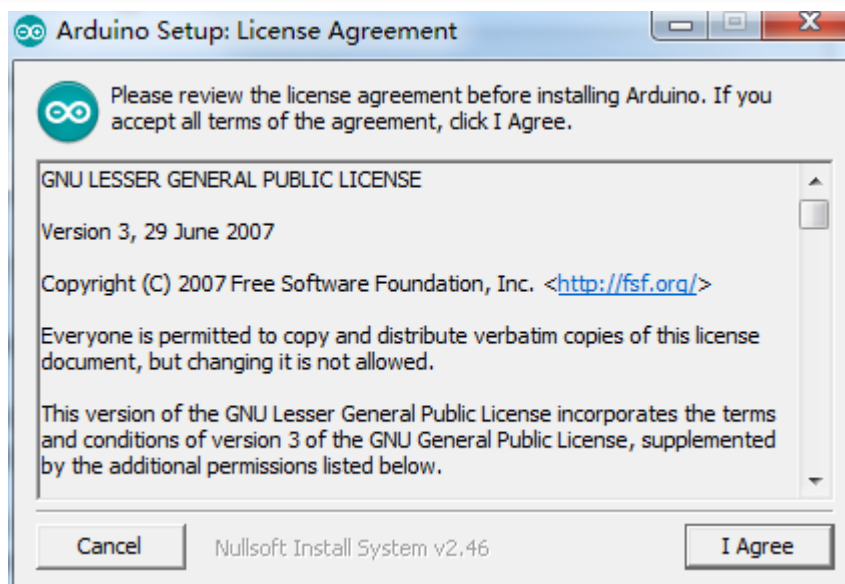
Darüber hinaus Version 1.8.0 ist in das Material, das wir zur Verfügung gestellt, und die Versionen unserer Materialien sind die neuesten Versionen, wenn dieser Kurs gemacht wurde.



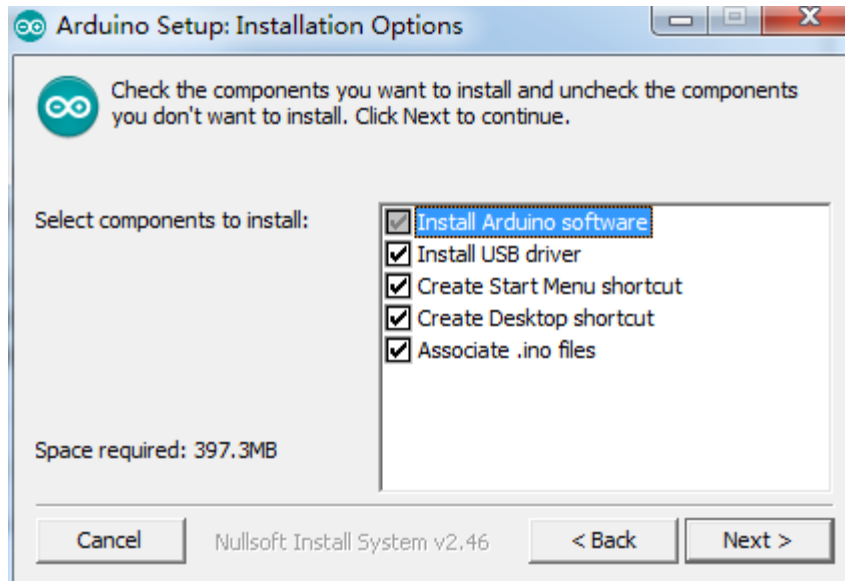
Installation von Arduino (Windows)

Sie können es mit der EXE-Installations-Paket oder dem grünen Paket installieren.

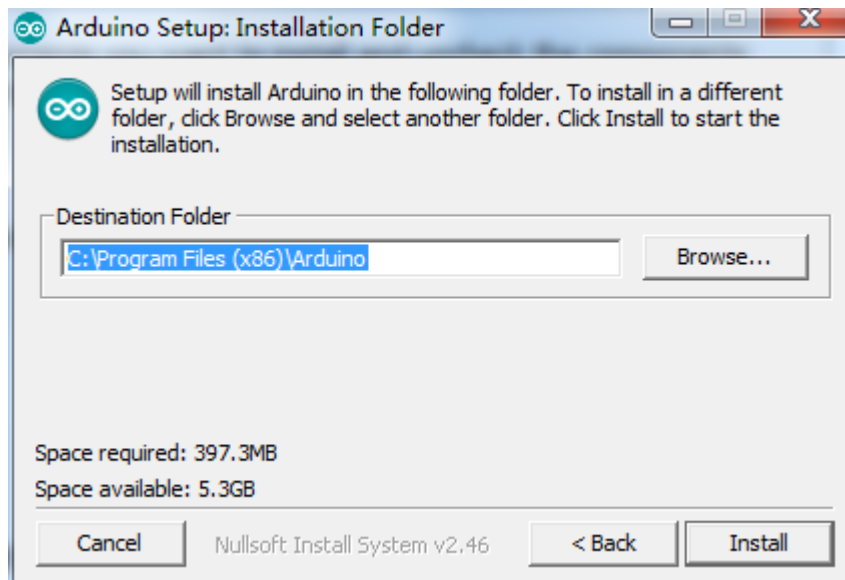
Installieren Sie Arduino mit dem exe. Installationspaket.



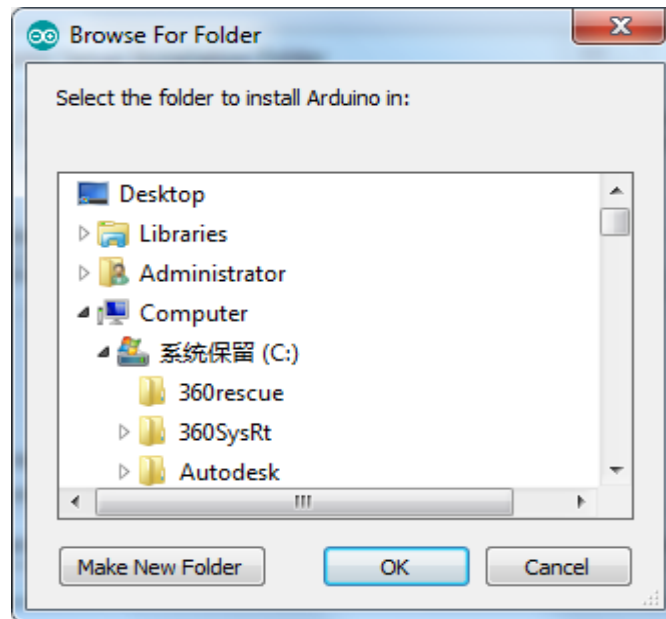
Klicken Sie auf "I Agree", um die folgende Schnittstelle zu sehen.



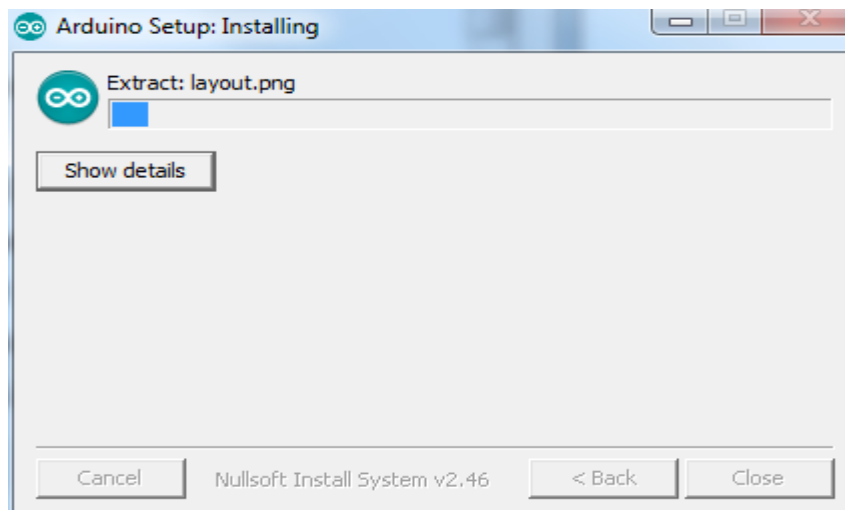
Klicken Sie auf Next.



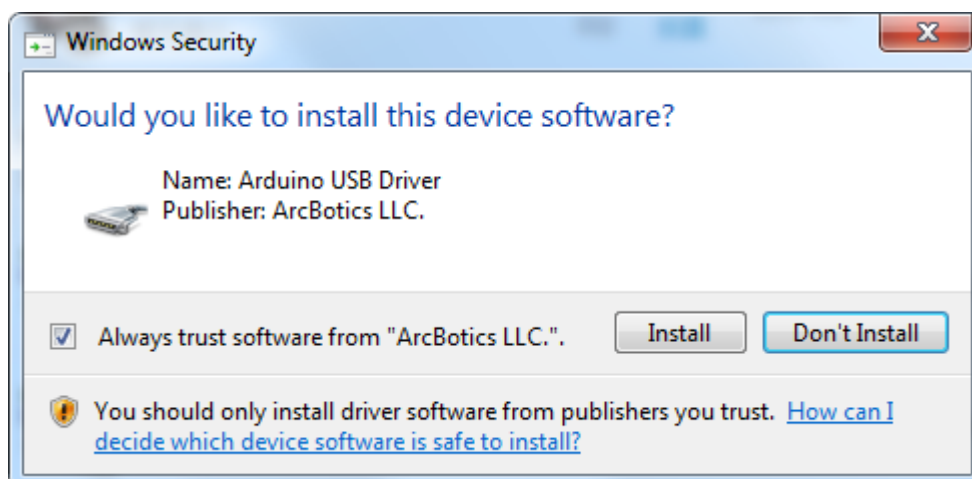
Sie können auf "Browse..." klicken, um einen Installationspfad oder Verzeichnis typ in dem gewünschten Verzeichnis auszuwählen.



Klicken Sie auf *installieren*, um die Installation zu starten.



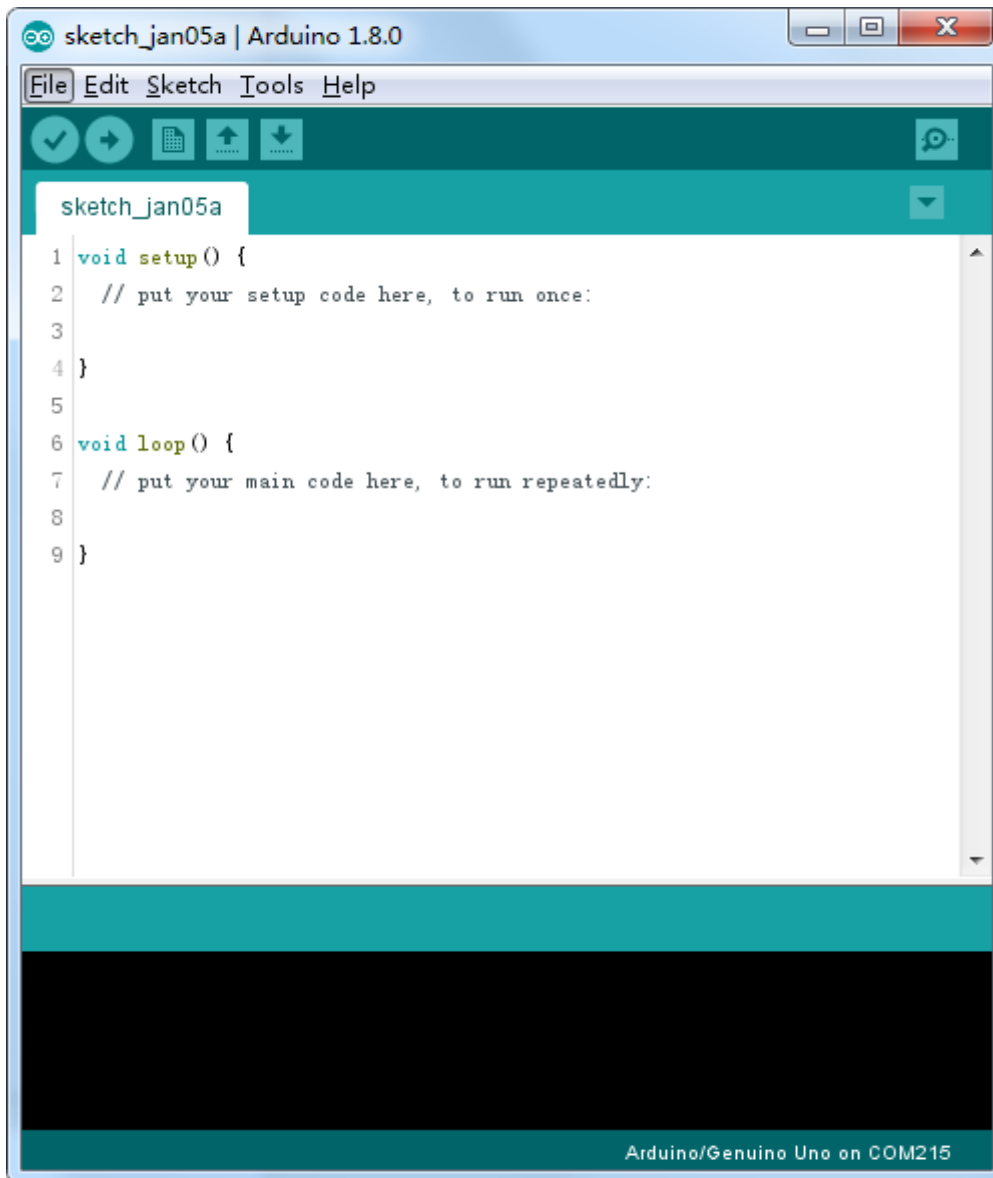
Zu guter Letzt die folgende Schnittstelle angezeigt wird, klicken Sie auf *installieren*, um die Installation abzuschließen.



Als nächstes erscheint das folgende Symbol auf dem Desktop.

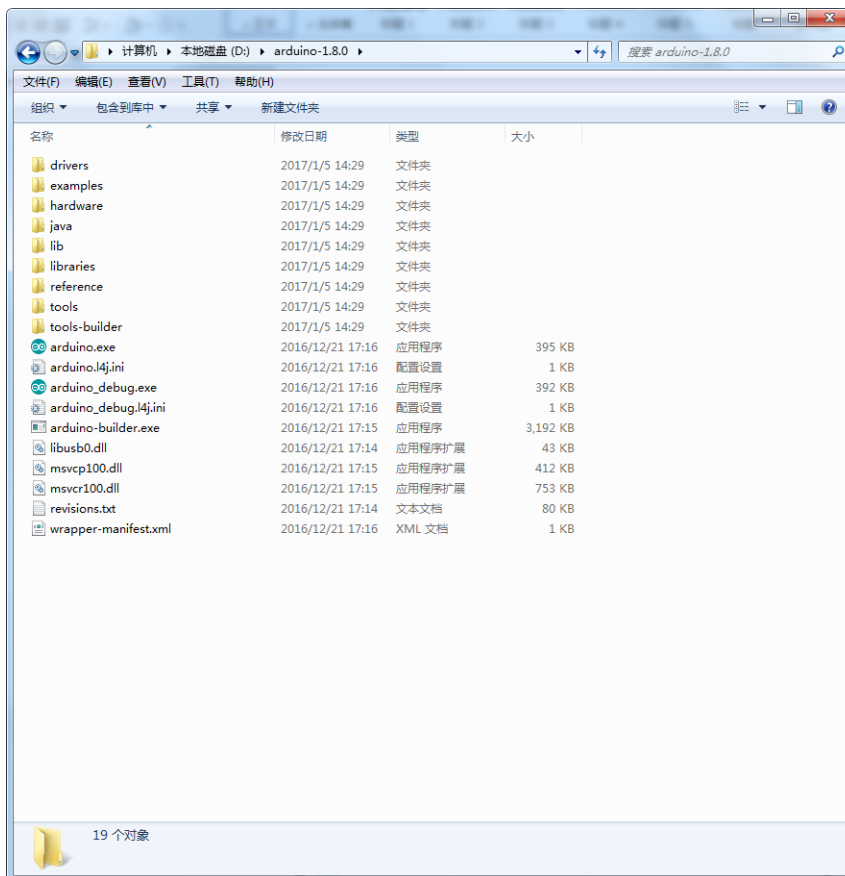


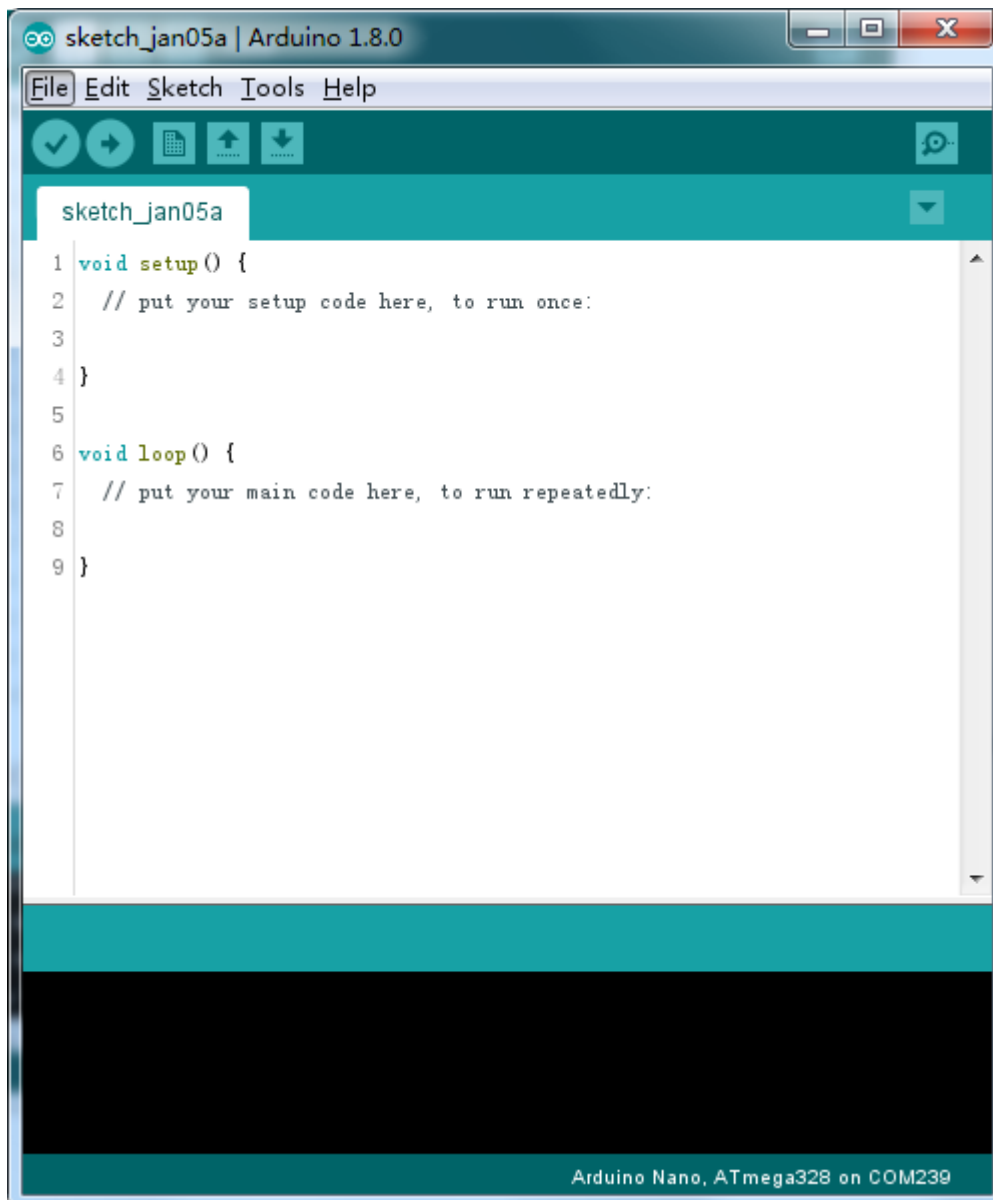
Doppelklicken Sie auf die gewünschte Entwicklung Umwelt gelangen



Sie können direkt wählen das Installationspaket für die Installation und den Inhalt unten überspringen und springen zum nächsten Abschnitt. Aber wenn Sie einige andere Methoden als das Installationspaket erfahren möchten, bitte lesen Sie den Abschnitt weiter.

Entpacken Sie die Zip-Datei heruntergeladen, doppelklicken Sie auf das Programm öffnen und geben die gewünschte Entwicklungsumgebung.





Allerdings braucht diese Installationsmethode separate Installation des Treibers.

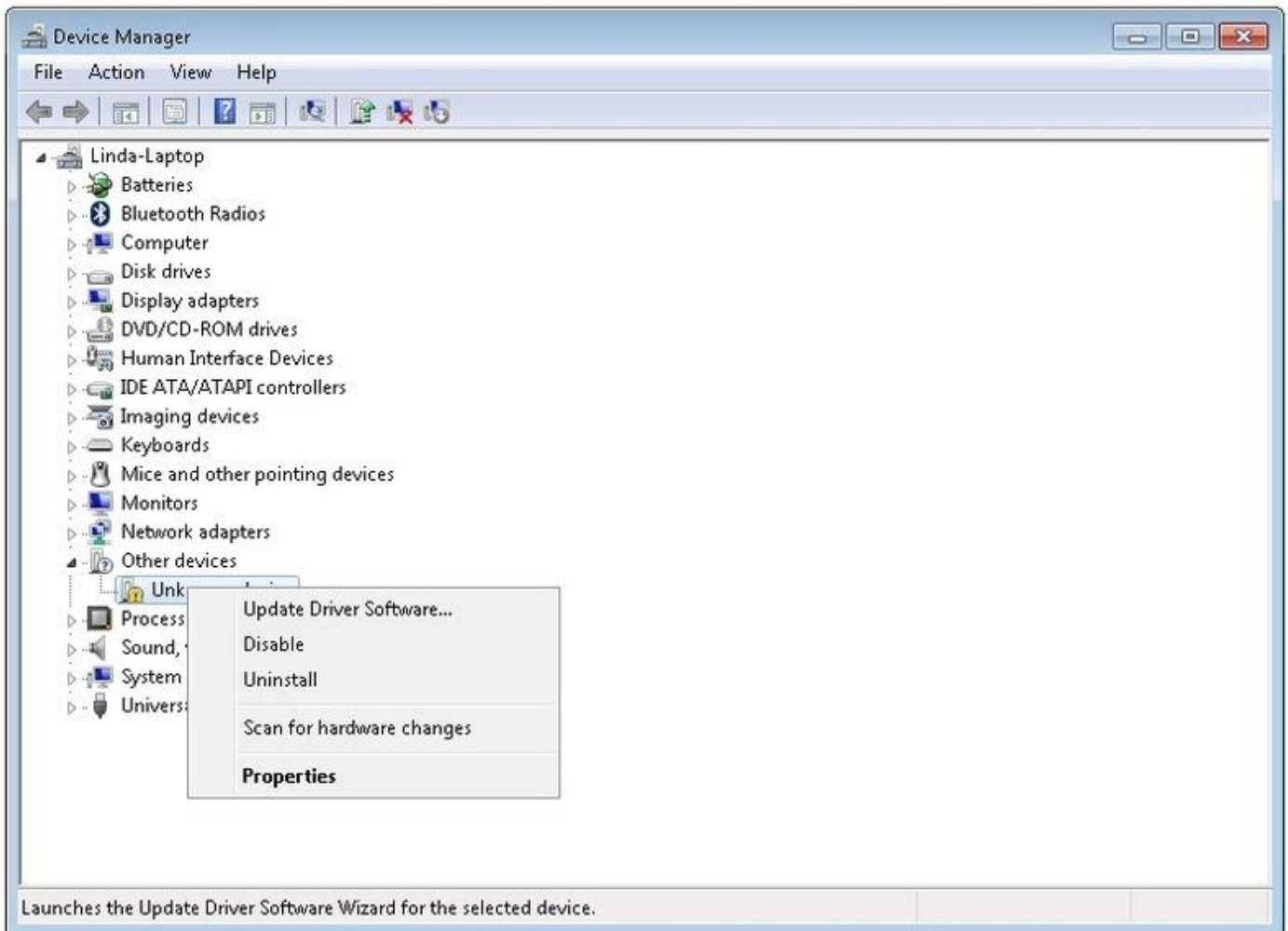
Der Arduino-Ordner enthält die Arduino-Programm selbst und die Treiber, die die Arduino, von einem USB-Kabel an den Computer angeschlossen werden können. Bevor wir die Arduino-Software starten, wirst du die USB-Treiber zu installieren.

Stecken Sie ein Ende des USB-Kabels in die Arduino und das andere Ende in einen USB-Anschluss auf Ihrem Computer. Die Betriebsanzeige LED leuchtet auf und Sie erhalten eine Meldung "Neue Hardware gefunden" von Windows. Ignorieren Sie diese Nachricht und stornieren Sie alle Versuche, die Windows macht, versuchen Sie es und installieren von Treibern automatisch für Sie.

Die zuverlässigste Methode die USB-Treiber zu installieren ist, verwenden Sie den Geräte-Manager. Dies ist auf verschiedene Arten je nach Ihrer Version von Windows zugegriffen. In Windows 7 muss

du zunächst öffnen Sie die Systemsteuerung, und wählen die Option, um Symbole anzuzeigen, und sollten Sie den Geräte-Manager in der Liste finden.

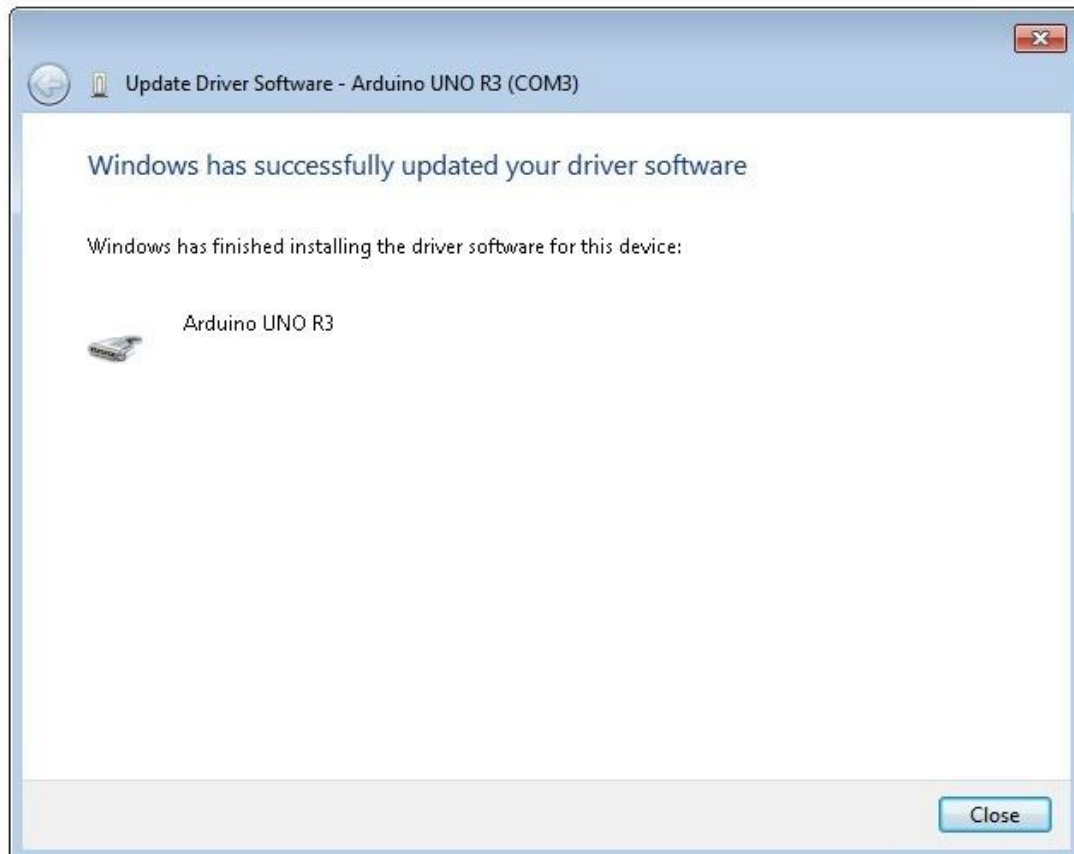
Unter "Andere Geräte" sollten Sie ein Symbol für "unbekanntes Gerät" mit ein wenig gelb Dreieck daneben Warnung sehen. Dies ist Ihre Arduino.



Presse auf dem Gerät mit der rechten Maustaste und wählen Sie den obersten Menüpunkt (Treibersoftware aktualisieren...). Sie werden dann aufgefordert, entweder "Suche automatisch nach Treibersoftware" oder "Suche mein Computer nach aktualisierter Treibersoftware". Wählen Sie die Option zum Durchsuchen und navigieren zu der X\arduino1.8.0\Treiber.



Klicken Sie auf "Weiter" und Sie erhalten eine Sicherheit Warnung, wenn das passiert, können die Software installiert werden. Sobald die Software installiert wurde, erhalten Sie eine Bestätigungsmeldung.



Windows-Benutzer können die Installationsanweisungen für Mac und Linux Systeme überspringen und direkt zu Lektion 2. Mac und Linux-Benutzer können weiterhin diesen Abschnitt zu lesen.

Installation von Arduino (Mac OS X)

Laden und entpacken Sie die Zip-Datei, drücken Sie doppelklicken Sie auf die "Arduino.app", um Arduino IDE eingeben; Wird das System Sie bitten, Java-Laufzeitbibliothek zu installieren, wenn Sie es nicht in Ihrem Computer haben.

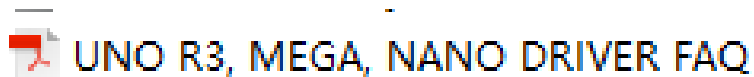


Installation von Arduino (Linux)

Du musst den "Installieren"-Befehl verwenden. Wenn Sie die Ubuntu-System verwenden, empfiehlt es sich, Arduino IDE von Software Center von Ubuntu zu installieren.



TIPPS: Sollten Sie Probleme bei der Installation der Treiber haben, lesen Sie bitte die **UNO R3, MEGA, NANO DRIVER FAQ.**



Lektion 1: Bibliotheken hinzufügen

Sobald Sie mit der Arduino-Software vertraut sind und wenn Sie die eingebauten Funktionen verwenden, möchten Sie vielleicht ihre Funktionalität mit zusätzlichen Bibliotheken erweitern.

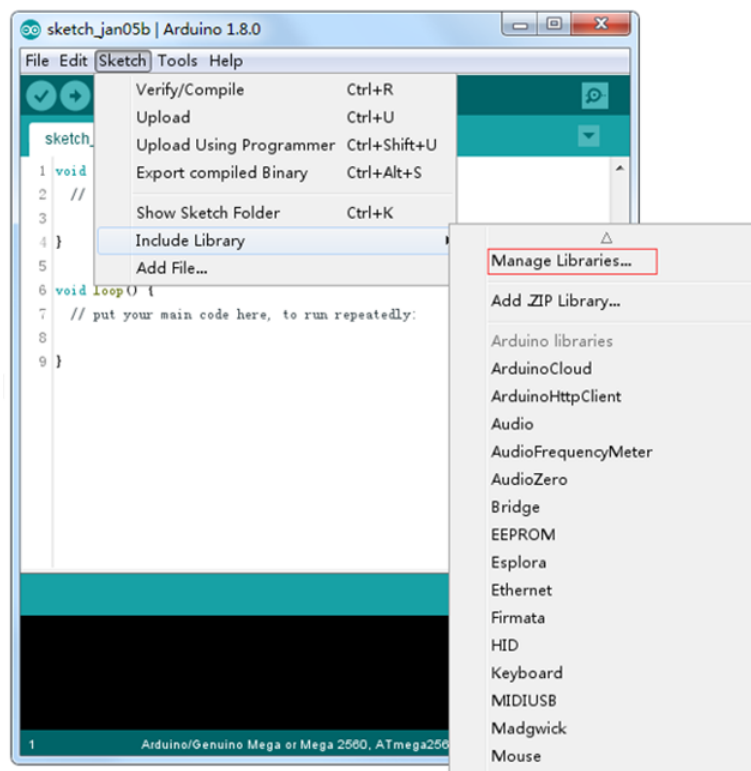
Was sind Bibliotheken?

Bibliotheken sind eine Sammlung von Code, der es Ihnen einfach macht, eine Verbindung mit einem Sensor, einer Anzeige, einem Modul, etc. herzustellen. Zum Beispiel ermöglicht es die eingebaute LiquidCrystal-Bibliothek Ihnen, mit den LCD-Zeichenanzeigen zu sprechen. Es gibt Hunderte von zusätzlichen Bibliotheken, die im Internet für Herunterladen verfügbar ist. Die eingebauten Bibliotheken und einige dieser zusätzlichen Bibliotheken sind in der Referenz aufgelistet. Um diese zusätzlichen Bibliotheken zu verwenden, müssen Sie sie installieren.

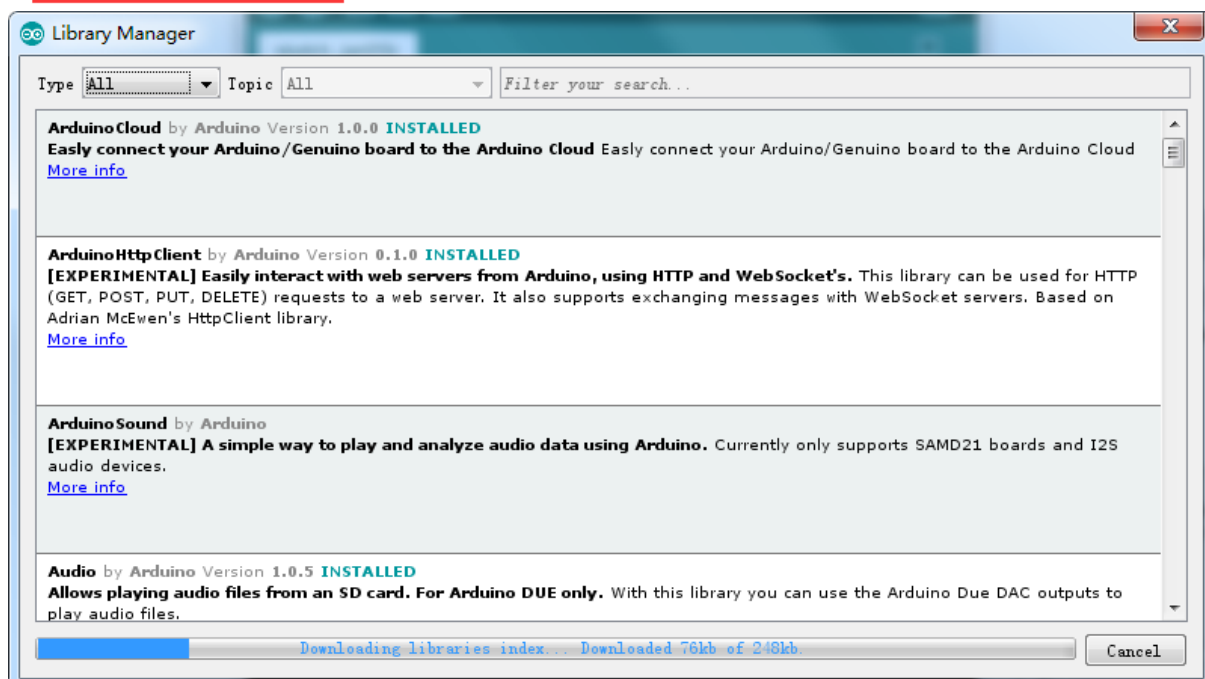
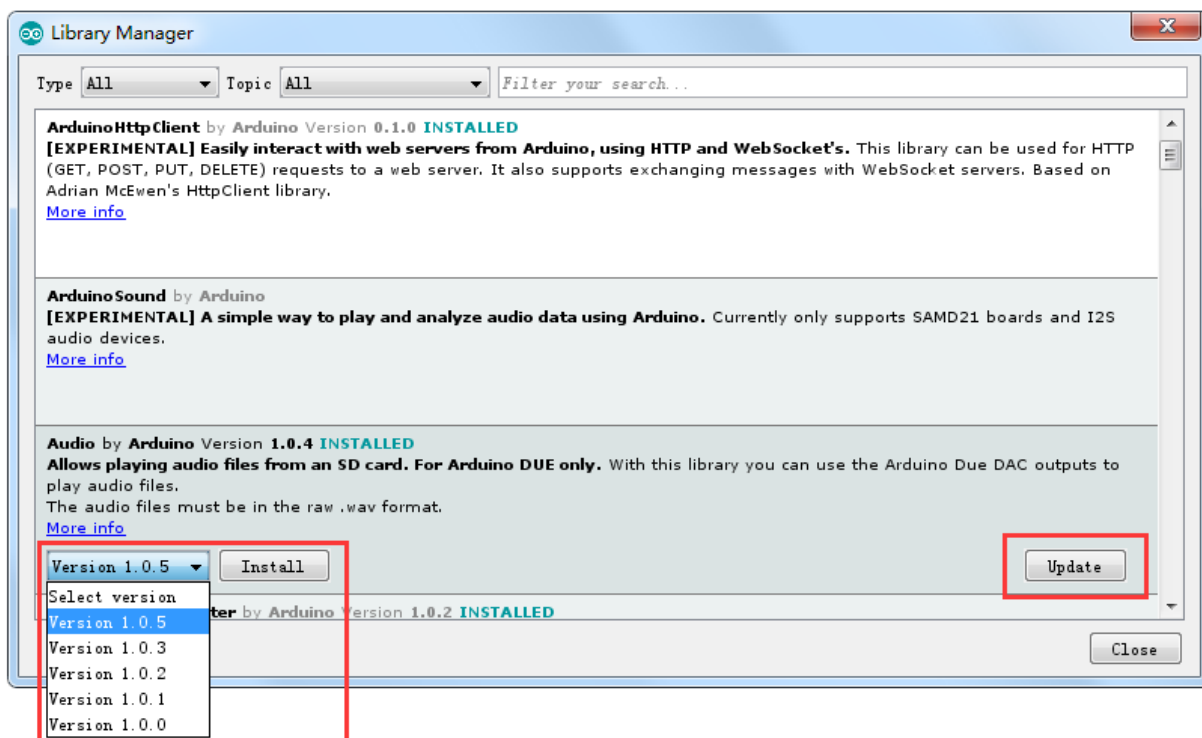
Wie eine Bibliothek installiert wird

Mit dem Bibliotheksmanager

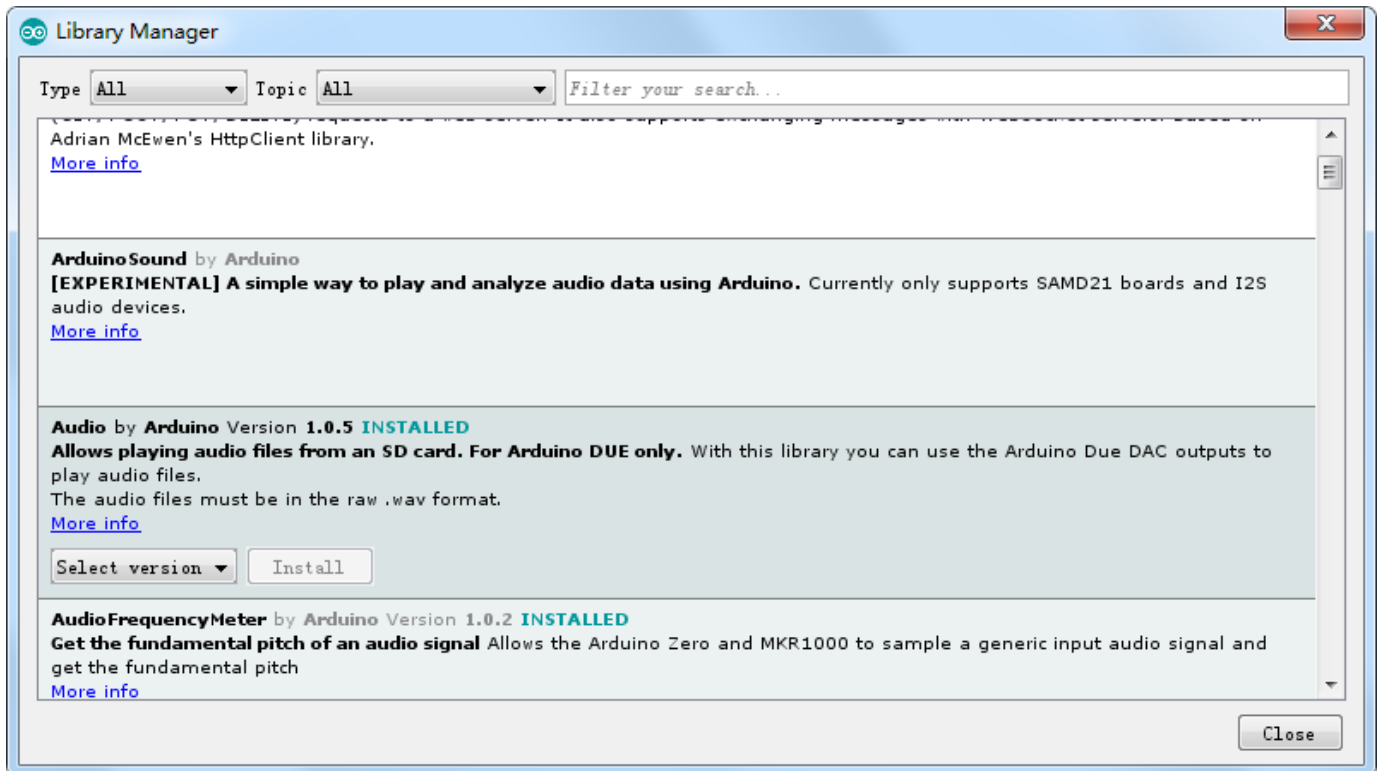
Um eine neue Bibliothek in Ihrer Arduino-IDE zu installieren, können Sie den Bibliotheksmanager (ab IDE-Version 1.6.2 verfügbar) verwenden. Öffnen Sie die IDE, und klicken Sie Sketch > Include > Library > Manage Libraries.



Der Bibliotheksmanager wird geöffnet, und Sie werden eine Liste von Bibliotheken, die bereits installiert sind oder bereit für Installation sind. In diesem Beispiel wird die Bridge-Bibliothek installiert. Blättern Sie die Liste nach unten, um sie zu finden, dann wählen Sie die Version der Bibliothek, die Sie installieren möchten, aus. Manchmal ist nur eine Version der Bibliothek verfügbar. Falls das Auswahlménü der Version nicht erscheint, machen Sie sich keine Sorge; es ist normal. Es gibt Zeiten, wo haben Sie Geduld mit ihm, genauso wie in der Abbildung gezeigt. Bitte aktualisieren Sie es und warten.



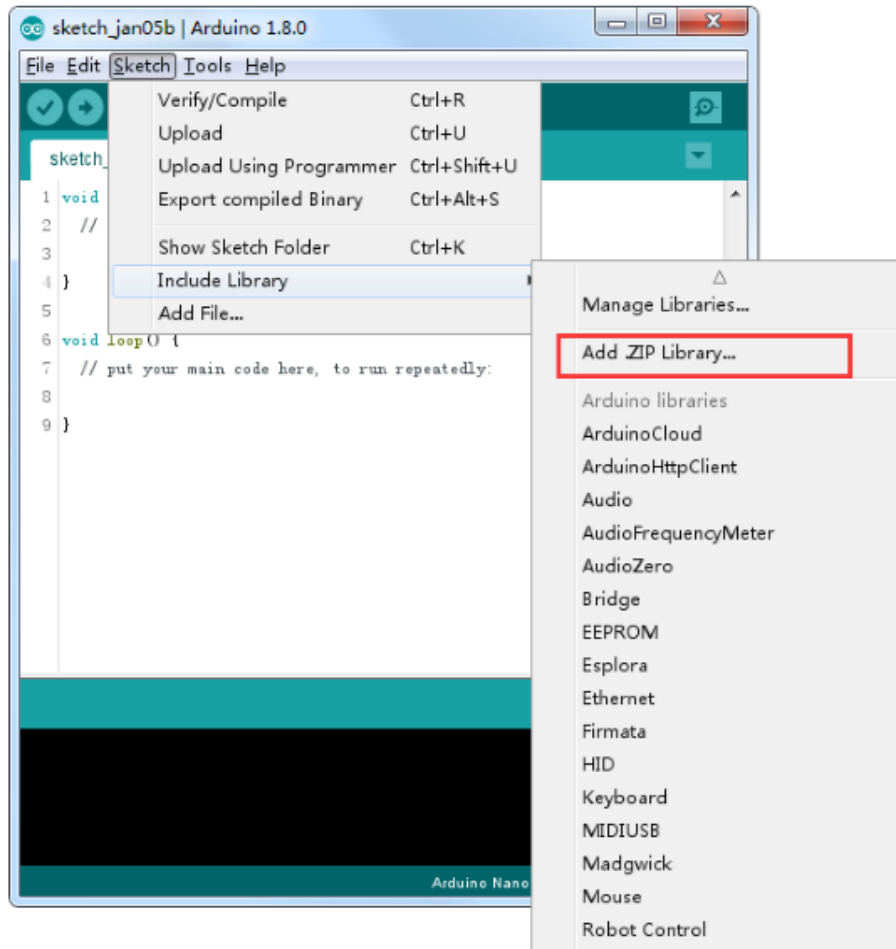
Schließlich klicken Sie auf Installieren, und warten Sie darauf, dass die IDE die neue Bibliothek installiert. Der Prozess des Herunterladens könnte in Abhängigkeit von der Verbindungsgeschwindigkeit Zeit brauchen. Sobald dieser Prozess abgeschlossen ist, sollte eine installierte Kennzeichnung neben der Bridge-Bibliothek erscheinen. Sie können dann den Bibliotheksmanager schließen.



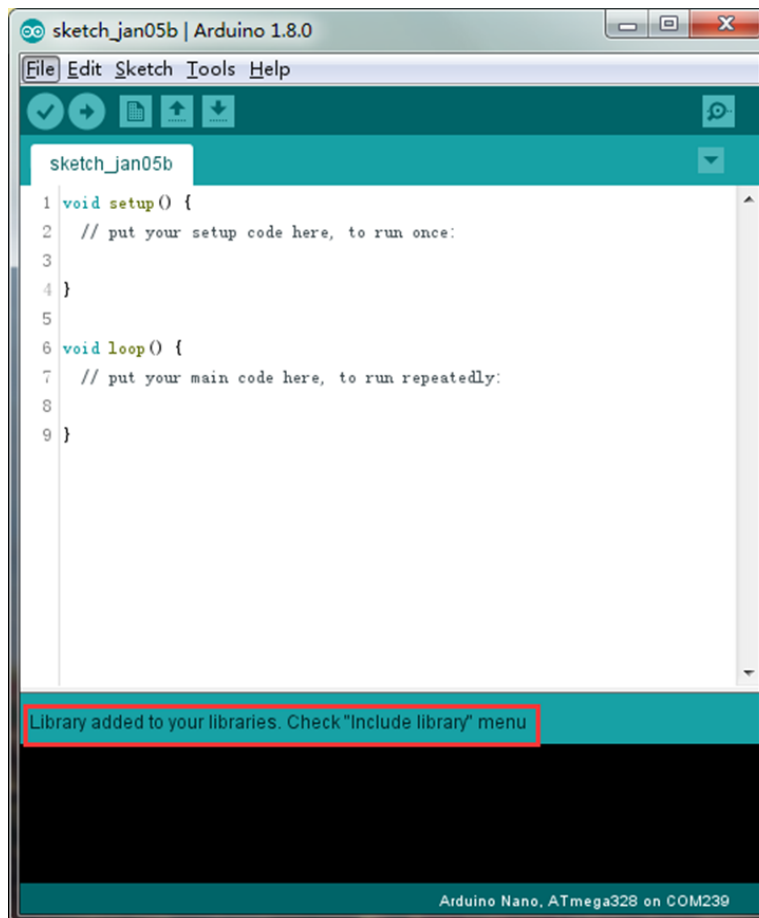
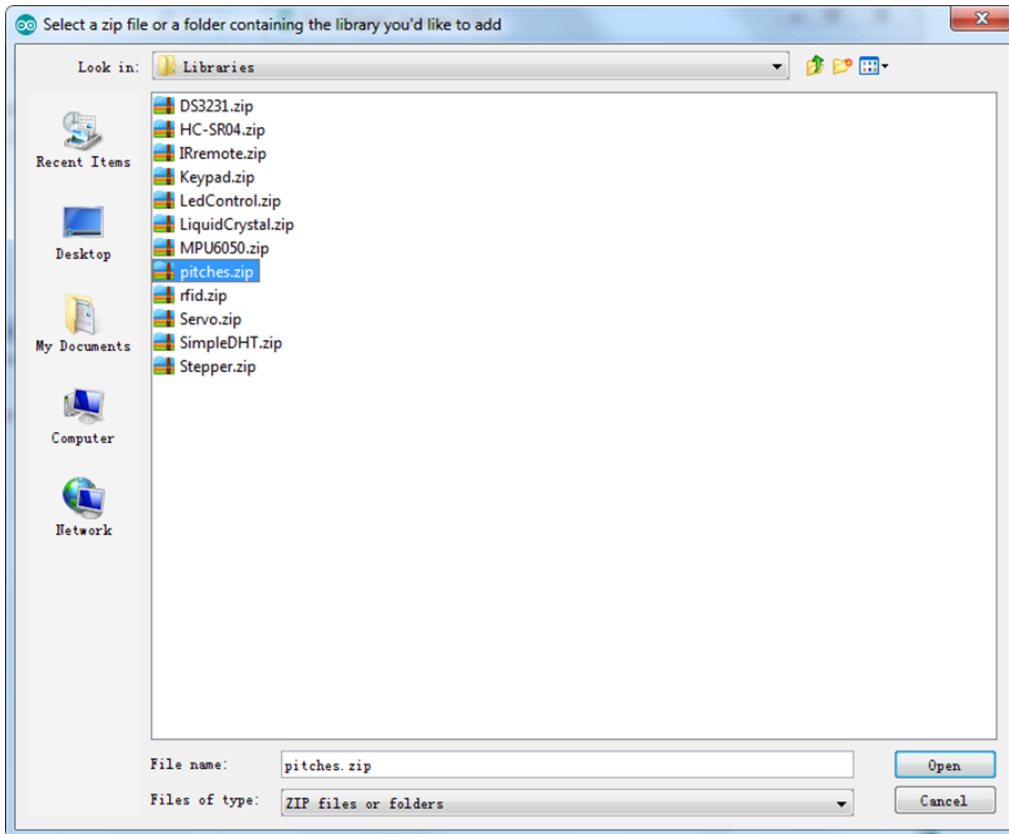
Nun können Sie eine neue Bibliothek, die in dem Bibliotheksmenü verfügbar ist, finden. Wenn Sie Ihre eigene Bibliothek hinzufügen möchten, öffnen Sie eine neue Aufgabe auf GitHub.

Eine .Zip-Bibliothek importieren

Bibliotheken werden oft als eine Zip-Datei oder -Ordner verteilt. Die Bezeichnung des Ordners ist die Bezeichnung der Bibliothek. In dem Ordner sind die folgenden Dateien: .cpp-Datei, .h-Datei, häufig ein Stichwort.txt-Datei, Beispielordner, und andere Dateien für die Bibliothek notwendig. Ab Version 1.0.5 können Sie Bibliotheken von einem Drittanbieter in der IDE installieren. Entpacken Sie die heruntergeladene Bibliothek nicht; lassen Sie sie wie sie ist. In der Arduino-IDE navigieren Sie zu Sketch > Include Library > Add .ZIP Library.



Sie werden zur Auswahl der Bibliothek, die Sie hinzufügen möchten, veranlasst. Navigieren Sie zum Speicherort der Zip-Datei und öffnen Sie sie.



Kehren Sie auf das Sketch > Import Library-Menü zurück. Nun sollten Sie die Bibliothek am unteren Rand des Drop-Down-Menüs finden. Sie ist bereit, in Ihrem Sketch verwendet zu werden. Die Zip-Datei wird in dem Bibliotheksordner in Ihrem Arduino-Sketch-Verzeichnis weiter erweitert.

NB: Die Bibliothek kann in Sketchen verwendet werden, aber Beispiele für die Bibliothek werden nicht in File > Examples angezeigt, bevor die IDE erneut gestartet wurde.

Diese beiden sind die am häufigsten verwendeten Ansätze. MAC und Linux-Systeme können ebenfalls behandelt werden. Die manuelle Installation unter eingeführt werden, da eine Alternative nur selten eingesetzt werden kann und Benutzer mit keinen Anforderungen können es zu überspringen.

Manuelle Installation

Um die Bibliothek zu installieren, beenden Sie zunächst das Arduino-Anwendungsprogramm. Dann entpacken Sie die Zip-Datei, die die Bibliothek enthält. Zum Beispiel, wenn Sie eine Bibliothek mit der Bezeichnung „ArduinoParty“ installieren, dekomprimieren Sie ArduinoParty.zip. Sie sollte einen Ordner mit der Bezeichnung ArduinoParty, in dem Dateien wie ArduinoParty.cpp und ArduinoParty.h enthalten sind, umfassen. (Falls die .cpp-Datei und die .h-Datei nicht in einem Ordner enthalten sind, müssen Sie einen solchen Ordner erstellen. In diesem Fall erstellen Sie einen Ordner mit der Bezeichnung „ArduinoParty“, und verschieben Sie alle Dateien wie ArduinoParty.cpp und ArduinoParty.h., die in der Zip-Datei enthalten werden, in den Ordner.)

Schieben Sie den ArduinoParty-Ordner in diesen Ordner (Ihren Bibliotheksordner). Under Windows wird er wahrscheinlich als „My Documents\Arduino\libraries“ bezeichnet. Für Mac-Anwender wird er wahrscheinlich als „Documents/Arduino/libraries“ bezeichnet. Auf Linux wird er der Ordner von „Bibliotheken“ in Ihrem Sketchbook sein.

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

`My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp`

`My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h`

`My Documents\Arduino\libraries\ArduinoParty\examples`

....

oder wie folgt aussehen (auf Mac und Linux):

`Documents/Arduino/libraries/ArduinoParty/ArduinoParty.cpp`

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

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

....

Es gibt vielleicht mehr Dateien als die .cpp-Datei und die .h-Datei, so stellen Sie sicher, dass sie alle da sind. (Die Bibliothek wird nicht funktionieren, wenn die .cpp-Datei und die .h-Datei unmittelbar in den Bibliotheksordner gesetzt werden, oder wenn sie sich in einem verschachtelten zusätzlichen Ordner befinden. Zum Beispiel: Documents\Arduino\libraries

\ArduinoParty.cpp und Documents\Arduino\libraries\ArduinoParty\ArduinoParty\ArduinoParty.cpp werden nicht funktionieren.)

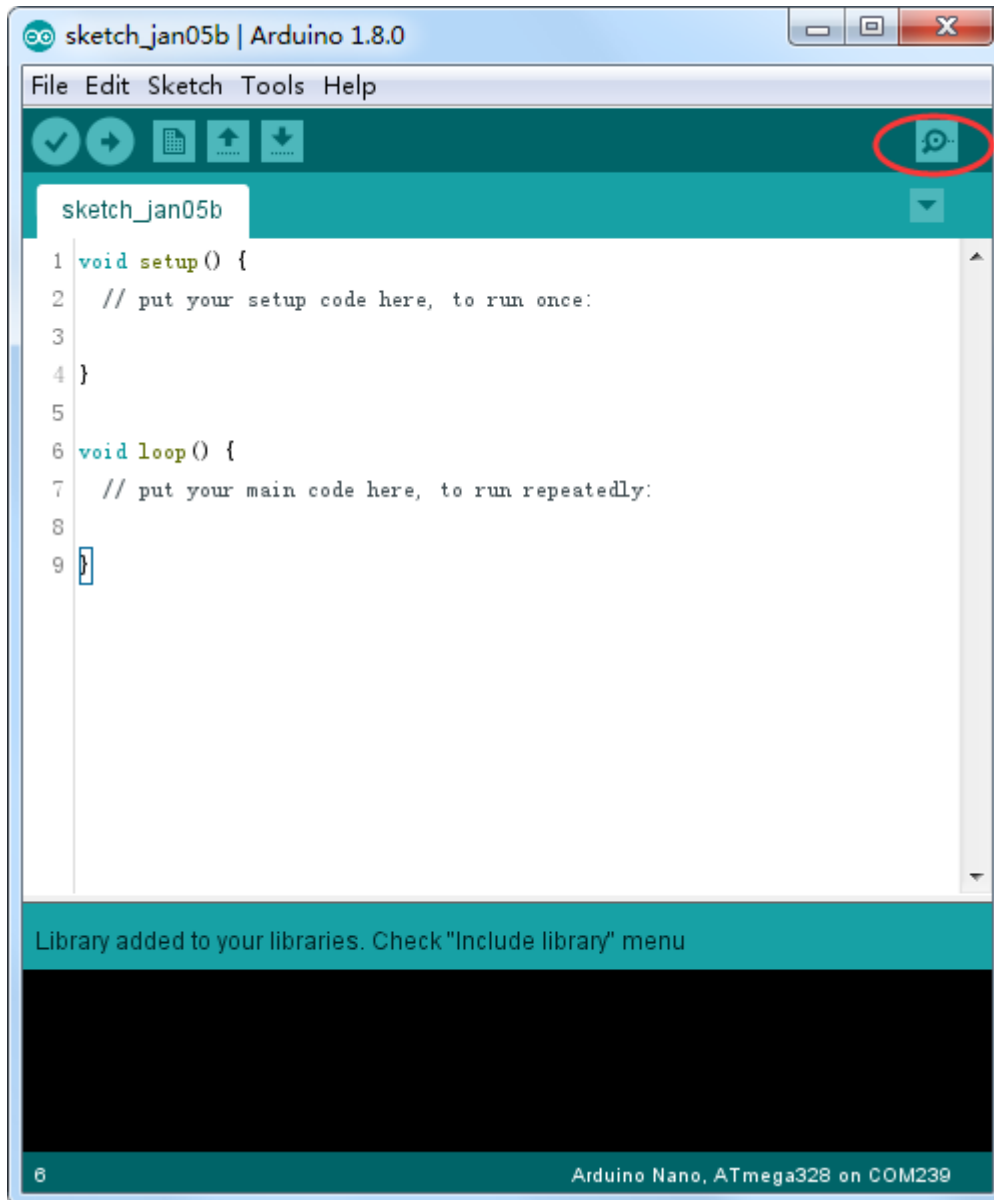
Starten Sie das Arduino-Anwendungsprogramm erneut. Stellen Sie sicher, dass die neue Bibliothek in dem Menü Sketch > Import Library erscheint. Fertig. Sie haben eine Bibliothek installiert!

Arduino Serieller Monitor (Windows, Mac, Linux)

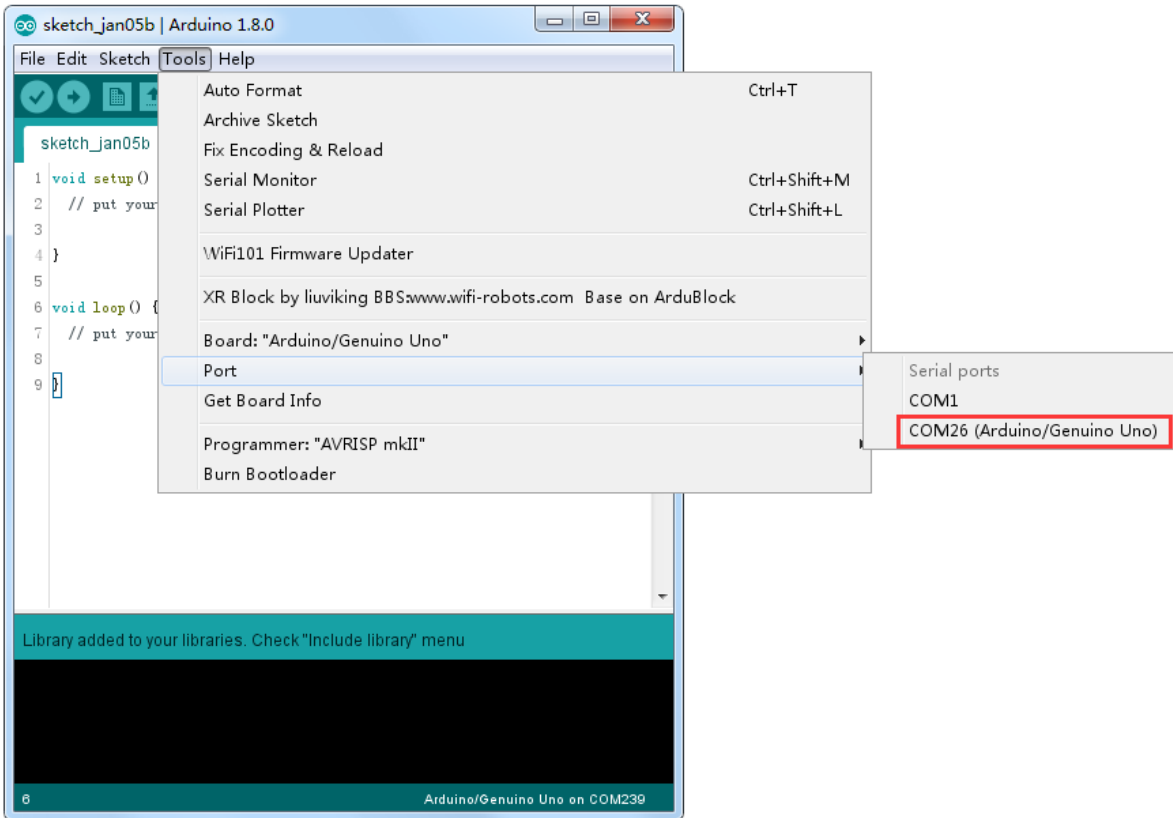
Der Arduino Integrated Development Environment (IDE) ist der Software-Seite der Arduino-Plattform. Und da mit einem Terminal ein großer Teil ist der Arbeit mit Arduinos und anderen Mikrocontroller, beschlossen sie, einen seriellen Terminal mit der Software enthalten. Innerhalb der Arduino-Umgebung nennt man den Serieller Monitor.

Herstellen einer Verbindung

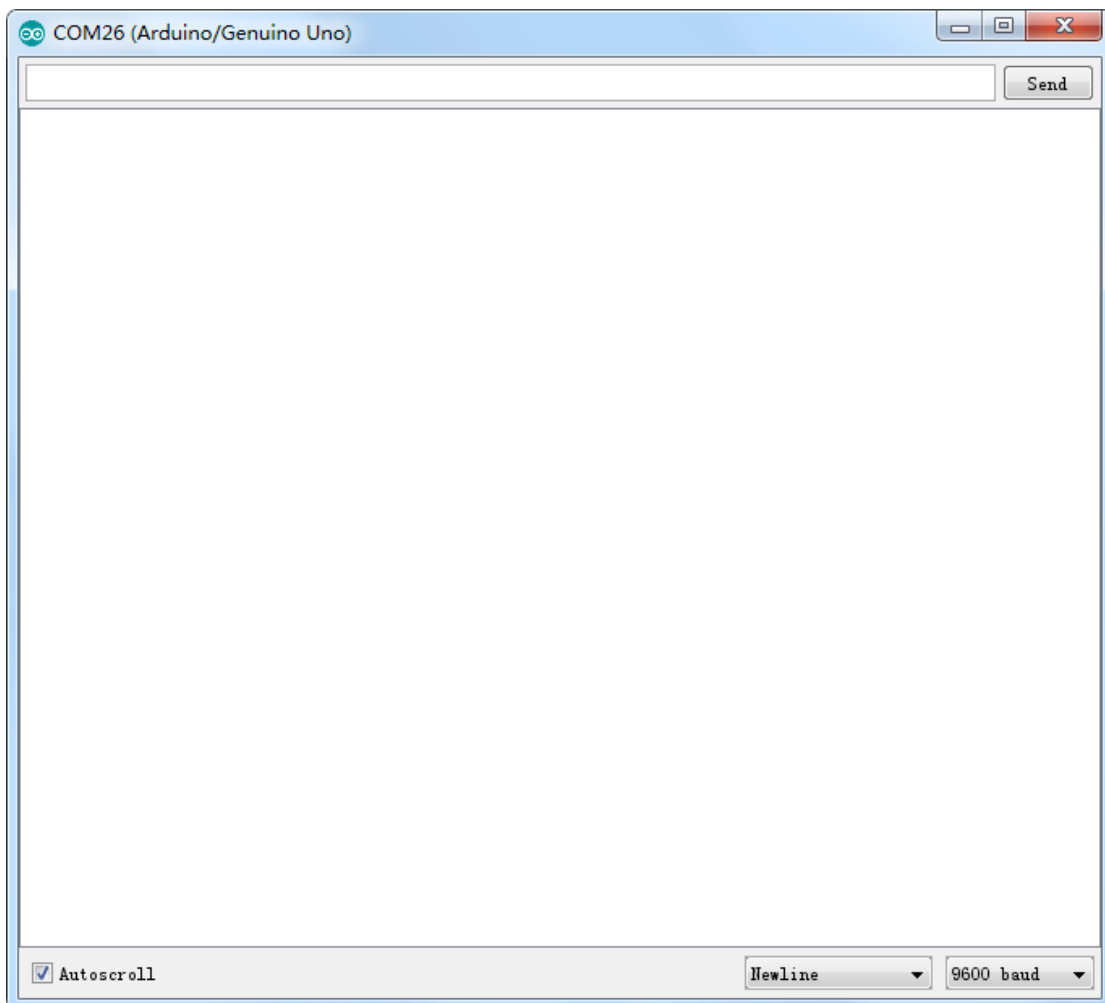
Seriellen Monitor kommt mit allen Version der Arduino IDE. Um es zu öffnen, klicken Sie einfach das Serieller Monitor-Symbol.



Auswählen welchen Port soll in den Seriellen Monitor ist dasselbe wie einen Hafen für das Hochladen von Arduino Code auswählen. Gehen Sie zu *Werkzeuge* -> *Serielle Schnittstelle*, und wählen Sie den richtigen Schnittstelle.

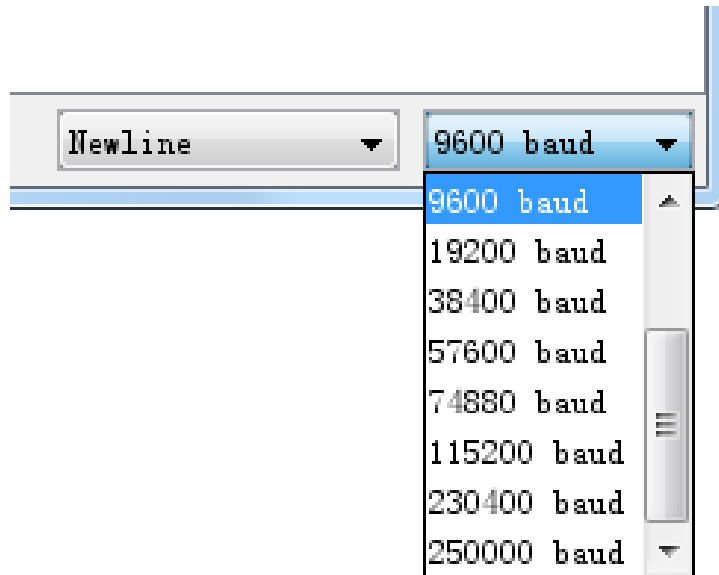


Einmal geöffnet, sollte Folgendes angezeigt werden:

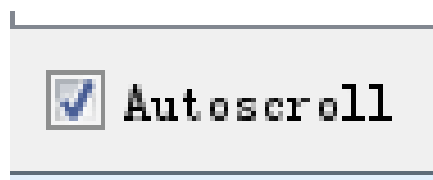


Einstellungen

Die Serieller Monitor hat einer begrenzten Einstellung, aber genug, um die meisten Ihrer seriellen Kommunikations-Bedürfnisse zu behandeln. Die erste Einstellung, die du ändern kannst ist die Baud-Rate. Klicken Sie auf die Baud-Rate Dropdown-Menü, wählen die richtige Baudrate (9600 Baud).



Schließlich, setzen Sie das Terminal Autoscroll oder nicht durch Anklicken des Kästchens in der unteren linken Ecke.



Vorteile

Der Serieller Monitor ist eine sehr schnelle und einfache Möglichkeit herstellen eine serielle Verbindung mit Ihrem Arduino. Wenn Sie bereits in der Arduino IDE arbeiten, gibt es wirklich keine Notwendigkeit, einem separaten Terminal zur Datenanzeige zu öffnen.

Nachteile

Der Mangel an Einstellungen lässt viel zu wünschen übrig auf den Serial Monitor, und für die fortgeschrittene serielle Kommunikation, kann es nicht den Trick tun.

Lektion 2: Blinken

Überblick

In dieser Lektion werden Sie erfahren, wie Sie Ihr UNO R3-Controller-Board programmieren, um die eingebaute LED von Arduino blinken zu lassen.

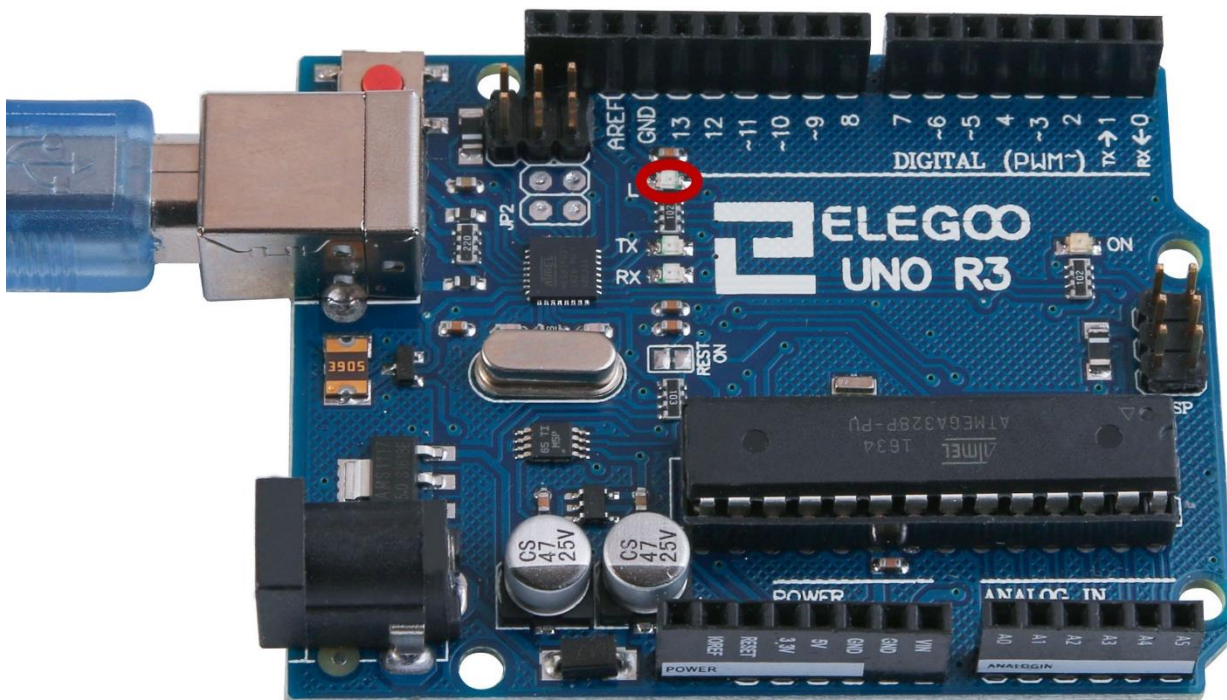
Erforderliche Komponente

(1) Elegoo UNO-R3

Grundsatz

Entlang den beiden Seiten verfügt das UNO R3-Board über Reihen von Verbindern, die für die Verbindung mit mehreren elektronischen Geräten und Plug-in ‚Schutzschilden‘, die die Leistungsfähigkeit vom UNO R3-Board erweitern können, verwendet werden.

Es hat auch eine einzelne LED, die Sie von Ihren Sketchen kontrollieren können. Diese LED ist auf dem UNO R3-Board aufgebaut und wird oft als ‚L-LED‘ bezeichnet, weil dies ist, wie sie auf dem Board gekennzeichnet wird.



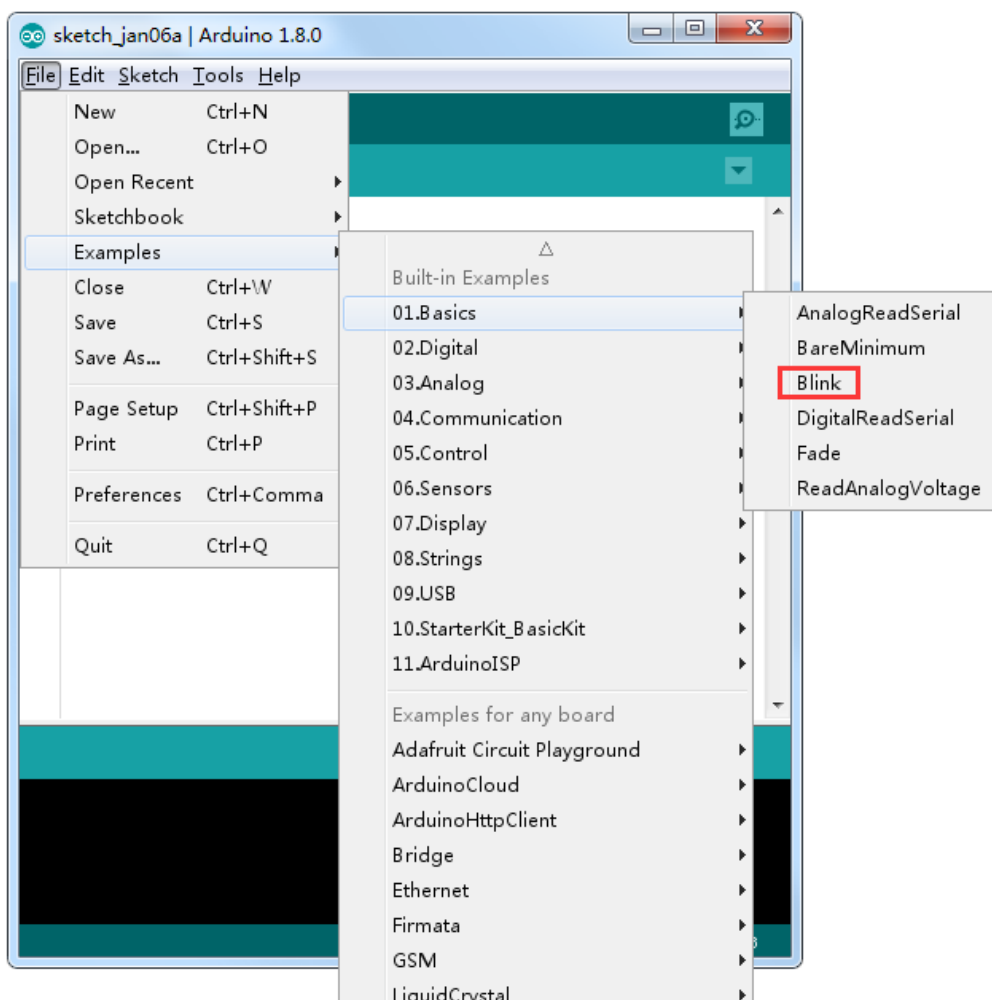
Sie könnten finden, dass die ,L'-LED Ihres UNO R3-Boards bereits blinkt, wenn Sie sie an einen USB-Stecker anschließen. Der Grund liegt darin, dass die Boards, die mit dem ,Blinken'-Sketch ausgestattet sind, üblicherweise vorinstalliert sind.

In dieser Lektion werden wir das UNO R3-Board mit unserem eigenen Blinken-Sketch umprogrammieren, und dann werden wir die Geschwindigkeit, bei der es blinkt, ändern.

In der Lektion 0 richten Sie Ihre Arduino-IDE ein, und stellen Sie sicher, dass Sie den richtigen seriellen Anschluss finden können, so dass er an Ihr UNO R3-Board angeschlossen werden kann. Nun ist die Zeit gekommen, dass die Verbindung zum Test Ihnen zur Verfügung stellt und dass Ihr UNO R3-Board programmiert werden kann.

Die Arduino-IDE enthält eine riesige Sammlung von beispielhaften Sketchen, die Sie hochladen und verwenden können. Darunter wird ein beispielhafter Sketch für das Blinken der ,L'-LED enthalten.

Laden Sie den ,Blinken'- Sketch, den Sie in dem Menüsystem der IDE unter File > Examples > 01.Basics finden werden.



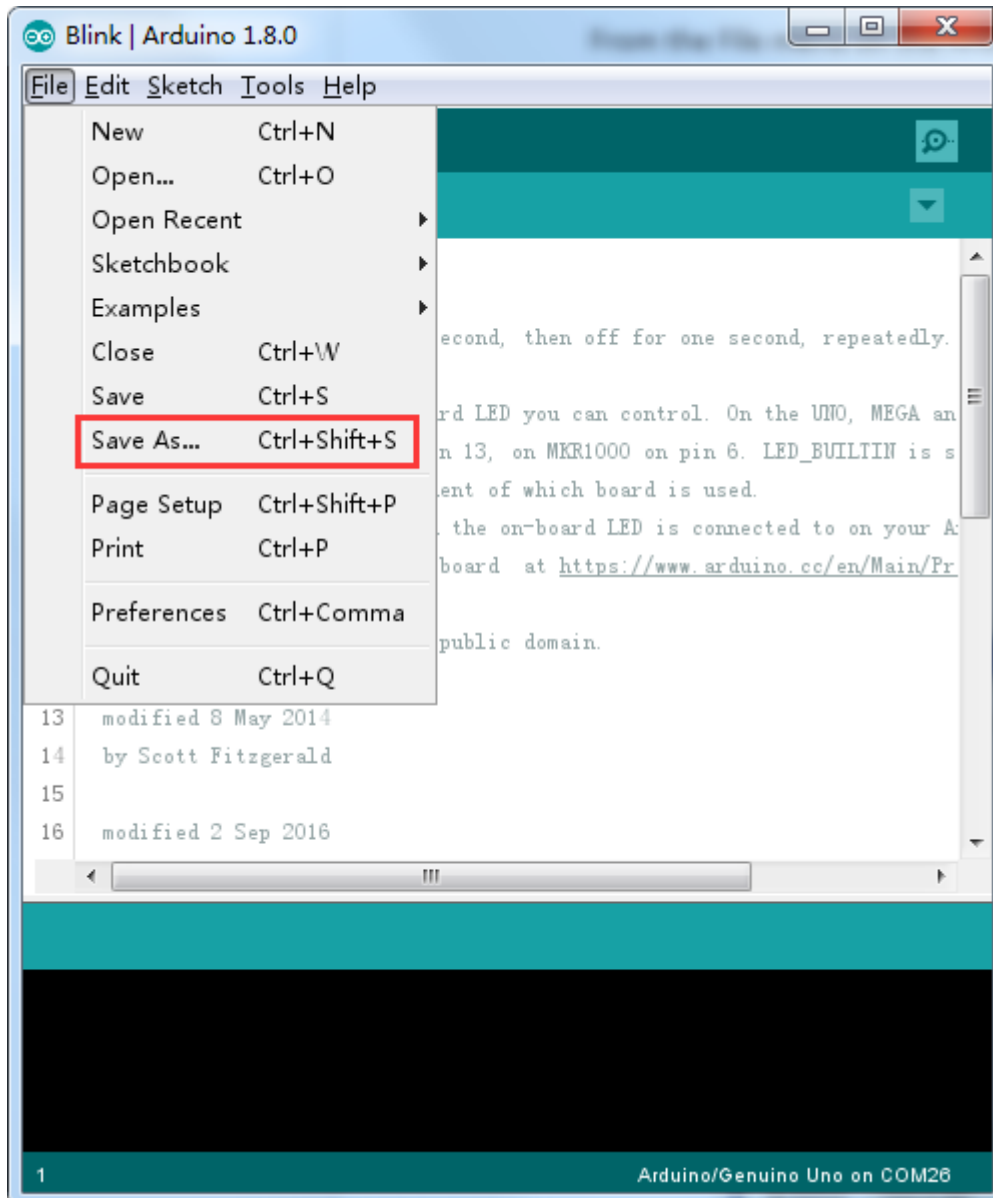
Wenn das Fenster vom Sketch geöffnet wird, erweitern Sie es, so dass Sie den gesamten Sketch in dem Fenster sehen.



Die Beispielsketches mit der Arduino-IDE sind ‚schreibgeschützt‘. Das heißt, dass Sie sie auf ein UNO R3-Board hochladen können, aber falls Sie sie ändern, können Sie sie nicht als die gleiche Datei speichern.

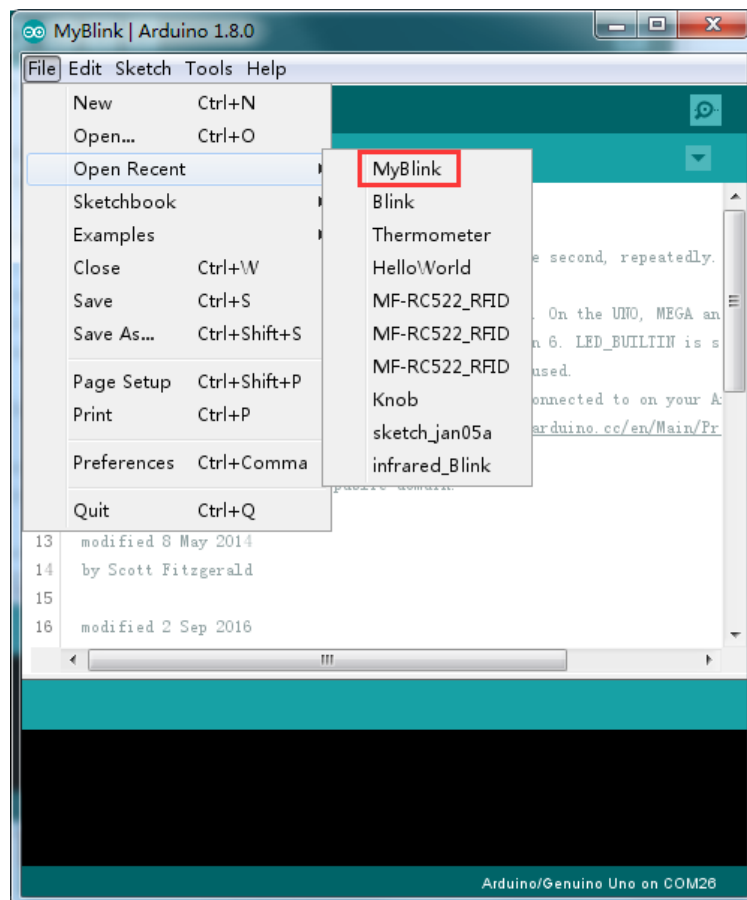
Weil wir den Sketch ändern werden, müssen Sie zunächst Ihre eigene Kopie speichern.

Im Datei-Menü auf der Arduino-IDE wählen Sie ‚Save As...‘ aus, und dann speichern Sie den Sketch mit der Bezeichnung ‚MyBlink‘.



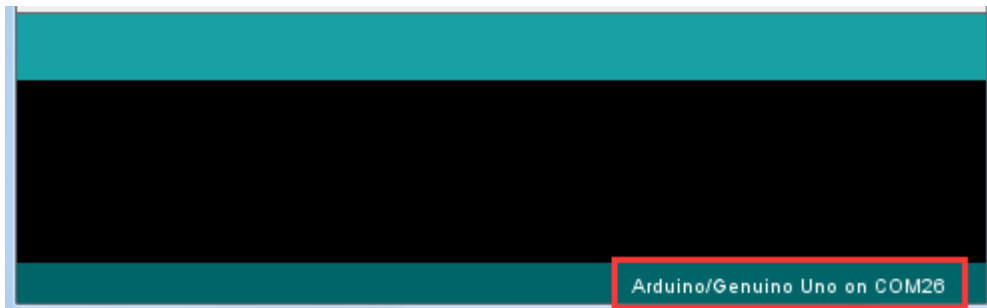


Sie haben Ihre Kopie von ‚Blinken‘ in Ihrem Sketchbook gespeichert. Dies bedeutet, wenn Sie sie wieder finden möchten, können Sie sie nur mit der Menüoption File > Sketchbook öffnen



Fügen Sie Ihr Arduino-Board mit dem USB-Kabel Ihrem Computer bei, und überprüfen Sie, ob der ‚Board-Typ‘ und der ‚Serielle Anschluss‘ richtig eingestellt sind. Ggf. sollten Sie auf die Lektion 0 verweisen.

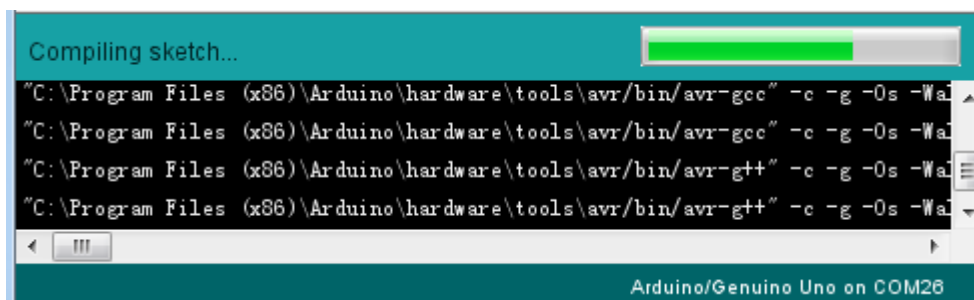
Die Arduino-IDE wird Ihnen die aktuellen Einstellungen für Board am unteren Rand des Fensters anzeigen.



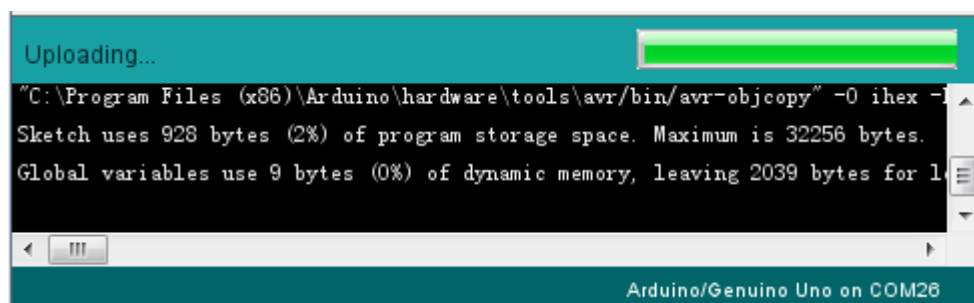
Klicken Sie auf die Taste ‚Hochladen‘. Sie ist die zweite Taste von links in der Symbolleiste.



Wenn Sie den Statusbereich der IDE beobachten, werden Sie einen Fortschrittsbalken und eine Reihe von Meldungen sehen. Zunächst wird ‚Compiling Sketch...‘ angegeben. Dies wandelt den Sketch in ein Format, das für das Hochladen des Boards angemessen ist.



Anschließend wird der Status in ‚Hochladen‘ umgewandelt. Auf diesem Punkt sollte die LED auf Arduino mit der Wechselung des Sketches zu flackern beginnen.



Schließlich wird der Status in den Status ‚Hochladen Beendet‘ wechseln. Eine Meldung wird erscheinen, was anzeigt, dass 1.084 Bytes von den 32.256 Bytes aktuell für den Sketch verfügbar sind.

```
Done uploading.
"C:\Program Files (x86)\Arduino\hardware\ttools\avr\bin\avr-objcopy" -O ihex -
Sketch uses 928 bytes (2%) of program storage space. Maximum is 32256 bytes.
Global variables use 9 bytes (0%) of dynamic memory, leaving 2039 bytes for 1
Arduino/Genuino Uno on COM26
```

Bitte beachten Sie, dass Sie in der Phase ‚Compiling Sketch...‘ vielleicht die folgende Fehlermeldung erhalten:

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

Dies kann bedeuten, dass Ihr Board gar nicht angeschlossen ist, oder dass die Treiber nicht installiert sind (falls erforderlich), oder dass der falsche serielle Anschluss ausgewählt wurde.

In diesem Fall gehen Sie zur Lektion 0 zurück und überprüfen Sie Ihre Installation.

Sobald der Prozesse des Hochladens abgeschlossen ist, sollte das Board erneut starten und zu blinken beginnen.

Öffnen Sie den Code

Beachten Sie, dass ein großer Teil dieses Sketches aus Kommentaren besteht. Sie sind keine tatsächlichen Programmanweisungen; vielmehr erklären sie nur, wie das Programm funktioniert. Von ihnen können Sie profitieren.

Alles zwischen /* und */ an der Spitze des Sketches ist ein Block-Kommentar; er erklärt, wozu der Sketch ist.

Einzeiliger Kommentar, der mit // und allem bis zum Ende der Zeile beginnt, wird als ein Kommentar betrachtet.

Code in der ersten Zeile ist:

Code kopieren

```
1. int led = 13;
```

Wie der vorstehende Kommentar erklärt es, dass eine Bezeichnung dem Pin, an dem die LED angebracht ist, gegeben wird. Auf den meisten Arduinos einschließlich UNO und Leonardo ist dies 13.

Als nächstes erhalten wir die ‚Setup‘-Funktion. Noch einmal, wie der Kommentar angibt, dass dies ausgeführt

wird, wenn die Reset-Taste gedrückt wird. Dies wird auch ausgeführt, wenn das Board aus irgendeinem Grund zurückgesetzt wird, wie z.B. das erste Anlegen einer Spannung an das Board, oder nachdem ein Sketch hochgeladen wurde.

Code kopieren

1. Ungültiges Einrichten() {
2. // initialisieren Sie den digitalen Pin als einen Ausgang
3. pinMode(led, AUSGANG);
4. }

Jeder Arduino-Sketch muss eine ‚Setup‘-Funktion aufweisen, und die Position, wo Sie vielleicht Ihre eigenen Anweisungen hinzufügen möchten, liegt zwischen dem/der {und dem/der}.

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

Es ist auch für einen Sketch zwingend, eine ‚Loop‘-Funktion aufzuweisen. Anders als die ‚Setup‘-Funktion, die nur einmal ausgeführt wird, wird die ‚Loop‘-Funktion nach einem Reset sofort wieder starten, nachdem sie die Ausführung ihrer Befehle beendet hat.

Code Kopieren

1. Ungültige Schleife () {
2. digitalWrite(led, HIGH); // Schalten Sie die LED ein (HIGH ist der Spannungspegel)
3. Verzögerung (1000); // Warten Sie auf eine Sekunde
4. digitalWrite(led, LOW); // Schalten Sie die LED aus, indem Sie die Spannung LOW einstellen
5. Verzögerung (1000); // Warten Sie auf eine Sekunde
6. }

Innerhalb der Loop-Funktion wird der LED-Pin zuerst durch die Befehle eingeschaltet (HOCH), dann ‚verzögert‘ es für 1000 Millisekunden (1 Sekunde), dann schalten Sie den LED-Pin aus, und warten Sie auf eine weitere Sekunde.

Nun werden Sie Ihre LED schneller blinken lassen. Wie Sie vielleicht bereits erraten haben, liegt der Schlüssel dazu in Verwechslung des Parameters in () für den ‚Verzögerung‘-Befehl.

```
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 Verzögerungsphase dauert Millisekunden, und wenn Sie die LED doppelt so schnell blinken lassen, stellen Sie den Wert von 1000 auf 500 ein. So wird eine Verzögerungszeit auf eine halbe Sekunde dauern, anstatt einer ganzen Sekunde zu dauern.

Laden Sie den Sketch wieder hoch, und Sie sollten sehen, dass die LED schneller zu blinken beginnt.

Lektion 3: LED

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie die Helligkeit einer LED ändern, indem Sie verschiedene Werte des Widerstands verwenden.

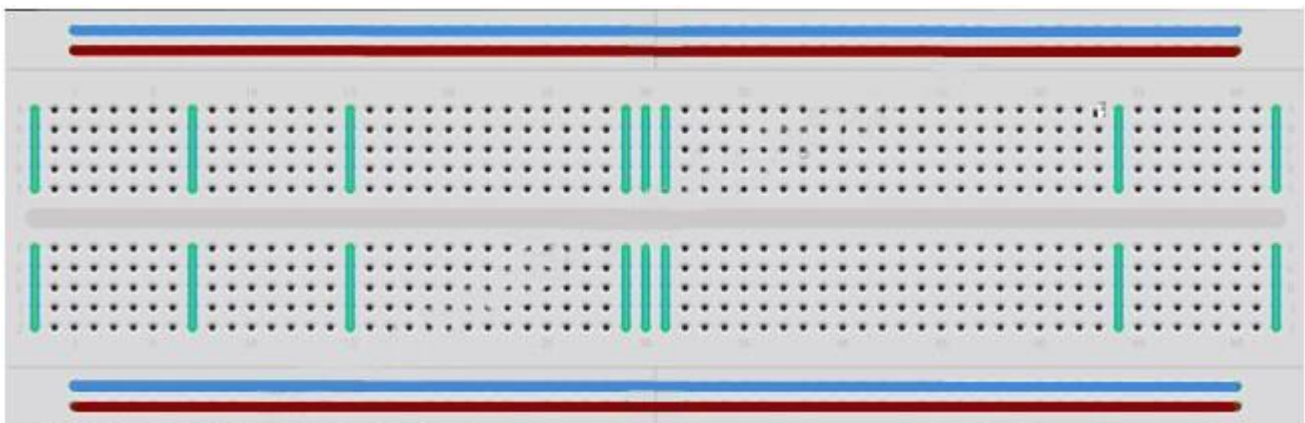
Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) 5mm rote LED
- (1) 220 Ohm-Widerstand
- (1) 1k Ohm-Widerstand
- (1) 10k Ohm-Widerstand
- (2) M-M Kabel

Komponentenanweisung

BREADBOARD MB-102:

Ein Versuchsaufbau ermöglicht es Ihnen, Prototypen der Schaltungen schnell herzustellen, ohne dass Sie die Anschlüsse löten. Nachstehend ist ein Beispiel.



Versuchsaufbaue weisen verschiedene Größen und Konfigurationen auf. Die einfachste Art ist nur ein Raster von Löchern in einem Kunststoffblock. Im Inneren sind Metallstreifen, die elektrische Verbindung zwischen Löchern in den kürzeren Reihen bereitstellen. Schieben Sie die Schenkel von zwei unterschiedlichen Komponenten in die gleiche Reihe, um sie elektrisch miteinander zu verbinden. Ein tiefer Kanal läuft in der Mitte nach unten, was anzeigt, dass es dort einen Bruch in Verbindungen

gibt, das heißt, dass Sie einen Chip einstecken können, ohne dass die Beine auf einer Seite des Kanals nicht miteinander verbunden sind. Einige Versuchsaufbaue weisen zwei Streifen von Löchern entlang der langen Kanten des Aufbaus, die vom Hauptnetz getrennt werden, auf. Sie weisen Streifen, die über die Länge des Aufbaus verlaufen und einen Weg für die Verbindung der gemeinsamen Spannung bereitstellen, auf. Sie sind normalerweise paarweise für +5 Volt und die Masse. Diese Streifen werden als Schienen bezeichnet, und sie ermöglichen es Ihnen, Strom mit vielen Komponenten oder Punkten in dem Aufbau zu verbinden.

Während Versuchsaufbaue für Prototypen toll sind, haben sie aber einige Einschränkungen. Weil die Verbindungen steckbar und zeitlich begrenzt sind, ist dies unzulässig, sie als Lötverbindungen verwendet zu werden. Wenn Sie intermittierende Probleme mit einer Schaltung haben, könnte dies durch die schlechte Verbindung auf dem Versuchsaufbau verursacht werden.

LED:

LEDs erzeugen tolle Anzeigelichte. Sie verbrauchen sehr wenig Strom, aber dauern so ziemlich für immer.

In dieser Lektion werden Sie vielleicht alle häufigsten LEDs verwenden: a 5mm rote LED. 5mm bezieht sich auf den Durchmesser der LED. Weitere allgemeine Größen sind 3mm und 10mm.

Sie können die LED nicht unmittelbar mit einer Batterie oder einer Spannungsquelle verbinden, weil 1) die LED eine positive und eine negative Leitung aufweist, falls sie auf eine falsche Weise angeschlossen werden, wird die LED nicht leuchten, und 2) eine LED mit einem Widerstand für die Begrenzung oder das ‚Drosseln‘ der Menge an Strom, der durch sie fließt, ausgestattet sein muss; andernfalls wird sie ausbrennen!



Wenn Sie keinen Widerstand für eine LED verwenden, könnte sie fast sofort zerstört werden, weil zu viel Strom durch sie fließt, sie erhitzt und die ‚Anschlussstelle‘, wo das Licht erzeugt wird, zerstört.

Es gibt zwei Möglichkeiten, um anzugeben, welche die positive Leitung der LED ist und welche die negative ist.

Erstens ist die positive Leitung länger.

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

Wenn Sie gerade eine LED, deren flache Kante neben der längeren Leitung steht, haben, sollten Sie davon ausgehen, dass die längere Leitung die positive Leitung ist.

WIDERSTÄNDE:

Wie die Bezeichnung verrät, widerstehen Widerstände dem Stromfluss. Je höher der Wert des Widerstands, desto mehr widersteht der Widerstand dem Stromfluss, daher wird weniger elektrischer Strom durch ihn fließen. Wir werden diesen Grundsatz verwenden, um zu steuern, wie viel Strom durch die LED fließt und wie hell sie strahlt.

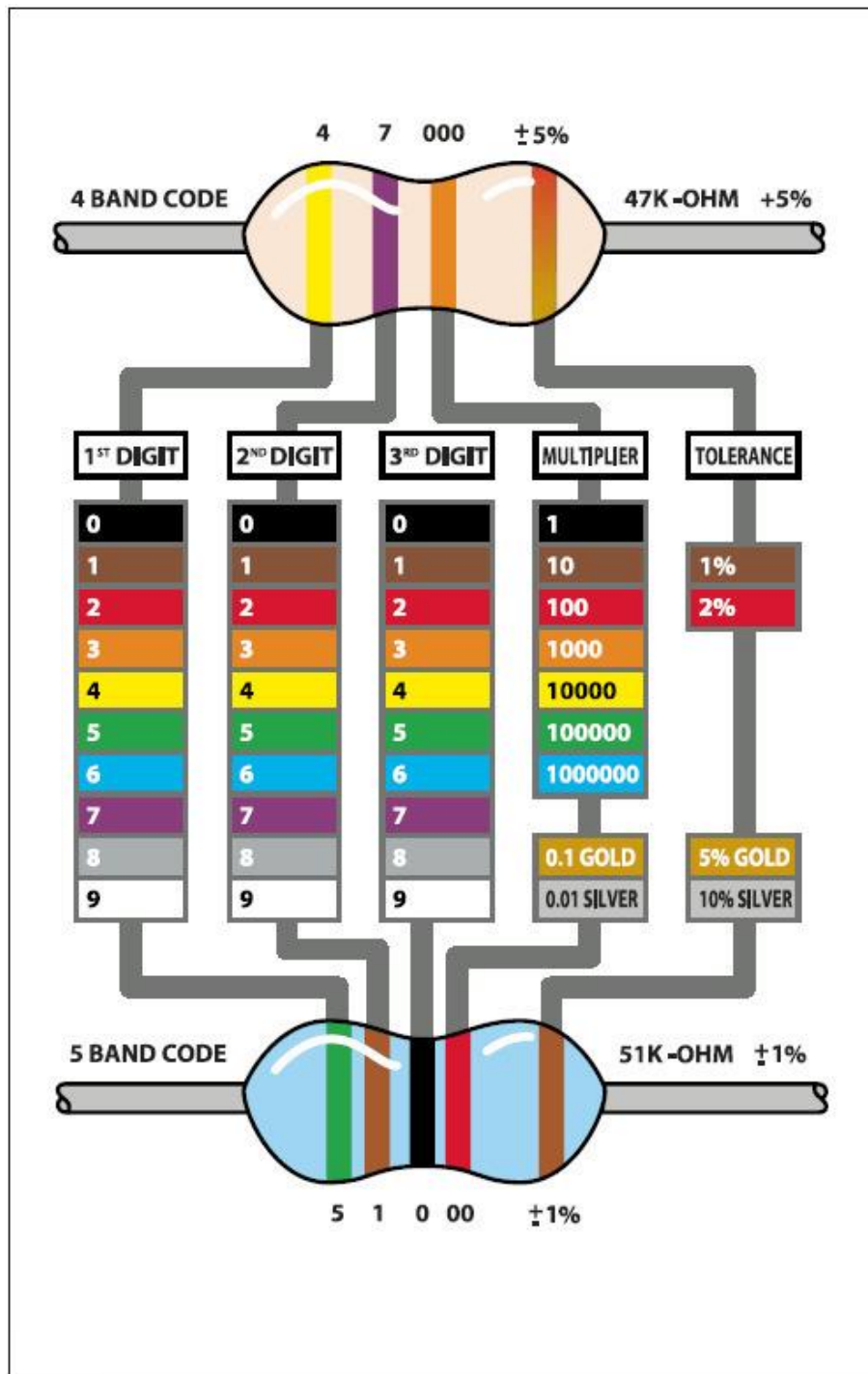


Aber zunächst mehr über Widerstände...

Die Einheit des Widerstands wird als Ohm, das in der Regel als Ω , den griechischen Buchstaben Omega, abgekürzt wird, bezeichnet. Weil ein Ohm ein niedriger Wert von Widerstand ist (es widersteht überhaupt nicht viel Strom), kennzeichnen wir die Werte der Widerstände auch in $k\Omega$ (1.000Ω) und $M\Omega$ ($1.000.000\Omega$). Sie werden als Kilo-Ohm und Mega-Ohm bezeichnet.

In dieser Lektion werden wir verschiedene Werte von Widerstand verwenden: 220Ω , $1k\Omega$ und $10k\Omega$. Alle diese Widerstände sehen gleich aus, mit der Ausnahme, dass Streifen mit verschrienen Farben auf ihnen angebracht werden. Diese Streifen können Ihnen den Wert des Widerstands angeben.

Der Farbcode des Widerstands weist drei farbige Streifen und einen Goldstreifen an einem Ende auf.

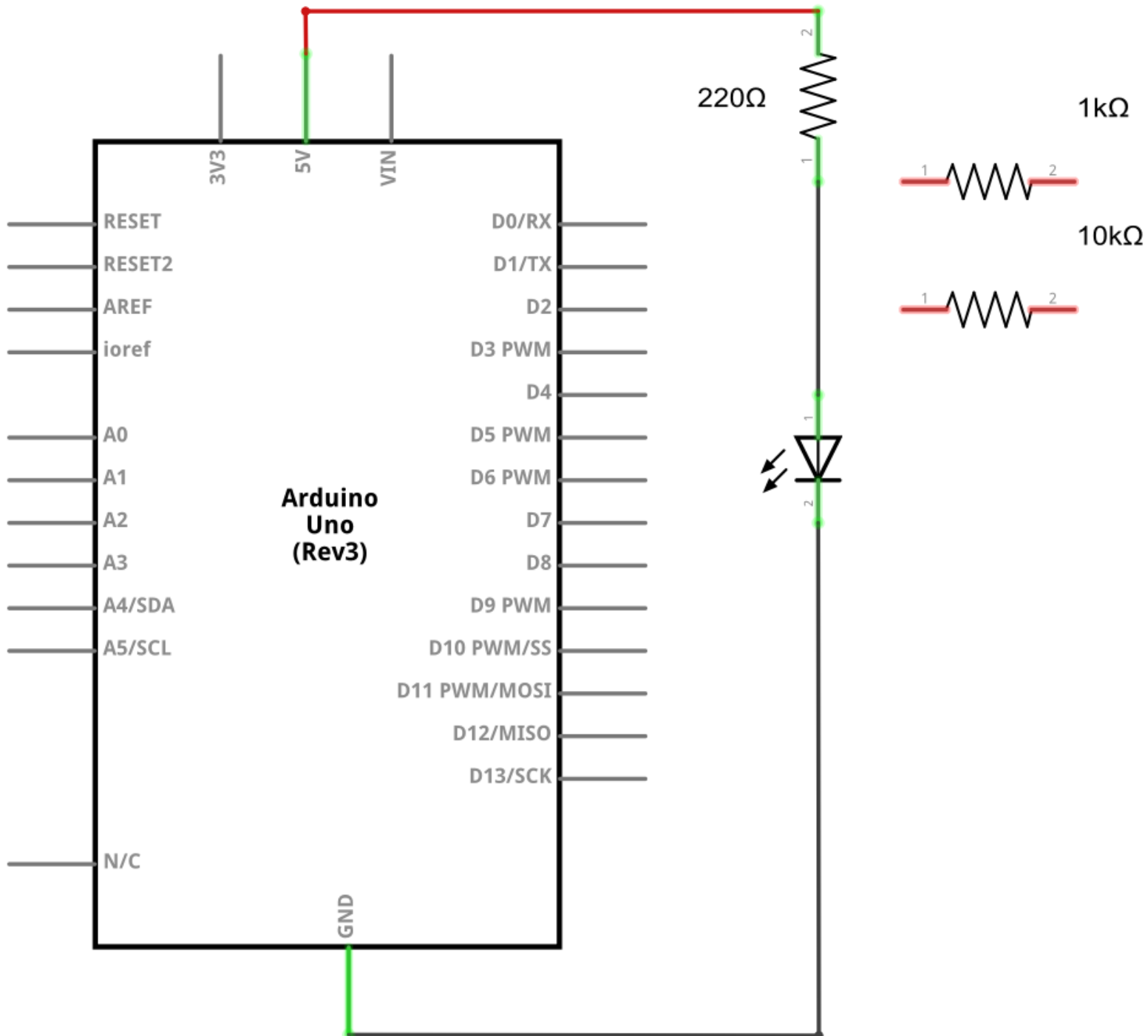


Anders als LEDs weisen die Widerstände keine positive oder negative Leitung auf. Sie können beliebig angeschlossen werden.

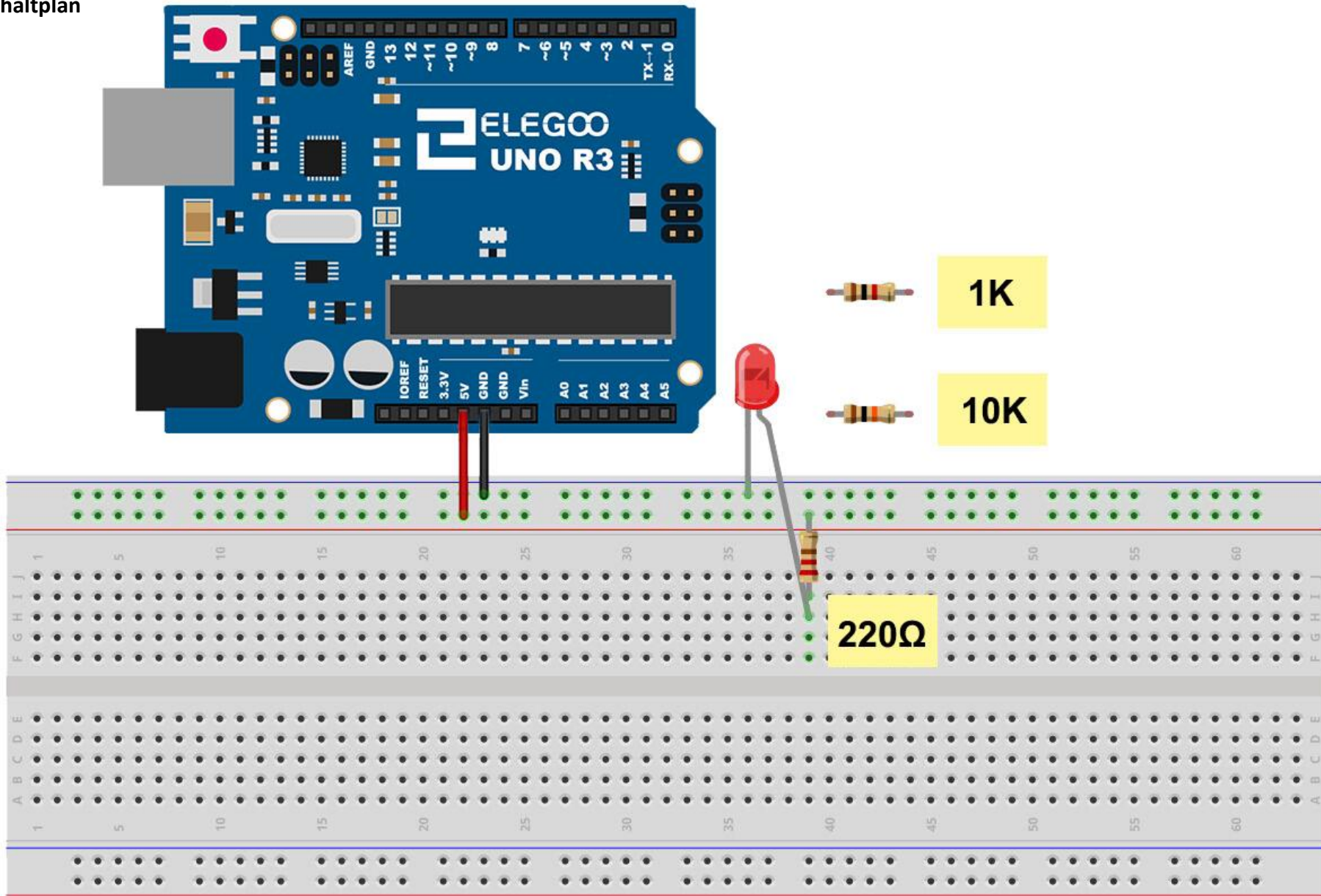
Wenn Sie diese Methode zu kompliziert finden, lesen Sie den Farbe-Ring auf unsere Widerstände direkt an seinen Widerstandswert zu bestimmen. Oder Sie können stattdessen ein digital-Multimeter.

Verbindung

Schema



Schaltplan



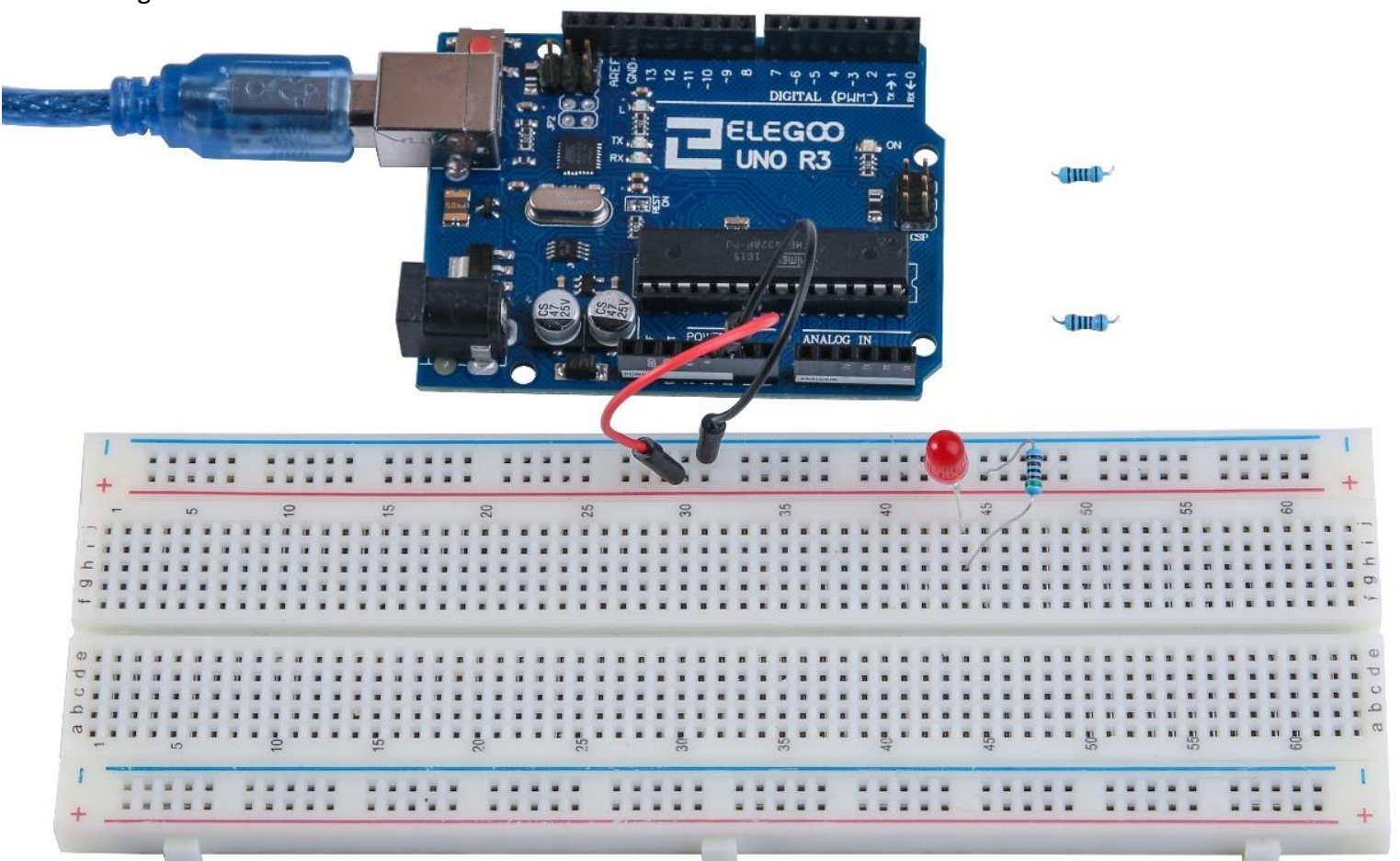
Das UNO ist eine bequeme Quelle von 5 Volt, die wir verwenden werden, um die LED und den Widerstand mit Strom zu versorgen. Für Sie ist es nicht erforderlich, etwas mit Ihrem UNO zu tun, außer dass Sie ihm in ein USB-Kabel einstecken sollten.

Mit dem Vorhandensein des 220Ω-Widerstands sollte die LED sehr hell sein. Wenn der 220Ω-Widerstand durch den 1kΩ-Widerstand ersetzt wird, wird die LED ein wenig dunkler sein. Schließlich setzen Sie den 10kΩ-Widerstand an der Stelle, wird die LED nur sichtbar sein. Ziehen Sie das rote Überbrückungskabel aus dem Versuchsaufbau heraus, berühren Sie es in das Loch, und entfernen Sie es, so dass es wie ein Schalter wirkt. Sie sollten nur in der Lage sein, den Unterschied zu bemerken.

Zurzeit geht 5V auf einem Schenkel des Widerstands, geht der andere Schenkel des Widerstands auf der positiven Seite der LED, und geht die andere Seite der LED nach GND. Aber wenn wir den Widerstand entfernen, so dass er hinter der LED kommt, wie unten gezeigt, wird die LED noch leuchten.

Sie möchten wahrscheinlich den Widerstand 220Ω wieder an seine Stelle setzen.

Es spielt keine Rolle, auf welche Seite der LED wir den Widerstand setzen, solange er sich dort irgendwo befindet.



Lektion 4: RGB LED

Übersicht

RGB LEDs liefern eine lustige und einfache Möglichkeit, Ihren Programmen einige Farben zu verleihen. Da sie wie 3 übliche LEDs in einer aussehen, ist es deshalb nicht viel anders, wie sie verwendet und verbunden werden.

Sie weisen meistens 2 Versionen auf: allgemeine Anode oder allgemeine Kathode.

Allgemeine Anode benötigt 5 V auf dem allgemeinen Pin, während allgemeine Kathode wird mit der Masse verbunden.

Wie bei jeder LED müssen wir einige Widerstände inline anschließen (insgesamt 3), so können wir den verbrauchten Strom begrenzen.

In unserem Sketch werden wir mit der LED in dem Status der roten Farbe starten, dann verblasst die rote Farbe zu der blauen Farbe, und schließlich wieder zu der roten Farbe. In dieser Vorgehensweise werden wir den meisten der Farben, die erreicht werden können, durchlaufen.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Versuchsaufbau
- (4) M-M Kabel
- (1) RGB LED
- (3) 220 Ohm-Widerstände

Komponentenanweisung

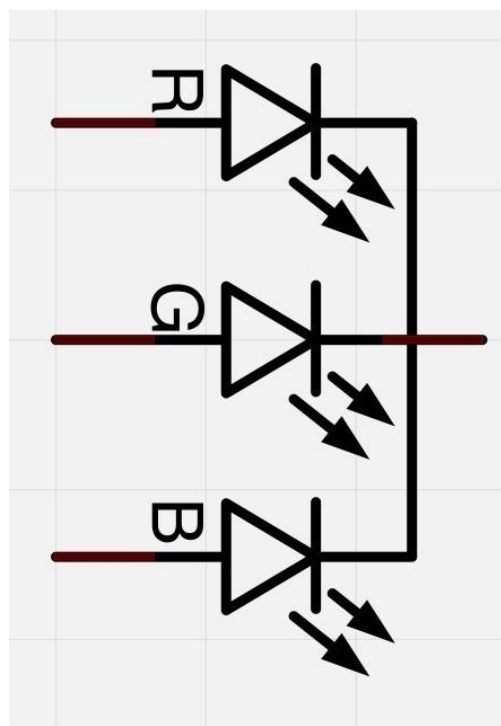
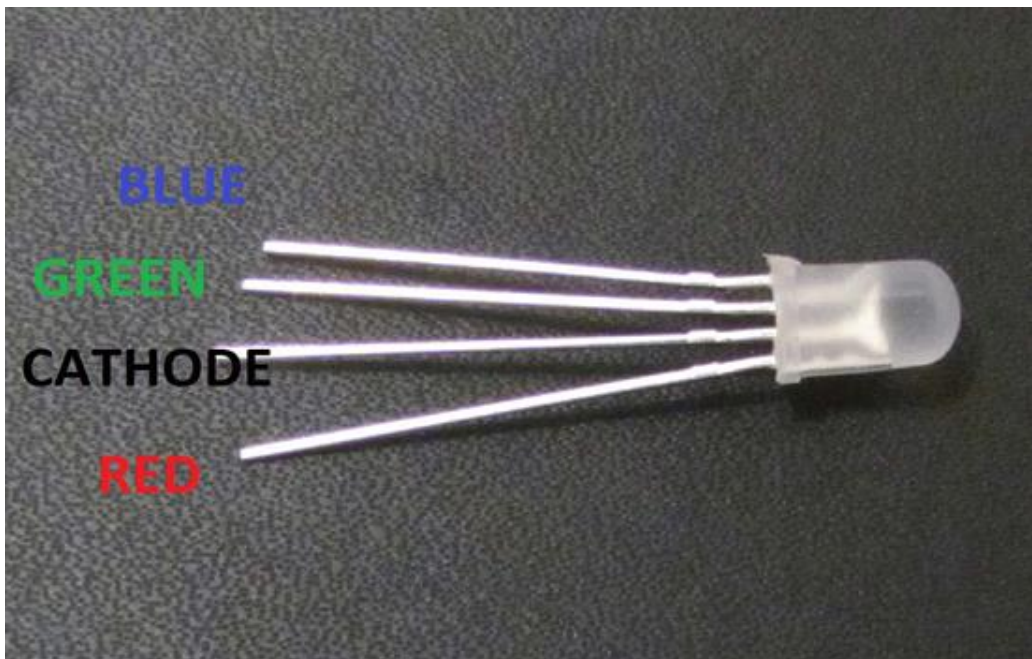
RGB:

Auf den ersten Blick sehen RGB (Rot, Grün, Blau)-LEDs nur wie normale LEDs aus. Jedoch gibt es innerhalb des üblichen LED-Pakets eigentlich drei LEDs, eine rote, eine grüne und ja, eine blaue LED. Durch die Steuerung der Helligkeit von jeder der einzelnen LEDs können Sie fast jede Farbe, die Sie wünschen, mischen.

Wir mischen Farben auf dieselbe Weise wie Sie Farben auf einer Palette mischen - durch Einstellung der Helligkeit von jeder der drei LEDs. Die schwierige Sache in diesem Prozess könnte die Verwendung

von Widerständen mit verschiedenen Werten (oder variablen Widerständen) sein, was wir in der Lektion 2 behandelt haben, aber viel Arbeit ist zu erledigen! Zum Glück für uns verfügt UNO R3-Board über eine analogWrite-Funktion, die Sie verwenden können, um mit Pins ~Ausgang einer variablen Menge an Strom auf angemessene LEDs zu kennzeichnen.

Die RGB LED hat vier Leitungen. Eine Leitung geht zum positiven Anschluss jeder der einzelnen LEDs innerhalb des Pakets, und eine einzelne Leitung ist an alle drei negativen Seiten der LEDs angeschlossen.



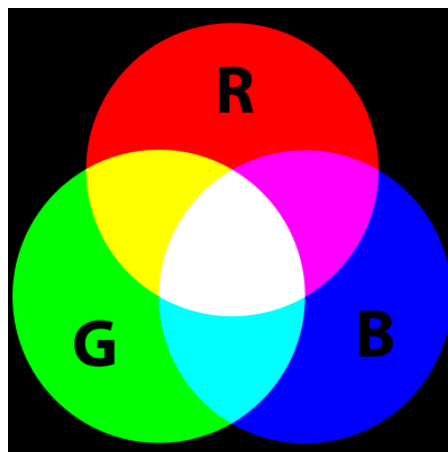
Der gemeinsame negative Anschluss des LED-Pakets ist der zweite Pin auf der flachen Seite. Seine Leitung ist auch die längste der vier Leitungen und wird mit der Masse verbunden.

Jede LED innerhalb des Pakets erfordert ihren eigenen Widerstand 270Ω , um zu verhindern, dass zu viel Strom durch sie fließt. Die drei positiven Leitungen der LEDs (eine rote, eine grüne und eine blaue) sind durch diese Widerstände an die UNO-Ausgangspins angeschlossen.

FARBE:

Der Grund, dass Sie jede Farbe, die Sie wünschen, durch Variation der Mengen von rotem, grünem und blauem Licht mischen, liegt darin, dass Ihre Augen drei Arten von Lichtempfänger haben (Rot, Grün und Blau). Ihre Augen und Ihr Gehirn verarbeiten die Mengen von Rot, Grün und Blau, und wandeln sie in eine Farbe des Spektrums um.

Durch die Verwendung der drei LEDs spielen wir gewissermaßen einen Trick auf den Augen. Diese gleiche Idee wird in Fernsehern verwendet, in denen LCD rote, grüne und blaue Farbpunkte nebeneinander hat, was jedes Pixel bildet.



Wenn wir die Helligkeit aller drei LEDs gleich einstellen, wird die Farbe des Lichtes überall Weiß sein. Wenn wir die blaue LED ausschalten, so dass nur die roten und grünen LEDs die gleiche Helligkeit haben, wird daher das Licht Gelb erscheinen.

Wir können die Helligkeit von jeder der roten, grünen und blauen Teile der LED getrennt steuern, so dass es möglich ist, jede Farbe, die wir wünschen, zu mischen.

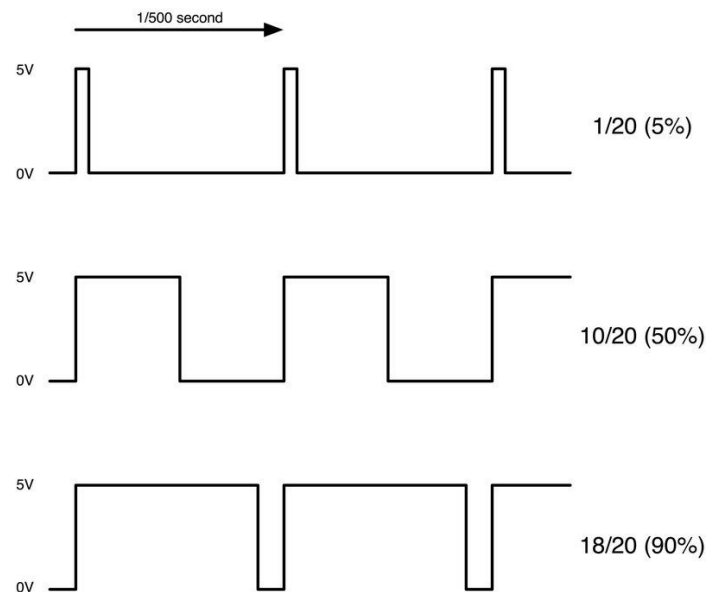
Schwarz ist eine Farbe wie Abwesenheit von Licht. Dadurch können wir mit unserer LED alle drei Farben ausschalten, um dem Schwarz am nächsten zu erhalten.

PWM Theorie

Pulsweitenmodulation (PWM) ist eine Technik für Leistungssteuerung. Ebenfalls verwenden Sie sie

hier, um die Helligkeit jeder der LEDs zu steuern.

Das folgende Diagramm zeigt das Signal aus einer der PWM-Pins auf dem UNO an.



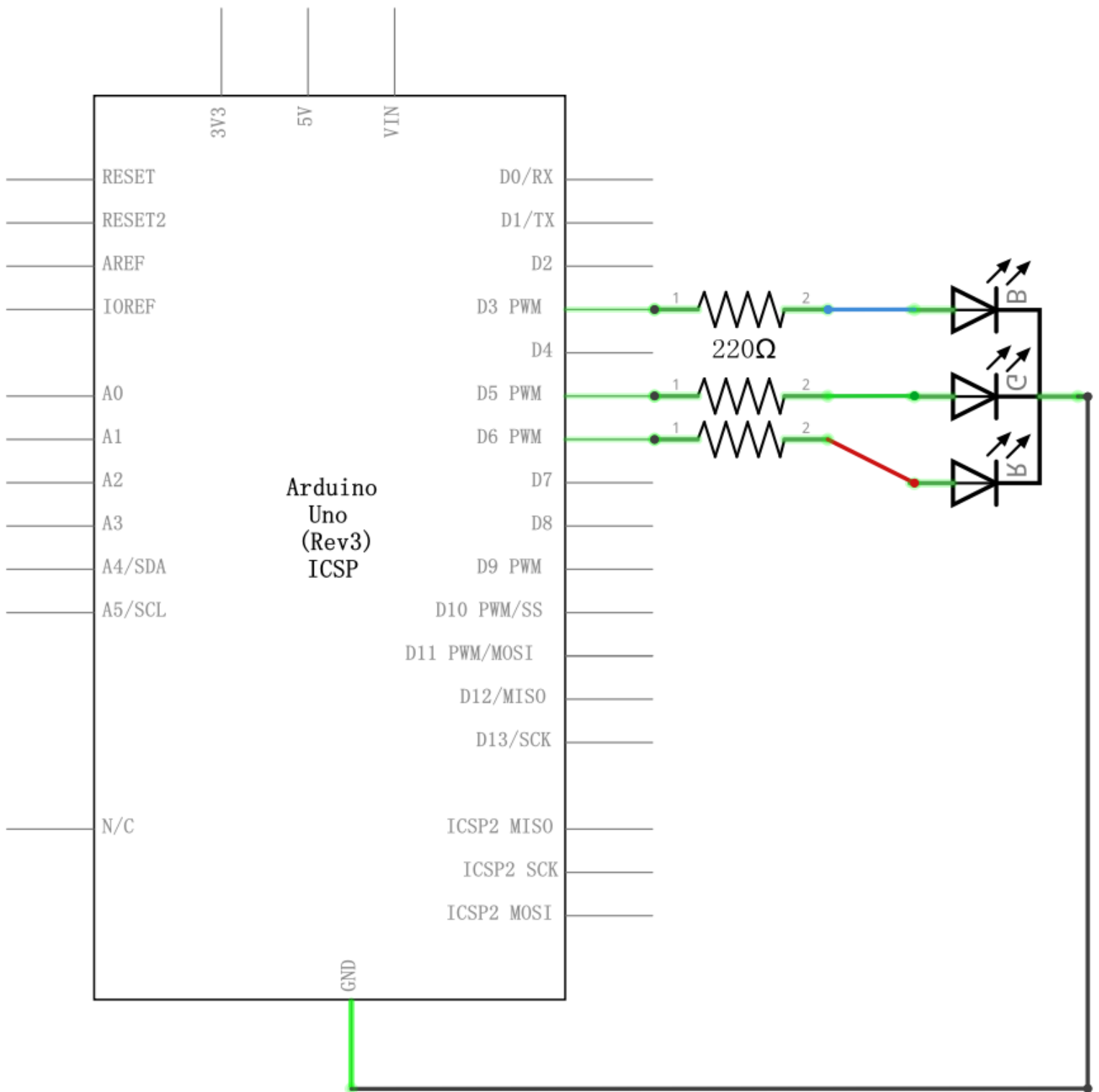
Ungefähr alle 1/500 einer Sekunde wird der PWM-Ausgang einen Impuls erzeugen. Die Länge dieses Impulses wird durch ‚analogWrite‘-Funktion gesteuert. So wird ‚analogWrite(0)‘ gar keinen Impuls erzeugen, und ‚analogWrite(255)‘ wird einen Impuls, der dauert, bis der nächsten Impuls erzeugt wird, erzeugen, so dass der Ausgang tatsächlich immer eingeschaltet ist.

Wenn wir einen Wert, der irgendwo zwischen 0 und 255 liegt, in analogWrite festlegen, werden wir einen Impuls erhalten. Wenn der Ausgangsimpuls nur für 5% der Zeit hoch ist, werden wir nur 5% der vollen Leistung erhalten, unabhängig davon, was wir betreiben.

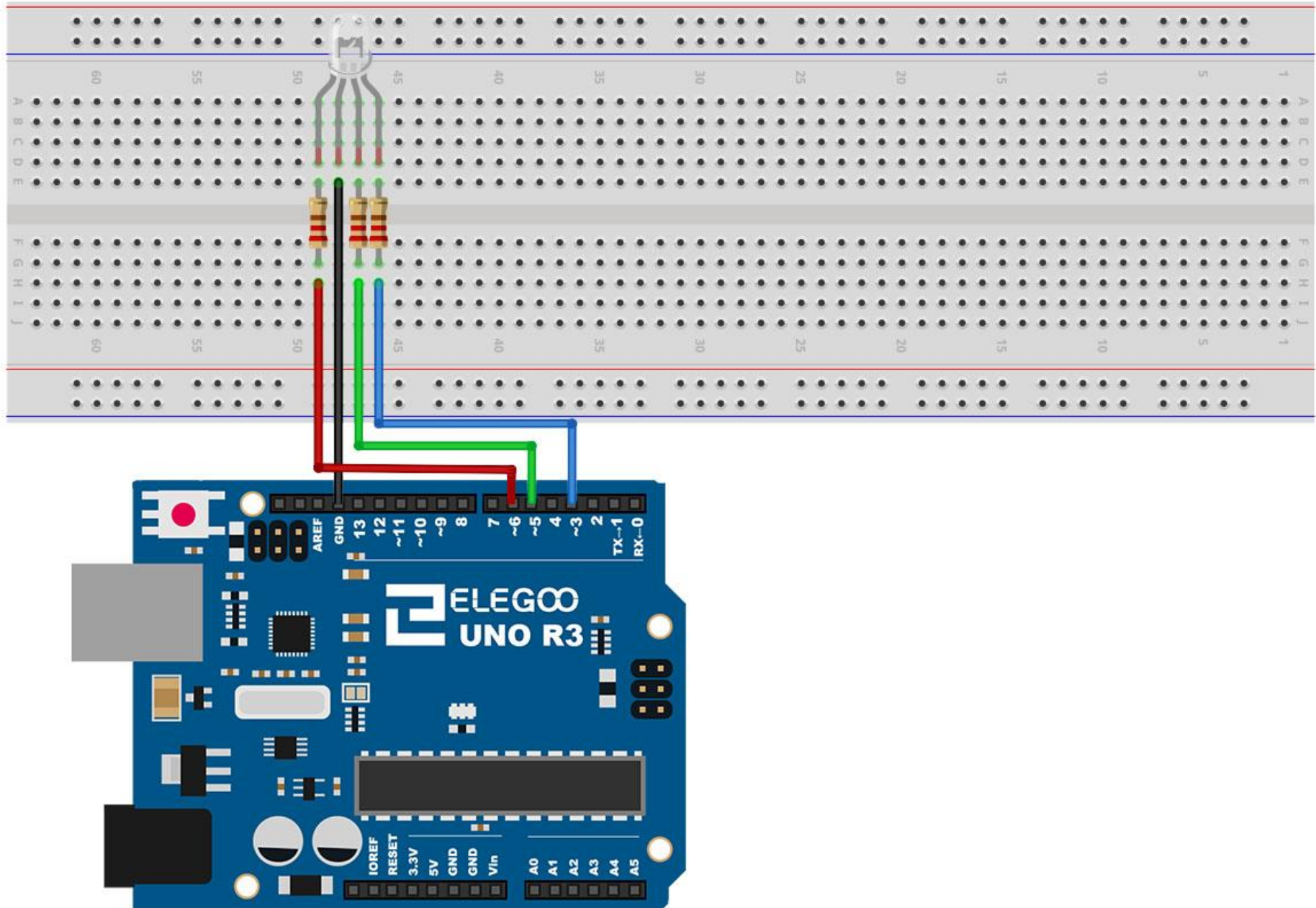
Wenn jedoch der Ausgang bei 5V für 90% der Zeit liegt, wird die Last 90% ihrer gelieferten Leistung erhalten. Bei dieser Geschwindigkeit können wir Ein- und Ausschalten der LEDs nicht sehen, so sieht es für uns nur so aus, dass die Helligkeit geändert wird.

Verbindung

Schema



Schaltplan



Code

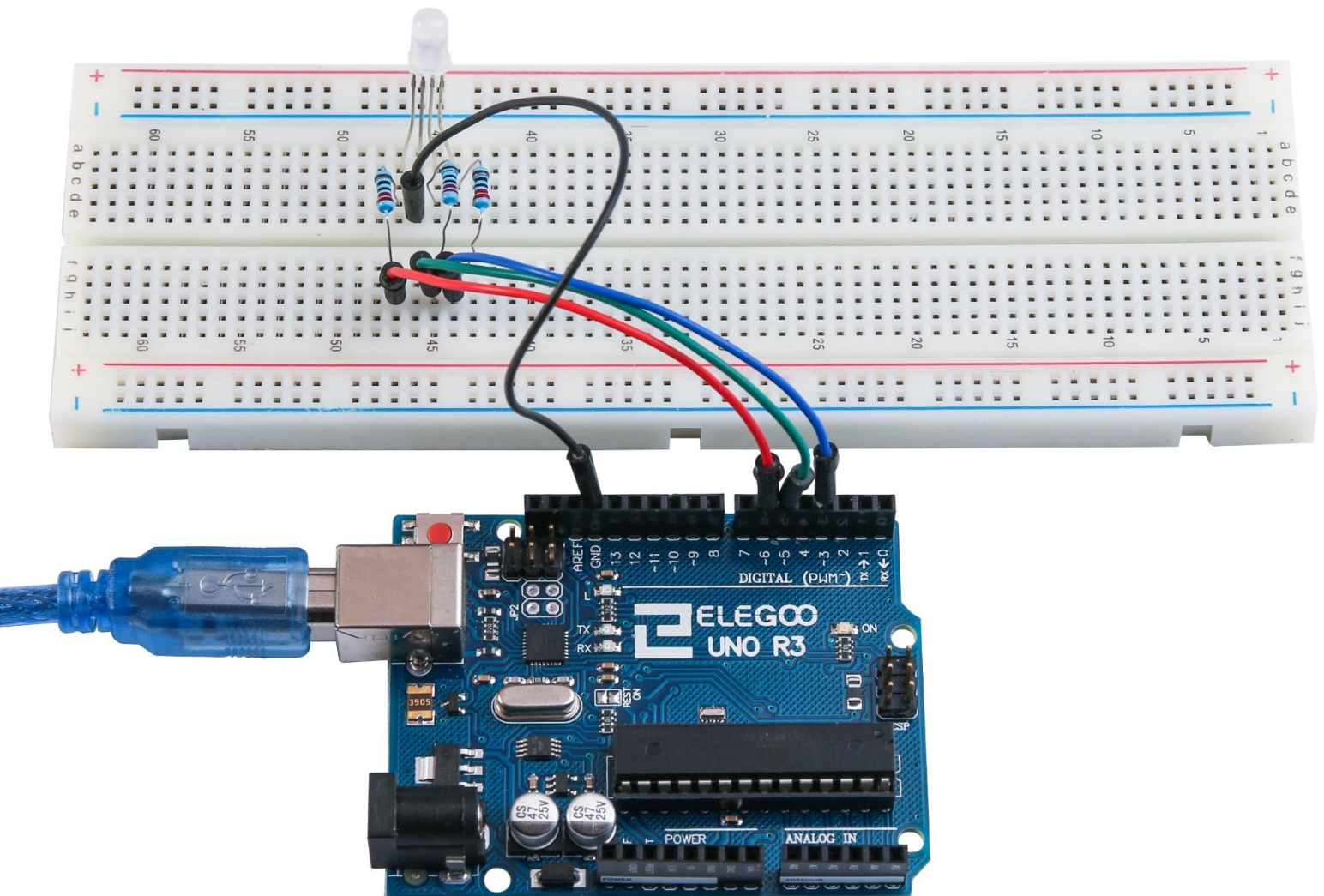
Nach der Verdrahtung, bitte öffnen Sie das Programm in die Code-Ordner-Lektion 4-RGB-LED, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefodert werden, siehe [Lektion 2 Einzelheiten zum Programm hochladen](#).

Unser Code wird FOR-Schleifen verwenden, um die Farben durchzulaufen.

Die erste FOR-Schleife wird von ROT auf GRÜN wechseln.

Die zweite FOR-Schleife wird von GRÜN auf BLAU wechseln.

Die letzte FOR-Schleife wird von BLAU auf ROT wechseln.



Lektion 5: Digitale Eingänge

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie Drucktasten mit digitalen Eingängen verwenden, um eine LED ein- und auszuschalten.

Drücken Sie die Taste näher an der Spitze des Versuchsaufbaus, so wird die LED eingeschaltet; drücken Sie die andere Taste, so wird die LED ausgeschaltet.

Erforderliche Komponenten

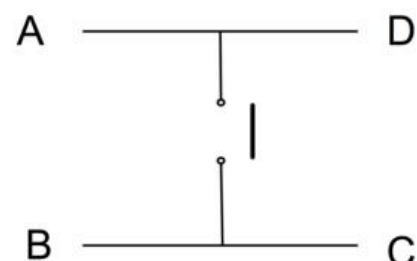
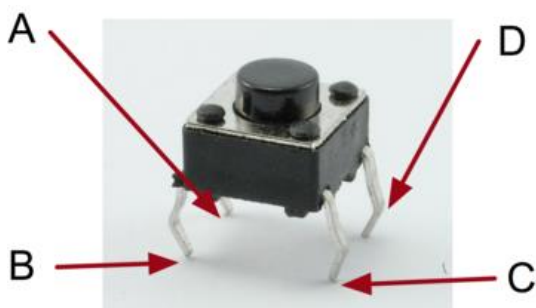
- (1) Elegoo UNO R3
- (1) Versuchsaufbau
- (1) 5mm rote LED
- (1) 220 Ohm-Widerstand
- (2) Druckschalter
- (6) M-M Kabel

Komponentenanweisung

DRUCKSCHALTER:

Schalter sind wirklich einfache Komponenten. Wenn Sie eine Taste drücken oder einen Hebel klappen, verbinden sie zwei Kontakte zusammen, so dass der Strom durch sie fließen kann.

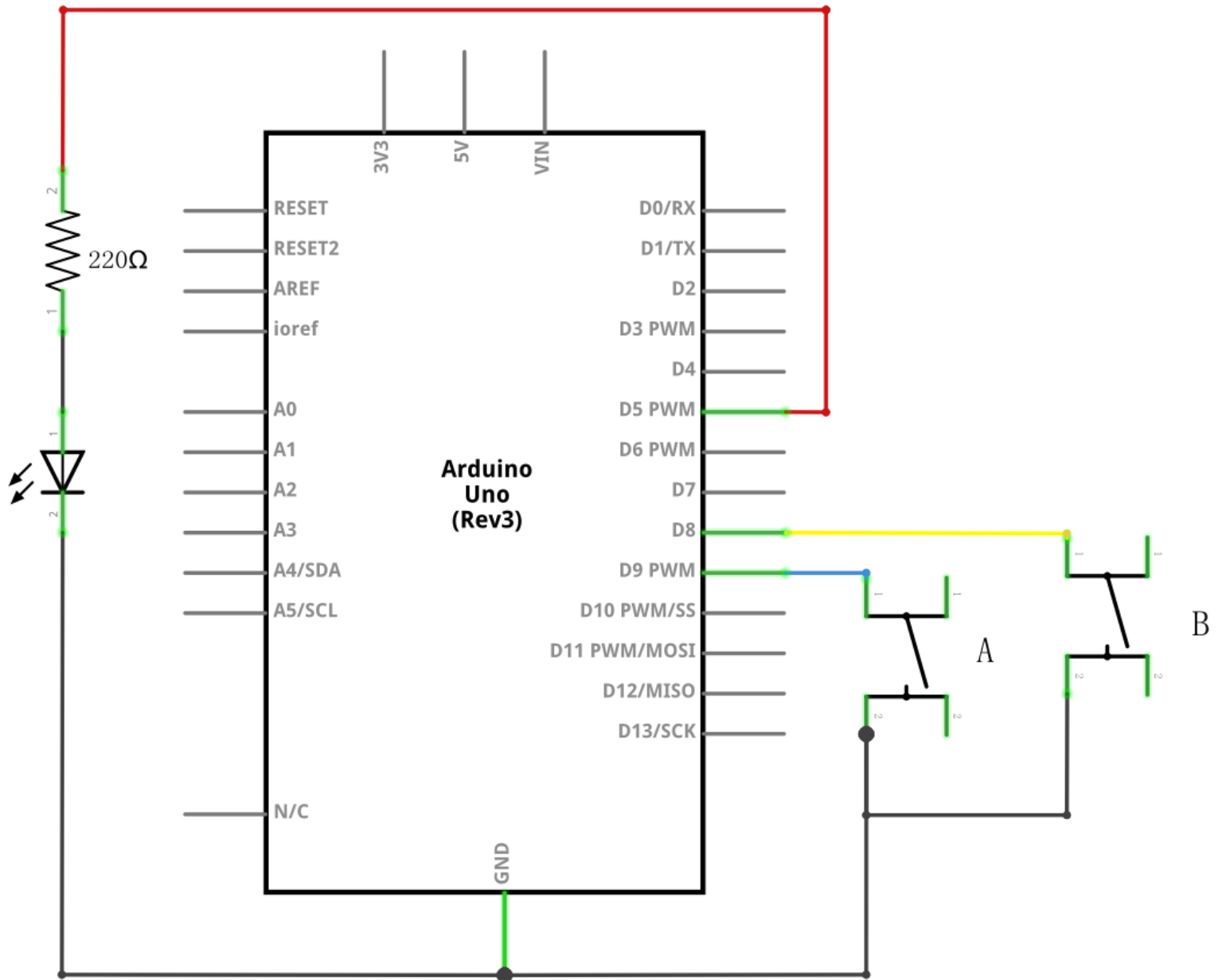
Die kleinen taktile Schalter, die in dieser Lektion verwendet werden, weisen vier Anschlüsse, die ein wenig Verwirrung erzeugen können, auf.



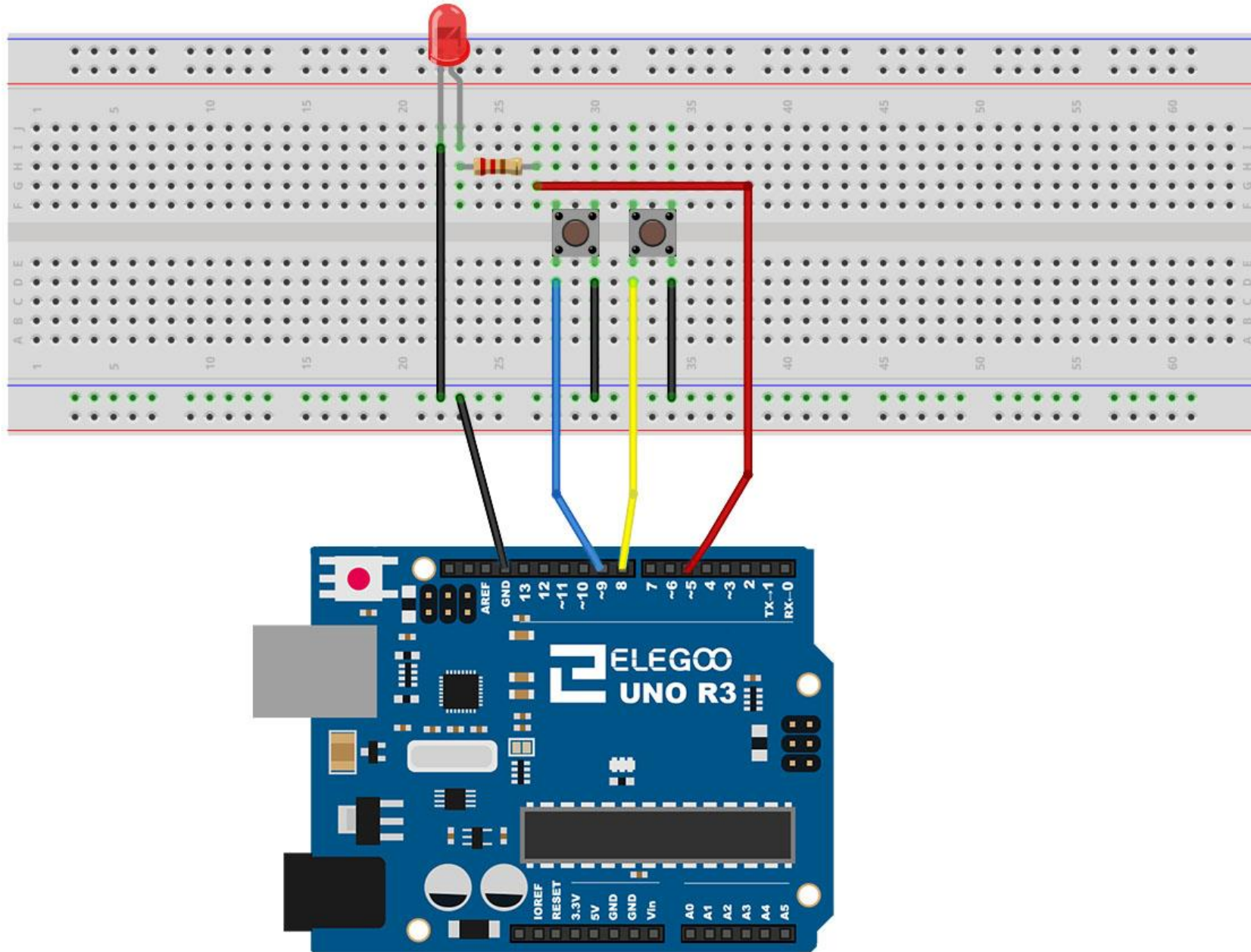
Eigentlich gibt es nur zwei echte elektrische Anschlüsse. Innerhalb des Schalterpakets sind Pins B und C miteinander verbunden, wie die Pins A und D sind.

Verbindung

Schema



Schaltplan



Obwohl die Körper der Schalter quadratisch sind, ragen die Pins aus den gegenüberliegenden Seiten des Schalters heraus. Dies bedeutet, dass die Pins nur ausreichend weit voneinander getrennt werden, wenn sie richtig auf dem Versuchsaufbau angeordnet werden.

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

Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Laden Sie den Sketch auf Ihr UNO-Board. Drücken Sie die obere Taste, um die LED einzuschalten, während Sie die untere Taste drücken, um sie auszuschalten.

Der erste Teil des Sketches definiert drei Variablen für die drei Pins, die verwendet werden. Der ,ledPin' ist der Ausgangspin, und der ,buttonApin' wird sich auf den Schalter näher an der Spitze des Versuchsaufbaus beziehen, und der ,buttonBpin' wird sich auf den anderen Schalter beziehen.

Die ,Setup'-Funktion stellt fest, dass der ledPin wie gewohnt als einen AUSGANG betrachtet wird, aber nun behandeln wir zwei Pins. In diesem Fall verwenden wir das Set von pinMode als ,EINGANG_PULLUP' wie folgt:

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

Der Pin-Modus von EINGANG_PULLUP bedeutet, dass der Pin als einen Eingang verwendet wird, aber dass, wenn nichts anderes an den Eingang angeschlossen ist, sollte er ,nach oben gezogen' auf HIGH wird. Mit anderen Worten ist der Standardwert für den Eingang HIGH, es sei denn, dass er durch die Aktion des Drückens der Taste auf LOW gezogen wird.

Aus diesem Grund sind die Schalter an GND angeschlossen. Wenn ein Schalter gedrückt wird, schließt er den Eingangspin an GND an, so dass er nicht mehr HIGH ist.

Da der Eingang normalerweise HIGH ist, und auf LOW fällt, nur wenn die Taste gedrückt wird, wird die Logik ein wenig auf den Kopf gestellt. Wir werden dies in der ,Loop'-Funktion behandeln.

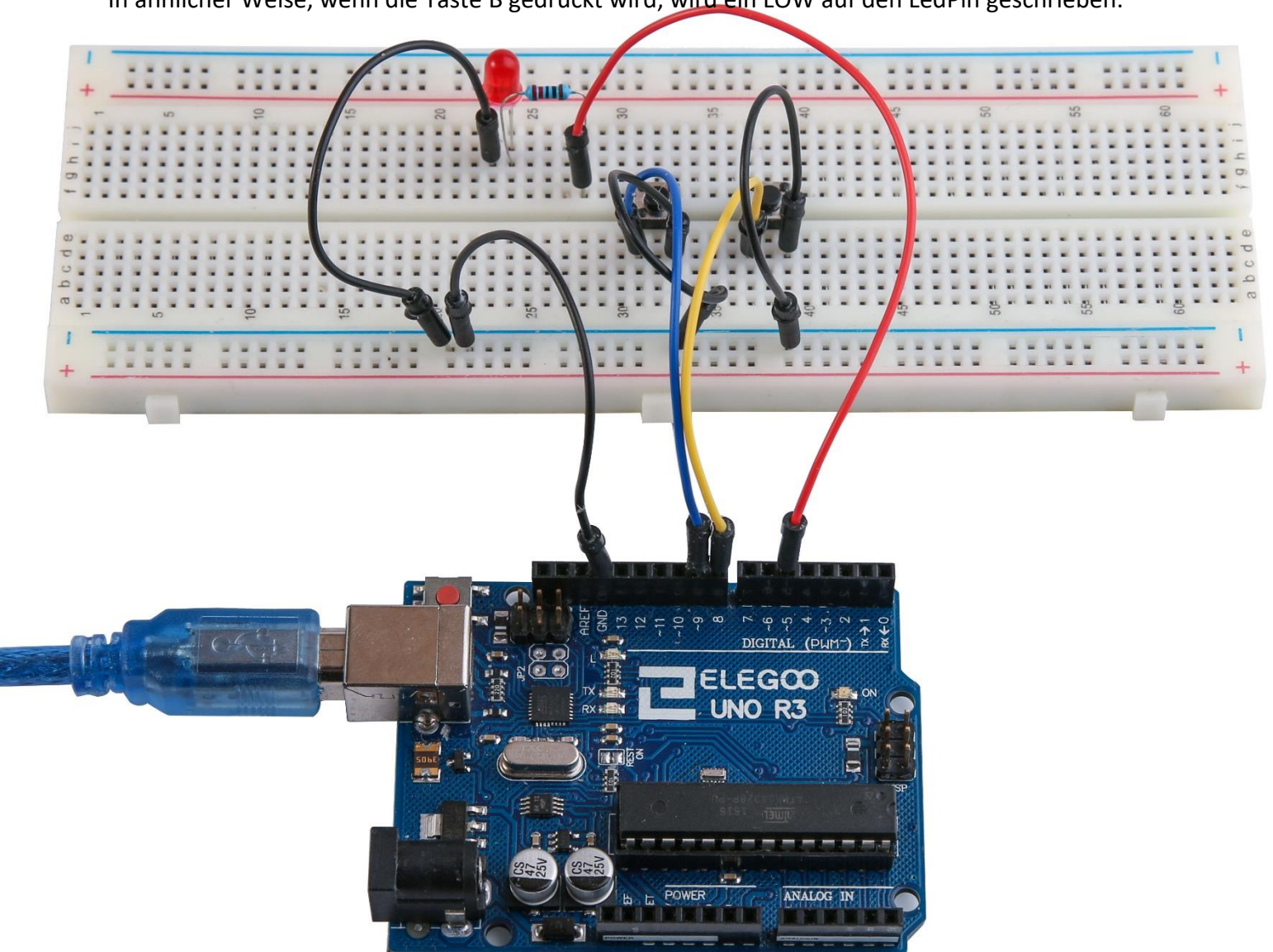
```
void loop()  
{  
if (digitalRead(buttonApin) == LOW)
```

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

In der ‚Loop‘-Funktion gibt es zwei ‚IF‘-Anweisungen, eine für jede Taste. Jede macht eine ‚digitalRead‘ auf dem angemessenen Eingang.

Denken Sie daran, wenn die Taste gedrückt wird, wird der entsprechende Eingang LOW sein. Wenn die Taste A niedrig ist, schaltet ein ‚digitalWrite‘ auf dem ledPin sie ein.

In ähnlicher Weise, wenn die Taste B gedrückt wird, wird ein LOW auf den LedPin geschrieben.



Lektion 6: Töne erzeugen

Übersicht

In dieser Lektion werden Sie erfahren, wie ein Ton mit einem aktiven Summer erzeugt wird.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Aktiver Summer
- (2) F-M Kabel

Komponentenanweisung

SUMMER:

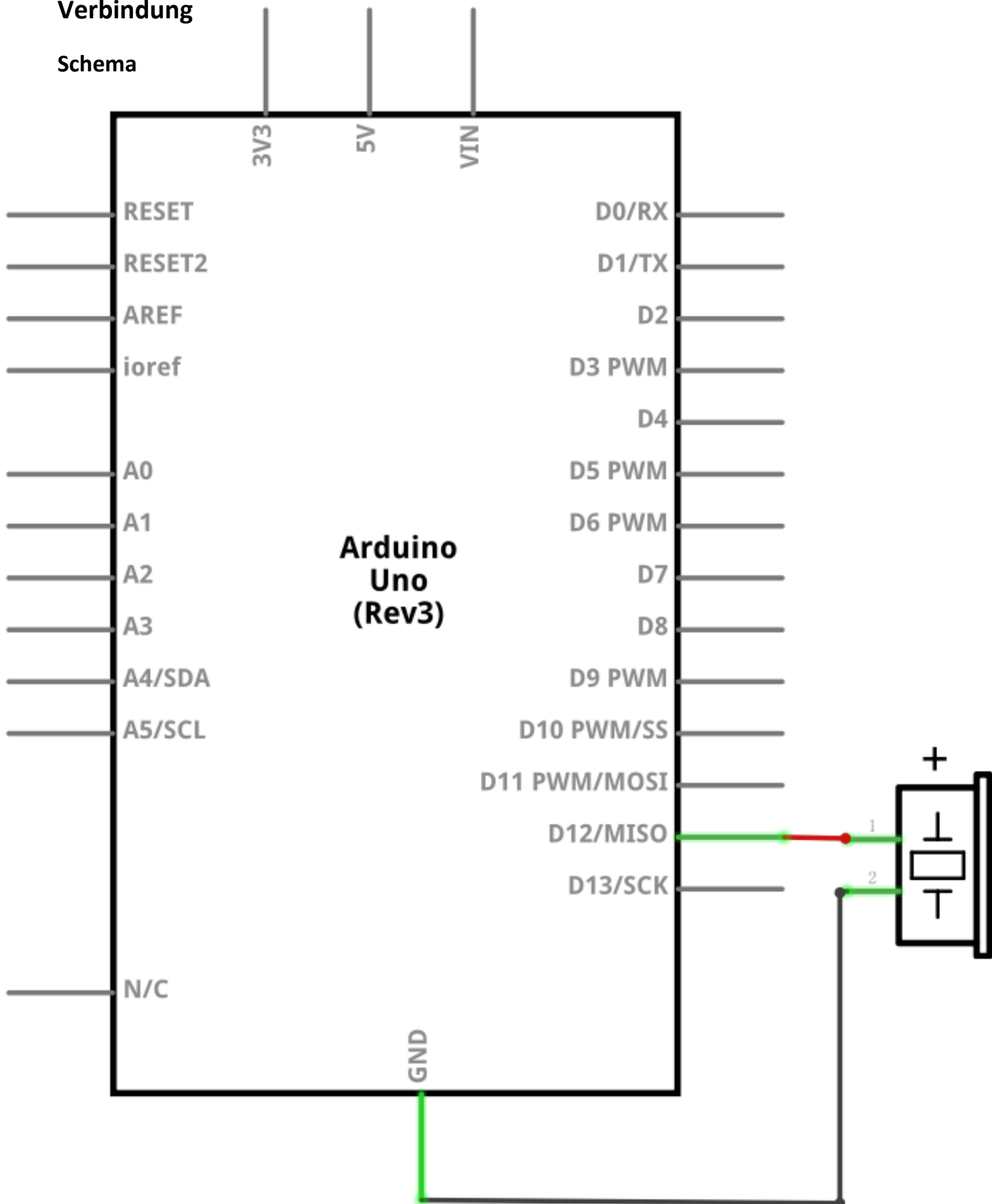
Elektronische Summer sind mit Gleichstrom betrieben und mit einer eingebauten Schaltung ausgestattet. Sie werden häufig in Computern, Druckern, Kopierern, Alarmgebern, elektronischen Spielzeugen, elektronischen Geräten für Kraftfahrzeuge, Telefonen, Zeitmessern und anderen elektronischen Produkten für Voice-Geräte, angewendet. Summer können als aktive und passive Summer kategorisiert. Drehen Sie die Pins der zwei Summer nach oben. Der Summer, der mit einer grünen Leiterplatte ausgestattet ist, ist ein passiver Summer, während der andere Summer, der mit einem schwarzen Klebeband ausgestattet ist, ist ein aktiver Summer.

Der Unterschied zwischen den beiden liegt darin, dass ein aktiver Summer über eine eingebaute oszillierende Quelle verfügt, so wird er einen Ton erzeugen, wenn er elektrisiert ist. Ein passiver Summer verfügt über keine solche Quelle, so wird er nicht piepsen, wenn Gleichstromsignale verwendet werden; stattdessen müssen Sie Rechteckwellen, deren Frequenz zwischen 2K und 5K liegt, verwenden, um ihn zu betreiben. Wegen der mehreren eingebauten oszillierenden Schaltungen ist der aktive Summer oft teurer als der passive Summer.

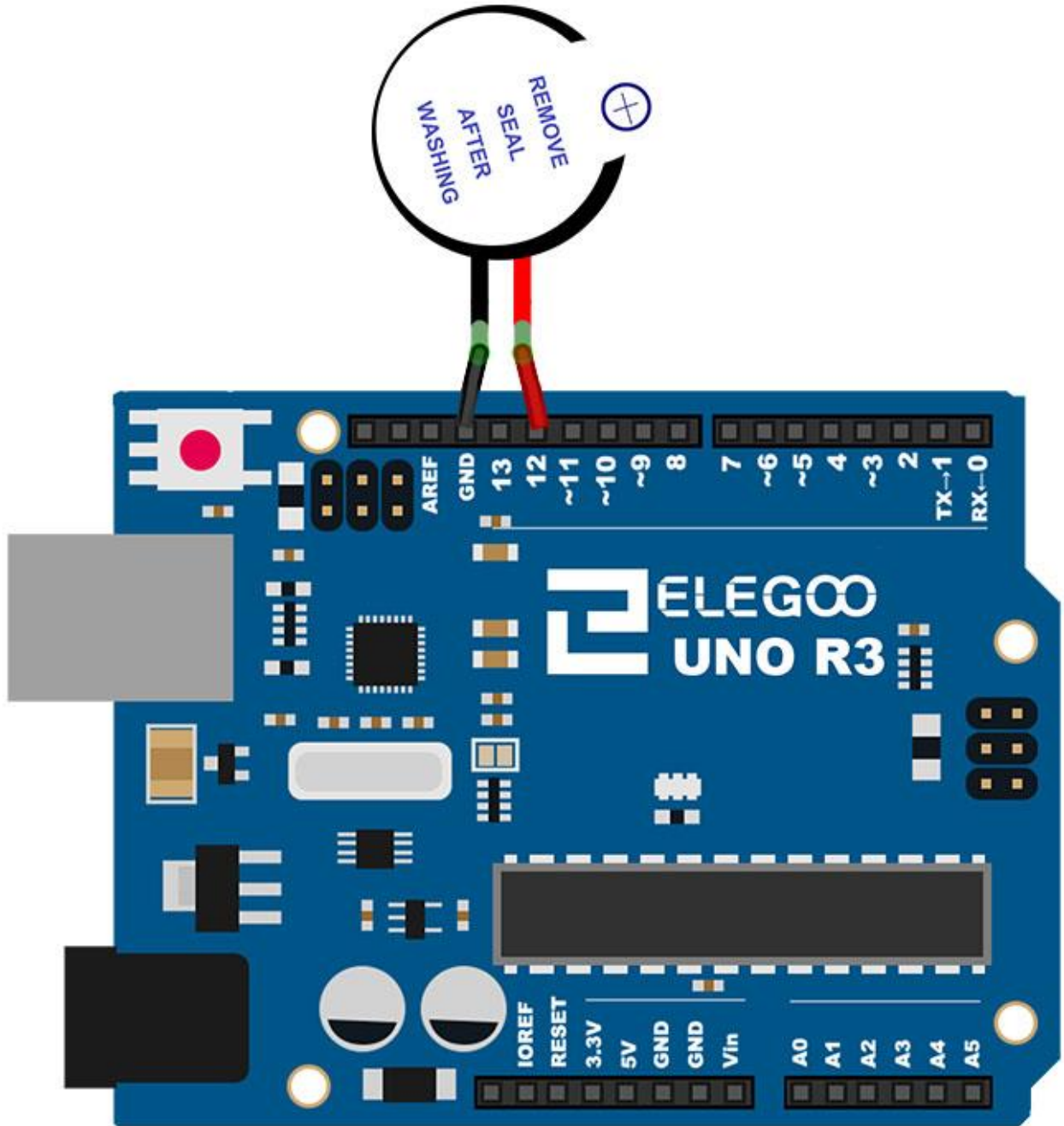


Verbindung

Schema

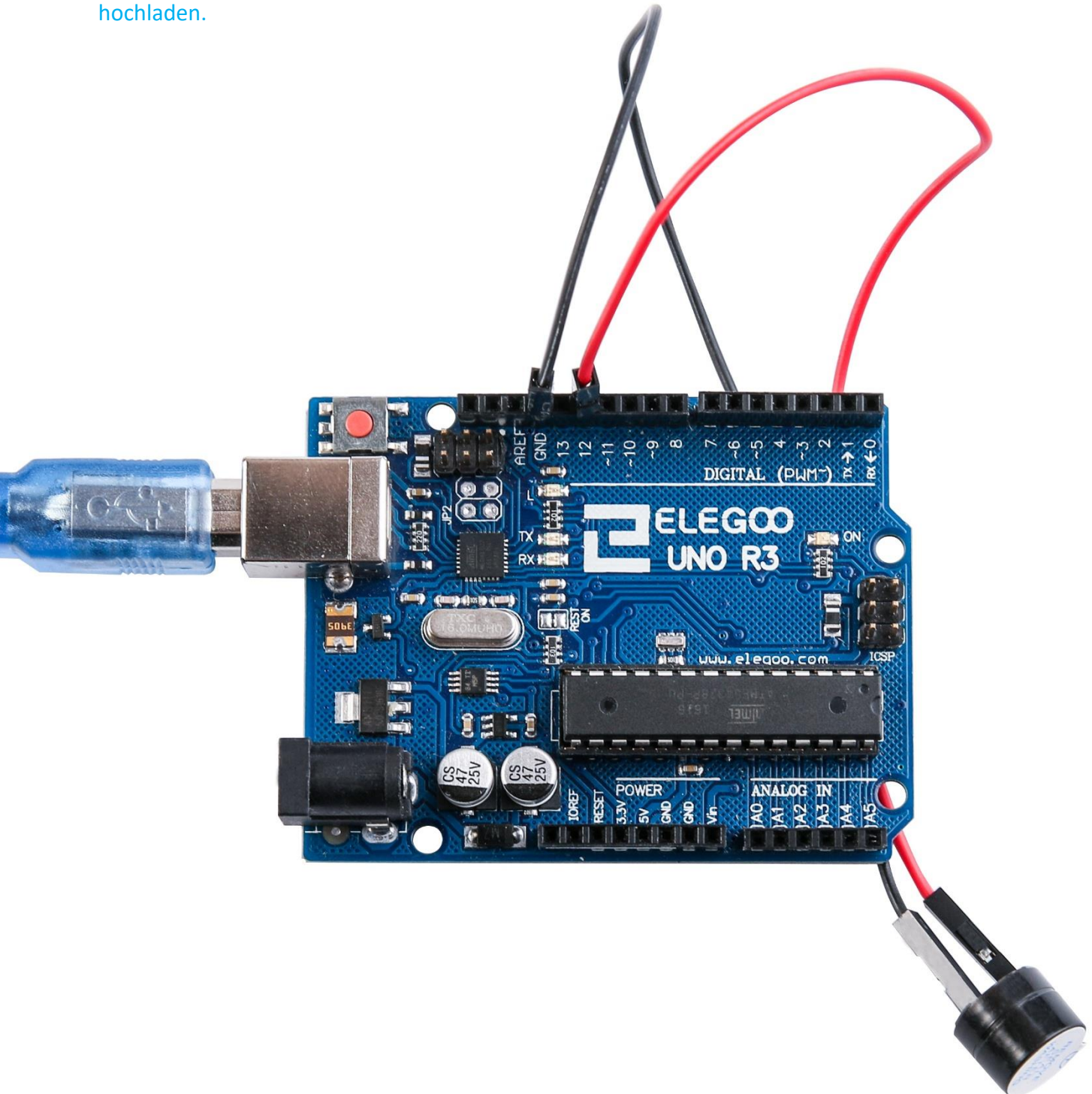


Schaltplan



Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefodert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.



Lektion 7: Passive Summer

Übersicht

In dieser Lektion werden Sie erfahren, wie die passiven Summer verwendet werden.

Der Zweck des Experiments ist, acht verschiedene Töne zu erzeugen, und jeder Ton dauert 0,5 Sekunden: ab Altstimme Do (523Hz), Re (587Hz), Mi (659Hz), Fa (698Hz), So (784Hz), La (880Hz), Si (988Hz) bis Oberstimme Do (1047Hz).

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Passiver Summer
- (2) F-M Kabel

Komponentenanweisung

Passiver Summer:

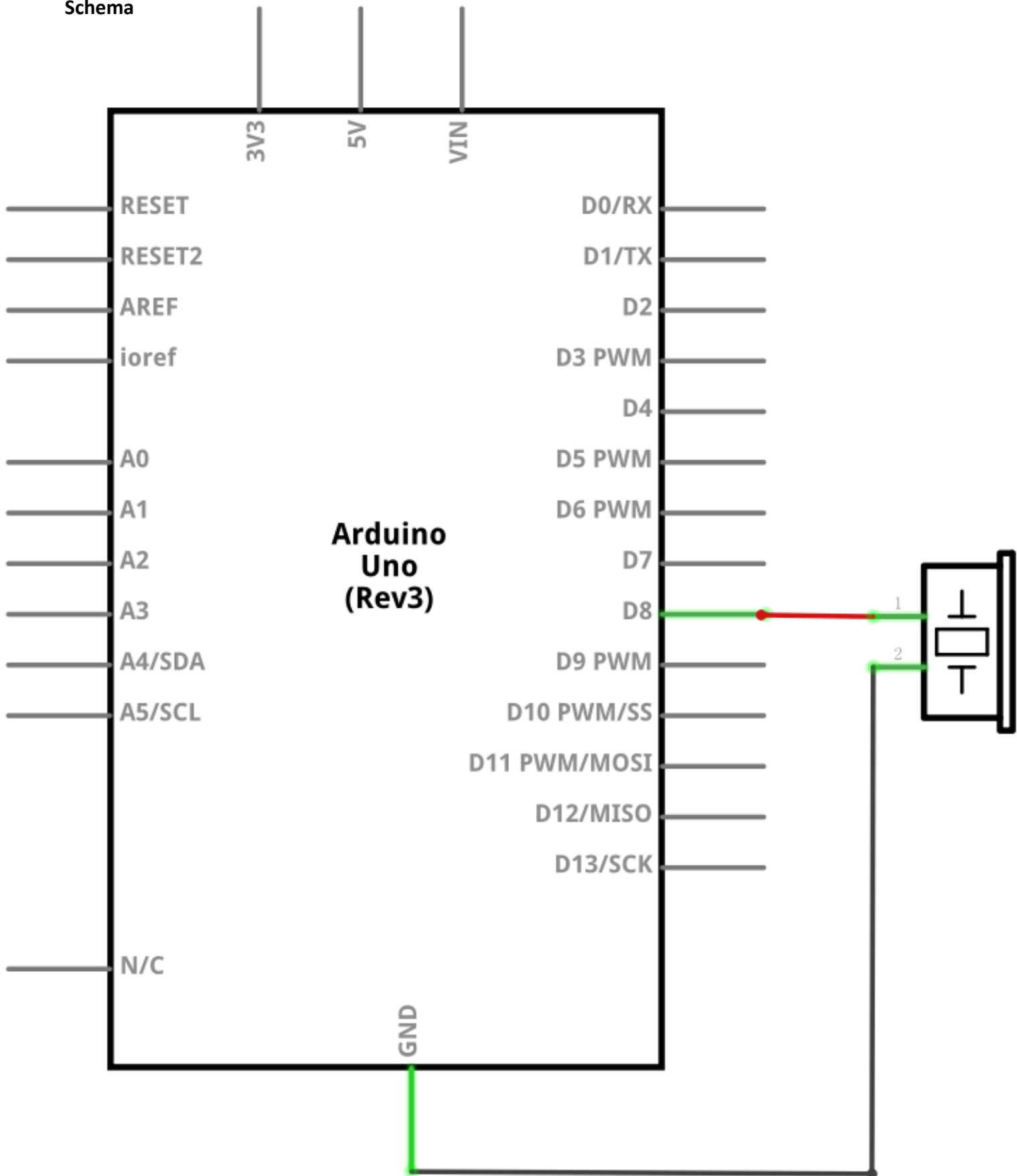
Der Arbeitsgrundsatz des passiven Summers liegt darin, dass PWM verwendet wird, um die Luft vibrieren zu machen. Solange die Schwingungsfrequenz entsprechend geändert wird, können verschiedene Töne erzeugt werden. Zum Beispiel, wenn ein Impuls von 523Hz gesendet wird, kann Altstimme Do erzeugt werden, wenn ein Impuls von 587Hz gesendet wird, kann Mittelton Re erzeugt werden, und wenn ein Impuls von 659Hz gesendet wird, kann Mittelton Mi erzeugt werden. Mit dem Summer können Sie ein Lied spielen.

Wir sollten darauf achten, die analog Write ()-Funktion vom UNO R3-Board nicht verwenden, um einen Impuls, der an den Summer gesendet wird, zu erzeugen, weil der Impulsausgang von analog Write () festgelegt wurde (500Hz).

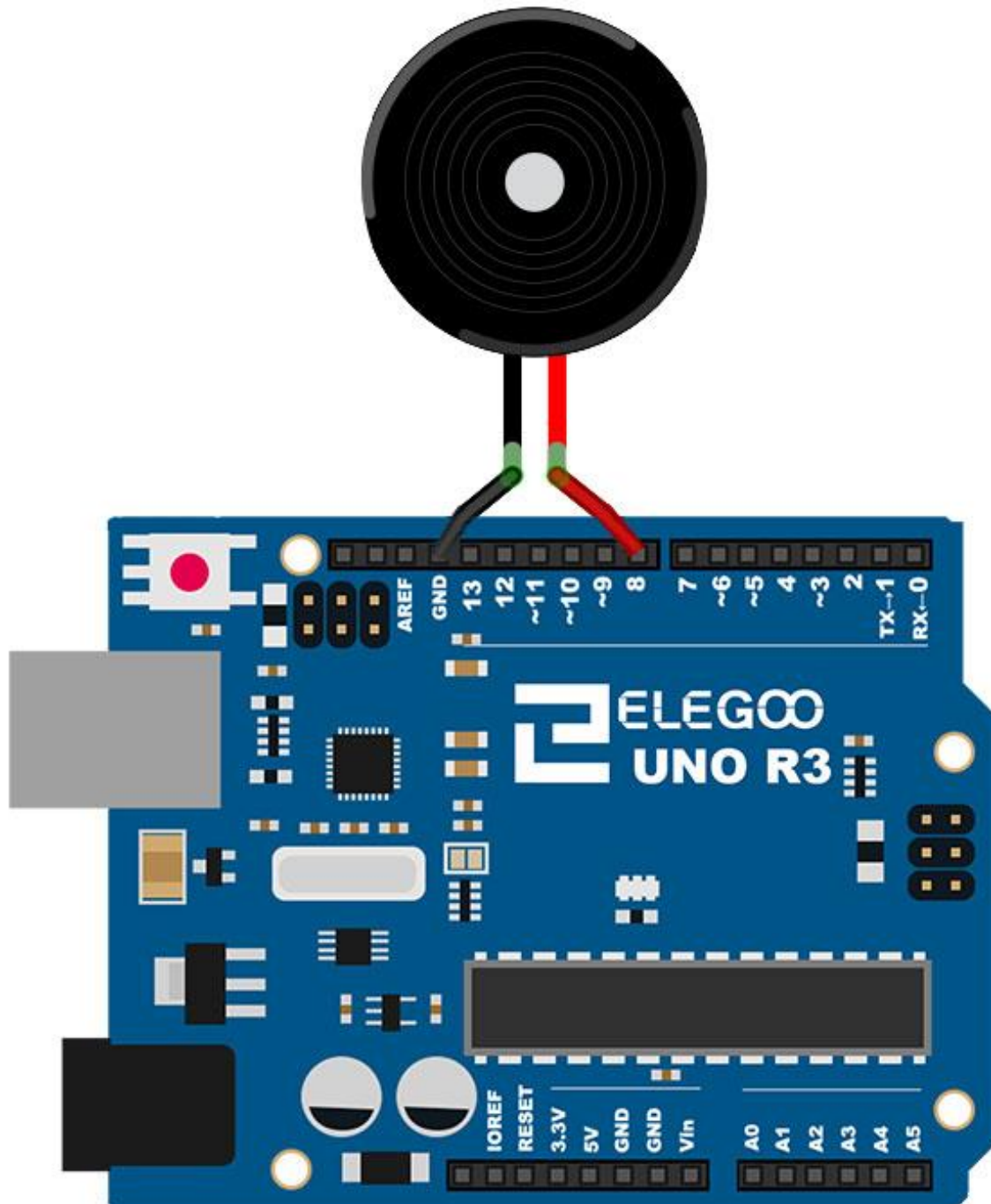


Verbindung

Schema



Schaltplan

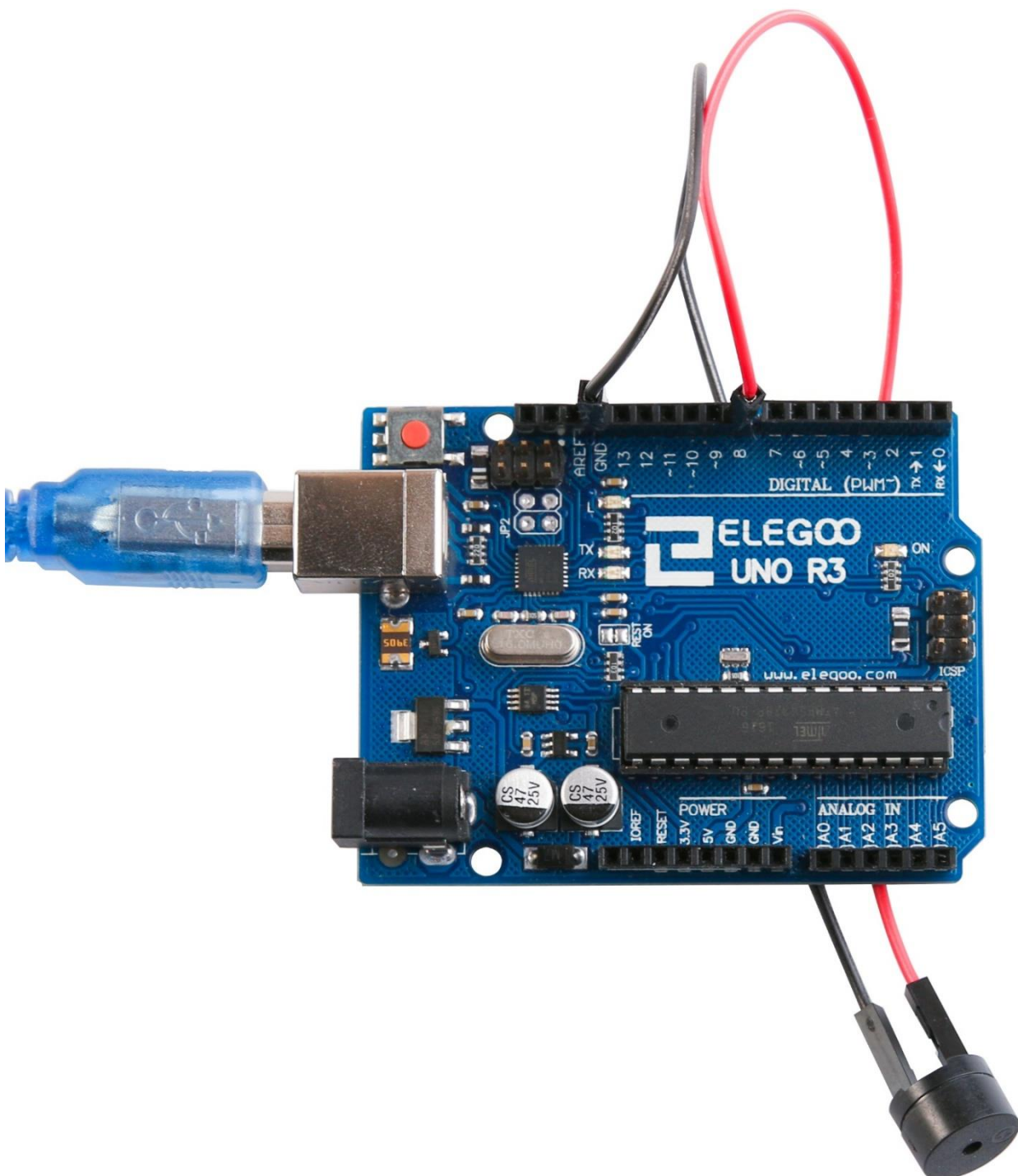


Verdrahten Sie den Summer, um ihn an das UNO R3-Board anzuschließen, wobei das rote (positive) Kabel mit dem Pin8 und das schwarze Kabel (negativ) mit der GND verbunden werden.

Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek <Pitches> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.



Lektion 8: Kugelschalter

Übersicht

In dieser Lektion werden wir erfahren, wie wir einen Kugelschalter zur Erkennung des kleinen Neigungswinkels verwenden.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Kugelschalter
- (2) F-M Kabel



Komponentenanweisung

Neigungssensor:

Neigungssensoren ermöglicht es Ihnen, Ausrichtung oder Neigung zu erkennen. Sie sind klein, billig, geringer Stromverbrauch und benutzerfreundlich. Wenn sie angemessen verwendet werden, werden sie nicht verschleifen. Aufgrund ihrer Einfachheit werden sie häufig in Spielzeugen, kleinen Geräten und Vorrichtungen verwendet. Manchmal werden sie aus offensichtlichen Gründen als „Quecksilberschalter“, „Neigungsschalter“ oder „Rollkugelsensor“ genannt.

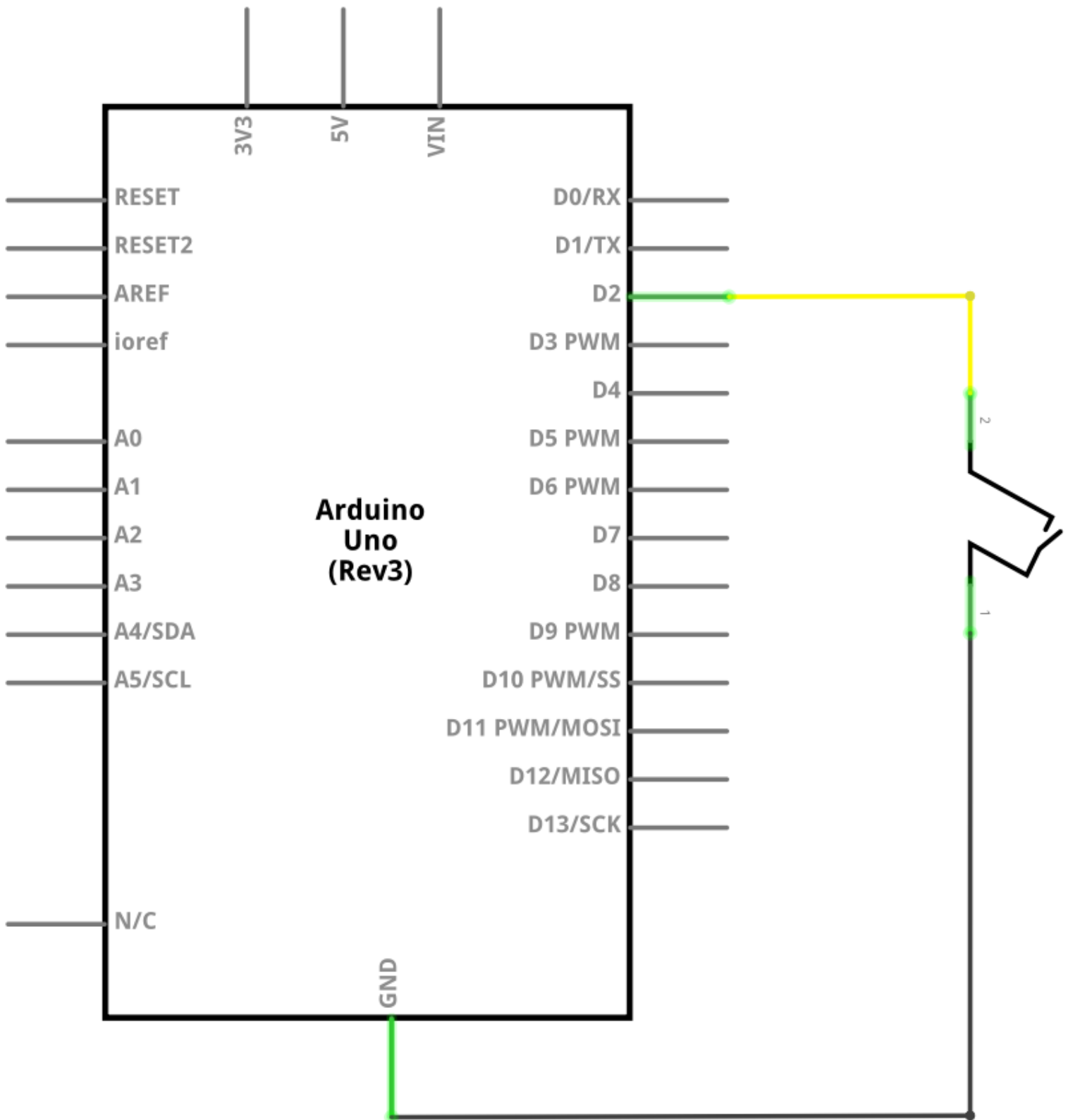
Sie bestehen üblicherweise aus einem Hohlraum in irgendeiner Form (zylindrisch ist beliebt, aber nicht immer) mit einem leitenden freien Ding darin, wie einem Klumpen von Quecksilber oder Rollkugel. An einem Ende des Hohlraums gibt es zwei leitende Elemente (Pole). Wenn der Sensor ausgerichtet ist, so richtet dieses Ende abwärts, rollt das Ding auf die Pole und verkürzt es sie, als ein Schalter wirkt.

Weil es nicht so präzise oder flexibel wie ein Vollbeschleunigungsmesser ist, können Neigungsschalter Bewegung oder Ausrichtung erkennen. Ein weiterer Vorteil liegt darin, dass die großen Neigungsschalter automatisch anschalten können.

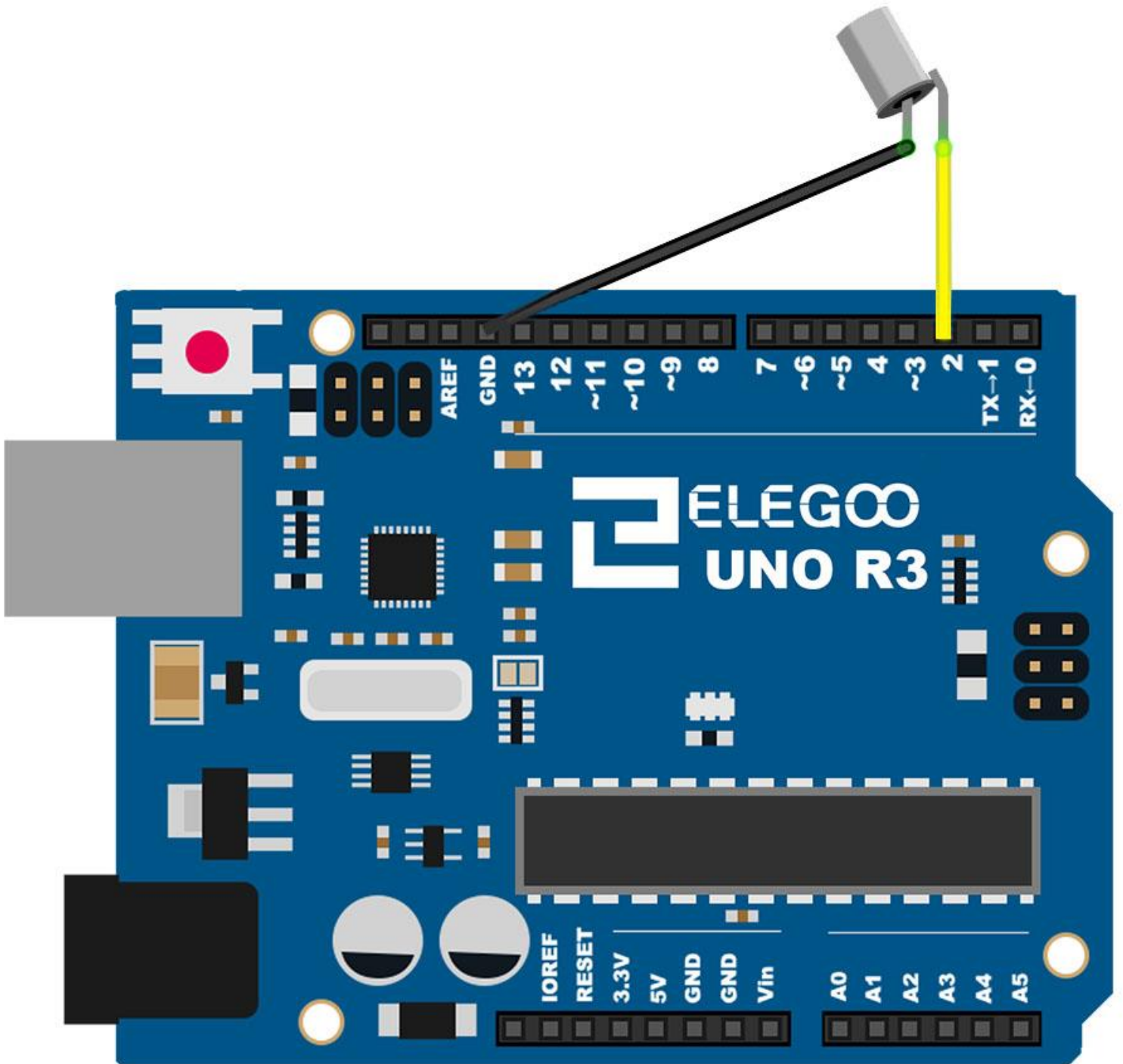
Im Gegenteil geben Beschleunigungsmesser digitale oder analoge Spannung, die durch eine zusätzliche Schaltung analysiert wird, aus.

Verbindung

Schema

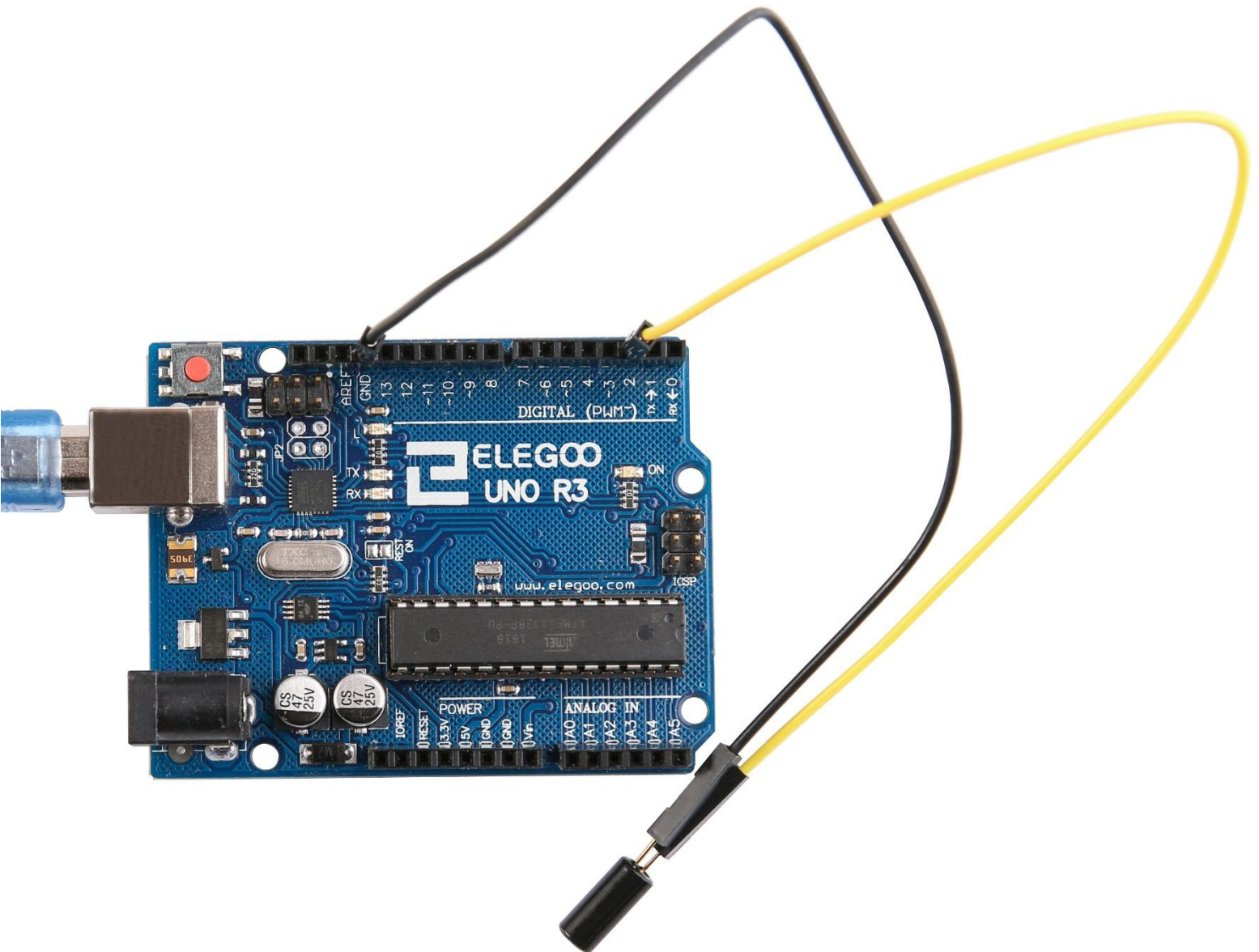


Schaltplan



Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefodert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.



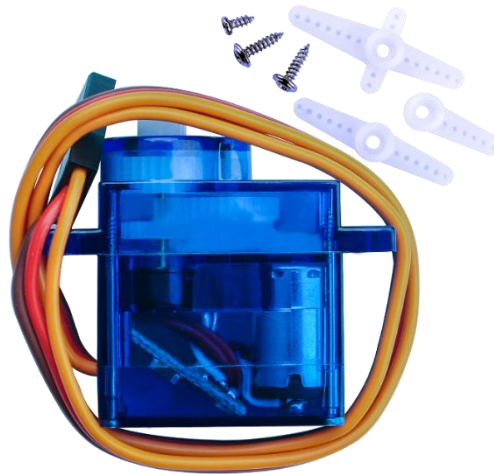
Lektion 9: Servomechanismus

Übersicht

Servomechanismus ist eine Art vom Getriebemotor, der nur 180 Grade drehen kann. Er ist durch das Senden der elektrischen Impulse aus Ihrem UNO R3-Board kontrolliert. Diese Impulse erzählen dem Servomechanismus, in welche Position er bewegen sollte. Das Servo hat drei Drähte, die braune ist die Masseleitung und sollte an den GND Anschluss der UNO angeschlossen werden, die rote ein Stromkabel ist und sollte an den 5v Anschluss und Orange ist der Signalleitung und sollte an den Graben #9-Port angeschlossen werden.

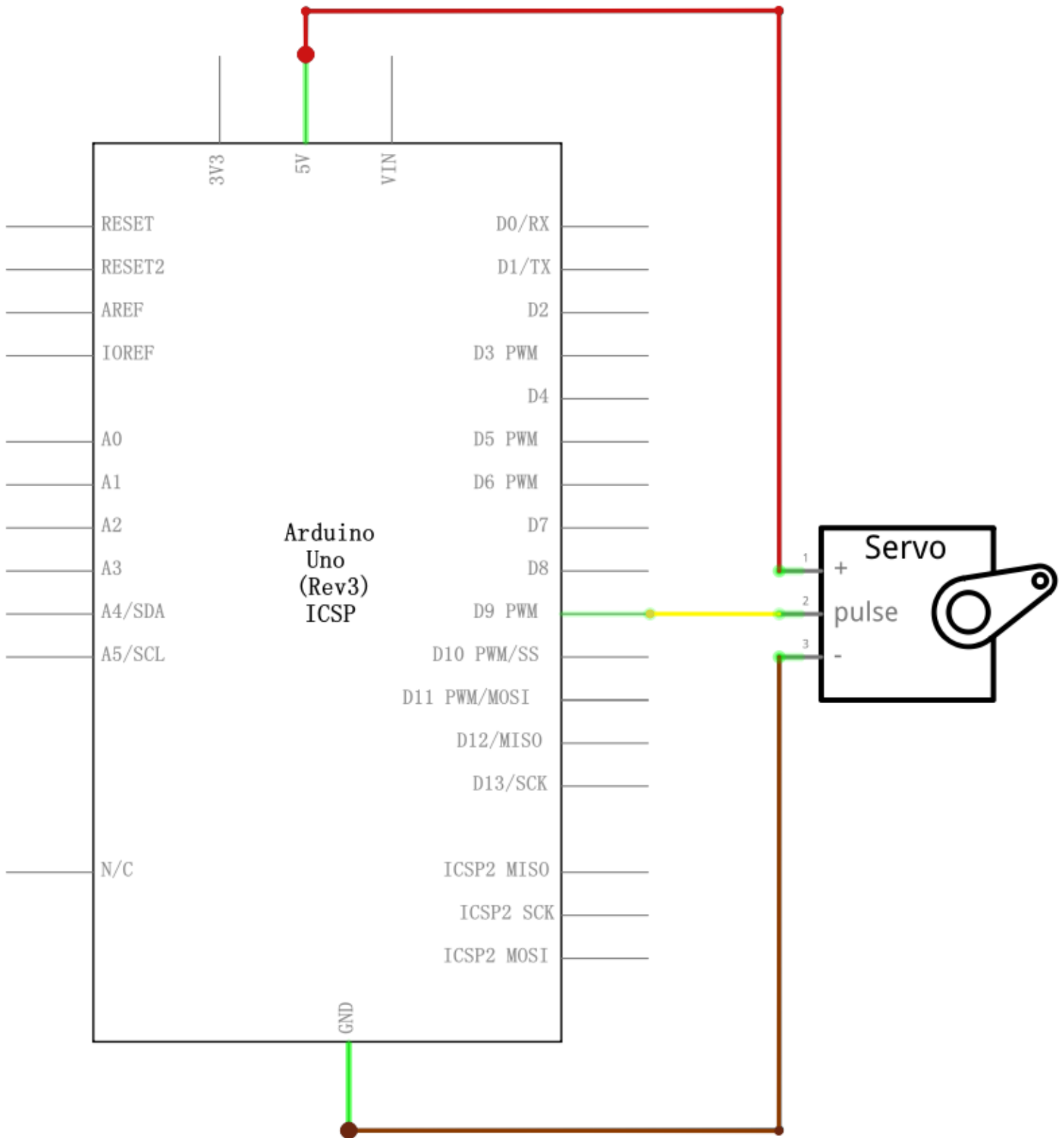
Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Servomechanismus
- (2) F-M Kabel

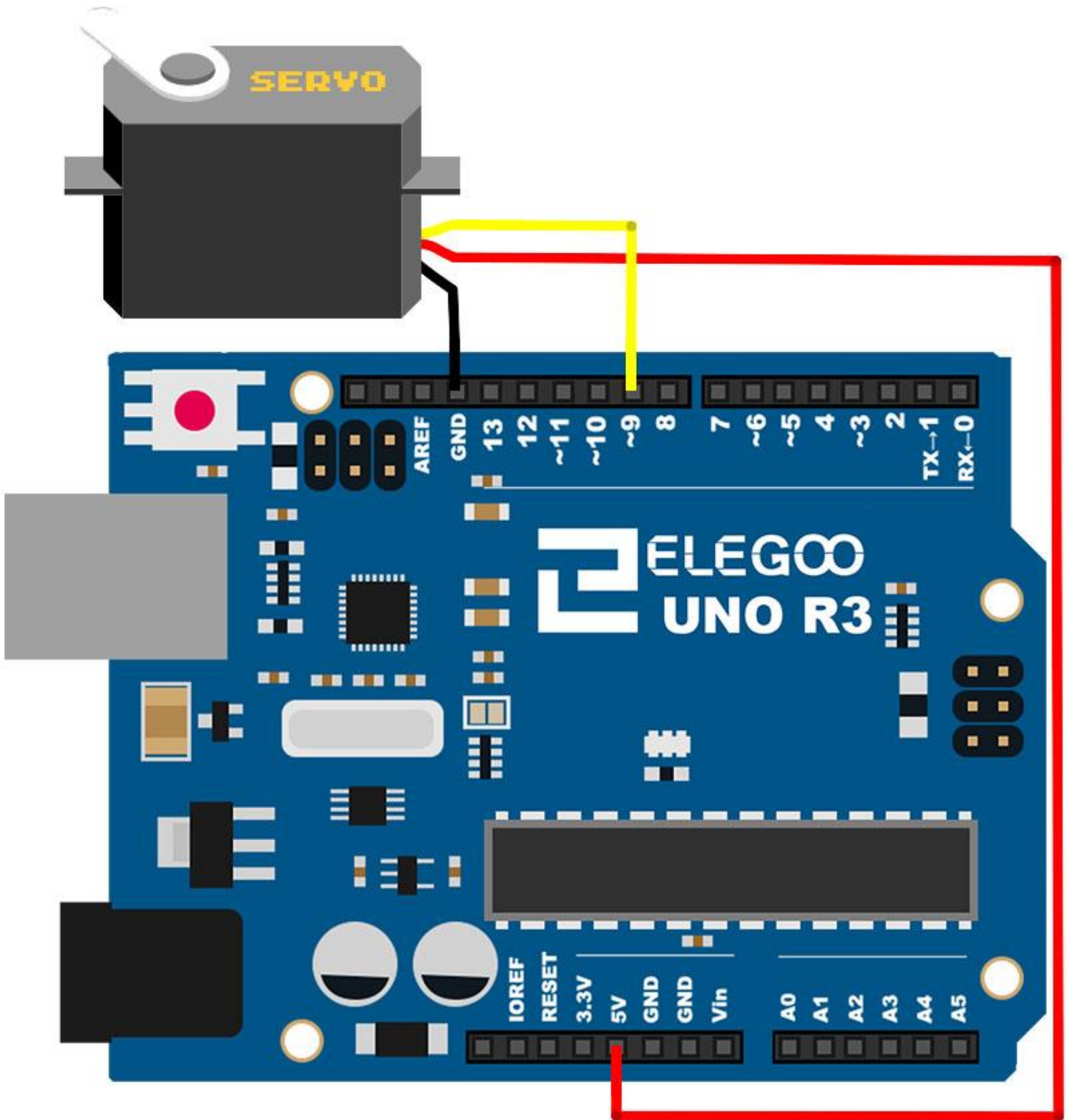


Verbindung

Schema



Schaltplan

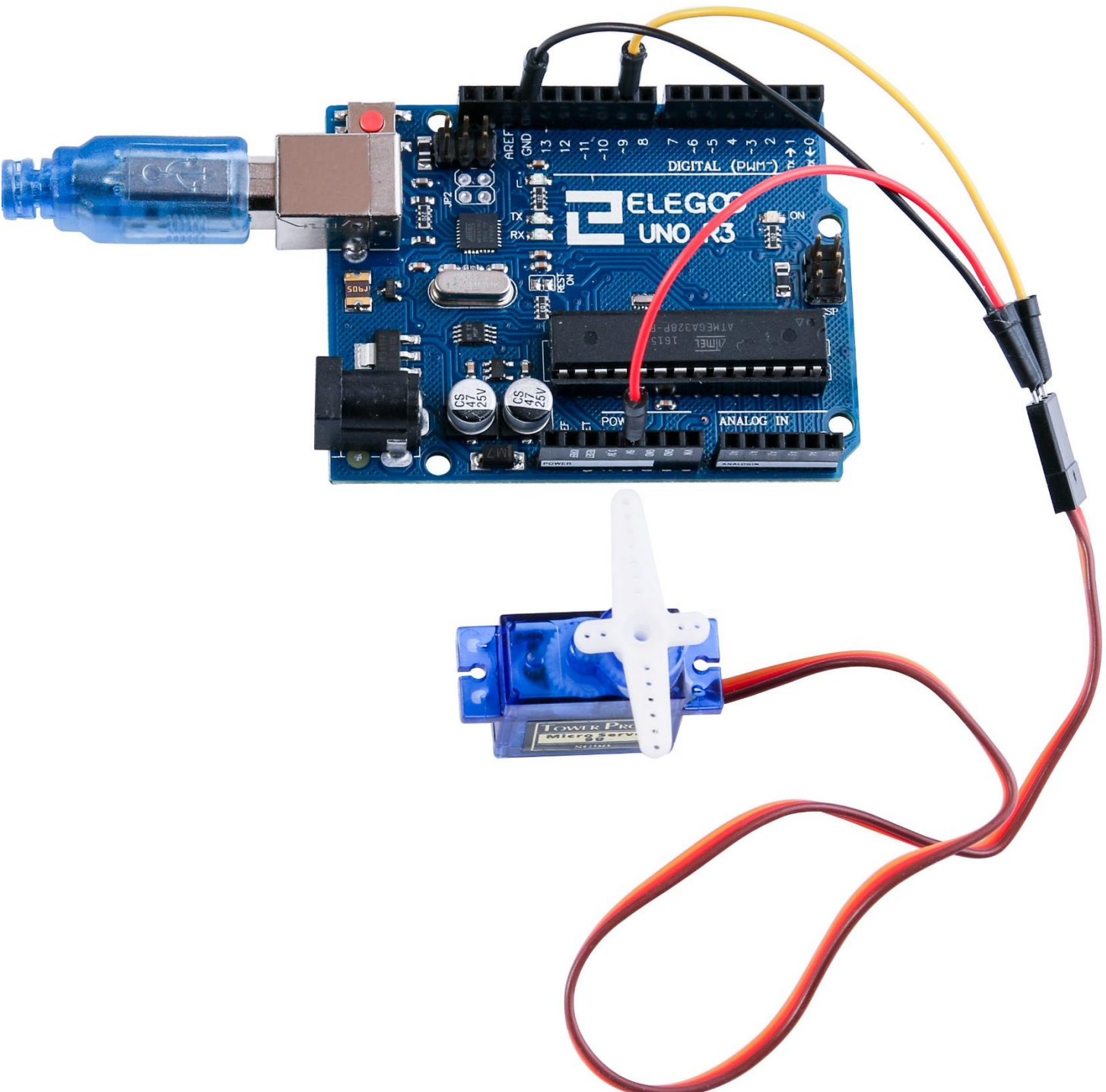


Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek <Servo> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.

Auf dem Bild ist das braune Kabel über die schwarzen M-M-Kabel, das rote, über die rote M-M-Kabel und den orangenen über die gelbe M-M-Kabel angepasst.



Lektion 10: Ultraschall-Sensor-Modul

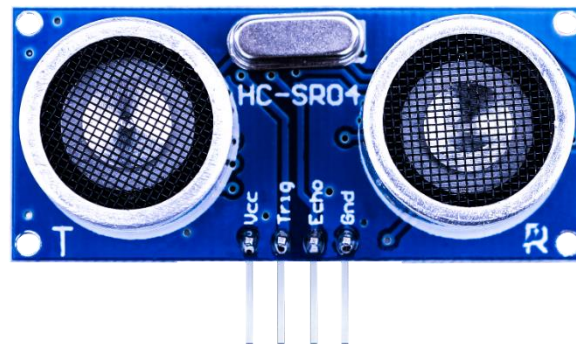
Übersicht

Ultraschall-Sensoren sind ideal für alle Arten von Programmen, die Abstandsmessungen benötigen, wie zum Beispiel Vermeidung vor Hindernissen.

Die HC-SR 04 sind preiswert und einfach zu bedienen, da wir eine Bibliothek, die speziell für diese Sensoren entwickelt wird, verwenden.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Ultraschall-Sensor-Modul
- (4) F-M Kabel



Komponentenanweisung

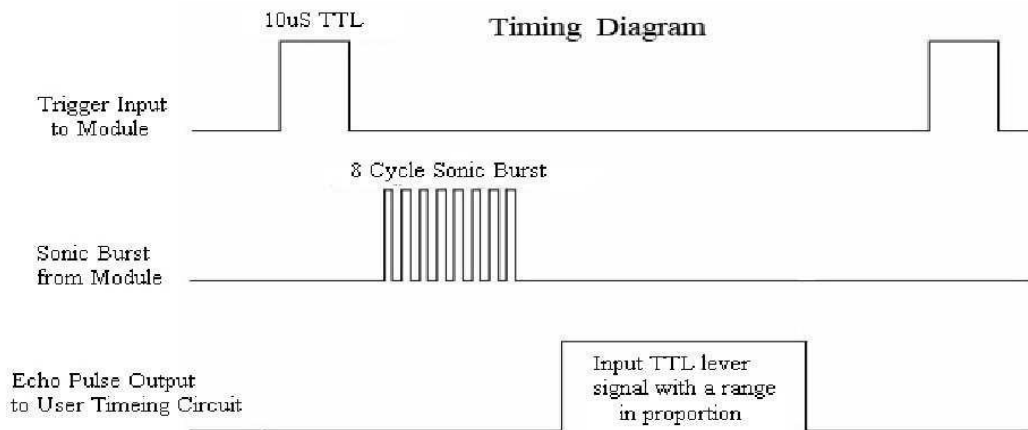
Ultraschall-Sensor

Ultraschallbereich-Modul HC-SR 04 bietet 2cm - 400cm berührungslose Messungsfunktion, und die Messgenauigkeit kann 3mm erreichen. Das Modul enthält Ultraschallsender, Empfänger und Steuerschaltung. Der grundlegende Arbeitsgrundsatz:

- (1) Verwenden Sie IO-Trigger für mindestens 10us High-Pegel-Signal,
- (2) Das Modul sendet automatisch acht 40kHz und erkennt, ob ein Impulssignal zurückgesendet wird.
- (3) FALLS das Signal über High-Pegel zurückgesendet wird, wird die Zeitdauer des hohen Ausgangs IO ab Senden von Ultraschall-Tone-Turning berechnet.

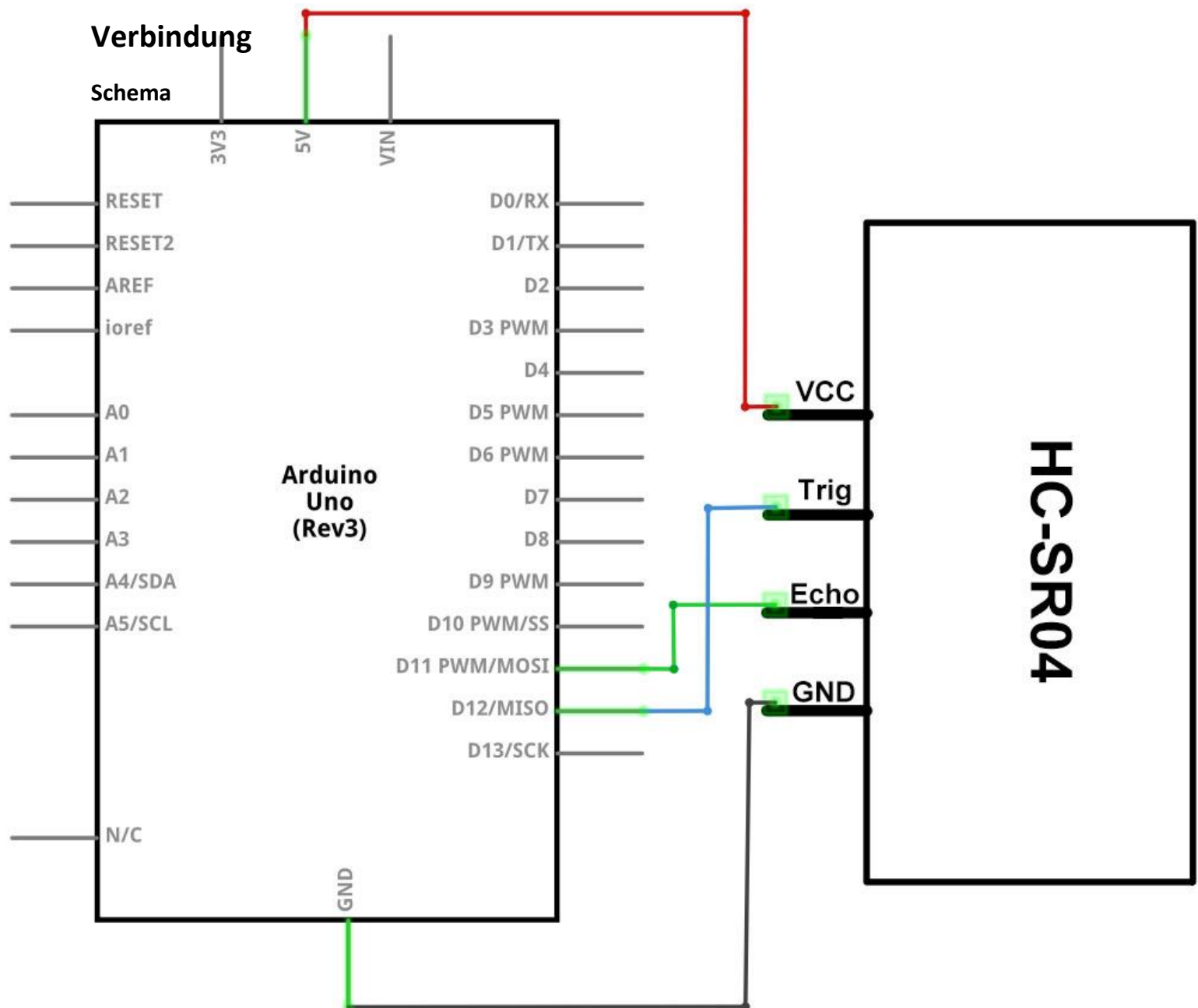
$$\text{Testabstand} = (\text{Zeit des High-Niveaus} \times \text{Schallgeschwindigkeit (340M / S)}) / 2$$

Das Timing-Diagramm ist unten dargestellt. Sie müssen den Trigger-Eingang nur mit einem kurzen 10uS Impuls versorgen, um den Bereich zu starten, und dann wird das Modul einen 8-Zyklus-Impuls von Ultraschall bei 40 Hz senden, und sein Echo erhöhen. Das Echo ist ein Gegenstand mit einem Abstand, der die Impulsbreite ist und dessen Bereich im Verhältnis ist. Sie können den Bereich durch das Zeitintervall zwischen dem Senden vom Trigger-Signal und dem Empfangen des Echosignals berechnen. Formel: $\mu\text{S} / 58 = \text{Zentimeter}$ oder $\mu\text{S} / 148 = \text{Zoll}$; oder: der Bereich = Zeit des High-Pegels* Geschwindigkeit (340M/S) / 2; wir schlagen Ihnen vor, über 60ms Messzyklus zu verwenden, um zu verhindern, dass das Signal des Trigger das Echosignal deckt.

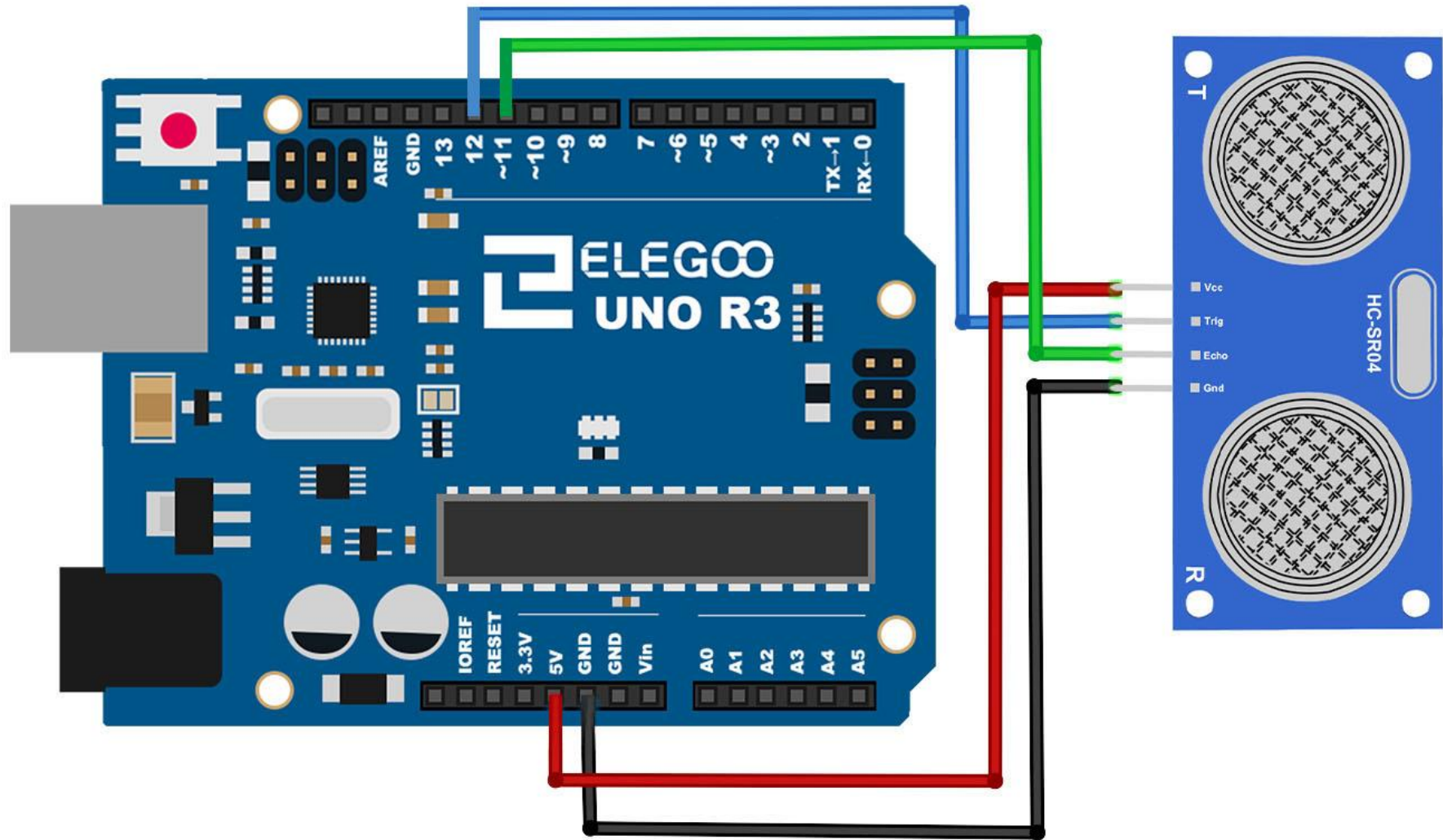


Verbindung

Schema



Schaltplan



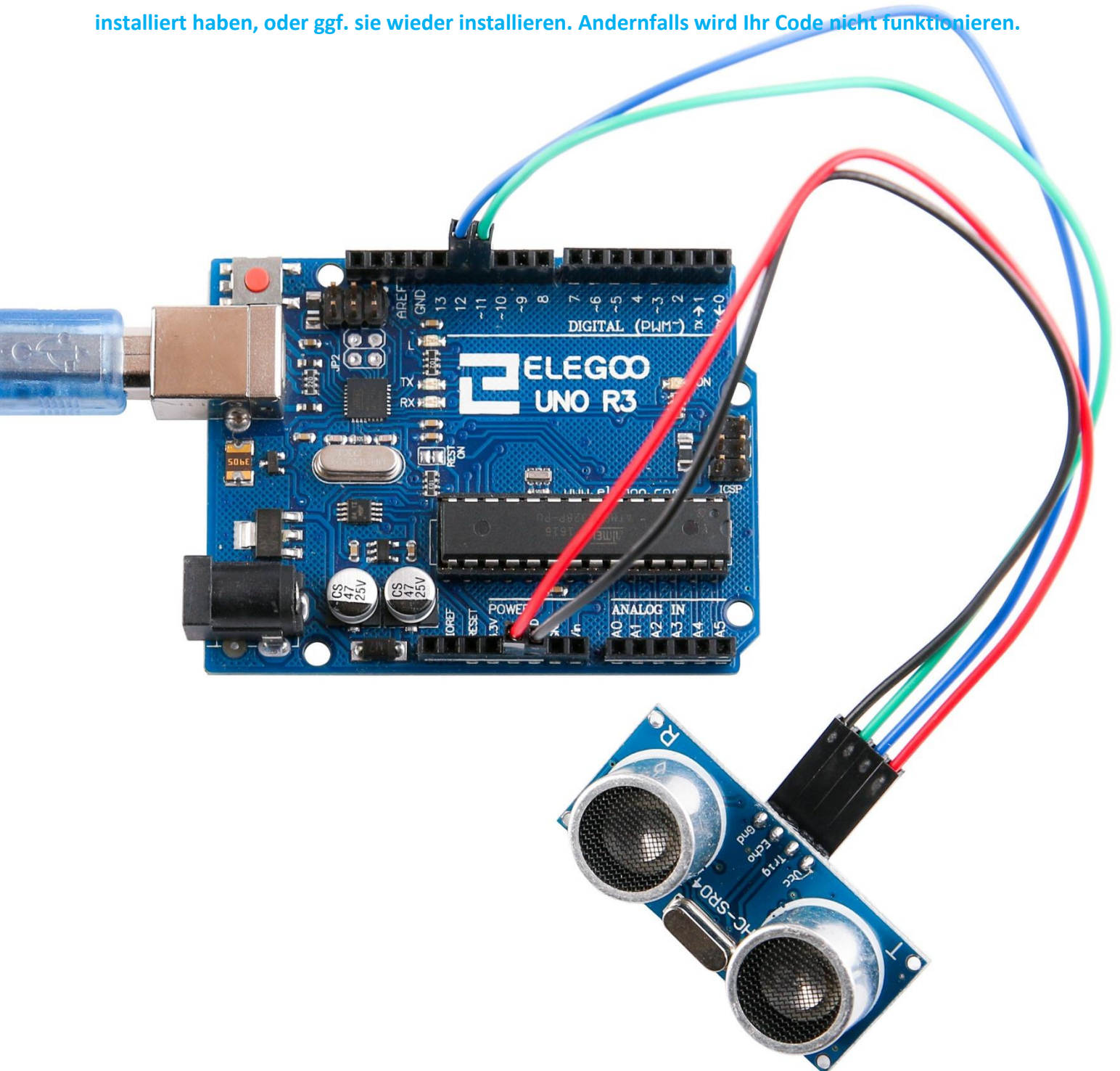
Code

Verwenden Sie eine Bibliothek, die für diese Sensoren entwickelt wird, um unseren Code kurz und einfach zu machen.

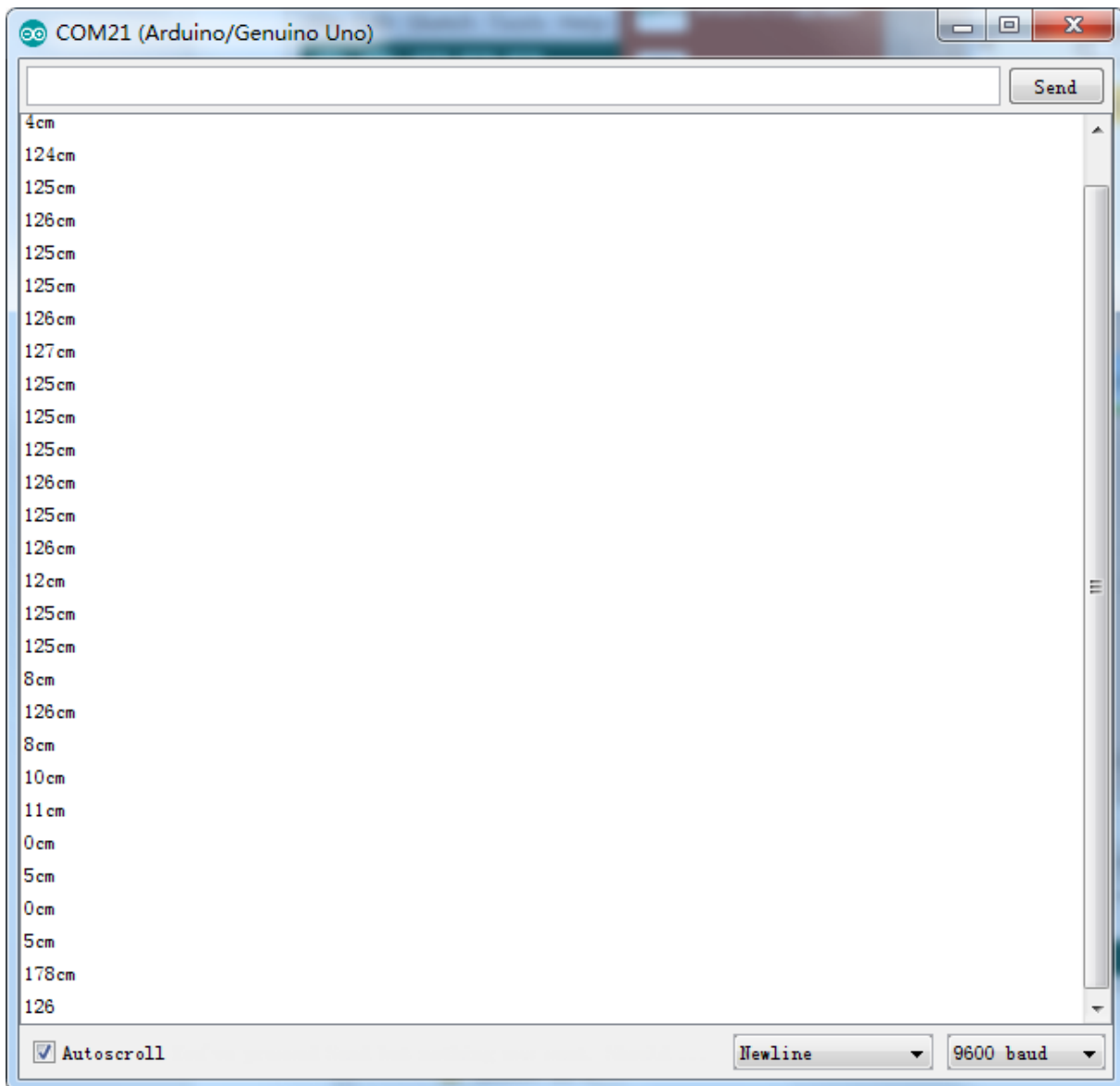
Die Bibliothek wird zu Beginn unseres Codes enthalten, und dann durch Verwendung von einfachen Befehlen kann das Verhalten des Sensors gesteuert werden.

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefodert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek <HC-SR04> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.



Open the monitor then you can see the data as blow:



Lektion 11 Tastenmodul

Übersicht

In diesem Programm werden wir erfahren, wie wir eine Tastatur in ein UNO R3-Board integrieren, so dass UNO R3 die Tasten, die von einem Benutzer gedrückt werden, ablesen.

Tastaturen werden in allen Arten von Geräten, einschließlich Mobiltelefone, Faxgeräte, Mikrowellen, Öfen, Türschlösser, etc. verwendet. Sie sind praktisch überall. Für eine große Menge von elektronischen Geräten werden sie für Benutzereingabe verwendet.

So ist es für Herstellung von vielen verschiedenen Arten der kommerziellen Produkte sehr wertvoll, wenn Sie wissen, wie Sie eine Tastatur mit einem Mikrocontroller wie einem UNO R3-Board verbinden.

Zum Schluss, wenn alle richtig angeschlossen und programmiert sind und wenn eine Taste gedrückt wird, wird sie auf dem seriellen Monitor auf Ihrem Computer angezeigt. Immer wenn Sie eine Taste drücken, wird sie auf dem seriellen Monitor angezeigt. Später werden wir in einem anderen Programm die Tastaturschaltung verbinden, so dass sie auf einer LCD angezeigt wird. Aber jetzt starten wir aus Gründen der Einfachheit einfach mit dem Anzeigen der Taste, die auf dem Computer gedrückt wird.

Für dieses Programm werden wir die Matrix-Tastatur verwenden. Dies ist eine Art von Tastatur, die einem Kodierungsschema folgt, so dass sie viele weniger Ausgangspins als die vorhandenen Tasten aufweist. Zum Beispiel weist die Matrix-Tastatur, die wir verwenden, 16 Tasten (0-9, A-D, *, #) auf, aber sie hat nur 8 Ausgangspins. Wenn wir eine lineare Tastatur verwenden würden, sollte es 17 Ausgangspins (ein für jede Taste und ein Massepin) aufweisen, um sicherzustellen, dass sie funktionieren. Das Matrix-Kodierungsschema erlaubt aber viele weniger Ausgangspins und dadurch viele weniger Verbindungen, die für das Funktionieren der Tastatur gebildet werden müssen. Auf diese Weise sind sie effizienter als lineare Tastaturen, da sie weniger Verdrahtungen haben.

Erforderliche Komponenten

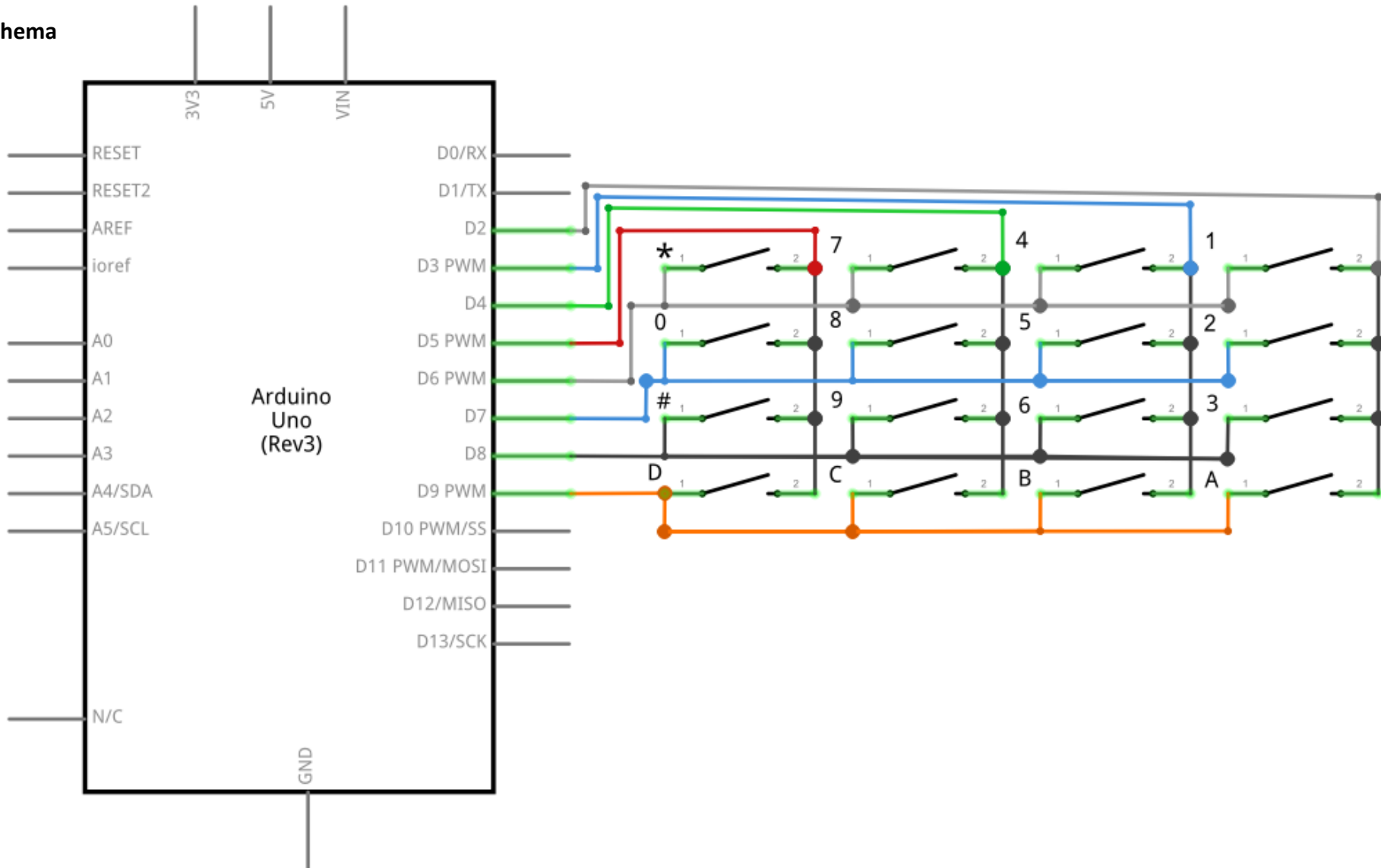
(1) Elegoo UNO R3

(1) Membranschalter-Modul

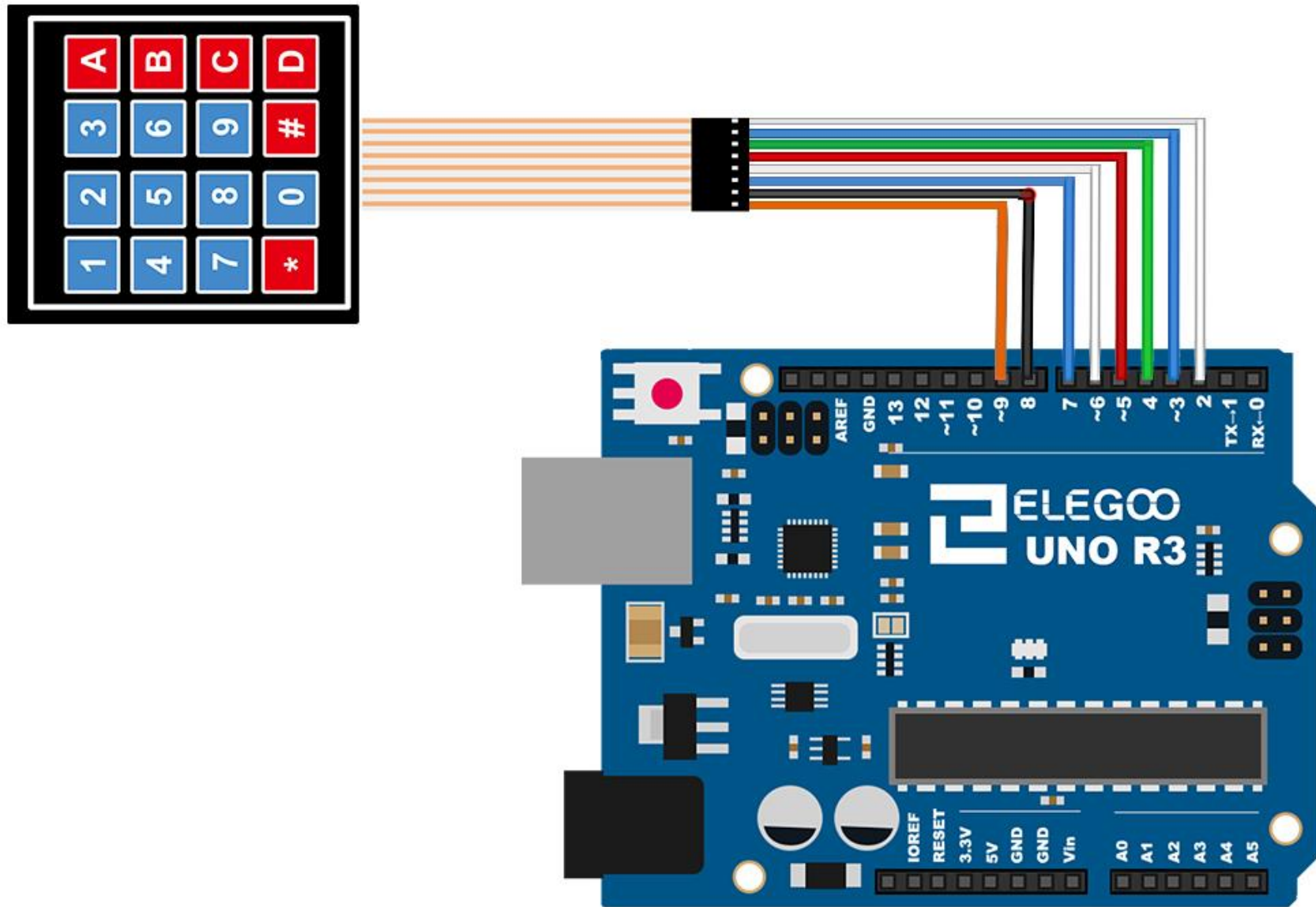
(8) M-M Kabel

Verbindung

Schema



Schaltplan



Wenn Sie die Pins an das UNO R3-Board anschließen, verbinden wir sie mit den digitalen Ausgangspins, D9-D2. Wir verbinden den ersten Pin der Tastatur mit D9, den zweiten Pin mit D8, den dritten Pin mit D7, den vierten Pin mit D6, den fünften Pin mit D5, den sechsten Pin mit D4, den siebten Pin mit D3, und den achten Pin mit D2.

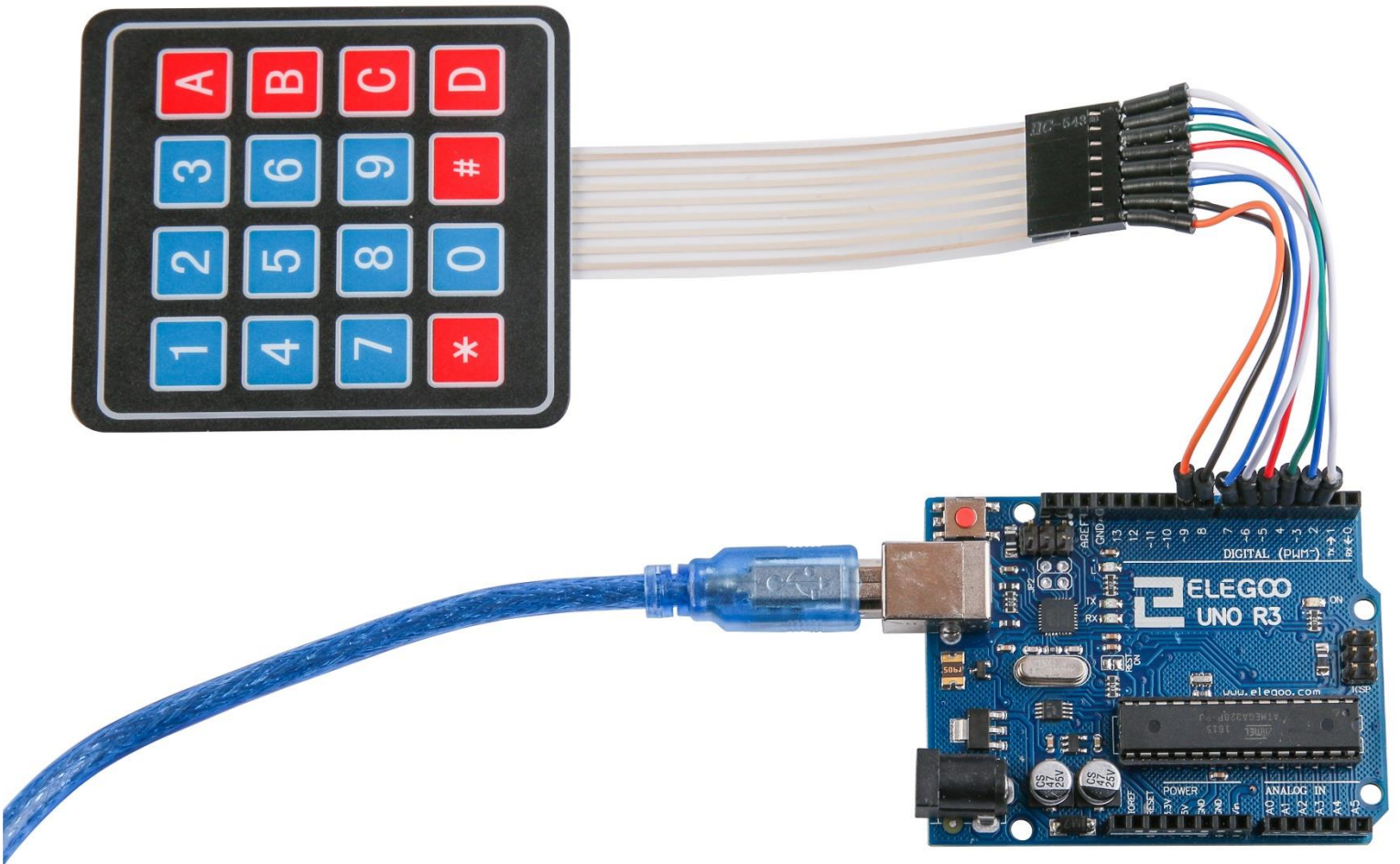
Diese sind Verbindungen in einer Tabelle:

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

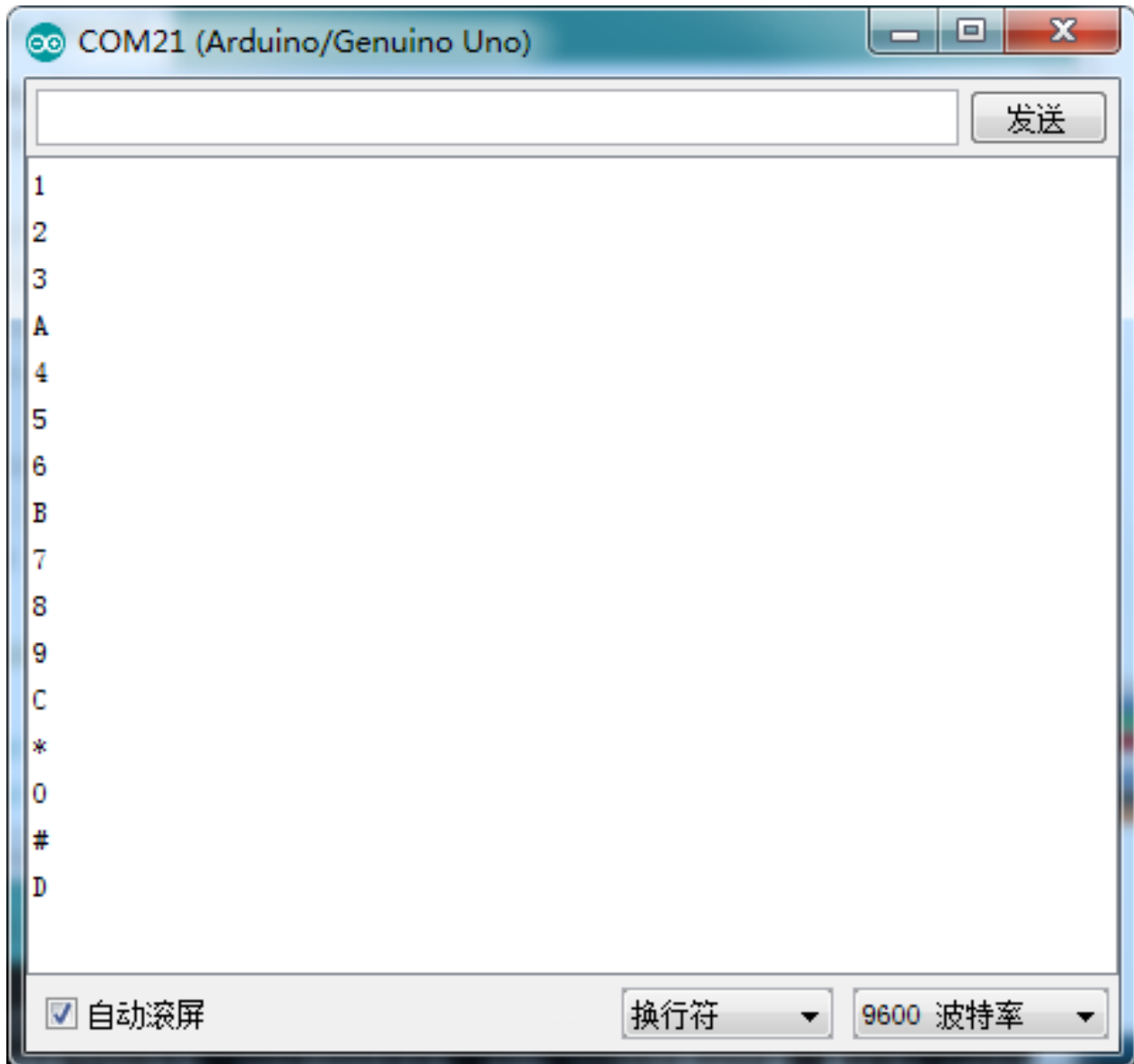
Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek <Keypad> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.



Mit diesem Code, sobald wir eine Taste auf der Tastatur drücken, sollte sie auf dem seriellen Monitor der Arduino-Software angezeigt werden, nachdem der Code kompiliert und auf das UNO R3-Board hochgeladen wurde.



Lektion 12: DHT11 Temperatur- und Feuchtesensor

Übersicht

In diesem Tutorial werden Sie erfahren, wie Sie einen DHT11 Temperatur- und Feuchtesensor verwenden.

Er ist für die meisten Programme, die Temperatur- und Feuchtigkeitsmessungen verfolgen müssen, präzise genug.

Hier werden Sie eine Bibliothek, die speziell für diese Sensoren, die den Code kurz und einfach schreiben lässt, entwickelt wird, wieder verwenden.

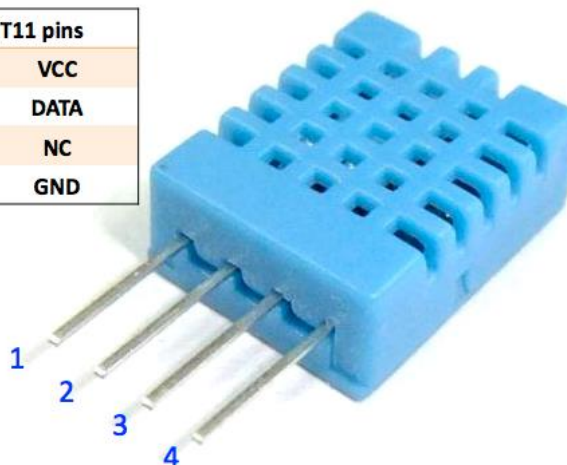
Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) DHT11 Modul
- (3) F-M Kabel

Komponentenanweisung

Temperatur- und Feuchtesensor:

DHT11 pins	
1	VCC
2	DATA
3	NC
4	GND



DHT11 digitaler Temperatur- und Feuchtesensor ist ein zusammengesetzter Sensor, der einen kalibrierten digitalen Signalausgang der Temperatur und der Feuchtigkeit aufweist. Anwendung einer Erfassungstechnologie von dedizierten digitalen Modulen und einer Sensortechnologie von Temperatur und Feuchtigkeit stellt sicher, dass das Produkt eine hohe Zuverlässigkeit und eine ausgezeichnete langfristige Stabilität aufweist. Der Sensor enthält eine resistive Wahrnehmung von

nassen Komponenten und eine NTC-Temperaturmesseinrichtung, und wird mit einem hochleistungsfähigen 8-Bit-Mikrocontroller verbunden.

Anwendungen: HVAC, Luftentfeuchter, Prüf- und Inspektionsgeräte, Konsumgüter, Automobile, automatische Controller, Datenlogger, Wetterstationen, Haushaltsgeräte, Feuchtigkeitsregulatoren, medizinische und andere Feuchtemess- und Feuchtsteuerungstechniken.

Produktparameter

Relative Luftfeuchtigkeit:

Auflösung: 16 Bit

Wiederholbarkeit: $\pm 1\%$ RH

Genauigkeit: Bei $25\text{ }^{\circ}\text{C}$ $\pm 5\%$ RH

Austauschbarkeit: Vollständig austauschbar

Reaktionszeit : $1/e$ (63%) von $25\text{ }^{\circ}\text{C}$ 6s

1m/s Luft 6s

Hysteresis: $<\pm 0,3\%$ RH

Langfristige Stabilität: $<\pm 0,5\%$ RH / Jahr in

Temperatur:

Auflösung: 16Bit

Wiederholbarkeit: $\pm 0,2\text{ }^{\circ}\text{C}$

Reichweite: Bei $25\text{ }^{\circ}\text{C}$ $\pm 2\text{ }^{\circ}\text{C}$

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

Elektrische Eigenschaften

Versorgungsspannung: DC 3,5 ~ 5,5V

Versorgungsstrom: Messung 0,3 mA Standby 60 μ A

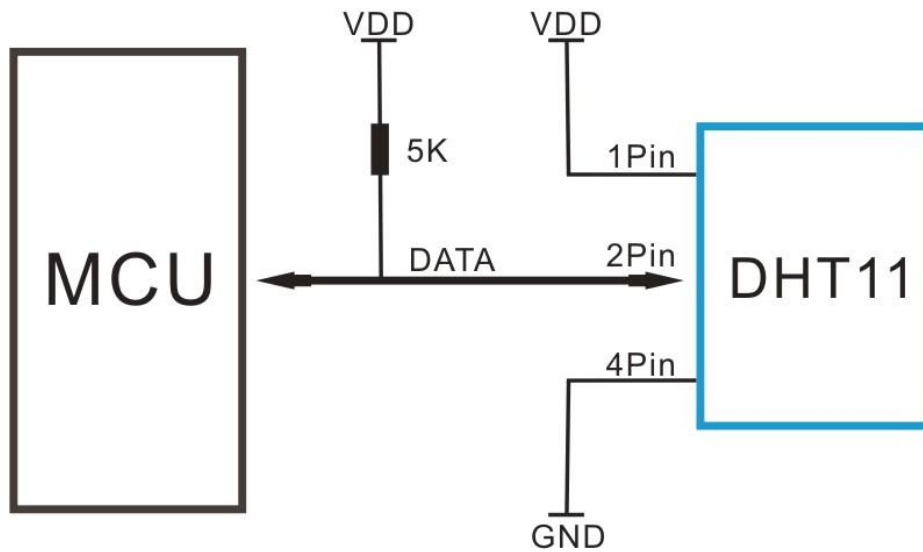
Probenahmeintervall: mehr als 2 Sekunden

Beschreibung über Pin:

1. Die VDD-Versorgungsspannung von 3,5 ~ 5,5 V DC
2. DATA serielle Daten, ein einzelner Bus
3. NC, leerer Pin

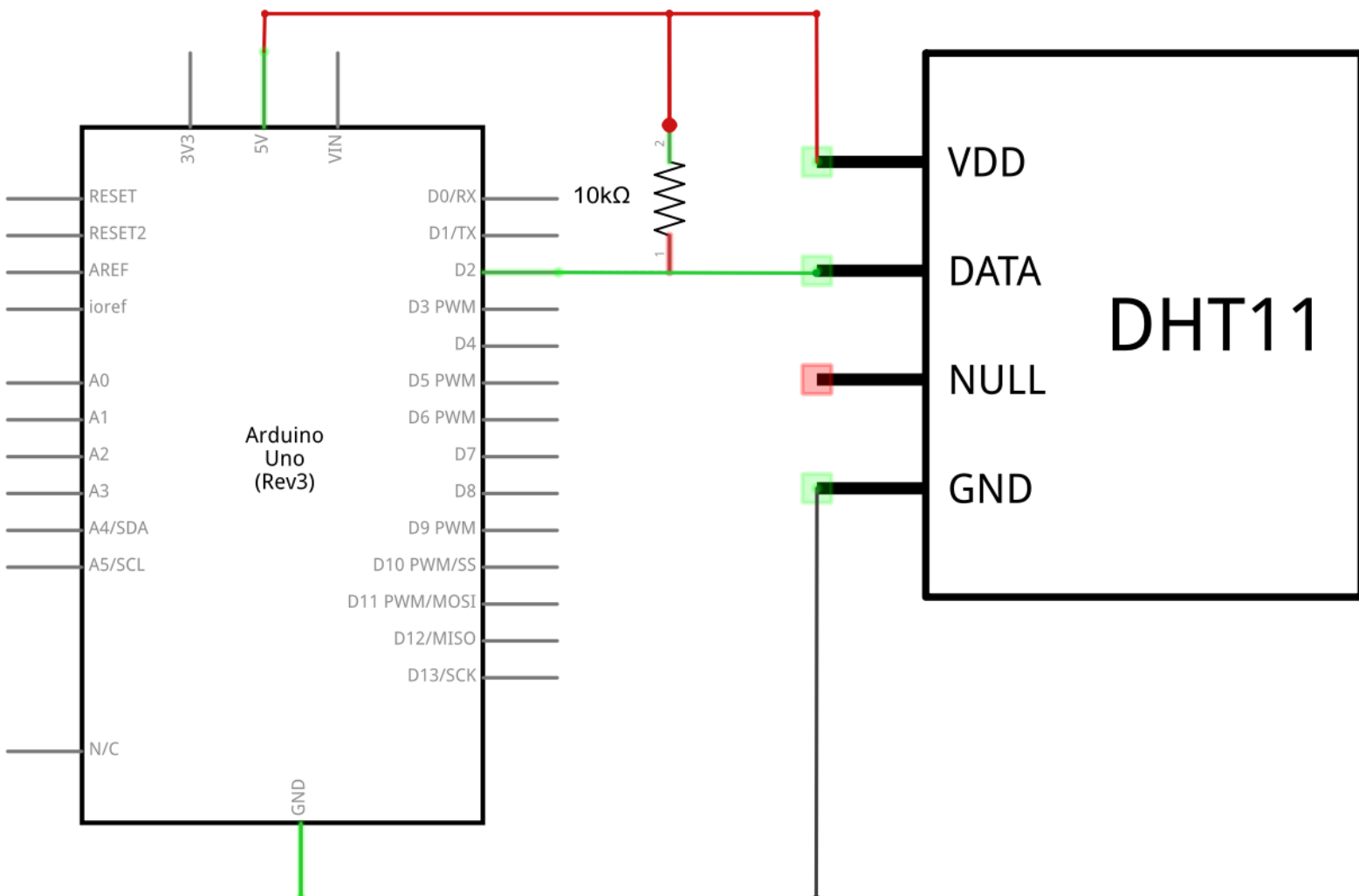
4. GND Masse, die negative Energie

Typische Anwendung

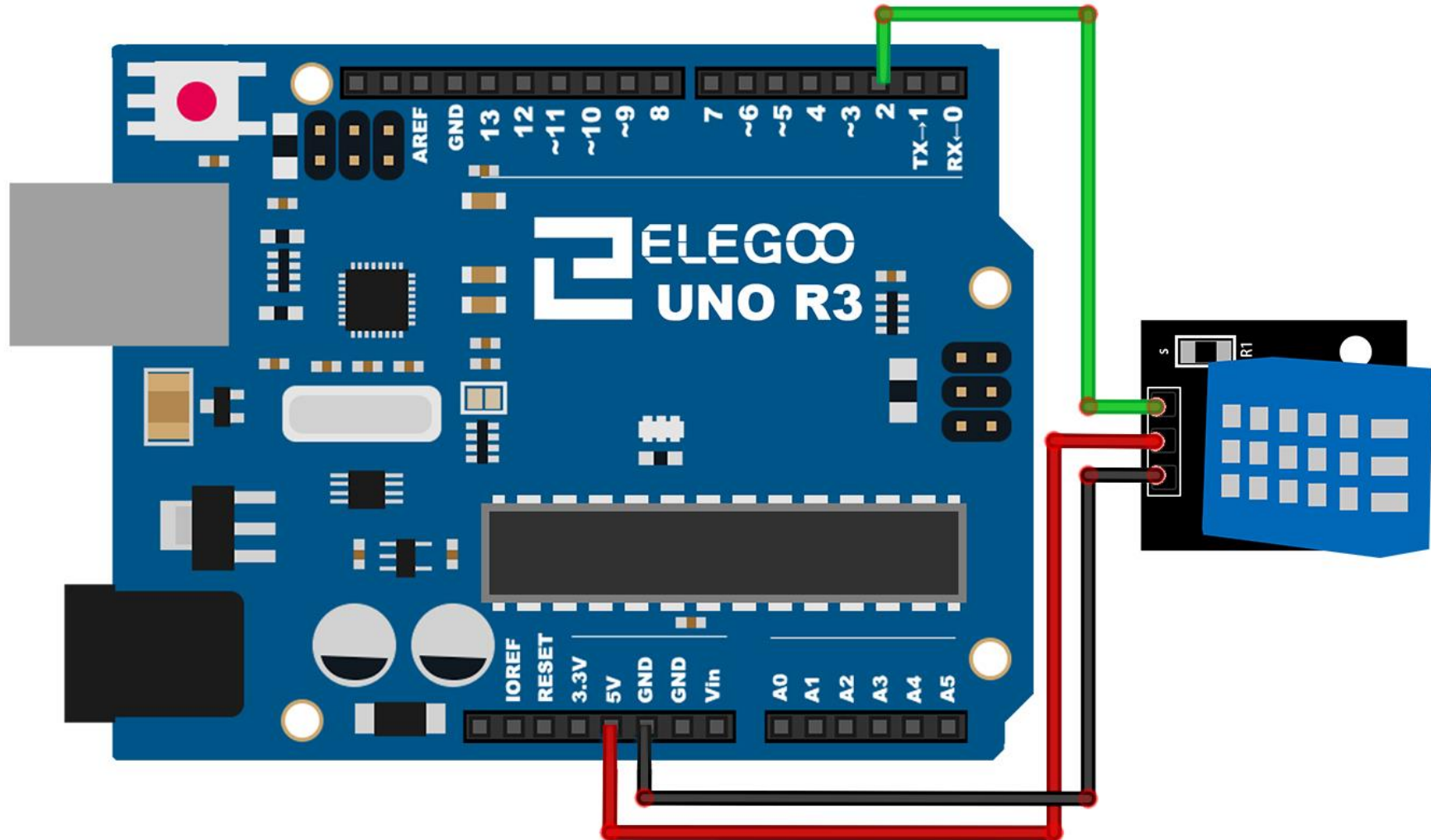


Verbindung

Schema



Schaltplan



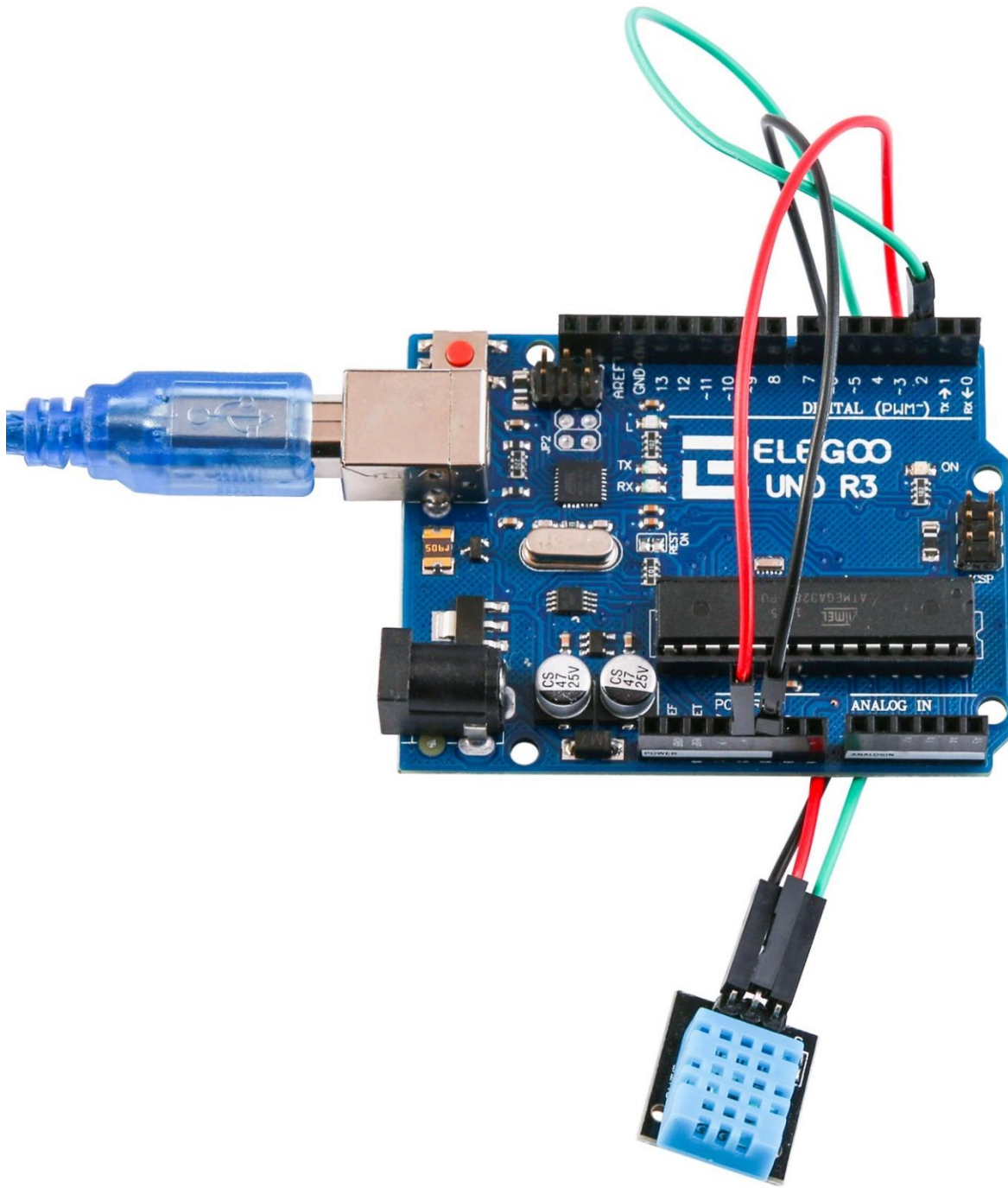
Wie Sie sehen können, dass Sie nur 3 Anschlüsse an den Sensor brauchen, da einer der Pins nicht verwendet wird.

Die Anschlüsse sind: Spannung, Masse und Signal, der an jeden analogen Pin auf dem UNO angeschlossen wird.

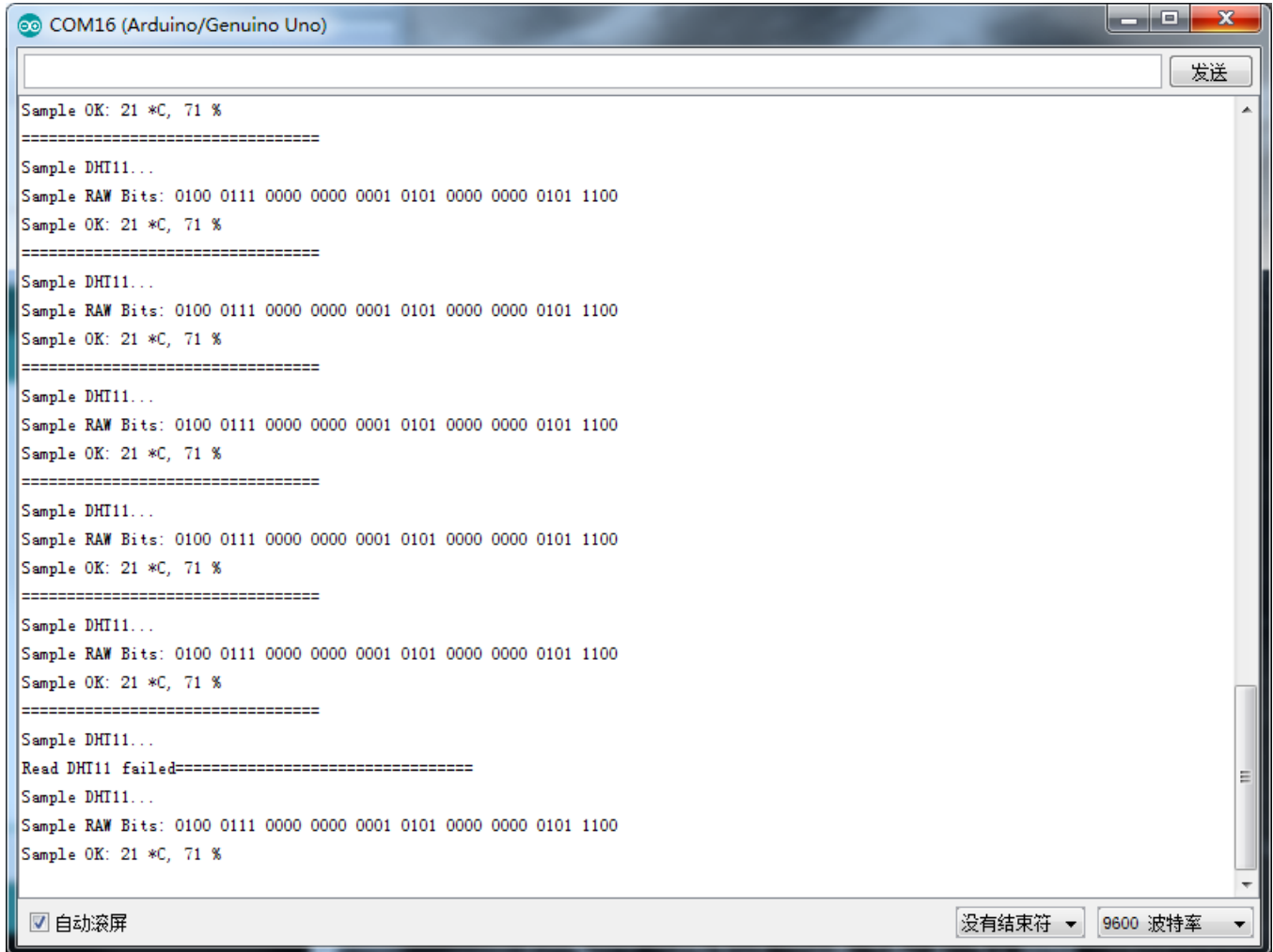
Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek <SimpleDHT> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.



Laden Sie das Programm hoch, und danach schalten Sie den Monitor ein, so können wir die Daten wie folgt sehen: (Die Temperatur der Umgebung wird angezeigt, und wir können sehen, dass sie bei 21 Grad liegt)



Lektion 13: Analoges Joystick-Modul

Übersicht

Analoge-Joysticks bieten eine gute Möglichkeit an, eine gewisse Kontrolle in Ihre Programme hinzuzufügen.

In diesem Tutorial werden wir erfahren, wie wir das analoge Joystick-Modul verwenden.

Erforderliche Komponenten

(1) Elegoo UNO R3

(1) Joystick-Modul

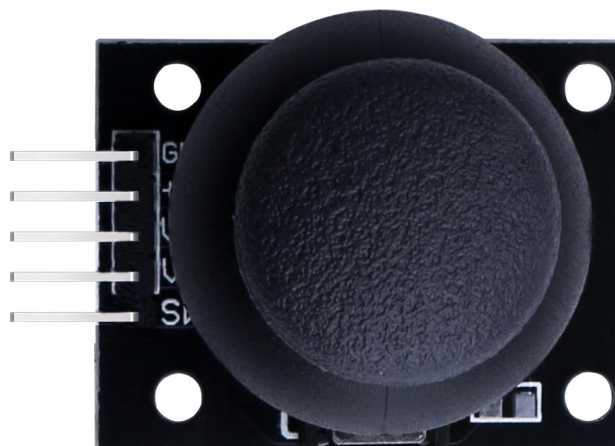
(5) F-M Kabel

Komponentenanweisung

Joystick

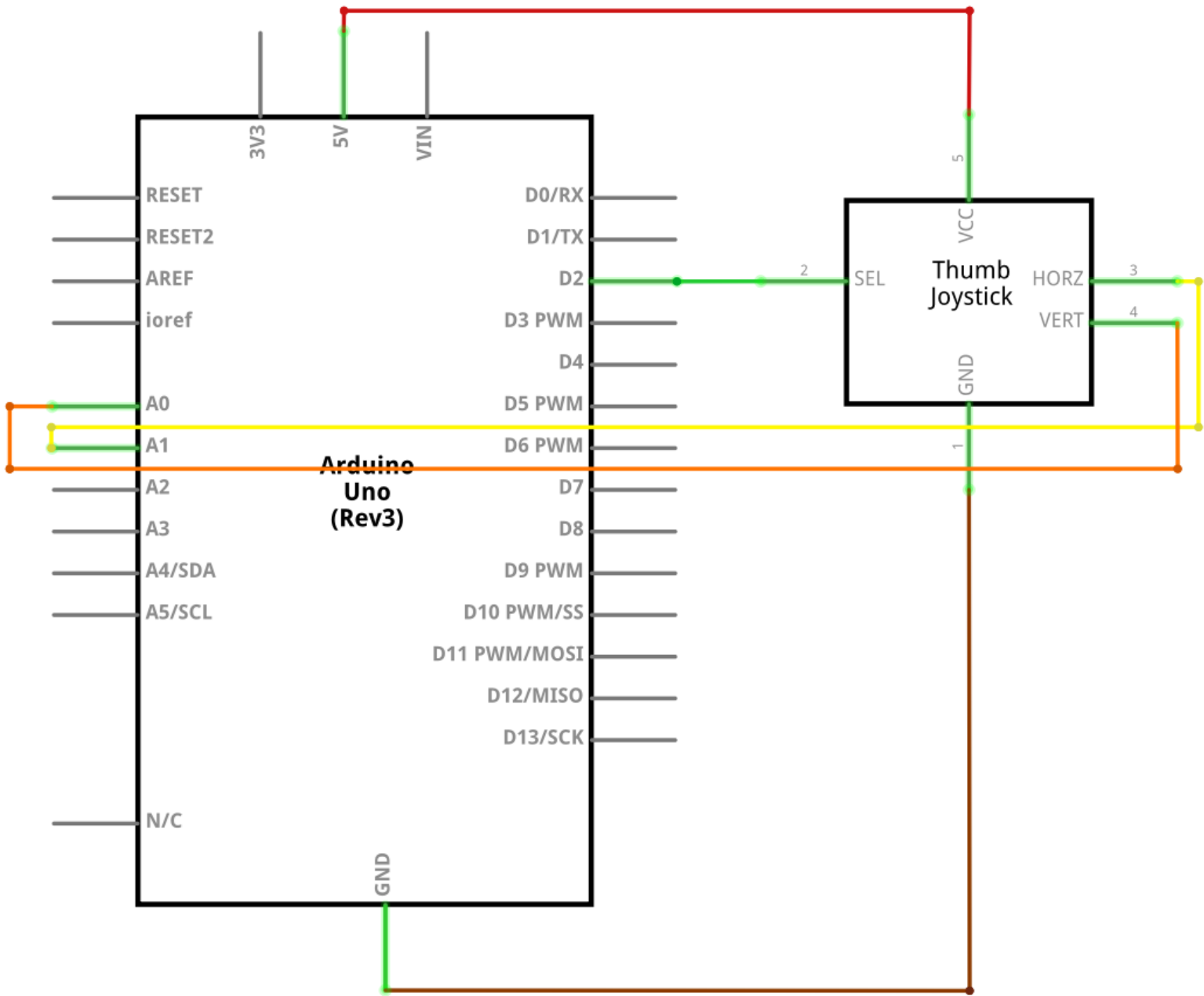
Das Modul weist 5 Pins auf: Vcc, Masse, X, Y, Key. Achten Sie darauf, dass die Etiketten auf Ihrem Modul etwas unterschiedlich sein können, je nachdem, wo Sie das Modul bekamen. Der Thumb-Stick ist analog und sollte viel genauere Ablesungen als einfache ‚gerichtete‘ Joysticks, die Tasten in einigen Formen oder mechanische Schalter verwenden, liefern. Zusätzlich können Sie den Joystick nach unten drücken (relativ stark), um einen Druckknopf ‚drücken, um auszuwählen‘ zu aktivieren.

Wir müssen analoge Arduino-Pins verwenden, um die Daten aus den X/Y-Pins abzulesen, und einen digitalen Pin zur Ablesung der Taste verwenden. Der Key-Pin wird mit der Masse verbunden, wenn der Joystick nach unten gedrückt wird, ansonsten ist er schwimmend. Um stabile Ablesungen auf dem Key-/Select-Pin zu erhalten, muss er über einen Pull-up-Widerstand mit Vcc verbunden werden. Die eingebauten Widerstände auf den digitalen Arduino-Pins können verwendet werden. Ein Tutorial für die Aktivierung der Pull-up-Widerstände für Arduino-Pins ist, dass sie als Eingänge konfiguriert werden.

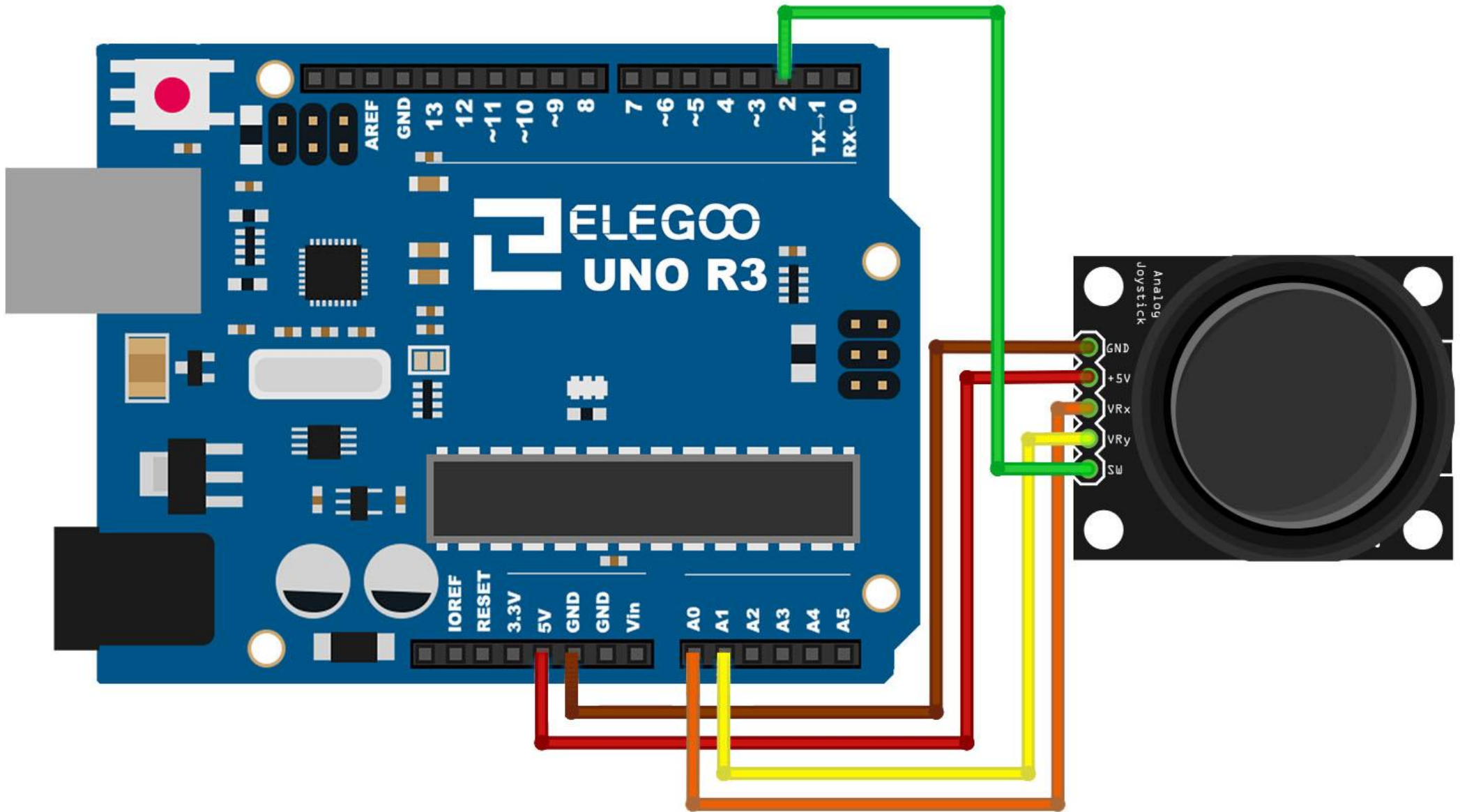


Verbindung

Schema



Schaltplan



Wir brauchen 5 Anschlüsse an den Joystick.

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

„Y und X“ sind analoge Anschlüsse, und „Key“ ist ein digitaler Anschluss. Wenn der Schalter nicht erforderlich ist, können Sie dann nur 4 Pins verwenden.

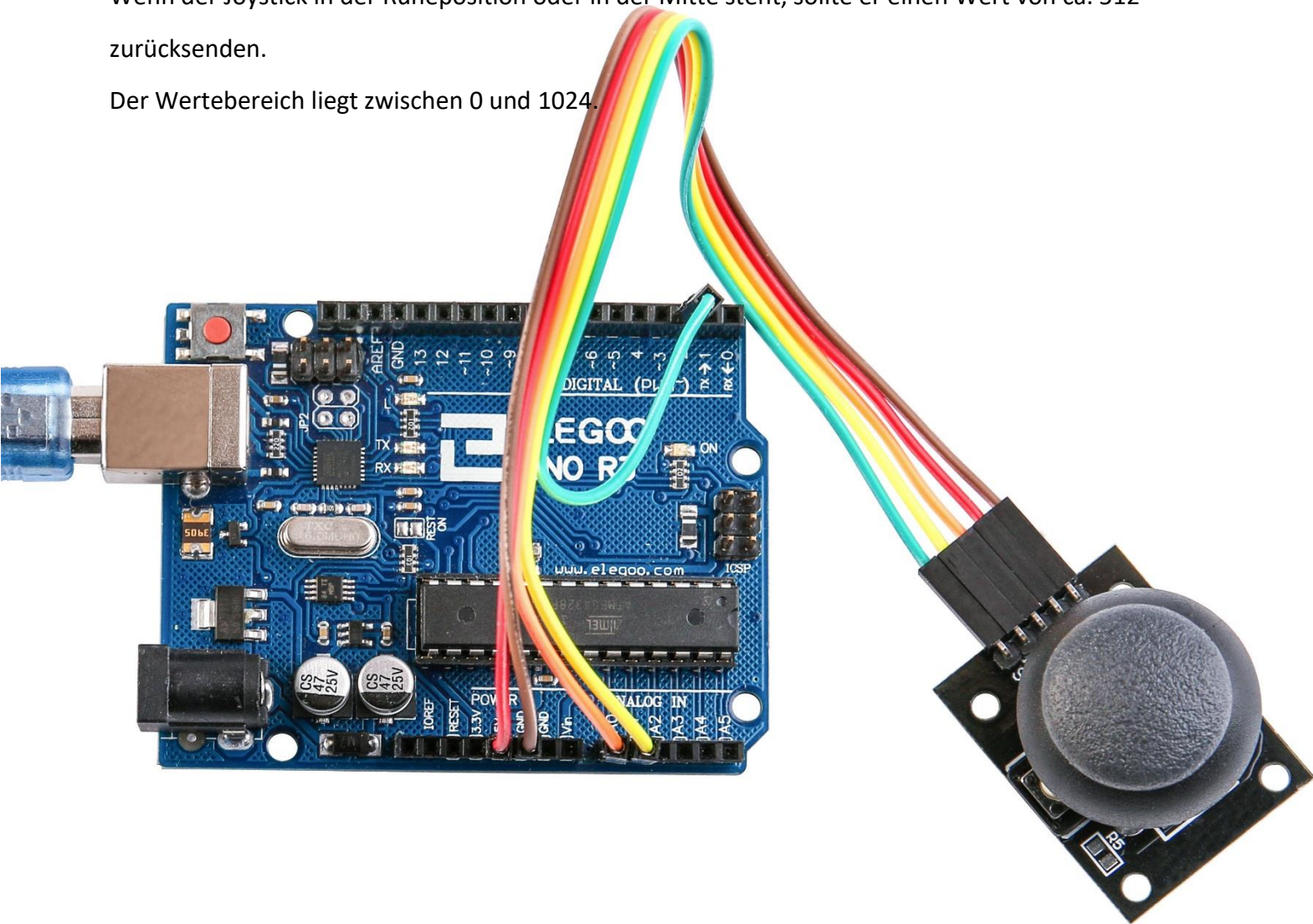
Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

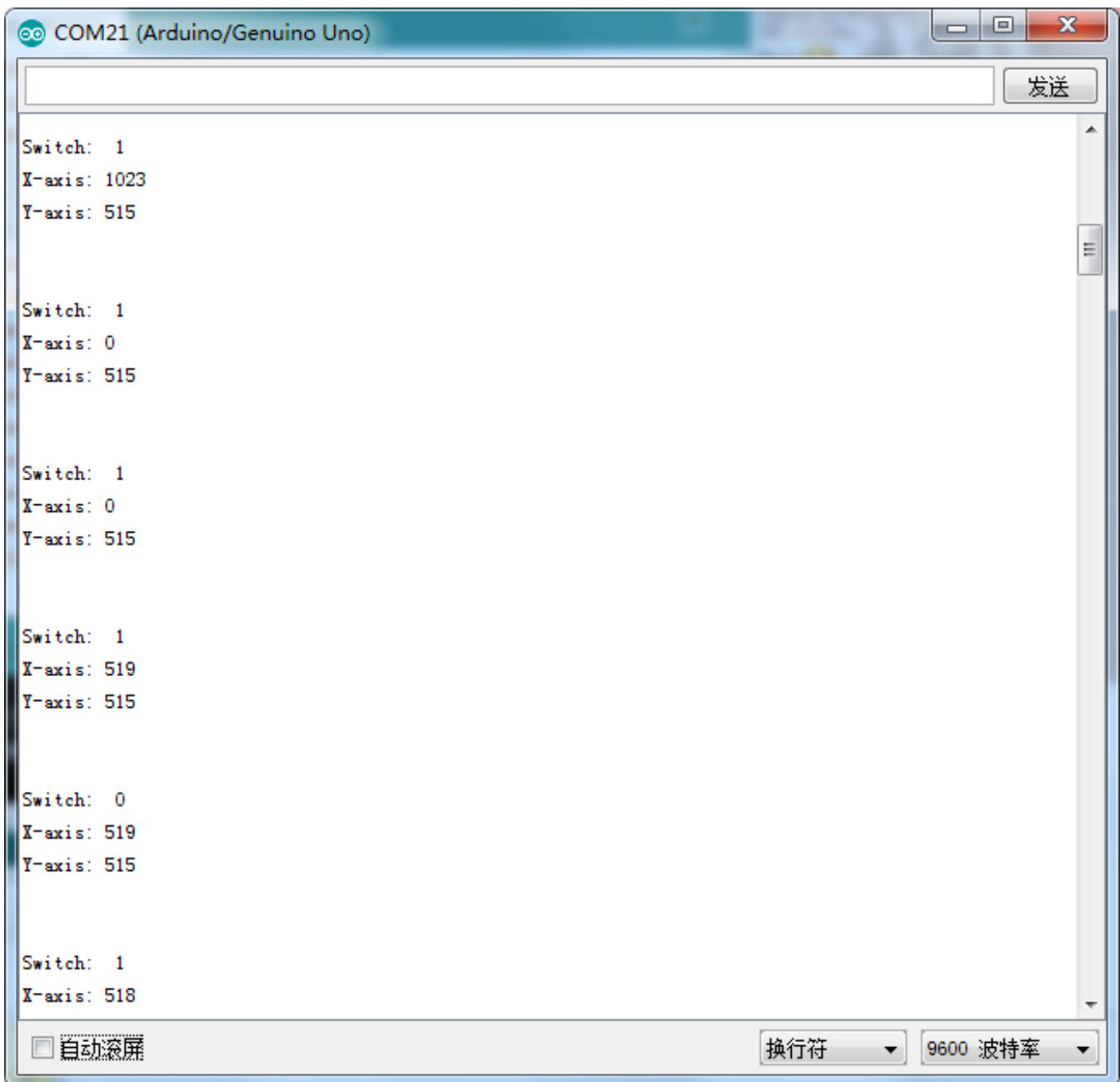
Analoge Joysticks sind grundsätzlich Potentiometer, so senden sie Analogwerte zurück.

Wenn der Joystick in der Ruheposition oder in der Mitte steht, sollte er einen Wert von ca. 512 zurücksenden.

Der Wertebereich liegt zwischen 0 und 1024.



Open the monitor then you can see the data as blow:



Lektion 14: IR-Empfängermodul

Übersicht

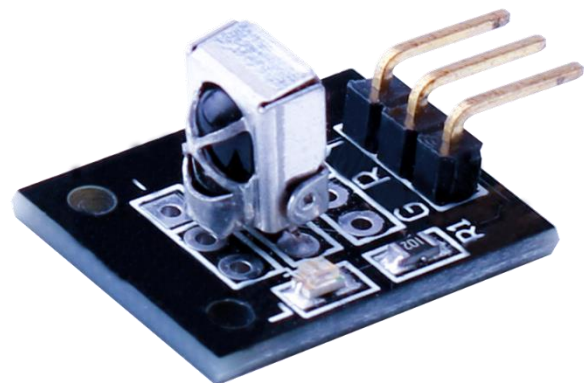
Die Verwendung einer IR-Fernbedienung bietet eine gute Möglichkeit an, Ihr Programm drahtlos gesteuert zu werden.

Infrarotfernbedienungen sind einfach und leicht zu bedienen. In diesem Tutorial werden wir den IR-Empfänger an das UNO anschließen, und dann werden wir eine Bibliothek, die für diesen besonderen Sensor entwickelt wird, verwenden.

In Ihrem Sketch werden Sie alle IR-Hexadezimalcodes, die auf der Fernbedienung verfügbar sind, haben, und werden Sie auch erkennen, ob der Code erkannt wurde, und ob Sie eine Taste nach unten gedrückt halten.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) IR-Empfängermodul
- (1) IR-Fernbedienung
- (3) F-M Kabel



Komponentenanweisung

IR-EMPFÄNGERSENSOR:

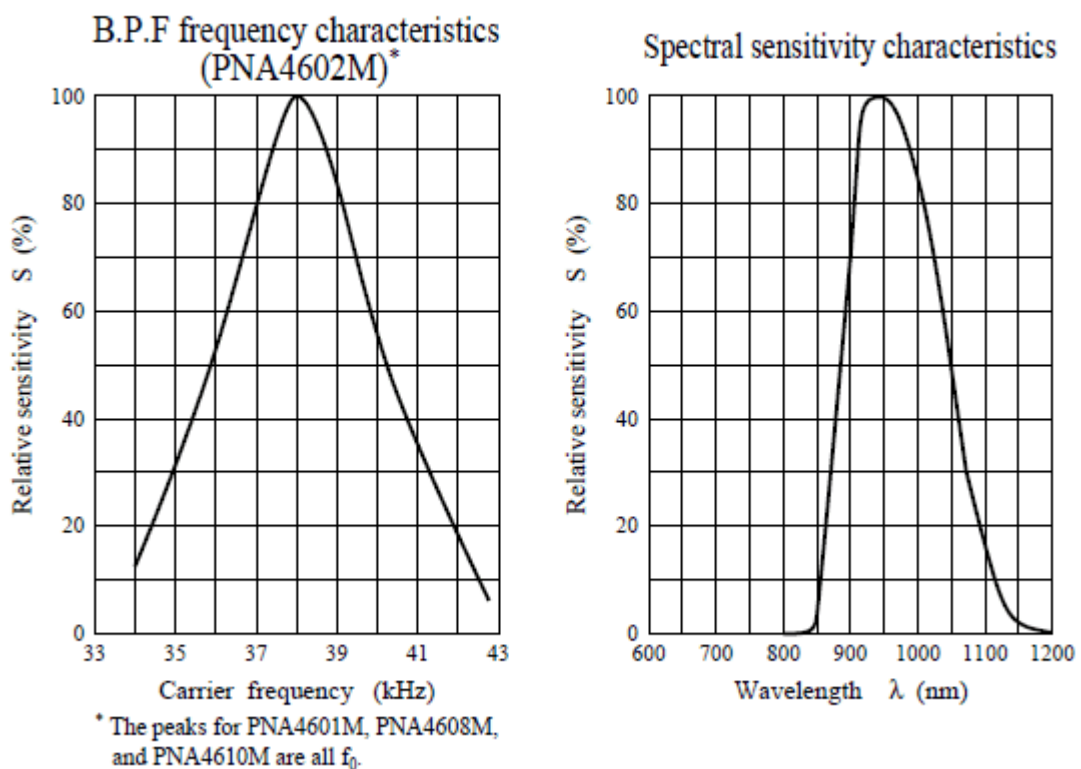
IR-Detektoren sind kleine Mikrochips mit einer Fozelle, die abgestimmt werden, Infrarotlicht zu hören. Sie werden fast immer für die Fernbedienungserkennung verwendet - alle Fernseher und DVD-Player haben einen dieser Mikrochips auf der Vorderseite, um das IR-Signal aus der Fernbedienung zu hören. Innerhalb der Fernbedienung ist es eine passende IR-LED, die IR-Impulse emittiert, um dem Fernseher anzuweisen, einzuschalten, auszuschalten oder Kanäle zu wechseln. IR-Licht ist für menschliche Augen nicht sichtbar, was bedeutet, dass es ein wenig mehr Zeit nimmt, um ein Setup zu testen.

Es gibt ein paar Unterschiede zwischen diesen Fozellen und den sogenannten CdS-Fozellen:

IR-Detektoren sind speziell für die Filterung des Infrarotlichtes entwickelt, und bei der Erkennung des sichtbaren Lichtes sind sie nicht so gut. Auf der anderen Seite sind Fotozellen bei der Erkennung des gelben/grünen sichtbaren Lichtes gut, aber bei der Erkennung des IR-Lichtes nicht so gut.

- Innerhalb der IR-Detektoren gibt es einen Demodulator, der moduliertes IR bei 38KHz sucht. Nur eine leuchtende IR-LED wird nicht erkannt, und es muss die Pulsweitenmodulation (PWM), die bei 38KHz blinkt, sein. Fotozellen weisen keine Art von Demodulator auf, und sie können jede Frequenz (einschließlich Gleichstrom) innerhalb einer Reaktionsgeschwindigkeit der Fotozelle (die bei ca. 1KHz liegt) erkennen.
- IR-Detektoren werden digital ausgegeben - entweder sie 38KHz IR und Ausgang-LOW (0V) erkennen, oder sie keinen Ausgang-HIGH (5V) erkennen. Fotozellen wirken wie Widerstände, und der Widerstand ändert sich, je nachdem, wie viel Licht sie ausgesetzt sind.

Was Können Sie Messen



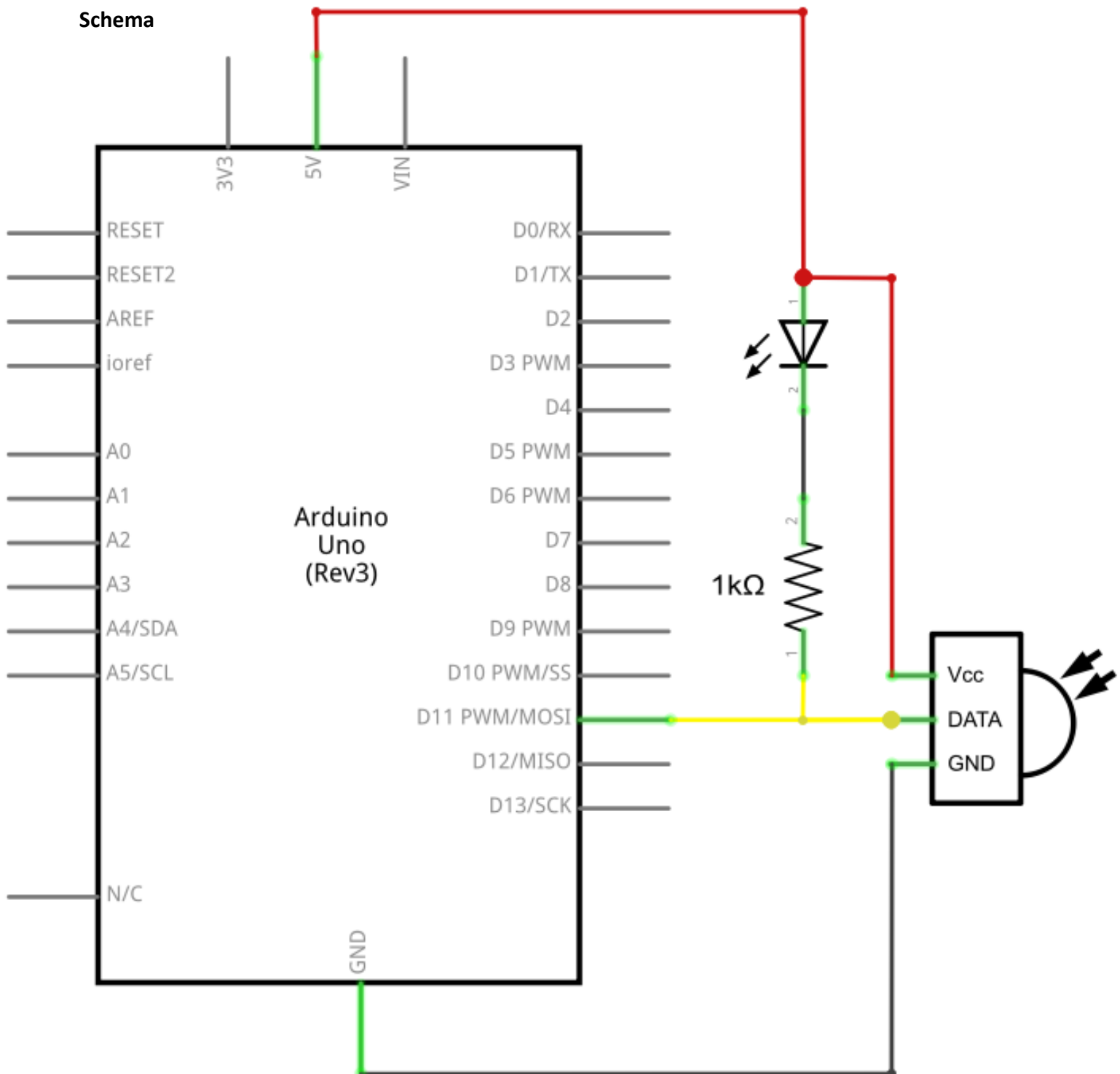
Wie Sie aus diesen Kurvenbildern des Datenblattes sehen, dass die erkannte Spitzenfrequenz bei 38KHz liegt und dass die LED-Spitzenfarbe 940nm ist. Sie können ab ca. 35KHz bis 41KHz verwenden, aber die Empfindlichkeit wird zurückgehen, so dass sie aus der Ferne auch nicht erkennen wird.

Ebenso können Sie auch LEDs ab 850 bis 1100 nm verwenden, aber sie werden nicht so gut wie ab 900 bis 1000nm funktionieren, so stellen Sie sicher, dass Sie passende LEDs erhalten! Überprüfen Sie das Datenblatt für Ihre IR-LED, um die Wellenlänge zu bestätigen.

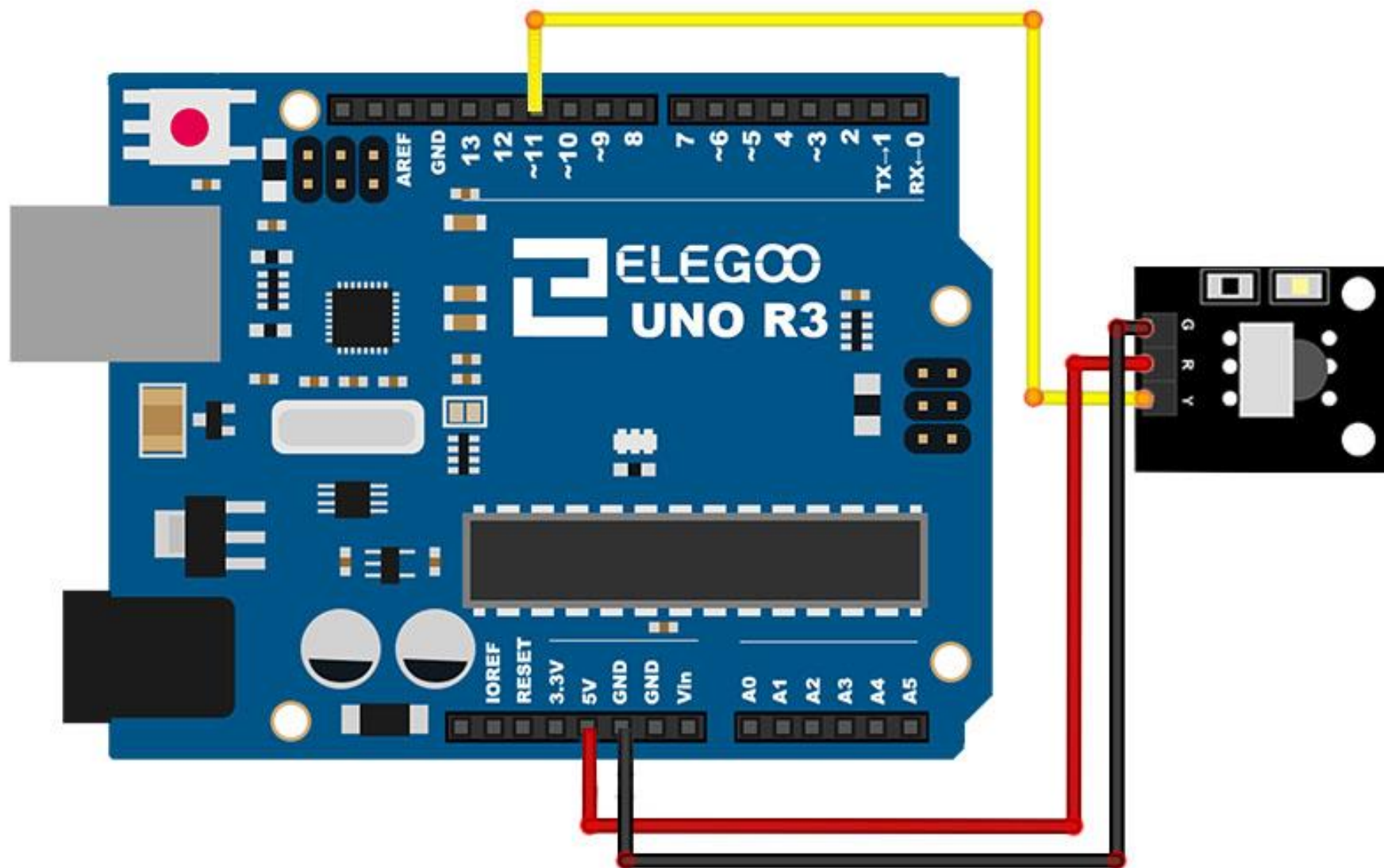
Versuchen Sie, eine LED von 940 nm zu erhalten - denken Sie daran, dass 940 nm ist kein sichtbares Licht (Infrarot)!

Verbindung

Schema



Schaltplan



Es gibt 3 Anschlüsse an IR-Empfänger.

Die Anschlüsse sind: Signal, Spannung und Masse.

Der Anschluss „-“ ist die Masse, „S“ ist Signal, und der mittlere Pin ist 5V-Spannung.

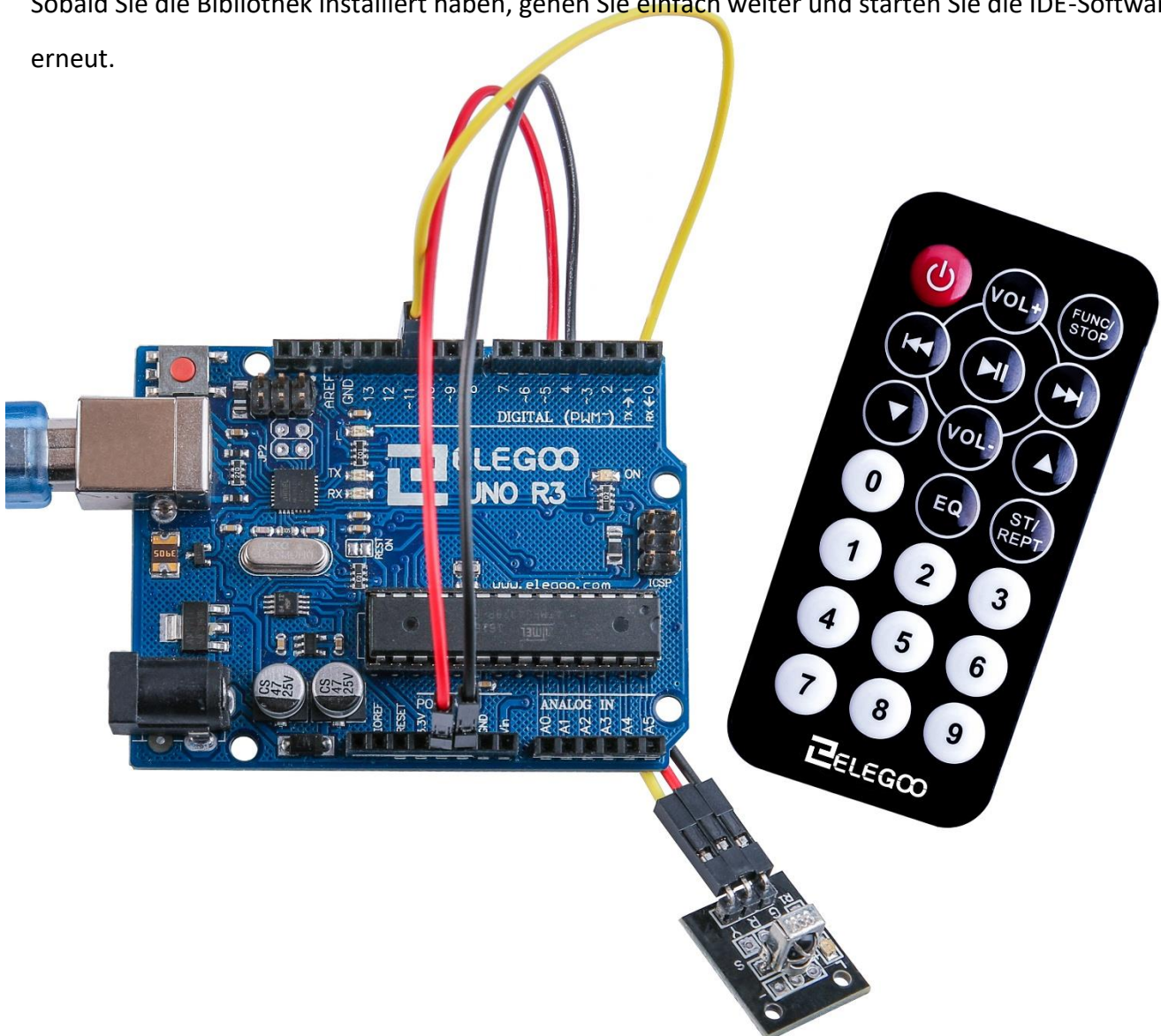
Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefodert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

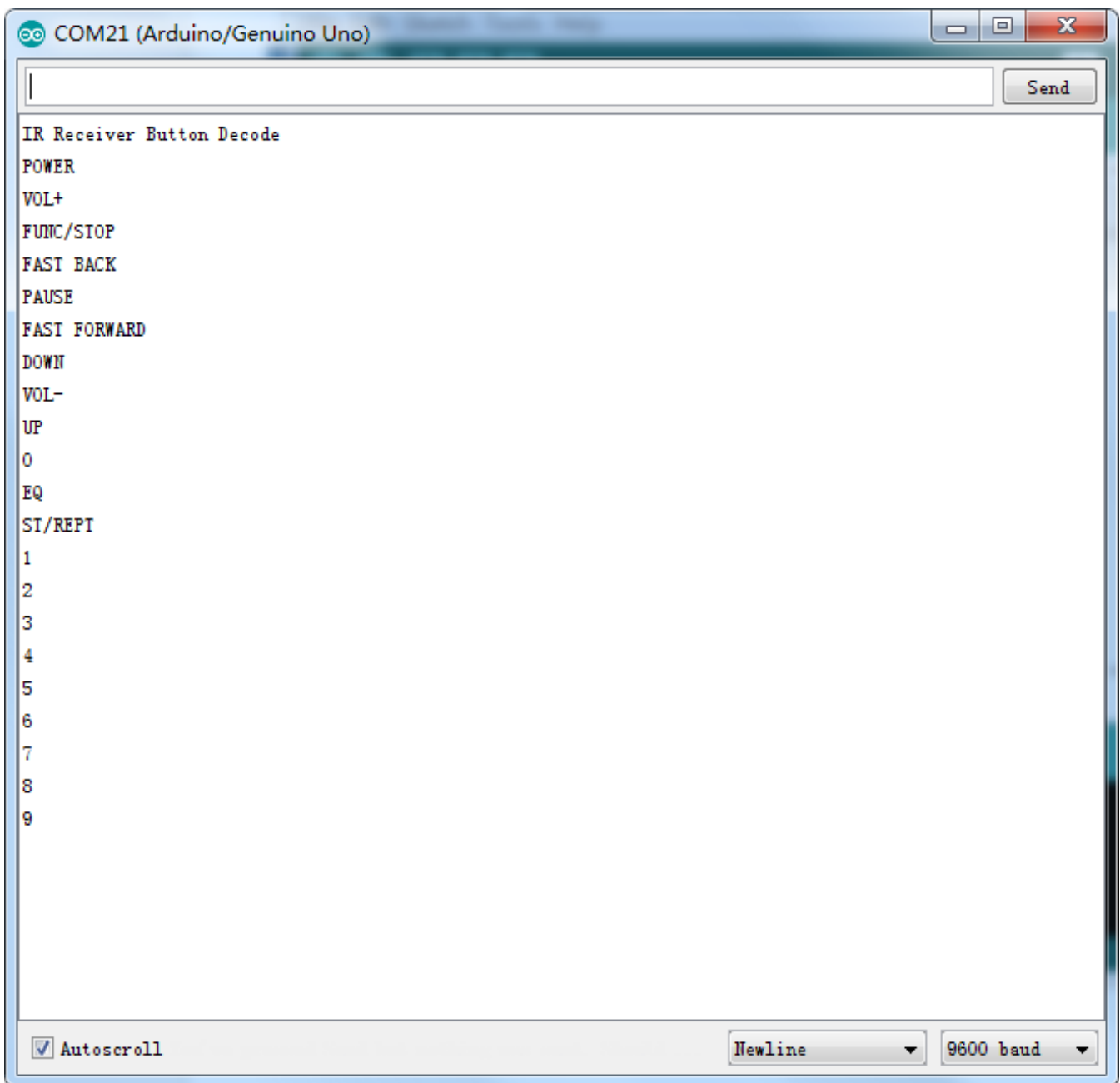
Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek <IRremote> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.

Als nächstes werden wir <RobotIRremote> aus dem Bibliotheksordner entfernen, weil die Bibliothek mit der Bibliothek, die wir verwenden werden, Konflikte hat. Sie können sie einfach zurück in die Bibliothek schieben, sobald Sie Ihren Mikrocontroller fertig programmiert haben.

Sobald Sie die Bibliothek installiert haben, gehen Sie einfach weiter und starten Sie die IDE-Software erneut.



Open the monitor then you can see the data as blow:



Lektion 15: MAX7219 LED Dot-Matrix-Modul

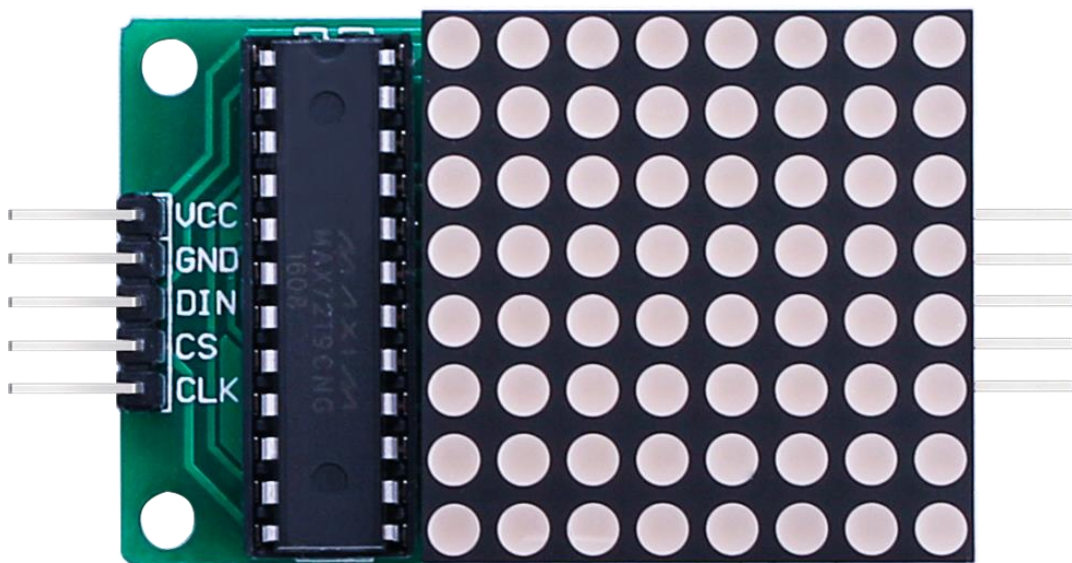
Übersicht

In diesem Tutorial werden Sie ein MAX7219 anschließen und werden Sie den Text blättern.

Da diese Module den MAX7219 LED-Treiber-Chip verwenden, können wir die 64 LEDs jedes Moduls mit nur drei Pins auf unserem UNO ein- und ausschalten.

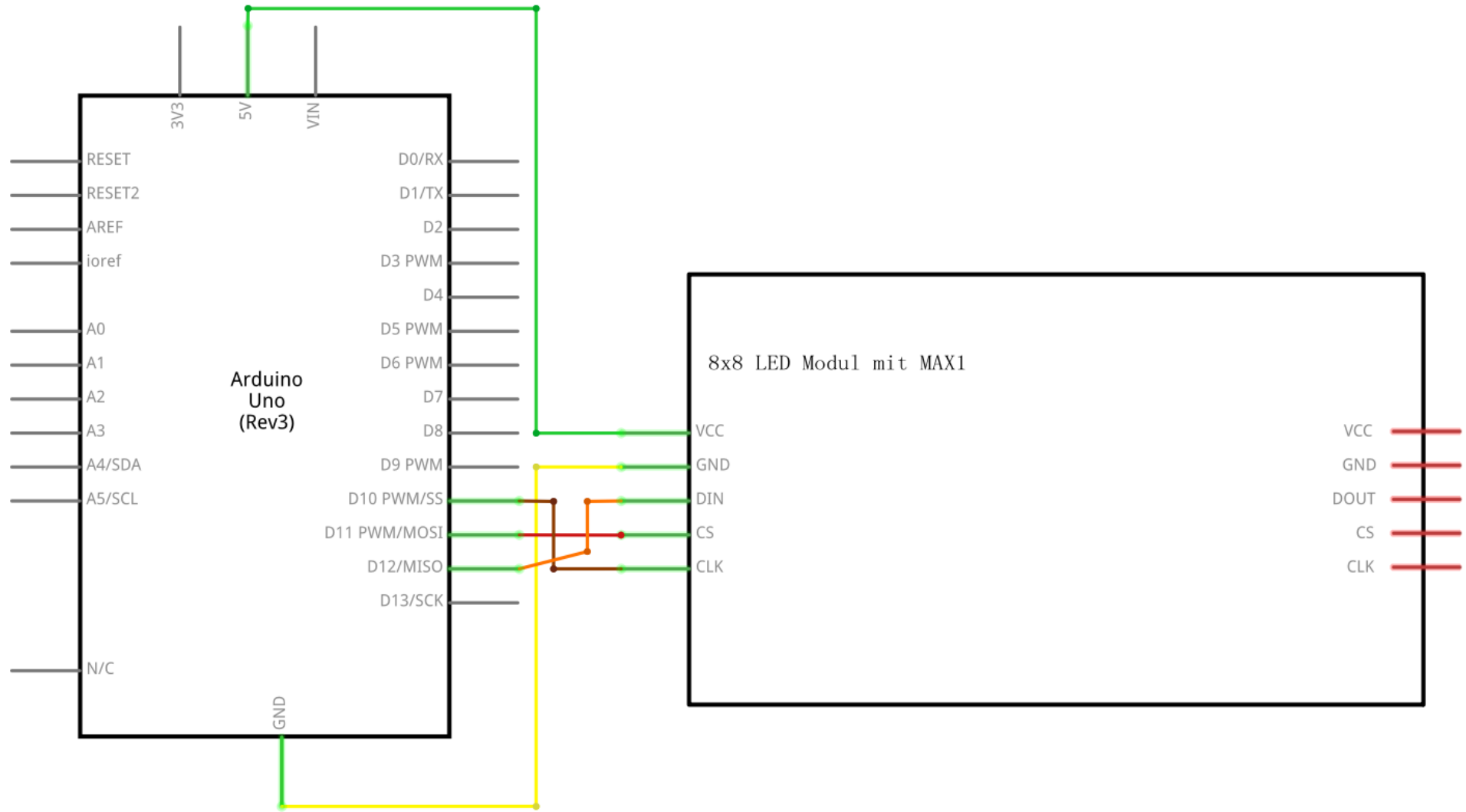
Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Max7219 Modul
- (5) F-M Kabel

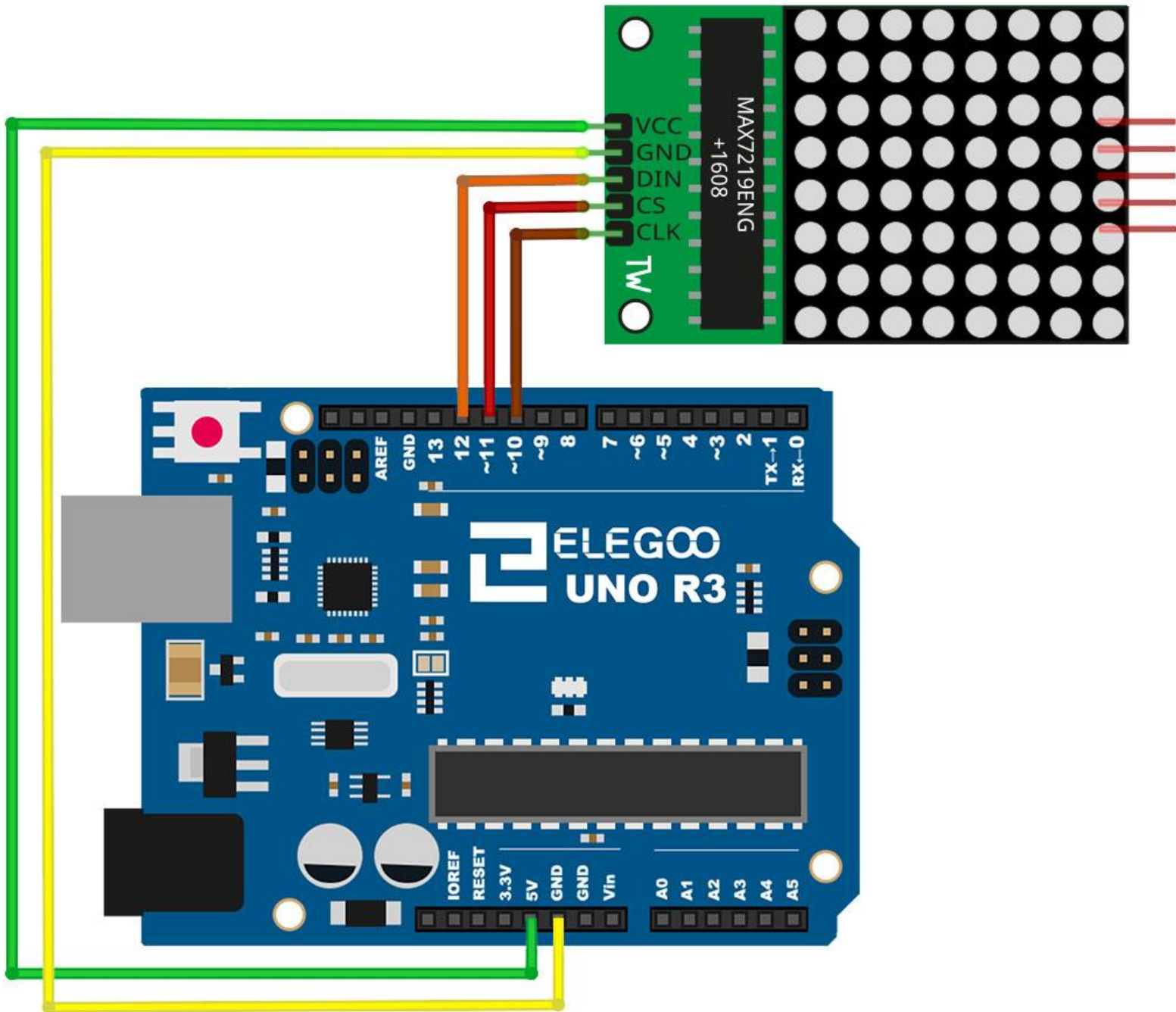


Verbindung

Schema



Schaltplan



VCC und Masse sind mit dem Arduino verbunden.

Pin 12 wird an DIN, Pin 10 wird an CS und Pin 11 wird an CLK angeschlossen.

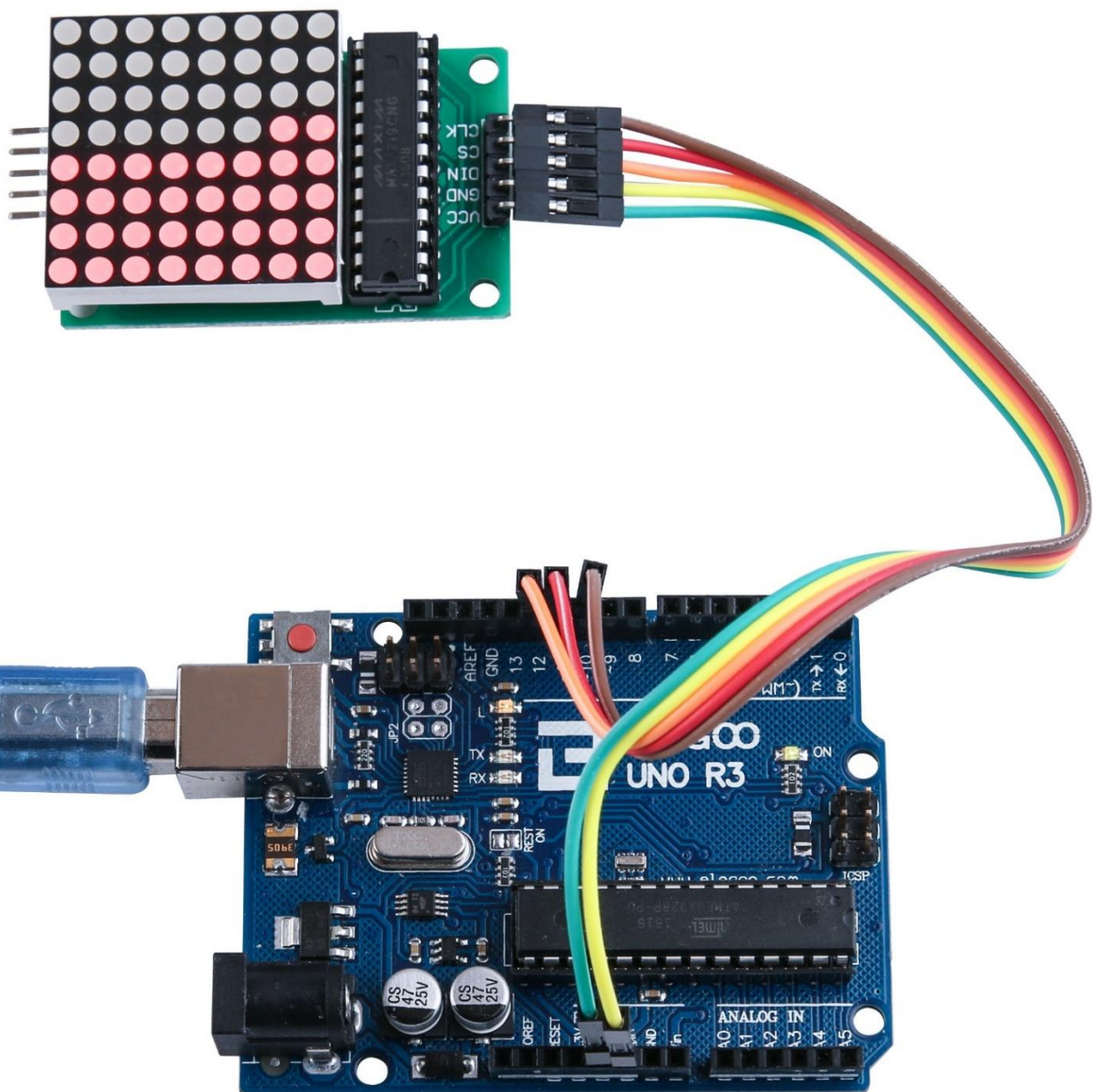
Wir verbinden dann die Module in Form von Daisy-Chain zusammen.

Code

Unser Sketch wird die „Maxmatrix“-Bibliothek verwenden, um mit MAX7219 Modulen zu kommunizieren.

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

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



Lektion 16 MPU-6050-Modul

Übersicht

In dieser Lektion lernen wir, MPU6050 Modul zu verwenden, die eines der besten IMU (Trägheit Maßeinheit) Sensoren, mit Arduino kompatibel ist. Sensoren, IMU wie MPU 6050 werden in Selbstausgleich Roboter, UAVs, Smartphones, etc. verwendet.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) MPU6050-Modul
- (5) F-M Kabel

Komponente-Einführung

MPU6050 SENSOR

Die InvenSense MPU-6050 Sensor enthält einen MEMS-Beschleunigungssensor und einen MEMS-Gyros in einem einzigen Chip. Es ist sehr genau, da sie 16-Bit analog-digital Umwandlung Hardware für jeden Kanal enthält. Daher fängt es die x-, y- und Z-Kanal zur gleichen Zeit. Der Sensor nutzt den I2C-Bus Schnittstelle mit dem Arduino.

Die MPU-6050 ist nicht teuer, insbesondere angesichts der Tatsache, dass es einen Beschleunigungssensor und ein Gyroskop vereint.

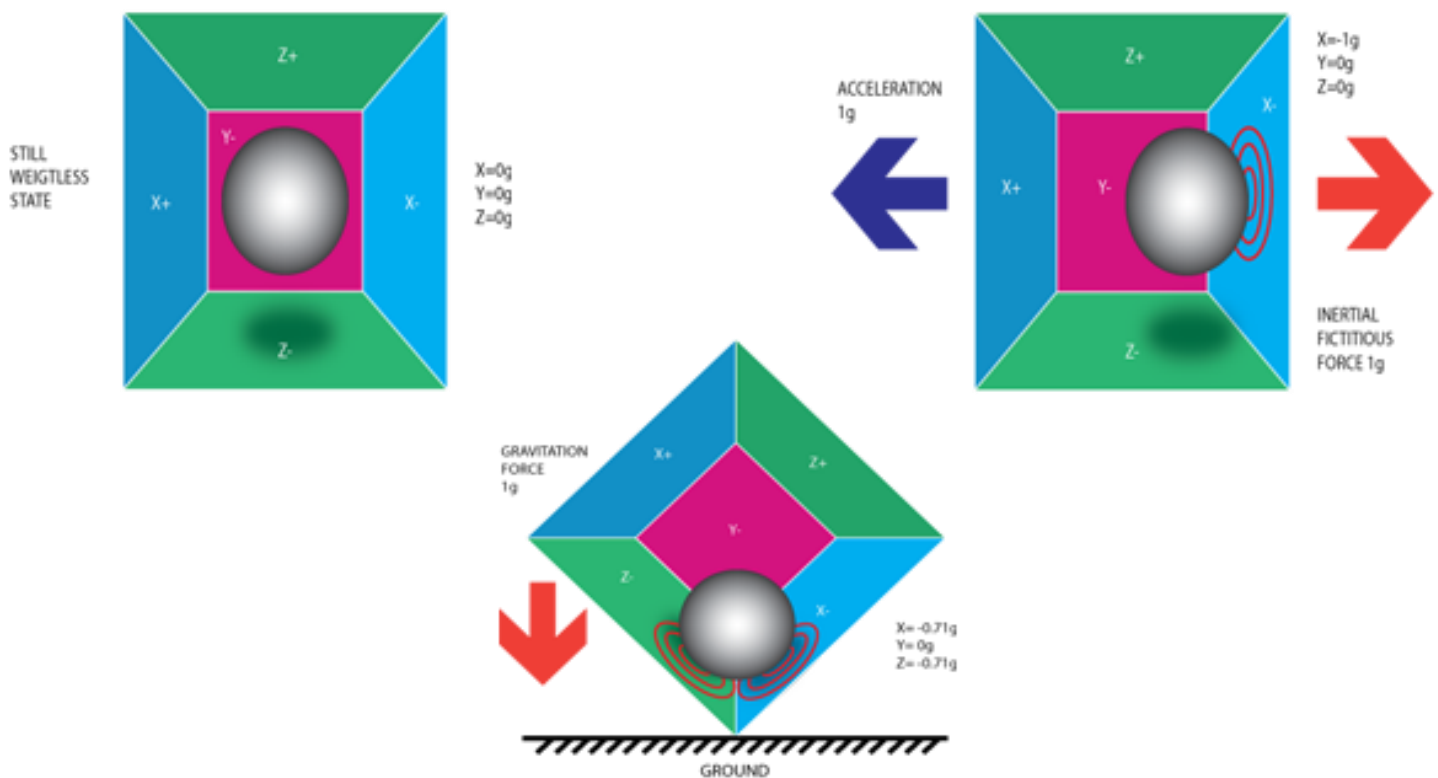


IMU-Sensoren sind eine der am meisten unvermeidlichen Art von Sensoren, die heute in allen Arten von elektronischen Geräten benutzt. Man sieht sie in Smartphones, Wearables, Spiel-Controller. IMU-Sensoren helfen uns dabei, die Haltung eines Objekts, befestigt an den Sensor im dreidimensionalen Raum. Diese Werte in der Regel in den Winkeln, damit helfen Sie uns, ihre Haltung zu bestimmen. So dienen sie in Smartphones, um die Ausrichtung zu erkennen. Und auch in tragbare Geräte wie der Nike Kraftstoff Band oder fit Bit, die IMU Sensoren verwendet, um Bewegung zu verfolgen.

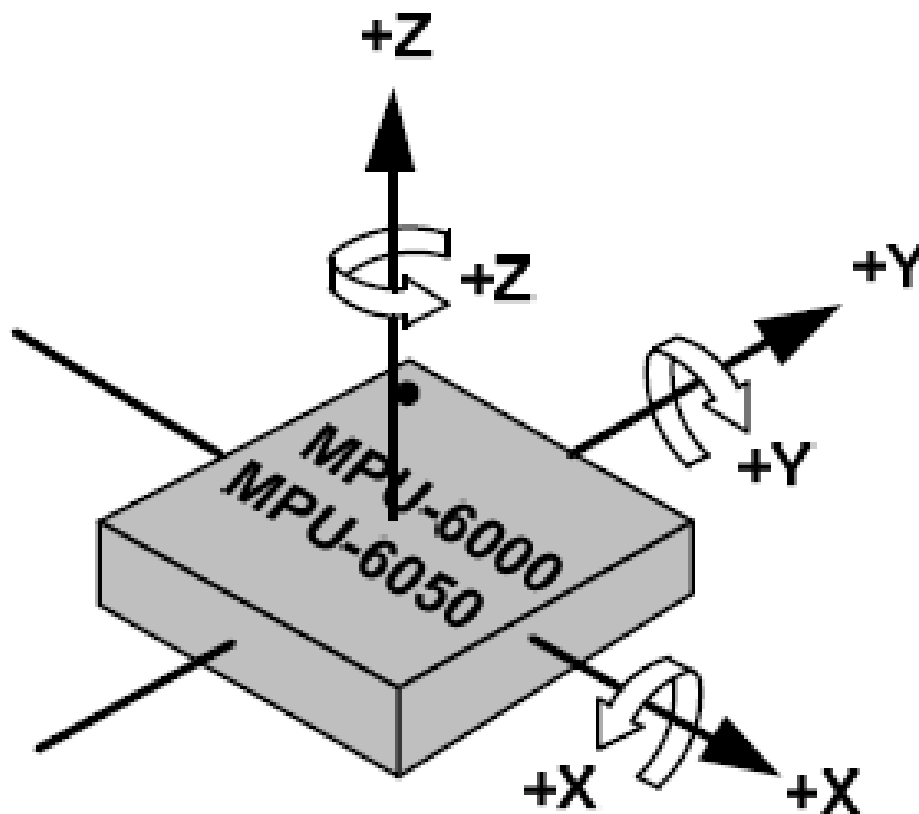
Wie funktioniert es?

IMU-Sensoren in der Regel besteht aus zwei oder mehr teilen. Sie diese nach Priorität auflisten, sie sind: Beschleunigungssensor, Gyroskop, Magnetometer und Höhenmesser. Die MPU-6050 ist eine 6-DOF (Freiheitsgrade) oder sechs Achsen IMU Sensor, was bedeutet, dass es sechs Werte als Ausgang gibt. Drei Werte aus der Beschleunigungsmesser und drei aus der Kreisel. Die MPU-6050 ist ein Sensor basierend auf MEMS (Micro Electro Mechanical Systems) -Technologie. Der Beschleunigungssensor und das Gyroskop sind eingebettet in einem einzigen Chip. Dieser Chip nutzt I2C (Inter Integrated Circuit) Protokoll für die Kommunikation.

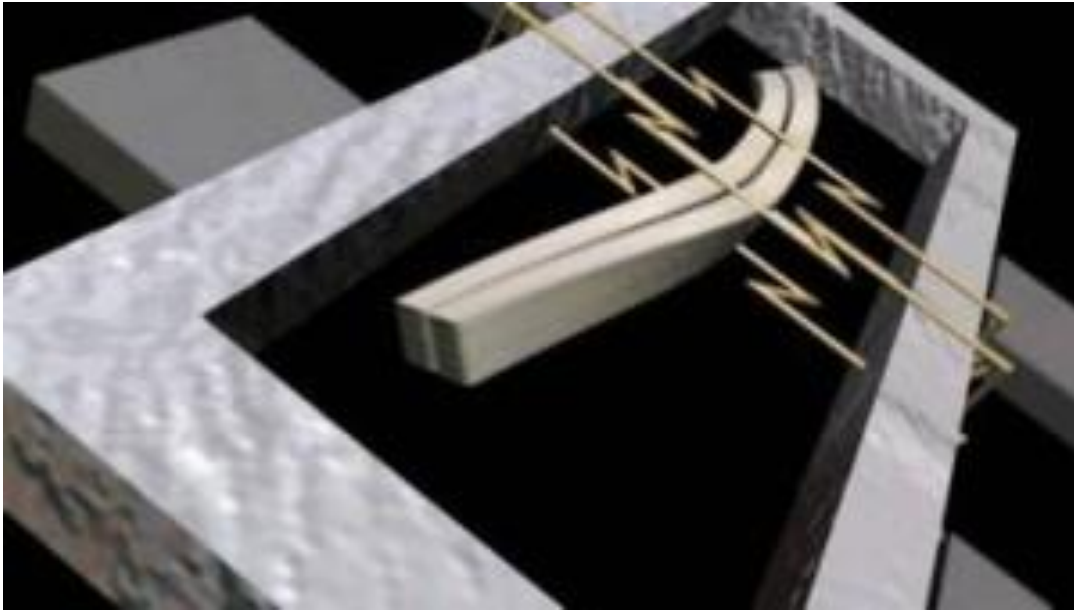
Wie funktioniert ein Beschleunigungssensor?



Ein Beschleunigungssensor arbeitet nach dem Prinzip des Piezoelektrischen Effekt. Hier stellen Sie sich einen quaderförmigen Kasten mit einem kleinen Ball drin, wie im Bild oben. Die Wände dieser Box sind mit Piezoelektrischen Kristallen gefertigt. Wenn Sie das Kontrollkästchen kippen, ist der Ball gezwungen, in die Richtung der Neigung, aufgrund der Schwerkraft zu bewegen. Die Wand mit denen der Ball kollidiert, schafft kleine Piezoelektrische Ströme. Insgesamt, gibt es drei Paare von gegenüberliegenden Wänden in einen Quader. Jedes Paar entspricht einer Achse im 3D-Raum: X, Y und Z-Achsen. Je nach Strömung, produziert von den Piezo-elektrischen-Wänden können wir die Richtung der Neigung und seine Größe bestimmen. Überprüfen Sie für weitere Informationen dies.



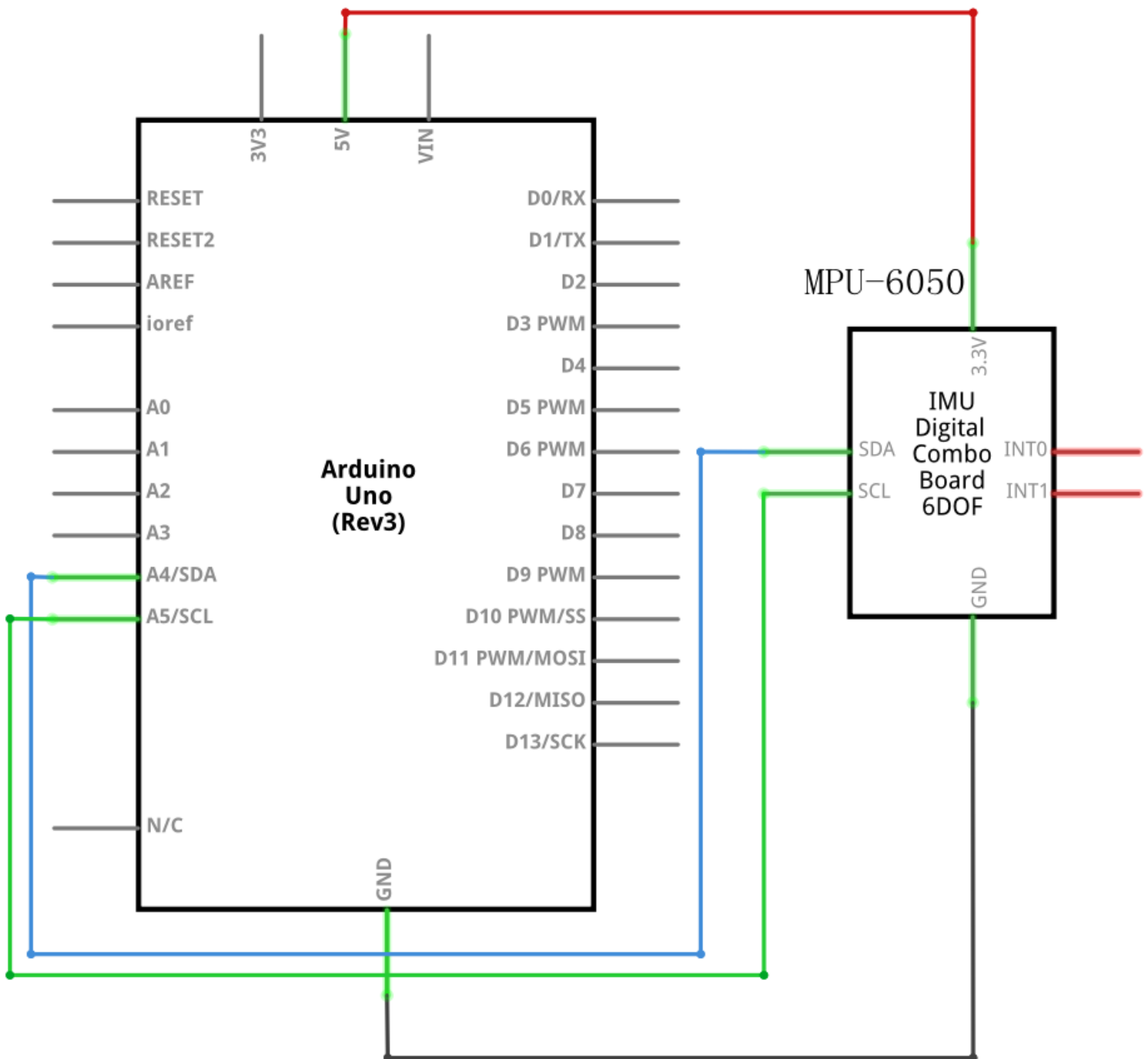
Wie funktioniert ein Gyroskop?



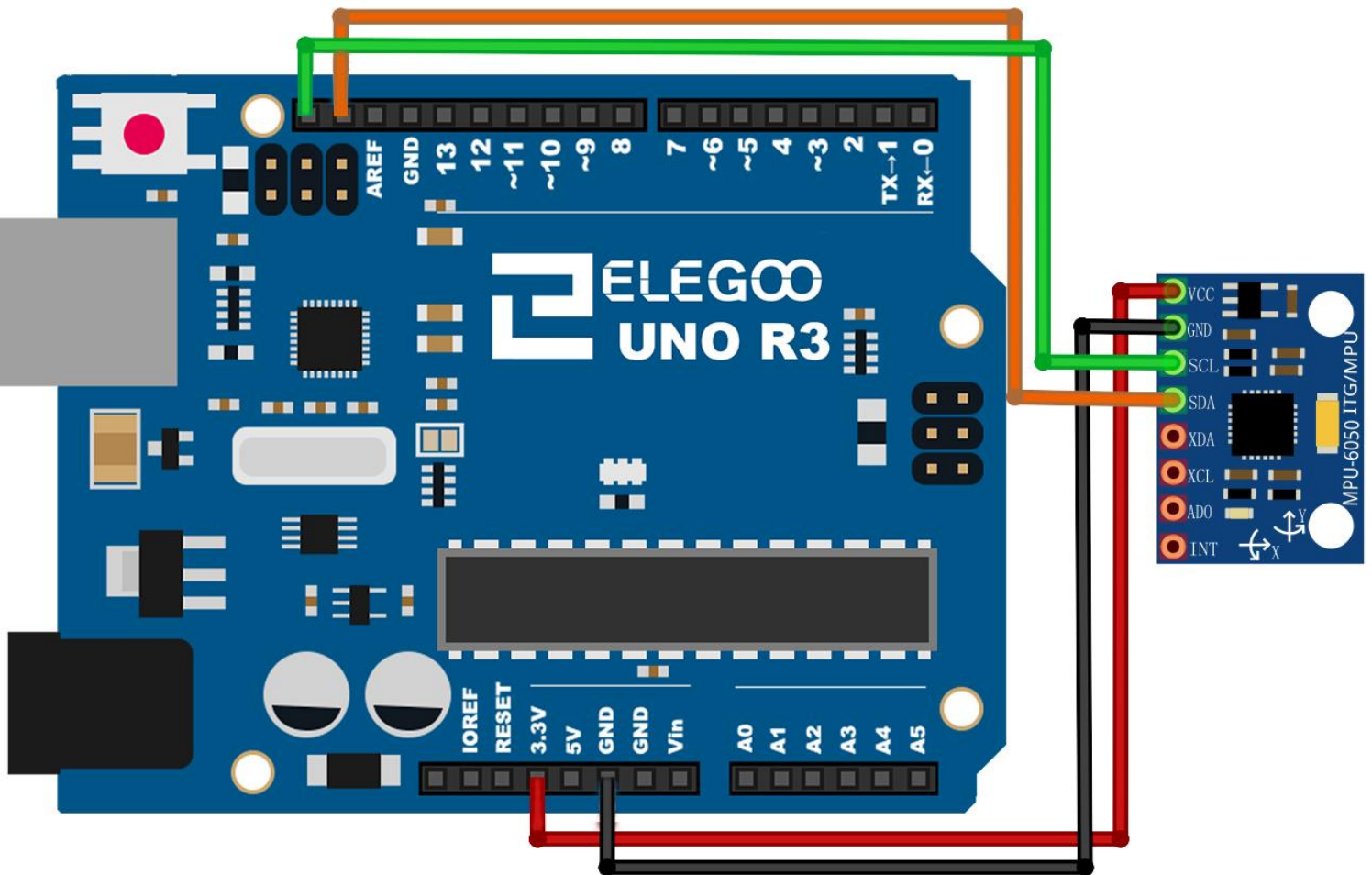
Gyroskope arbeiten nach dem Prinzip der Coriolis Beschleunigung. Stellen Sie vor, dass es eine Gabel wie Struktur, die in ständigen hin und her Bewegung ist. Es wird mit elektrischen Piezo-Kristalle gehalten. Wenn Sie versuchen, diese Vereinbarung zu kippen, erleben Sie die Kristalle eine Kraft in Richtung der Neigung. Dies wird verursacht durch die Trägheit der beweglichen Gabel. Die Kristalle erzeugen eine Strömung im Konsens mit dem Piezoelektrischen Effekt, und dieser Strom wird verstärkt. Die Werte werden dann vom Host Mikrocontroller veredelt.

Verbindung

Schema



Schaltplan



Die MPU-6050 kommuniziert mit dem Arduino durch das I2C Protokoll. Arduino ist die MPU 6050 verbunden, wie im folgenden Diagramm dargestellt. Hier, wenn Ihr MPU 6050 Modul 5V Pin hat, können dann Sie es auf Ihrem Arduino 5V Pin verbinden. Andernfalls musst du es herstellen die 3,3V Pin. Als nächstes ist der GND des Arduino mit dem GND der MPU 6050 verbunden.

Das Programm, das wir hier läuft, nutzt auch dem Arduino-Interrupt-Pin. Daher verbinden Sie Arduino der digitale Stift 2 (Interrupt-Pin 0) mit dem Stift als INT auf die MPU-6050 gekennzeichnet. Als nächstes müssen wir die I2C-Zeilen einrichten. Dafür schließen Sie den Stift als SDA auf die MPU-6050 auf dem Arduino analog Pin 4 (SDA) gekennzeichnet. Und den Pin als SCL auf der MPU-6050 mit dem Arduino analog Pin 5 (SCL) gekennzeichnet. Nun haben Sie Verkabelung der Arduino MPU 6050 fertig.

Software-Bibliotheken, die erforderlich ist

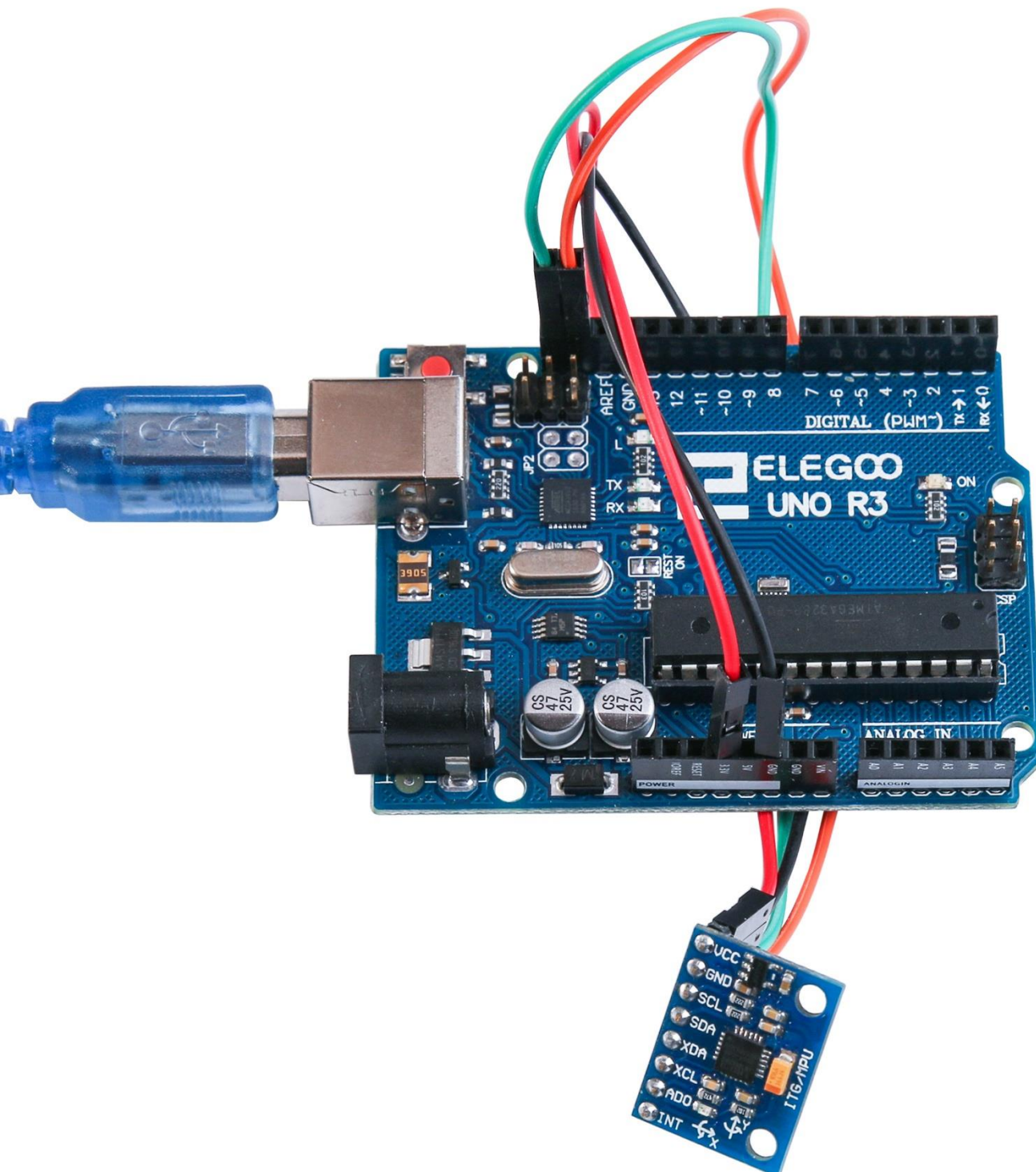
MPU6050

Der Code

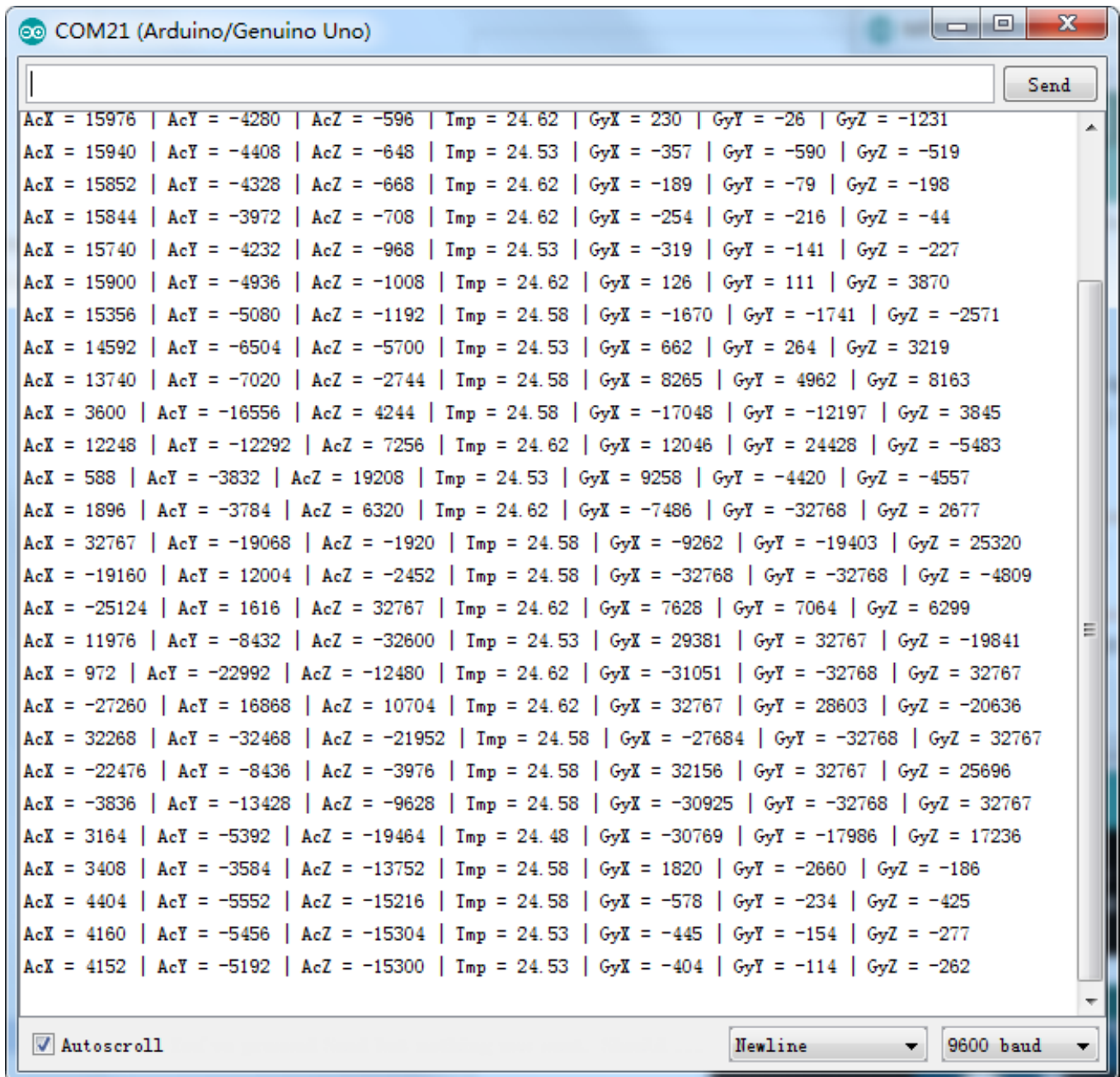
Kurzes Beispiel-Skizze ist eine sehr kurze Skizze und es zeigt alle Roh Werte (Beschleunigungsmesser, Gyro und Temperatur). Es sollte funktionieren auf Arduino "Uno", "Nano", "Leonardo" und auch "Due".

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefodert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek <MPU6050> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.



Schalten Sie den Monitor ein, so können Sie die Daten wie folgt sehen:



Lektion 17: HC-SR501 PIR Sensor

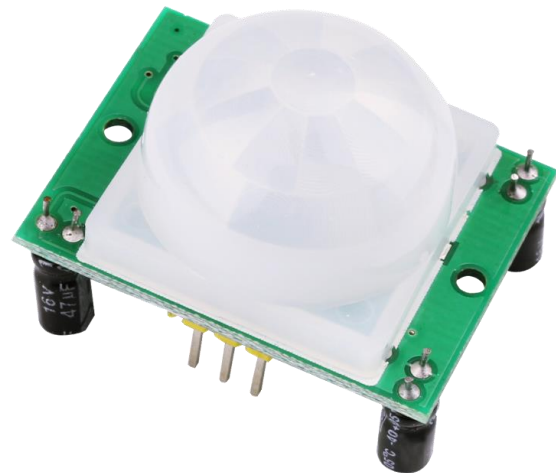
Übersicht

In dieser Lektion werden Sie erfahren, wie Sie einen PIR-Bewegungsmelder mit einem UNO verwenden.

Das UNO ist das Herz des Programms. Es ‚hört‘ den PIR-Sensor, und wenn eine Bewegung erkannt ist, wird die LED angewiesen, ein- oder auszuschalten.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) HC-SR501
- (3) F-M Kabel



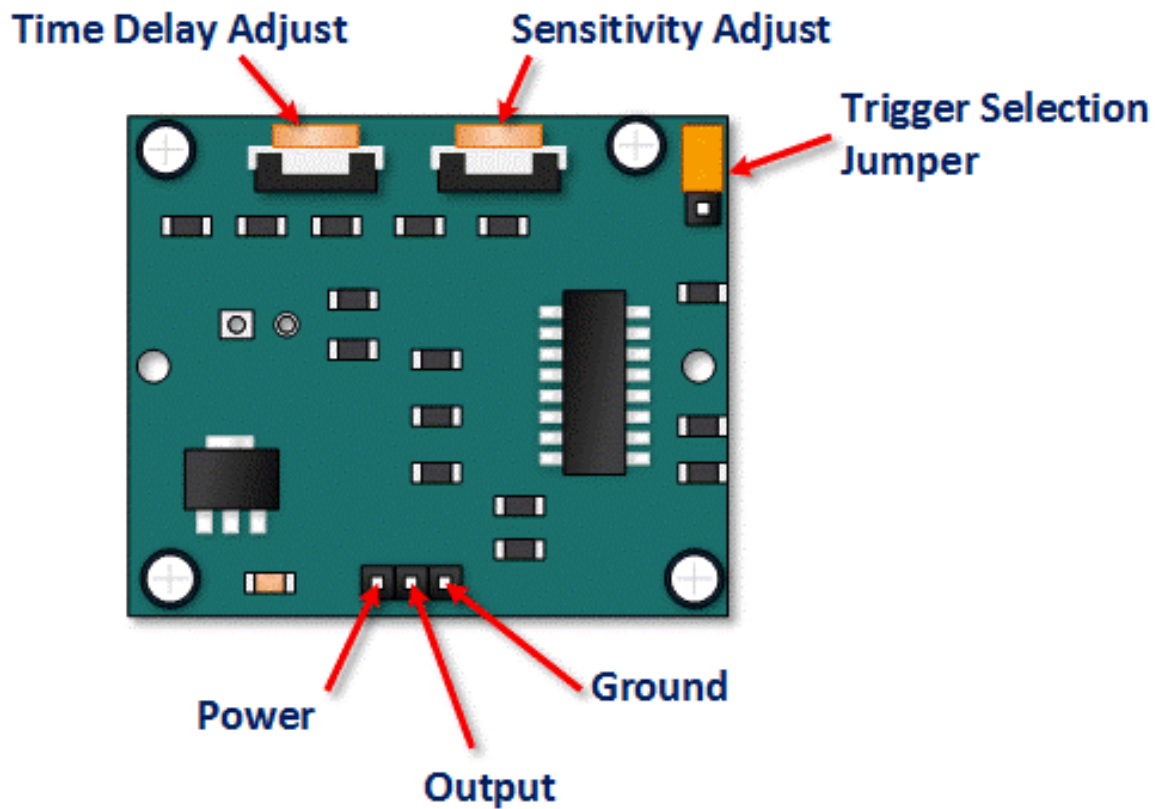
Komponentenanweisung

PIR-SENSOR:

PIR-Sensoren sind komplizierter als viele der anderen Sensoren, die in diesem Tutorial erklärt werden (wie Fotozellen, FSRs und Neigungsschalter), weil es mehrere Variablen, die Sensoreingang und -ausgang beeinflussen, gibt.

Der PIR-Sensor selbst verfügt über zwei Steckplätze. Jeder Steckplatz besteht aus einem speziellen Material, das IR-empfindlich ist. Die hier verwendete Linse tut nicht so viel, so sehen wir, dass die beiden Steckplätze über bestimmten Abstand hinaus ‚sehen‘ (grundsätzlich die Empfindlichkeit des Sensors). Wenn der Sensor sich im Leerlauf befindet, erkennen die beiden Steckplätze dieselbe Menge an IR, und die Umgebungsmenge wird aus dem Raum oder den Wänden oder dem Außenbereich abgestrahlt.

Wenn ein warmer Körper wie ein Mensch oder ein Tier vorbeigeht, fängt er zunächst eine Hälfte des PIR-Sensors, der eine positive Differenzänderung zwischen den beiden Hälften verursacht, ab. Wenn der warme Körper den Erkennungsbereich verlässt, geschieht das Umgekehrte, wobei der Sensor eine negative Differenzänderung erzeugt. Diese Änderungsimpulse sind, was erkannt wird.



Pin or Control	Function
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

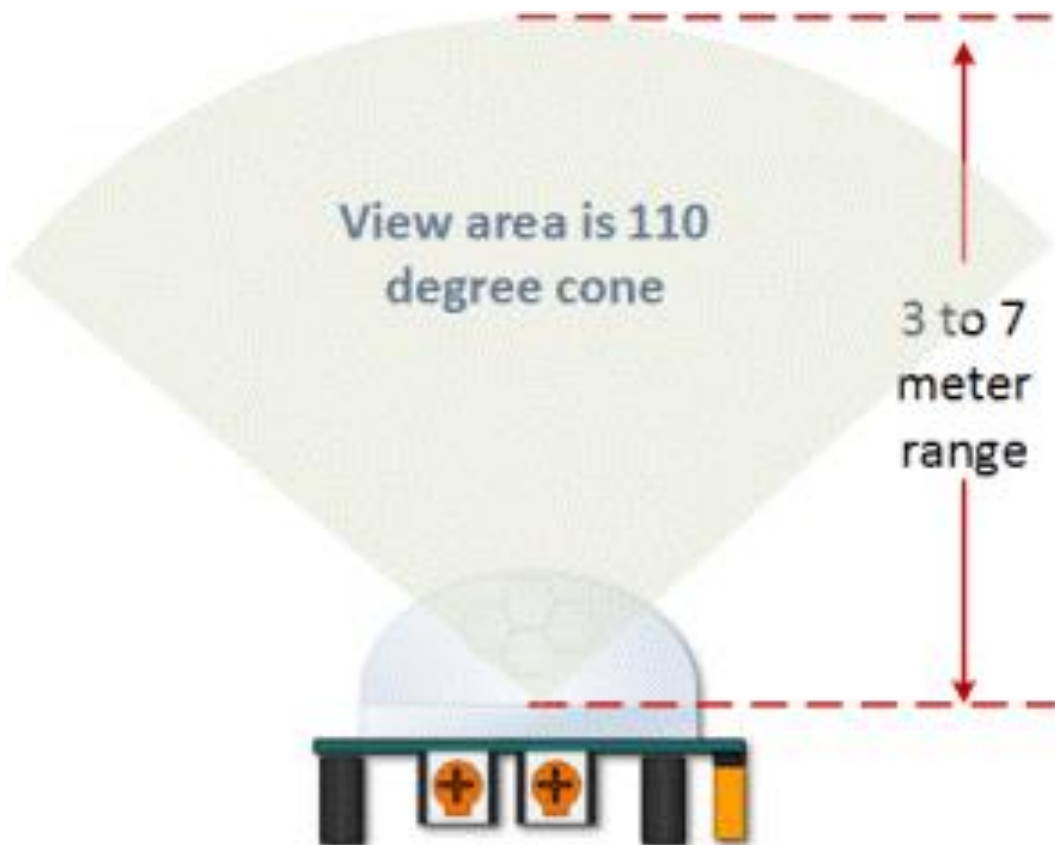
Die SR501 wird Infrarot-Änderungen erkennen und als Bewegung interpretiert seine Ausgabe niedrig eingestellt werden. Was ist oder wird nicht interpretiert, wie Bewegung weitgehend abhängig von Benutzer-Einstellungen und Anpassungen ist.

Gerät-Initialisierung

Das Gerät benötigt fast eine Minute zu initialisieren. Während dieser Zeit kann und oft Fehldetektionen Signale ausgegeben werden. Schaltung oder Controller-Logik muss dieser Initialisierung Zeit berücksichtigt werden.

Gerätebereich der Erkennung

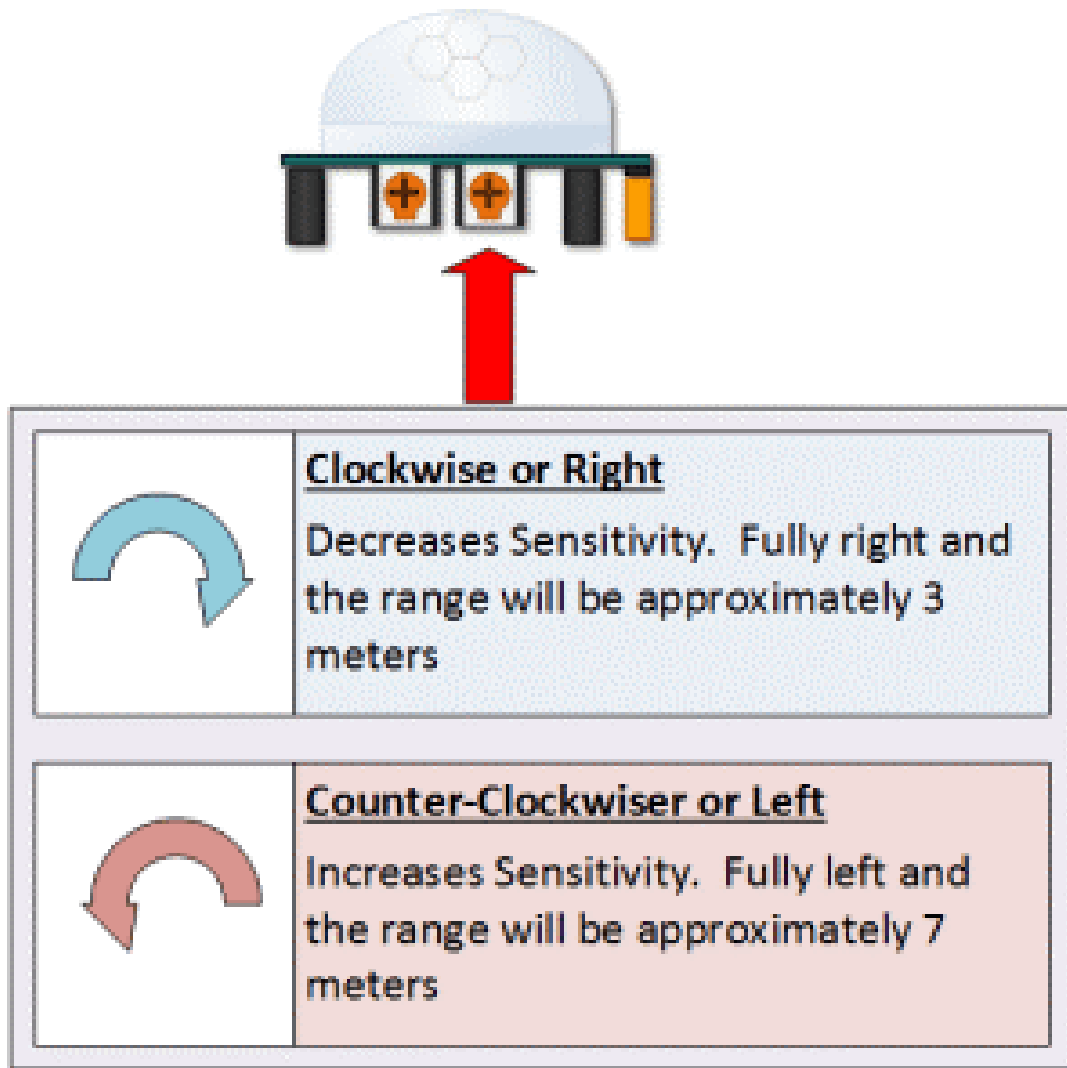
Das Gerät erkennt Bewegung innen 110-Grad-Kegel mit einer Reichweite von 3 bis 7 Metern.



HC-SR501-View-Bereich

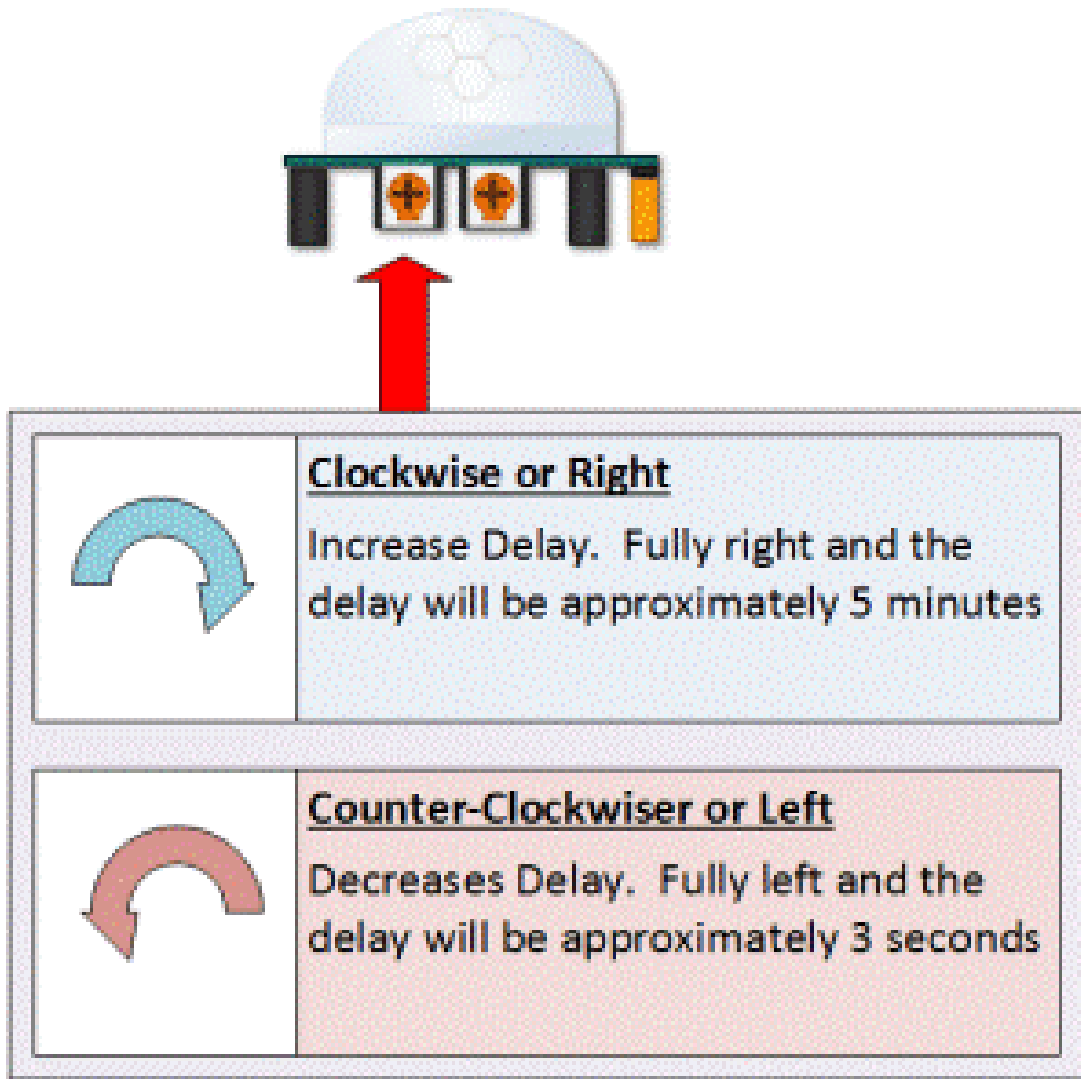
Einstellung des PIR-Bereichs (Empfindlichkeit)

Wie bereits erwähnt, ist der Verstell Bereich von ca. 3 bis 7 Meter. Die folgende Abbildung zeigt diese Einstellung.



HC SR501 Empfindlichkeitseinstellung-Einstellung der Verzögerung

Die Anpassung der Verzögerung bestimmt, wie lange die Ausgabe der PIR-Sensor-Modul nach Erkennung Bewegung hoch bleiben wird. Der Bereich liegt zwischen ca. 3 Sekunden bis fünf Minuten.



HC SR501 Einstellung der Verzögerung

3 Sekunden aus, nach der zeitlichen Verzögerung abgeschlossen ist - WICHTIG

Die Ausgabe dieses Geräts wird NIEDRIG gehen (oder aus) für ca. 3 Sekunden, NACHDEM die zeitliche Verzögerung abgeschlossen ist. Das heißt, wird ALLE Bewegungserkennung während diesem Zeitraum von drei Sekunden blockiert.

Zum Beispiel:

Stellen Sie sich vor, bist du in der einzigen Trigger Modus und Ihre zeitliche Verzögerung wird 5 Sekunden festgelegt.

Der PIR-SENSOR erkennt Bewegungen und für 5 Sekunden hoch angesetzt.

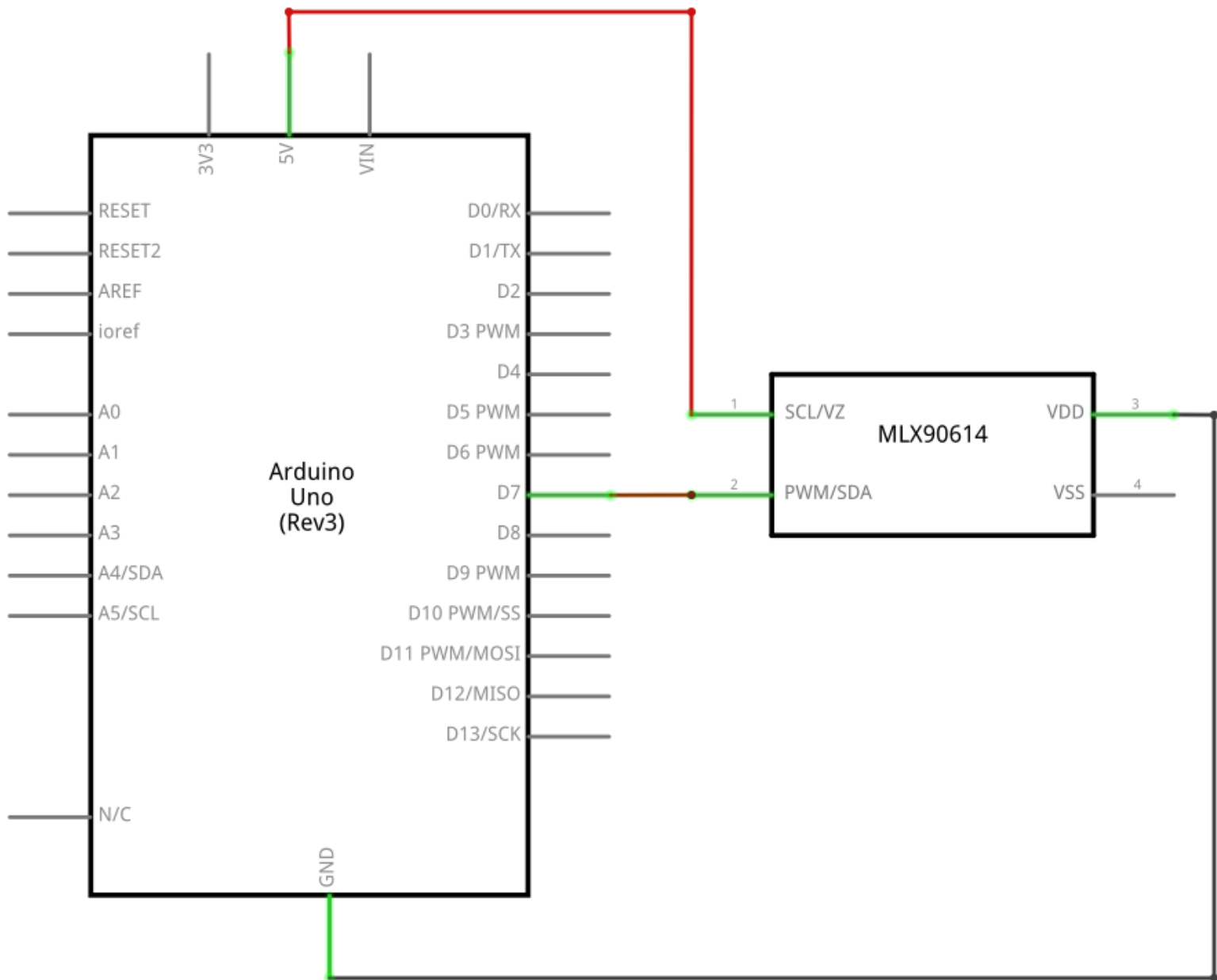
Nach fünf Sekunden wird der PIR-SENSOR setzt seinen Ausgang niedrig für ca. 3 Sekunden.

Während der drei Sekunden wird der PIR-SENSOR nicht Bewegung zu erkennen.

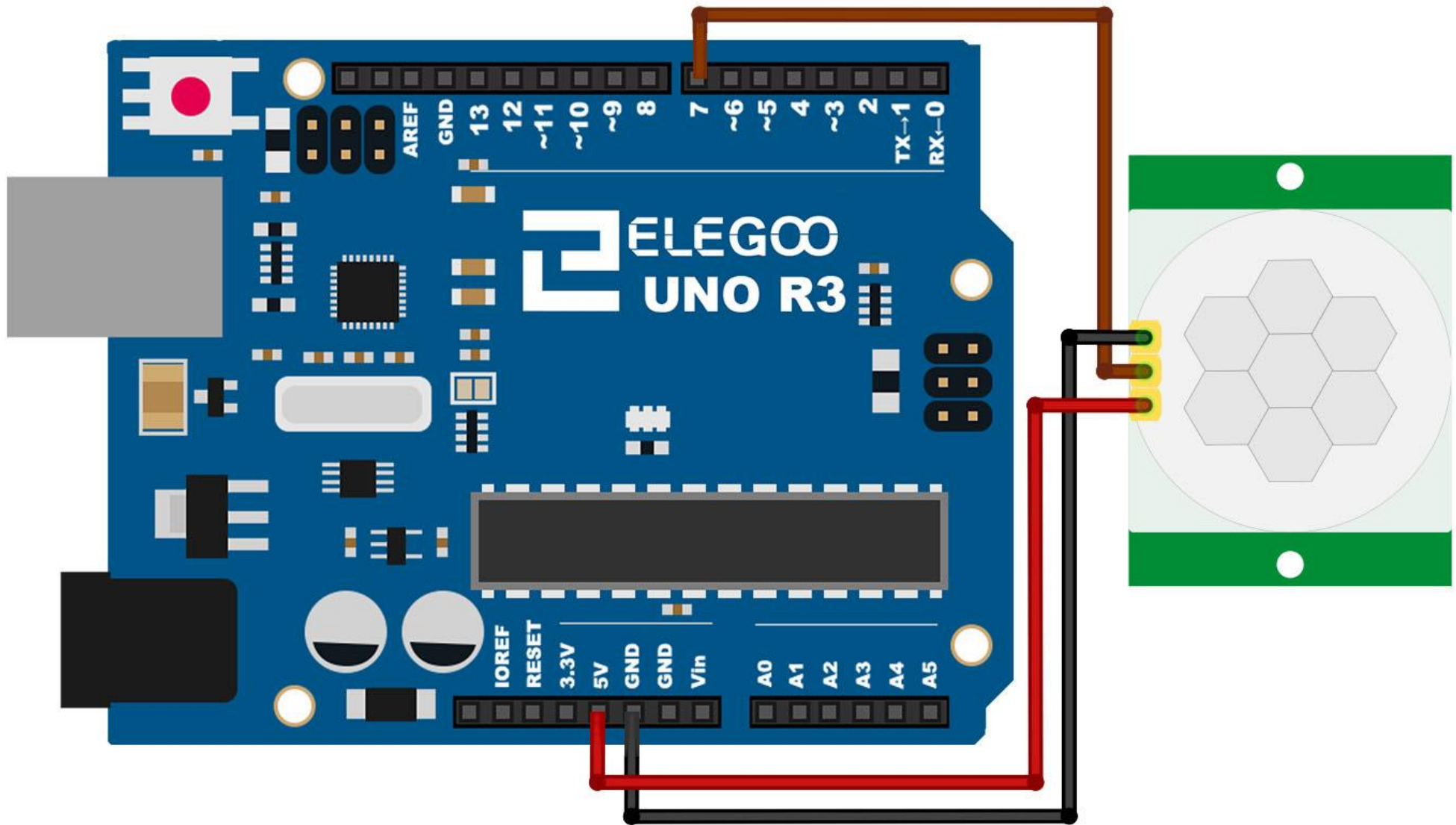
Nach drei Sekunden wird der PIR-SENSOR erkennt Bewegungen wieder und erkannten Bewegung erneut die hohe Leistung.

Verbindung

Schema



Schaltplan



Verbindung PIR-Sensoren mit einem Mikrocontroller ist wirklich einfach. Der PIR-SENSOR dient als ein digitaler Ausgang, so dass alle Sie tun müssen, ist für den Stift zu hochdrehen hören (erkannt) oder niedrig (nicht erkannt).

Es ist wahrscheinlich, dass Sie wollen Wiederauslösen, so sicher sein, setzen Sie den Jumper in der H-Position!

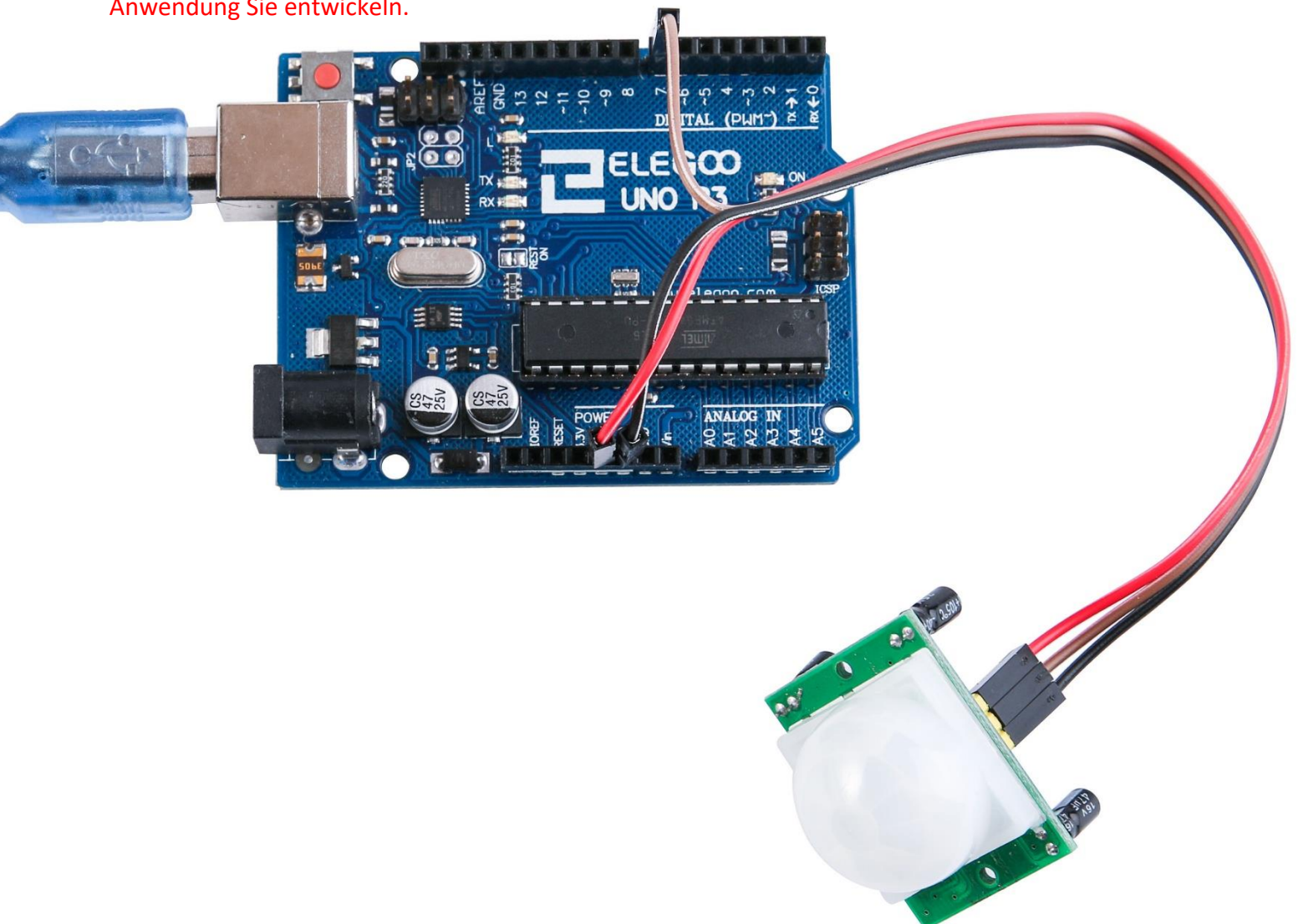
Der PIR-SENSOR mit 5V schalten und verbinden von Boden zu Boden. Dann verbinden Sie den Ausgang zu einem digitalen Stift. In diesem Beispiel verwenden wir Pin 7.

Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Die Skizze wird einfach auf Ihre Arduino LED an Pin 13 angeschlossen, sobald Bewegung erkannt wird.

Achten Sie darauf, Hüten Sie sich vor und irgendwie behandeln die 1 Minute Initialisierung in welcher Anwendung Sie entwickeln.



Lektion 18: Wasstand-Erkennung-Sensormodul

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie ein Wasserstand-Erkennung-Sensormodul verwenden. Dieses Modul kann die Wassertiefe erkennen, und die Kernkomponente ist eine Verstärkerschaltung, die aus einem Transistor und mehreren kammförmigen PCB-Routings besteht. Wenn sie in das Wasser gesetzt werden, werden diese Routings einen Widerstand, der sich mit der Änderung der Wassertiefe ändern kann, darstellt. Dann wird das Signal der Wassertiefe in das elektrische Signal umgewandelt, und wir können die Änderung der Wassertiefe mittels der ADC-Funktion von UNO R3 feststellen.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Wasserfüllstanderkennung-Sensormodul
- (3) F-M Kabel



Komponentenanweisung

Wassersensor:

Ein Wassersensorblock wird für Wassererkennung entwickelt, und er kann häufig in Erkennungen des Niederschlags, des Wasserstands, und sogar des Flüssigkeitsaustritts eingesetzt werden. Der Block besteht hauptsächlich aus drei Teilen: einem elektrischen Blockverbinder, einem 1 M Ω -Widerstand und einige Leitungen der blanken Leistungsdrähte.

Dieser Sensor arbeitet mit einer Reihe von freiliegenden Spuren, die mit der Masse verbunden werden. Die Erkennungsspuren befinden sich vernetzt zwischen den Massespuren.

Die Erkennungsspuren weisen einen schwachen Pull-up-Widerstand von 1M Ω auf. Der Widerstand wird den Wert der Sensorspur hoch gesetzt, bis dass ein Wassertropfen die Sensorspur auf die Massespur verkürzt. Ob Sie es glaubet oder nicht, die Schaltung wird mit den digitalen I/O-Pins von Ihrem UNO R3-Board arbeiten, oder Sie können ihn mit den analogen Pins zusammen verwenden, um die Menge an das Wasser, das Kontakt zwischen den geerdeten Spuren und den Sensorspuren verursacht hat, zu erkennen.

Der Wassersensor kann den Wasserstand bewerten, indem er mit einer Reihe von freiliegenden parallelen Kabeln den Wassertropfen / die Wassergröße misst. Er kann die Wassergröße einfach in ein analoges Signal umwandeln, und der Ausgangsanalogwert kann direkt in der Programmfunktion verwendet werden, dann wird die Funktion des Wasserstandalarms aktiviert.

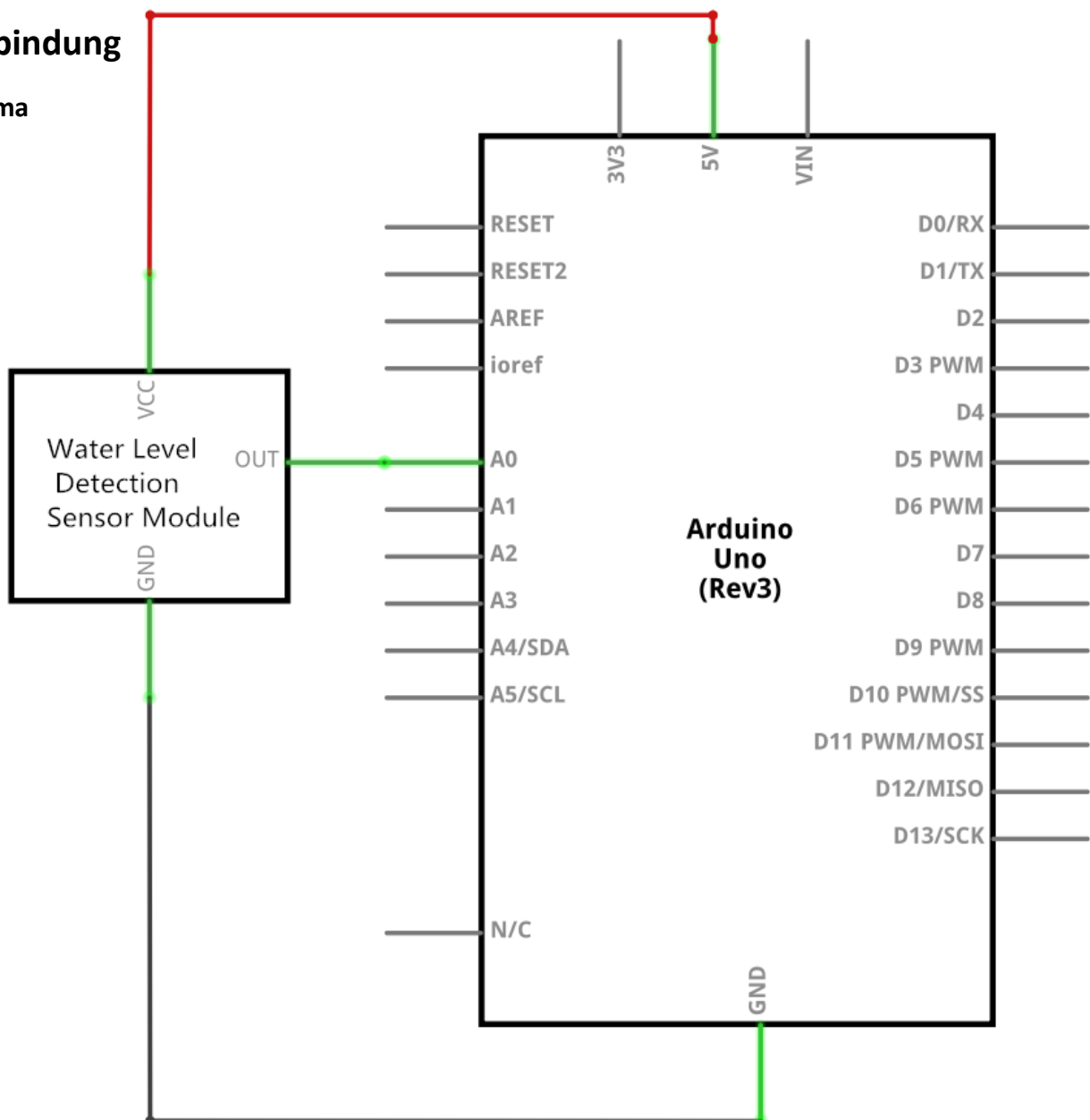
Er hat einen geringen Stromverbrauch und eine hohe Empfindlichkeit.

Eigenschaften:

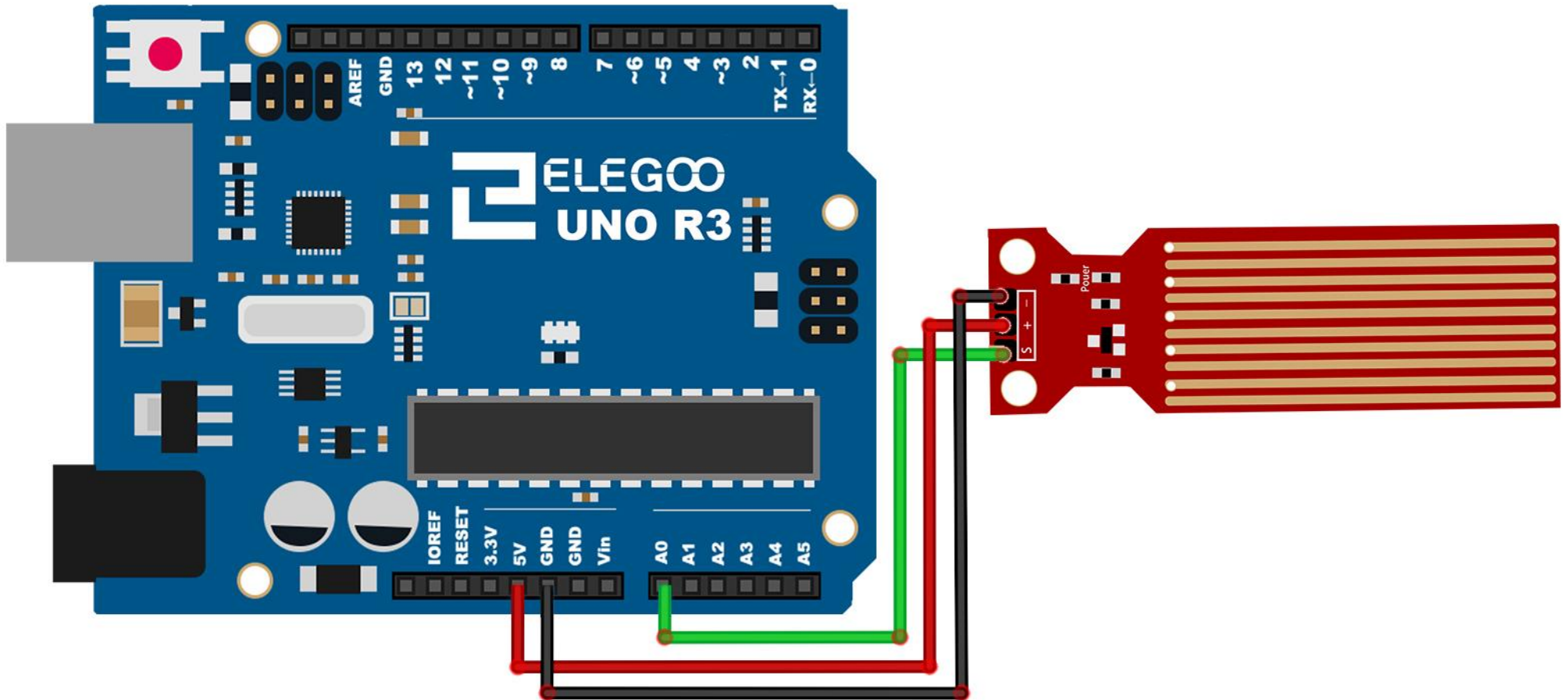
- 1. Arbeitsspannung : 5V
- 2. Arbeitsstrom : <20 mA
- 3. Schnittstelle: Analog
- 4. Erkennungsbreite: 40mm × 16mm
- 5. Arbeitstemperatur : 10 °C ~ 30 °C
- 6. Ausgangsspannungssignal : 0 ~ 4,2 V

Verbindung

Schema



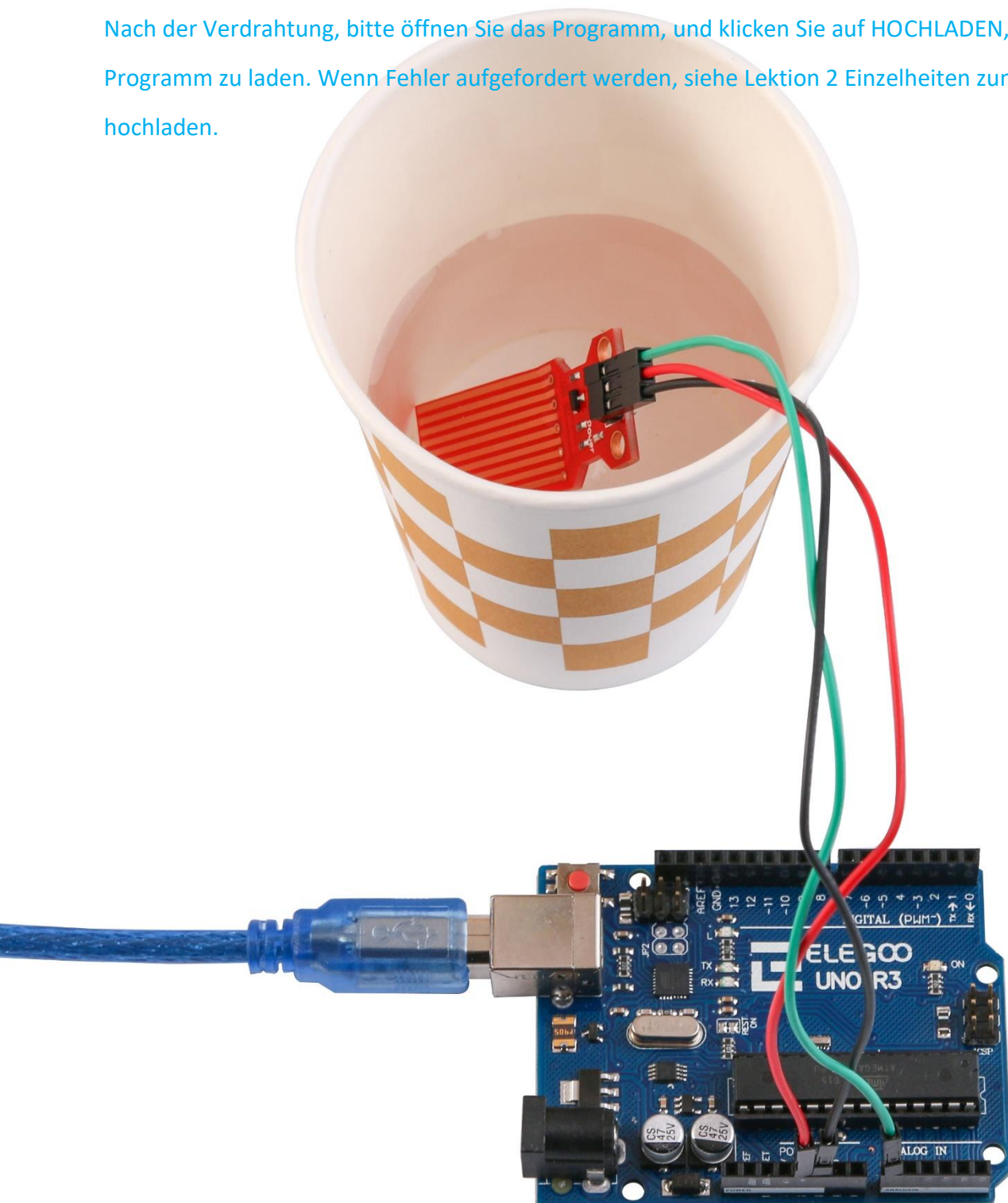
Schaltplan



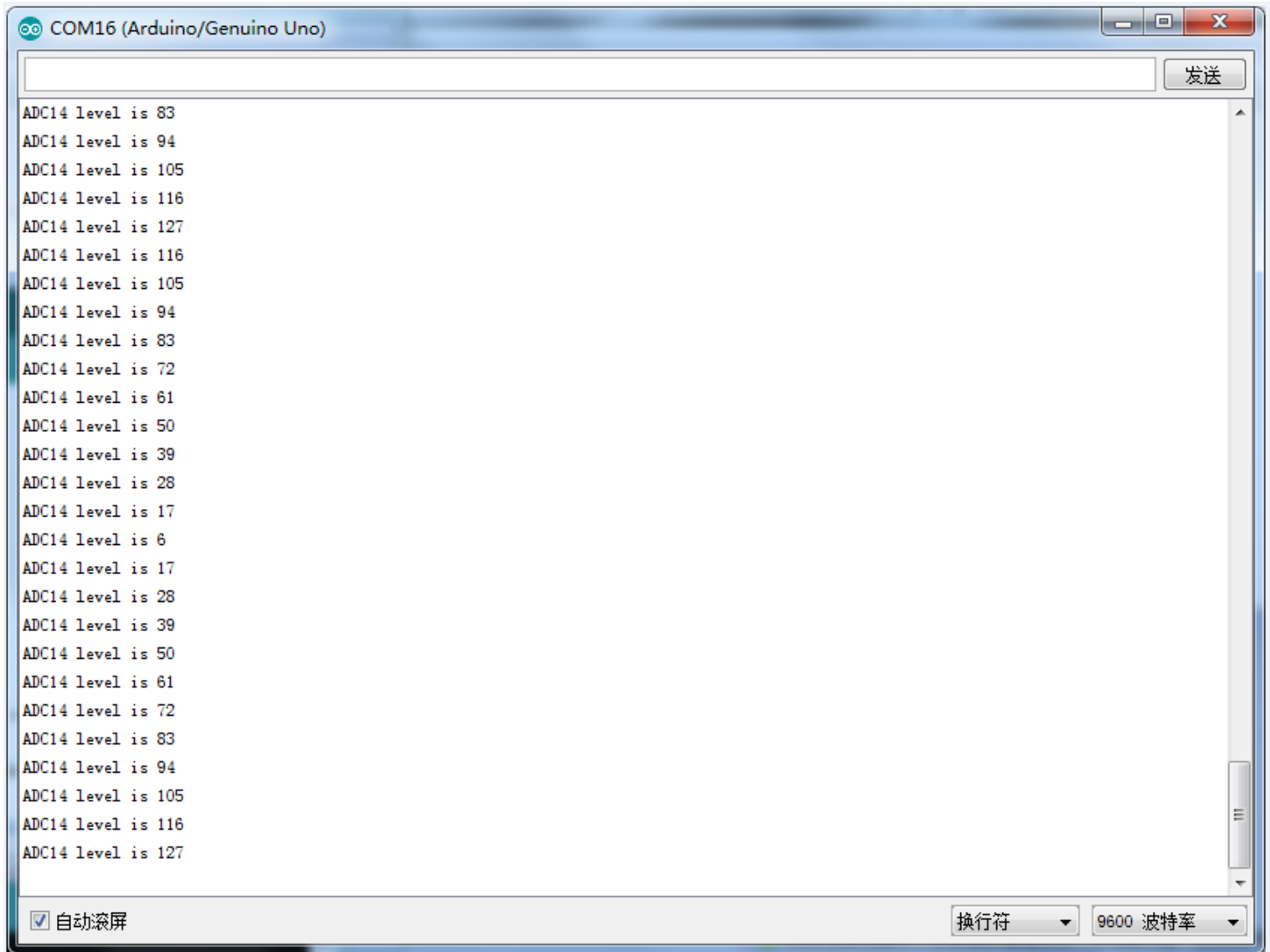
Hinweis über Verdrahtung: Spannungsversorgung (+) wird mit 5V-Pin des UNO R3-Boards verbunden, Masselektrode (-) wird mit GND verbunden. Signalausgang (S) wird mit den Anschlüssen (A0-A5), die die Funktion hat, irgendein analoges Signal in das UNO R3-Board einzugeben, verbunden, aber er sollte den gleichen Demo-Code wie das Programm definieren.

Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefodert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.



Schalten Sie den Monitor ein, so können Sie die Daten wie folgt sehen:



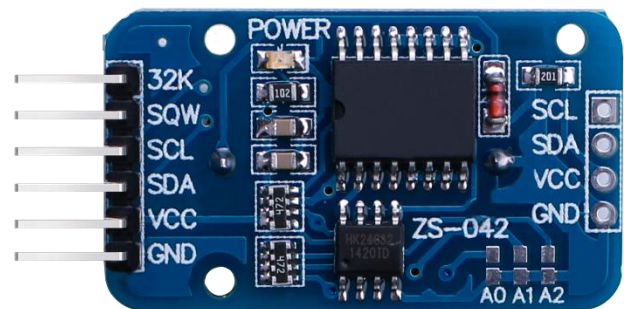
Lektion 19: Echtzeituhren-Modul

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie DS3231, Uhrenmodul, das das Jahr, den Monat, den Tag, die Stunde, die Minute, die Sekunde und die Woche anzeigt, verwenden. Das Uhrenmodul wird durch ein Sicherungsbatterie-Erhaltungsladegerät, das verwendet werden kann, nur wenn es mit nur drei Datenkabeln verbunden wird, unterstützt.

Eroforderliche Komponente

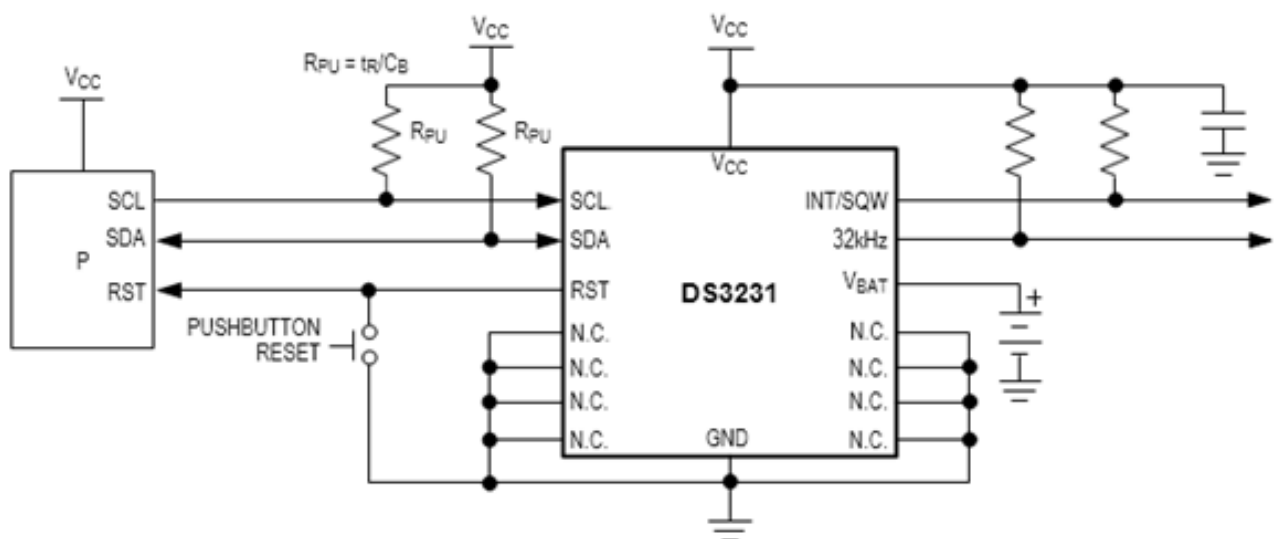
- (1) Elegoo UNO R3
- (1) DS3231 Echtzeituhren-Modul
- (4) F-M Kabel



Komponentenanweisung

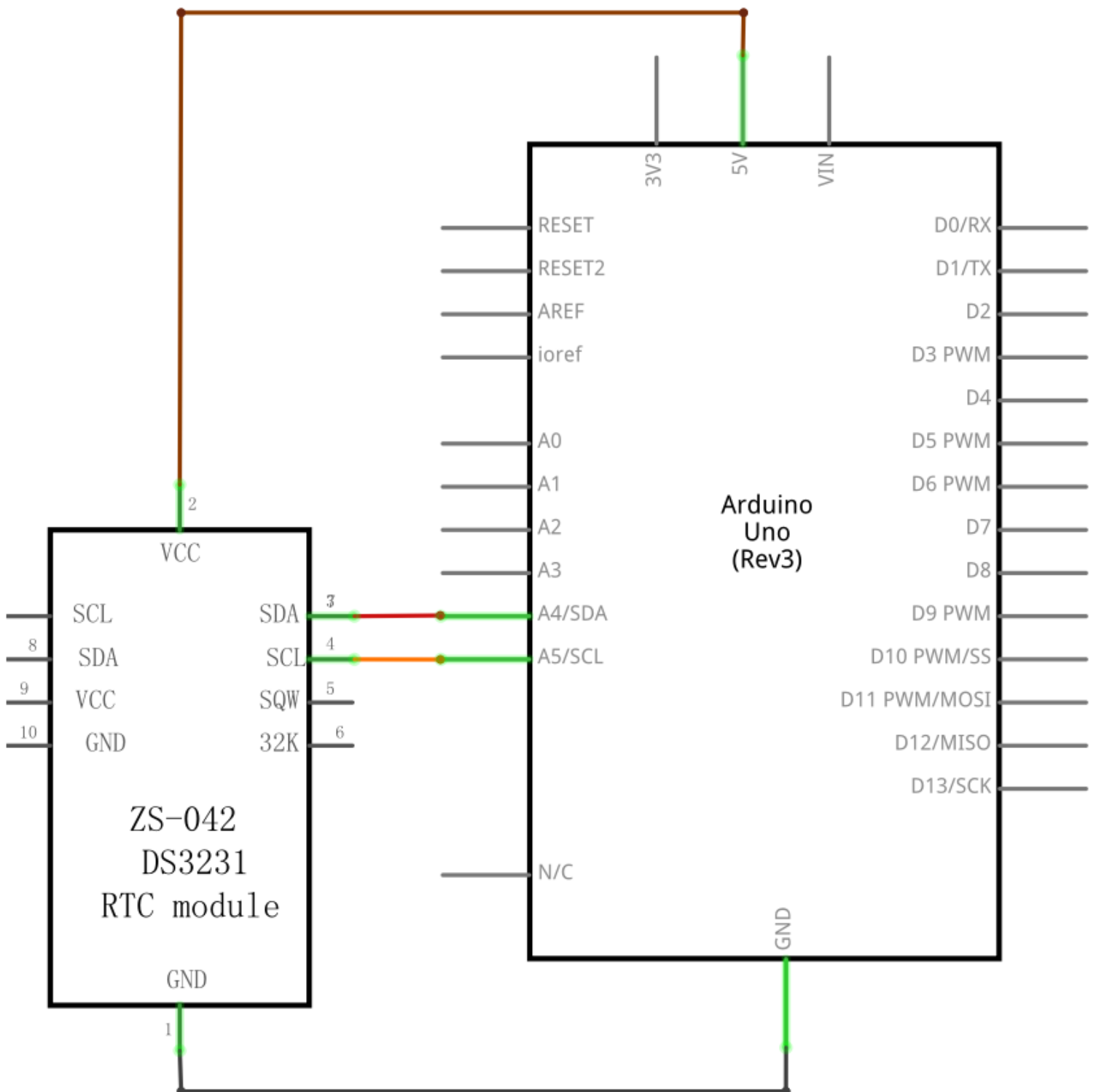
DS3231

DS3231 ist ein einfacher Zeitmessungschip. Es weist eine eingebaute Batterie auf, so kann die Uhr die Zeit weiterhin messen, auch wenn sie herausgezogen wird.

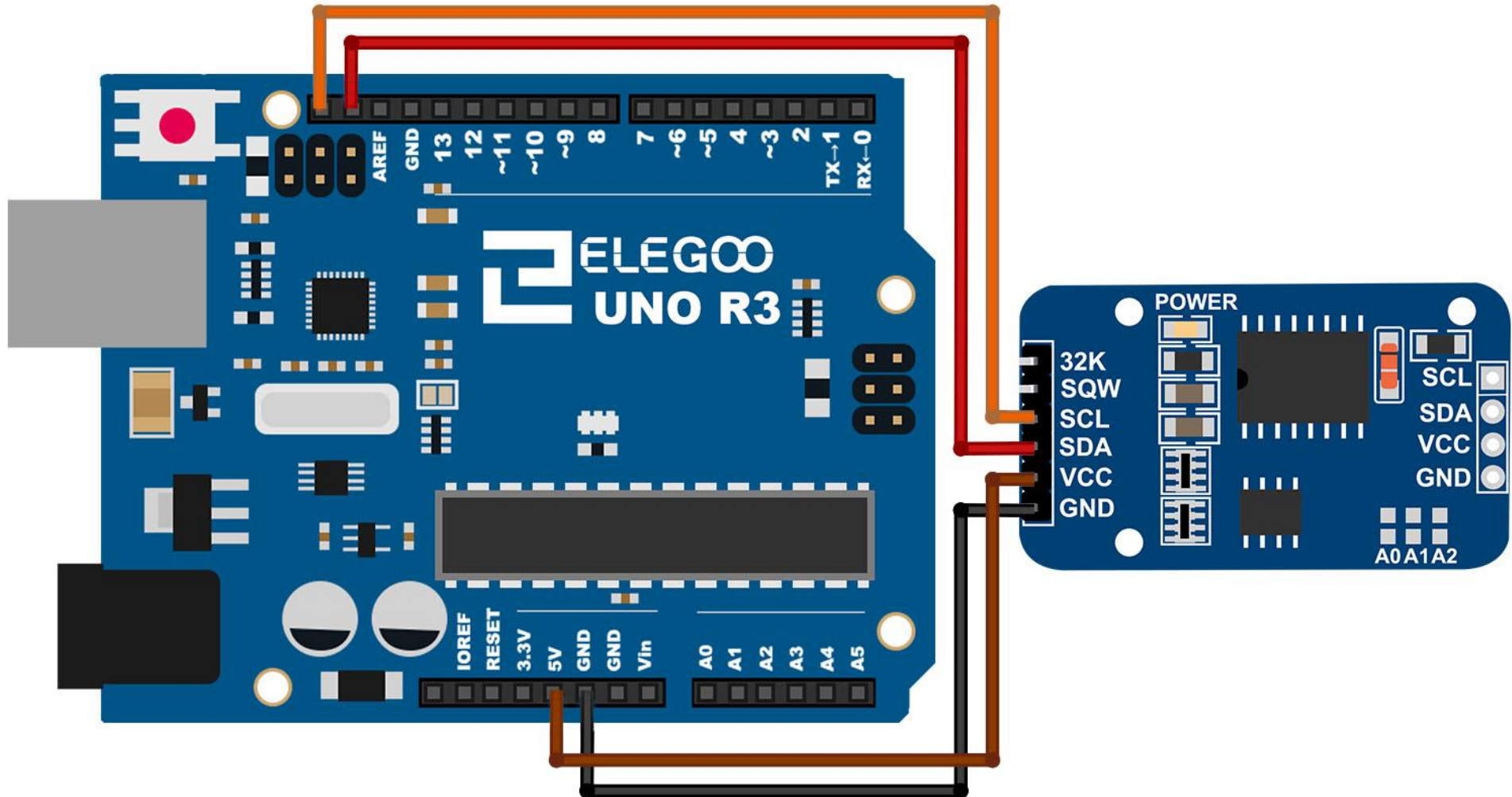


Verbindung

Schema



Schaltplan



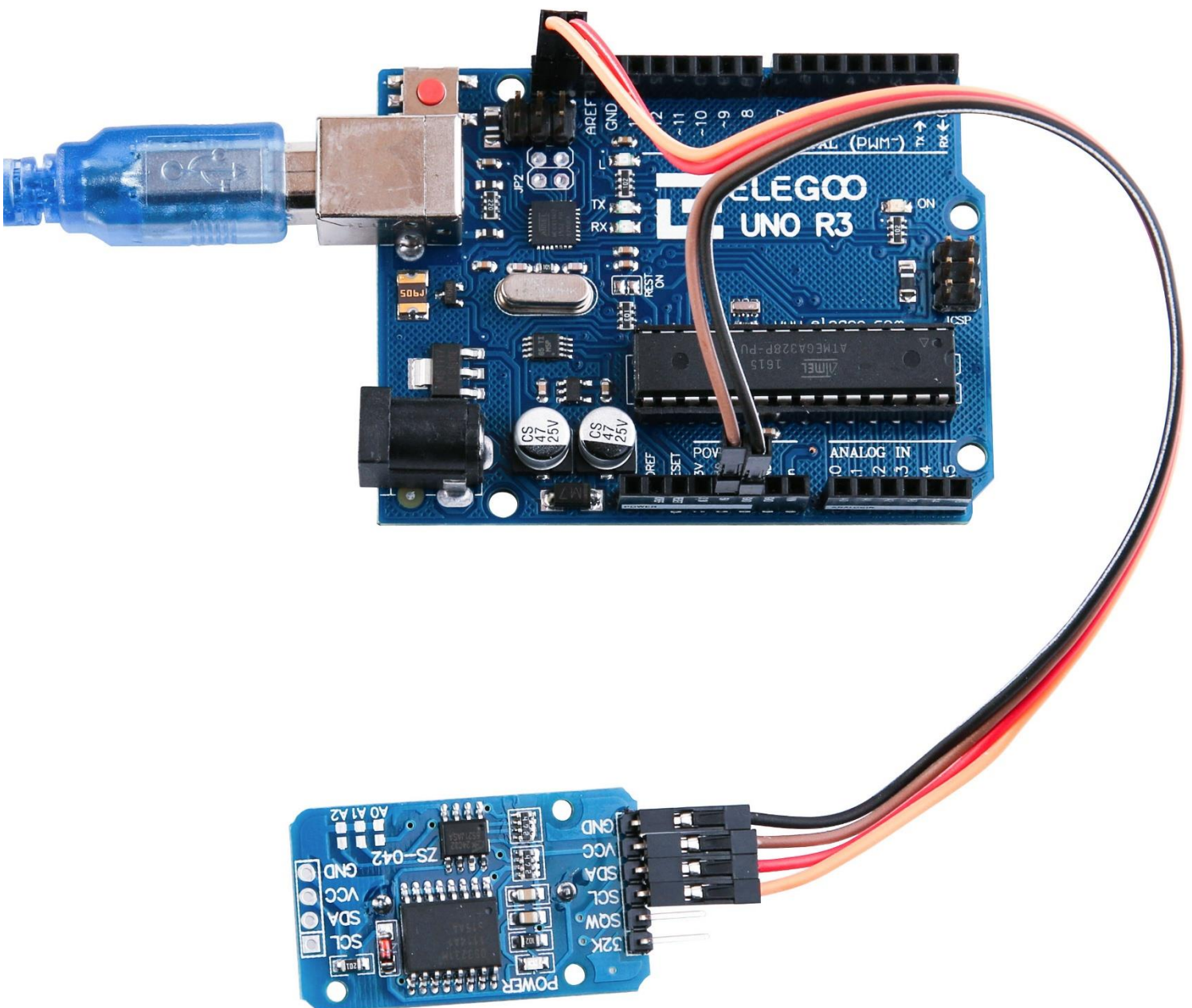
Richten Sie gemäß dem folgenden Bild ein.

Ignorieren Sie 31K- und SQW-Pins; sie sind unnötig. Stecken Sie den SCL-Pin in den A5-Anschluss Ihres UNO R3-Boards und den SDA-Pin in den A4-Anschluss ein. Stecken Sie den VCC-Pin in den 5V-Anschluss und den GND-Pin in den GND-Anschluss ein.

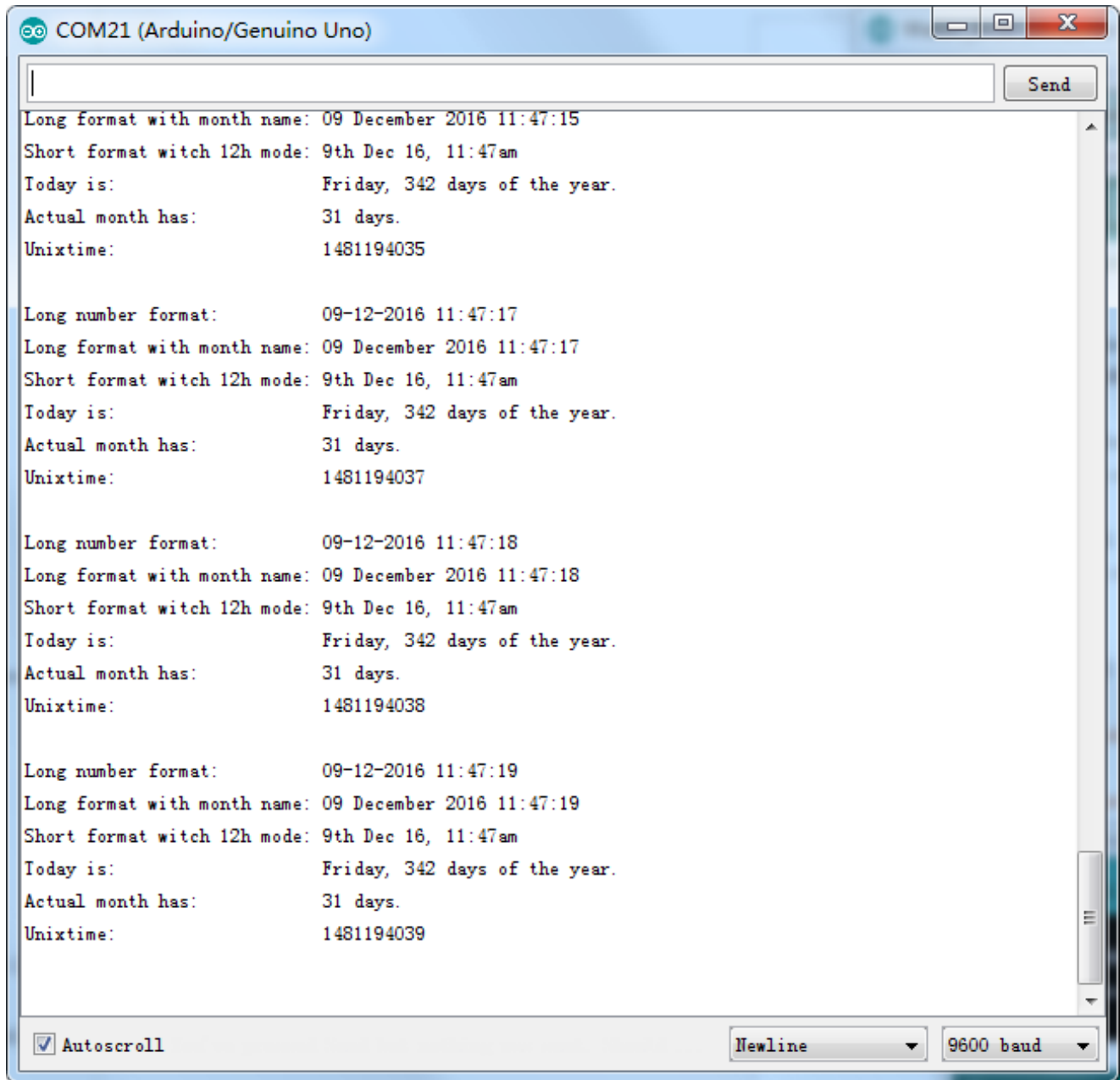
Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

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



Schalten Sie den Monitor ein, so können Sie sehen, dass das Modul die Zeit wie folgt anzeigt:



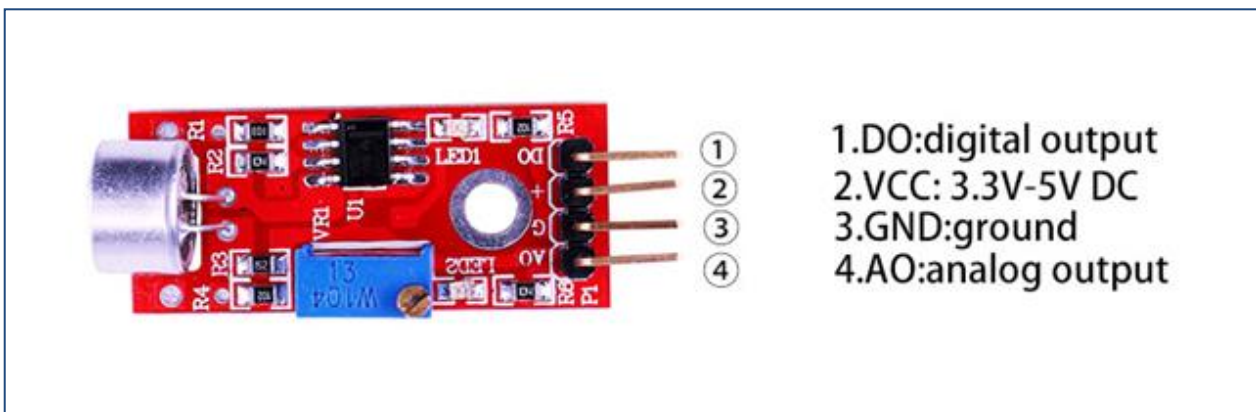
Lektion 20: Schallsensor-Modul

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie ein Schallsensor-Modul verwenden. Dieses Modul verfügt über zwei Ausgänge:

AO: Analogausgang, Echtheit-Ausgangsspannungssignal des Mikrofons

DO: Wenn die Intensität des Tons einen bestimmten Schwellenwert erreicht, ist der Ausgang ein HIGH- oder LOW-Pegel-Signal. Die Empfindlichkeit des Schwellenwerts kann durch Einstellen des Potentiometers erreicht werden.



Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Schallsensor-Modul
- (3) F-M Kabel

Komponentenanweisung

Mikrofon:

Wandler sind Vorrichtungen, die Energie von einer Form in eine andere Form umwandeln. Ein Mikrofon ist ein Wandler, der die Schallenergie in das elektrische Signal umwandelt. Er wirkt entgegengesetzt zu einem Lautsprecher. Mikrofone sind in verschiedenen Formen und Größen verfügbar. Je nach der Anwendung könnte ein Mikrofon verschiedene Technologien verwenden, um den Schall in elektrische Signale umzuwandeln. Hier werden wir über das Electret-Kondensatormikrofon, das häufig in Mobiltelefonen, Laptops, etc. verwendet werden, diskutieren.

Wie die Bezeichnung verrät, ist das Elektret-Kondensatormikrofon ein Parallelplattenkondensator, und arbeitet es nach dem Grundsatz einer variablen Kapazität. Es besteht aus zwei Platten, d.h. der feststehenden Platte (als Rückplatte genannt) und der anderen beweglichen Platte (als Diaphragma genannt), und ein kleiner Spalt steht zwischen den beiden Platten. Ein elektrisches Potential lädt die Platte auf. Wenn der Schall das Diaphragma trifft, beginnt es zu bewegen, wodurch die Kapazität zwischen den beiden Platten geändert wird, was wiederum zum Fließen eines variablen elektrischen Stroms führt.



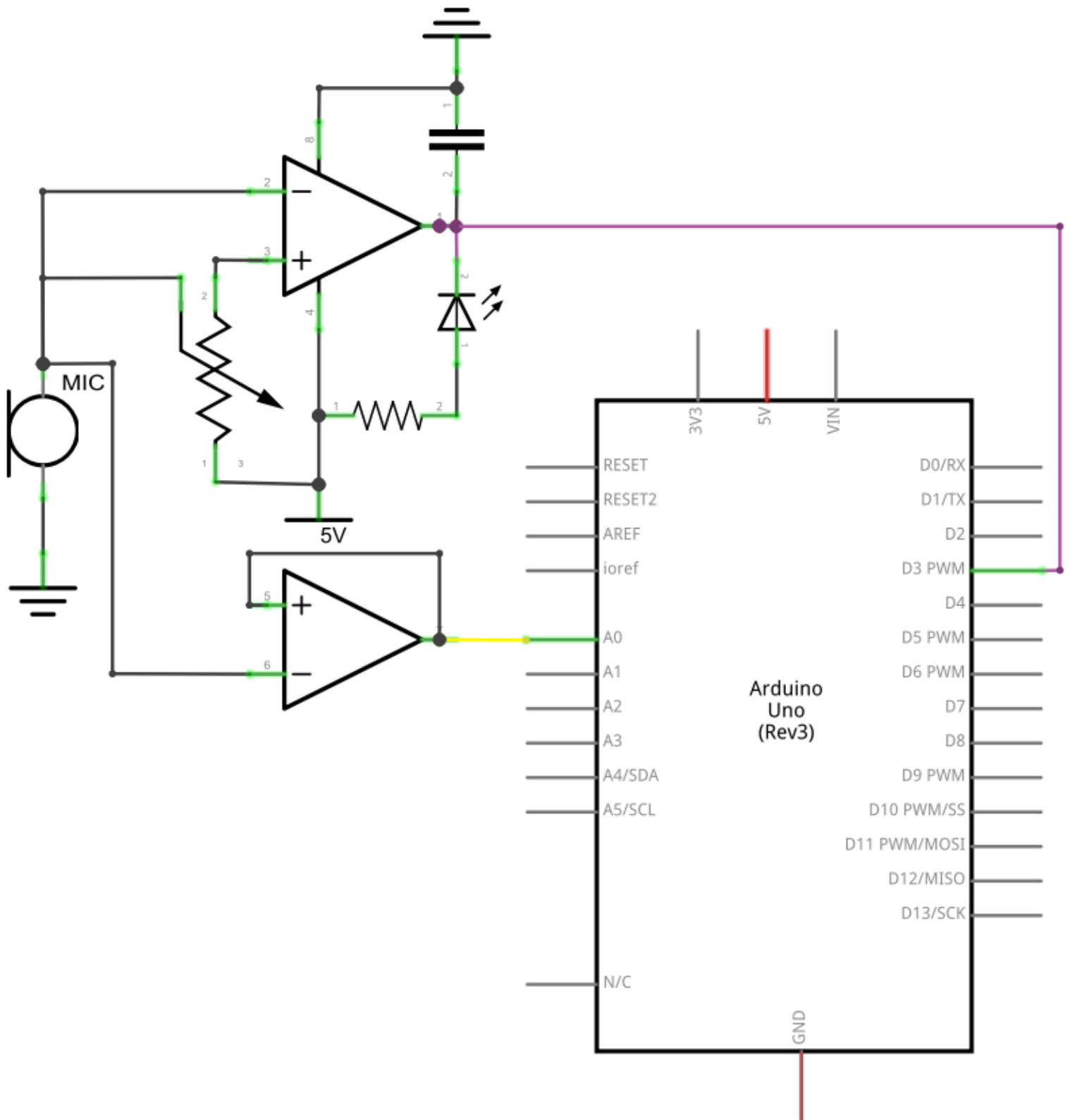
Diese Mikrofone werden häufig in elektronischen Schaltungen verwendet, um geringfügigen Schall oder Luftschwingungen, die wiederum in elektrische Signale für weitere Verwendung umgewandelt werden, zu erkennen. Die beiden Schenkel, wie im Bild oben dargestellt, werden verwendet, um eine elektrische Verbindung mit der Schaltung zu machen.



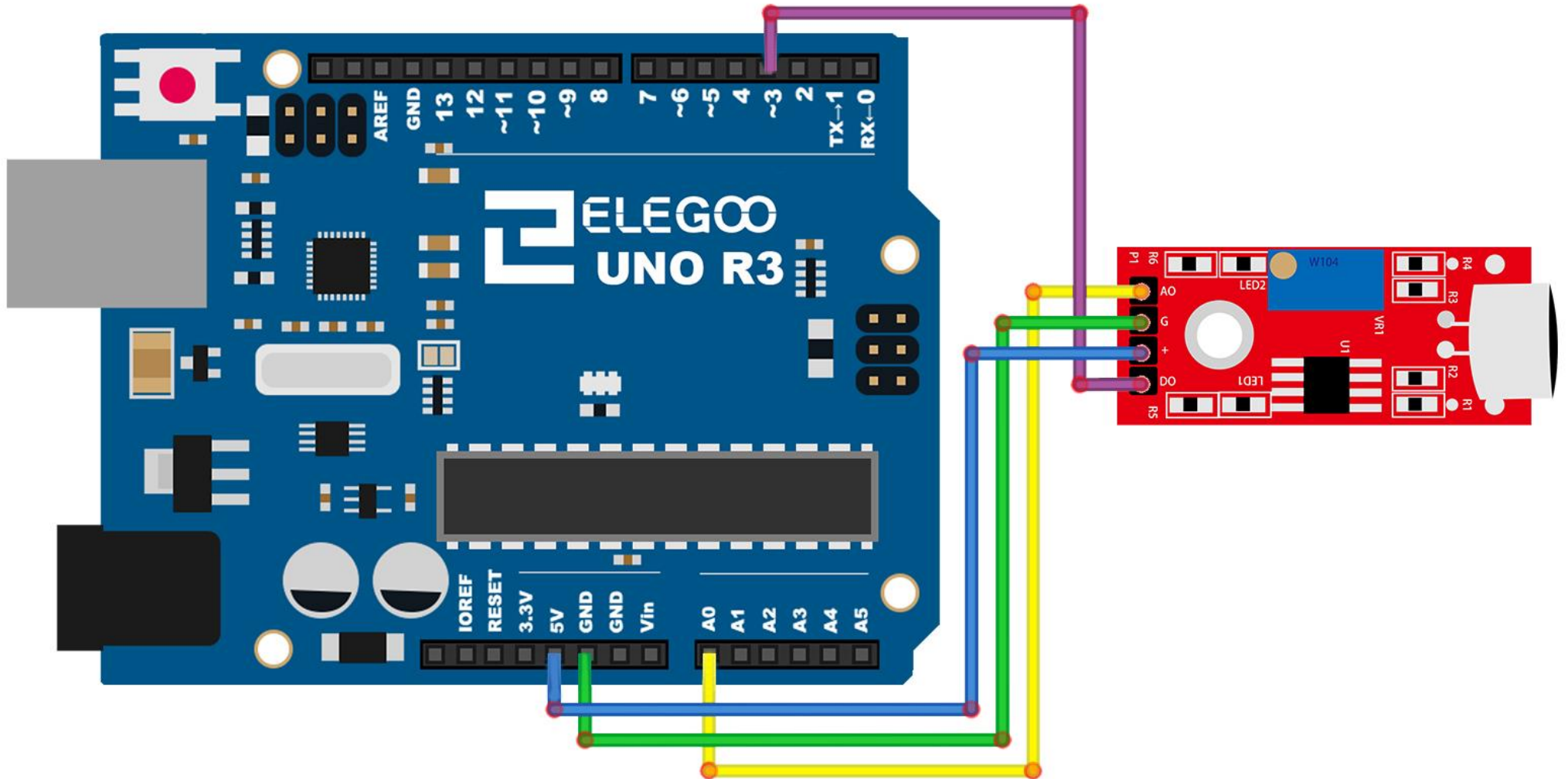
Ein fester leitender Metallkörper kapselt die verschiedenen Teile des Mikrofans. Die obere Fläche wird mit einem porösen Material mit der Hilfe von Klebstoff bedeckt. Er wirkt wie ein Filter für die Staubpartikel. Schallsignale/ Luftschwingungen laufen durch das poröse Material durch und fallen auf das Diaphragma durch das Loch, wie im Bild oben dargestellt.

Verbindung

Schema



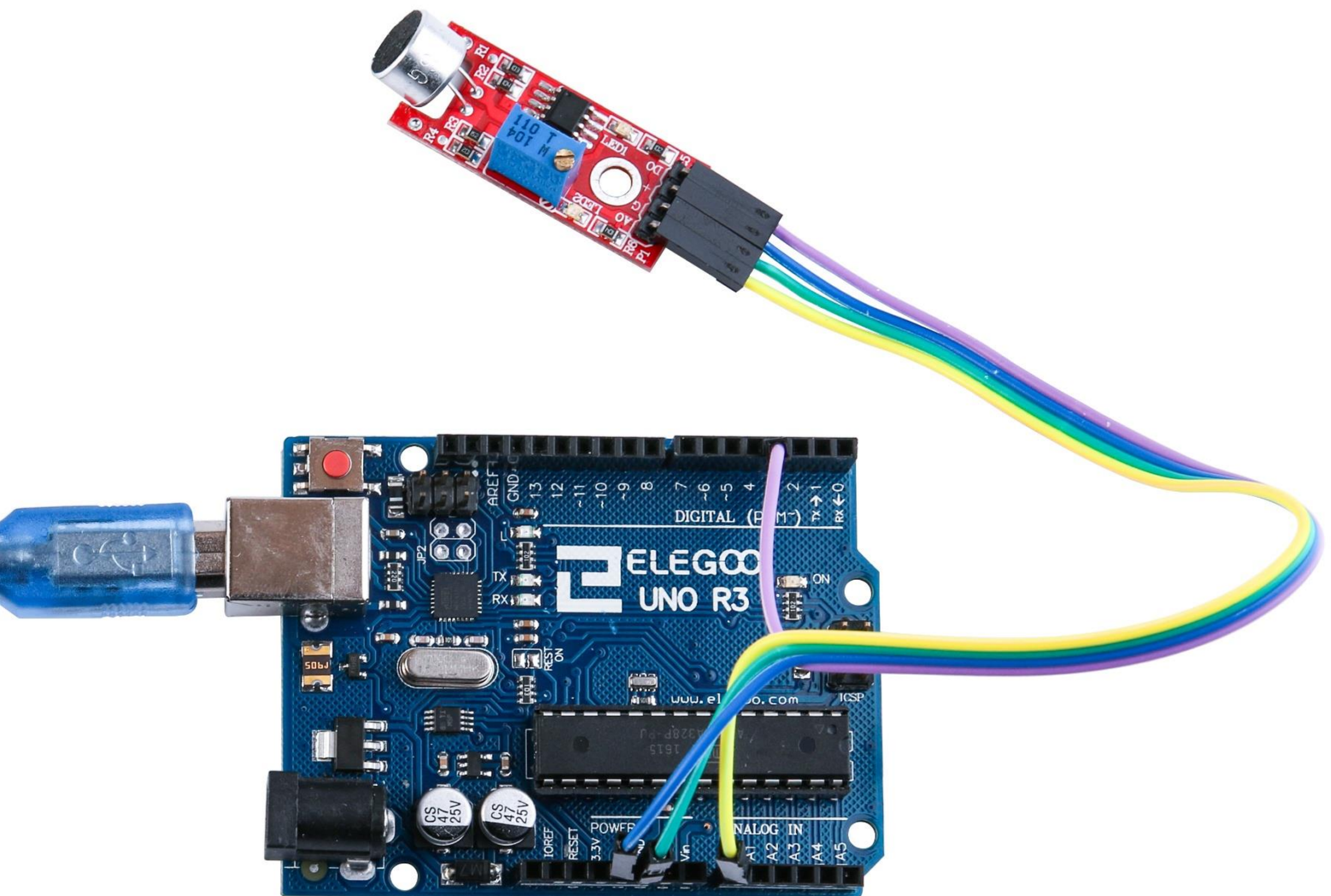
Schaltplan



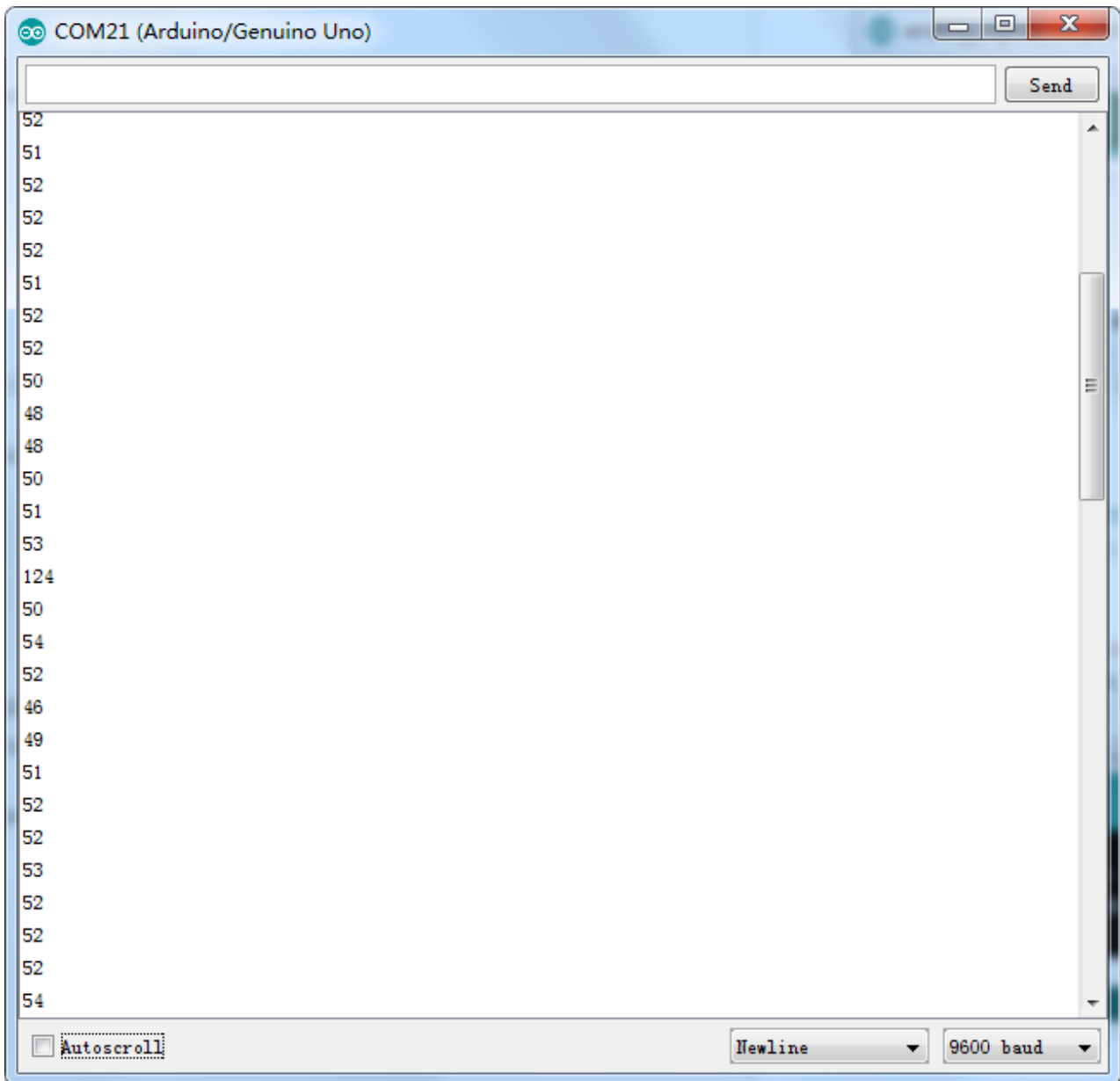
Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Dieses Modul bietet zwei Signal Ausgabemodi, wofür wir zwei Codes schrieb: "Digitale_Signalausgabe" und "Analoger_Signalausgabe". Der Code, Digitale_Signalausgabe, funktioniert so, dass wenn die Stimme, bei Erreichen eines bestimmten Wertes, ein digitales Signal auslöst, die Digital #11-Pin auf dem Arduino hohen Ausgangspegel wird und das Kennzeichen L zur gleichen Zeit beleuchtet werden wird. Diese auslösenden Wert kann entsprechend der oben erwähnten Empfindlichkeit-Anpassungsmethode geändert werden. Auf der anderen Seite der Code, Analoger_Signalausgabe, liest den analogen Wert des Moduls und direkt auf den seriellen Monitor anzeigen, ebenso kann dieser Wert auch nach der oben erwähnten Empfindlichkeit-Anpassungsmethode geändert werden.



Schalten Sie den Monitor ein, so können Sie die Daten wie folgt sehen:



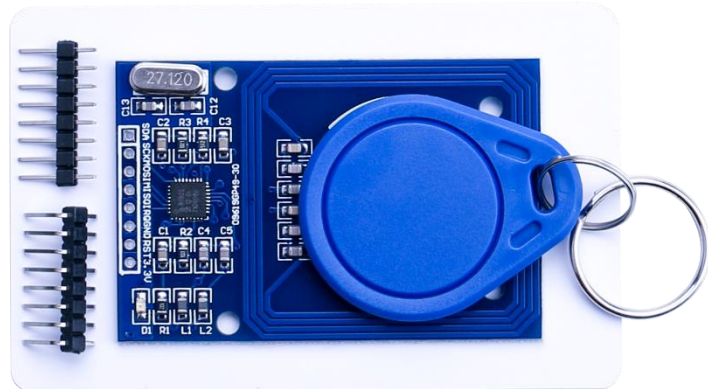
Lektion 21: RC522 RFID-Modul

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie RC 522 RFID Lesemodul auf UNO R3 verwenden. Dieses Modul verwendet Serial Peripheral Interface (SPI)-Bus, um mit Controllern wie Arduino, Raspberry Pi, Beagle Board, etc. zu kommunizieren.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) RC522 Modul
- (3) F-M Kabel



Komponentenanweisung

RC522:

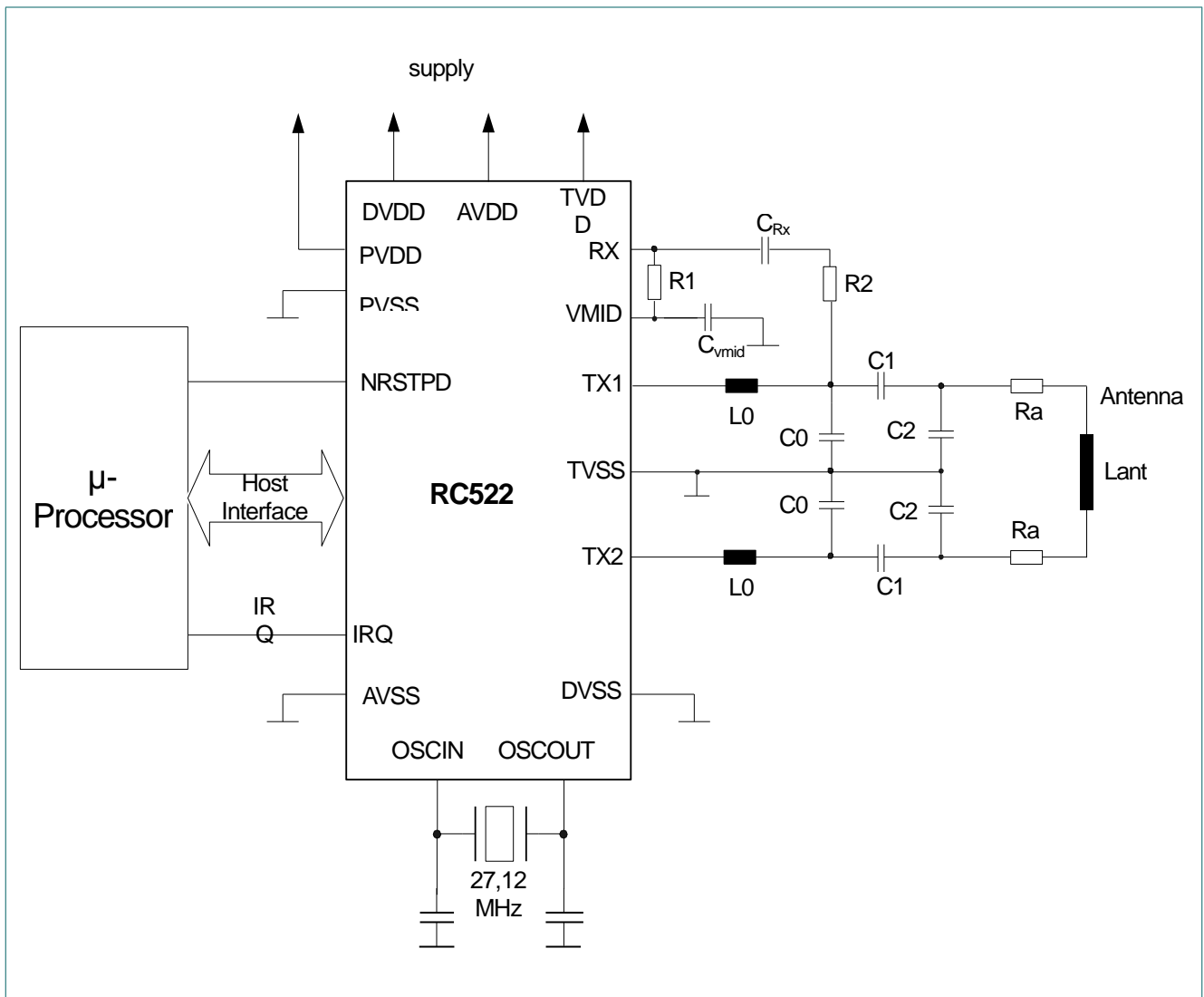
MFRC522 ist ein hochintegriertes Lese- / Schreibgerät für berührungslose Kommunikation bei 13,56MHz. Das MFRC522 Lesegerät unterstützt ISO 14443A / MIFARE® -Modus.

Das interne Senderteil von MFRC522 kann eine Lese- / Schreibgerätantenne, die für die Kommunikation mit ISO/IEC 14443A/MIFARE®-Karten und Transponder ohne zusätzliche aktive Schaltung entwickelt wird, betreiben. Das Empfängerteil bietet eine robuste und effiziente Implementierung einer Demodulation und einer Dekodierungsschaltung für Signale aus den Karten und Transpondern, die mit ISO/IEC 14443A/MIFARE® kompatibel sind, an. Das digitale Teil handhabt die komplette ISO/IEC 14443A Rahmung und Fehlererkennung (Parity & CRC). MFRC522 unterstützt MIFARE®Classic (z.B. MIFARE® Standard) Produkte. MFRC522 unterstützt berührungslose Kommunikation unter Verwendung der MIFARE® höheren Übertragungsraten bis 848 kbit/s in beiden Richtungen.

Verschiedene Host-Schnittstellen werden implementiert:

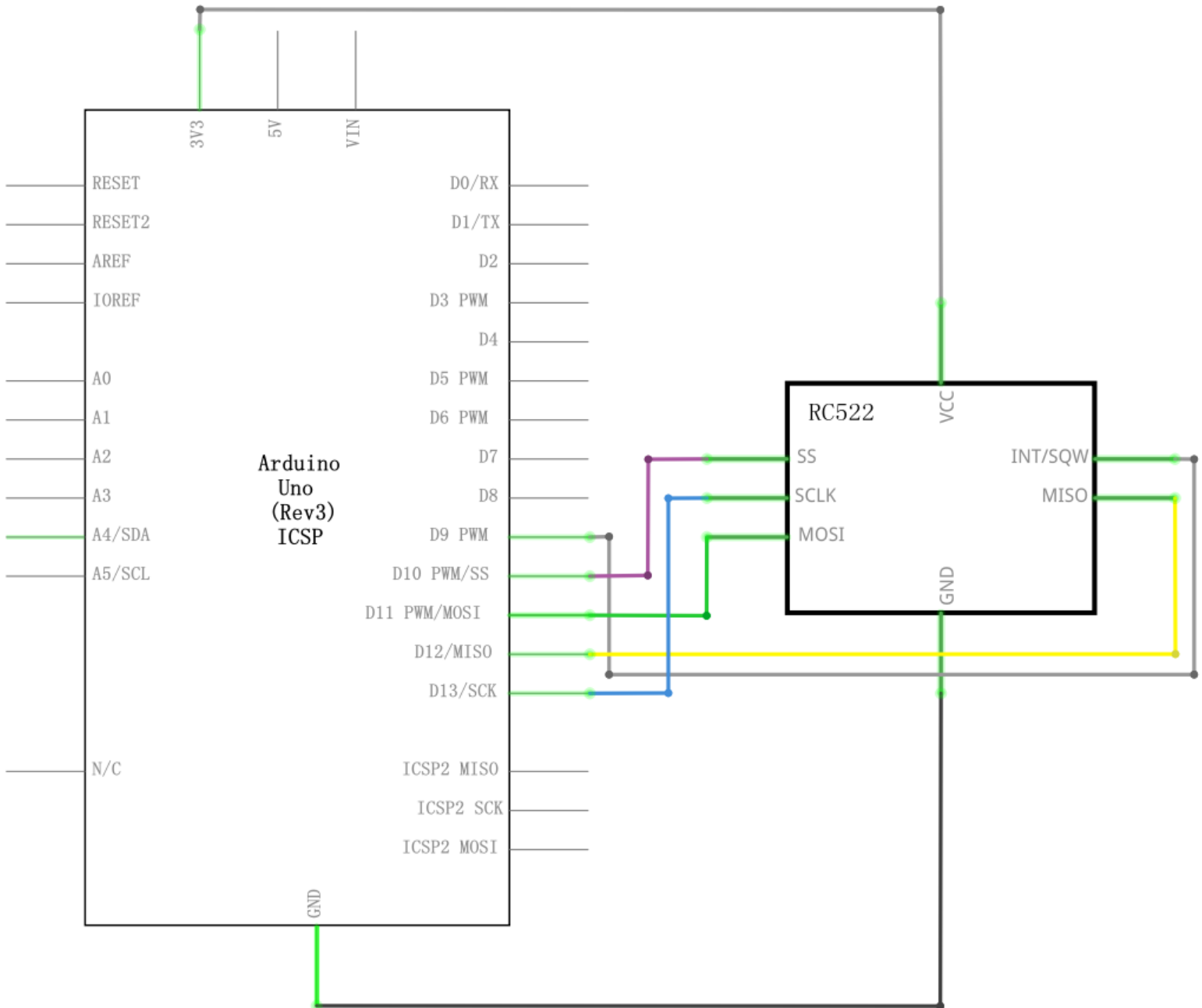
- SPI-Schnittstelle
- Serieller UART (ähnlich wie RS232 mit Spannungspegeln nach Pad Spannungsversorgung)
- 12C-Schnittstelle.

In der folgenden Abbildung wird ein typisches Schaltungsdiagramm dargestellt, mit einer komplementären Antennenverbindung zum MFRC522.

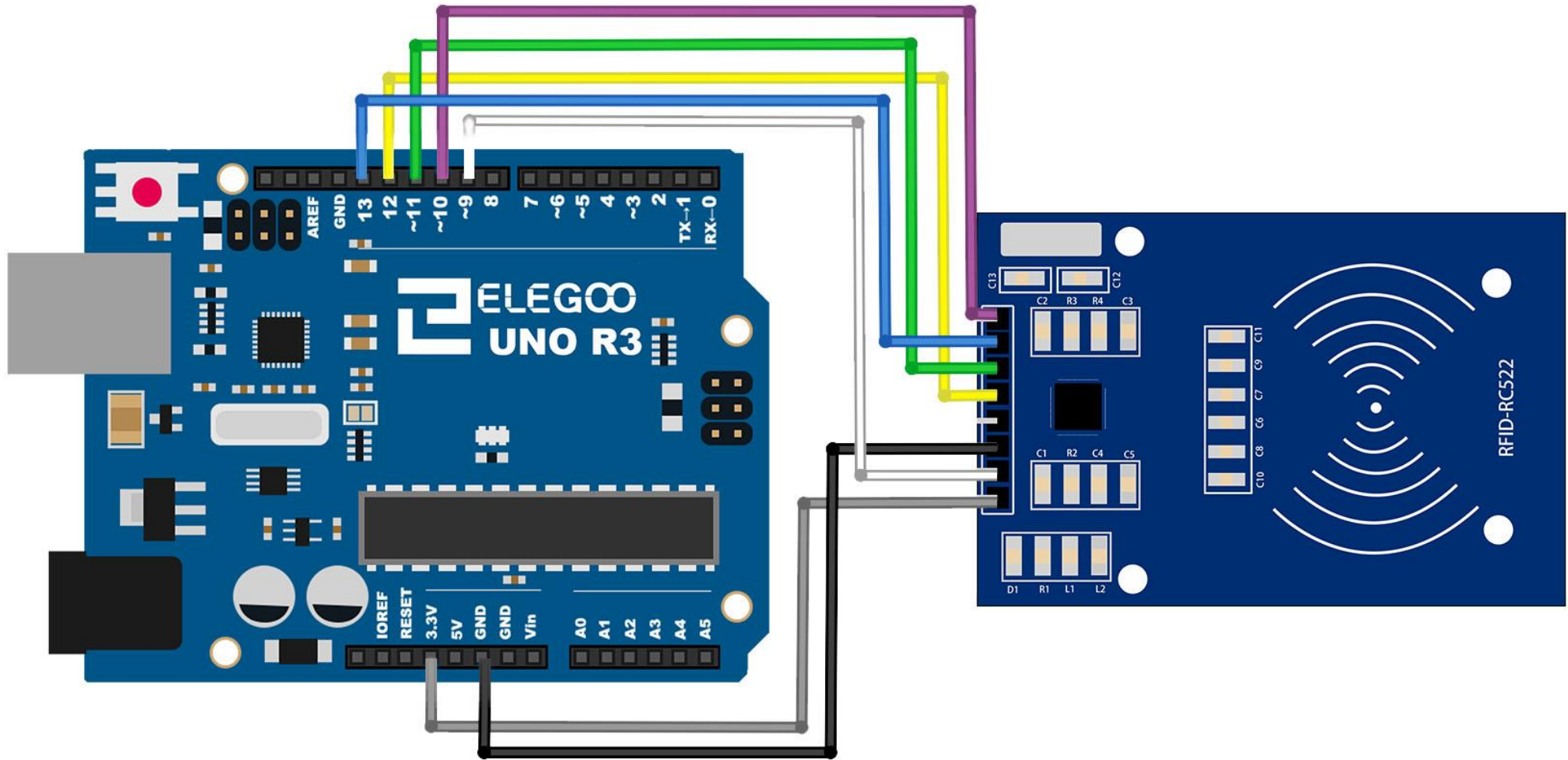


Verbindung

Schema



Schaltplan



Code

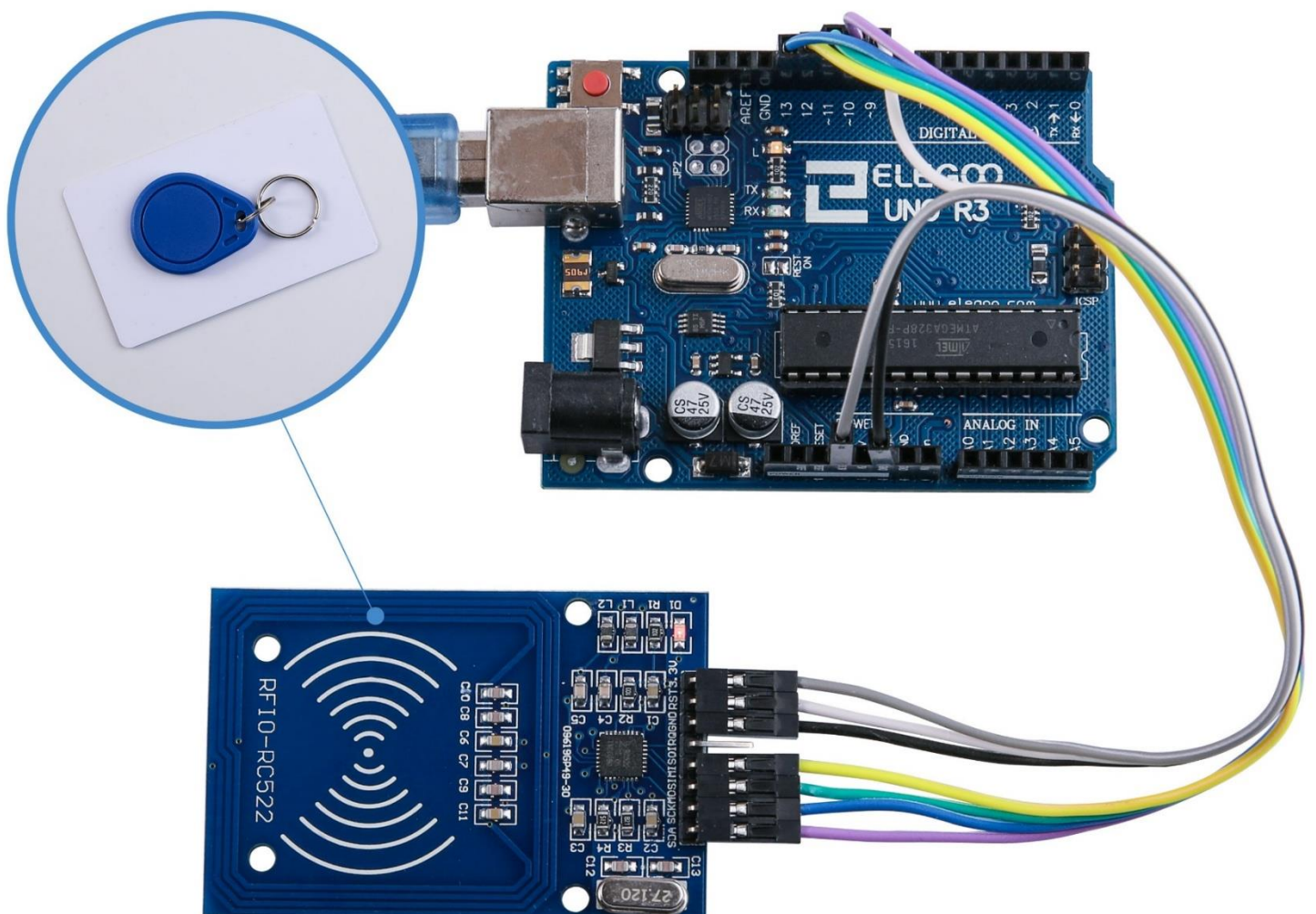
Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefodert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek <rfid> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.

	MFRC522	Arduino	Arduino	Arduino	Arduino	Arduino
	Reader/PCD	Uno	Mega	Nano v3	Leonardo/Micro	Pro Micro
Signal	Pin	Pin	Pin	Pin	Pin	Pin
RSI/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 9 // Configurable, see typical pin layout above
#define SS_PIN 10 // Configurable, see typical pin layout above
```

The locations of SPI pins vary with different chips, and you have to make a minor modification of the function.



Schalten Sie den Monitor ein, so können Sie die Daten wie folgt sehen:

Lektion 22: LCD-Anzeige

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie verdrahten und eine alphanumerische LCD-Anzeige verwenden.

Die Anzeige besitzt eine LED-Hintergrundbeleuchtung, und kann zwei Reihen mit bis zu 16 Zeichen auf jeder Reihe anzeigen. Sie können Rechtecke für jedes Zeichen und die Pixeln, die jedes Zeichen bilden, auf der Anzeige sehen. Die Anzeige ist nur weiß auf blau, und ist für das Anzeigen des Texts bestimmt.

In dieser Lektion werden wir das Arduino-Beispielprogramm für die LCD-Anzeige ausführen, aber in der nächsten Lektion werden wir unsere Anzeige unter Verwendung von Sensoren die Temperatur und das Lichtniveau anzeigen lassen

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) LCD1602 Anzeige
- (1) Potentiometer
- (1) Versuchsaufbau
- (16) F-M Kabel



Komponentenanweisung

LCD1602:

VSS: Ein Pin, der mit der Masse verbunden wird

VDD: Ein Pin, der mit einem Netzteil von +5V verbunden wird

VO: Ein Pin, der den Kontrast von LCD 1602 einstellt

RS: Ein von Register ausgewählter Pin, der kontrolliert, worin in dem LCDs Speicher Sie Daten schreiben. Sie können entweder das Datenregister, das kontrolliert, was auf dem Bildschirm passiert, oder ein Befehlsregister, das kontrolliert, wo der LCD-Kontroller Befehle über den nächsten Schritt sucht, auswählen.

R/W: Ein Lese-/Schreibpin, der das Lese- oder das Schreibmodus auswählt

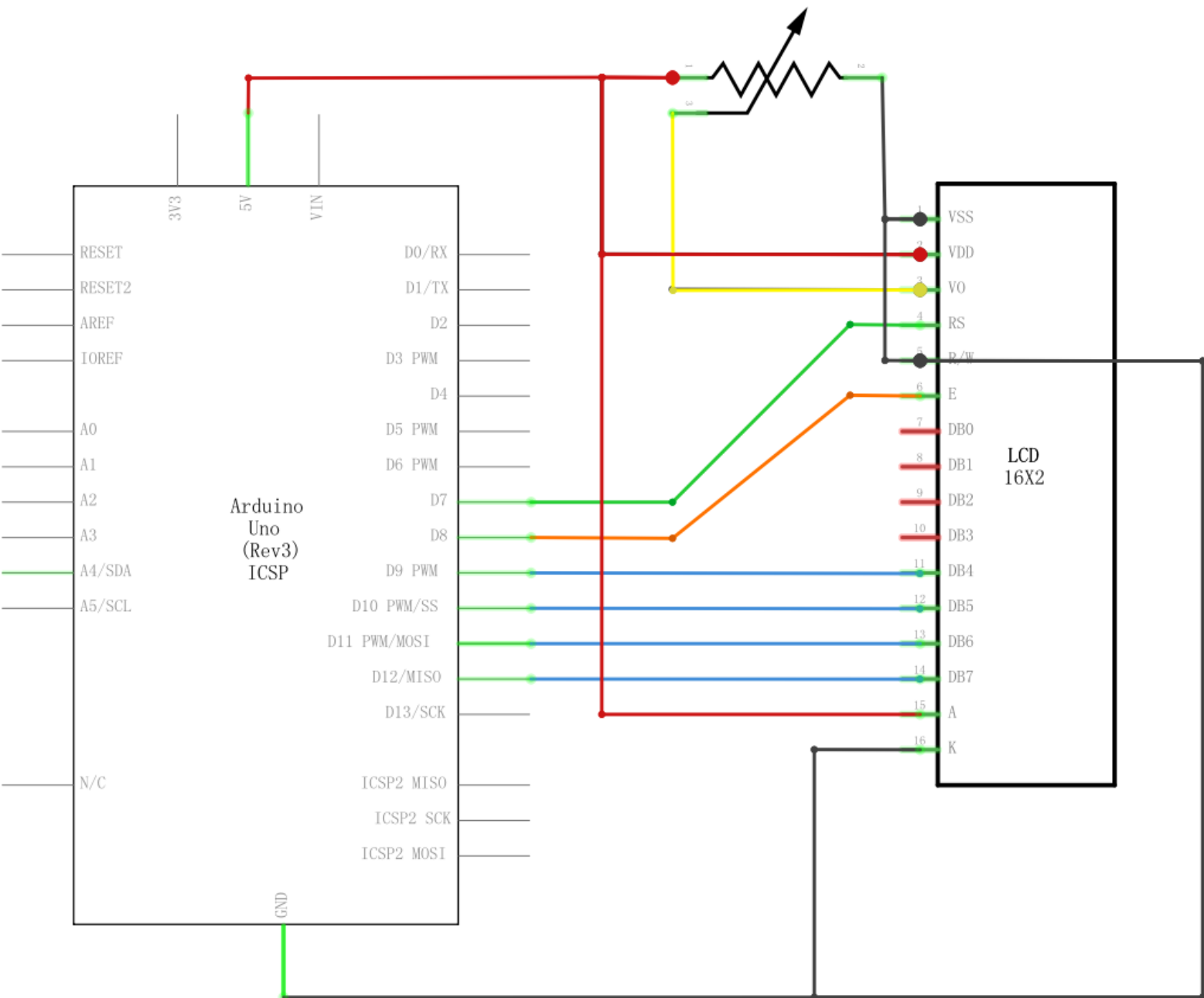
E: Ein aktivierter Pin, der bei Versorgung mit niedriger Spannung das LDC-Modul veranlasst, entsprechende Befehle auszuführen

D0-D7: Pins, die Daten lesen und schreiben

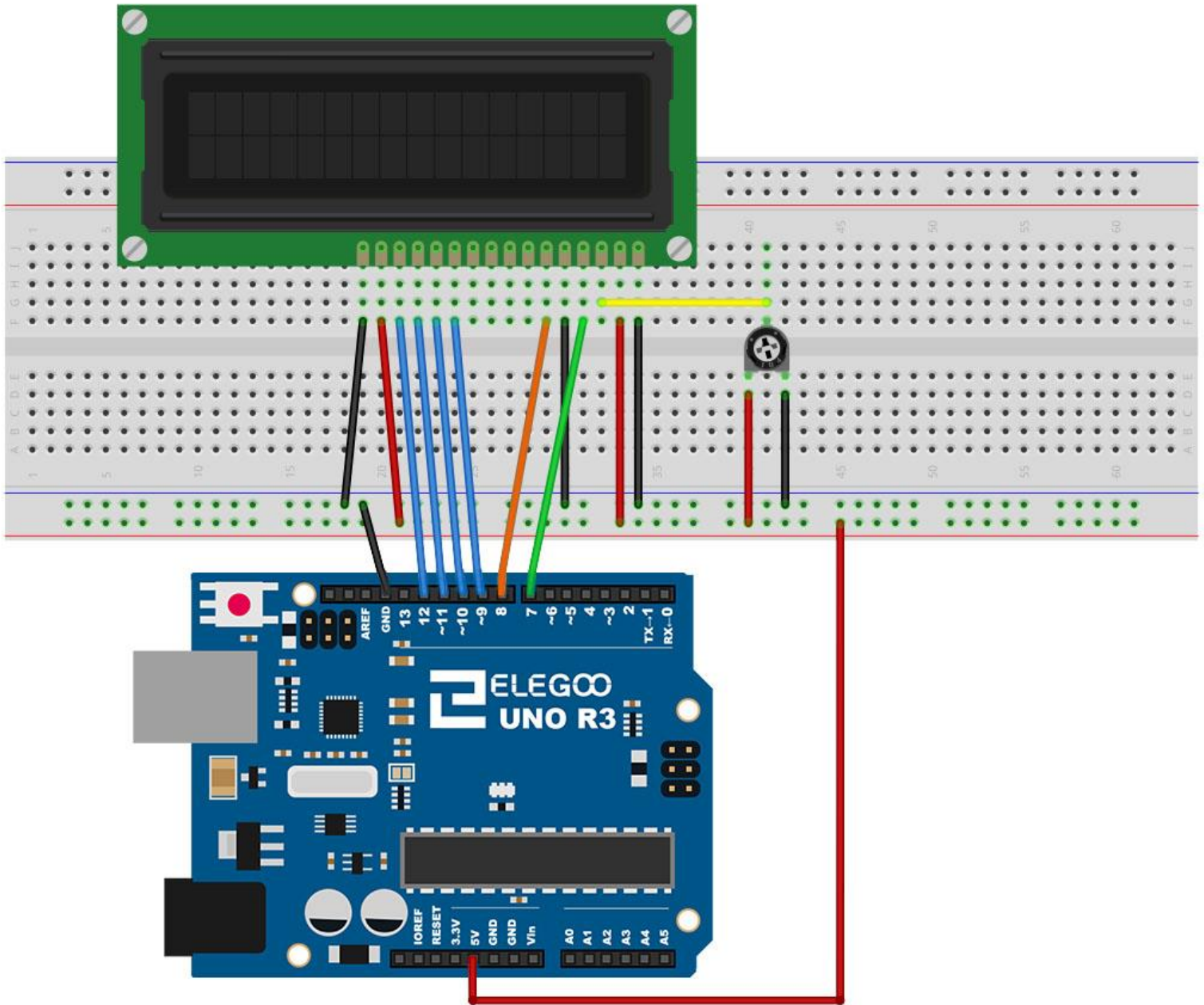
A und K: Pins, die die LED-Hintergrundbeleuchtung kontrollieren

Verbindung

Schema



Schaltplan



Die LCD-Anzeige benötigt sechs Arduino-Pins, und alle der Pins werden als digitale Ausgänge eingesetzt. Die LCD-Anzeige benötigt auch 5 V- und GND-Anschlüsse.

Es gibt eine Reihe von Verbindungen, die hergestellt werden. Stellen Sie die Anzeige mit der Spitze des Versuchsaufbaus auf, was hilft, ihre Pins ohne zu viel Zählung zu identifizieren, besonders wenn die Reihen des Versuchsaufbaus mit Reihe 1 wie die obere Reihe des Aufbaus nummeriert werden. Vergessen Sie nicht, dass die lange gelbe Leitung den Schieber der Pot an den Pin 3 der Anzeige anschließt. Die ‚Pot‘ wird verwendet, um den Kontrast der Anzeige zu kontrollieren.

Sie könnten finden, dass keine Header-Pins an ihrer Anzeige befestigt sind. Wenn ja, befolgen Sie die Anweisungen im nächsten Abschnitt.

Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Jetzt haben Sie das physikalische Setup, und alles, was wir nun brauchen, ist der Code.

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

In diesem Beispiel verwenden wir unterschiedliche Pins, so können wir die Codezeilen wie folgt finden:

```
LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
```

und wechseln Sie es in:

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

Laden Sie den Code auf Ihr Arduino-Board auf, und Sie sollten die Meldung ‚hello, world‘ auf der Anzeige sehen, verfolgt von einer Zahl, die von Null aufwärts zählt.

An erster Stelle sollten Sie die Zeile in dem Sketch bemerken:

```
#include <LiquidCrystal.h>
```

Dies erzählt Arduino, dass wir die Liquid Crystal-Bibliothek verwenden möchten.

Als nächstes ist die Zeile, die wir ändern müssen. Dies legt fest, welche Pins von Arduino an welche Pins der Anzeige angeschlossen werden.

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

Die Argumente dafür sind wie folgt:

Anzeige Pinbezeichnung Anzeige Pinnummer Arduino-Pin (in diesem Beispiel) RS 4 7 E 6 8 D4 11 9 D5
12 10 D6 13 11 D7 14 12

Nach dem Hochladen dieses Codes stellen Sie sicher, dass die Hintergrundbeleuchtung leuchtet, und stellen Sie das Potentiometer immer ein, bis dass Sie die Textmeldung sehen.

In der ‚Setup‘-Funktion haben Sie zwei Befehle:

```
lcd.begin(16, 2);
```

```
lcd.print(„hello, world!“);
```

Der erste Befehl erzählt der Liquid Crystal-Bibliothek, wie viele Spalten und Reihen die Anzeige hat. In der zweiten Zeile wird die Meldung, die Sie auf der ersten Zeile der Anzeige sehen, angezeigt.

In der ‚Loop‘-Funktion haben Sie auch zwei Befehle:

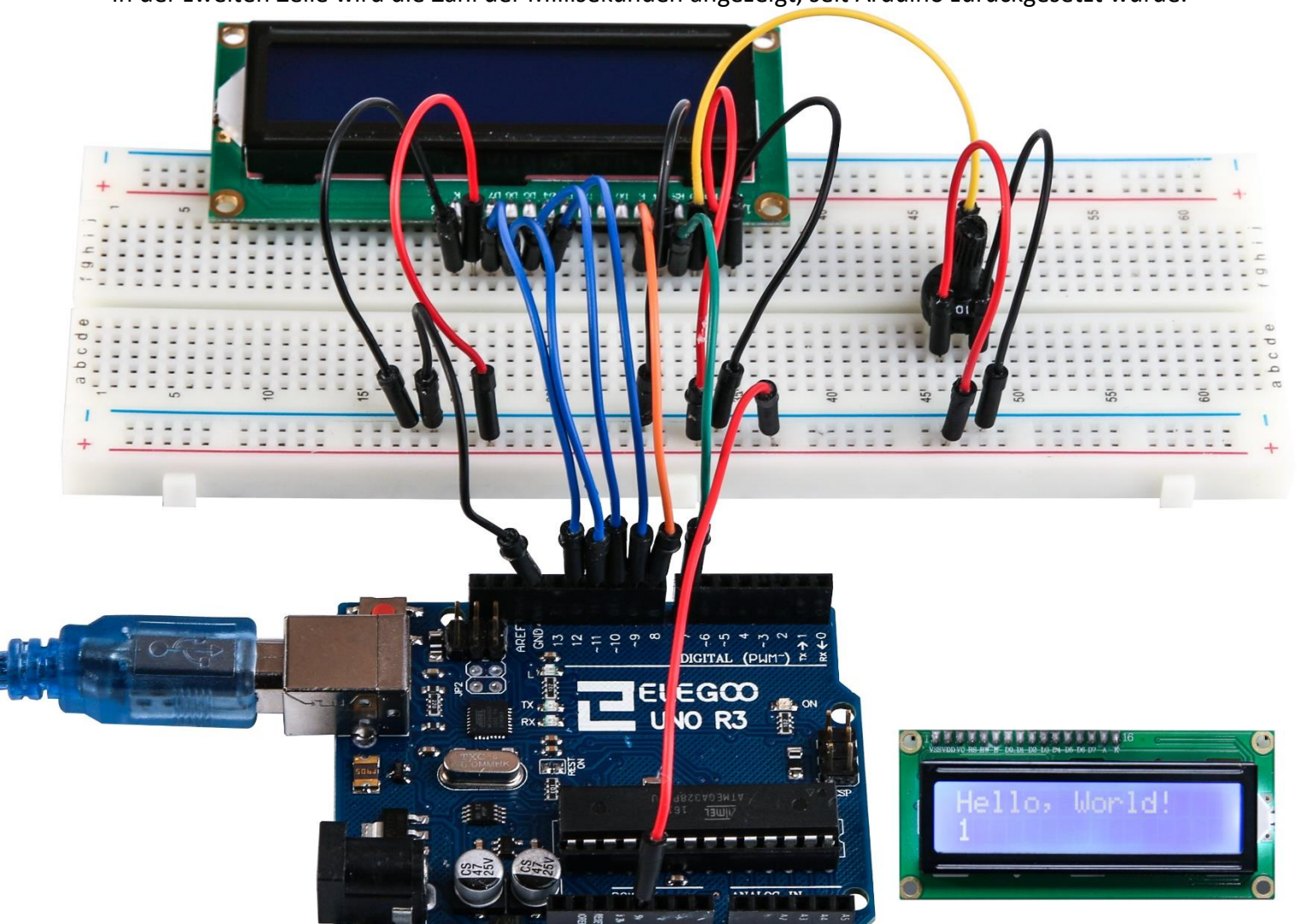
```
lcd.setCursor(0, 1);
```

```
lcd.print(millis()/1000);
```

Der erste Befehl setzt den Cursor (wo der nächste Text erscheinen wird) auf Spalte 0 & Reihe 1.

Sowohl die Spalten- als auch die Reihenzahlen beginnen mit 0 anstatt mit 1

In der zweiten Zeile wird die Zahl der Millisekunden angezeigt, seit Arduino zurückgesetzt wurde.



Lektion 23: Thermometer

Übersicht

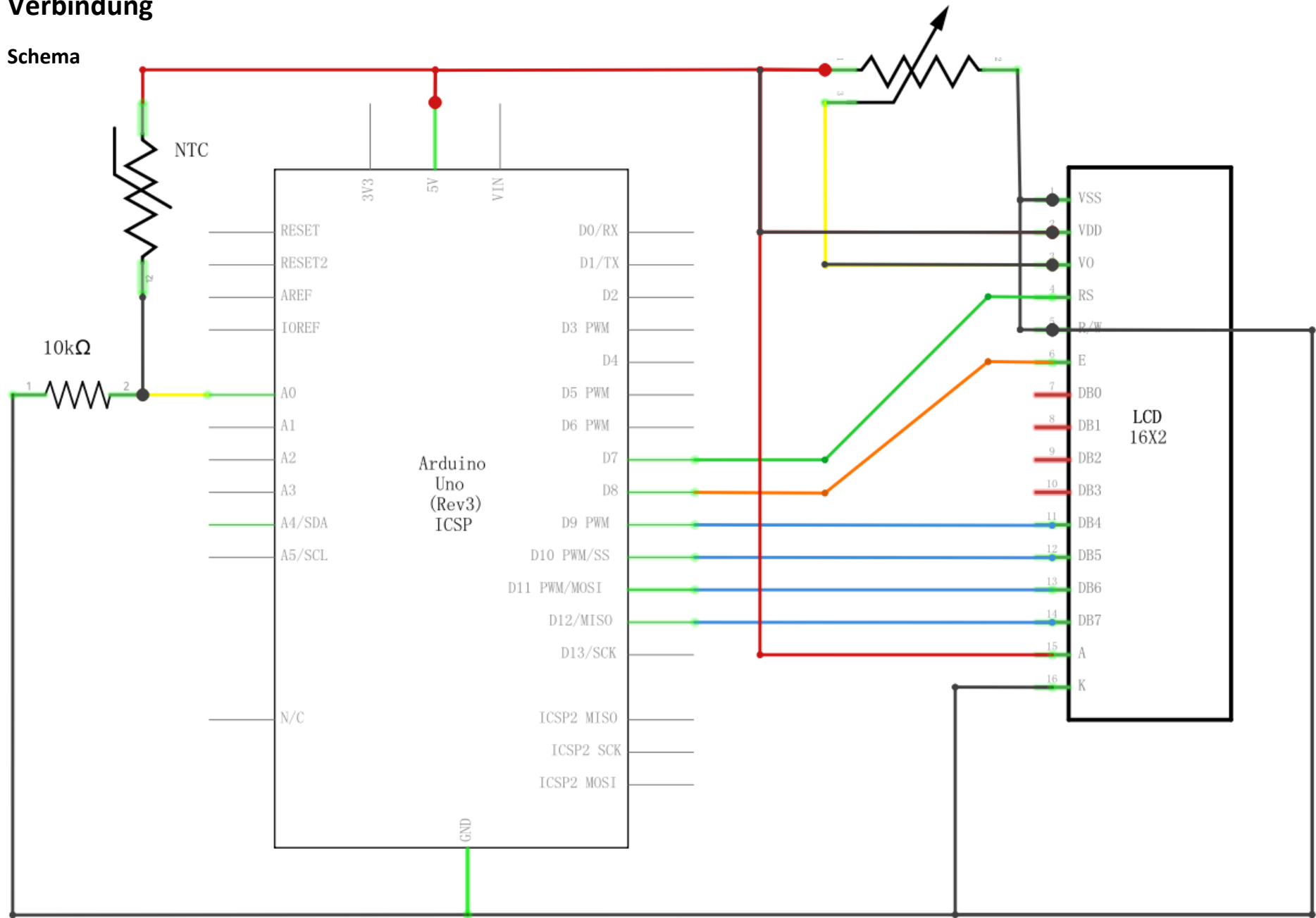
In dieser Lektion werden Sie eine LCD-Anzeige verwenden, um die Temperatur anzuzeigen.

Erforderliche Komponenten

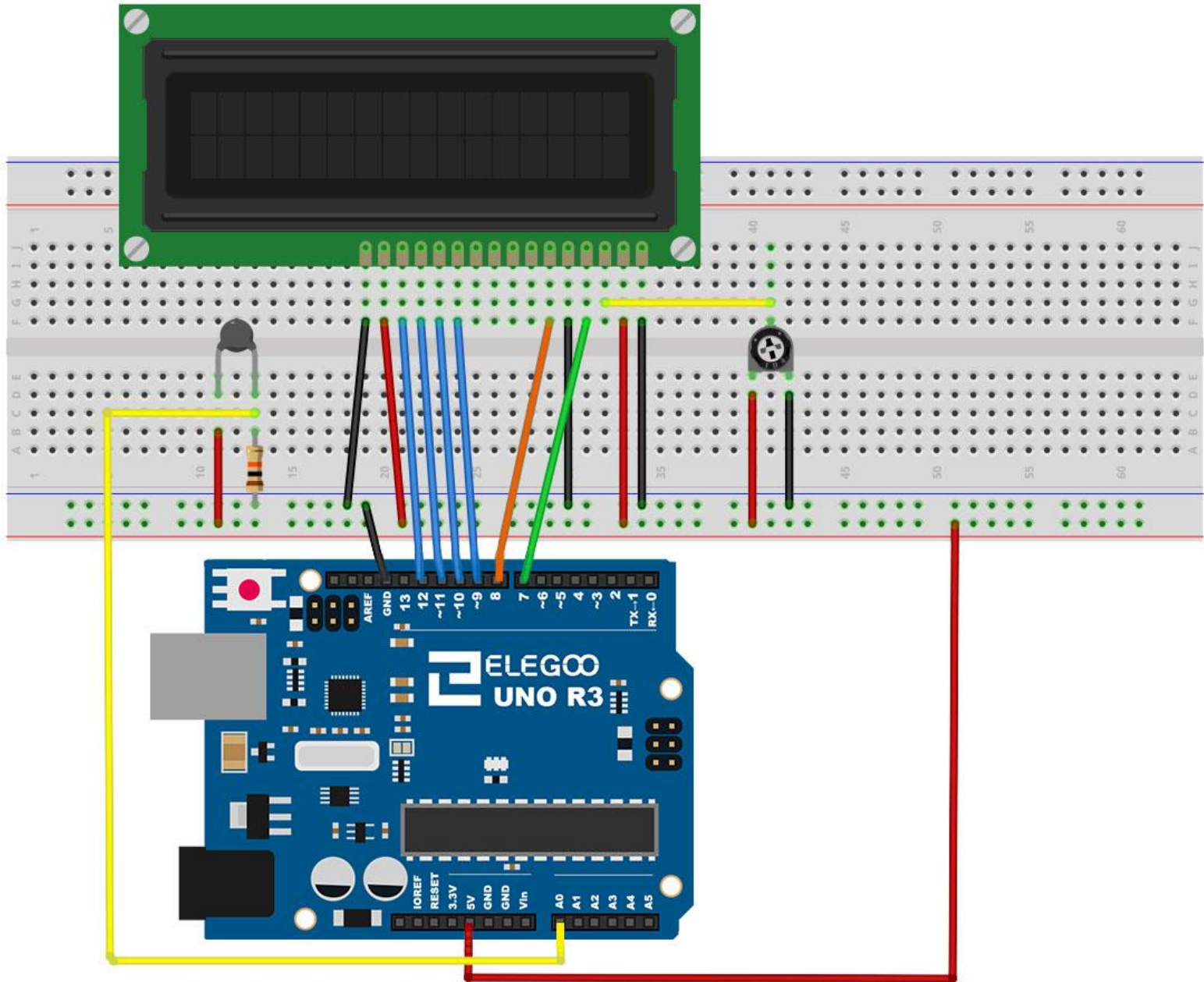
- (1) Elegoo UNO R3
- (1) LCD1602 Anzeige
- (1) 10k Ohm-Widerstand
- (1) Thermistor
- (1) Potentiometer
- (1) Versuchsaufbau
- (16) F-M Kabel

Verbindung

Schema



Schaltplan



Das Layout vom Versuchsaufbau basiert auf dem Layout der Lektion 16, so können Dinge vereinfacht, wenn Sie dies noch auf dem Versuchsaufbau haben.

Es gibt einige Überbrückungskabel in der Nähe der Pot, die etwas auf diesem Layout verschoben wurde.

Die Fozozelle, 1kΩ-Widerstand und Thermistor sind alle neuen Ergänzungen des Aufbaus.

Code

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

Der Sketch herfür basiert auf den in der Lektion 14. Laden Sie es auf Ihren Arduino hoch, und Sie sollten finden, dass die Temperaturmessung erhöht wird, wenn Sie den Temperatursensor erwärmen, indem Sie Ihre Finger auf den Temperatursensor setzen.

Es ist sinnvoll, eine Kommentarzeile über dem Befehl ,lcd' zu setzen.

```
//           BS  E  D4 D5  D6 D7  
LiquidCrystal lcd(7, 8, 9, 10, 11, 12);
```

Dies macht Dinge einfacher, wenn Sie darüber entscheiden, die von Ihnen verwendeten Pins zu wechseln.

In der ,Loop'-Funktion gibt es nun zwei lustige geschehende Dinge. Erstens müssen wir die analoge Temperatur aus dem Temperatursensor in eine tatsächliche Temperatur umwandeln, und zweitens müssen wir herausfinden, wie sie angezeigt werden.

Vor allem schauen wir auf die Berechnung der Temperatur.

```
int tempReading = analogRead(tempPin);  
float tempVolts = tempReading * 5.0 / 1024.0;  
float tempC = (tempVolts - 0.5) * 100.0;  
float tempF = tempC * 9.0 / 5.0 + 32.0;
```

Die originale Ablesung aus Temperatursensor wird zuerst mit 5 multipliziert, und dann durch 1024 geteilt, um uns die Spannung (zwischen 0 und 5) an dem analogen ,tempPin'-Eingang zu geben.

Um die Spannung aus TMP 36 in eine Temperatur in Grad C umzuwandeln, müssen Sie 0,5V von der Messung subtrahieren und dann mit 100 multiplizieren lassen.

Um diese Spannung in eine Temperatur in Fahrenheit umzuwandeln, müssen Sie es mit 9/5 multiplizieren lassen und dann 32 addieren.

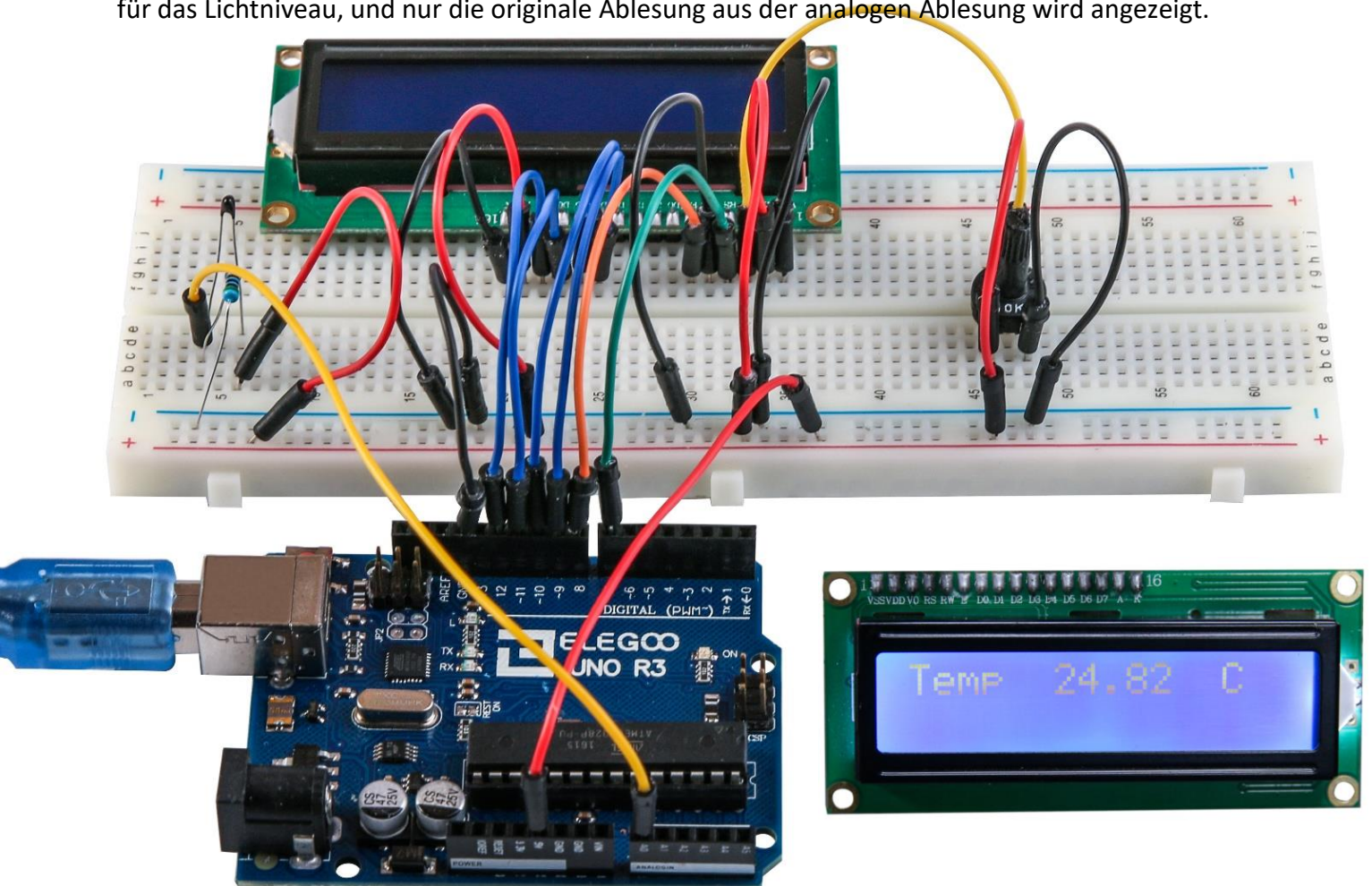
Anzeigen der geänderten Ablesungen auf einer LCD-Anzeige kann knifflig sein. Das Hauptproblem liegt darin, dass die Ablesung nicht immer die gleiche Zahl von Ziffern sein könnte. Wenn die Temperatur von 101,50 bis 99,00 geändert wird, besteht es bei der zusätzlichen Ziffer aus der alten Ablesung die Gefahren, auf der Anzeige verblieben zu werden.

Um dies zu vermeiden, schreiben Sie die ganze Zeile auf der LCD jedes Mal um die Schleife.

```
// -----  
lcd.print(„Temp      F “);  
lcd.setCursor(6, 0);  
lcd.print(tempF);
```

Der ziemlich merkwürdige Kommentar soll Sie an die 16 Spalten der Anzeige erinnern. Sie können dann eine Reihe von dieser Länge mit Leerzeichen, wo die tatsächliche Ablesung gehen wird, drucken. Um in die Lücken auszufüllen, setzen Sie den Cursor in die Position, wo die Ablesung erscheinen sollte, und dann drucken Sie sie.

Genau der gleiche Ansatz wird für das Anzeigen des Lichtniveaus verwendet. Es gibt keine Einheiten für das Lichtniveau, und nur die originale Ablesung aus der analogen Ablesung wird angezeigt.



Lektion 24: Acht LEDs mit 74HC595

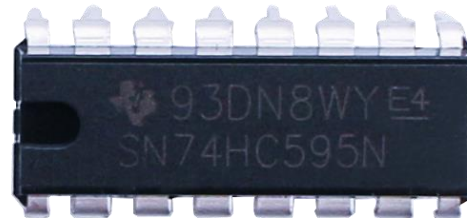
Übersicht

In dieser Lektion werden Sie erfahren, wie Sie acht große rote LEDs mit einem UNO verwenden, ohne dass Sie auf 8 Ausgangspins verzichten müssen!

Obwohl Sie acht LEDs mit jeweils einem Widerstand an den UNO-Pin verdrahten könnten, werden die Pins auf Ihrem UNO schnell verbraucht. Wenn Sie nicht so viele Sachen mit Ihrem UNO verbinden, ist es in Ordnung, dies zu tun – aber bevor Sie wissen, dass keine Pins bei Ihnen verblieben sind, brauchen wir oft Tasten, Sensoren, Servomechanismen, etc. So werden Sie stattdessen einen Chip mit der Bezeichnung 74HC595 Seriell-Parallel-Wandler verwenden. Dieser Chip hat acht Ausgänge (vorzugsweise) und drei Eingänge, die Sie verwenden, um Daten ein wenig in einer Zeit einzuspeisen. Dieser Chip lässt es ein wenig langsamer sein, die LEDs zu betreiben (Sie können nur die LEDs ca. 500.000 Male pro Sekunde statt 8.000.000 Male pro Sekunde ändern), aber es ist auch sehr schnell, viel schneller, als dass Menschen erkennen können, so lohnt es sich!

Erforderliche Komponenten

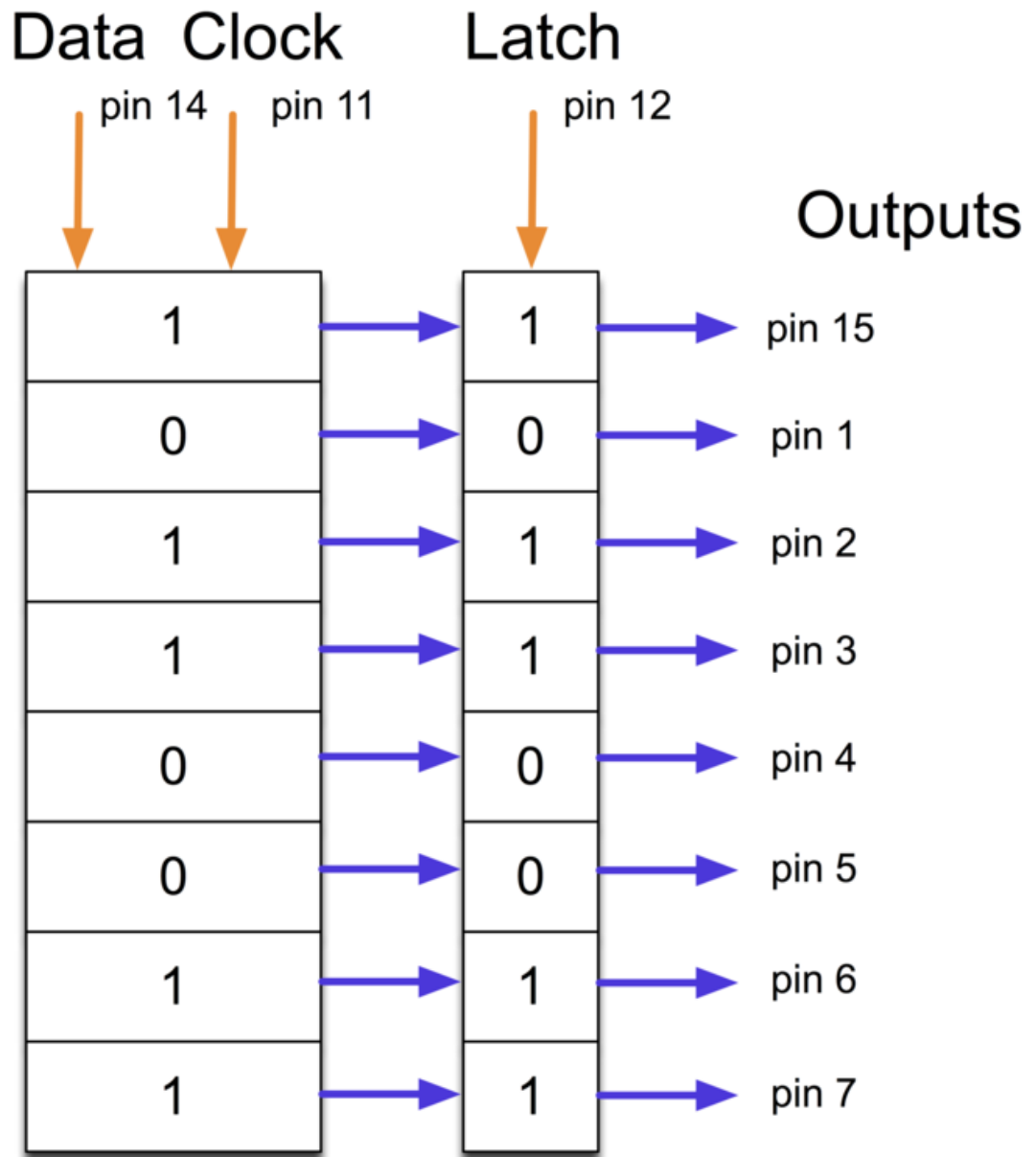
- (1) Elegoo UNO R3
- (1) Versuchsaufbau
- (8) LEDs
- (8) 220 Ohm-Widerstände
- (1) 74HC595 IC
- (14) M-M Kabel



Komponentenanweisung

74HC595 Schieberegister:

Das Schieberegister ist eine Art von Chip, der acht Speicherplätze besitzen kann, und jeder der Speicherplätze kann entweder eine 1 oder eine 0 sein. Um jeder dieser Wert ein- oder auszuschalten, speisen wir die Daten mit Pins ‚Data‘ und ‚Clock‘ von dem Chip ein.

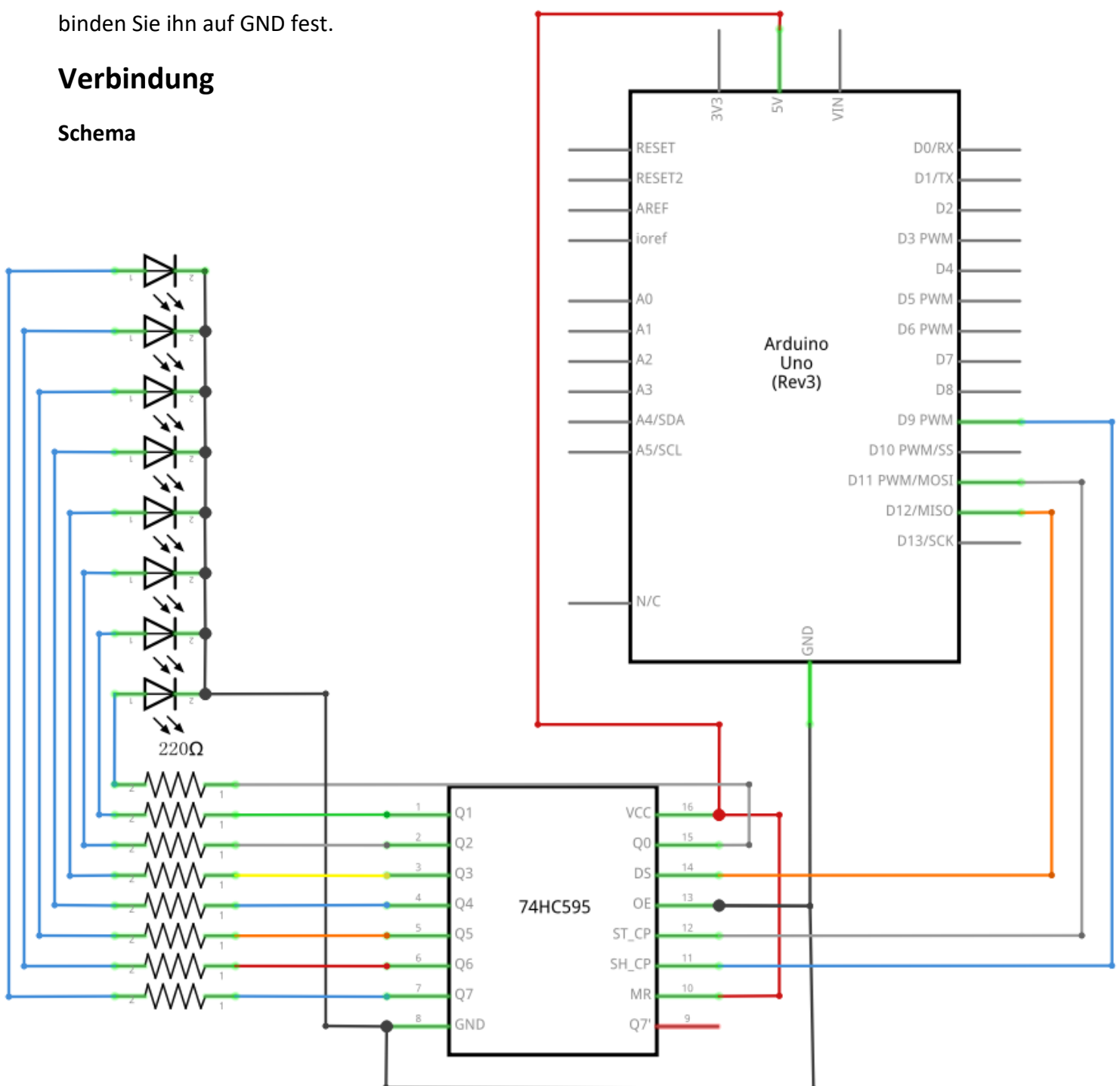


Der Clock-Pin muss acht Impulse empfangen. Falls der Daten-Pin bei jedem Impuls hoch ist, wird eine 1 in das Schieberegister geschoben; andernfalls, eine 0. Wenn alle acht Impulse empfangen wurden, ermöglicht der ‚Latch‘-Pin es Ihnen, solche acht Werte in das Latch-Register zu kopieren. Dies ist notwendig; ansonsten würden LEDs flackern, da die Daten in das Schieberegister geladen werden.

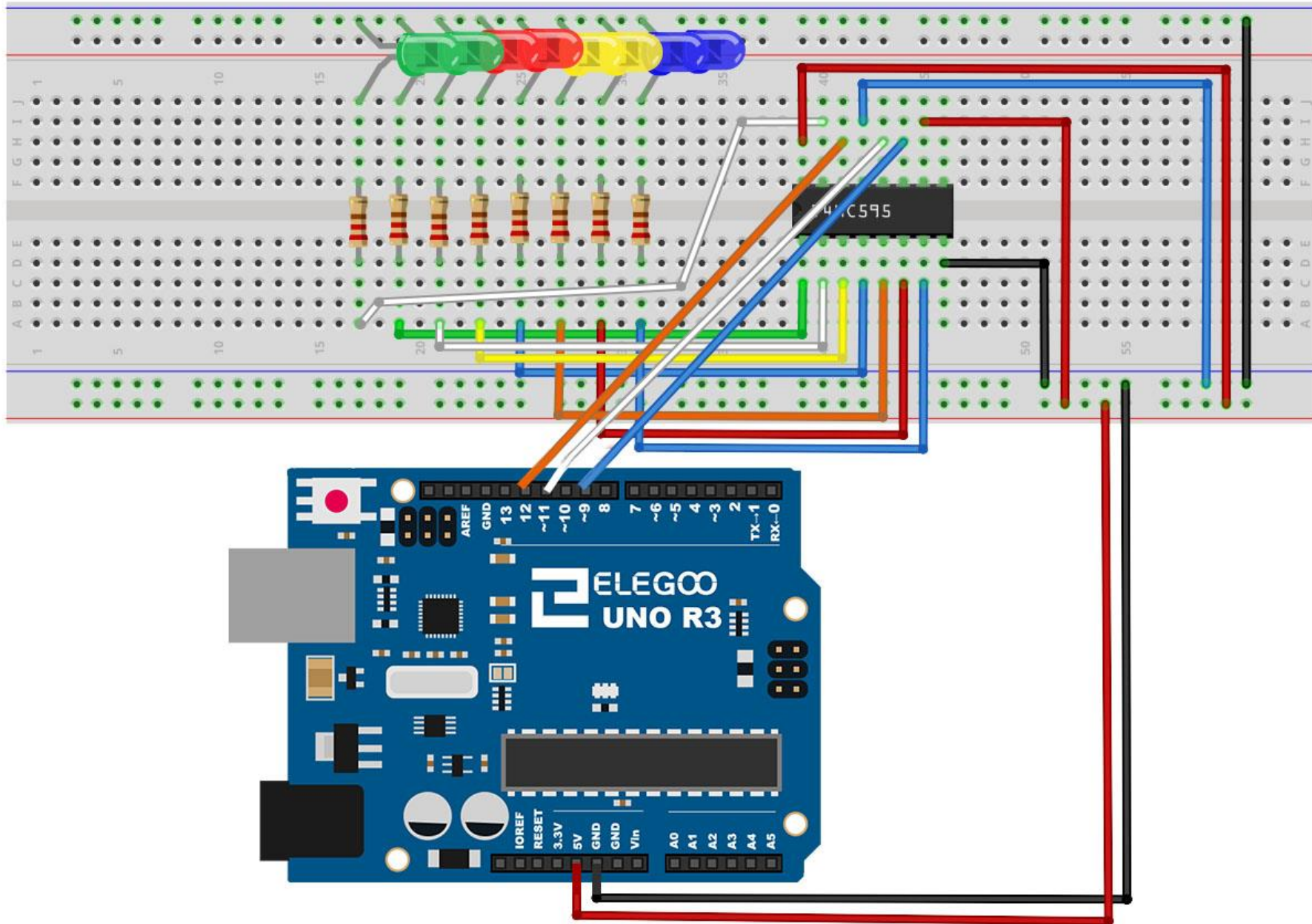
Der Chip verfügt auch über einen Output-Enable (OE)-Pin, der alle Ausgänge auf einmal aktivieren oder deaktivieren kann. Sie könnten ihn einem PWM-fähigen UNO-Pin beilegen, und ‚analogWrite‘ verwenden, um die Helligkeit der LEDs zu kontrollieren. Dieser Pin ist LOW-aktiv, so binden Sie ihn auf GND fest.

Verbindung

Schema



Schaltplan



Weil acht LEDs und acht Widerstände uns zur Verbindung stellen, werden tatsächlich ziemlich viele Verbindungen hergestellt.

Es ist vielleicht am einfachsten, zunächst den 74HC595 Chip einzustecken, weil so ziemlich alles andere an ihn angeschlossen wird. Stecken Sie ihn ein, so dass die kleine U-förmige Kerbe nach oben zum Versuchsaufbau hin zeigt. Pin 1 des Chips zeigt aber nach links zur Kerbe hin.

- Digitale 4 aus dem UNO geht auf Pin #14 des Schieberegisters
- Digitale 5 aus dem UNO geht auf Pin #12 des Schieberegisters
- Digitale 6 aus dem UNO geht auf Pin #11 des Schieberegisters

Alle außer einem dieser Ausgänge aus der integrierten Schaltung (IC) befindet sich auf der linken Seite des Chips. Daher ist hier auch, wo die LEDs sich befinden, um eine die Verbindung einfach herzustellen.

Nach dem Chip setzen Sie die Widerstände in die richtige Position. Sie müssen darauf achten, dass die Leitung eines Widerstands die Leitung eines anderen Widerstands nicht berührt. Sie sollten dies wieder überprüfen, bevor Sie das UNO mit Strom versorgen. Wenn es schwierig für Sie ist, die Widerstände anzuordnen, ohne dass sie keine Berührung miteinander haben, können Sie die Leitungen verkürzen, was hilfreich ist, so dass sie näher an der Oberfläche des Versuchsaufbaus liegen.

Danach setzen Sie die LEDs auf den Versuchsaufbau. Die längeren positiven LED-Leitungen müssen zum Chip hin zeigen, egal auf welcher Seite des Versuchsaufbaus sie stehen.

Bringen Sie die Überbrückungskabel an, wie oben gezeigt. Vergessen Sie das Überbrückungskabel, das vom Pin 8 von IC auf der GND-Spalte des Versuchsaufbaus geht, nicht.

Laden Sie später den aufgelisteten Sketch ein wenig, und probieren Sie ihn aus. Jede LED sollte nacheinander leuchten, bis dass alle der LEDs eingeschaltet sind, und anschließend erlöschen alle der LEDs und wiederholt der Zyklus sich.

Code

Als erstes sollten wir diese drei Pins, die wir verwenden werden, definieren. Sie sind die digitalen Ausgänge vom UNO, die an Latch-, Clock- und Data-Pin von 74HC595 angeschlossen werden.

```
int latchPin = 5;
```

```
int clockPin = 6;
```



```
int dataPin = 4;
```

Als nächstes wird eine Variable mit der Bezeichnung von ‚leds‘ festgelegt. Dies wird verwendet, das Modell, in dem LEDs derzeit ein- oder ausgeschaltet werden, zu halten. Daten mit dem Typ ‚byte‘ vertritt die Zahlen, die acht Bits verwenden. Jedes Bit kann entweder ein- oder ausgeschaltet wird, so ist dies ideal, zu verfolgen, welche von unseren acht LEDs ein- oder ausgeschaltet sind.

```
byte leds = 0;
```

Diese ‚Setup‘-Funktion stellt nur die drei Pins, die wir derzeit als digitale Ausgänge verwenden werden, ein.

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

Die ‚Loop‘-Funktion schaltet ursprünglich alle der LEDs aus, indem sie die Variable ‚leds‘ den Wert 0 gibt. Sie ruft dann ‚updateShiftRegister‘, das das ‚leds‘-Modell an das Schieberegister senden wird, auf, so dass alle der LEDs erlöschen werden. Später werden wir behandeln, wie ‚updateShiftRegister‘ funktioniert.

Die Loop-Funktion pausiert für eine halbe Sekunde, und dann beginnt sie von 0 bis 7 zu zählen, indem sie die ‚FOR‘-Schleife und die Variable ‚i‘ verwendet. Jedes Mal verwendet sie die Arduino-Funktion ‚bitSet‘, um das Bit, das die LED in der Variable ‚leds‘ steuert, einzustellen. Sie ruft dann auch ‚updateShiftRegister‘ auf, so dass die LEDs aktualisiert werden, um anzugeben, was in der Variable ‚leds‘ ist.

Es gibt dann eine Verzögerung von einer halben Sekunde, bevor ‚i‘ erhöht wird und bevor die nächste LED leuchtet.

```
void loop()  
{  
  leds = 0;  
  updateShiftRegister();
```

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

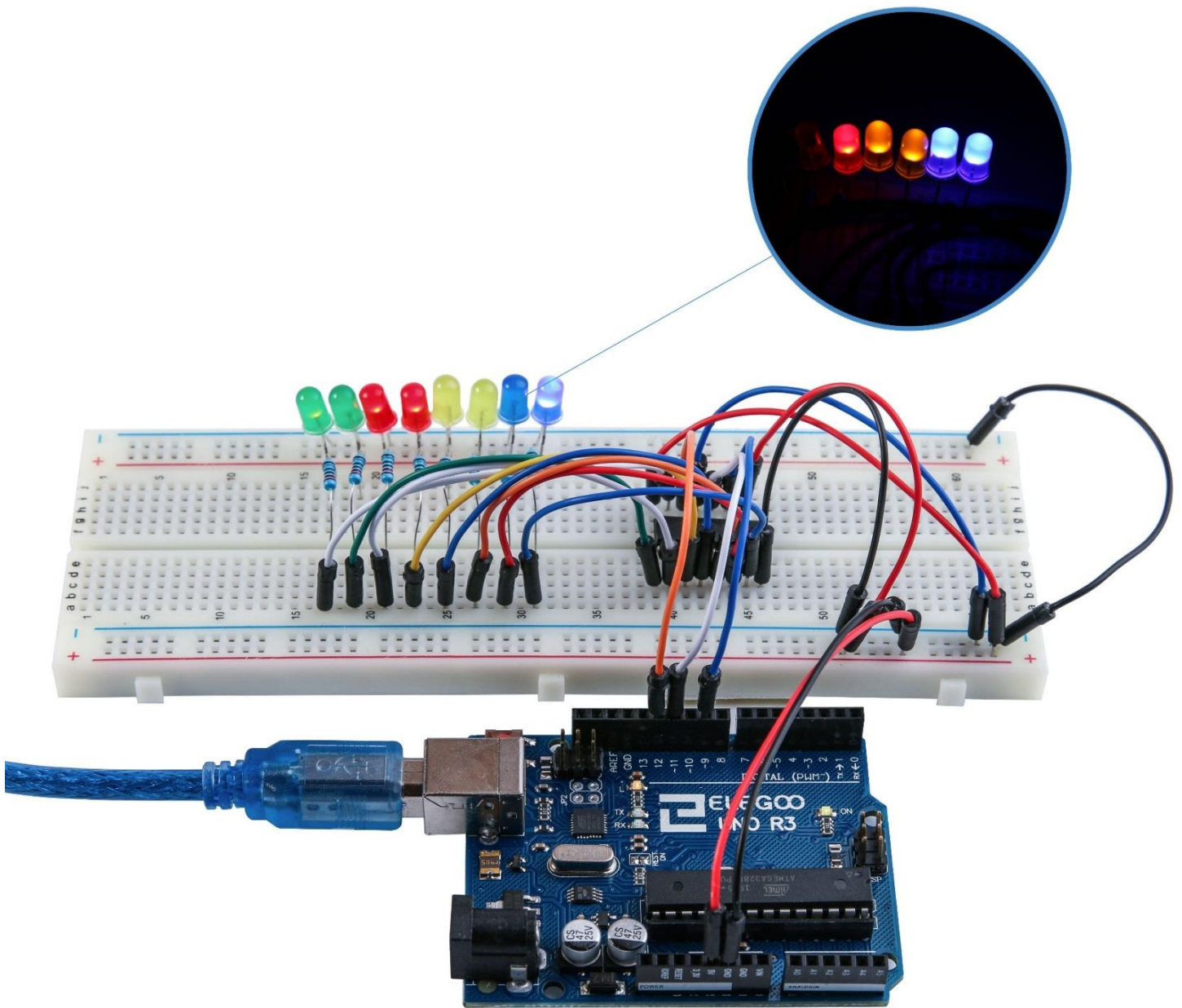
Vor allem stellt die ‚updateShiftRegister‘-Funktion den latchPin auf LOW ein, und dann ruft das UNO die ‚shiftOut‘-Funktion auf, bevor der ‚latchPin‘ wieder auf HIGH gesetzt wird. In diesem Prozess werden vier Parameter verwendet. Die ersten beiden Parameter sind die Pins, die jeweils für Daten und Clock verwendet werden.

Der dritte Parameter gibt an, an welchem Ende der Daten Sie starten möchten. Wir werden mit dem äußersten rechten Bit, das als das ‚Least Significant Bit‘ (LSB) bezeichnet wird, beginnen.

Der letzte Parameter ist die Ist-Daten, die in das Schieberegister eingespeist werden, und in diesem Fall ist er ‚leds‘.

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

Wenn Sie eine der LEDs ausschalten, anstatt einzuschalten, sollten Sie eine ähnliche Arduino-Funktion (bitClear) mit der ‚leds‘-Variable aufrufen. So wird das Bit von ‚leds‘ auf 0 eingestellt, und dann sollten Sie nur es verfolgen, indem Sie ‚updateShiftRegister‘ zur Aktualisierung der tatsächlichen LEDs aufrufen.



Lektion 25: Der Serielle Monitor

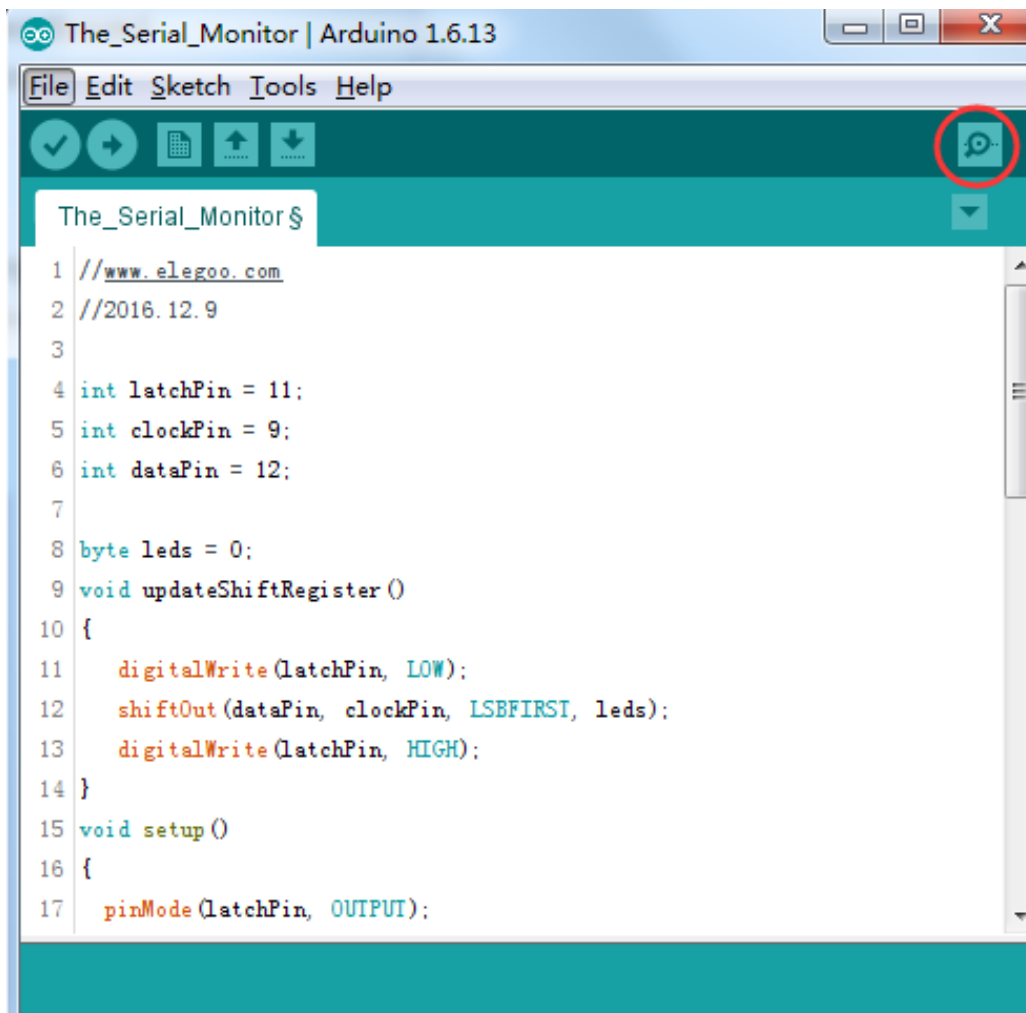
Übersicht

In dieser Lektion werden Sie auf Basis der Lektion 24 die Anlage zur Steuerung der LEDs von Ihrem Computer mit dem seriellen Arduino-Monitor anbringen. Der serielle Monitor ist die ‚Halteverbindung‘ zwischen dem Computer und Ihrem UNO. Er ermöglicht es Ihnen, Textmeldungen zu schicken, zu empfangen und nützlich zu debuggen, sowie das UNO mit einer Tastatur zu steuern! Zum Beispiel können Sie Befehle von Ihrem Computer schicken, um LEDs einzuschalten.

In dieser Lektion werden Sie genau die gleichen Teile und ein ähnliches Layout vom Versuchsaufbau wie Lektion 24 verwenden. Falls Sie dies nicht bereits getan haben, befolgen Sie jetzt Lektion 24 sofort.

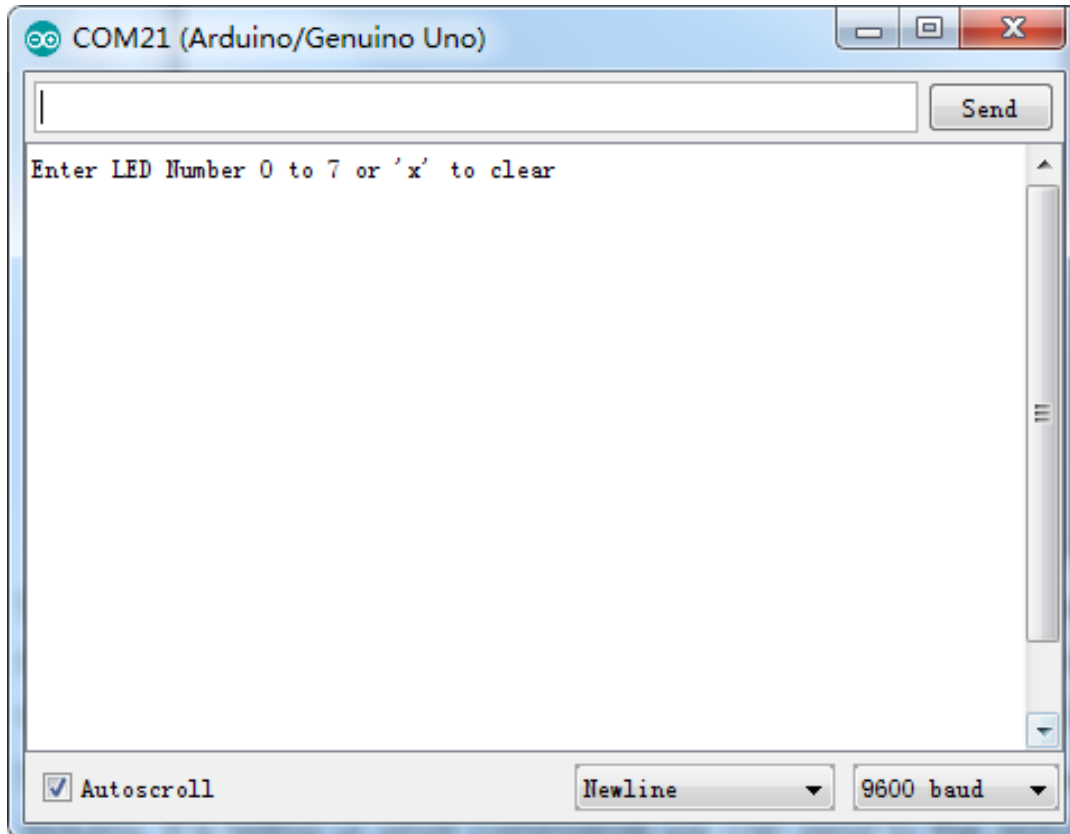
Schritte

Nachdem Sie diesen Sketch auf das UNO hochgeladen haben, klicken Sie die äußerste rechte Taste in der Symbolleiste in Arduino-IDE, wie auf dem Bild unten eingekreist.



Das folgende Fenster wird geöffnet.

Click the Serial Monitor button to turn on the serial monitor. The basics about the serial monitor are introduced in details in Lesson 1.



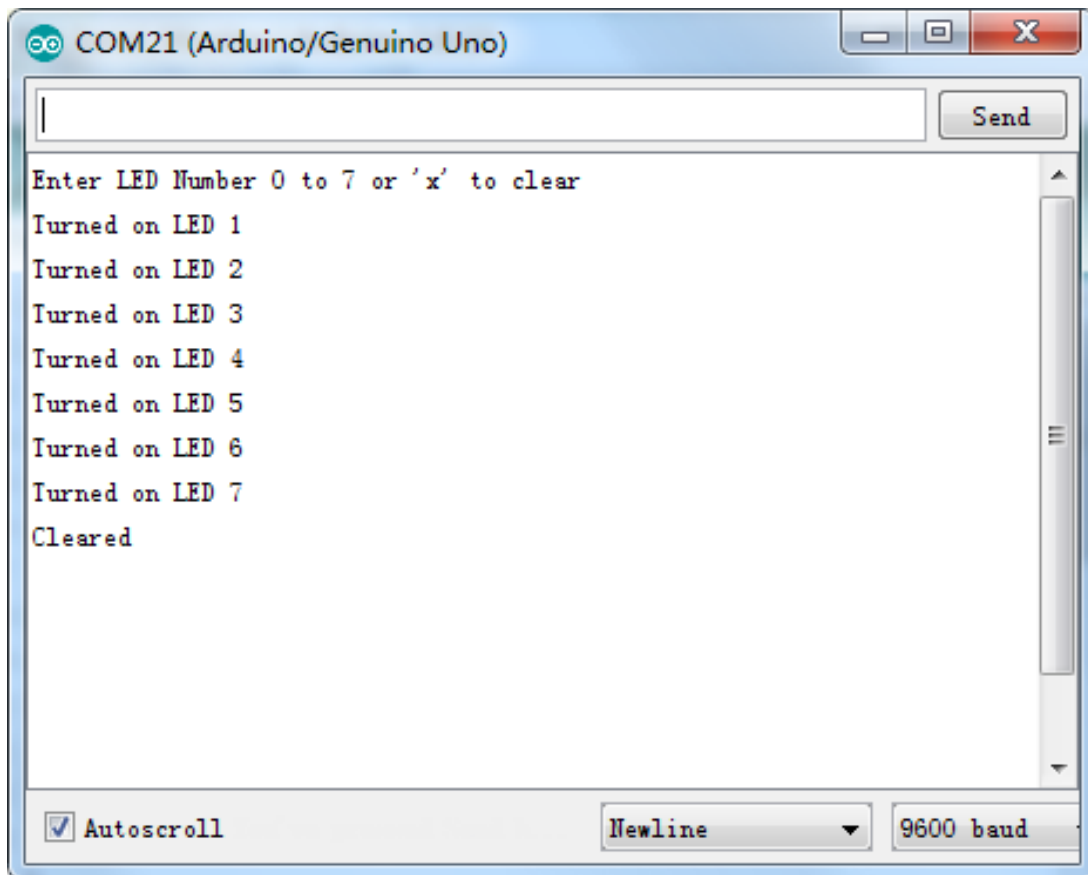
Dieses Fenster wird als den seriellen Monitor bezeichnet, und es ist ein Teil der Arduino-IDE-Software. Seine Aufgabe ist, es Ihnen zu ermöglichen, Meldungen von Ihrem Computer an ein UNO-Board (über USB) zu senden und Meldungen von dem UNO zu empfangen.

Die Meldung „Enter LED Number 0 to 9 or ‚x‘ to clear“ wurde schon durch den Arduino geschickt. Sie gibt uns an, welche Befehle wir an Arduino senden können: Sie können entweder ‚x‘ (um alle LEDs auszuschalten) oder die Zahl der LED, die Sie einschalten möchten (wo 0 die untere LED ist und 1 die nächste LED herauf ist, bis zu 7 für die obere LED) senden.

Versuchen Sie, die folgenden Befehle in den oberen Bereich des seriellen Monitors, der der Höhe der ‚Senden‘-Taste liegt, einzugeben. Drücken Sie die Taste ‚Senden‘, nachdem Sie jedes dieser Zeichen: x 0 3 5 eingegeben haben.

Wenn die LEDs schon ausgeschaltet sind, beeinflusst die Eingabe x gar nicht, aber bei der Eingabe jeder Zahl sollte die entsprechende LED leuchten, und Sie werden eine Bestätigungsmeldung vom

UNO-Board erhalten. Der serielle Monitor erscheint wie unten gezeigt.



Geben Sie x wieder ein, und drücken Sie die Taste ‚Senden‘, um alle LEDs auszuschalten.

Code

Weil Sie erwarten könnten, dass der Sketch auf Basis des Sketches, der in Lektion 6 verwendet wird, steht, werden wir hier nur die neuen Bits decken. Sie werden finden, dass dies ist nützlich, um auf den vollen Sketch in Ihrer Arduino-IDE zu verweisen.

In der ‚Setup‘- Funktion gibt es drei neuen Zeilen am Ende:

```
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 aktiviert die serielle Kommunikation, so können die UNO-Befehle über den USB-Anschluss senden. Der Wert 9600 ist als ‚baud rate‘ der Verbindung bezeichnet. Dies gibt an, wie schnell die Daten gesendet werden. Sie können diesen Wert durch einen höheren Wert ersetzen, aber der Wert des seriellen Monitors vom Arduino muss gleichzeitig in den gleichen Wert gewechselt werden. Wir werden später darüber diskutieren; aber nun lassen Sie ihn bei 9600 liegen.

Die Zeile, die mit ‚while‘ beginnt, stellt sicher, dass es etwas am anderen Ende des USB-Anschlusses, mit dem Arduino vorm Senden der Meldungen sprechen kann, gibt. Andernfalls könnte die Meldung gesendet werden, aber könnte sie nicht angezeigt werden. Diese Zeile ist tatsächlich unnötig, wenn Sie das Programm von Arduino-Leonardo verwenden, weil das Arduino-UNO das Arduino-Board automatisch zurücksetzen kann, wenn Sie den seriellen Monitor öffnen, während dies bei dem Leonardo nicht passiert.

Das Letzte der neuen Zeilen in ‚setup‘ sendet die Meldung, die wir an der Spitze des seriellen Monitors sehen.

Die ‚Loop‘-Funktion ist, wo die 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 innerhalb einer ‚IF‘-Anweisung enthalten. Also, nichts anderes wird anschließend passieren, es sei denn, der Aufruf an die eingebaute Arduino-Funktion ist ‚true‘.

Serial.available() wird auf ‚true‘ zurückkehren, falls die Daten an das UNO gesendet wurde, und falls es gibt Daten, die zur Verarbeitung bereit sind. Eingehende Meldungen werden in einem sogenannten Puffer, und Serial.available() auf ‚true‘ zurückkehrt, wenn dieser Puffer nicht leer ist.

Wenn eine Meldung empfangen wurde, kommt dann der Code in der nächsten Zeile:

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

Dieser Code liest das nächste Zeichen aus dem Puffer, und entfernt es aus dem Puffer. Er verteilt es auch an die Variable ‚ch‘. Die Variable ‚ch‘ ist Typ ‚char‘, der ‚character‘ vertritt. Wie der Name verrät, hält es ein einzelnes Zeichen.

Wenn Sie die Anweisungen in der Eingabeaufforderung an der Spitze des seriellen Monitors verfolgt haben, wird dieses Zeichen entweder eine einstellige Zahl zwischen 0 und 7 oder das Buchstabe ‚x‘ sein.

Die ‚IF‘-Anweisung in der nächsten Zeile überprüft, ob es eine einzelne Zahl ist, wenn ‚ch‘ größer als oder gleich wie das Zeichen ‚0‘ sowie kleiner als oder gleich wie das Zeichen ‚7‘ ist. Es ist ein wenig merkwürdig, Zeichen auf diese Weise zu vergleichen, aber es ist vollkommen akzeptabel.

Jedes Zeichen wird durch eine eindeutige Nummer vertreten, die als seinen ASCII-Wert bezeichnet wird. Dies bedeutet, wenn wir Zeichen unter Verwendung von <= und >= vergleichen, ist es eigentlich die ASCII-Werte, die verglichen werden.

Wenn es den Test besteht, kommt dann der Code in der nächsten Zeile:


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

Nun führen wir arithmetische Funktion auf Zeichen aus! Wir subtrahieren die Ziffer ,0' von irgendeiner Ziffer, die eingegeben wurde. Also, wenn Sie ,0' eingeben, dann wird ,0' - ,0' gleich 0 sein. Wenn Sie ,7' eingeben, dann wird ,7' - ,0' glich die Zahl 7 sein, weil es eigentlich die ASCII-Werte, die in der Subtraktion verwendet werden.

Da wir die Nummer der LED, die wir einschalten möchten, wissen, müssen wir nur dieses Bit in der Variable ,leds' einstellen und das Schieberegister aktualisieren.

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

```
updateShiftRegister();
```

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

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

```
Serial.println(led);
```

In der ersten Zeile wird Serial.print anstatt Serial.println verwendet. Der Unterschied zwischen den beiden liegt darin, dass Serial.print keine neue Zeile nach dem Drucken von irgendwas, das in seinem Parameter steht, startet. Wir verwenden dies in der ersten Zeile, weil wir die Meldung in zwei Teilen drucken: das allgemeine Bit ,Turned on LED' und dann die Zahl der LED.

Die Zahl der LDE ist in einer ,int' Variable gehalten eher, als sie eine Zeichenfolge ist. Serial.print kann entweder eine Zeichenfolge ,int', die in doppelten Anführungszeichen eingeschlossen ist, oder fast jede Art von Variablen halten.

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

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

Wenn ja, werden dann alle dieser LEDs gelöscht und wird eine Bestätigungsmeldung gesendet.

Lektion 26: Fotozelle

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie Lichtintensität mit einem analogen Eingang messen. Aufgrund der Schaltung werden Sie die Lichtmenge verwenden, um die Zahl der zu beleuchtenden LEDs zu steuern.

Die Fotozelle befindet sich an der Unterseite des Versuchsaufbaus, wo die Pot darüber war.

Erforderliche Komponente

- (1) Elegoo UNO R3
- (1) Versuchsaufbau
- (8) LEDs
- (8) 220 Ohm-Widerstände
- (1) 1k Ohm-Widerstände
- (1) 74HC595 IC
- (1) Fotozelle
- (14) M-M Kabel



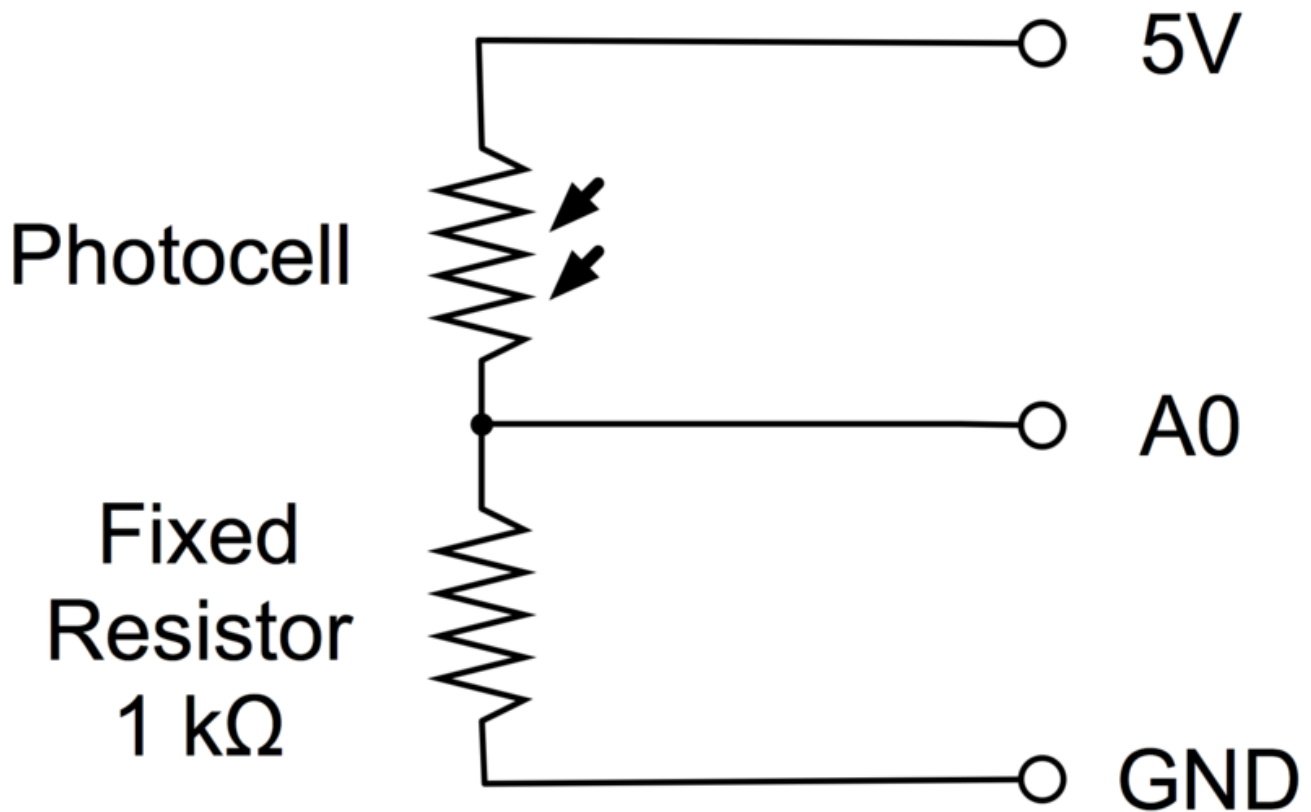
Komponentenanweisung

FOTOZELLE:

Die verwendete Fotozelle ist ein Typ, der als einen lichtabhängigen Widerstand (LDR) genannt. Wie der Name verrät, dass diese Komponente genau als ein Widerstand wirkt, mit der Ausnahme, dass der Widerstand sich abhängig davon, wie viel Licht auf ihn fällt, verändert.

Diese Fotozelle weist einen Widerstand von ca. 50kΩ bei fast vollständiger Dunkelheit und 500Ω bei hellem Licht auf. Um diesen unterschiedlichen Wert des Widerstands in irgendwas, das wir auf einem analogen Eingang vom UNO R3-Board messen können, umzuwandeln, muss er in Spannung umgewandelt werden.

Am einfachsten verbinden Sie die Fotozelle mit einem festen Widerstand.



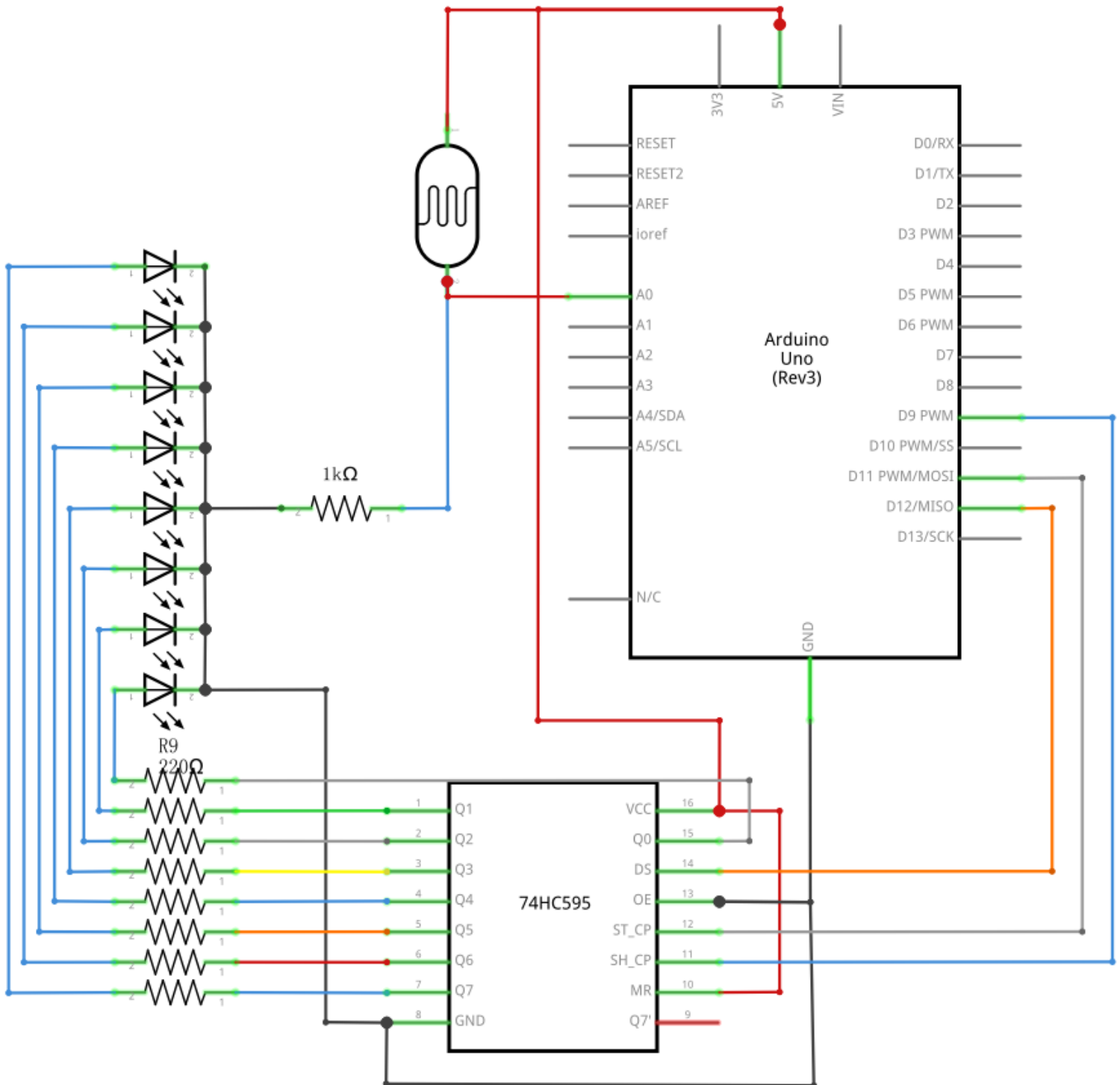
Der Widerstand und die Fotozelle verhalten sich zusammen wie eine Pot. Wenn das Licht sehr hell ist, ist der Widerstand der Fotozelle im Vergleich mit dem festen Widerstand sehr niedrig, und so ist es, als ob die Pot auf Maximum eingeschaltet würde.

Wenn die Fotozelle im matten Licht ist, wird der Widerstand größer als der festen Widerstand 1kΩ sein, so ist es, als ob die Pot auf GND eingeschaltet würde,

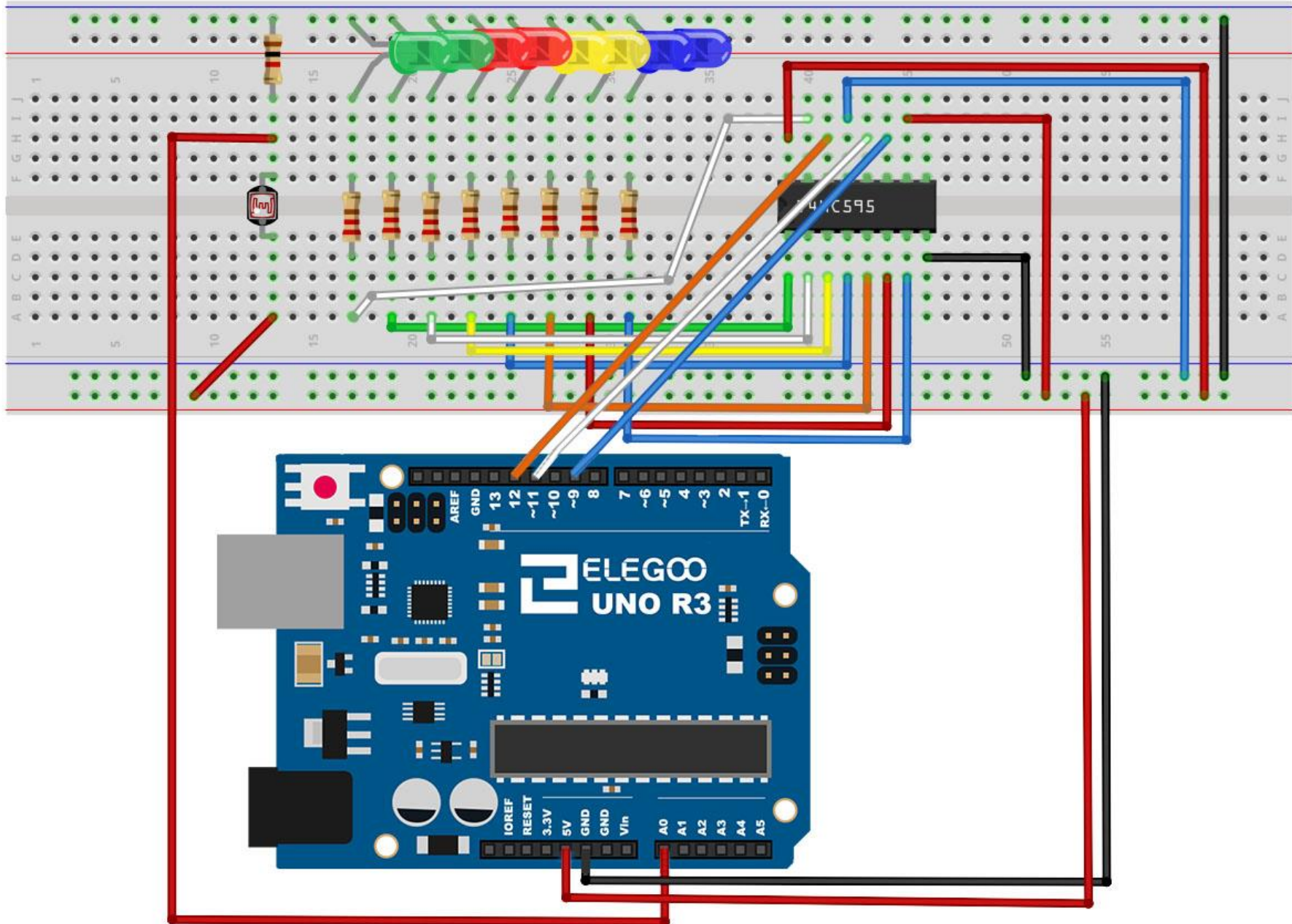
Laden Sie den Sketch, der im nächsten Abschnitt gegeben wird, hoch, und versuchen Sie, die Fotozelle mit ihrem Finger abzudecken, und dann halten Sie sie in der Nähe einer Lichtquelle.

Verbindung

Schema



Schaltplan



Code

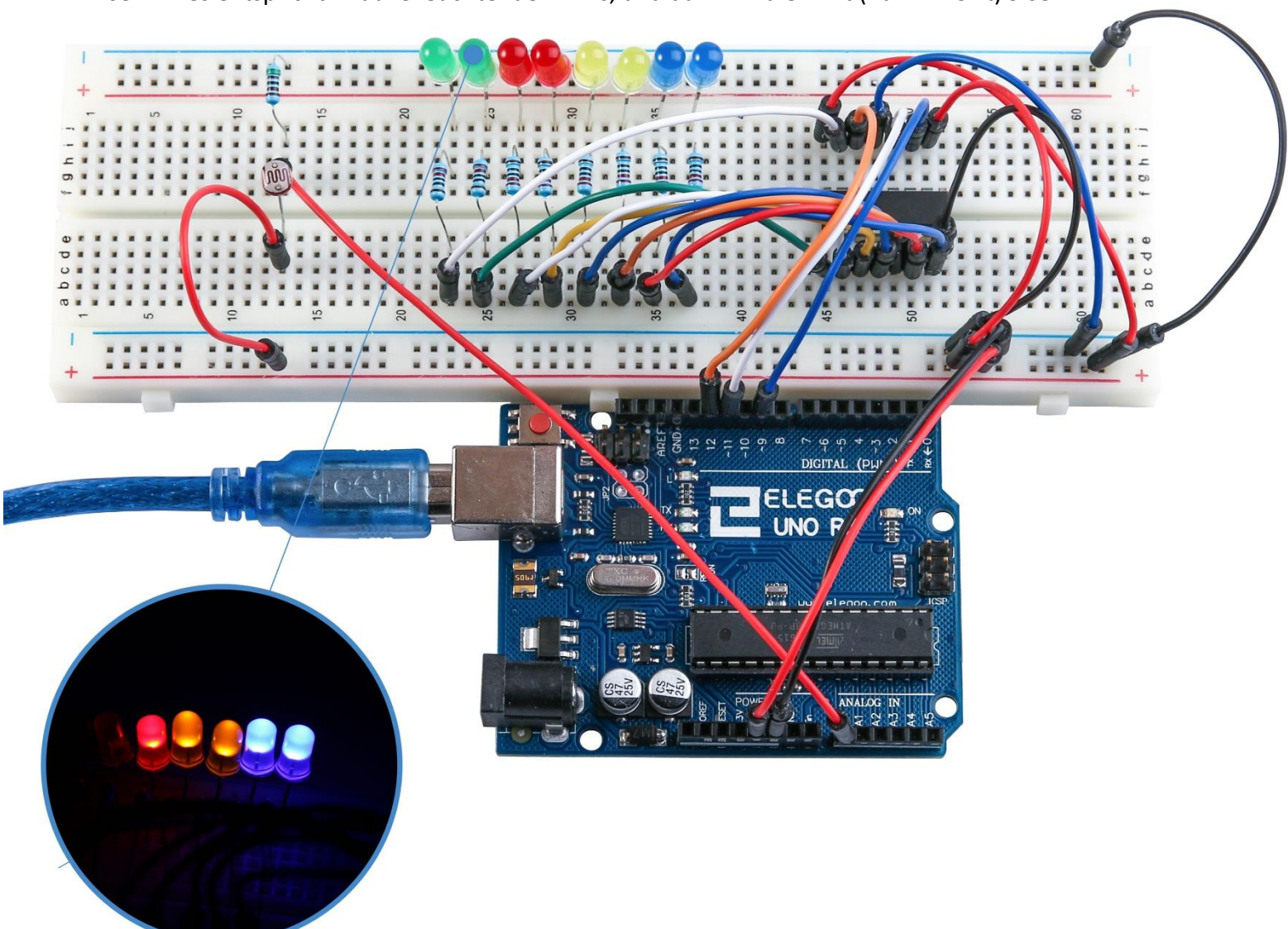
Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

An erster Stelle sollten wir bemerken, dass wir die Bezeichnung vom analogen Pin in ‚lightPin‘ wechseln, anstatt in ‚potPin‘ zu wechseln, da wir nicht mehr eine verbundene Pot haben.

Die einzige andere wesentliche Änderung des Sketches ist die Zeile, in der die Menge der LEDs, die leuchten werden, berechnet wird:

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

Dieses Mal werden wir das Raw-Lesen durch 57 teilen, an statt durch 114 zu teilen. Mit anderen Worten teilen wir es durch die Hälfte so viel, wie wir mit der Pot getan hatten, um es in neun Zonen aufzuteilen, von Leuchten keiner LEDs bis Leuchten aller acht LEDs. Dieser zusätzliche Faktor wird verwendet, den festen Widerstand von $1k\Omega$ zu erklären. Dies bedeutet, wenn die Fozelle einen Widerstand von $1k\Omega$ (gleich wie der feste Widerstand) aufweist, wird das Raw-Lesen $1023 / 2 = 511$ sein. Dies entspricht mit alle leuchtenden LEDs, und dann wird ein Bit (numLEDSLit) 9 sein.



Lektion 27: 74HC595 und Segmentanzeige

Übersicht

Nach dem Lernen von Lektion 24, Lektion 25 und Lektion 26 werden wir das Schieberegister 74HC595 verwenden, um die Segmentanzeige zu kontrollieren. Die Segmentanzeige wird die Zahlen von 9 bis 0 anzeigen.

Erforderliche Komponenten

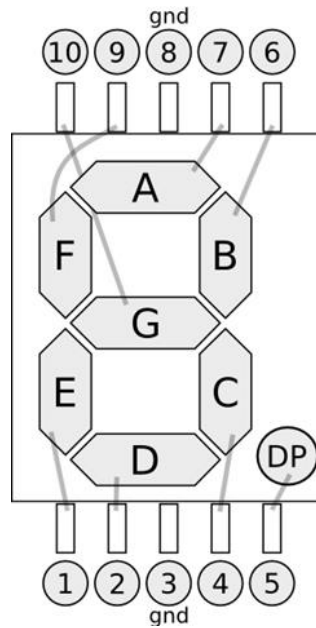
- (1) Elegoo UNO R3
- (2) Versuchsaufbau
- (1) 74HC595
- (1) Segmentanzeige
- (8) 220 Ohm-Widerstand
- (20) M-M Kabel



Komponentenanweisung

Siebensegmentanzeige:

Unten ist das Siebensegment-Pin-Diagramm

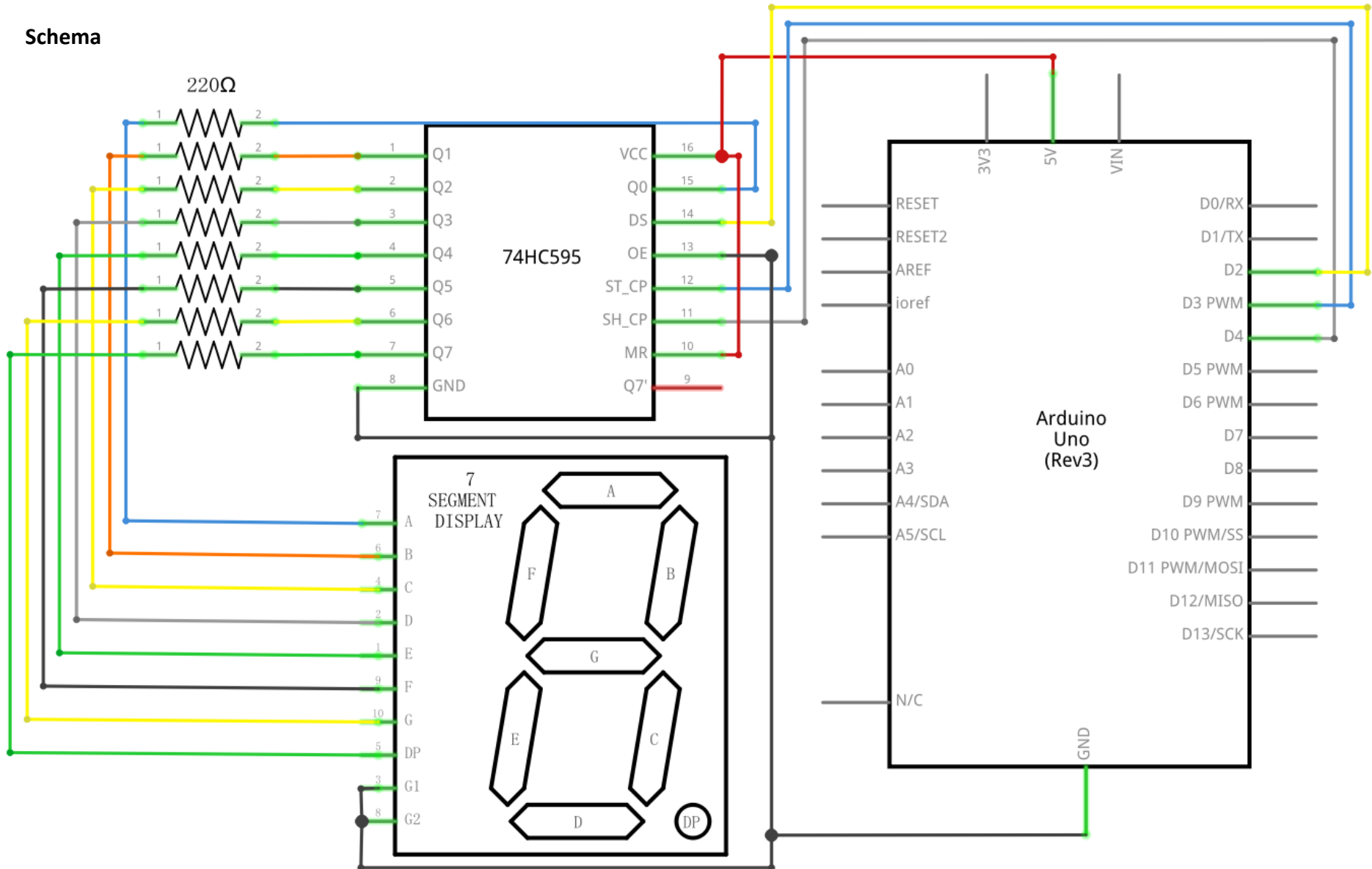


Zehn Ziffern 0-9, die jeweils mit jedem Segment entsprechen, sind wie folgt (die folgende Tabelle gilt für Siebensegmentanzeigergerät mit gemeinsamer Kathode, wenn Sie eine gemeinsame Anode verwenden, sollte die Tabelle alle 100 durch 1 ersetzt werden):

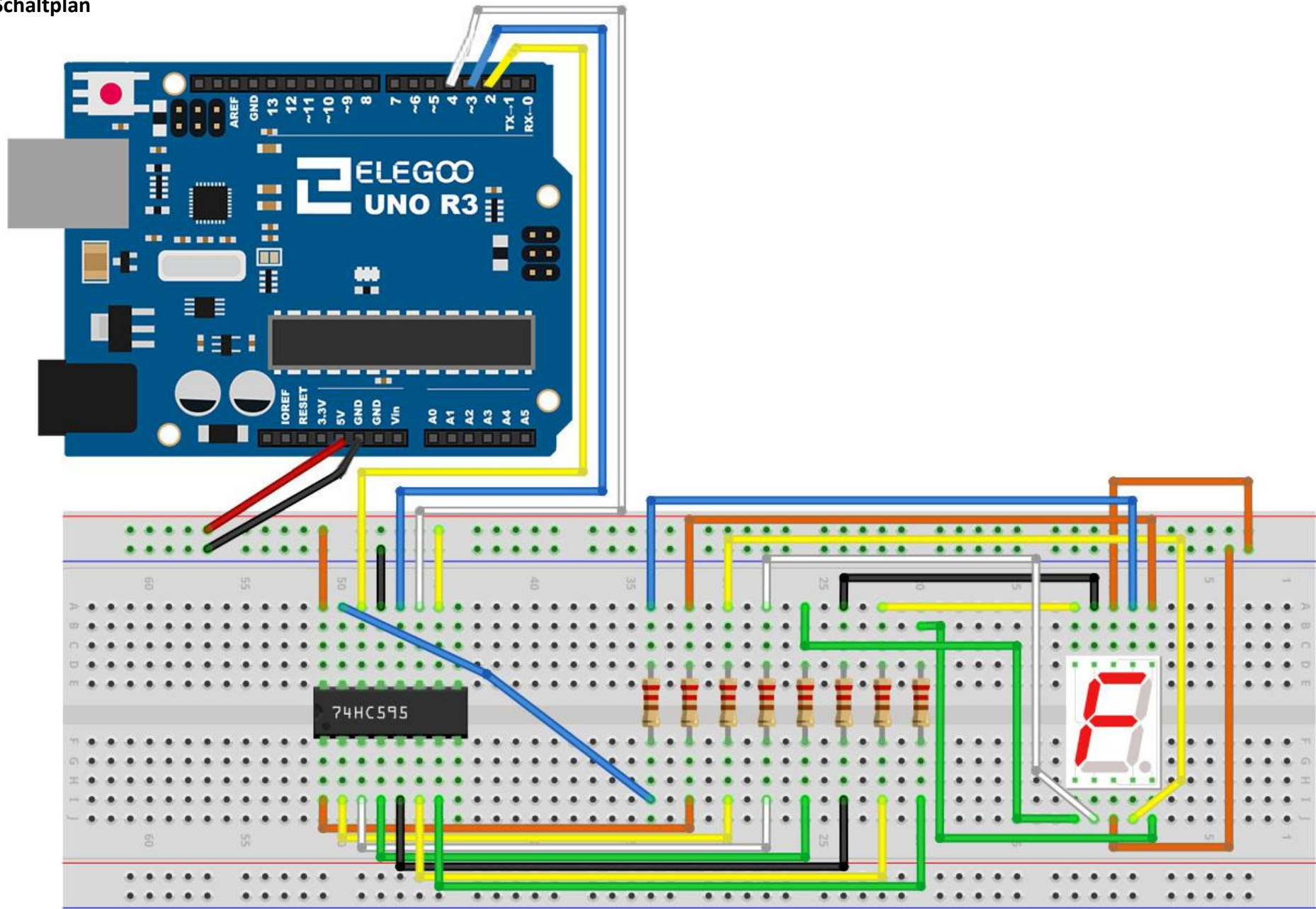
Anzeige	dp	a	b	c	d	e	f	g
0	0	1	1	1	1	1	1	0
1	0	0	1	1	0	0	0	0
2	0	1	1	0	1	1	0	1
3	0	1	1	1	1	0	0	1
4	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1
6	0	1	0	1	1	1	1	1
7	0	1	1	1	0	0	0	0
8	0	1	1	1	1	1	1	1
9	0	1	1	1	1	0	1	1

Verbindung

Schema



Schaltplan



In der folgenden Tabelle wird die Korrespondenztabelle von der Siebensegmentanzeige und dem Pin 74HC595 angezeigt:

Pin 74HC595	Sieben zeigt bemerkenswerten Steuerungspin (Hub)
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 1: Verbinden Sie 74HC595

Erstens verbinden Sie die Leitung mit dem Stromnetz und der Masse:

- Vcc (Pin 16) und MR (Pin 10) werden mit Spannung 5V versorgt
- GND (Pin 8) und OE (Pin 13) werden mit der Masse verbunden

Verbinden Sie DS-, ST_CP- und SH_CP-Pins:

- DS (Pin 14) wird an den Pin 11 vom UNO R3-Board (blaue Leitung) angeschlossen
- ST_CP (Pin 12, Latch-Pin) wird an Pin 8 vom UNO R3-Board (grüne Leitung) angeschlossen
- SH_CP (Pin 11, Clock-Pin) wird an Pin 12 vom UNO R3-Board (gelbe Leitung) angeschlossen

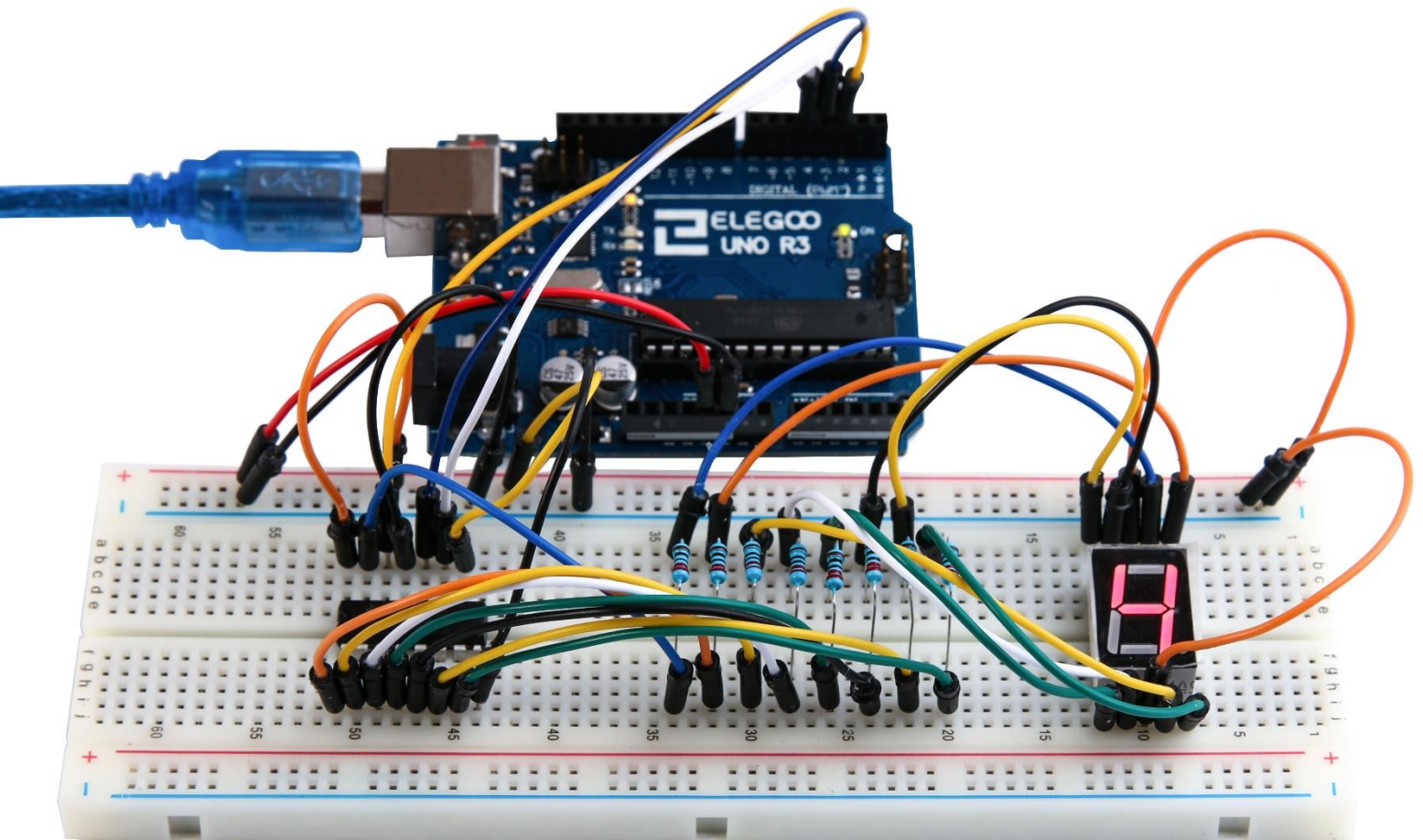
Schritt 2: Verbinden Sie die Siebensegmentanzeige

- Schließen Sie Pins 3 und 8 der Siebensegmentanzeige an das UNO R3-Board GND an (In diesem Beispiel wird die gemeinsame Kathode verwendet, wenn Sie die gemeinsame Anode verwenden, schließen Sie den Pins 3 und 8 an das UNO R3-Board + 5V)

- Gemäß der vorstehenden Tabelle schließen Sie 7 HC595 Q0 ~ Q7 an die Pins (A ~ G und DP), mit denen die Siebensegmentanzeige entspricht, an und dann schließen Sie jeden Fuß an einen 220 Ohm-Widerstand in Reihenschaltung an

Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.



Lektion 28: 4-Stellige 7-Segmentanzeige

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie eine 4-stellige 7-Segmentanzeige verwenden.

Wenn Sie 1-stellige 7-Segmentanzeige mit der gemeinsamen Anode verwenden, schließen Sie den Pin der gemeinsamen Anode an die Stromquelle an; wenn es gemeinsame Kathode ist, schließen Sie den Pin der gemeinsamen Kathode an die GND an.

Wenn Sie 4-stellige 7-Segmentanzeige verwenden, wird der Pin der gemeinsamen Anode oder der gemeinsamen Kathode verwendet, um die angezeigte Ziffer zu kontrollieren. Obwohl nur eine Ziffer arbeitet, ermöglicht der Grundsatz der Persistenz der Vision es Ihnen, alle angezeigten Zahlen zu sehen, aber weil jede der Abtastgeschwindigkeit so schnell ist, bemerken Sie kaum die Intervalle.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Versuchsaufbau
- (1) 74HC595
- (1) 4-stellige 7-Segmentanzeige
- (4) 220 Ohm-Widerstand
- (20) M-M Kabel

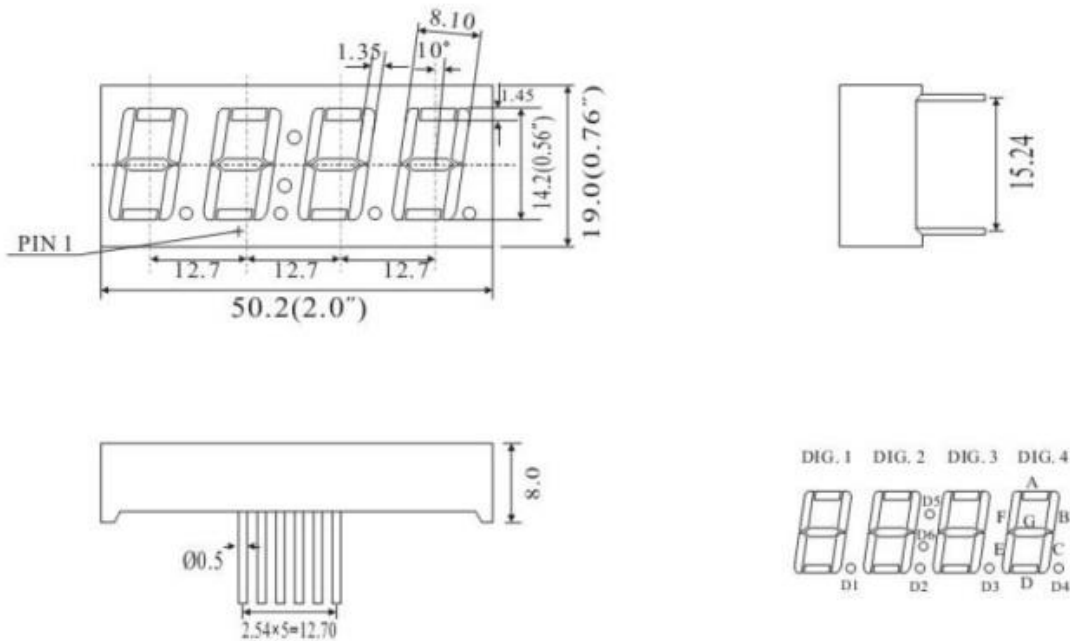
Komponentenanweisung

4-stellige 7-Segmentanzeige:



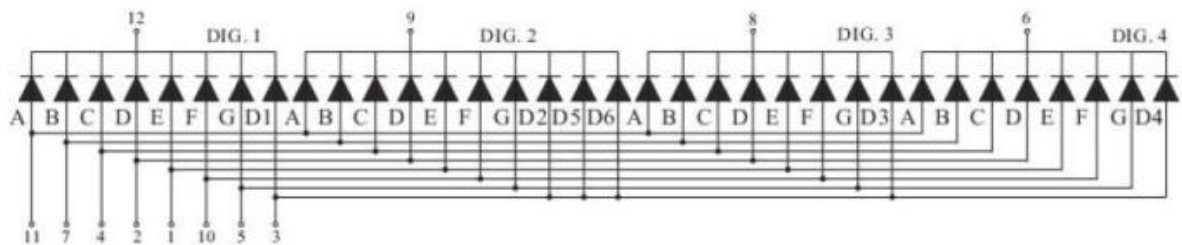
Package Dimensions

CPS05643AB

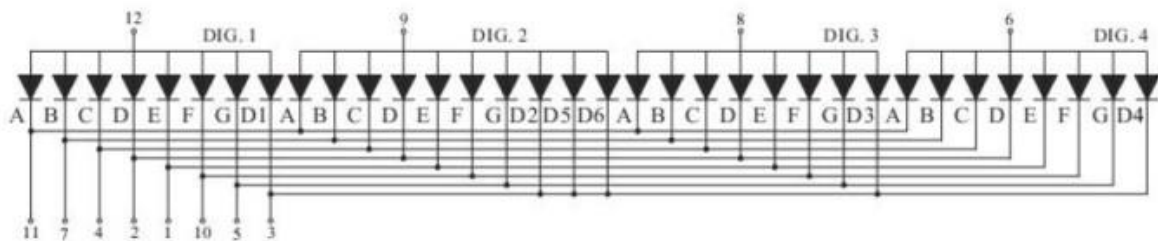


UNIT: MM(INCH) TOLERANCE: ±0.25(0.01")

Internal Circuit Diagram



5643A

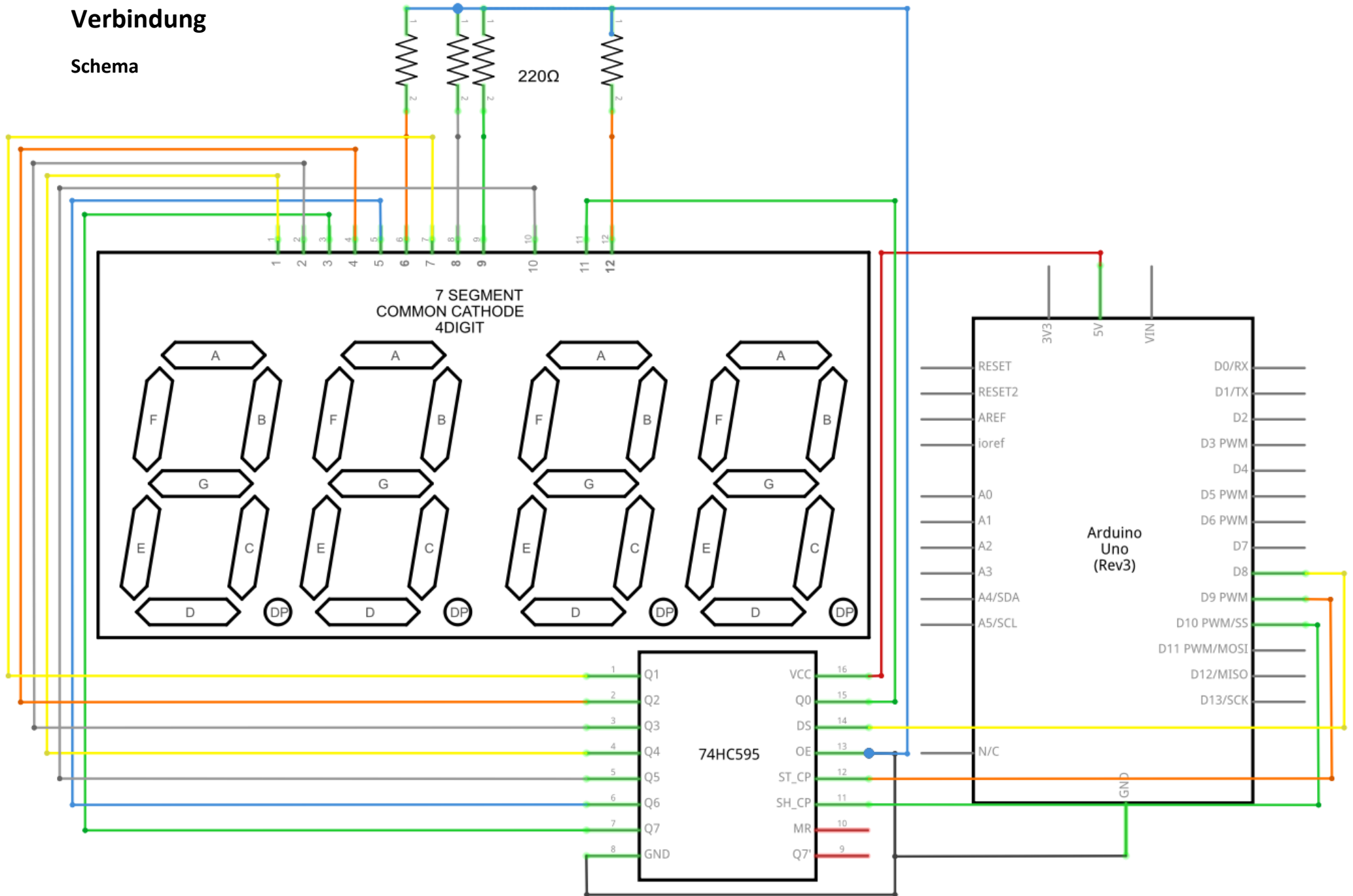


5643B

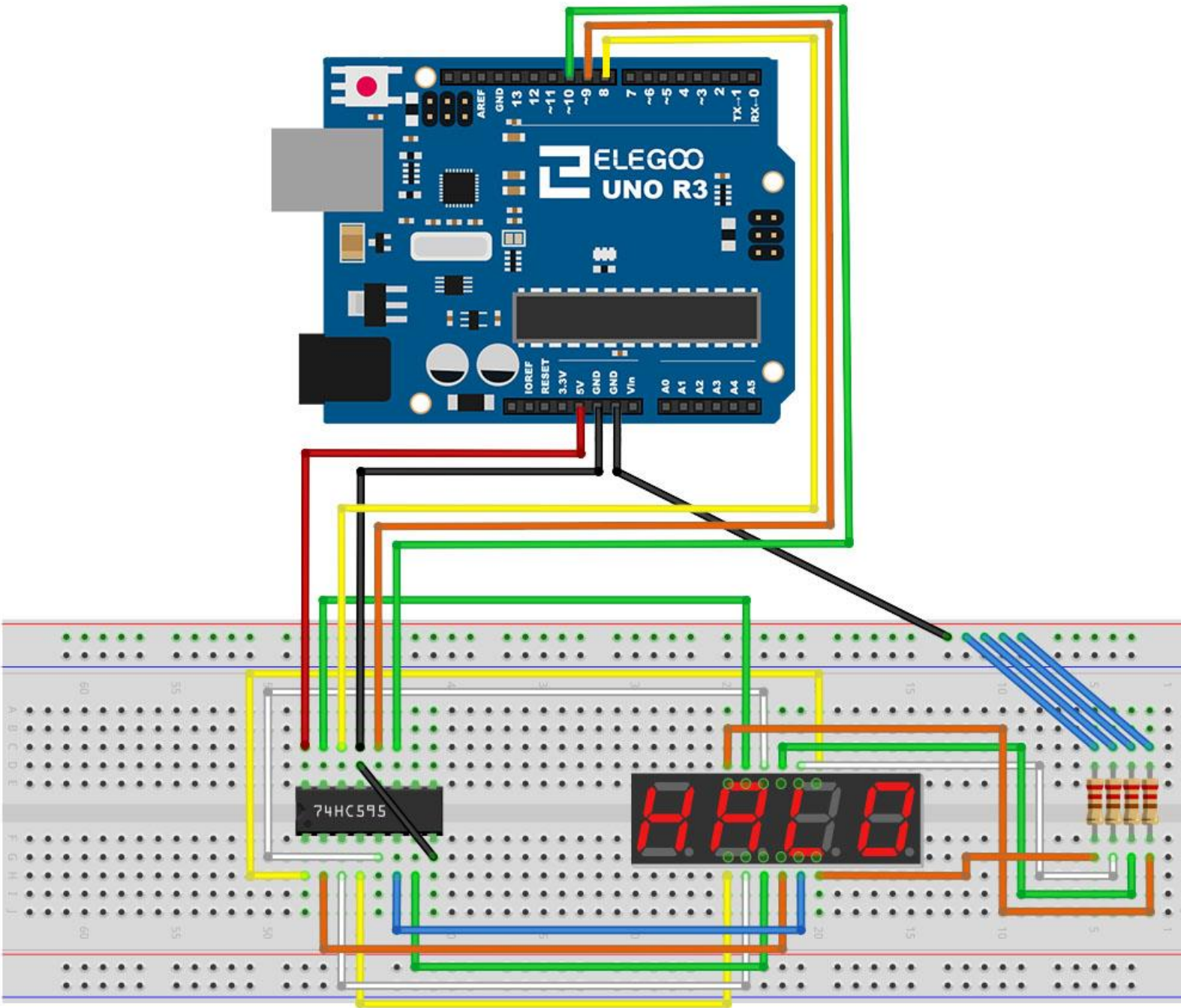
Four Digits Displays Series

Verbindung

Schema

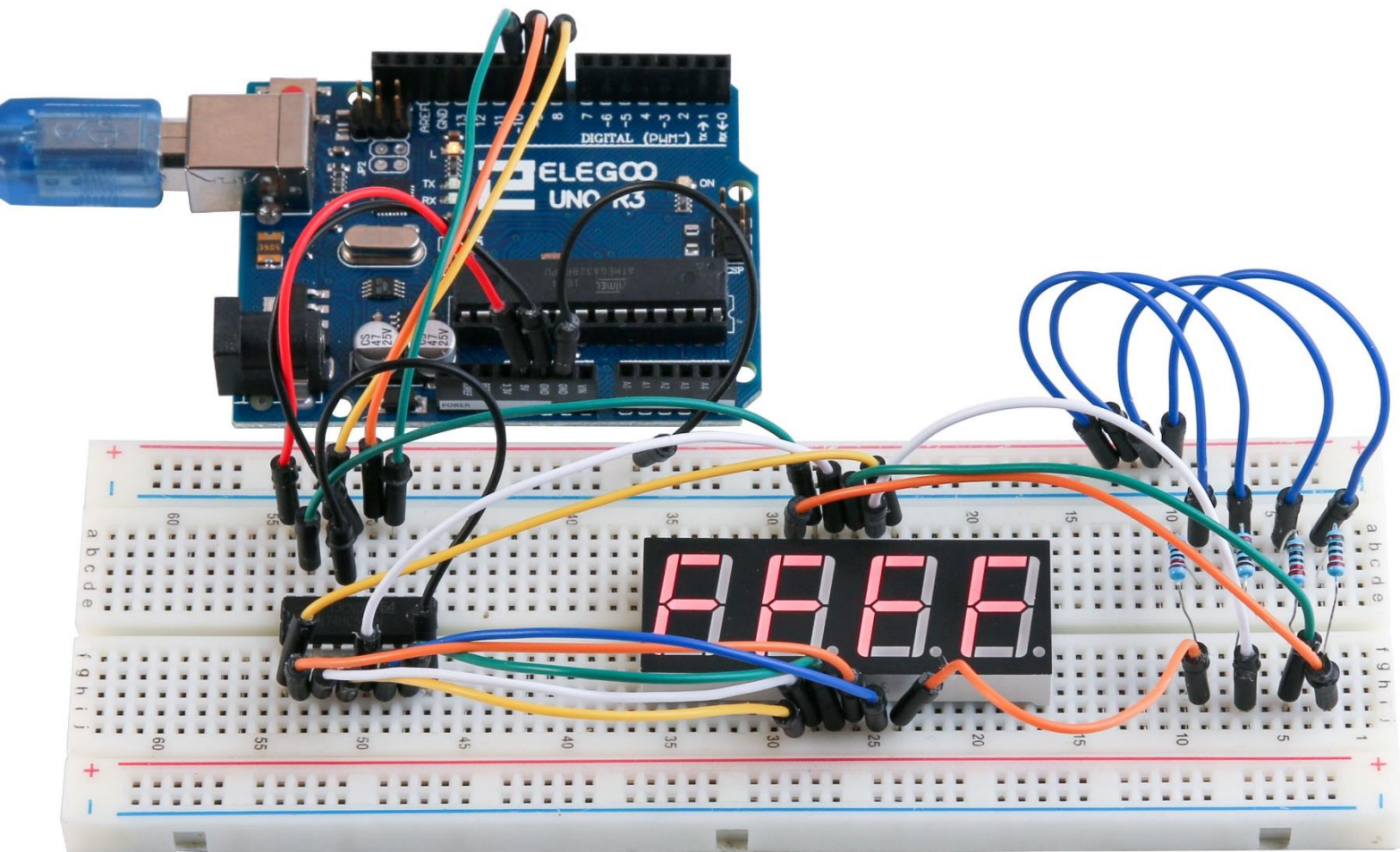


Schaltplan



Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefördert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.



Lektion 29: DC Motoren

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie einen kleinen Gleichstrommotor mit einem UNO R3 und einem Transistor steuern.

Erforderliche Komponenten

- (1) Elego UNO R3
- (1) Versuchsaufbau
- (1) 6 V Gleichstrommotor
- (1) PN2222
- (1) 1N4007
- (1) 220-Ohm-Widerstand
- (3) M-M Kabel

Komponente-Einführung

Steckbrett Netzteil

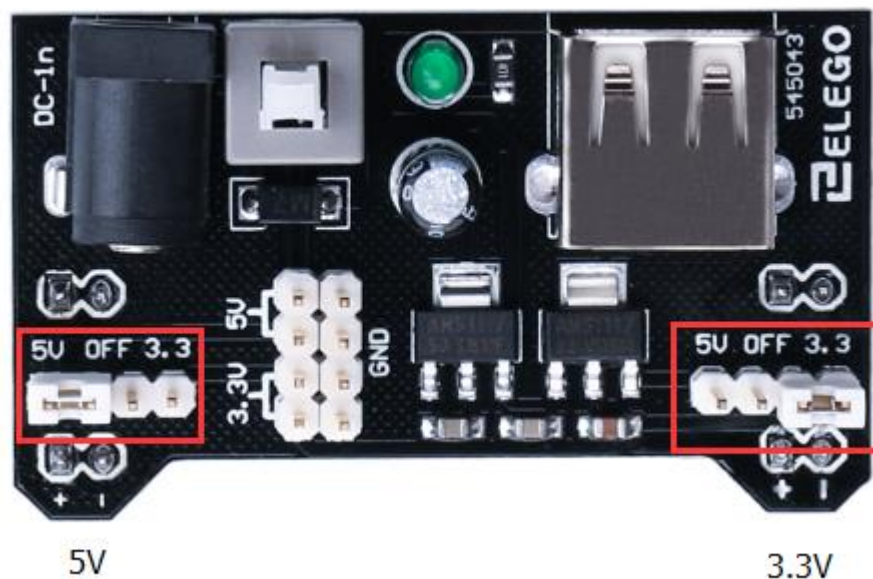
Die kleine DC-Motor wird voraussichtlich mehr Strom als ein UNO-R3-Board verwenden, die digitale Ausgabe direkt verarbeiten kann. Wenn wir den Motor direkt auf eine UNO-R3-Board-Pin zu verbinden versucht, gibt es eine gute Chance, dass es die UNO R3-Board beschädigen könnten. Daher verwenden wir ein Netzteil Modul bietet Stromversorgung.



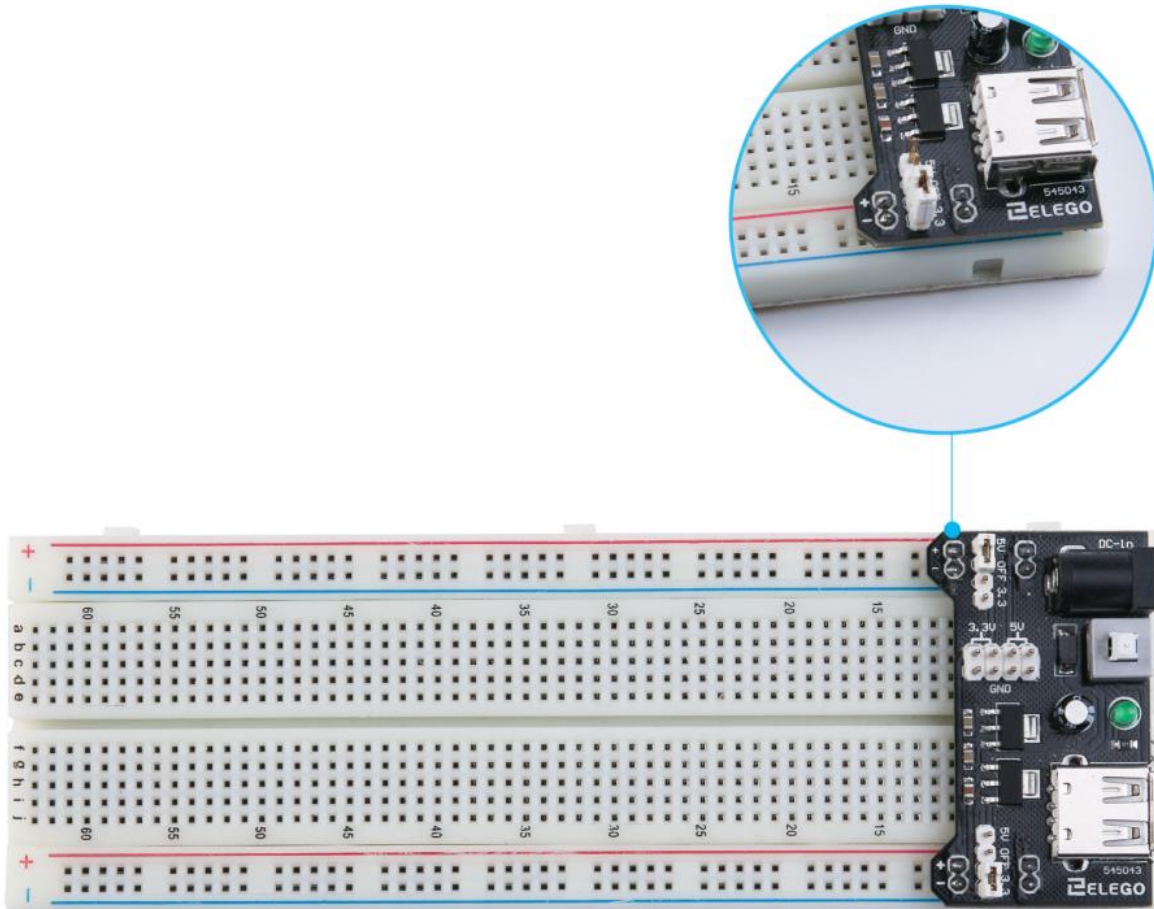
Produkt-Spezifikationen:

- Ein-/Ausshalter Verriegelung
- LED-Betriebsanzeige
- Eingangsspannung: 6,5-9v (DC) über 5,5 x 2,1 mm Stecker
- Ausgangsspannung: 3,3V / 5v
- Max. Ausgangsstrom: 700 mA
- Unabhängige Kontrolle Schiene Ausgabe. 0V, 3,3V, 5v, Steckbrett
- Output-Header-Pins für die bequeme externe Nutzung
- Größe: 2,1 x 1,4 Zoll
- USB-Gerät Anschluss an Bord, externes Netzteil

Einrichten von Ausgangsspannung:



Die linken und rechten Ausgangsspannung kann unabhängig voneinander konfiguriert werden. Um die Ausgangsspannung zu markieren, verschieben Sie Jumper an die entsprechenden Pins. Hinweis: Betriebsanzeige LED und das Steckbrett Stromschienen werden nicht einschalten, wenn beide Jumper in der Stellung "Aus" sind.

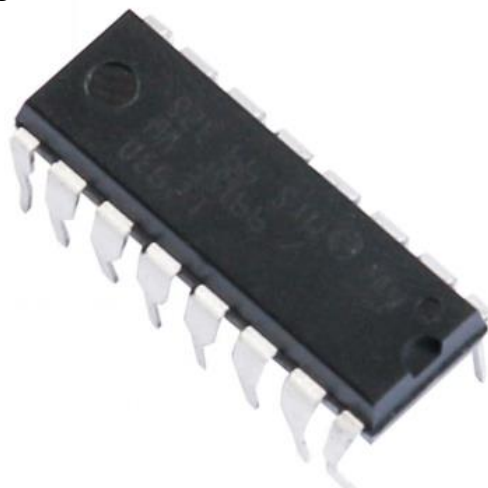


Wichtiger Hinweis:

Stellen Sie sicher, dass Sie das Modul korrekt auf dem Steckbrett ausrichten. Dem negativen Pin (-) am Modul reiht sich mit der blauen Linie (-) auf Steckbrett und, dass den positiven Pin (+) an der roten Linie (+) ausgerichtet. Andernfalls führen Sie versehentlich Umkehrung die Kraft für Ihr Projekt.

L293D

Dies ist ein sehr nützlicher Chip. Es kann zwei Motoren tatsächlich unabhängig voneinander steuern. Verwenden wir nur die Hälfte den Chip in dieser Lektion, die meisten der Pins auf der rechten Seite des Chips sind für die Steuerung eines zweiten Motors.



Produkt-Spezifikationen:

- Mit Unitrode L293 und L293D Produkte jetzt von Texas Instruments
- Weitbereichs Versorgungsspannung: 4,5 V bis 36 V
- Getrennte Eingabe-Logik-Versorgung
- Interne ESD-Schutz
- Thermische Abschaltung
- Hohe Störfestigkeit Eingänge
- Funktionell ähnlich wie SGS L293 und SGS L293D
- Ausgabe der aktuellen 1 A pro Kanal (600 mA für L293D)
- Haupt Leistung aktueller 2 A pro Kanal (1,2 A für L293D)
- Ausgang Klemme Dioden für induktive Überspannungsschutz (L293D)

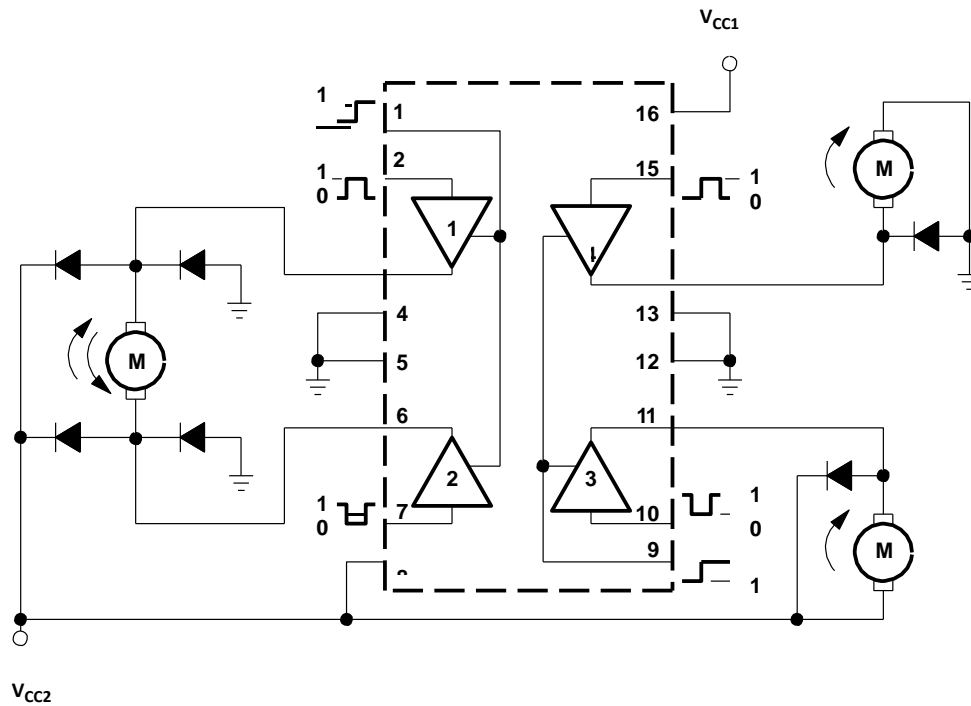


Beschreibung/Bestellinformationen

Die L293 und L293D sind vierfach Hochstrom-Hälfte-H Treiber. Die L293 bietet bidirektionale Laufwerk Ströme von bis zu 1 A bei Spannungen von 4,5 V bis 36 V. Die L293D bietet bidirektionale Laufwerk Ströme von bis zu 600 mA bei Spannungen von 4,5 V bis 36 V. Beide Geräte sollen induktive Lasten wie z. B. Relais, Magnete, dc und Bipolare Schrittmotoren sowie andere High-Strom/Hochspannungs-Lasten in positiv-Supply-Anwendungen zu fahren. Alle Eingänge sind TTL-kompatibel. Jeder Ausgang ist eine komplette Totempfhahl Ansteuerschaltung mit einem Darlington-Transistor-Waschbecken und einer Pseudo-Darlington-Quelle. Die Treiber sind paarweise aktiviert, wobei die Treiber 1 und 2 von 1.2 EN und die Treiber 3 und 4 von 3.4EN aktiviert werden. 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 ausgeschaltet und im hochohmigen Zustand. Mit den richtigen Dateneingängen bildet jedes Treiberpaar einen Voll-H (oder Brücken-) reversiblen Antrieb, der für Magnet- oder Motoranwendungen geeignet ist.

Blockschaltbild



Ich habe satt mit unentzifferbaren Pin-belegung Diagramme innerhalb Datenblätter, so dass ich meine eigene entworfen habe, dass ich denke gibt mehr relevante Informationen.

Es gibt 3 Drähte an den Arduino angeschlossen, 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**

Verwenden Sie diese Pin-belegung:

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

Ja, führen Sie es mit nur einem Motor verbunden.

Arduino-Verbindungen

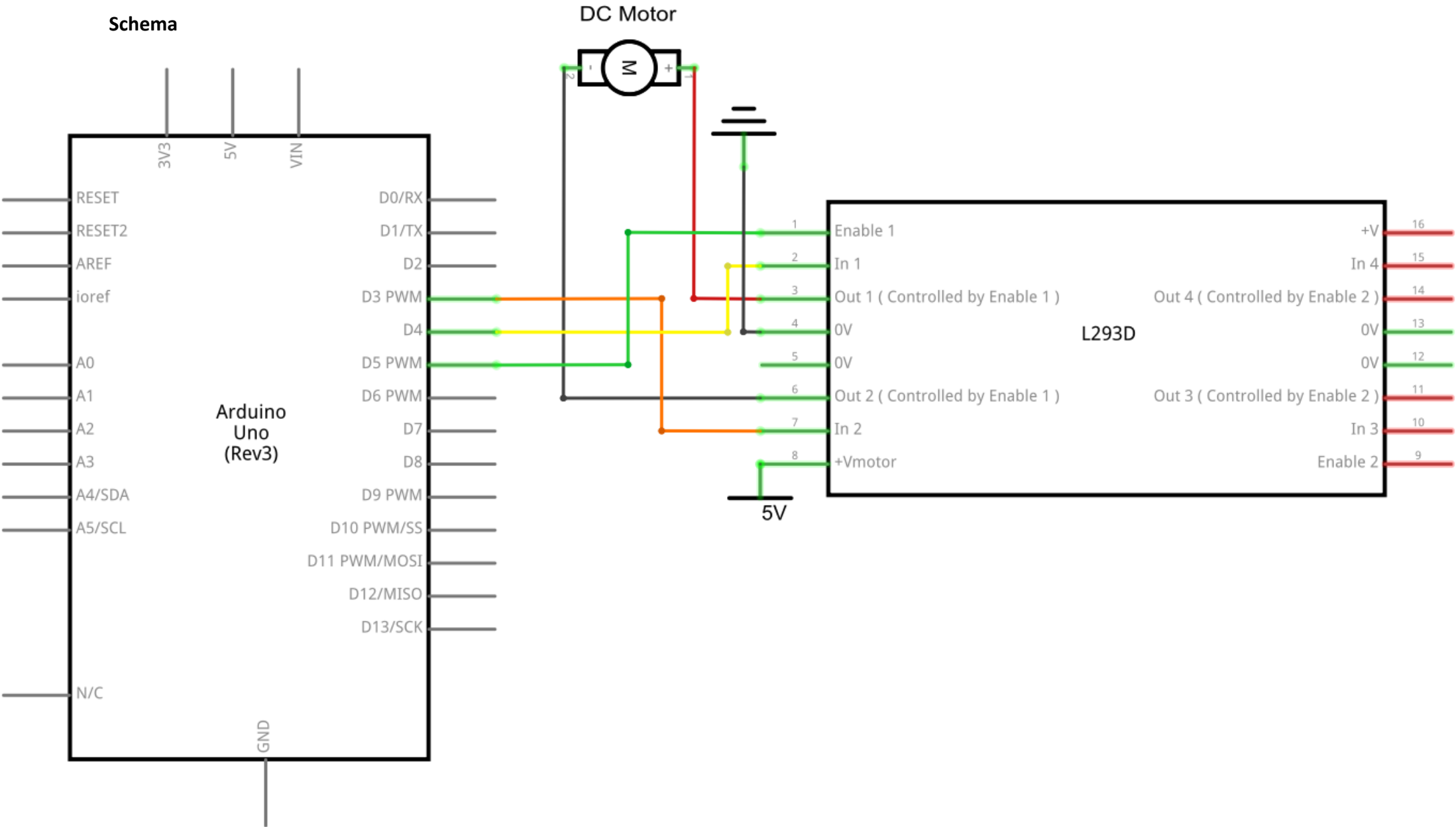
M1-PWM - schließen Sie diesen an einen PWM Pin auf dem Arduino. Sie sind auf die Uno gekennzeichnet, Pin 5 ist ein Beispiel. Ausgegeben Sie eine Ganzzahl zwischen 0 und 255, wo 0 wird ausgeschaltet, 128 ist halber Geschwindigkeit und 255 ist Max Geschwindigkeit.

M1 und M1 Richtung 0/1 1/0 - verbinden diese zwei zu zwei digitale Arduino Pins. Ein Ausgangspin so HOCH und den andern Pin als GERING und der Motor werden in eine Richtung drehen.

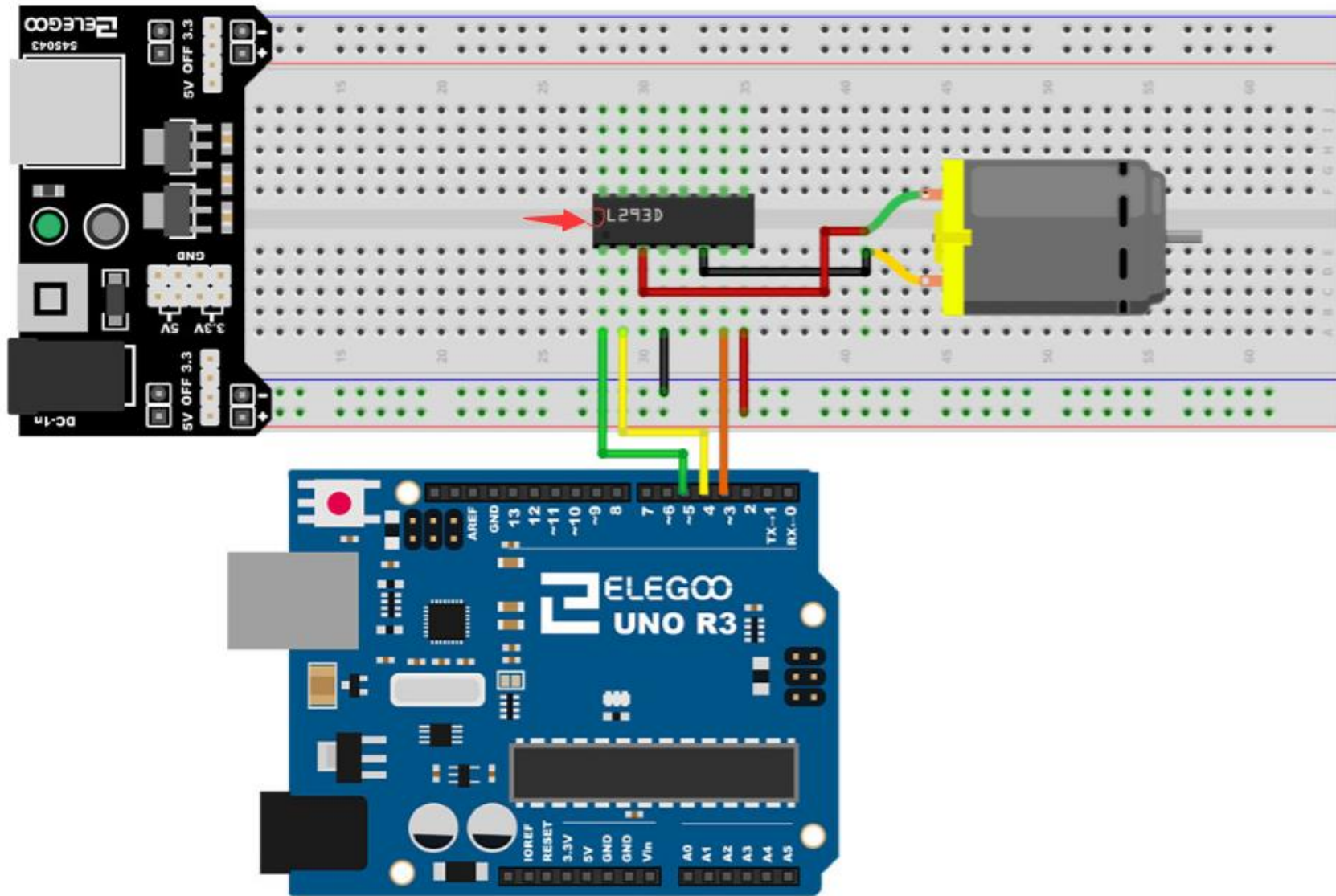
Umkehren der Ausgänge auf NIEDRIG und HOCH, und der Motor in die andere Richtung drehen.

Verbindung

Schema

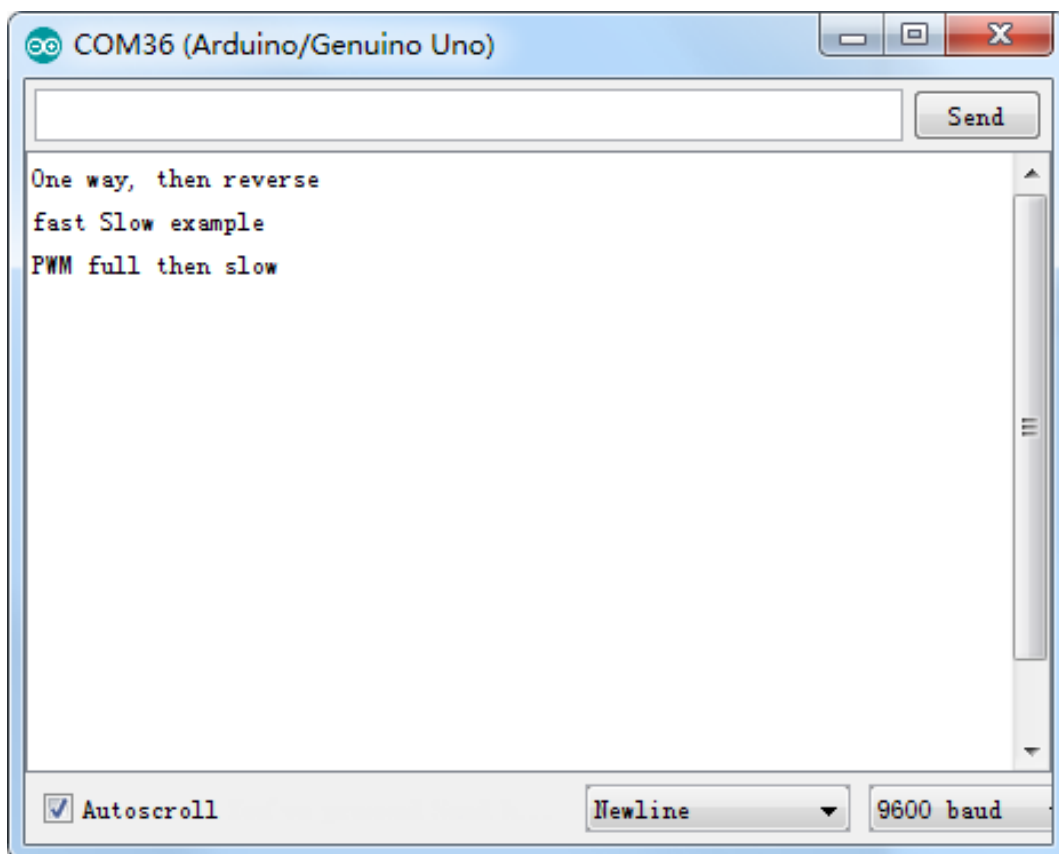


Schaltplan



Der folgende Code verwendet keine separate Stromversorgung (z.B. eine Batterie), es wird stattdessen die 5v Strom aus dem Arduino. Beachten Sie, dass dies ohne die L293D riskant wäre es zu kontrollieren.

Sie sollten niemals einen Motor direkt auf den Arduino verbinden, weil beim Wechsel ein Motor-aus-Sie bekommen eine elektrische Rückmeldung. Mit einem kleinen Motor dies schadet Ihrem Arduino, und mit einem großen Motor, Sie sehen einem interessanten Flammen und Funken Wirkung.

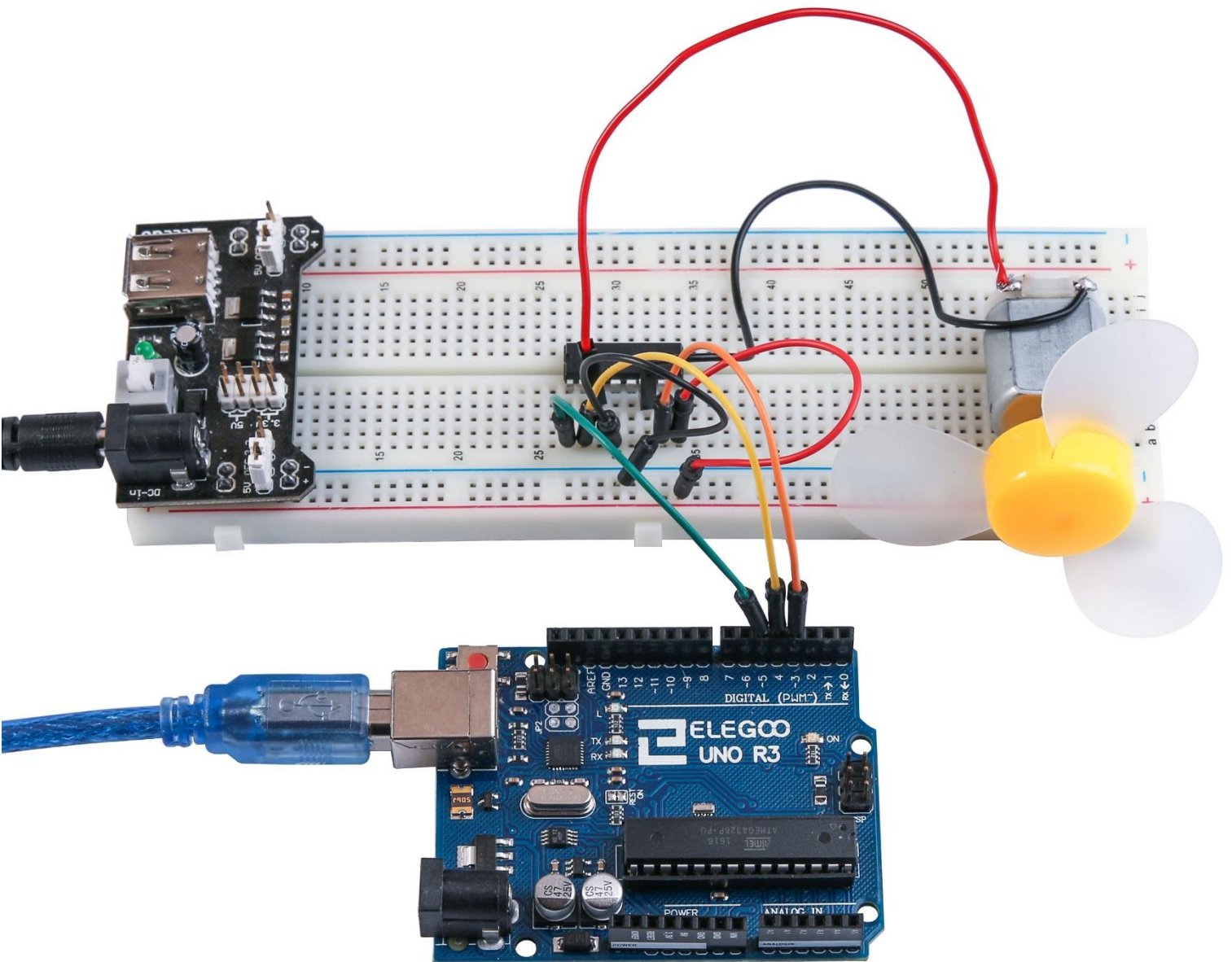


Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Schalten Sie nach Programm laden den Power-Schalter. Der Motor dreht leicht im Uhrzeigersinn und gegen den Uhrzeigersinn für 5-mal. Dann wird es weiterhin dramatisch im Uhrzeigersinn drehen. Nach einer kurzen Pause wird es dramatisch gegen den Uhrzeigersinn drehen. Dann die Controllerplatine

PWM-Signal um den Motor anzutreiben senden wird der Motor wird langsam seine maximale DREHZAHL auf ein Minimum reduzieren und wieder auf das Maximum erhöhen. Schließlich kommt es zum Stillstand für 10 s, bis der nächste Zyklus beginnt.



Lektion 30: Relais

Übersicht

In dieser Lektion werden Sie erfahren, wie Sie ein Relais verwenden.

Erforderliche Komponenten

- (1) Elegoo UNO R3
- (1) Versuchsaufbau
- (1) PN2222
- (1) 1N4007
- (1) 220 Ohm-Widerstand
- (1) 6v DC Motor
- (1) Relais
- (1) Netzteilmodul vom Versuchsaufbau
- (1) 9v-Stromversorgungsadapter
- (8) M-M Kabel



Komponentenanweisung

Relais:

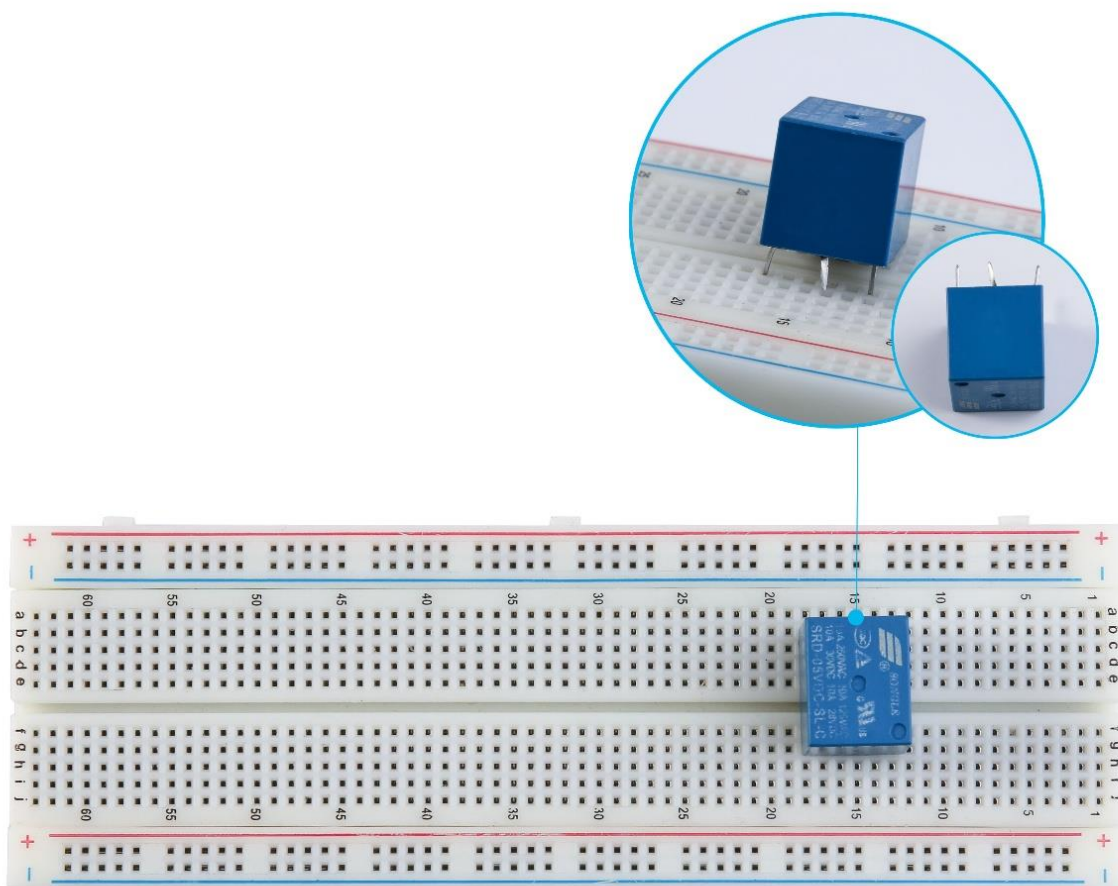
Ein Relais ist ein elektrisch betriebener Schalter. Viele Relais verwenden einen Elektromagnet, um einen Schalter mechanisch zu bedienen, aber andere Betriebsgrundsätze wie in Halbleiterrelais werden auch verwendet. Relais werden verwendet, wo es notwendig ist, eine Schaltung durch ein Niederleistungssignal (mit vollständiger galvanischer Trennung zwischen Controller und kontrollierten Schaltungen) kontrolliert zu werden, oder wo einige Schaltungen durch ein Signal kontrolliert werden müssen. Die ersten Relais wurden in Ferntelegraphenschaltungen als Verstärker verwendet. Sie wiederholten das Signal aus einer Schaltung, und übertrugen es auf eine andere Schaltung wieder. Relais wurden umfassend in Telefonzentralen und frühen Computers für Ausführung der logischen Bedienungen verwendet.

Relais mit dem Typ A, das die hohe Leistung, die einen elektrischen Motor oder andere Lasten direkt kontrolliert, behandeln kann, wird als ein Schütz genannt. Halbleiterrelais kontrollieren Schaltungen

mit unbeweglichen Teilen, stattdessen, dass eine Halbleitervorrichtung verwendet wird, um die Umschaltung auszuführen. Relais mit kalibrierten Betriebseigenschaften, ggf. mit mehreren Betätigungsspulen werden verwendet, um die elektrische Schaltung vor Überlastung oder Störungen zu schützen. In den modernen elektrischen Energiesystemen werden diese Funktionen durch digitale Instrumente mit der Bezeichnung „Schutzrelais“ ausgeführt.

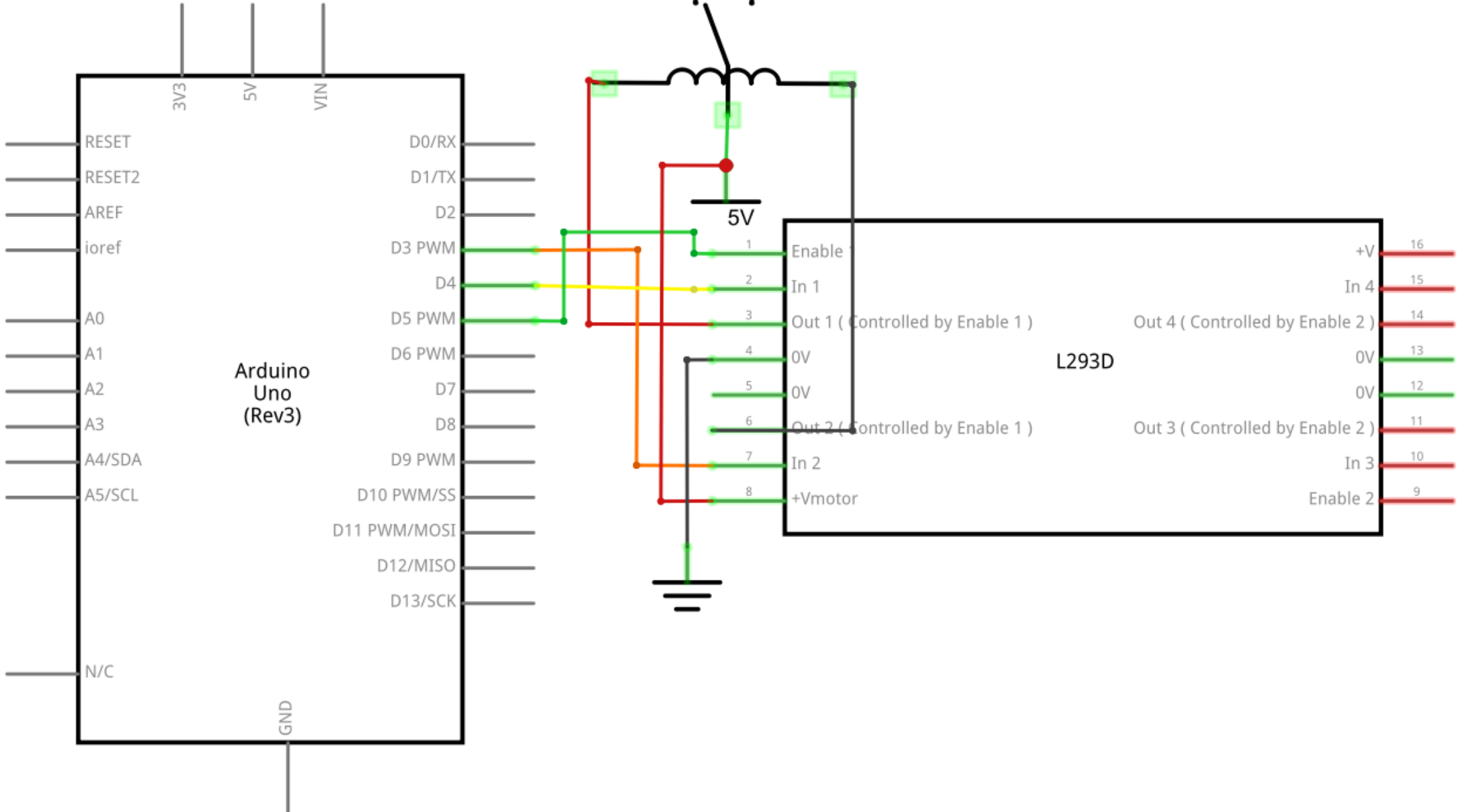
Unten ist das Schema, das darstellt, wie ein Relais mit Arduino (Laden Sie auf arduino.cc herunter) betrieben wird.

Sie können über das Relais in das Bread-Board eingefügt verwechselt werden. Wie die Abbildung unten zeigt musst du einen der Stifte des Relais leicht beugen, dann können Sie es in das Bread-Board einfügen.

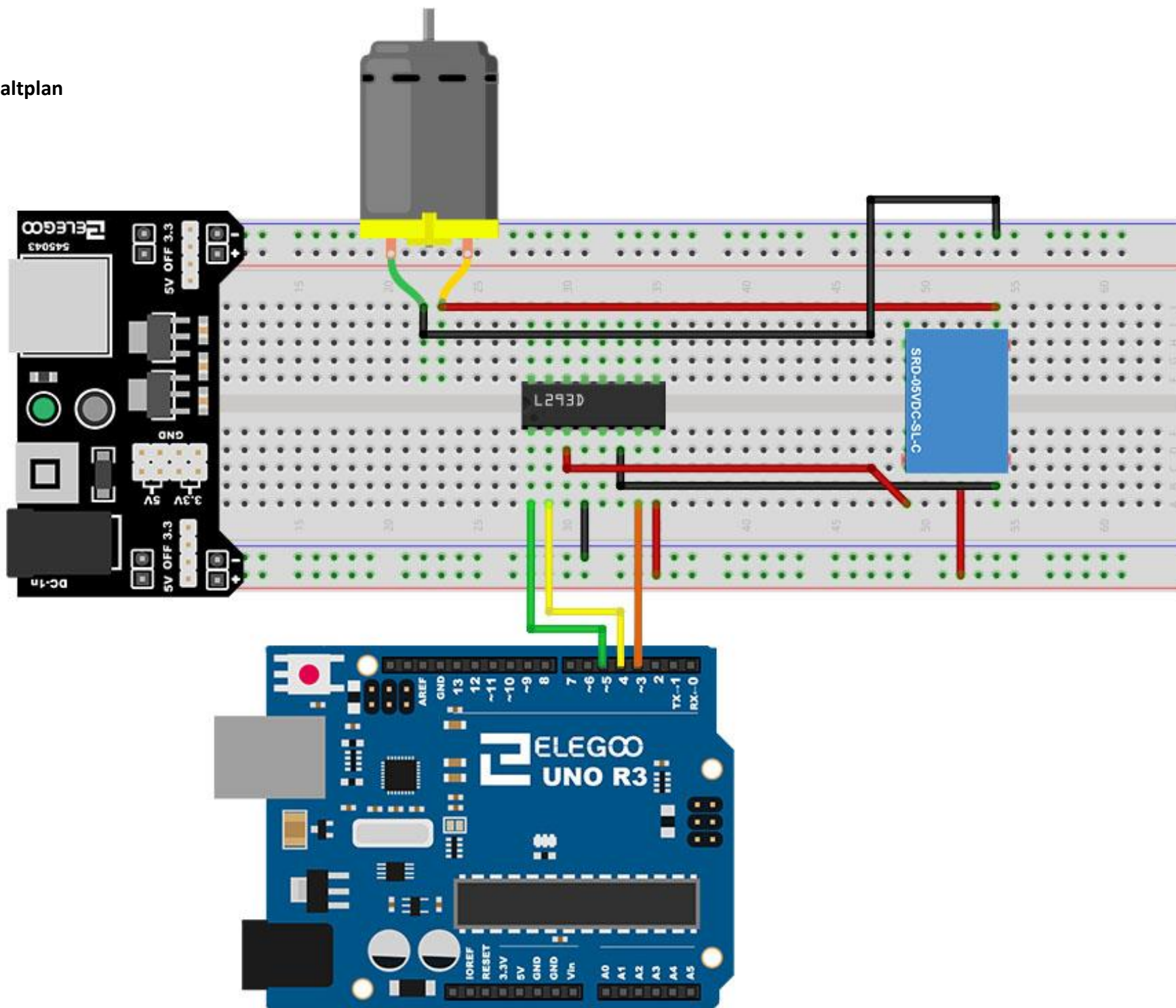


Verbindung

Schema

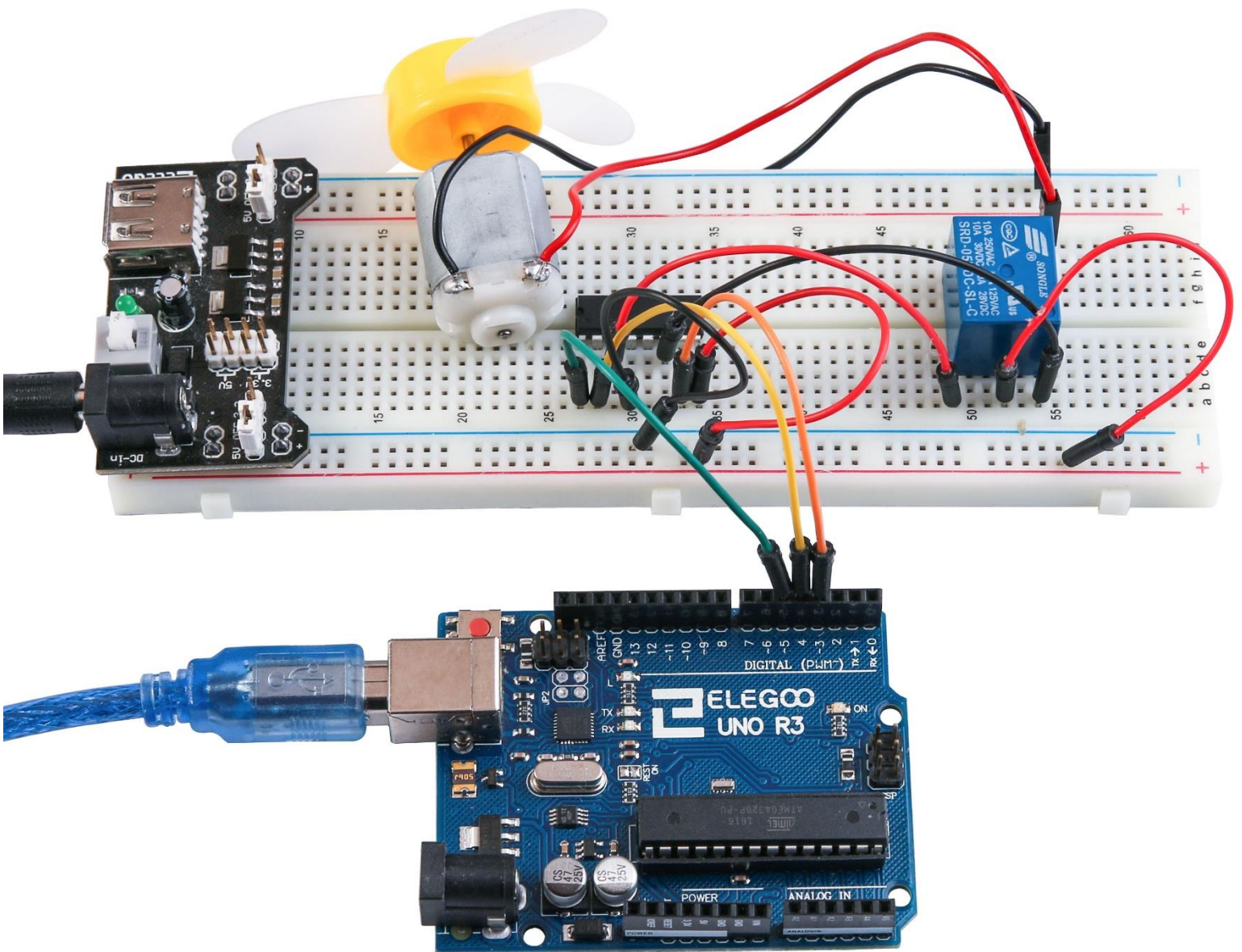


Schaltplan



Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.



Lektion 31: Schrittmotor

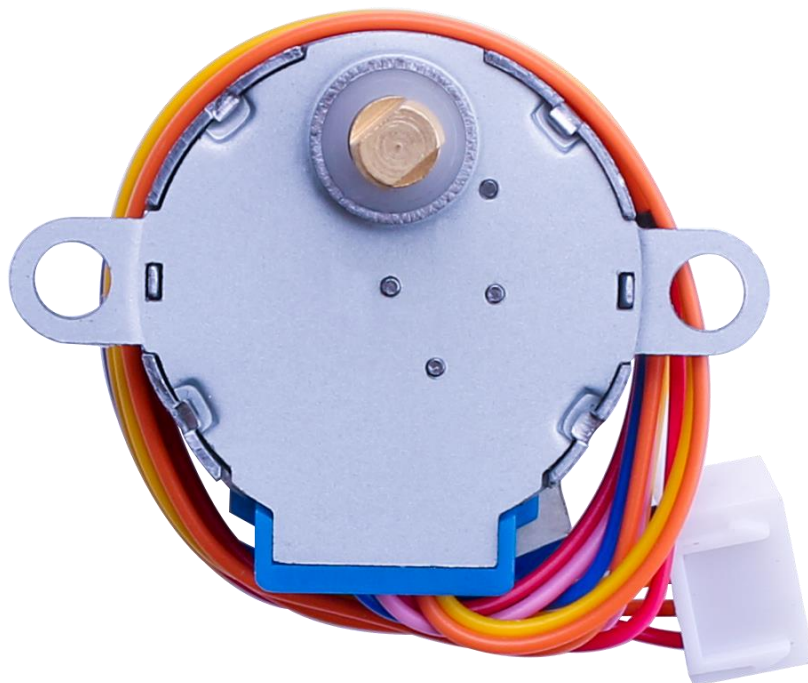
Übersicht

In dieser Lektion werden Sie eine lustige und einfache Art und Weise erfahren, um einen Schrittmotor zu betreiben.

Der Stepper, der von uns verwendet wird, ist mit seiner eigenen Treiberplatine ausgestattet, so kann der Schrittmotor einfach mit unserem UNO verbunden werden.

Erforderliche Komponenten

- (1) Elego UNO R3
- (1) ULN2003 Schrittmotor-Treibermodul
- (1) Schrittmotor
- (9) F-M Kabel



Komponentanweisung

Schrittmotor

Ein Schrittmotor ist ein elektromechanisches Gerät, das elektrische Impulse in diskrete mechanische Bewegungen umsetzt. Die Welle oder Spindel ein Schrittmotor dreht sich in diskreten Schritten, wenn elektrische Befehl Impulse in der richtigen Reihenfolge zugewiesen werden. Die Rotation des Motors hat mehrere direkte Beziehungen zu diesem angewandten Eingangsimpuls. Die Reihenfolge der angewandten Impulse steht in direktem Zusammenhang mit der Drehrichtung Motorwellen. Die Drehzahl der Motorwellen bezieht sich direkt auf die Frequenz der Eingangsimpulse und die Länge der Rotation bezieht sich direkt auf die Anzahl der Eingangsimpulse angewendet. Einer der bedeutendsten Vorteile von einem Schrittmotor ist seine Fähigkeit, in einem Open-Loop-System genau kontrolliert werden. Open-Loop-Regelung bedeutet, dass keine Feedback-Informationen über Position benötigt wird. Diese Art der Steuerung entfällt die Notwendigkeit teure Wahrnehmung und Feedback Geräte wie optischen Encodern. Ihre Position ist bekannt durch die Einlaufhöhe Pulse Überblick zu behalten.

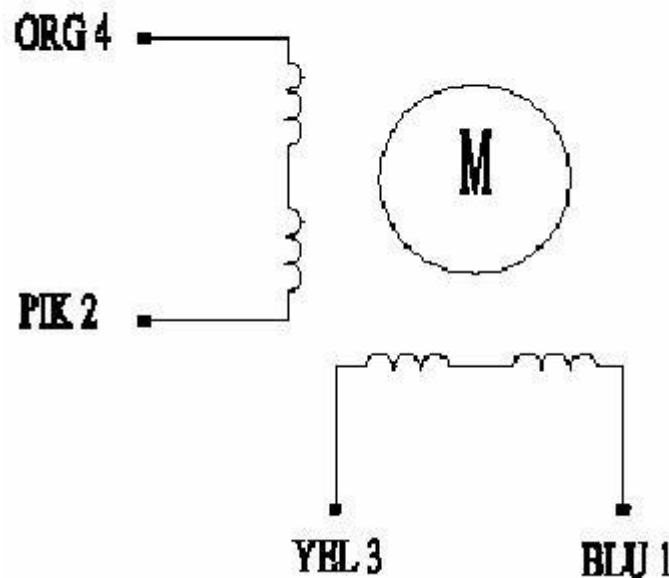
Schrittmotor 28BYJ-48-Parameter

- Modell: 28BYJ-48
- Nennspannung: 5VDC
- Anzahl der Phase: 4
- Geschwindigkeitsänderung: 1/64
- Schrittlänge Winkel: 5,625° /64
- Frequenz: 100Hz
- DC-Widerstand: 50 Ω \pm 7%(25°C)
- Leerlauf-Zugkraft Frequenz: > 600Hz
- Leerlauf Auszug Frequenz: > 1000Hz
- Zugkraft Drehmoment >34,3 mN.m (120Hz)
- Selbstpositionierende Drehmoment >34,3 mN.m
- Reibungsmoment: 600-1200 gf.cm
- Ziehen in Drehmoment: 300 gf.cm
- Isolierte Widerstand >10M Ω (500V)
- Insulated elektrische Spannung: 600VAC/1mA/1s

- Isolierungsgrad: A
- Anstieg der Temperatur <40K(120Hz)
- Lärm <35dB (120Hz, Ohne Last, 10cm)

Einlage-Schaltungen

WIRING DIAGRAM



Der bipolare Schrittmotor hat in der Regel vier Drähte aus ihm heraus. Im Gegensatz zu unipolaren Steppern haben bipolar Stepper keine gemeinsamen Zentrum Verbindung. Sie haben zwei unabhängige Sätze von Spulen statt. Sie können diese von unipolaren Steppern unterscheiden, durch Messung des Widerstandes zwischen den Drähten. Finden Sie zwei Paare von Drähten mit gleicher Widerstand. Hast du die Leitungen des Messgeräts an zwei Drähte, die nicht angeschlossen sollte angeschlossen (also nicht verbunden mit der gleichen Spule), unendlichen Widerstand (oder keine Kontinuität) angezeigt werden.

ULN2003 Treiber-Board



Product Description

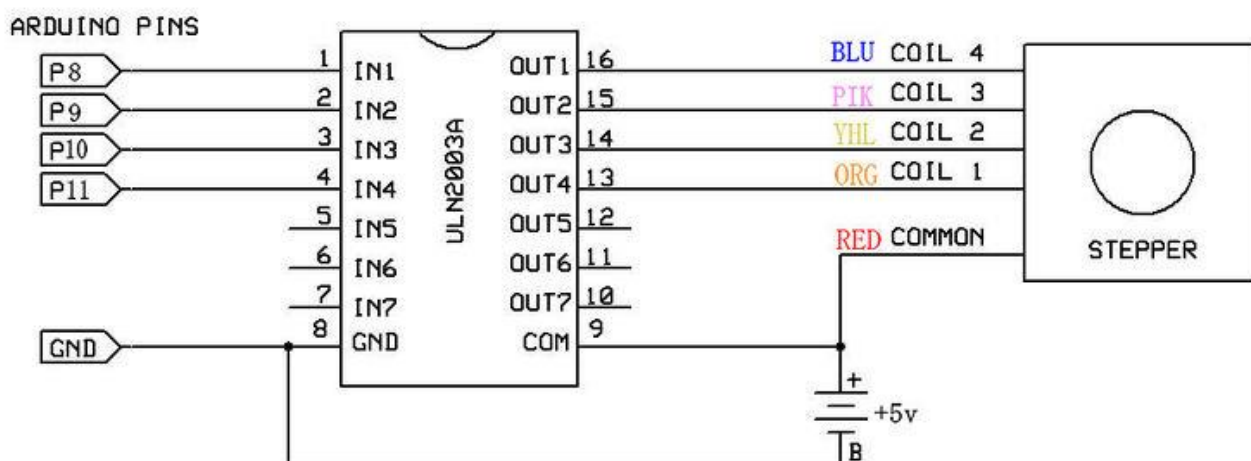
- o Größe: 42mmx30mm
- o Verwenden Sie ULN2003 Treiber-Chip, 500mA
- o A. B. C. D LED zeigt die vier-Phasen-Schrittmotor Motor technisch einwandfreiem Zustand.
- o Weiße Buchse ist der vier-Phasen-Schrittmotor Motor Standard-Buchse.
- o Kraft-Pins sind getrennt.
- o Wir hielten den Rest-Pins des ULN2003 Chips für Ihre weitere Prototypentwicklung.

Am einfachsten Schnittstellen zwischen einem unipolaren Schrittmotor mit Arduino ist einem Ausbrechen für ULN2003A Transistor Array Chip zu verwenden. Die ULN2003A enthält sieben Darlington-Transistor-Treiber und ist so etwas wie mit sieben TIP120 Transistoren in einem Paket. Die ULN2003A kann bis zu 500 mA pro Kanal und verfügt über einem internen Spannungsabfall über 1V wenn diese Option aktiviert. Es enthält auch interne Klemme Dioden um Spannungsspitzen zu zerstreuen, wenn induktive Lasten zu fahren. Um den Schrittmotor zu steuern, gelten Sie jeweils der Spulen in einer bestimmten Reihenfolge Spannung.

Die Sequenz würde wie folgt gehen:

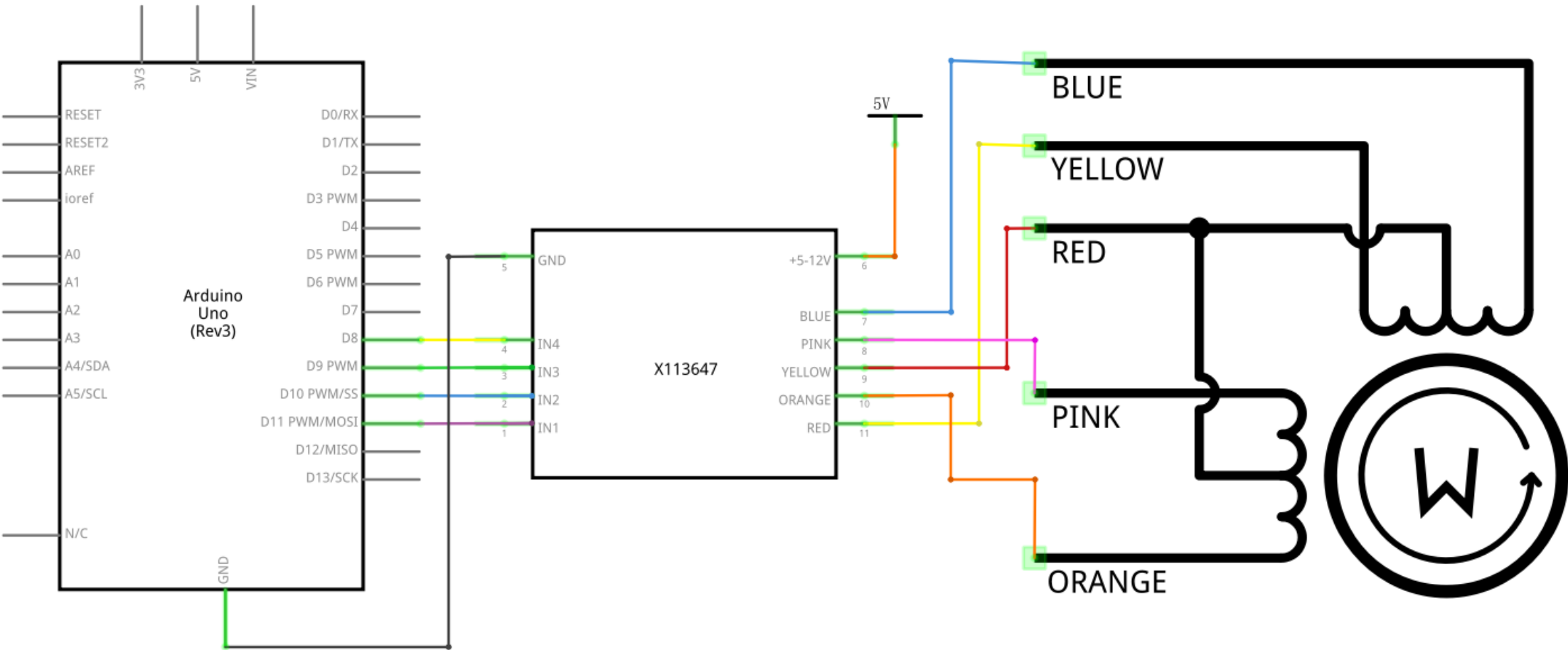
Lead Wire Color	---> CW Direction (1-2 Phase)							
	1	2	3	4	5	6	7	8
4 ORG	-	-						-
3 YEL		-	-	-				
2 PIK				-	-	-		
1 BLU						-	-	-

Hier sind Schaltpläne, die zeigt, wie man eine unipolare Schrittmotoren Schnittstelle Motor mit vier Controller-Pins mit einem ULN2003A, und zeigt, wie man mit vier COM-Schnittstelle

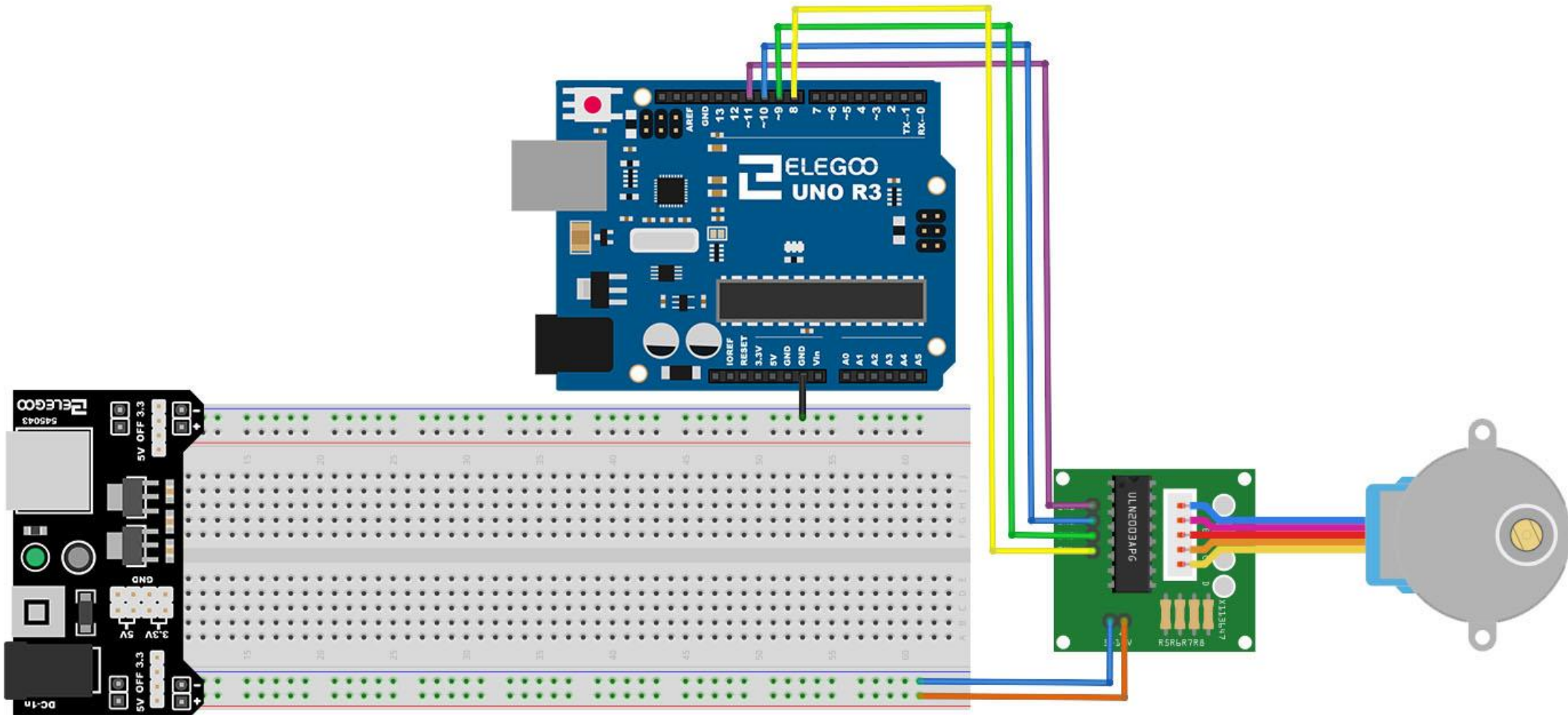


Verbindung

Schema



Schaltplan



Wir werden 4 Pins verwenden, um den Stepper zu steuern.

Pins 8-11 steuern den Schrittmotor.

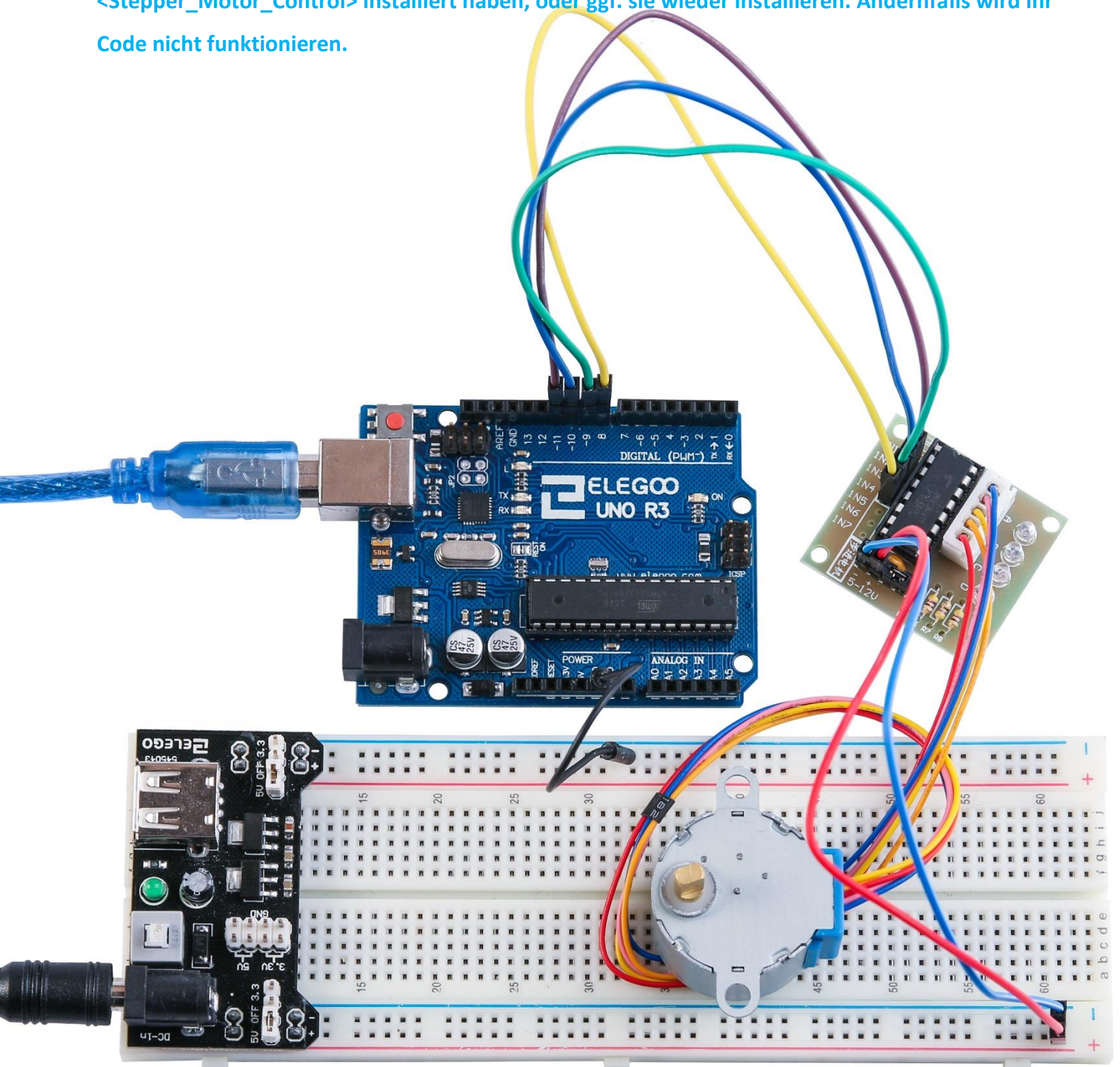
Wir verbinden 5V und Masse vom UNO mit dem Schrittmotor.

Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefodert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek

<Stepper_Motor_Control> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.



Lektion 32: Steuerung des Schrittmotors mit Fernbedienung

Übersicht

In dieser Lektion werden Sie eine lustige und einfache Art und Weise erfahren, um einen Schrittmotor mit einer IR-Fernbedienung aus einem Abstand zu steuern.

Der Stepper, der von uns verwendet, ist mit seiner eigenen Treiberplatine ausgestattet, so kann der Schrittmotor einfach mit unserem UNO verbunden werden.

Da wir den Motor nicht direkt von dem UNO betreiben möchten, werden wir ein preiswertes kleines Netzteil für den Versuchsaufbau, das genau in unseren Versuchsaufbau eingesteckt wird und den Versuchsaufbau mit einer Leistung von 9V 1Amp versorgt, verwenden.

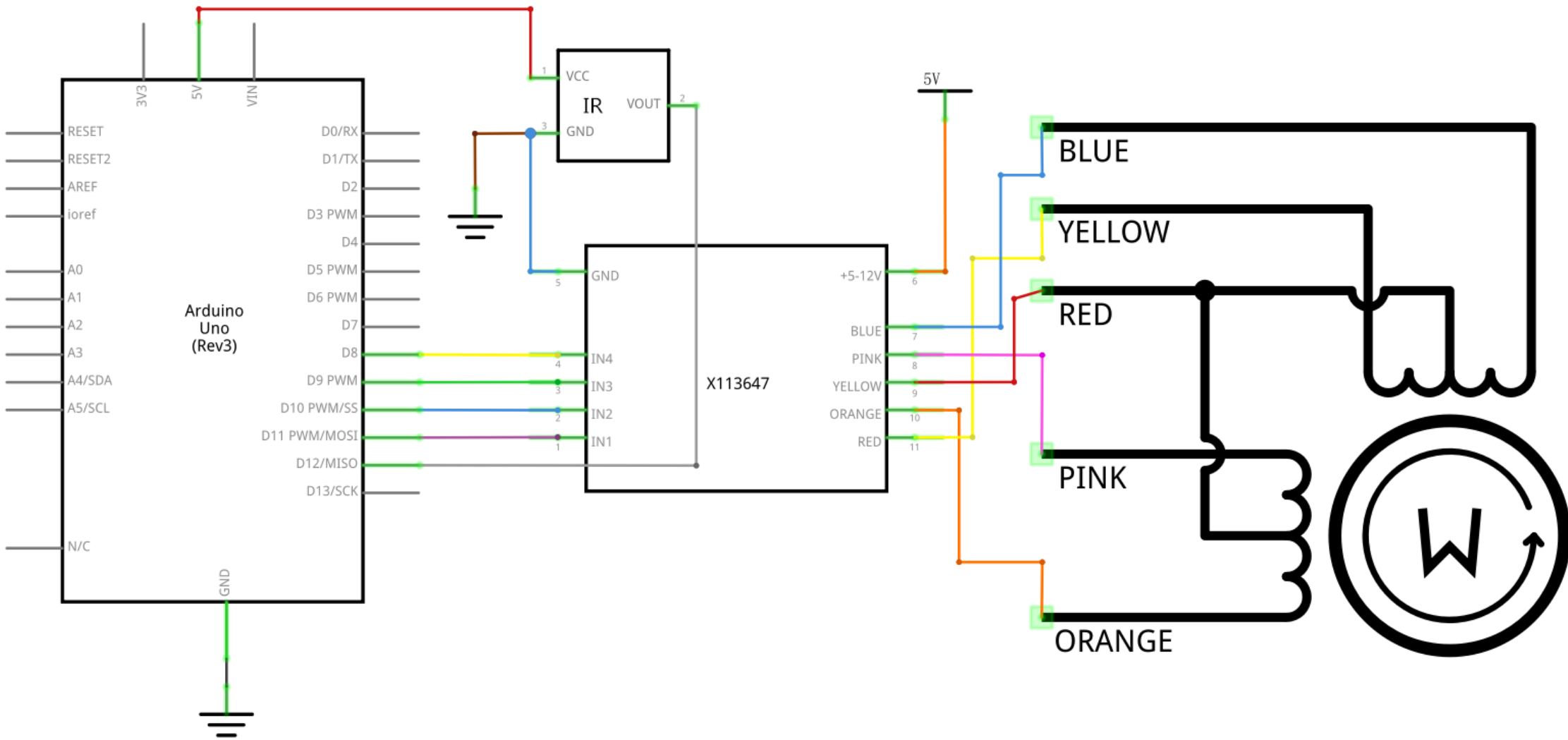
Der IR-Sensor wird direkt mit dem UNO verbunden, da er fast keinen Strom verbraucht.

Erforderliche Komponenten

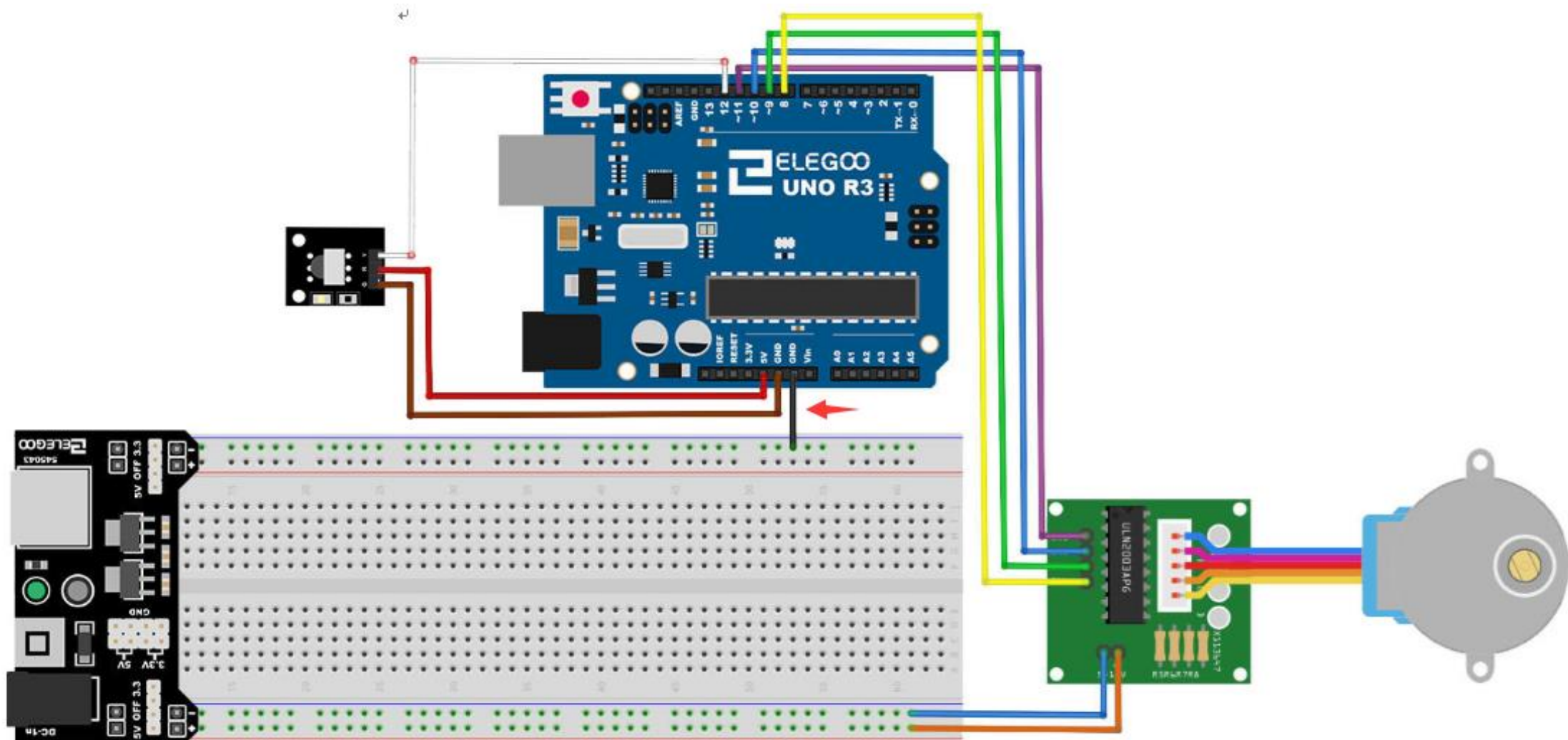
- (1) Elego UNO R3
- (1) Versuchsaufbau
- (1) IR-Empfängermodul
- (1) IR-Fernbedienung
- (1) ULN2003 Schrittmotor-Treibermodul
- (1) Schrittmotor
- (1) Netzteilmodul
- (1) Adapter
- (9) F-M Kabel

Verbindung

Schema



Schaltplan



Wir werden 4 Pins verwenden, um den Stepper zu steuern, und werden 1 Pin verwenden, um den IR-Sensor zu steuern.

Pins 8-11 steuern den Schrittmotor, und Pin 6 empfängt die IR-Information.

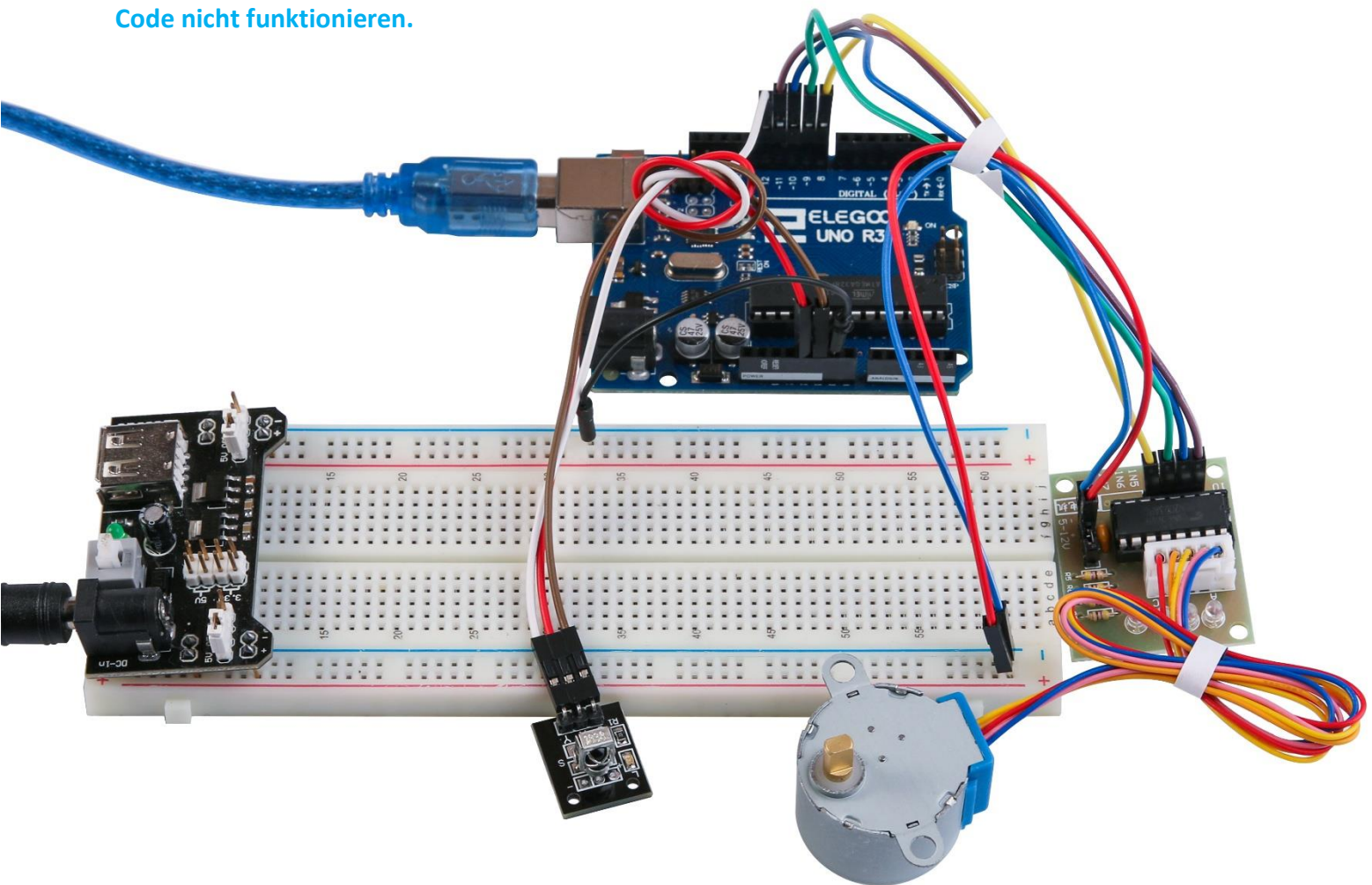
Wir verbinden 5V und Masse vom UNO mit dem Sensor. Vorsorglich verwenden wir ein Netzteil für Versuchsaufbau, da er mehr Strom verbrauchen kann und wir das Netzteil vom UNO nicht beschädigen möchten.

Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek

<Stepper_Motor_Control> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.



Lektion 33: Steuerung des Schrittmotors mit Encoder

Übersicht

In dieser mehrteiligen Lektion werden Sie erfahren, wie Sie verschiedene Schrittmotoren mit einem Drehgeber steuern.

In diesem ersten Teil werden wir den preiswerten und beliebten Schrittmotor, der mit seinem eigenen Steuerungsboard ausgestattet ist, verwenden: Der 28 BYJ-48 Schrittmotor mit dem ULN2003 Board.

Der 28BYJ-48 Motor ist nicht sehr schnell oder sehr stark, aber es ist toll für Anfänger, experimentierend mit der Steuerung eines Schrittmotors mit einem Arduino zu beginnen.

Wir werden einige Codes schreiben, um sicherzustellen, dass der Motor sich in die Richtung, in die wir den Drehgeber drehen, bewegt, und wir werden auch die Menge der Schritte, die wir gemacht haben, verfolgen, so können wir den Motor in die Ausgangsposition zurückgehen lassen, indem wir den Schalter auf dem Drehgeber nach unten drücken.

Für den nächsten Teil werden wir einen Nema 17 Schrittmotor und ein EasyDriver Board verwenden.

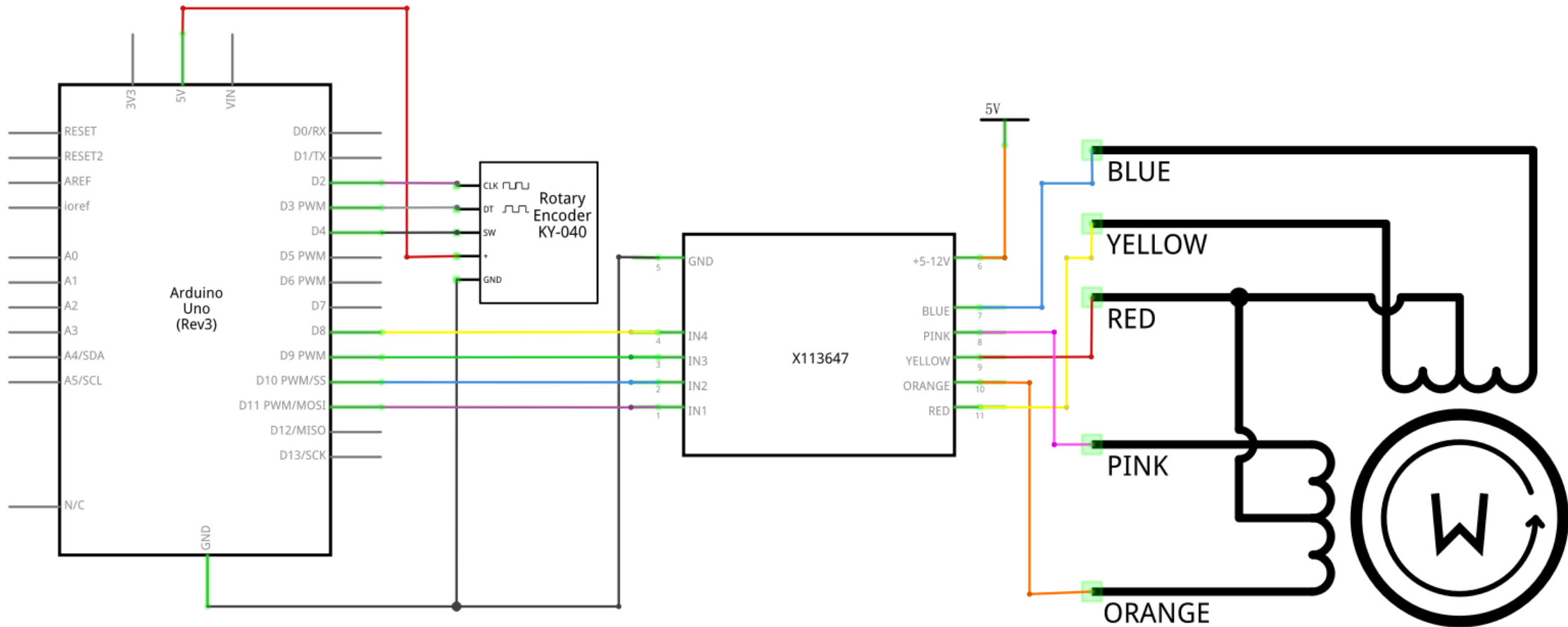
Erforderliche Komponenten

- (1) Elego UNO R3
- (1) Versuchsaufbau
- (1) Drehgeber
- (1) ULN2003 Schrittmotor -Treibermodul
- (1) Schrittmotor
- (1) Netzteilmodul
- (1) Adapter
- (9) F-M Kabel

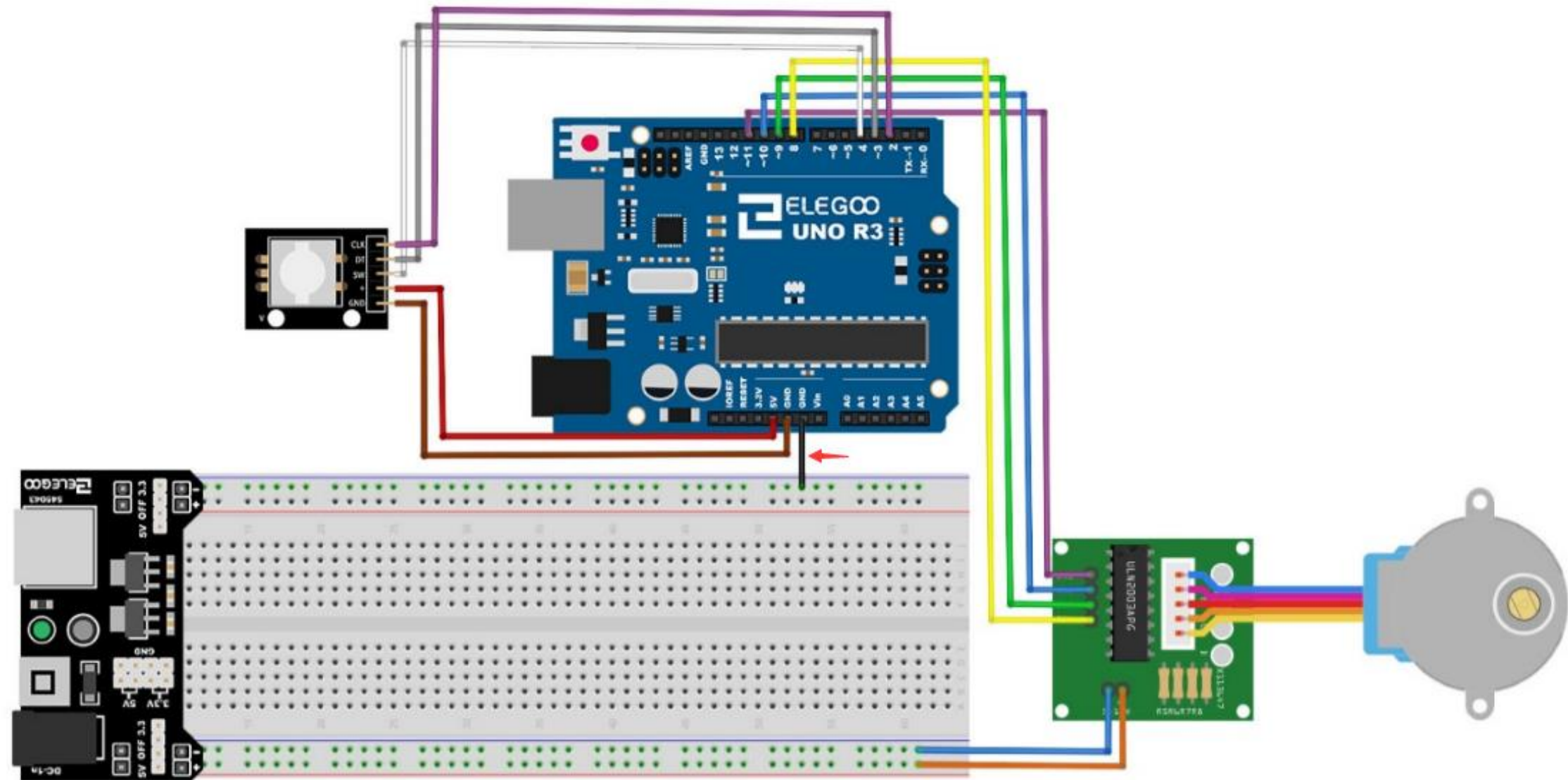


Verbindung

Schema



Schaltplan



Wir verwenden 4 Pins, um den Stepper zu steuern, und 3 Pins, um das Drehgeber-Modul zu steuern. Pins 8-11 steuern den Schrittmotor, und Pins 2-4 empfangen Information aus dem Drehgeber. Wir verbinden 5V und Masse vom UNO mit Drehgeber, und vorsorglich verwenden wir ein Netzteil für Versuchsaufbau, um den Schrittmotor mit Strom zu versorgen, da er mehr Strom, der von dem UNO versorgt werden kann, verbrauchen kann.

Als Referenz verbinden wir auch die UNO-Masse mit dem Versuchsaufbau.

Code

Nach der Verdrahtung, bitte öffnen Sie das Programm, und klicken Sie auf HOCHLADEN, um das Programm zu laden. Wenn Fehler aufgefordert werden, siehe Lektion 2 Einzelheiten zum Programm hochladen.

Bevor Sie dies ausführen dürfen, stellen Sie sicher, dass Sie schon die Bibliothek <Stepper_Motor_Control> installiert haben, oder ggf. sie wieder installieren. Andernfalls wird Ihr Code nicht funktionieren.

Wir verwenden einige Variablen, um die aktuelle Position zu speichern, da wir die Position des Schrittmotors verfolgen möchten, so können wir ihn in die Ausgangsposition zurückgehen lassen. Wir haben auch einige Fehlerüberprüfungscodes, um sicherzustellen, dass der Drehgeber keine Schritte verpasst, andernfalls könnte unsere Motorposition ungenau sein.

