

HowTo Fritzing [Teil 2]

Im ersten Teil dieser Blogserie haben Sie eine einfache Schaltung mit einem Arduino Uno, einer LED, einem Widerstand und einem Temperatursensor erstellt. Bei dieser einfachen schematischen Zeichnung soll es aber nicht bleiben. In diesem Blogbeitrag wird aus der schematischen Zeichnung ein elektrischer Schaltplan.

Dieser wird gerne in Foren oder sozialen Netzen verteilt, um Fehler zu finden oder Wissen weiterzugeben. Natürlich reicht auch die schematische Zeichnung dafür, aber in vielen Fällen ist es dann doch einfacher einen elektrischen Schaltplan zu haben.

Damit Sie diesen Blogbeitrag durcharbeiten können, ist Teil 1 dieser Serie notwendig. Sie können aber auch unter <https://github.com/M3taKn1ght/Blog-Repo> die Fritzing-Datei aus Teil 1 herunterladen und damit diesen Teil erarbeiten.

Öffnen der Ansicht Schaltplan

Nachdem Sie das Projekt aus Teil 1 geöffnet haben, wird zuerst die Willkommen – Ansicht angezeigt. Gerne können Sie noch einmal überprüfen, ob alle Bauteile korrekt auf dem Steckbrett positioniert sind und auch die virtuelle Verdrahtung soweit stimmt. Diese sollte Aussehen, wie in Abbildung 1.

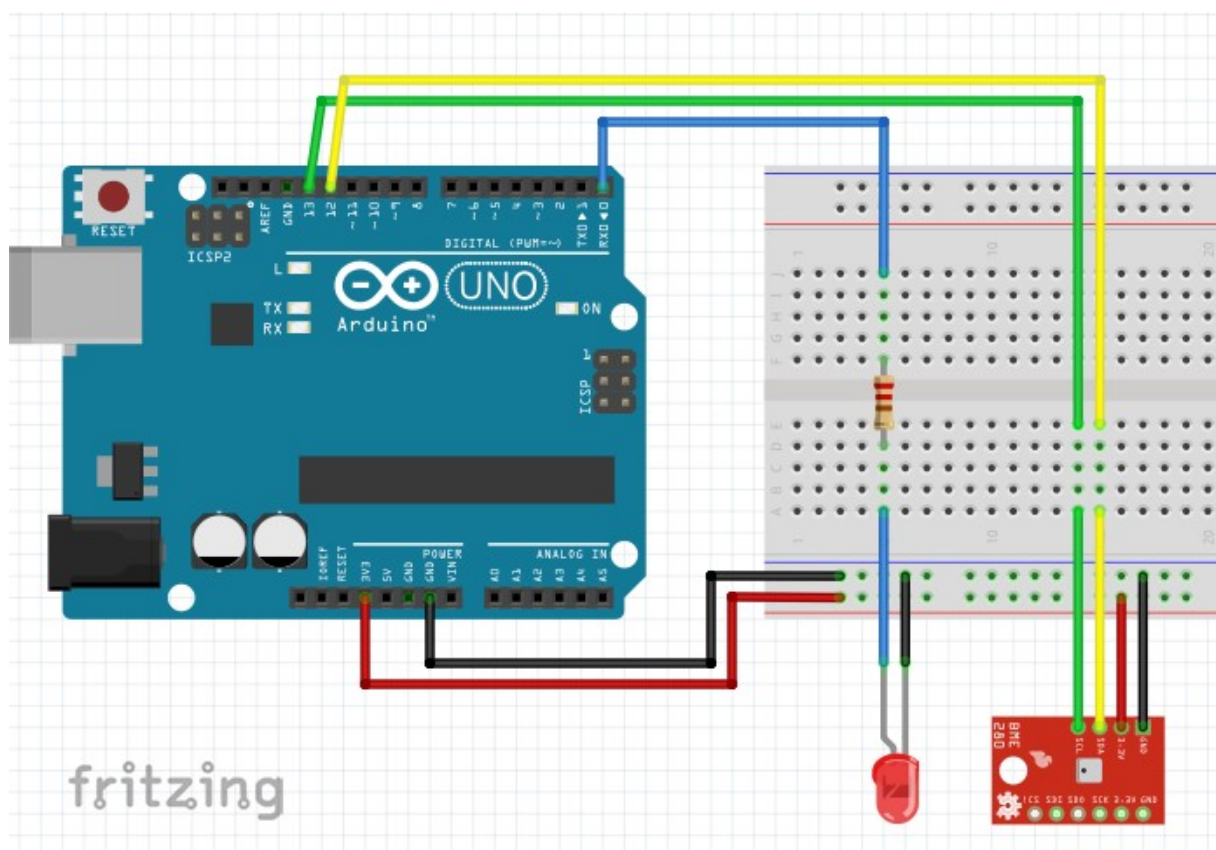


Abbildung 1: Finale schematische Zeichnung aus Teil 1

Stimmt alles, können Sie nun in die Ansicht Schaltplan wechseln.

Sie werden feststellen, dass mit dem Wechsel der Ansicht, sich auch die Ansicht der Bauteile verändert haben. Hierbei handelt es sich um die genormten elektrischen Bauteilsymbole oder um die schematischen Darstellungen der Bauteile. Dabei haben alle Bauteile die gleichen Anschlüsse, wie in der Steckbrettansicht.

Ebenfalls fällt auf, dass alle Bauteile willkürlich in der Ansicht platziert wurden und die angedeuteten Leiterbahnen kreuz und quer verlaufen.

Wichtig ist die Information im unteren Abschnitt, dass „0 von 6 Netzen geroutet – 7 Verbinder noch zu routen“ sind. Damit teilt Ihnen Fritzing mit, dass die Bauteile noch nicht verbunden wurden.

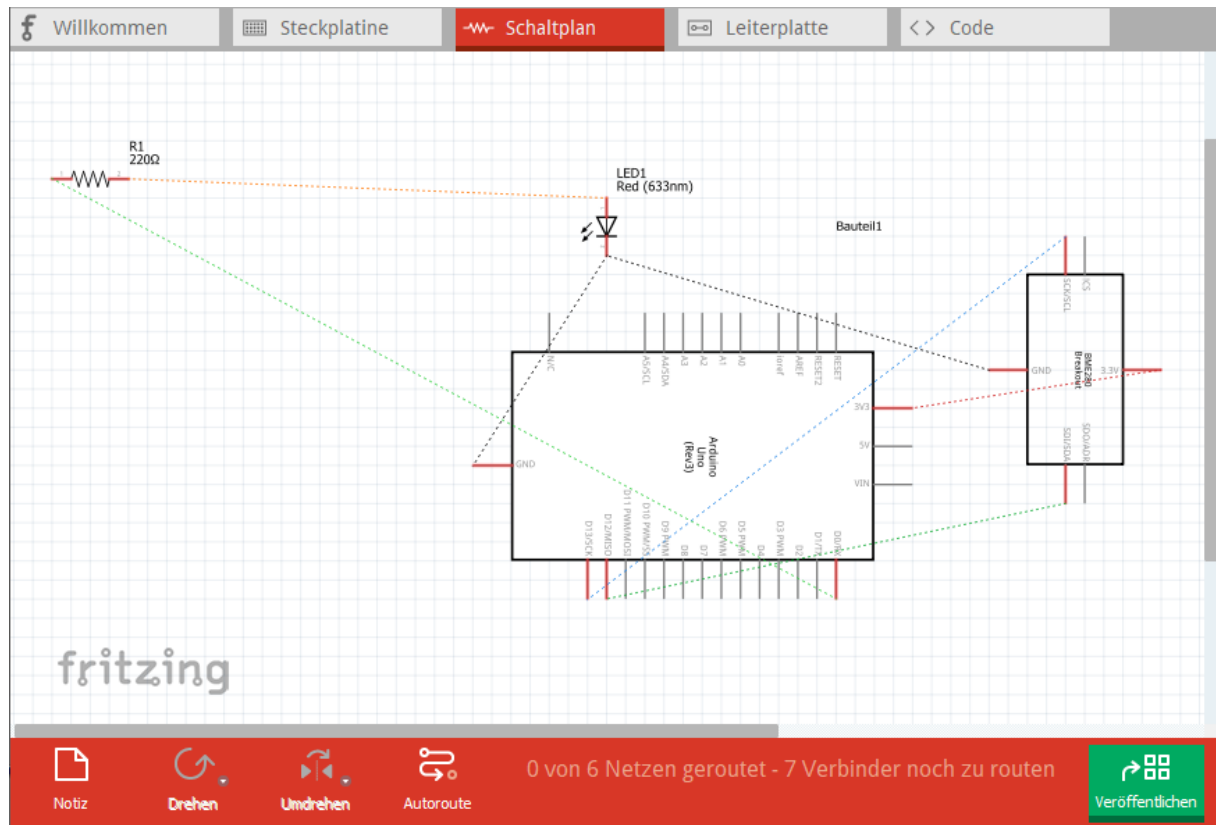


Abbildung 2: Öffnen der Ansicht Schaltplan

Das Autorouting nutzen

Ein erster Versuch, alle fehlenden Bauteile mit Leiterbahnen zu versehen, wäre das Autorouting. Dabei handelt es sich um einen Algorithmus der die Leiterbahnen automatisch erzeugt. Probieren Sie es gerne einmal aus, jedoch wird das Ergebnis weniger zufriedenstellend sein, als Sie erhoffen. Es wird so ähnlich aussehen wie in Abbildung 3.

Sieht man sich die Leiterbahnen an, so wird schnell deutlich, dass das Routing per Hand die bessere Wahl ist.

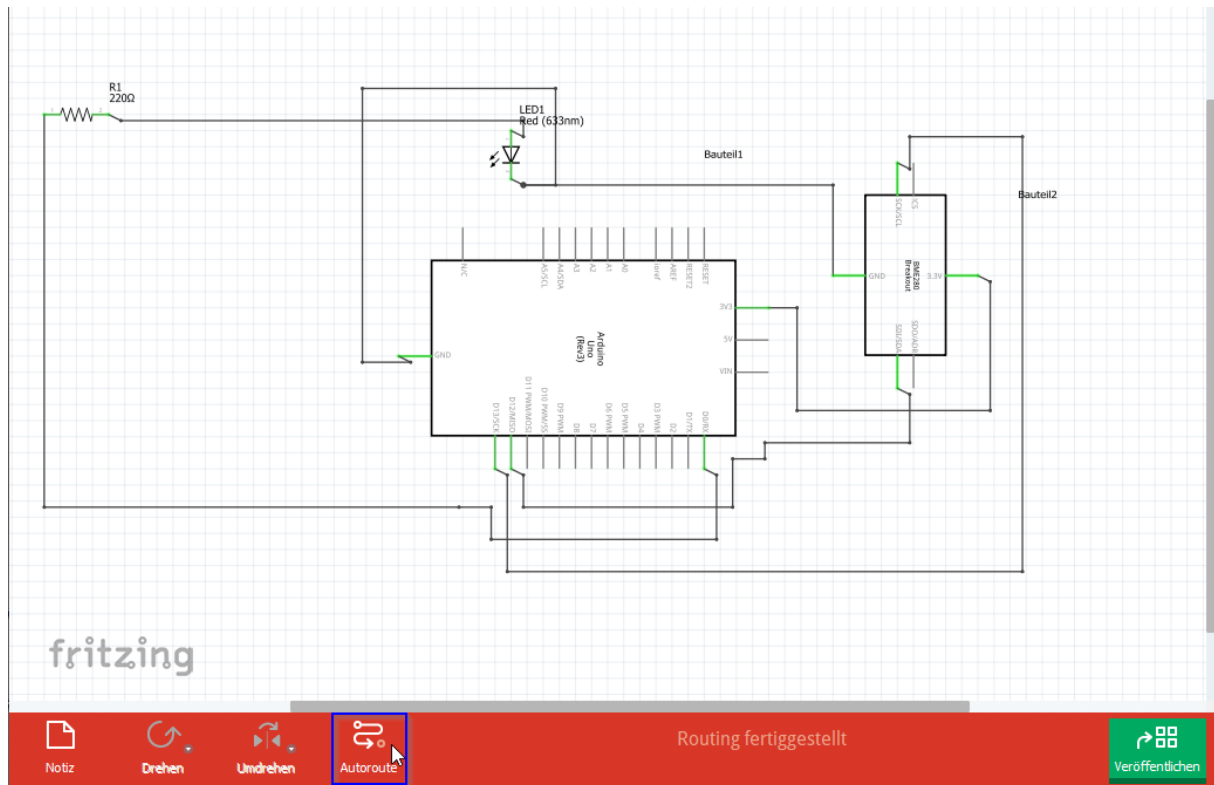


Abbildung 3: Autorouting von Fritzing verwenden

Ein erster Grund, warum das Autorouting nicht wirklich klappt, sind die bisher willkürlichen gewählten Positionen der elektrischen Elemente.

Das Routing per Hand

Gerade bei komplexen schematischen Zeichnungen kann das viel Zeit in Anspruch nehmen, daher beschränkt sich diese Blogserie auf ein einfaches Beispiel. Haben Sie die Option „Autoroute“ verwendet, so machen Sie dies bitte rückgängig. Dies geht entweder über das erneute Öffnen der Fritzing-Datei, ohne woher zu speichern oder die Option Bearbeiten -> Rückgängig (Strg + Z).

Bevor die ersten Leiterbahnen erzeugt werden, sollten die Bauteile neu positioniert werden. Das vereinfacht im späteren Verlauf das Routing der Leiterbahnen erheblich. Gleichzeitig wird auch die Orientierung der Bauteile angeglichen, die Luftlinie der Leiterbahnen dient dabei als Orientierungshilfe.

Zunächst greifen wir die LED und positionieren diese, durch gedrückt halten der linken Maustaste, unterhalb von Ground des Arduinos, siehe Abbildung 4.

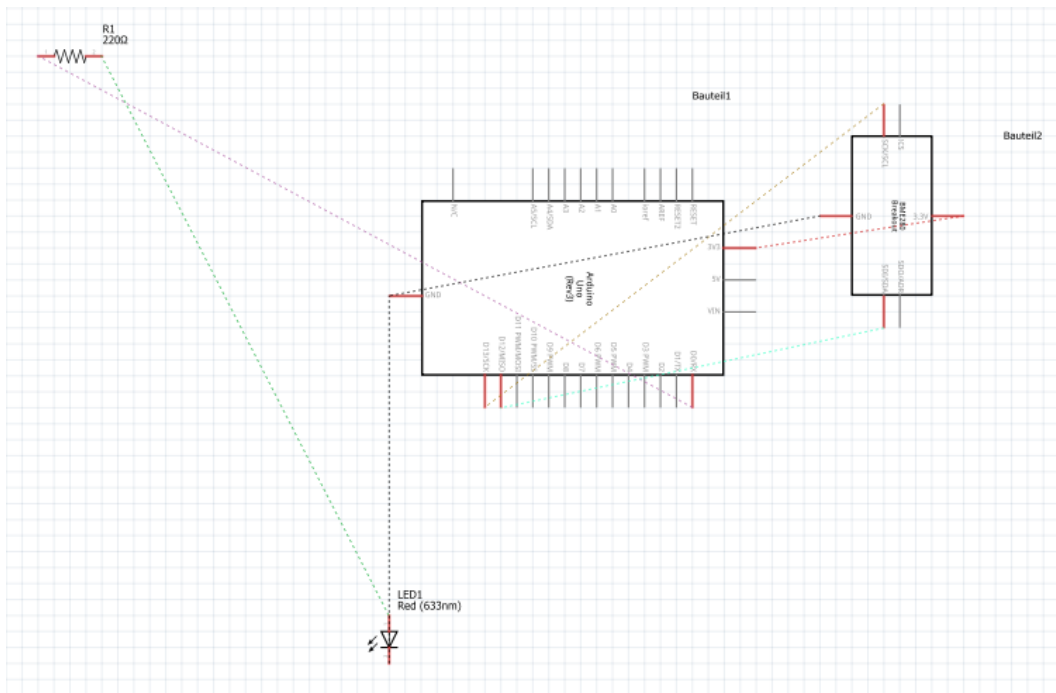


Abbildung 4: LED neu positionieren

Im nächsten Schritt drehen wir die LED um 90° im Uhrzeigersinn, siehe Abbildung 5.

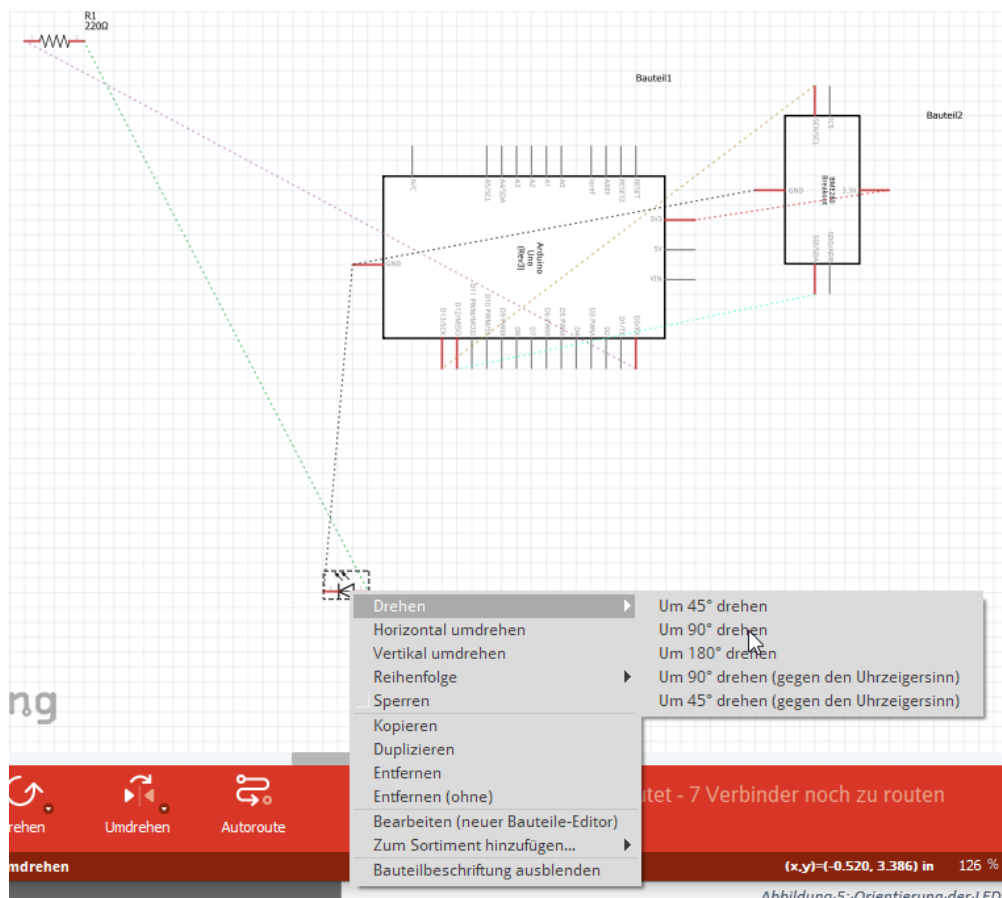


Abbildung 5: Orientierung der LED anpassen

Dass der Anschluss des LED nun nicht mehr genau unter dem Ground-Anschluss des Arduinos liegt, lassen Sie zunächst außer Acht. Positionieren Sie erst einmal die anderen Bauteile. Dabei gehen Sie wie folgt vor:

- Den Widerstand (R1) auf gleicher horizontaler Linie wie die LED positionieren und um 180° drehen
- BME280 um 90° gegen den Uhrzeigersinn drehen und unterhalb vom Arduino positionieren

Das Ergebnis sollte nun ungefähr so aussehen wie in Abbildung 6.

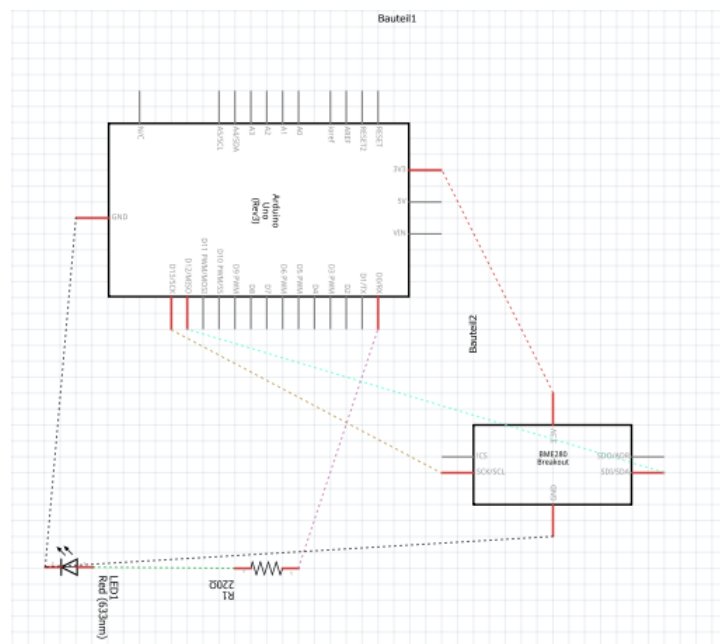


Abbildung 6: Bauteile neu positioniert und orientiert

Damit sind alle Bauteile erst einmal an der korrekten Stelle, nun müssen die entsprechenden Leiterbahnen für die einzelnen Bauteile gezogen werden. Ähnlich wie in Teil 1 nutzen wir die Griffpunkte, siehe Abbildung 7, um eine Leiterbahn von Kathode LED zu Ground Arduino zu ziehen.



Abbildung 7: Griffpunkt um Leiterbahn zu generieren

Bei dieser Leiterbahn wollen wir wieder die vierte Regel vom Styleguide anwenden. Diese Regel besagt, dass Leiterbahnen immer rechtwinklig und nie quer verlaufen dürfen. Hier nutzen wir auch wieder die Option, Biegepunkte auf den Leiterbahnen zu erstellen, siehe Abbildung 8. In der Abbildung ist schon die Leiterbahn entsprechend dem Styleguide angepasst worden.

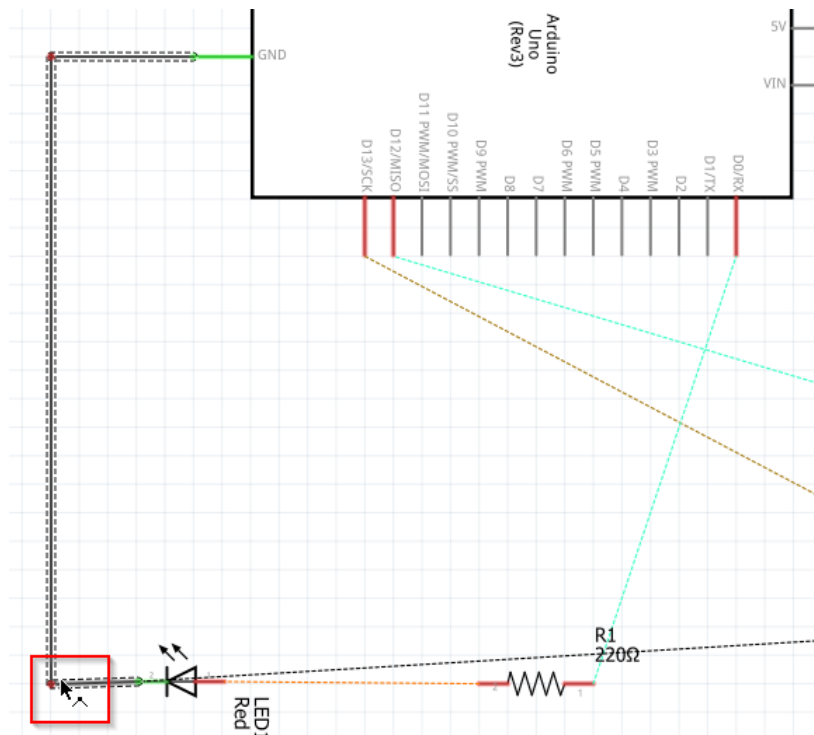


Abbildung 8: Biegepunkt auf Leiterbahn erzeugen

Die restlichen Leiterbahnen zwischen LED, Widerstand und Arduino werden nach demselben Verfahren generiert. Beachten Sie dabei die Luftlinien die Fritzing vorgibt, siehe Abbildung 9.

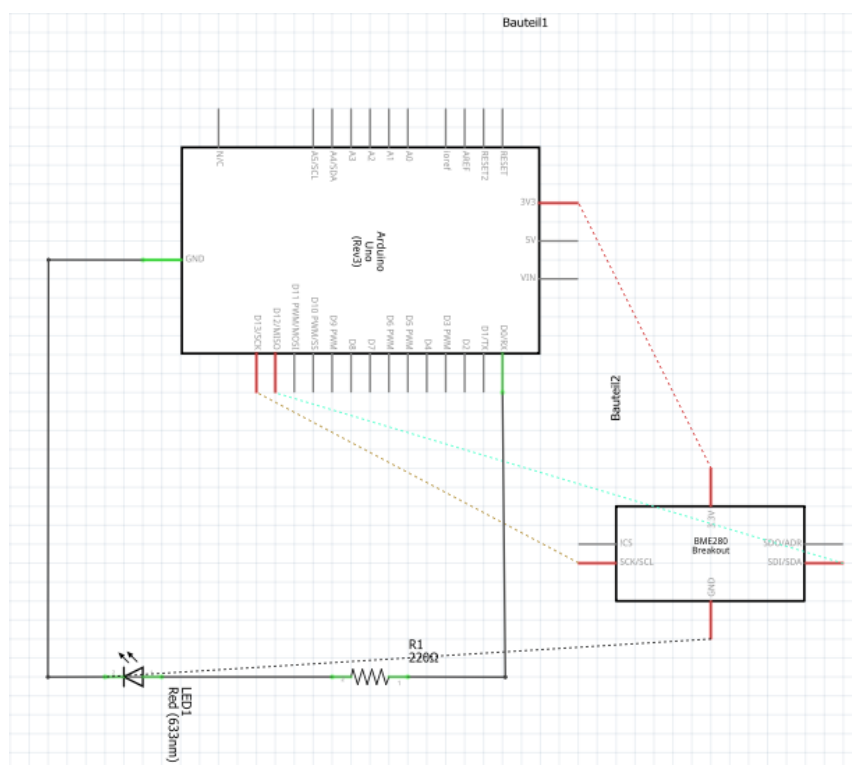


Abbildung 9: Fertige Leiterbahn zwischen LED, Widerstand und Arduino

Nun wird das BME280 mit dem Arduino verbunden. Bei dem Ground-Anschluss vom BME gibt es zwei mögliche Varianten der Anbindung. Die erste Möglichkeit ist, den Ground vom BME direkt mit dem Ground vom Arduino zu verbinden, siehe Abbildung 10.

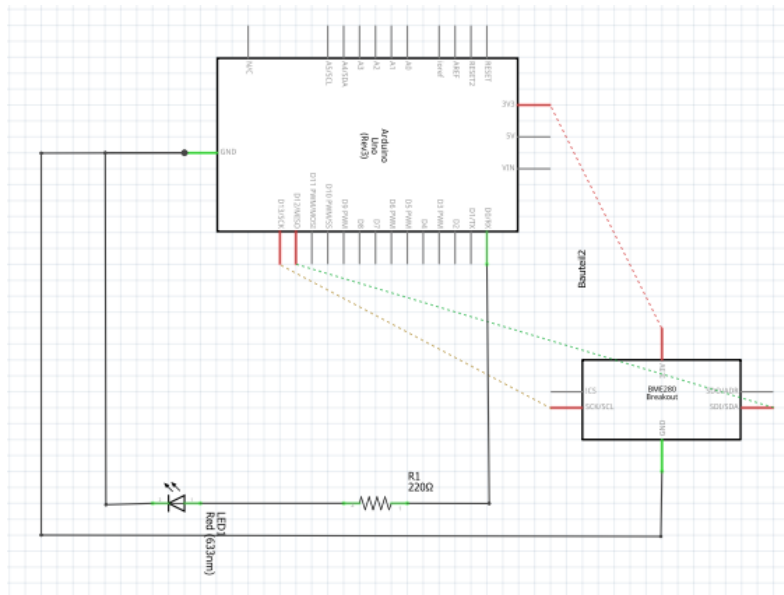


Abbildung 10: BME280 direkt an Arduino Ground

Das sieht aber nicht professionell aus und könnte bei komplexeren Schaltungen zu vielen unnötigen virtuellen Leiterbahnenlängen führen. Wie Sie ggf. bereits bemerkt haben könnten, hat Fritzing einen dickeren Punkt beim Arduino Ground-Anschluss eingezeichnet. Diesen dickeren Punkt nennt man in der Elektrotechnik Knotenpunkt. Dieser Knotenpunkt ist unsere zweite Möglichkeit beim Einzeichnen einer Leiterbahn und diesen werden wir nun auch nutzen. Dazu ziehen Sie eine Leiterbahn von Ground des BME280 zum Biegepunkt hinter der Katode der LED, siehe Abbildung 11 rote Umrandung. Hierdurch verschieben wir den Knotenpunkt und sparen dadurch Leiterbahnlänge.

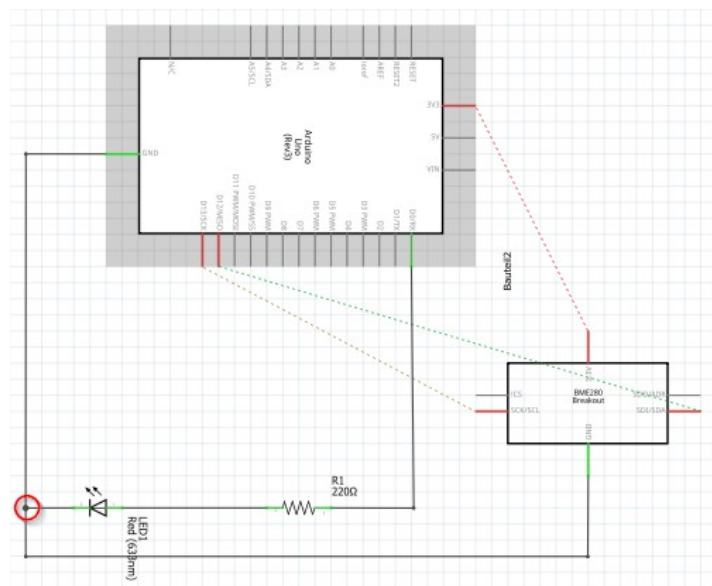


Abbildung 11: Knotenpunkt an Biegepunkt Katode LED

Zuletzt werden noch die letzten Leiterbahnen gelegt. Bevor Sie nun versuchen kreuzende Leiterbahnen zu vermeiden, wie es der Styleguide vorgibt, kann ich jetzt schon versichern, dass Ihnen dies nicht gelingen wird. Da aber beim Kreuzen der Leiterbahnen kein Knotenpunkt erzeugt wird, sollte ein geübter Leser eines Schaltplanes keine Probleme damit haben. Das Endergebnis der Verbindungen sollte Abbildung 12 ähneln.

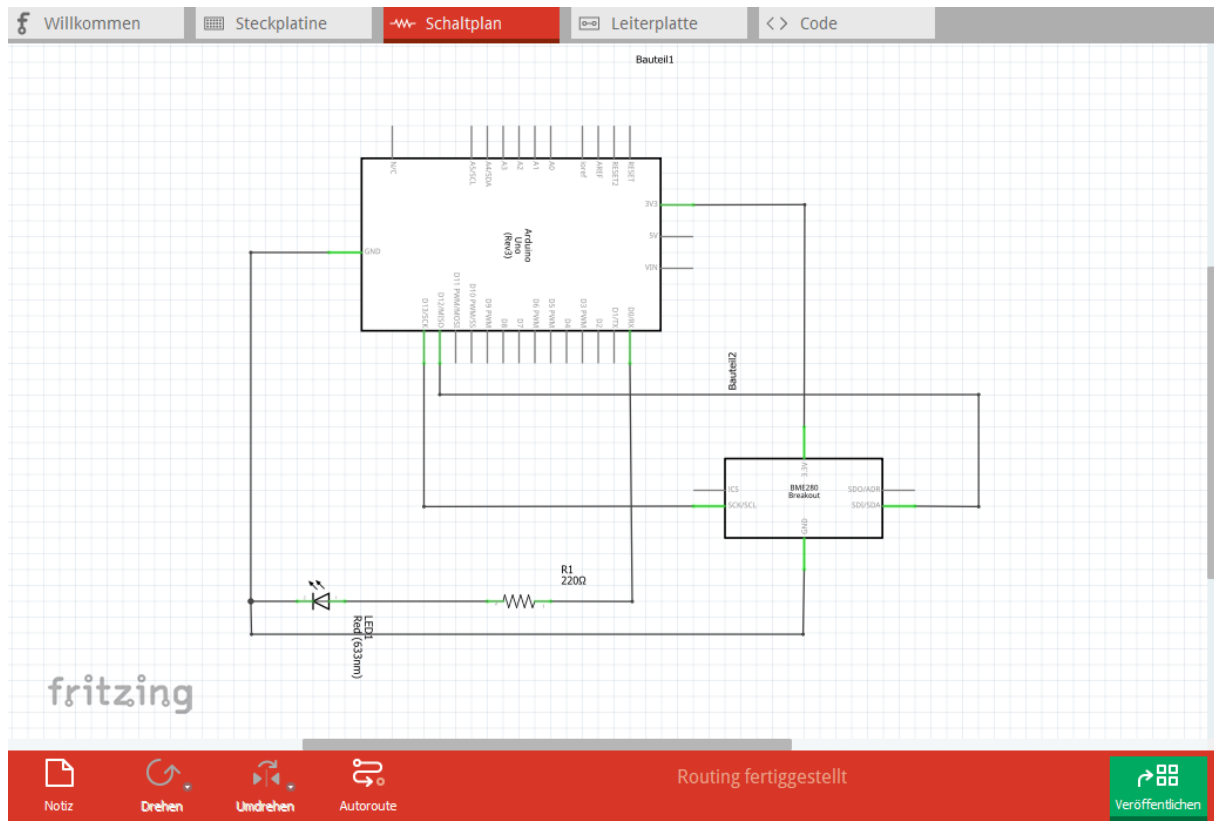


Abbildung 12: Schaltplan fertig verbunden

Nachdem Sie alle Leiterbahnen gelegt haben, werden Sie feststellen, dass Fritzing im unteren Bereich die Meldung ausgibt, dass das „Routing fertiggestellt“ wurde. Dies bedeutet, dass keine offenen Verbindungen mehr vorhanden sind.

Allerdings sind wir an dieser Stelle mit diesem Blogbeitrag noch nicht fertig. Wir wenden nun die Regel sechs unseres Styleguides an. Diese besagt: „Leiterbahnen sollten nie andere Leiterbahnen kreuzen. Sollte das nicht möglich sein, sollten die Leiterbahnen verschiedene Farben haben“. Da in dieser Schaltplanzeichnung Leiterbahnen kreuzen, werden die Farben dem Steckplattenlayout angepasst. Daraus folgt, dass:

- Ground die Farbe schwarz erhält
- 3,3V Spannungsversorgung die Farbe rot
- SCL die Farbe grün
- SDA die Farbe gelb
- Die Leiterbahn zwischen Digital-Ausgang 0 vom Arduino, über Widerstand zu Anode LED die Farbe blau

Das einfärben der Leiterbahnen ist identisch zu den Methoden bei der Steckplatine. Welche Sie bevorzugen liegt an Ihnen, das Ergebnis sollte wie in Abbildung 13 aussehen.

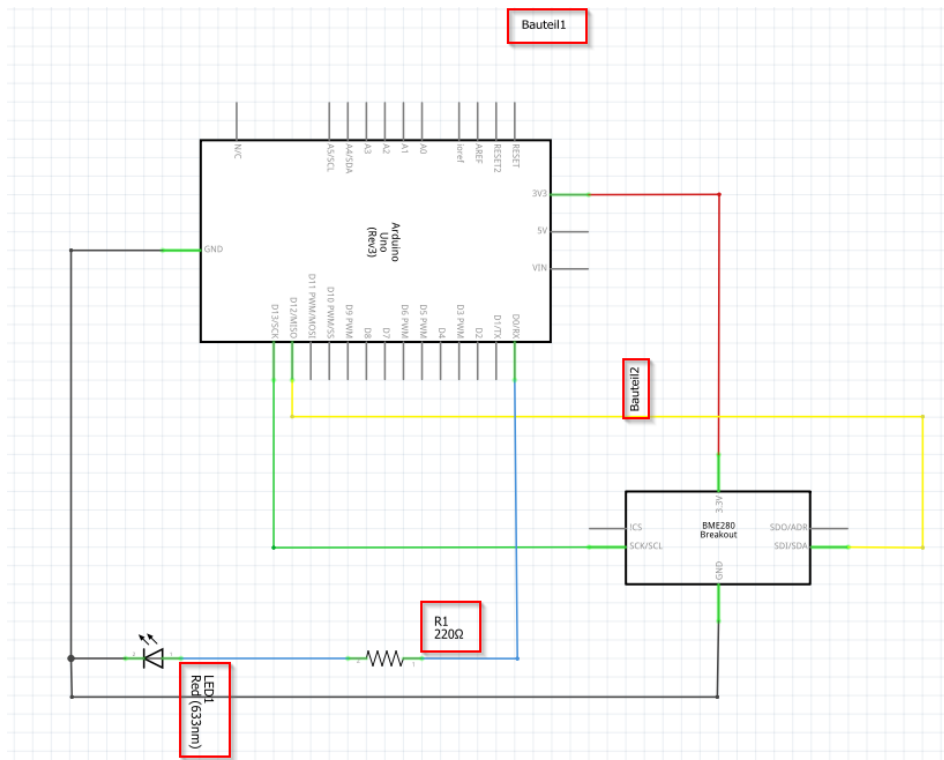


Abbildung 13: Leiterbahnen eingeführt

Zuletzt sollen noch die störenden Beschriftungen, in Abbildung 13 rot markiert, richtig positioniert und der Text korrigiert werden. Für den Text reicht es einen Doppelklick mit der linken Maustaste auf die Beschriftung zu machen und im neuen Dialog den korrekten Text einzugeben, siehe Abbildung 14.

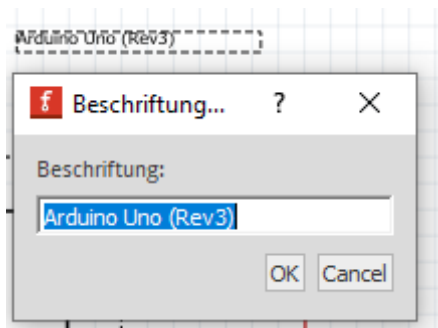


Abbildung 14: Beschriftung anpassen

Bei der Orientierung und der Positionierung gehen Sie genauso vor, als ob Sie das Bauteil verschieben würden. Die entsprechenden Menüs und das Verhalten sind identisch. Das endgültige Ergebnis sehen Sie in Abbildung 15.

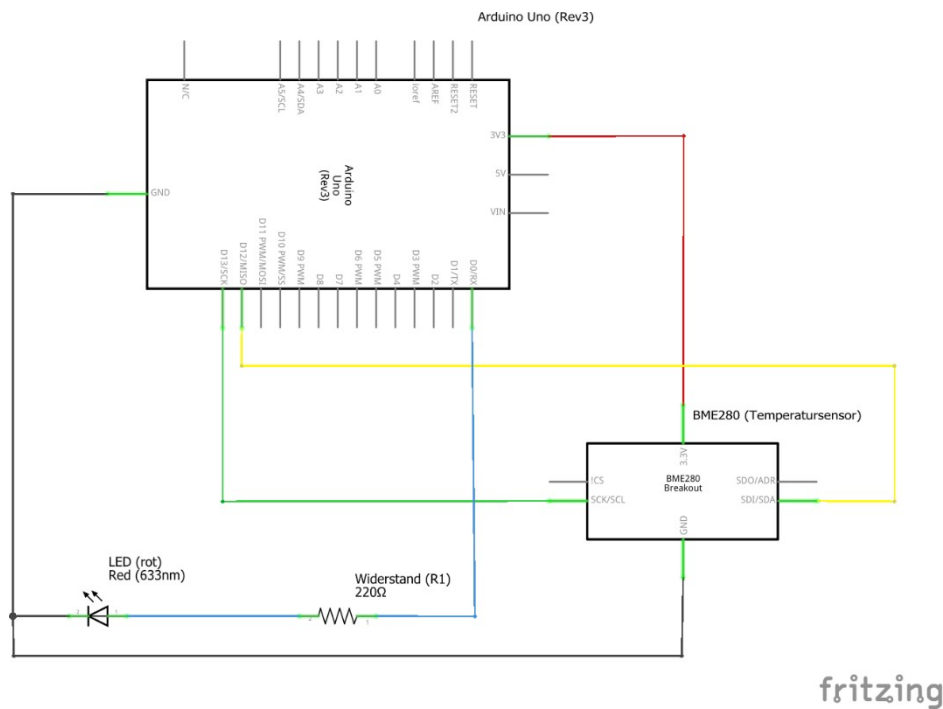


Abbildung 15: Fertiger Schaltplan

In dem aktuellen Blogartikel haben Sie gelernt aus einer einfachen Steckplatinschaltung einen Schaltplan zu erzeugen. Natürlich gibt es noch weitere Möglichkeiten, um den Schaltplan zu optimieren, aber die Grundkenntnisse wurde vermittelt. Probieren Sie es ruhig einmal aus, wie verschiedene Positionen der Bauteile das Aussehen des Schaltplanes verändert.

Weitere Projekte für AZ-Delivery von mir, finden Sie unter <https://github.com/M3taKn1ght/Blog-Repo>.