

HowTo Fritzing [Teil 1]

Jeder von uns hat das schon einmal erlebt, man sieht im Internet ein großartiges Projekt, aber bis auf die wenigen Bilder und vielleicht ein paar Kommentare im Quellcode ist es einfach nicht möglich dieses Projekt nachzubauen. Neben den Fotos wäre da eine schematische Zeichnung sehr hilfreich. Bei vielen Blogbeiträgen von AZ-Delivery kriegen Sie diese Zeichnungen, in Form von Fritzing-Bildern, direkt geliefert.

Daher will ich Ihnen im ersten Teil dieser Serie das Tool Fritzing und ein paar grundlegende Funktionen dieser Applikation näherbringen. Gleichzeitig, so gehört sich das nicht nur beim Programmieren, gibt es gewisse Regeln, um eine solche Zeichnung lesbarer zu machen.

Der zweite Teil widmet sich mit dem Erstellen von Schaltplänen.

Im dritten Teil dieser Blogserie, erkläre ich dann genauer, wie Sie externe Bauteile Ihren Fritzing-Parts hinzufügen und wo die besten Quellen sind, um diese zu finden.

Im vierten und letzten Teil der Serie, gehe ich dann auf das Erzeugen von eigenen Bauteilen ein. Sowie, welche Regeln Fritzing dafür hat und welche Tools Sie dafür neben Fritzing noch benötigen.

Was ist Fritzing

Geht man auf [Fritzing.org](https://fritzing.org), so findet man direkt die passende Antwort.

„Fritzing ist eine Open-Source-Hardware-Initiative, die Elektronik als kreatives Material für jedermann zugänglich macht. Wir bieten ein Software-Tool, eine Community-Website und Dienstleistungen im Geiste von Processing und Arduino an und fördern ein kreatives Ökosystem, das es den Benutzern ermöglicht, ihre Prototypen zu dokumentieren, sie mit anderen zu teilen, Elektronik im Klassenzimmer zu unterrichten und professionelle Leiterplatten zu entwerfen und herzustellen.“

Zitat 1: Übersetzung von Fritzing.org

Um es einfach zu halten, Fritzing ist eine Software für jedermann, um damit Elektronikschaltpläne zu erzeugen, zu teilen und zu dokumentieren. Open-Source-Hardware-Initiative geht auf den Open-Source-Gedanken zurück. Der Quellcode von Fritzing ist dabei jedermann unter <https://github.com/fritzing/fritzing-app> zugänglich und generell kann jeder, der C++ programmieren und QT kann, an diesem Projekt mitwirken.

Die Applikation ist erhältlich für Windows, macOS und Linux und dafür möchte Fritzing lediglich eine kleine Spende. Zusätzlich neben der Verfügbarkeit auf allen gängigen Betriebssystemen, wird diese Applikation auch noch in eine Vielzahl von Sprachen übersetzt. Die aktuelle Version 0.9.4 umfasst insgesamt 21 Sprachen!

Woher bekomme ich Fritzing

Für die meisten Betriebssystem, gerade bei Windows und macOS, sollten Sie die aktuelle Version unter <https://fritzing.org/download/> downloaden und nach Anleitung für das jeweilige Betriebssystem installieren. Das gleiche gilt auch bei Linux, Vertreter hier ubuntu, wobei in den Paketverwaltungen es durchaus ältere Versionen zum Installieren geben könnte. Empfehlen würde ich dennoch die Installation der aktuellen Version der Homepage.

Teilweise findet man auch im Internet sogenannte „portable versions“, also eine Version, die auf einen USB-Stick installiert und ausgeführt wird, ohne das Admin-Rechte auf einem PC nötig sind. Hierbei ist aber auch wieder auf die zugrunde gelegte Version zu achten. Eine portable Version von Fritzing 0.9.4 habe ich bisher nicht finden können.

Der erste Start

Hat man Fritzing installiert, so kann man es auch direkt starten. Registrierung oder eine andere Art von Lizenzwesen, um die Applikation nutzen zu können, gibt es nicht. Den Aufbau der Software unterteile ich in vier Gruppen, siehe Abbildung 1.

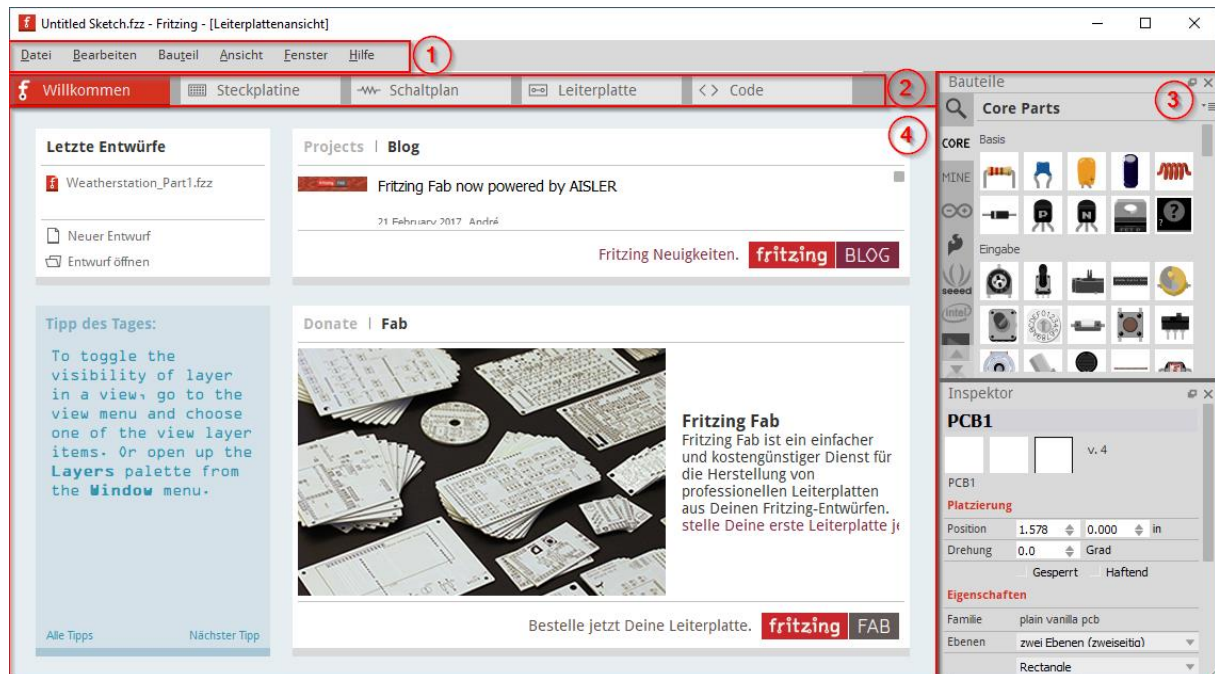


Abbildung 1: Fritzing direkt nach dem Start

Ganz oben ist die Menüleiste (1), die bei allen Applikationen vorhanden ist. Mit ihr kann das Aussehen und die Einstellungen durch die verschiedenen Untermenüs verändert, sowie Projekte geladen und gespeichert werden.

Darunter gibt es eine Schnellauswahl (2) für die fünf benötigten Ansichten. Diese können auch über die Menüleiste -> Ansicht oder die Shortcuts Strg + 1 - 5 ausgewählt werden, aber ein Klick auf die richtige Ansicht geht dann schon mal schneller. Gleich zu Anfang möchte ich die verschiedenen Ansichten kurz erläutern.

Ansicht	Bemerkung	Shortcut
Willkommen	Eine einfache Ansicht, die oben links die letzten Entwürfe anzeigt oder uns die Möglichkeit gibt, einen neuen Entwurf oder vorhandenen Entwurf zu öffnen. Zusätzlich gibt es immer mal wieder neue Infos aus dem Blog oder zusätzliche Infos. Hier werden Sie wohl wenig Zeit verbringen	Strg + 1
Steckbrett	Die Ansicht Steckbrett wird noch ausführlicher erklärt, aber hier werden die Projekte zu den schematischen Zeichnungen zusammengesteckt.	Strg + 2
Schaltplan	Die Ansicht Schaltplan ist eher für die (Hobby-)Elektroniker gedacht, die einen ordentlichen Schaltplan haben wollen. Hier wird der Aufbau auf dem Steckbrett in eine DIN gerechte Zeichnung umgewandelt. Wobei umwandeln hier nicht ganz stimmt, denn auch hier muss man die Leiterbahnen korrigieren. Zum späteren Zeitpunkt werde ich noch einmal auf die Ansicht zu sprechen kommen.	Strg + 3

Leiterplatte	Gerade wenn Projekte nicht nur auf dem Steckbrett, sondern auf einer Leiterplatte verewigt werden soll, werden (Hobby-)Elektroniker hier Zeit investieren. Die Ansicht kommt an ein Programm wie Target 3001! nicht ran, aber um einfache Leiterplatten zu kreieren reicht diese Ansicht allemal.	Strg + 4
Code	Da Fritzing in der Vergangenheit gerne für Arduino-Projekte genutzt wurde, gibt es die Möglichkeit für den Arduino Quellcode gleich in Fritzing umzusetzen. Sie können ausschließlich Arduino-Module auswählen und nutzen. Wollen Sie z.B. einen ESP32 programmieren, müssen Sie direkt in die Arduino IDE wechseln.	Strg + 5

Auf der rechten Seite befindet sich die Fensterleiste (3), die das Bauteil- und das Inspektorfenster anzeigt. Zusätzlich kann noch das Verlaufs- und Ebenenfenster eingeblendet werden. Theoretisch kann ich jedes einzelne Fenster frei auf meinem Desktop platzieren, aber in den meisten Fällen werden die beiden meist genutzten Fenster in der Fensterleiste gelassen.

Zuletzt gibt es dann noch die Hauptansicht (4), die sich je nach gewählter Ansicht (2) unterscheidet. Diese Ansicht ist, sehen wir mal von dem Verhalten der Fensterleiste ab, das einzige Element, welche sich verändert.

Der kleine Styleguide der schematischen Zeichnung

Bevor es an die ersten Zeichnungen geht, möchte ich noch einmal verdeutlichen, wie wichtig gewissen Regeln beim Kreieren von schematischen Zeichnungen sind. Jeder Programmierer hat schon mal vom Styleguide gehört und so gibt es diese Regeln auch für Zeichnungen. Keine Sorge, es wird nun kein halbes Studium aus dem Bereich technische Zeichnungen, sondern es geht um grundlegende Gedanken. Schauen wir uns einmal Abbildung 2 an.

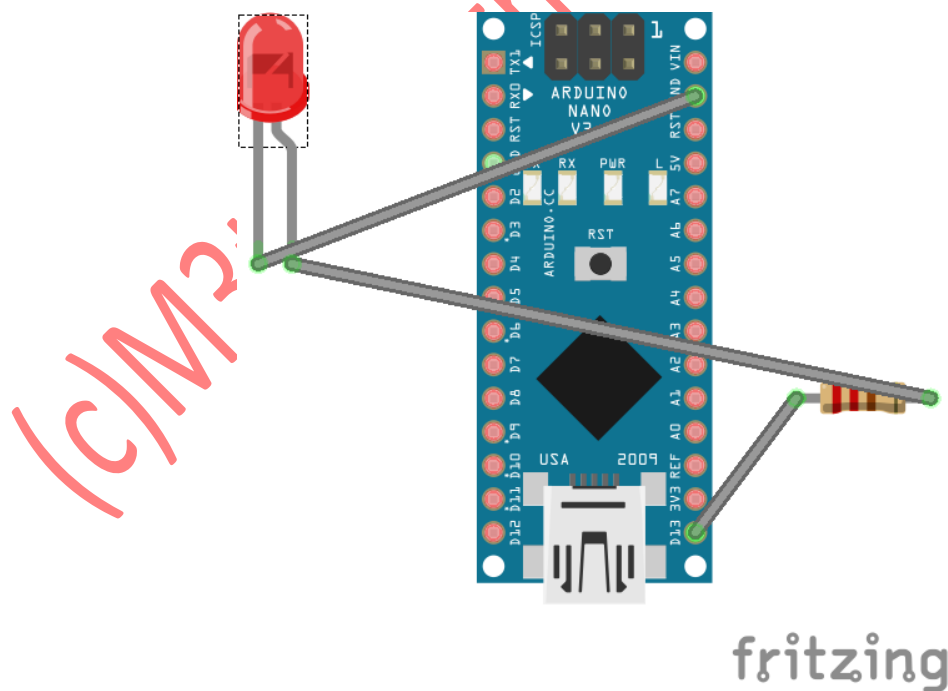


Abbildung 2: Beispiel Blink nicht sauber gezeichnet

Dies ist ein einfaches Beispiel, aber schon hier fällt auf, dass die Leiterbahnen und die Bauteile etwas seltsam angeordnet sind und es ist unklar warum die Leiterbahnen über den Arduino Nano verlaufen.

Denkt man nun an komplexere Projekte, wäre das Chaos beim Lesen der schematischen Zeichnung perfekt und ein Nachbau fast unmöglich. Anders sieht es da in Abbildung 3 aus.

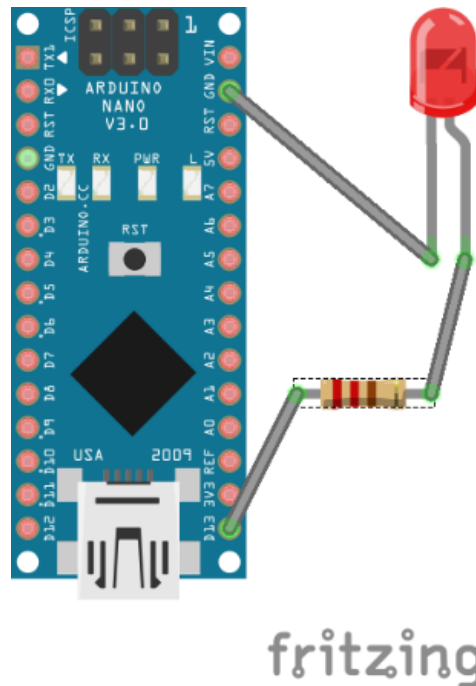


Abbildung 3: Beispiel Blink, keine Leiterbahnen über den Controller

Die Leiterbahnen kreuzen nun nicht mehr den Arduino Nano und die Zuordnung der Pins zu den Bauteilen ist klarer geworden. Im Vergleich zu Abbildung 2 erkennt man nun deutlich, dass keine anderen Pins, außer D13 und GND, vom Arduino Nano genutzt werden. Jedoch stören die schrägen Leiterbahnen ein bisschen. In diesem Fall würde es reichen, wenn einfach der Widerstand und die LED ein bisschen zurechtgerückt werden, aber man kann einfach ein oder mehrere Biegepunkte für die Leiterbahn setzen. Das Ergebnis ist in Abbildung 4 zu sehen.

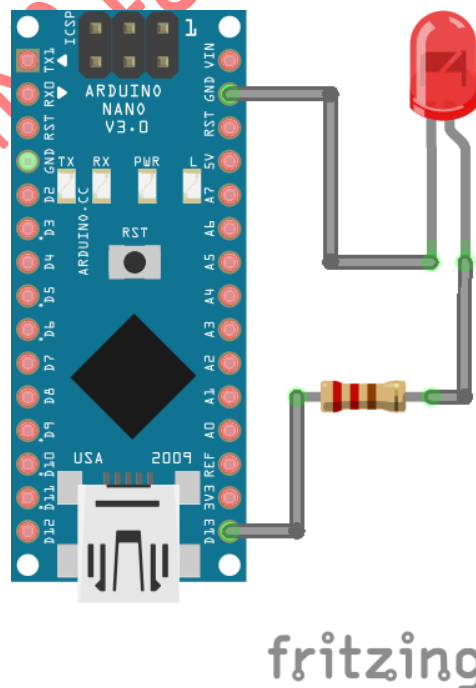


Abbildung 4: Beispiel Blink mit Biegepunkten an Bauteilen

Soweit sieht doch erst einmal alles logisch aus und jeder kann dieses Beispiel erst einmal nachbauen, ich denke da werden Sie mir Recht geben?

Aber wie sieht es den aus, wenn ich Ihnen nun Abbildung 5 zeige?

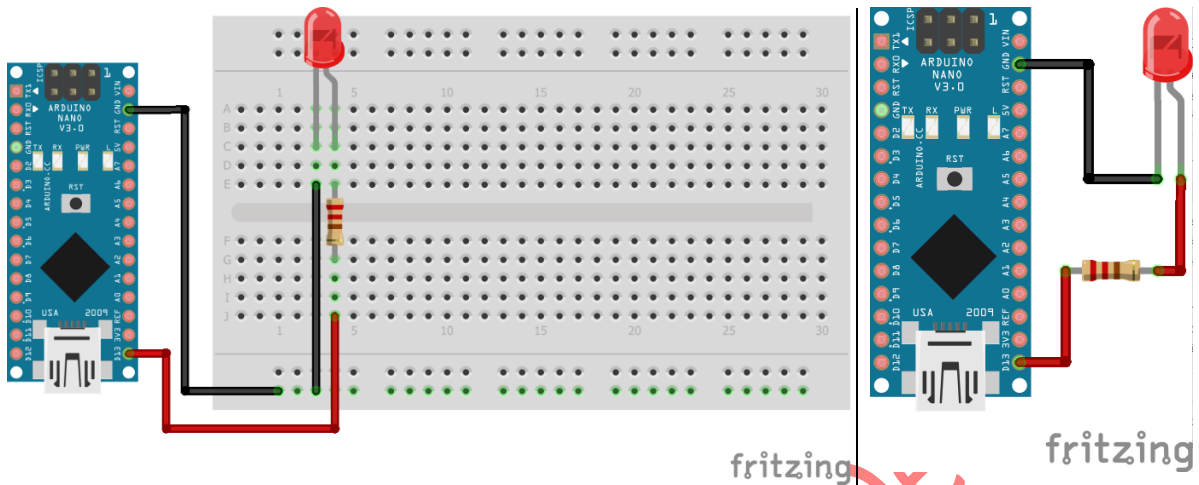


Abbildung 5: Beispiel Blink auf einem Steckbrett und ohne Steckbrett

Das linke Bild ist für einen unerfahrenen Bastler direkt nachbaubar. Das rechte Bild würde zwar auch reichen, aber das linke macht direkt klar, was gefordert ist. Gleichzeitig werden Farben genutzt, damit die Zeichnung noch verständlicher ist. Nur graue Linien würde bei größeren Projekten zu Kurzschlüssen führen, da man doch mal schnell den Überblick verliert. Aus der Elektrik wird der Nullleiter (Ground) schwarz und die Phase (Spannung) rot dargestellt, prinzipiell ist die Farbwahl allerdings frei.

Zusammenfassend lassen sich also folgende Regeln definieren:

1. Nutze Bauteile, die für das Projekt relevant sind oder nutze vergleichbare Bauteile
2. Die Anordnung der Bauteile sollte auf einem Steckbrett erfolgen oder aber sinnvoll gewählt werden
3. Leiterbahnen dürfen nie Bauteile kreuzen
4. Leiterbahnen verlaufen immer rechtwinklig und nie quer
5. Einsatz von Farben, um Leiterbahnen besser zu unterscheiden
6. Leiterbahnen sollten nie andere Leiterbahnen kreuzen. Sollte das nicht möglich sein, sollten die Leiterbahnen verschiedene Farben haben
7. Nutze so viel Platz wie nötig und so wenig Platz wie möglich für eine ordentliche schematische Zeichnung.

Die erste schematische Zeichnung

Nachdem nun die Grundlagen und der Styleguide erklärt wurde, wollen wir uns der ersten schematischen Zeichnung widmen. Dazu werden Bauteile genutzt, die Fritzing schon von Haus aus mitbringt, da wir uns erst später mit dem Import von Bauteilen beschäftigen.

Folgende schematische Zeichnung wollen wir am Ende erhalten, siehe Abbildung 6.

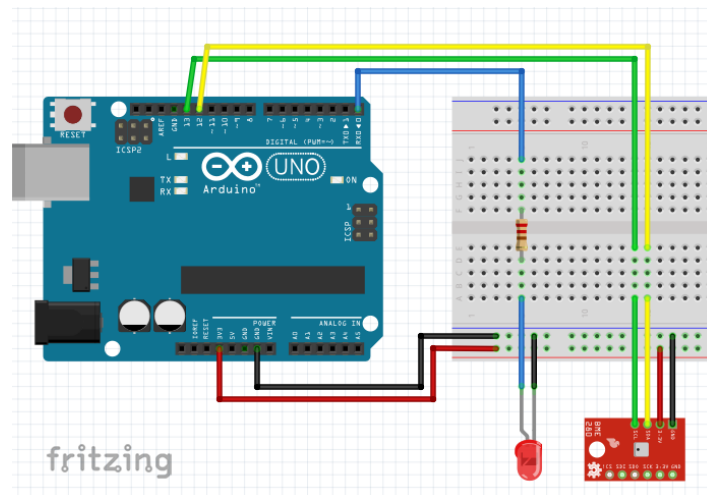


Abbildung 6: Fertige schematische Zeichnung

Die genaue Funktion ist erst einmal Nebensache, genauso wie die mögliche Programmierung des Arduino Unos, mit den verwendeten Bauteilen. Damit die schematische Zeichnung erstellt werden kann, muss Fritzing gestartet werden. Danach kann man direkt in die Steckbrett-Ansicht zu wechseln. Es ist hierbei egal, ob Sie den Shortcut Strg + 2 oder den Weg über die Menüleiste bzw. Schnellauswahl gehen. Sie sehen sofort, dass Fritzing standardmäßig das Steckbrett anzeigt, siehe Abbildung 7.

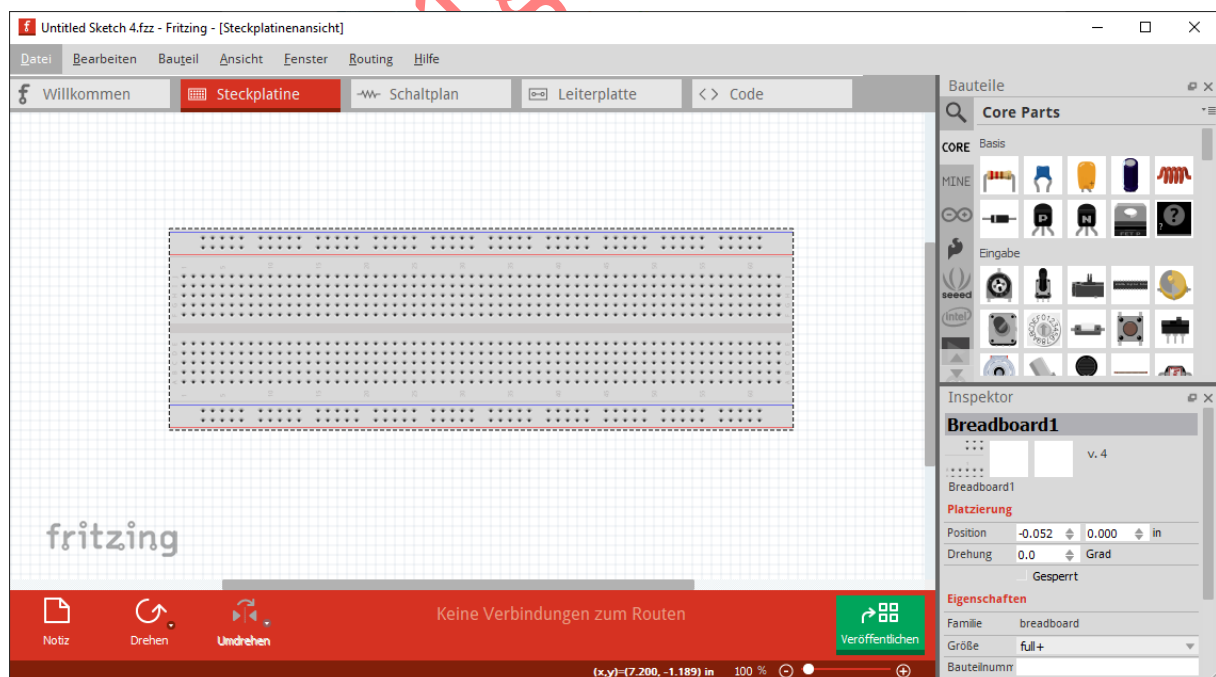


Abbildung 7: Ansicht Steckbrett

Sollte einmal nicht das Steckbrett angezeigt werden, so kommt nun eine erste Hürde für Sie. Fritzing-Bauteile haben englische Bezeichnungen, daher werden Sie bei der Suche (Lupensymbol in Bauteile-Fenster) mit dem Begriff „Steckbrett“ nichts finden. Daher suchen wir zunächst einmal nach „Breadboard“ und finde dazu eine kleine Liste mit unterschiedlichen Typen, siehe Abbildung 8.

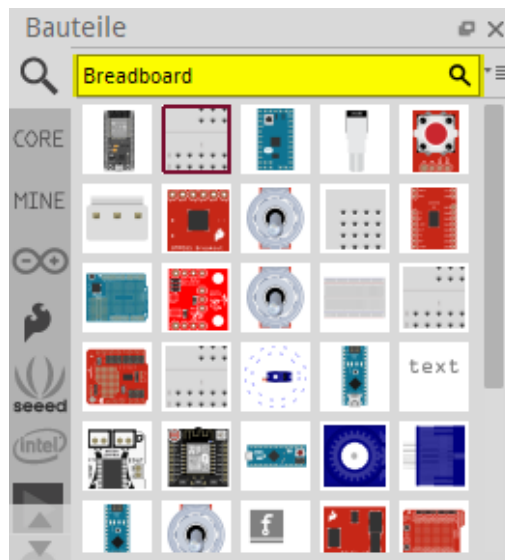


Abbildung 8: Steckbrett (Breadboard) suchen

Sie werden jetzt verwundert sein, warum bei der Suche nach einem Steckbrett, auf einmal noch andere Bauteile wie der oben links angezeigte ESP32-38Pin angezeigt wird. Nun das liegt daran, dass Fritzing bei der Suche von Begriffen nicht nur die Bauteilnamen prüft, sondern auch die Beschreibungen auf das Suchwort durchsucht. Daher finden Sie oft mehr zu einem Begriff, als nur das Bauteil was Sie nun gerade suchen.

Zum Einfügen eines weiteren Steckbrettes nutzen wir den zweiten Treffer, das zweite Symbol von links in der obersten Reihe, aus der Ergebnisliste und ziehen bei gedrückter linken Maustaste das Objekt in die Zeichenebene, siehe Abbildung 9.

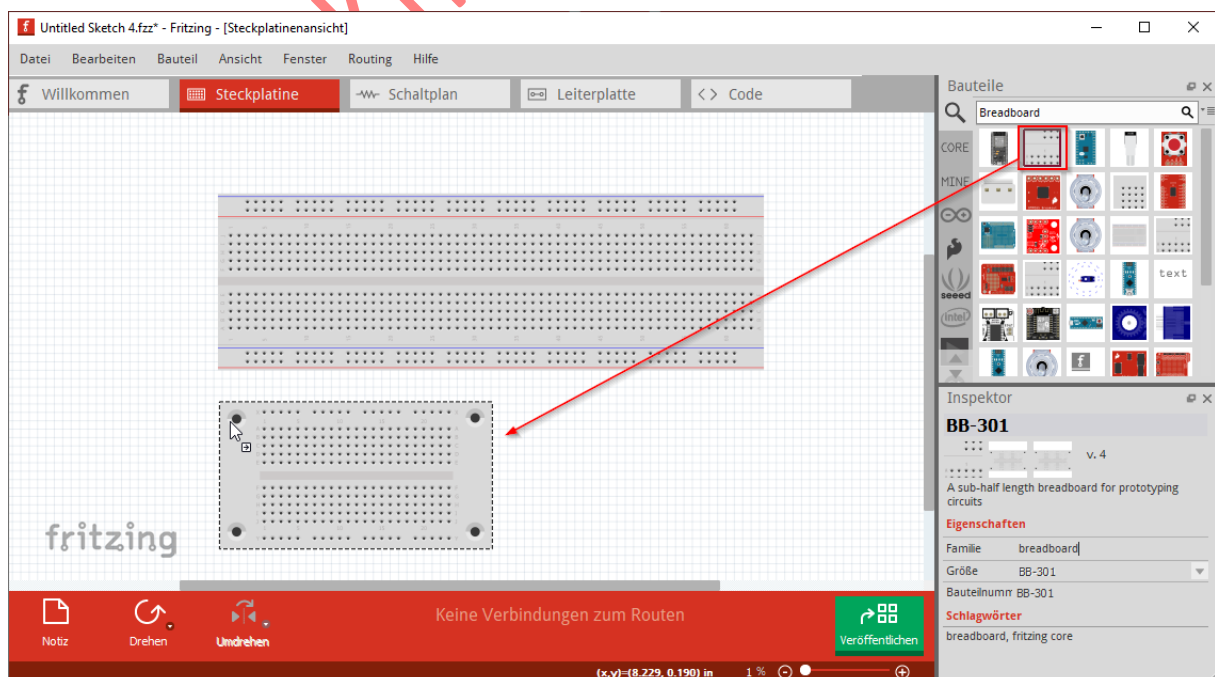


Abbildung 9: Steckbrett auf Zeichenebene anlegen

Sie sehen nun ein Steckbrett, aber es ist nicht das Steckbrett was gewollt war oder gerade in ihrer Bastelkiste liegt? Hier kommt das Inspektor-Fenster zum Einsatz. Einige Bauteile, wie auch das Steckbrett sind in einer Familie gruppiert und es ist möglich über die Eigenschaften diverse Parameter vom Bauteil zu verändern. Dies hat später auch Auswirkungen auf den Steckplan! Im Folgenden ändern Sie die Größe von BB-301 auf **full+** und Sie erhalten das Ergebnis wie in Abbildung 10.

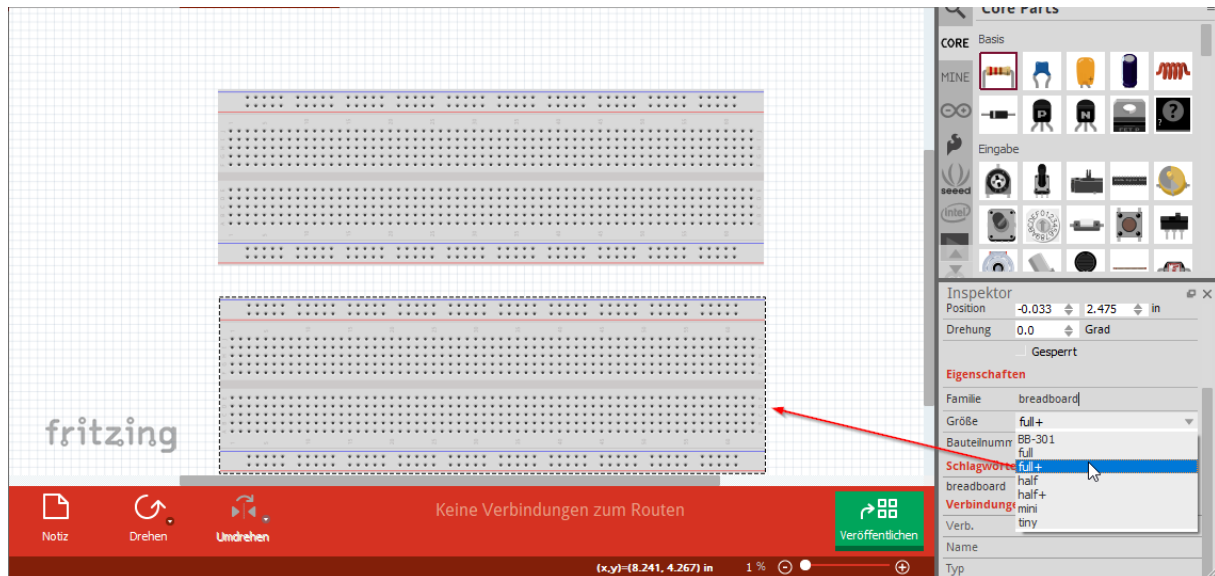


Abbildung 10: Steckbrett auf Größe full+ ändern

Zugegeben, das zweite Steckbrett ist für das Projekt gerade nicht nötig, daher wird es markiert und mit der Entf-Taste aus der Zeichenansicht gelöscht. Verdeutlichen wollte ich damit nur, wie Sie Bauteile auf die Zeichenebene bringen und wie Sie die Eigenschaften modifizieren können.

Setzen wir also die schematische Zeichnung für unser Beispiel-Projekt fort und ziehen folgende Bauteile in die Zeichenebene, wobei jetzt schon versucht wird, sinnvoll die Bauteile anzuordnen, siehe Abbildung 11.

1. Widerstand 220Ω, zu finden unter Core oder in der Suche unter „Resistor“
2. Rote LED, zu finden unter Core oder in der Suche unter „red LED“
3. BME280 Breakboard, zu finden unter SparkFun oder in der Suche unter „BME280“
4. Arduino Uno (Rev3), zu finden unter dem Arduino-Symbol oder in der Suche unter „Arduino Uno“

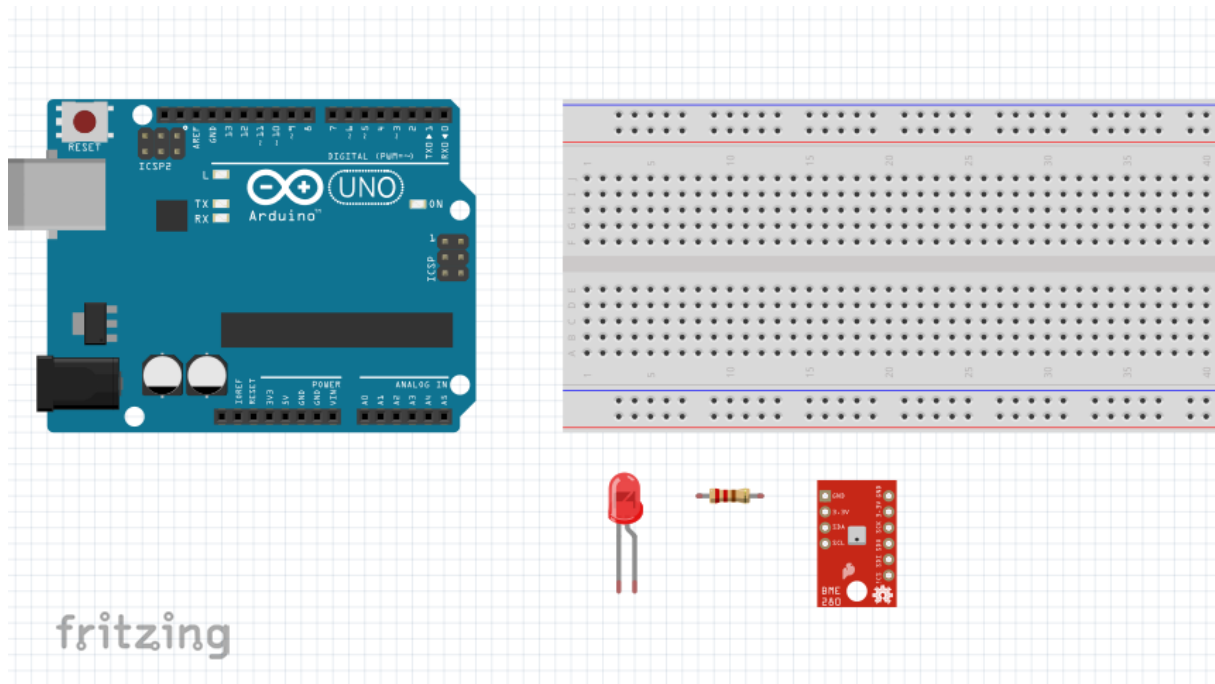


Abbildung 11: Benötigte Bauteile positionieren

Als erstes fällt auf, dass der Widerstand und auch der BME280 für den Anschluss falsch ausgerichtet sind. Beide Bauteile sollten um 90° gedreht werden und die LED am besten um 180°. Damit alles passt, wählen wir das Bauteil aus und öffnen mit der rechten Maustaste das Hilfsmenü. In der Kategorie drehen, wählen wir die gewünschte Drehbewegung aus, siehe Abbildung 12.

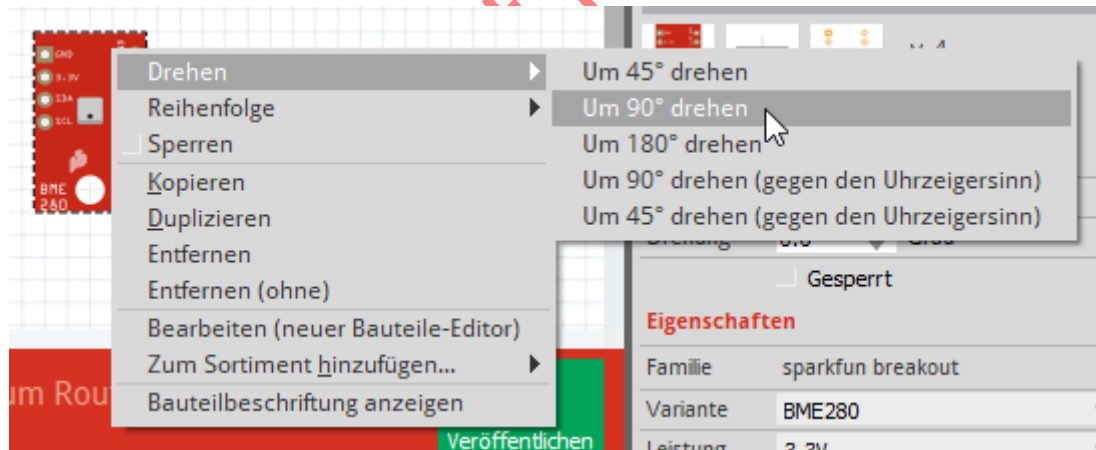


Abbildung 12: Bauteil neuorientieren

Sind alle Bauteile richtig orientiert, sollte die Zeichenfläche ungefähr wie in Abbildung 13 aussehen.

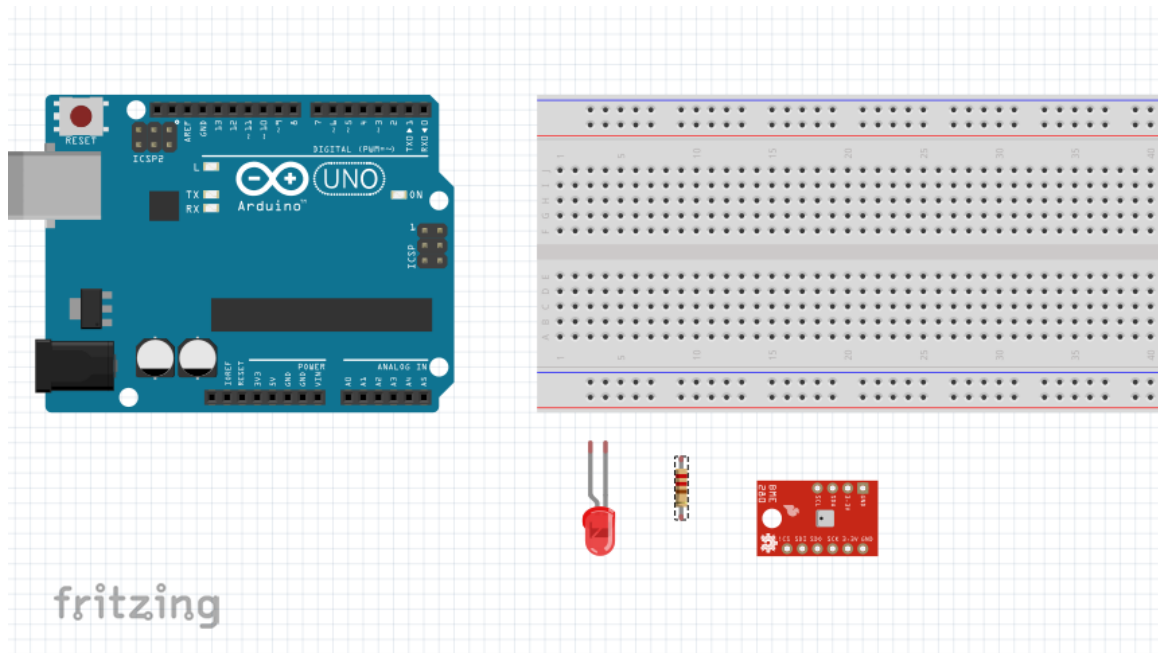


Abbildung 13: Bauteile richtig orientiert

Zunächst wird die LED und der Widerstand mit dem Arduino Uno verbunden. Dazu wird der Widerstand auf das Steckbrett platziert und der Widerstand entsprechend neu positioniert. Wie schon beim Einfügen eines Bauteils in die Zeichenfläche, greifen wir das Bauteil mit der linken Maustaste und bewegen es auf das Steckbrett. Sie werden feststellen, dass direkt ein kleines Symbol neben der Maustaste erscheint, das Ihnen mitteilt, dass das Bauteil ans Lochraster des Steckbrettes ausgerichtet wurde, siehe Abbildung 14.

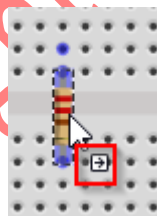


Abbildung 14: Widerstand auf Steckbrett ausgerichtet

Lassen Sie nun die linke Maustaste los, verfärben sich die Reihen auf Höhe vom Widerstand grün, siehe Abbildung 15

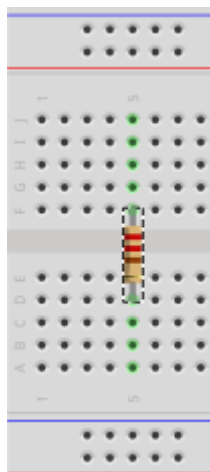


Abbildung 15: Widerstand auf Steckbrett gesteckt

Damit gibt uns Fritzing das Feedback, dass nun Verdrahtungen in der entsprechenden Reihe auch Auswirkungen auf den Widerstand und die angezeigten Stecklöcher hat. Gerade bei komplexeren Schaltungen ist das sehr hilfreich. Fritzing gibt aber kein Feedback, wenn wir versehentlich einen Kurzschluss bauen, hier müssen Sie den Überblick behalten. Zuletzt wird die Anode der LED mit dem Widerstand verbunden. Fahren Sie über ein Steckloch unterhalb vom Widerstand und Sie sehen ein neues Symbol neben dem Mauszeiger, siehe Abbildung 16.

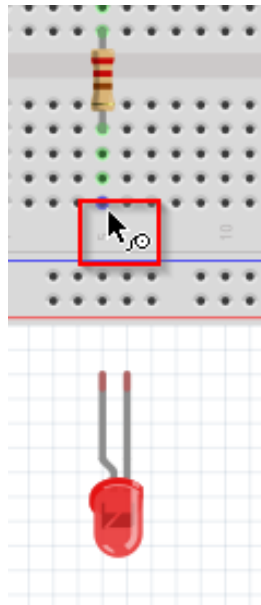


Abbildung 16: Verdrahtung auf Steckbrett anlegen

Damit gibt uns Fritzing das Feedback, dass wir an dieser Stelle eine Verdrahtung beginnen können. Da die LED mit dem Widerstand verbunden werden soll, halten Sie die linke Maustaste gedrückt und ziehen die neue Leiterbahn zu der Anode der LED und lassen los. Das Ergebnis sieht dann wie in Abbildung 17 aus.

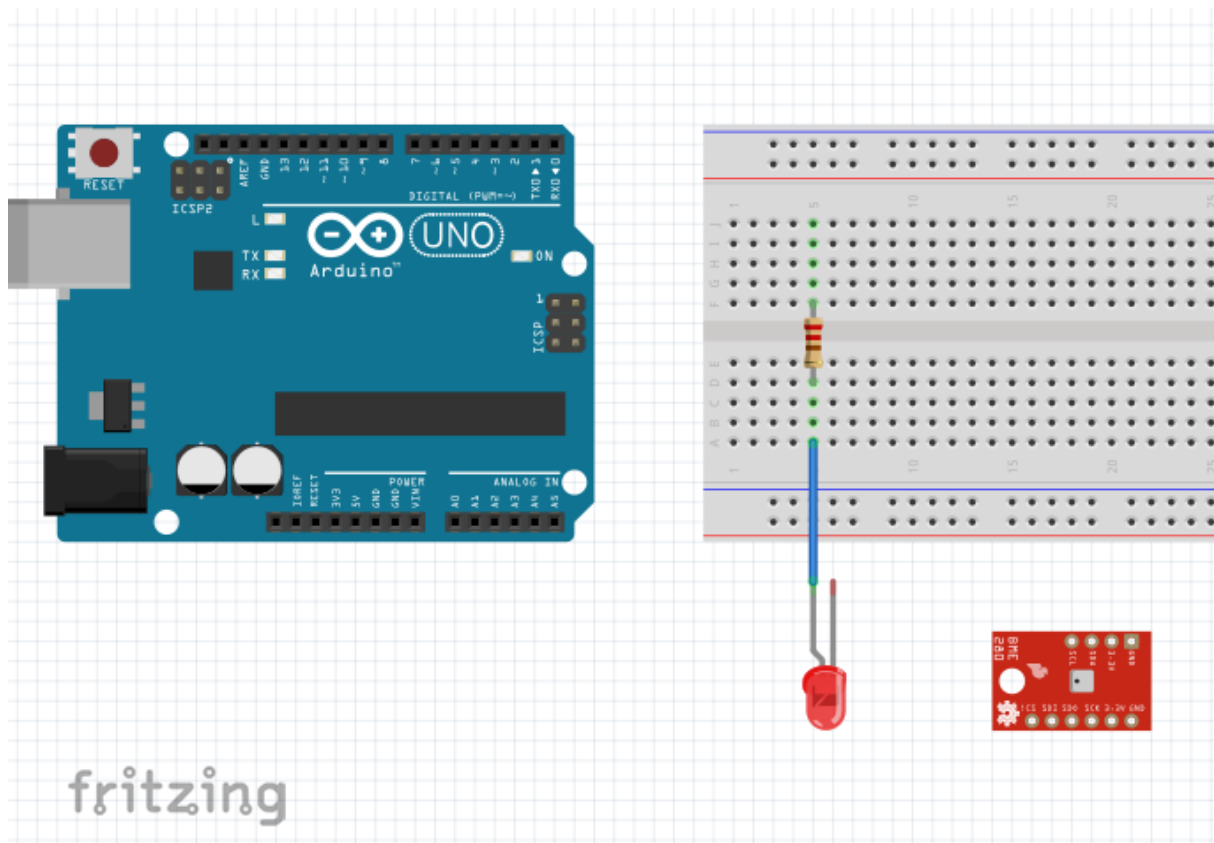


Abbildung 17: Leiterbahn zwischen Widerstand und LED

Mit dem jetzigen Wissen verbinden wir die andere Seite vom Widerstand mit dem Pin 0 am Arduino Uno und verbinden die Kathode der LED mit Ground der LED. Da das Steckbrett auf beiden Seiten einen Bus für die Spannungsversorgung hat, wird Ground vom Arduino mit dem Bus des Steckbrettes verbunden und von dort die Kathode der LED mit Ground verbunden. In Abbildung 18 sehen Sie nun das Zwischenergebnis der entsprechenden Steckverbindungen.

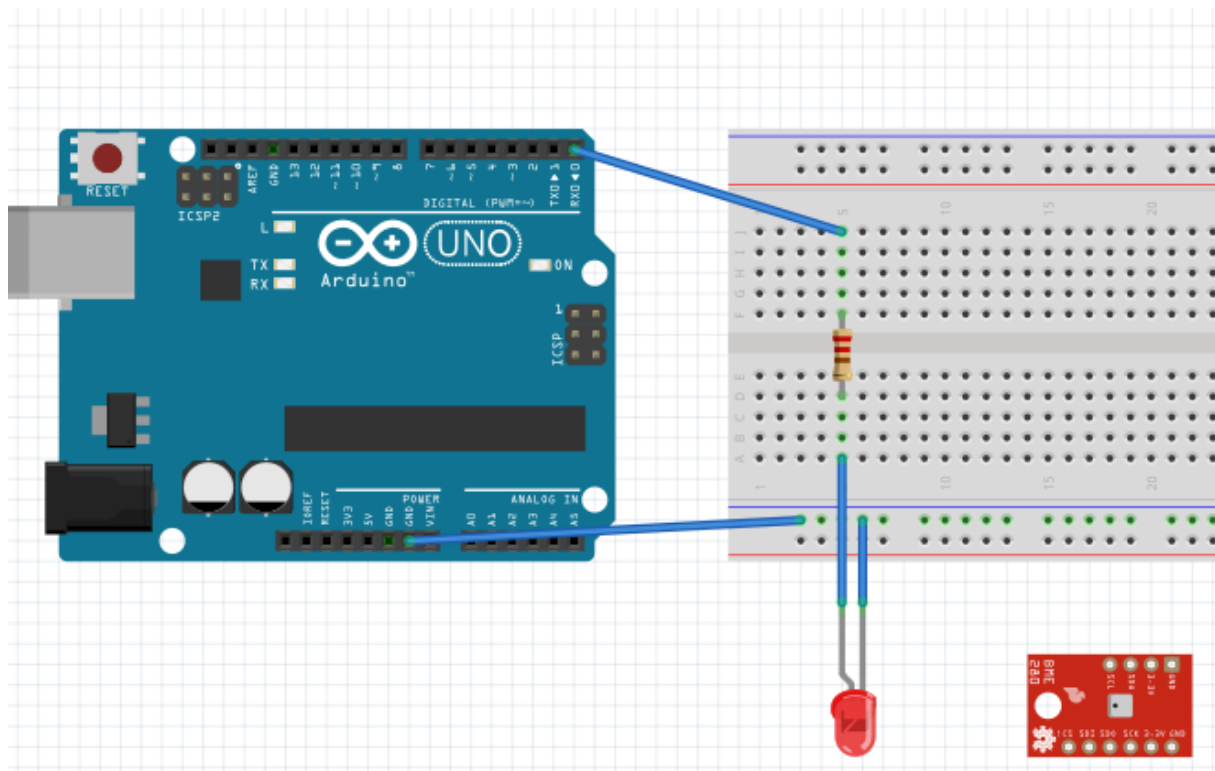


Abbildung 18: LED mit Arduino verdrahtet

Zum jetzigen Zeitpunkt werden die gleiche Farbe und der schräge Verlauf der zwei Verdrahtungen vernachlässigt, das wird zu einem späteren Zeitpunkt korrigiert.

Als nächstes wird der BME280 mit dem Arduino verbunden und auch die Stromversorgung hergestellt. Wie zuvor mit Ground, soll die Phase von 3,3 Volt über den Bus laufen. Gleichzeitig wird die Verdrahtung für die i2c-Kommunikation zwischen BME280 und Arduino Uno eingezeichnet. SCL wird an Pin 13 und SDA an Pin 12 vom Arduino Uno benötigt. Auf dem BME280 ist der entsprechende Pin beschriftet. Schaut man sich nun Abbildung 19 an, wird man feststellen, dass einige Probleme beim Aufbau der Schaltung entstehen können. Die Verdrahtungen sind teilweise schräg, die Farben der Verdrahtungen alle blau und bei der Power-Reihe vom Arduino ist nicht ersichtlich, was nun genau verdrahtet wurde.

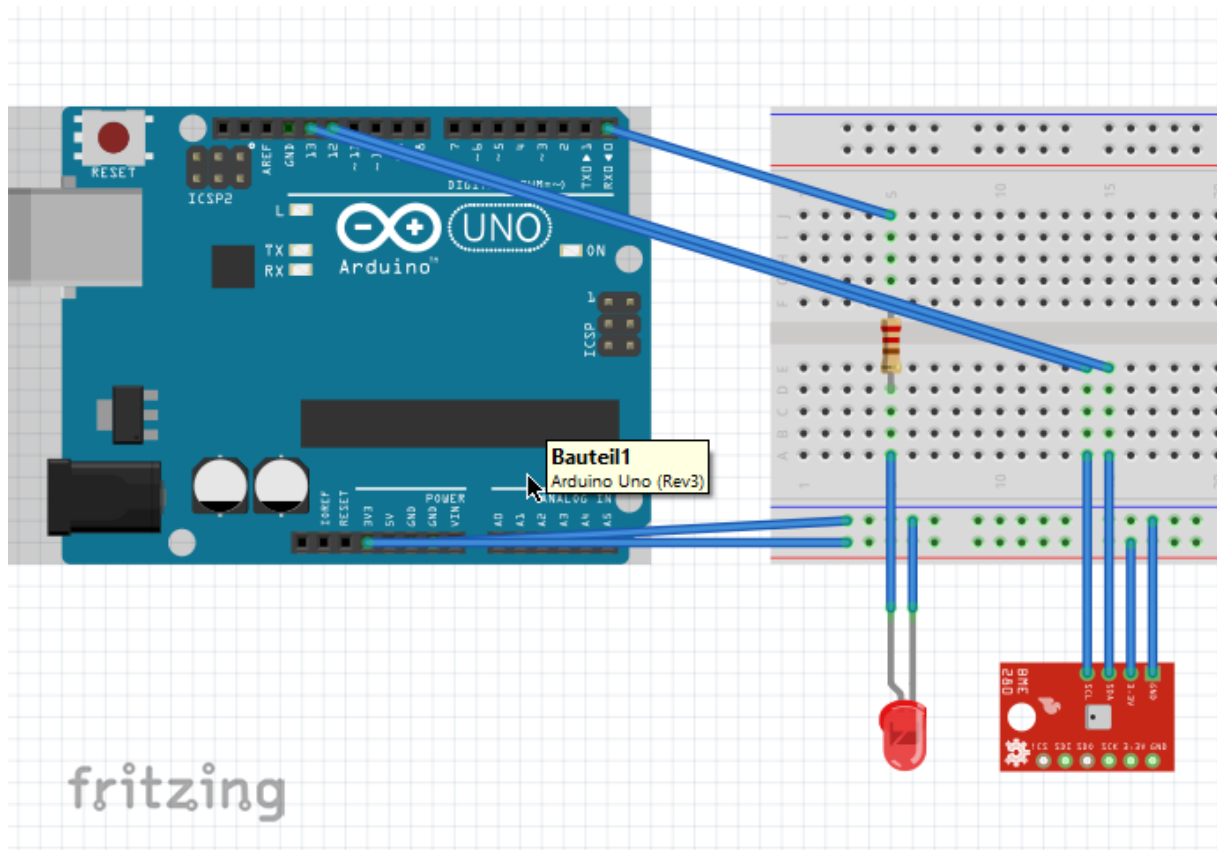


Abbildung 19: Fertige Verdrahtung der Bauteile mit Arduino

Nun kommen wir zum Styleguide, der vor dem Projekt erklärt wurde. Damit das Projekt für jedermann nachbaubar ist, muss nun das Feintuning der Zeichnung erfolgen. Zunächst werden alle Verdrahtungen rechtwinklig gezeichnet. Fährt man mit dem Mauszeiger über eine Verdrahtung, gibt Fritzing die Option, einen Biegepunkt einzufügen, siehe Abbildung 20.

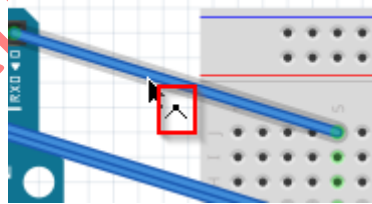


Abbildung 20: Biegepunkt bei Verdrahtung einfügen

Durch Drücken der linken Maustaste, sobald das Symbol erscheint und ziehen der Verdrahtung, kann der Biegepunkt an eine beliebige Position auf der Zeichenfläche gesetzt werden, siehe Abbildung 21.

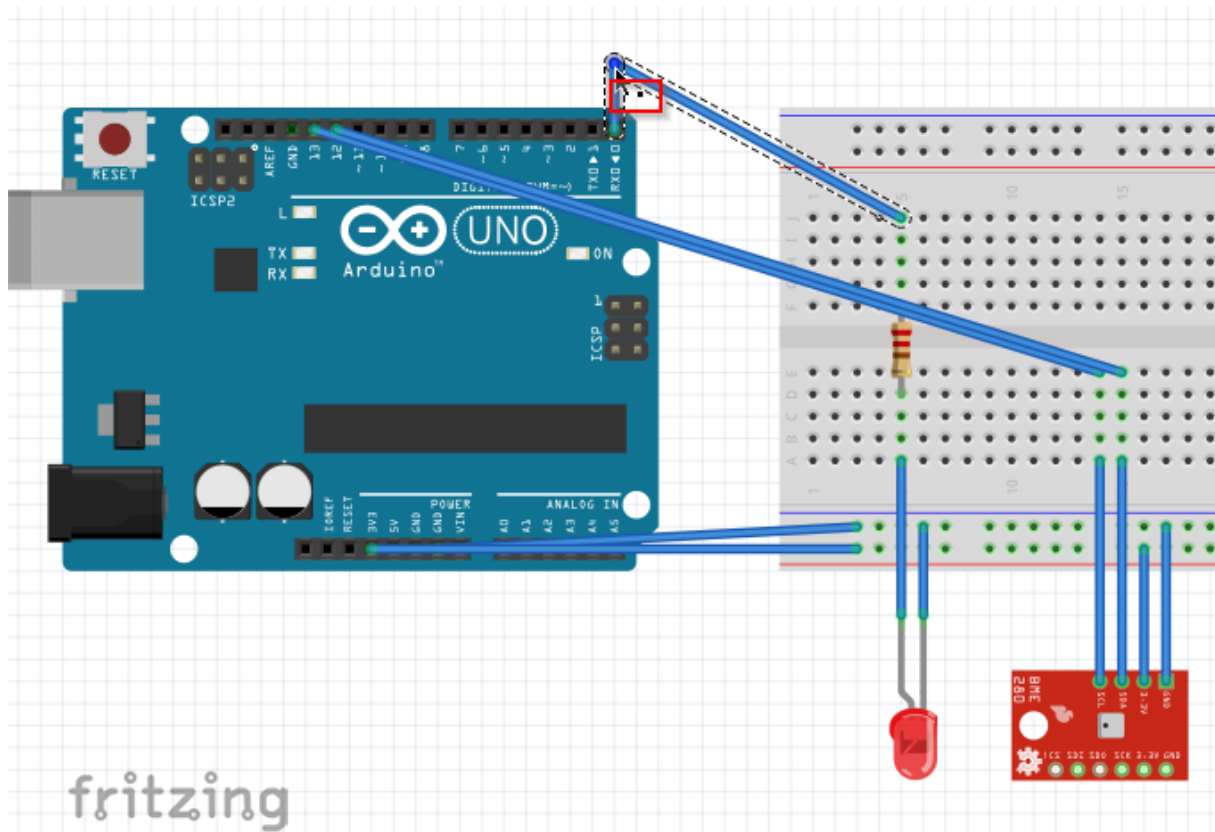


Abbildung 21: Neuer Biegepunkt für Verdrahtung

Gleichzeitig wird beim Loslassen der linken Maustaste ein weiteres Symbol neben dem Mauszeiger angezeigt. Der kleine schwarze Punkt, siehe Abbildung 21, sagt aus, dass Fritzing den Biegepunkt erkannt hat und somit dieser verschoben werden kann. Für unser Projekt werden nun alle Verdrahtungen rechtwinklig angeordnet, siehe Abbildung 22.

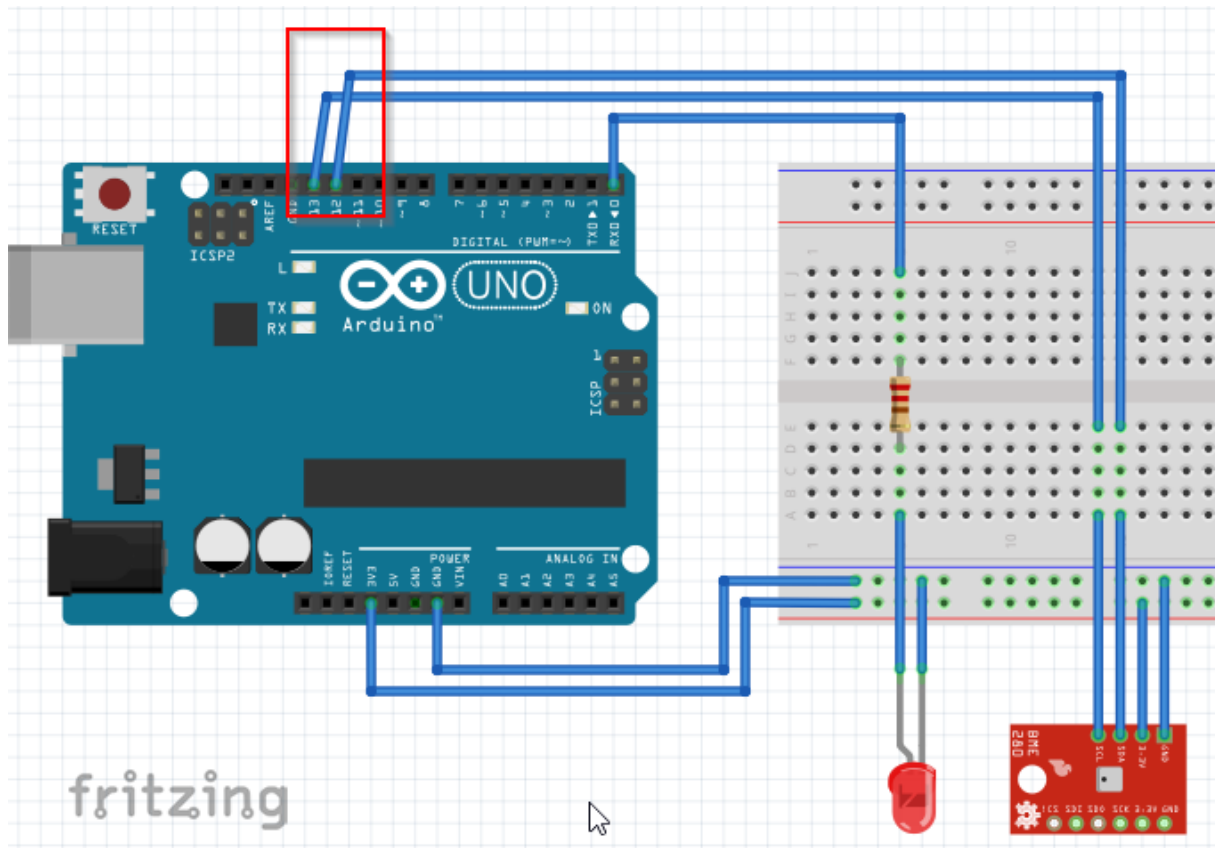


Abbildung 22: Verdrahtungen rechtwinklig zeichnen

Sie werden sich jetzt wahrscheinlich wundern, warum die Verdrahtung bei Pin 12 und 13 am Arduino Uno noch immer nicht rechtwinklig ist, obwohl dass das Bestreben ist. Die Antwort liegt in der Position der Buchsenleiste „Digital“ vom Arduino. Diese hält sich nicht an den Rasterabstand, was gut am Abstand zwischen Pin 7 und 8 im Vergleich zu VIN zu A0 zu erkennen ist. Daher passt ein Arduino Uno auch nicht auf ein Steckbrett und die Verdrahtungen können nicht rechtwinklig eingezeichnet werden. Dies könnte man nun beheben, würde aber eine Anpassung des kompletten Bauteilelements nach sich ziehen und wäre hier einfach zu viel. Jedoch stört nun noch die Farbe der Verdrahtungen, da diese alle blau sind. Damit der Plan noch verständlicher wird, soll nun die Farbe angepasst werden. Für die Phase und Ground soll es rot und schwarz sein, für SCL grün, SDA gelb und die Verbindung der LED-Anode, über den Widerstand, soll blau bleiben. Fahren Sie mit den Mauszeiger über eine Verdrahtung, bis diese grau umrandet wird und drücken Sie dann die rechte Maustaste. Im Hilfsmenü, siehe Abbildung 23, können Sie unter Drahtfarbe, die gewünschte Farbe auswählen.

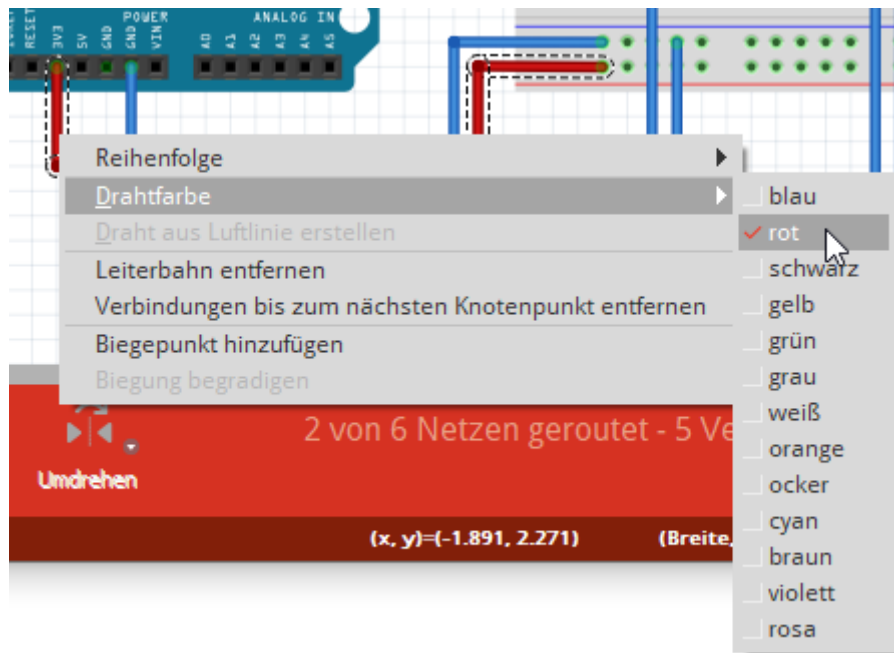


Abbildung 23: Farbe der Verdrahtung ändern

Wahlweise kann auch die Verdrahtung angewählt werden und über das Inspektor-Fenster die Farbe, unter dem Punkt Eigenschaften Farbe, geändert werden, siehe Abbildung 24.

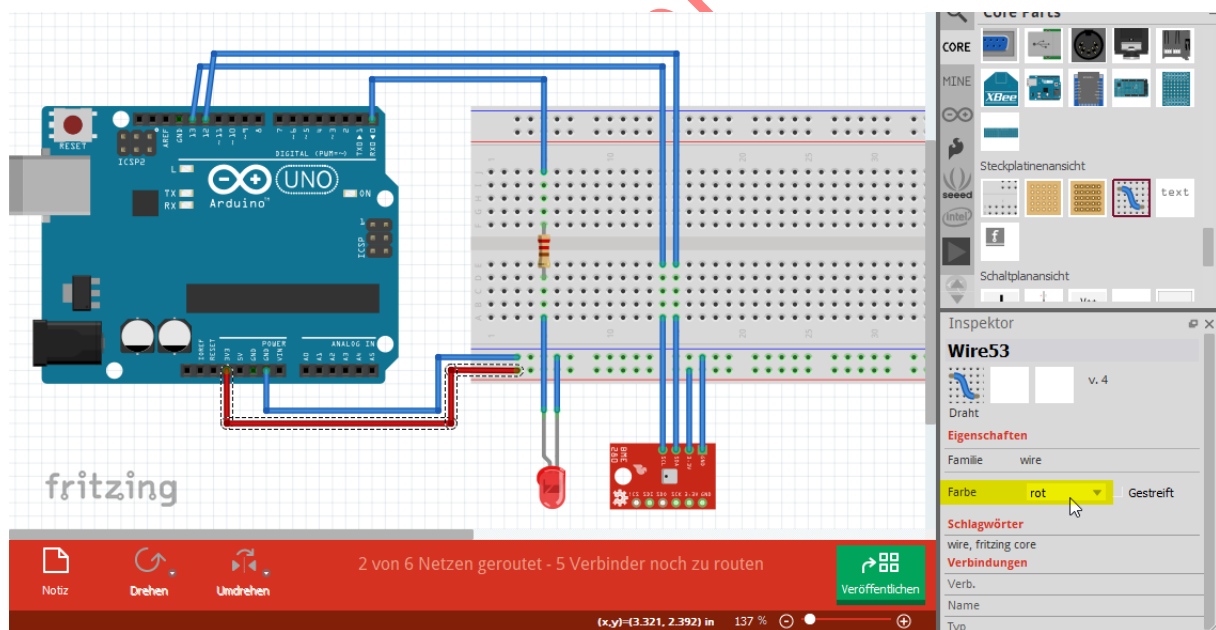


Abbildung 24: Verdrahtungsfarbe über Inspektor ändern

Bearbeiten Sie, mit einer der beiden Methoden, nun alle Verdrahtungsfarben. Das Ergebnis sollte ungefähr wie in Abbildung 25 aussehen.

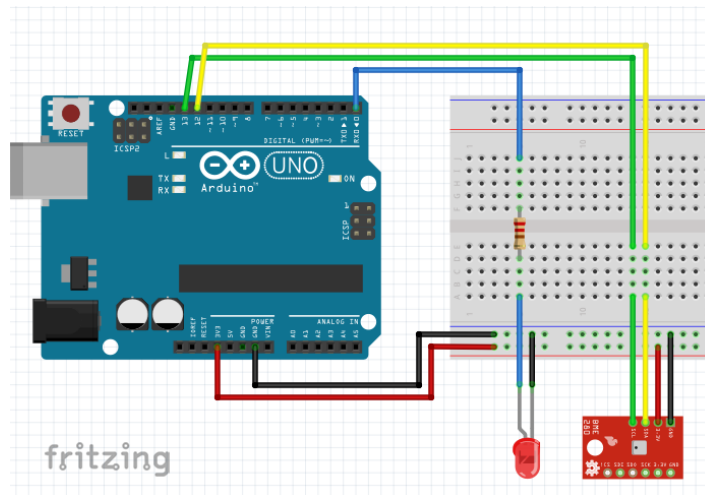


Abbildung 25: Fertige schematische Zeichnung

Damit haben Sie eine einfache schematische Zeichnung zu einem kleinen Projekt hergestellt. Im Grunde ist der Weg bis hierher gar nicht so schwer und für jeden leicht zu bewältigen. Ich hoffe Sie hatten Spaß daran und freuen sich schon auf Teil 2, wenn aus der schematischen Zeichnung ein Schaltplan und eine Leiterplatte wird.

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