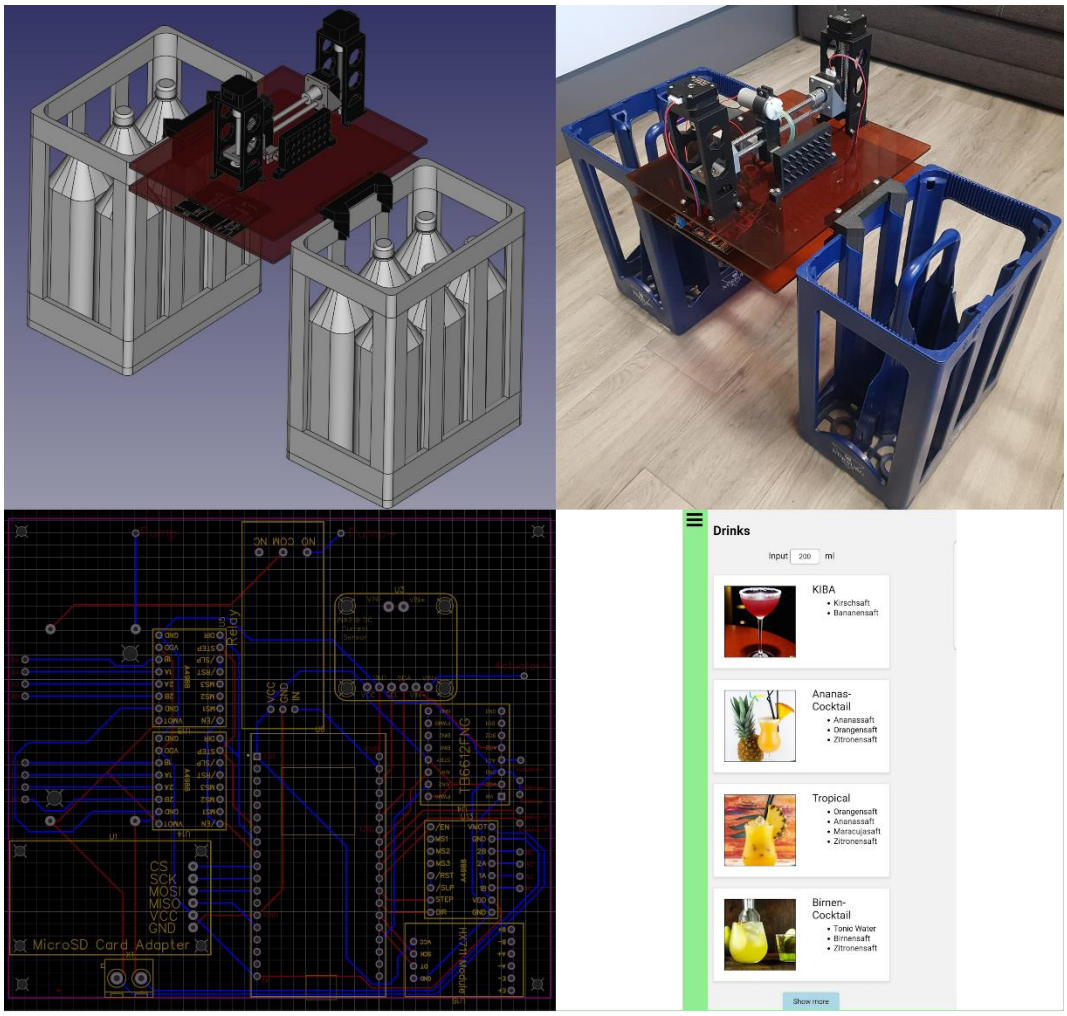
# Projekt:

# Automatischer Barkeeper 2.0




**Vom Prototypen zum Produkt**

von

Nick Fischer

Betreuungslehrer: Herr Nebe

**St. Michael Gymnasium, Bad Münstereifel**

# Inhaltsverzeichnis



# Abbildungsverzeichnis

# 1. Projektidee

Ziel des gesamten Projektes ist es, einen automatischen Getränkemixer zu entwickeln, der zuverlässig Flüssigkeiten mit Hilfe eines innovativen Pumpsystems befördern und mit Hilfe eines Browsers einfach und benutzerfreundlich bedient werden kann. Das innovativen Pumpdesign, welches es ermöglicht teils auch korrosive Flüssigkeiten zu transportieren, ohne dass diese dabei durch die Pumpe laufen müssen, konnte bereits in der letztjährigen Version erfolgreich umgesetzt werden. Dadurch wird die Pumpe nicht beschädigt und die Lebensdauer des Gerätes steigt signifikant. Auch die Bedienung über ein beliebiges Smartphone oder Tablet wurde erfolgreich implementiert. Dabei wurden Funktionen programmiert, welche es ermöglichen einen schon voreingestellten Drink auszuwählen oder einen selbst kreierten sich mixen zu lassen. Anpassungen und eigene Kreationen konnten zudem gespeichert werden und somit zu einem späteren Zeitpunkt erneut abgerufen werden.

Es gab jedoch noch umfangreiches Potential für Verbesserungen und die Umsetzung von einem Prototypen hin zu einem verkaufsfertigen Produkt. Diese werden im Folgenden beschrieben.

# 2. Ziele und Projektüberblick

Um die Ziele für dieses Jahr definieren zu können, ist es erstmal wichtig zu wissen was der Barkeeper bisher schon alles kann und was somit noch fehlt. Der Barkeeper ist schon in der Lage, per Knopfdruck automatisiert Getränke zu mixen und auszugeben. Dies bedeutet, dass bereits ein funktionierendes Pumpsystem als auch eine Benutzeroberfläche zum Steuern vorhanden ist, über welche Getränke ausgewählt und ausgegeben werden können. Auch die Herstellungskosten des Barkeepers konnten bereits stark gesenkt werden.

Wie das Regelwerk von Jugend Forscht nun besagt, muss ein Projekt, damit es noch einmal angemeldet werden und antreten darf, signifikante Änderungen vorweisen können und genau das war auch das oberste Ziel. Doch wie genau nimmt man solch signifikante Änderungen an einem schon vorhandenen Prototypen vor? Genau dies galt es herauszufinden. Als erstes habe ich mich noch einmal intensiv damit befasst, was die Kritikpunkte der Jury waren. Der Hauptkritikpunkt, der vorgebracht wurde, war, dass es zwar natürlich nett sei, den Barkeeper bei seiner Arbeit beobachten zu dürfen, jedoch der Barkeeper für den Praxisgebrauch zu langsam sei. Außerdem hatte ich mit dem Barkeeper zwar schon einen gewissen Grad der Skalierbarkeit vorzuweisen, jedoch konnte dieses System mit den bisherigen Anschlüssen noch durch eine andere Art von System, welches kostengünstiger wäre, ersetzt werden. Mit Skalierbarkeit ist hierbei die Menge der verfügbaren Anschlüsse für das Anschließen von Flaschen zum Mixen von Drinks gemeint. Dies hat mich dann zu meinen ersten konkreten Zielen geführt. Die Schnelligkeit zu erhöhen und die Skalierbarkeit zu verbessern. Als nächstes habe ich mir angeschaut, wie der Barkeeper bisher beschrieben wird. Und zwar als ein Prototyp und nicht als ein fertiges Produkt. Und dies sollte zum zweiten großen Aspekt des Barkeepers 2.0 werden, aus dem viele weitere kleinere und größere Ziele hervorgehen. Der Barkeeper sollte nun darauf ausgerichtet werden, nicht nur an diejenigen ausgeliefert zu werden, die ein DIY-Projekt haben wollen, sondern auch an die Endkonsumenten, welche den Barkeeper direkt nutzen möchten. Damit waren die nächsten Ziele den Barkeeper zuverlässiger zu gestalten und die Lebensdauer zu erhöhen. Außerdem mussten dafür jegliche Bugs, die bisher noch im Code aufgetreten sind, beseitigt und die Benutzererfahrung zum Bedienen des Barkeepers über die Webseite verbessert werden.

# 3. Software

## 3.1. Bilder

Die Bilder sind ein Aspekt, welcher schon vor dem Antreten auf dem Landeswettbewerb 2023 vorhanden war, jedoch noch nicht auf dem Regionalwettbewerb und damit ebenso wenig in der vorangegangenen schriftlichen Arbeit erwähnt wurden. Aus diesem Grund bin ich auch damals auf dem Landeswettbewerb 2023 noch nicht wirklich darauf eingegangen, weswegen dieser hier noch einmal kurz Erwähnung finden sollte. Spezifisch meine ich hierbei die Bilder der voreingestellten Drinks. Denn natürlich könnte man hierfür einfach selbst gemachte Bilder verwenden. Jedoch würden solche mit Hinblick auf die spätere Nutzung des Barkeepers durch Endverbraucher nicht sehr professionell erscheinen. Daher war die erste Idee, Bilder aus dem Internet zu verwenden. Das Problem mit solchen Bildern ist aber leider, dass viele dieser Bilder einer Lizenz unterliegen, welche das Benutzen der Bilder für kommerzielle Zwecke untersagt. Herauszufinden, welches Bild im Internet nun einer solchen Lizenz unterliegt, ist oft sehr schwer und intransparent gestaltet. Hier kommt Künstliche Intelligenz ins Spiel. Denn anstatt nach Bildern zu schauen, welche vielleicht nicht einer solchen Lizenz unterliegen, habe ich schon Anfang 2023 Künstliche Intelligenz verwendet, um mir die Bilder für die voreingestellten Drinks zu generieren. Der Vorteil davon ist, dass die Bilder nicht durch eine Lizenz geschützt sind, da diese so nicht existieren und damit einzigartig sind. Um die besten Bilder und damit die besten Ergebnisse erzielen zu können, habe ich verschiedenste Modelle benutzt. So habe ich auch momentan sehr bekannt gewordene Text-zu-Bild KIs benutzt, wie zum Beispiel „Stable Diffusion“, „Midjourney“ und „Dall-E“, aber auch weitere Webseiten, welche den Zugriff auf Sammlungen von verschiedenen Text-zu-Bild Modellen ermöglichen, verwendet.

## 3.2. SD-Karte

Da kurz nach dem Landeswettbewerb 2023 die Sommerferien gestartet sind, konnten ich mich in dieser Zeit nicht auf die großen Ziele konzentrieren, da diesen hauptsächlich ein hardwarelastiger Teil zu Grunde liegt. Jedoch wollte ich auch in den Sommerferien nicht untätig sein und habe mich somit hier erst einmal auf das konzentriert, was auch Zuhause gut umgesetzt werden konnte. Dies waren programmiertechnische Umsetzungen. Angefangen mit der Einbindung einer SD-Karte in die Software. Die Nutzung einer SD-Karte hatte zum Ziel mehr voreingestellte Drinks einbauen zu können, da der Flash-Speicher der Platine eine Limitierung auf 4 Megabyte bedeutete. Die Herausforderung hierbei war, das Senden der Bilder auf Anfrage der Benutzeroberfläche hin an den asynchron laufenden ESP zu implementieren, da es dafür keinerlei Dokumentation gab. Nach dem Ändern des Speicherorts, von dem die Bilder abgefragt werden sollten, konnten die ersten Bilder über die SD-Karte auf der Webseite angezeigt werden. Jedoch hat sich in den darauffolgenden Monaten herausgestellt, dass eine Implementierung der SD-Karte doch um ein Vielfaches schwerer werden würde, als gedacht. Denn durch die Verwendung der SD-Karte wurden Bilder mit einer viel zu hohen Quote gar nicht oder nur fehlerhaft geladen. Manchmal stürzte der ESP dadurch aber auch vollständig ab. Da dies ein Problem war, welches nichts mit dem eigentlichen Code zu tun hatte, sondern mit den internen Rechenstrukturen des ESP selber, müsste man sich zur Lösung des Problems die gesamte Programmierung des ESPs anschauen und vollständig verstehen. Da dies für das eigentliche Ziel, welches ich damit erreichen wollte, viel zu lange gedauert hätte und es nicht einmal wirklich zu den sekundären Zielen gehörte, habe ich mich dafür entschieden dieses Feature erst einmal rauszunehmen und später vielleicht wieder zu implementieren. Wie bereits gesagt, gehörte der Punkt eine SD-Karte zu nutzen aber nicht einmal wirklich zu den sekundären Zielen. Das eigentliche

sekundäre Ziel war es, eine größere Auswahl an Drinks bieten zu können. Um dieses Ziel trotz des kleinen Flashspeichers des ESP von gerade einmal 4 Megabytes erreichen zu können, habe ich mir Gedanken dazu gemacht, ob die Bilder, welche ich verwende, überhaupt so groß sein müssen. Da die Bilder hauptsächlich auf einem Handy geladen werden und damit nur sehr klein angezeigt werden, kann entsprechend die Qualität und damit auch die Größe der einzelnen Bilder heruntergestuft werden, ohne dass die Bilder dadurch merkbar schlechter aussehen. Dafür habe ich die Bilder mehrmals komprimiert und nach jedem Komprimierungsvorgang mit dem Originalbild verglichen. Dies habe ich für jedes Bild solange wiederholt, bis ein erkennbarer Qualitätsunterschied ersichtlich wurde und bin dann auf die vorherige Komprimierung zurückgegangen. Mit dieser Methodik konnte ich die Größe der Bilder von 712 kB auf 268 kB senken. Dies bedeutet, dass ich mit der Nutzung desselben Speicherplatzes nun 2,6-mal so viele Bilder speichern konnte. Konkret bedeutet dies, dass anstatt 78 voreingestellten Getränken nun 217 voreingestellte Drinks gespeichert werden können.

Obwohl die SD-Karte aktuell nicht benötigt wird, wurde auf der selbst designten Platine, welche weiter unten (siehe 4.2 Platine) noch ausführlich erklärt wird, ein Platz für den SD-Kartenleser vorgesehen, so dass es auch in Zukunft noch möglich ist eine SD-Karte einzubinden.

## 3.3. Bug fixes

Bug Fixing war nach dem Regional- und Landeswettbewerb und mit Hinblick auf das Erreichen der Serienreife der Maschine eine der wichtigsten Aufgaben. Insbesondere, da ich auf dem Regional- und Landeswettbewerb erstmals die Möglichkeit hatte den Barkeeper mit einem weiten Spektrum an Benutzern und deren Umgang mit dem Barkeeper, sowie Smartphones zu testen und so auch zum ersten Mal sehen konnte, wie der Barkeeper darauf reagieren würde, wenn viele Personen gleichzeitig mit dem Gerät verbunden sind. Diese Testphase war von großer Bedeutung, um Bugs festzustellen, aufzuschreiben und schlussendlich nach den Wettbewerben Zuhause in Ruhe beheben zu können. Beispiele solcher Bugs sind zum Beispiel, dass das Mixen eines Custom Drinks nicht mehr funktioniert hatte, weil der gesamte Barkeeper dabei abgestürzt ist, oder auch kompliziertere Bugs, die aufgefallen sind, weil manche Nutzer komplexere Kombinationen von Entscheidungen getroffen hatten, die dann schlussendlich zu einem Fehler führten.

## 3.4. Akkulaufzeit

Ein weiteres Problem, welches sich besonders auf dem Wettbewerb bemerkbar gemacht hatte, war die Akkulaufzeit. Denn dadurch, dass ich den Barkeeper, insbesondere auf dem Landeswettbewerb, drei Tage lang eingeschaltet hatte, ist mir aufgefallen, dass der Akku sehr schnell leer ging und nach einem Tag schon nur noch zwei von drei Balken aufwies. Nachdem ich mich mit dem Problem auseinander gesetzt habe um zu schauen, wie ich die Akkulaufzeit verbessern könnte, ist mir aufgefallen, dass solange sich der Akku in seiner Halterung befand und der ESP an den Strom angeschlossen war, die Motoren immer unter voller Spannung standen, selbst wenn diese momentan nicht benutzt wurden. Auf Grund dessen habe ich den Code so optimiert, dass wenn nicht gerade ein Drink gemixt wird, die Motoren ausgeschaltet bleiben und somit in dieser Zeit keinen Strom mehr verbrauchen. Dies hat die Akkulaufzeit signifikant verbessert. Trotzdem habe ich hiermit noch weiter experimentiert. Mir fiel auf, dass die verschiedenen Motoren während gerade ein Drink gemixt wurde, weiterhin immer unter Spannung standen. Dies galt auch für die Motoren, welche gerade nicht direkt angesteuert und bewegt wurden. Dies bedeutet, dass die Motoren, welche für das heranfahren an die Pumpe zuständig waren, auch unter Spannung standen, während gerade nur die Pumpe lief. Deswegen habe ich zusätzlich implementiert, dass Motoren ausschließlich unter Spannung stehen, wenn diese auch explizit angesteuert werden. Alle anderen

Motoren bleiben dabei auch während des Mixvorgangs ausgeschaltet. Mit diesen Verbesserungen im Code musste ich von der Implementierung, welche ich kurz nach Ende des Landeswettbewerbes vorgenommen habe, bis jetzt, trotz weiterem intensivem Testen des Barkeepers, den Akku nicht ein einziges Mal neu aufladen und der Akku hat bisher nur zwei der drei Balken verloren.

## 3.5. Anzahl der verbundenen Netzwerknutzer

Ein weiteres Problem, welches mir schon auf den Wettbewerben aufgefallen war, bestand darin, dass nach dem sich mehrere Menschen mit dem Barkeeper verbunden hatten, die nächsten Personen, welche den Barkeeper austesten wollten, Probleme hatten sich mit dem Barkeeper zu verbinden. Auf dem Landeswettbewerb habe ich, um dieses Problem zu umgehen, den Barkeeper immer, wenn das Problem auftrat, kurz vom Strom getrennt und wieder verbunden. Meine erste Vermutung war die, dass die Anzahl der Personen, welche sich gleichzeitig mit dem Netzwerk des Barkeepers verbinden konnten, limitiert war. Nachdem der Wettbewerb vorbei war und ich mich mit dem Problem befassen konnte, hat sich meine anfängliche Vermutung dann auch als richtig erwiesen. Der Barkeeper limitiert die Anzahl der Benutzer, welche sich mit dem eigens aufgebauten Netzwerk verbinden können, standardmäßig auf vier Geräte. Mit einer weiteren Codezeile, welche die Anzahl der Benutzer definiert, ist es mir gelungen diese Anzahl auf 16 Geräte zu erhöhen. Weitere Benutzer gleichzeitig damit zu verbinden scheint hardwareseitig limitiert zu sein.

## 3.6. Nutzungsformate

### 3.6.1. Party Version

Doch was ist aber, wenn man sich zum Beispiel auf einer Party befindet, auf der mehr als 16 Personen sind? Dafür ist die Party Version gedacht. Dies ist eine abweichende Version des Codes, welche einfach auf den ESP übertragen werden kann. Der Grundgedanke für eine solche erweiterte Version kam mir dadurch, dass ich den Barkeeper auf meinem eigenen Abiball demnächst aufstellen möchte. Was macht die Party Version nun anders als die normale Version des Barkeepers und wofür braucht man eine abweichende Version, wenn es anscheinend doch eine Lösung dafür gibt, dass mehr als 16 Personen im Netzwerk sind? Und das ist der Punkt. Es können nicht mehr als 16 Personen im Netzwerk sein. Deswegen musste ich mir eine andere Lösung für das Problem überlegen. Dafür habe ich auf die Lösung zurückgegriffen, mit der ich ursprünglich das Problem gelöst hatte, in dem ich den Barkeeper immer neu startete. Dies hat auch einigermaßen gut funktioniert, jedoch hieß dies einerseits, dass der Barkeeper immer wieder manuell vom Strom getrennt werden musste und andererseits, dass sich viele Smartphones einfach wieder automatisch mit dem Netzwerk verbunden haben, da das Passwort für das WLAN immer dasselbe blieb. Mein Lösungsansatz hierzu ist es nach dem Mixen eines Drinks, das Passwort für das WLAN zu resetten. Dies löst die Probleme, dass man manuell den Barkeeper vom Strom trennen muss und dass dies dadurch auch softwaretechnisch regelbar ist, da nach Änderung des Passwortes des WLANs, alle Geräte aus diesem WLAN automatisch entfernt werden. Die Smartphones können sich auch nicht automatisch wieder mit dem WLAN verbinden, da sich das Passwort nun geändert hat. Um dies zu erreichen, generiert der ESP eine zufällige 8-stellige Zahl, welche nach dem Mixen eines Drinks als neues Passwort festgelegt und in der seriellen Konsole zum Eintippen in das Handy ausgegeben wird. Man könnte diese Version noch mit einem LCD-Display erweitern, welches dann das momentane Passwort anzeigt. Wie bereits erwähnt, wird das Passwort immer nach dem Mixen eines Drinks geändert, jedoch ist dies nicht die einzige Situation, in der sich das Passwort ändert, denn es kann natürlich auch dazu kommen, dass sich 16 Personen mit dem Netzwerk verbunden haben, sich dann aber niemand von diesen 16 Personen einen Drink mixt. Um dieses Problem zu lösen, wird,

nachdem sich die erste Person mit dem Netzwerk verbindet, ein Timer für 15 Minuten gestartet. Nach Ablauf dieser Zeit und ohne dass gerade ein Drink gemixt wird, wird das Passwort für das Netzwerk auch resettet. Eine weitere Funktion der Partyversion ist, dass keine Drinks größer als 250ml erstellt werden können, um zu vermeiden, dass sich jemand einen Spaß erlaubt und das Glas überläuft.

### 3.6.2. Bar Version

Die Bar Version übernimmt alle Funktionen der Partyversion und baut darauf auf. Diese Version ist dafür gedacht, wie der Name schon impliziert, in einer Bar oder allgemein kommerziell eingesetzt zu werden. Zusätzlich zu den in der Party Version bereits erklärten Funktionen, erfordert das Mixen eines Drinks hier einen einmaligen, zufällig generierten Code, welcher dann vom Verkäufer zur Verfügung gestellt werden kann. Um die Codes einsehen zu können, gibt es in dieser Version in den Einstellungen einen Button, welcher nach Betätigung ein Pop-Up erscheinen lässt, welches den Nutzer auffordert, das voreingestellte Passwort „bar“ einzugeben. Danach wird man aufgefordert, ein eigenes Passwort festzulegen. Nachdem dieses festgelegt ist, kann man mit dem festgelegten Passwort nun auf die Codes zugreifen.

## 3.7. User Interface

D Die aber ohne Zweifel visuell erkennbarste und eine der wichtigsten softwaretechnischen Änderungen besteht aus der kompletten Überarbeitung des User Interfaces. Um das Design für die Benutzer so benutzerfreundlich wie möglich zu gestalten, habe ich mir genaustens angeschaut, wie die Benutzer während des Bedienens des Barkeepers auf dem Wettbewerb damit umgegangen sind. Dabei ist mir erst einmal aufgefallen, dass es sehr nervig ist, erst durch alle voreingestellten Drinks durchklicken zu müssen, wenn der Benutzer beispielsweise nur einen individualisierten Drink erstellen möchte. Aus diesem Grund habe ich für die voreingestellten Drinks ein separates Menü erstellt. Weiterhin ist mir aufgefallen, dass es lange dauert, wenn der Benutzer mit den Pfeiltasten durch alle voreingestellten Drinks navigieren muss und dies von vielen als nervig empfunden werden könnte. Deswegen habe ich mich entschieden dieses Mal, statt durch das Menü für die voreingestellten Drinks wieder mit Pfeilen navigieren zu können, ein Dropdown Menü zu implementieren. So kann der Benutzer sich mehrere Drinks gleichzeitig anschauen und daraufhin seine Auswahl treffen. Bei der Implementierung dieses Features ist mir jedoch aufgefallen, dass es öfters zu Abstürzen kam. Das Problem hierbei war, dass mit dem Dropdown Menü alle sieben Bilder der sieben voreingestellten Drinks gleichzeitig geladen wurden. Nach meinem Verständnis, kamen diese Abstürze daher, dass ein sogenanntes Wacht-Dog Feature, welches die Antwortzeit des Barkeepers überwacht um sicherzustellen, dass sich dieser nicht aufgehangen hat oder sich in einem endlos-loop befindet, den Barkeeper, falls dieser zu lange für eine Antwort benötigt, neustartet. Dadurch dass die Antwort 7 Bilder gleichzeitig beinhaltet, scheint dies zu lange zu dauern und den Watchdog auszulösen. Um dieses Problem zu umgehen, habe ich ein Feature eingebaut, welches sicherstellt, dass nie mehr als vier Bilder gleichzeitig gesendet werden. Weiterhin ist mir aufgefallen, dass es suboptimal ist, wenn nur der entsprechende Name des Drinks angezeigt wird, jedoch nicht

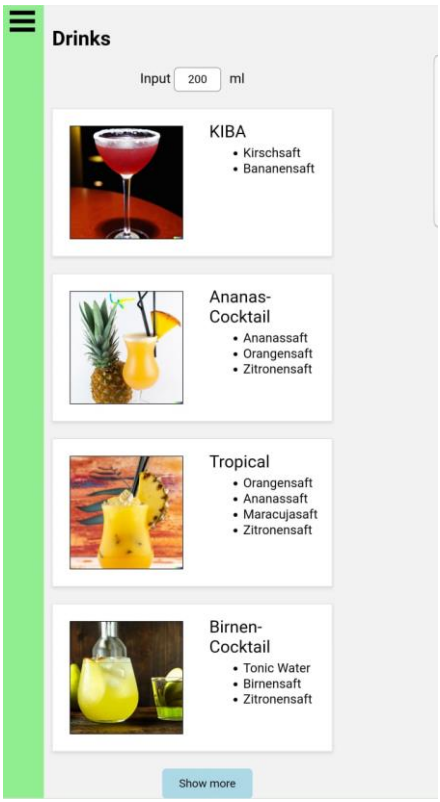



*Abbildung 1: Implementierung des Dropdown Menüs*

die dazugehörigen Zutaten. Die Lösung hierzu kam in Form eines steckbriefartigem Designs, welches den Namen und darunter die zugehörigen Zutaten listet, mit dem Bild seitlich des Titels und der Zutaten. Außerdem ist mir aufgefallen, dass es besser wäre, wenn die voreingestellten Drinks skalierbar wären. Dies bedeutet, dass man von nun an, bevor man auf den voreingestellten Drink geht, einstellen kann, wie groß der Drink sein soll und der Barkeeper dann automatisch die Zutaten entsprechend skaliert. Die letzte Veränderung, die ich vorgenommen habe, ist wahrscheinlich auch die umfangreichste. Für das Hauptmenü wurde immer noch das Design verwendet, bei welchem man mit Pfeilen durch die einzelnen Menüpunkte navigiert. Dies ist nicht nur, wie bereits festgestellt, langsam und unnötig, sondern benötigt auch eine komplett eigene programmierte HTML Seite mit dem einzigen Zweck zu den eigentlich gewollten Funktionen navigieren zu können. Es wäre viel angenehmer, wenn man mit nur zwei simplen Klicks auf die gewollte Funktion kommen könnte, welche man auch wirklich auswählen möchte. Dafür habe ich die komplette Navigation, um zu den einzelnen Funktionen zu kommen, überarbeitet, das Hauptmenü gelöscht und ein Seitenmenü programmiert. Wenn man nun den Barkeeper startet kommt der Nutzer standardmäßig auf die Seite der voreingestellten Drinks, da die meisten Anwesenden auf dem Wettbewerb es präferiert haben, einen schon voreingestellten Drink zu wählen, als sich einen eigenen Drink zu mixen. Von dort an kann man in der linken oberen Ecke das Seitenmenü öffnen und bekommt dann auf einen Blick alle Funktionen angezeigt, zu denen gesprungen werden kann. Dies eliminiert zusätzlich die Notwendigkeit eines Return Buttons auf jeder Seite und ist nicht nur um ein vielfaches benutzerfreundlicher, sondern sieht auch angenehmer aus als grelle gelbe Pfeile.


*Abbildung 2: Navigation durch Seitenmenü*

# 4. Hardware

## 4.1. Z-Achse

Einer der wohl wichtigsten Neuerungen des Barkeepers 2.0 ist die Z-Achse. Denn diese ist notwendig um eines der primären Ziele zu erreichen. Die Skalierbarkeit des Barkeepers. Diese Neuerung ist so fundamental, dass alle nachfolgenden Teilaspekte der Hardware auf dieser aufbauen. Erst nach Umsetzung der Z-Achse konnten alle weiteren Hardwareanpassungen umgesetzt werden. Denn ohne die Z-Achse standen die verschiedenen Pinbelegungen für die Platine noch nicht fest, die Maße des Barkeepers waren noch unklar, weswegen noch kein Gehäuse, keine Schlauchführung und kein

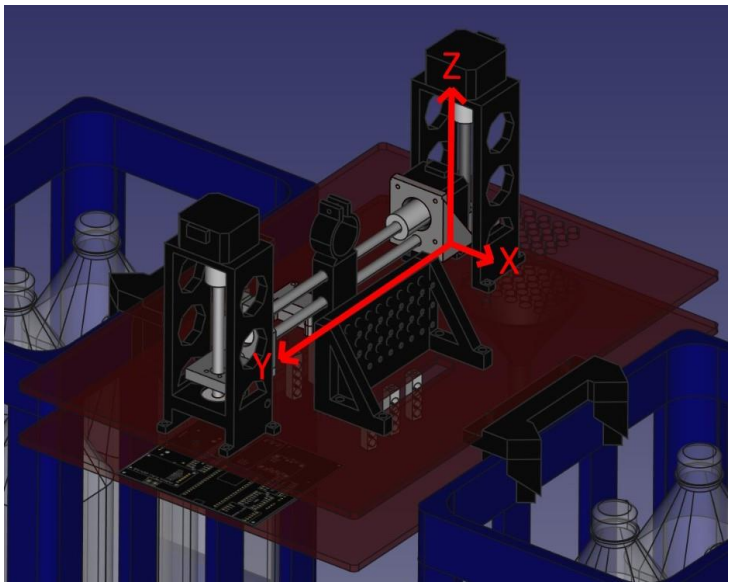



*Abbildung 3: Orientierung der Achsen*

Aufsatz für Kästen gebaut werden konnte. Somit galt es als oberste Priorität, die Z-Achse so schnell wie möglich konzipiert und gebaut zu bekommen.

Der schwierigste Aspekt am Bau der Z-Achse war, sich das grundlegende Design auszudenken. Denn wie sollte eine solche Z-Achse überhaupt aussehen? Dafür gab es viele verschiedene Möglichkeiten. Eine Möglichkeit bestand darin, das vorherige Design nahezu vollständig zu übernehmen. Dies würde bedeuten, dass der Verteiler weiterhin beweglich gewesen wäre. Dabei hätte sich der Verteiler in der X-Achse und der Druckluftgeber in der YZ-Achse bewegt. Dies war auch das erste Design, welches verfolgt wurde. Die erste Idee bestand darin, einen Motor für die Y-Achse zu verwenden, welcher statt einfach nur dem Andocker zu bewegen einen weiteren Motor, welcher nach oben gerichtet ist, bewegt. Dieser Motor wäre dann in der Lage, den Andocker hoch und runter zu fahren. Nachdem mit der Modellierung dieses Designs angefangen wurde, sind direkt mehrere Probleme daran aufgefallen. So bestand ein Problem dabei darin, dass der Verteiler und somit auch alle anderen Teile, welche mit dem Verteiler in Verbindung stehen, um einiges höher gesetzt werden müssten. Dies würde den Materialverbrauch und damit auch die Kosten erhöhen. Das größte Problem dieser Iteration bestand jedoch darin, dass dadurch, dass die Z-Achse relativ hoch sein müsste und je höher der zu erreichende Anschluss lag, immer mehr Hebelwirkung auf den Andocker, damit auf die Z-Achse und damit auch auf die Y-Achse ausgeübt werden würde. Ein weiteres Problem bestand darin, dass diese Version, wenn überhaupt umsetzbar, durch die schon bei wenigen Etagen hohe Hebelwirkung, nur geringfügig größer skalierbar sein würde und damit dem Primärziel der Skalierbarkeit widersprechen würde. Aufgrund all dieser Schwachstellen wurde dieses Design aufgegeben. Die nächste Idee bestand darin, zwei Motoren für die Z-Achse zu verwenden. Diese Motoren sollten außen angebracht werden und die gesamte Y-Achse hoch und runter fahren. Durch dieses Design wird eine signifikant höhere Stabilität gewährleistet und das Modell ist sehr einfach und sehr stark skalierbar, womit es perfekt für diesen Anwendungszweck geeignet ist.

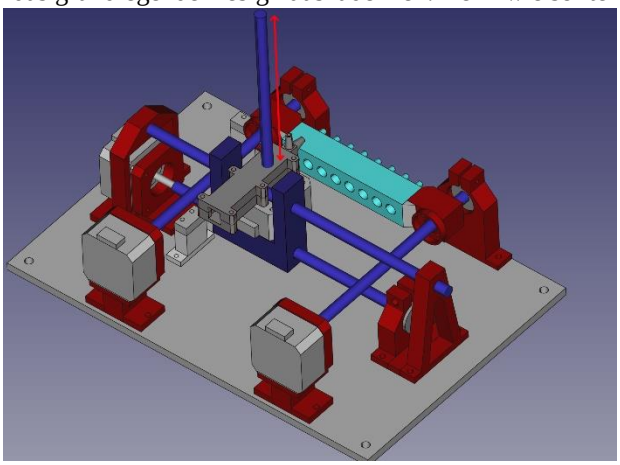


*Abbildung 4: Erste Designidee der Z-Achse*

### 4.1.1. Aktuator

Trotzdem war ich mit dem Design noch nicht zufrieden, denn dies würde höhere Kosten bedeuten, da so zwei weitere Motoren verbaut werden müssten. Bei der Analyse der Anforderungen an den Barkeeper wurde festgestellt, dass dieser in der X-Achse nicht weit nach vorne fahren, sondern in der X-Achse nur ein bis zwei Zentimeter überbrücken muss. Dies führte zu der Idee, einen Aktuator für  die X-Achse zu verwenden, denn durch einen Aktuator musste nur noch ein Motor, welchen den Aktuator antreibt, für die X-Achse verwendet werden. Außerdem muss sich damit der Verteiler selbst nicht mehr bewegen, wodurch die Zuverlässigkeit erhöht werden kann, da bei dem Verteiler keine beweglichen Teile mehr erforderlich waren. Außerdem werden hierdurch keine langen Spindeln, keine Kugellager und keine dazugehörigen Halterungen mehr benötigt.

#### 4.1.1.1. H-Brücke

Das erste Problem beim Verwenden eines Aktuators war die Steuerung. Denn während bei einem Stepper Motor, wie dem Nema17 die Richtung in welche dieser sich drehen soll, in der Software geändert werden kann, kann

der Aktuator immer nur in eine Richtung drehen, abhängig davon, wie herum dieser gepolt ist. Die Lösung dieses Problems bestand aus der Verwendung einer sogenannten H-Brücke. Dafür wurde die Dual H-Bridge TB6612FNG verwendet. Eine H-Brücke besteht im Prinzip aus 4 internen Schaltern, welche einzeln angesteuert werden können. Durch das korrekte Schalten dieser Schalter ist es möglich die Stromrichtung des Ausgangsstromes umzukehren und somit zu erreichen, dass wenn vorher der Motor links herum drehte, dieser danach rechts herum dreht.

#### 4.1.1.2. Präzision

Ein weiterer Nachteil eines Aktuator besteht darin, dass dieser einen einfachen Gleichstrommotor verwendet und keinen Schrittmotor. Wie der Name eines Schrittmotors hier bereits impliziert, dreht sich ein solcher Motor in präzisen Schritten, womit genau bestimmt werden kann wie viele Schritte sich dieser bewegen soll. Ein Gleichstrommotor besitzt kein solches Feature, wodurch es mit einem solchen Motor nicht möglich ist präzise zu bestimmen, wann genau dieser stoppen soll. Der erste Lösungsansatz zu diesem Problem bestand aus der Verwendung eines weiteren Endstops, jedoch hier nicht in Form eines Hebels, da dieser für diese spezifische Anwendung zu groß gewesen wäre, sondern eines Knopfes, welcher sich über der Spitze des Aktuators befindet. Bei jedem ausfahren des Aktuators fährt dieser mit heraus. Wenn dieser dann gegen den Verteiler stoßen würde, würde der Aktuator wissen, dass er nun stoppen muss. Dieses Methodik würde aber zwei weitere Probleme aufwerfen. Denn einerseits müsste dadurch an den Aktuator selber noch ein Teil angebaut werden, welches den Endstop hält, wodurch der Aktuator selber größer werden würde und designtechnisch auch nicht mehr so schön aussähe und vor allem müssten hierbei dann aber zwei Endstops verwendet werden, um an den Verteiler und an den Aktuator anzudocken. Dieses Problem wäre zwar umgehbar gewesen, indem man die durchschnittliche Zeit stoppt, bis der Aktuator wieder komplett eingefahren wäre, jedoch wäre dies unpräzise gewesen. Zudem würde der Aktuator jedes Mal für einen Moment durchdrehen oder erst gar nicht weiterdrehen, da dieser schon komplett eingefahren ist und immer noch unter Spannung steht. Dies würde sich wahrscheinlich auf die Lebensdauer und die damit verbundene Haltbarkeit des Barkeepers auswirken. Um die daraufhin angepasste endgültige Lösung zu verstehen, sollte erst einmal erklärt werden, dass wenn ein Motor Widerstand erfährt, dieser erst einmal versucht, stärker dagegen anzukämpfen, indem dieser mehr Strom zieht. Dies wird durch einen Anstieg in der Stromstärke bemerkbar. Und genau dieses Prinzip wird sich bei dieser Lösung zu Nutze gemacht, indem mit dem Ina219 die Stromstärke gemessen wird. Dabei wird pro Sekunde die Stromstärke in 25 Millisekunden Takten gemessen und wenn innerhalb einer gewissen Zeitspanne die Stromstärke höher als 90 Milliampere liegt, wird der Motor ausgeschaltet, da dies dann bedeutet, dass dieser am Verteiler angekommen ist. Das Messen über eine gewisse Zeit ist dafür da, um Schwankungen, welche während dem Herausfahren auftreten können, zu mitigieren. Für das Zurückfahren muss über eine gewisse Zeitspanne die Stromstärke mehr als 80 Milliampere betragen. Hierbei kann ein kleinerer Wert verwendet werden, da es beim Andocken an den Verteiler wichtig ist, dass genug Druck aufkommt, um einen luftdichten Verschluss zwischen Aktuator und Verteiler gewährleisten zu können. Zusätzlich wird ein äußerer Dichtungsring verwendet, um dies sicherstellen zu können.

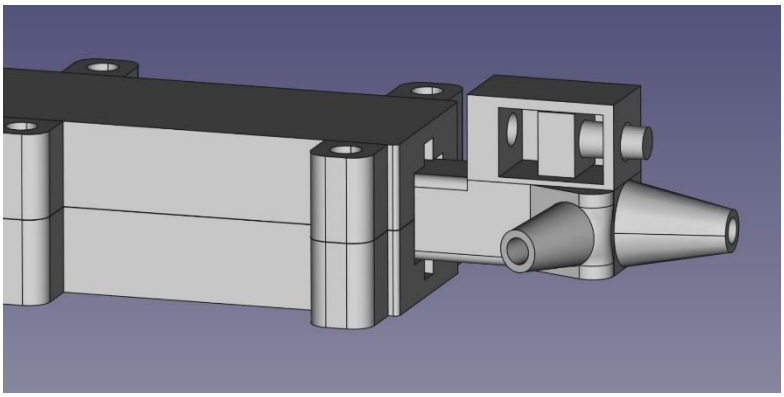


*Abbildung 5: Endstop des Aktuators*

#### 4.1.1.3. Spindel

Ein weiteres Problem lag in der mit dem Motor mitgelieferten Spindel. An diesem Punkt ist es sinnvoll, erst einmal zu erklären, wie ein Aktuator überhaupt funktioniert. Bei einem Aktuator dreht der Gleichstrommotor eine Spindel. Auf der Spindel befindet sich eine Mutter, welche sich drehen lässt. Die Mutter ist nun aber in ihrer Rotation durch das Teil, welches sich nach vorne und zurück schieben soll, fixiert. Wird die Spindel durch den Motor gedreht, bewegt sich dadurch die Mutter nach vorne oder nach hinten, wodurch sich gleichzeitig auch das Teil, in dem die Mutter fixiert ist, nach vorne oder nach hinten bewegt. Das Problem mit der mitgelieferten Metallspindel war nun, dass diese relativ viel Abstand vom Motor zum Anfang der eigentlichen Helix hatte. Dies bedeutete, dass sehr viel mehr Platz als eigentlich nötig benötigt wurde und damit der Aktuator in seiner Länge noch länger wäre, als dies sowieso schon der Fall ist. Dies hätte dazu geführt, dass das Gehäuse noch größer sein müsste und entsprechend das Gewicht, sowie die mit der größeren Länge verbundenen Hebelwirkung angestiegen wäre. Die Lösung dieses Problems kam in Form eines neuen 3D-Druckers. Dem Bamboo Lab X1 Carbon. Dies ist einer der modernsten und präzisesten 3D-Drucker für Konsumenten. Durch diese neue Errungenschaft war es möglich, eine besser passende Spindel und die dazugehörige Mutter selbst zu designen und dann zu drucken. Die nun gedruckte Spindel war nicht nur um einiges kürzer und damit passender für unsere Anwendung geeignet, sondern auch dadurch, dass diese aus PLA bestand und nicht aus Metall, auch um einiges leichter.

*Abbildung 6: Größe der Spindel und Mutter in Relation zu einem Finger*

#### 4.1.1.4. Länge

Aber auch mit einer verkürzten Spindel hat die Länge immer noch Sorgen bereitet und so wurde sich eine Lösung durch einen um 90° invertierten Motor mit einer Zahnradübersetzung überlegt. Damit wäre der Aktuator noch kürzer und entsprechend weniger Hebelwirkung vorhanden gewesen. Jedoch hat sich beim 3D-Drucken der Zahnräder gezeigt, dass dieses Prinzip durch die höheren Reibungskräfte ineffizienter und damit schwer umzusetzen gewesen wäre. Außerdem hätten mehr mechanische Teile einen unnötig größeren Verschleiß bedeutet, was wiederum mit dem primären Ziel der

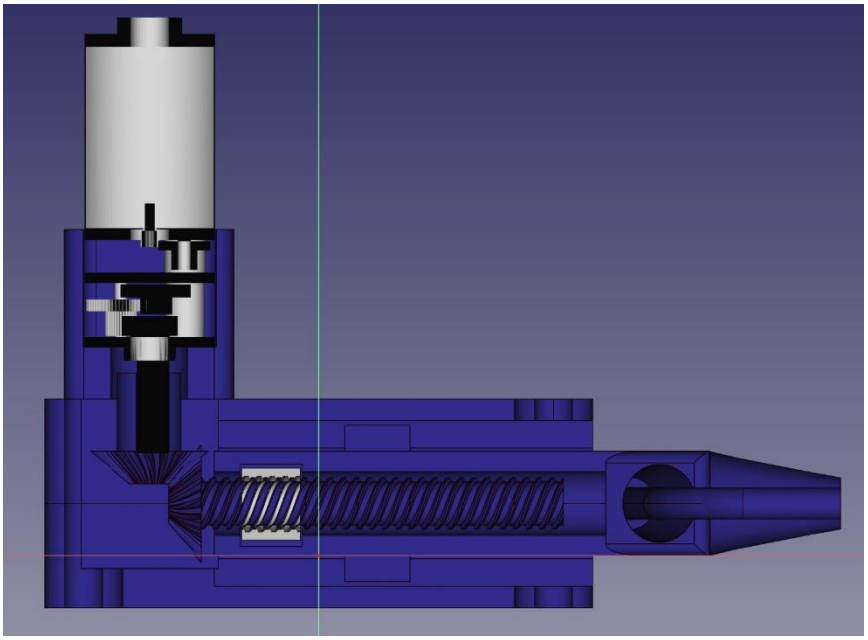


*Abbildung 7: Querschnitt des um 90° invertierten Designs*

längeren Haltbarkeit kollidiert wäre. Aus diesem Grund wurde sich dagegen entschieden, diese Idee weiterzuverfolgen. Als Lösung wurde die Länge der Spindel weiter angepasst, so dass diese nur noch genau auf das Ausfahren der entsprechenden Länge ausgelegt ist.

## 4.2. Platine

Die Konzeption einer Platine war einer der wichtigsten Punkte, welcher umzusetzen war. So würde eine Platine die Kosten des Barkeepers reduzieren, da dieser so in Zukunft schneller hergestellt werden kann und somit der Arbeitsaufwand, sowie die dadurch entstehenden Arbeitskosten gesenkt werden können. Außerdem erhöht dies die Zuverlässigkeit des Gerätes um ein Vielfaches, da sich hierbei keine Kabel lösen können, wie es bei einem Breadboard der Fall wäre und allgemein weniger Kabel vorhanden sind, wodurch auch die Wahrscheinlichkeit sinkt, dass ein Kabel Fehler aufweist. Infolge dessen steigt außerdem die Lebensdauer des Gerätes. Auch die Instandhaltung wird durch die Reduktion der Kabel einfacher, da ein auftretender Fehler so schneller und einfacher ausfindig zu machen ist. Außerdem wird der Barkeeper dadurch, dass die Leiterbahnen der Platine auf einer fast schon zweidimensionalen Ebene liegen und somit der Abstand zwischen der unteren Ebene mit der Platine und der oberen Ebene verringert werden kann, deutlich kompakter. Mit all diesen Zielen im Kopf habe ich mich damit beschäftigt die Platine zu designen. Dies war eines der schwierigsten Punkte, welche es am Barkeeper umzusetzen galt. Denn bis zu diesem Zeitpunkt hatte ich mich bereits stark mit dem Modellieren von Objekten, dem 3D-Druck, der Planung und der Zusammenschaltung von Elektronik auseinandergesetzt. Jedoch habe ich bis zu diesem Zeitpunkt noch nie eine eigene Platine designed. Dies bedeutete, dass ich mir selber von Grund auf beibringen musste eine Platine zu designen. Das Wichtigste war es erst einmal die richtige Software für das Designen einer Platine zu finden. Dies hat sich als schwieriger erwiesen als gedacht, da viele verschiedene Programme für das Designen von Platinen existieren. Meine einzige Anforderung an das Programm war, dass es kostenlos sein musste. Dabei bin ich dann auf Altium, Eagle und EasyEDA gestoßen. Schlussendlich habe ich mich für EasyEDA entschieden, da EasyEDA eine bereits von Nutzern erstellte Bibliothek aufweist, mit welcher man direkt Zugriff auf schon erstellte Platinen hatte, wie zum Beispiel den INA219 oder die A4988 Treiber für die Nema17 Motoren. Zusätzlich war dies auch die Software, welche von JLCPCB stammte. JLCPCB ist der Hersteller, bei welchem die entworfenen Platinen auch bestellt und hergestellt werden würden. Von hier an, habe ich mir viele Videos darüber angeschaut, wie eine Platine entworfen wird. Vor allen Dingen habe ich aber auch sehr viel selber ausprobiert. Das finale Design der Platine musste jedoch auf das finale Design der Z-Achse warten, denn mit der Z-Achse sind viele Komponenten hinzugekommen, wodurch sich viele Pinbelegungen geändert haben und damit auch das Design der Platine. So sind nach dem ersten Design der Platine, der Ina219 für das Messen der Stromstärke des Aktuators und die H-Bridge dazugekommen. Diese haben es erfordert die gesamte Platine von Grund auf neu aufzubauen. Eine Herausforderung die beim Design jeder Version der Platine bestand, lag darin, dass nur zwei Ebenen für die Leiterbahnen benutzt werden sollten, da das Hinzufügen von zwei weiteren Ebenen den Preis signifikant gesteigert hätte. Denn während das Board mit nur zwei Lagen ca. 10 € kostet, würde dasselbe Design mit vier Schichten schon ca. 35 € kosten. Dies würde einer Preissteigerung des 3,5-fachen entsprechen und sich dadurch preistechnisch gesehen sehr stark auf die Materialkosten des Barkeepers auswirken, welche es weiterhin galt so gering wie möglich zu halten. Nach Einfügen der richtigen Boards und dem Neudesignen der Platine, konnte Ende November die Bestellung des ersten Prototypens der Platine bei JLCPCB in Auftrag gegeben werden. Ein weiteres Problem hierbei war jedoch, dass JLCPCB beim Bestellen der Platine sehr viele Optionen lässt, welche ausgewählt werden können. Dies ist für viele ein sehr positiver Punkt, jedoch bedeutete dies für mich erstmal,

dass ich mich noch genauer mit den einzelnen Optionen auseinandersetzen musste und mir Stück für Stück anschauen musste, was welche Option bringt und ob diese für unsere Anwendungszwecke sinnvoll wäre oder nicht. Nach der Entscheidungsfindung wurde die Platine bestellt. Hiernach mussten die verschiedenen Komponenten auf der Platine zusammengelötet werden. Da es klar war, dass es schwer sein würde, mit Lötzinn alle feinen Pins ordentlich an die Platine gelötet zu bekommen, wurde schon im Voraus Lötpaste bestellt. Diese kann man auf alle Pins aufschmieren. Um diese schnell zu erhitzen wurde probiert, die Pins mit einer Heißluftpistole zu löten. Das Problem hierbei war jedoch, dass diese die Wärme zu großflächig verteilt, wodurch auch viele der anderen Komponenten warm wurden und fast sogar den Lötzinn auf den anzulötenden Platinen wie der H-Bridge geschmolzen hat. Auf Grund dessen wurde doch mit einem Lötkolben Pin für Pin gelötet. Dies hat einwandfrei funktioniert, so dass in den darauffolgenden Tagen die ersten Tests durchgeführt werden konnten. Es war schon von Anfang an klar, dass der erste Prototyp mit einer hohen Wahrscheinlichkeit Fehler aufweisen würde und dies war auch der Fall. Einer der offensichtlicheren Fehler bestand darin, dass für die Endstops zwar jeweils ein Pin für den Anschluss an den ESP erstellt wurde, jedoch vergessen wurde, dass jeder Endstop natürlich auch einen eigenen Ground Pin benötigt. Um hiermit trotzdem testen zu können, wurden diese einfach an andere Ground Pins gelötet. Ein weiterer Fehler war, dass die Pins auf der Platine für die H-Bridge spiegelverkehrt waren. Dies kam daher, dass ich vorher, als ich das Programm noch nicht so gut kannte, nicht verstanden hatte, dass die H-Bridge auf der Platine auf der Unterseite angebracht war. Nachdem ich das im Design jedoch korrigiert und auf die obere Seite gepackt hatte, hatte ich jedoch übersehen, dass dabei auch die Pins auf der Platine gedreht werden mussten. Dies habe ich für die ersten Tests dadurch behoben, indem ich die H-Bridge auf der Unterseite der Platine angebracht habe. Der letzte Fehler, welcher sich schon beim Löten ergeben hatte, bestand darin, dass das Modell welches aus der benutzerdefinierten Bibliothek für das Step-Down Modul heruntergeladen wurde, nicht vollständig akkurat war und die Output Pins voneinander 1mm näher zusammen waren, als diese eigentlich sein sollten. Durch das Auftragen von etwas mehr Lötpaste konnte aber auch dieses Problem zunächst gelöst werden. Nachdem diese drei Probleme zunächst umgangen waren, konnte mit den Tests gestartet werden. Der Aktuator und alle damit verbundenen Funktionen, sowie die Endstops haben einwandfrei funktioniert. Die Schrittmotoren haben sich jedoch nicht bewegt und konnten auch nicht einmal auf Spannung gebracht werden. Nach einer erneuten Prüfung der Pinbelegung und der Verknüpfung auf der Platine, fiel auf, dass die Pins um die Motoren in den Sleep Mode zu versetzen auf Ground gezogen wurden. Dies ist für die meisten Pins auch richtig, wenn man diese ausgeschaltet haben möchte. Bei diesen Pins handelte es sich aber um so genannte „active low pins“. Dies sind Pins, welche die damit verbundenen Funktionen dann deaktivieren, wenn an diesen eine Spannung anliegt. Um sicher zu gehen, dass dies auch wirklich der Fall war, wurden mit einem Cuttermesser die Leiterbahnen, die zu den entsprechenden Sleep Mode pins geführt haben, durchtrennt und die Erdungsleiterbahn (GND), welche durch die Sleep Mode Pins noch zu anderen Grounds führten, überbrückt. Dadurch haben die Motoren nun auch funktioniert, womit die gesamte Elektronik einwandfrei lief.

Aufgrund dessen konnte jetzt angefangen werden, das Platinendesign zu überarbeiten und die Fehler zu beheben. Nachdem auch dies erledigt wurde, mussten die einzelnen Komponenten nun erst einmal von der nun alten Platine wieder abgelötet werden. Da ich schon vorher realisiert hatte, dass es schwer werden würde, dies ohne eine ordentliche Entlötpumpe, welche selbst warm wird, zu bewerkstelligen, habe ich mir eine solche entsprechend gekauft. Mit dieser Entlötpumpe konnten fast problemlos der Lötzinn und die damit verbundenen Komponenten von der Platine entfernt werden. Nachdem die neue Platine angekommen war, wurden auch hier wieder alle Komponenten angelötet und die Platine konnte getestet werden. Hierbei sind keine weiteren Probleme aufgetreten. Nachdem nun nachgewiesen war, dass alle Komponenten mit der Platine funktionierten habe ich die Kabel welche von der Platine zu den Komponenten führen ein letztes Mal abgelötet, Löcher in die obere Platine gebohrt, durch welche die Kabel direkt zur unteren Ebene geführt werden können, die Kabel gekürzt und wieder an die Platine gelötet. Der Sinn dabei war, die Kabel vor allem auf der oberen, als aber auch auf der unteren Ebene, übersichtlicher und somit auch wartbarer zu gestalten.

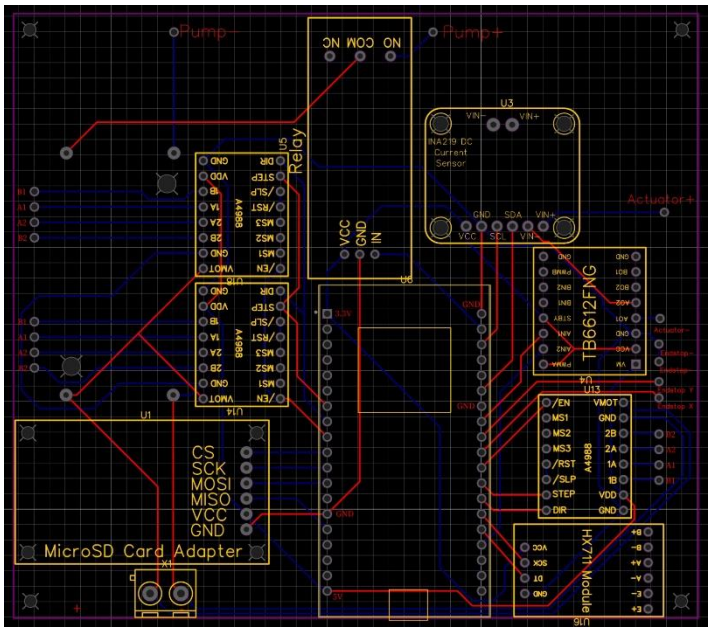



*Abbildung 8: Finales Platinendesign in EasyEDA*

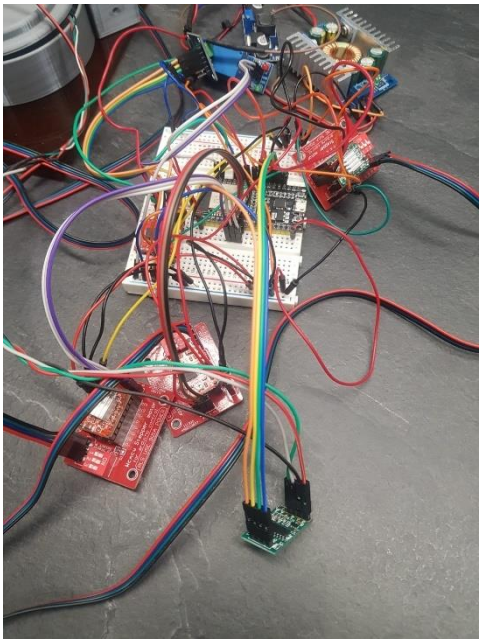


*Abbildung 10: Unterschied vorher nachher, Breadboard*

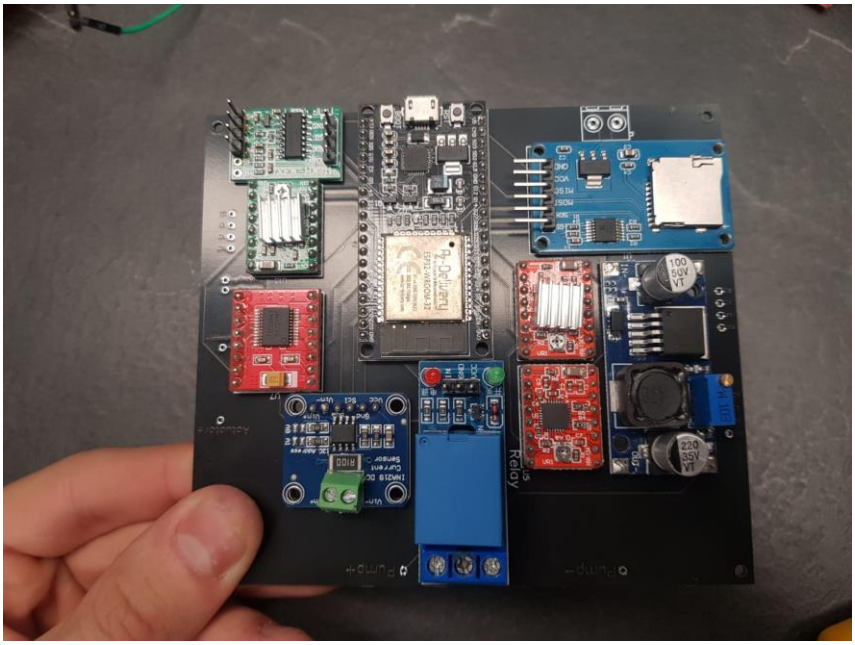


*Abbildung 9: Unterschied vorher nachher, Platine*

## 4.3. Aufsatz für Getränkekästen

Der Aufsatz für Getränkekästen spricht weitere Ziele des Automatischen Barkeepers 2.0 an. Und zwar das Ziel, den Barkeeper noch kompakter und stabiler zu gestalten, die Lebensdauer zu erhöhen, sowie die Kosten noch weiter zu senken. Das Vorgängermodell war aufklappbar, damit an den Seiten die verschiedenen Flaschen mit den Flüssigkeiten angeschlossen werden konnten. Zudem stand dieser eigenständig. Dies bedeutet aber, dass mehrere lange Metallstangen benötigt werden, welche von oben nach unten reichen müssen, um die obere und untere Ebene zu verbinden und auch als Ständer dienen. Zusätzlich wurde eine weitere Acrylplatte auf dem Boden benötigt, damit die Metallstangen nicht verrutschen können. Eine Halterung des Barkeepers für Kästen löst diese Probleme, denn hierdurch können die Metallstangen und die untere Acrylplatte vermieden werden und somit die Materialkosten weiter gesenkt werden. Außerdem sind Kästen stabiler als eine selbstgebaute Halterung, da diese meist aus massivem, spritzgegossenem Plastik bestehen. Zudem haben die meisten Menschen solche Kästen Zuhause und wenn nicht, sind diese in jedem Getränkeladen ohne Probleme für einen günstigen Preis kaufbar. Die erste Überlegung für das Erstellen der Halterung war, die obere Platte des Barkeepers einfach zwischen zwei präzise gedruckten Schichten zu klemmen und die untere Platte wieder mit Metallstangen an der oberen Platte zu befestigen. Dabei müssten die Metallstangen aber nicht mehr zum Boden reichen, sondern nur noch bis zur unteren Platte. Nachdem aber auf die Platine umgestiegen wurde, konnten die zweite Platte so nah an der oberen Platte angebracht werden, dass es logischer und ökonomisch günstiger erschien auch die zweite Platte einfach an dem Adapter für die Getränkekästen zu befestigen, so dass keine Metallstangen mehr benötigt wurden und die Kosten damit ohne Einbußen weiter gesenkt werden konnten. Der letzte Verbesserungsschritt ergab sich aus dem Problem, dass schon bei einem leichten Stoß gegen die Getränkekästen, der Barkeeper aus seiner Halterung rutschen könnte. Um dieses Problem zu beheben wurden in die Platten Löcher gebohrt und die Halter an den Platten mit Schrauben befestigt. Da die Gefahr besteht, dass sich die Schrauben mit der Zeit lösen, wenn diese ausschließlich im Plastik befestigt sind, wurden Gewindeeinsätze verwendet. Diese aus Messing bestehenden Gewindeeinsätze werden nach dem 3D-Druck in das Plastik mit Hilfe eines Lötkolbens eingeschmolzen und halten dadurch, dass diese aus Metall bestehen, auch wiederholtem lösen und befestigen problemlos stand. Das einzige Problem war nun noch, dass der Akku auf Grund der geringen Höhe der unteren zur oberen Schicht nicht mehr auf die untere Ebene passte. Dies wurde gelöst, indem die Halterung für den Akku auf der Unterseite der unteren Ebene befestigt wurde und die Kabel von dort aus durch ein gebohrtes Loch direkt auf die Oberseite der unteren Ebene geführt wurden. Da der Akku relativ schwer ist, wurden auch hier zu Befestigung der Schrauben an die untere Platte Gewindeeinsätze benutzt und die Akkuhalterung direkt neben der Kastenhalterung platziert. Dadurch wird die entstehende Hebelwirkung auf den Barkeeper bestmöglich reduziert.

## 4.4. Geschwindigkeit

Die Geschwindigkeit des Barkeepers ist einer der großen Kritikpunkte gewesen. Glücklicherweise ließ sich dieser Punkt aber relativ einfach lösen, da die Motoren mit Half-Steps liefen. Dadurch sind diese zwar präziser aber auch langsamer. Eine solch hohe Präzision war jedoch nicht notwendig, da durch die konische Form des Aktuators, die Dichtigkeit trotzdem gegeben ist. Folglich konnte auf Full-Steps umgestellt werden und somit die Geschwindigkeit der Motoren, welche für die Y- und Z-Achse zuständig sind, verdoppelt werden. Auch an der X-Achse wurden Maßnahmen getroffen um die Geschwindigkeit zu erhöhen. So wurde die Spannung des Motors erhöht, wodurch sich auch die Geschwindigkeit, mit der der Motor raus und rein fährt, erhöhte. Durch die Möglichkeit die Spindel 3D-Drucken zu können, war es möglich die Steigung der Helix zu erhöhen, was im

Umkehrschluss dazu führte, dass pro Umdrehung der Spindel der Aktuator weiter herausgedrückt wird. Dies führt zu einem schnelleren Aus- und Einfahren. Hierfür wurden mehrere Versionen mit verschiedenen Steigungen gleichzeitig gedruckt und getestet, um den Optimalwert für die Steigung zu finden. Denn eine höhere Steigung führt auch dazu, dass mehr Reibung zwischen Mutter und Spindel entsteht und der Motor somit mehr Kraft aufwenden muss, um diese gedreht zu bekommen.

## 4.5. Schlauchführung

### 4.5.1. Flüssigkeitsschlauchführung

Die Flüssigkeitsschlauchführung war nun noch eines der größten, verbliebenen Probleme. Denn besonders jetzt, da die Anzahl der Anschlüsse gesteigert wurde, würde es, ohne ordentliche Schlauchführung zu einem Wirrwarr aus Schläuchen kommen. Ein weiteres Problem ergab sich jetzt aus der Menge der Schläuche. Der bisherige Barkeeper hatte 8 Schläuche, welche direkt über das Glas geführt wurden. Bei 16 oder sogar 24 Schläuchen ist dies aber nicht mehr möglich. Zur Lösung dieses Problems sind mir gleich mehrere Ideen eingefallen. Bei allen Ideen sah ich es als die sinnvollste Lösung an, die Schläuche durch die Platten durch zu führen um damit zumindest schon einmal eine Art Führung zu haben. Die Frage war damit nur noch wo auf der Platte die verschiedenen Schläuche durch die Platten gehen sollten. Dafür war es erst einmal wichtig zu klären, wo sich überhaupt hinterher die Ausgabe befinden sollte. Entschieden wurde, die Ausgabe mittig zwischen den beiden Kästen an der Vorderseite, also gegenüber der Platine, zu platzieren. Es bestand aber auch die Möglichkeit, dass die Ausgabe komplett mittig angebracht werden sollte. Dies wurde immer designabhängig entschieden. Das erste Design, welches erstellt wurde, bestand daraus die Schläuche an den jeweiligen Seiten der Kästen in die Platine einzuführen und dann darunter mit einem 3D-gedruckten Rohrsystem in die Mitte zu transportieren. Das zweite System bestand daraus, dass Schläuche vorne in die Platte, dafür aber auch wieder links und rechts neben der Z-Achse in die Platten einzuführen. Hier würde darunter dann kein Rohrsystem verbaut werden müssen, sondern nur noch ein 3D-gedruckter viereckiger Trichter. Die dritte Möglichkeit bestand darin das gesamte Design in der Y-Achse so zu kürzen, dass gerade so noch Platz ist, die Schläuche mittig zwischen den beiden Kästen vorne gemeinsam durchzuführen. Hierbei könnte dann auch ein Metalltrichter

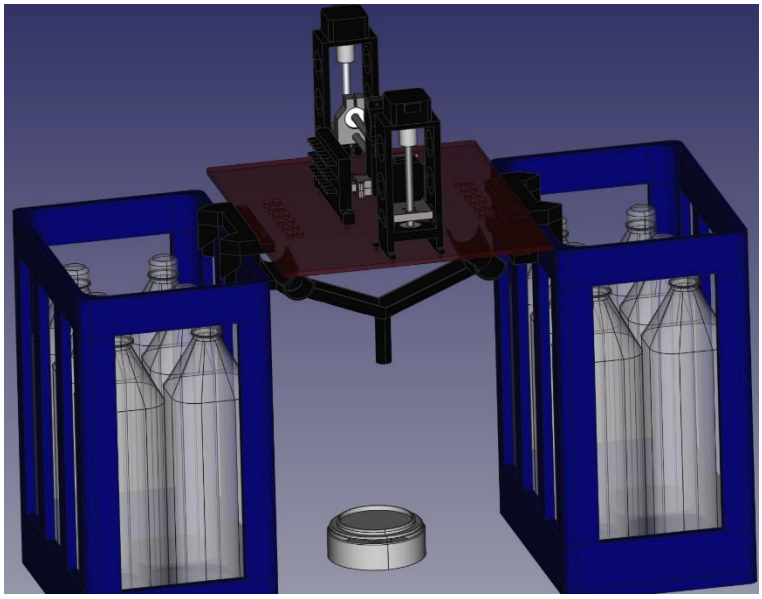


*Abbildung 12: Design 1*

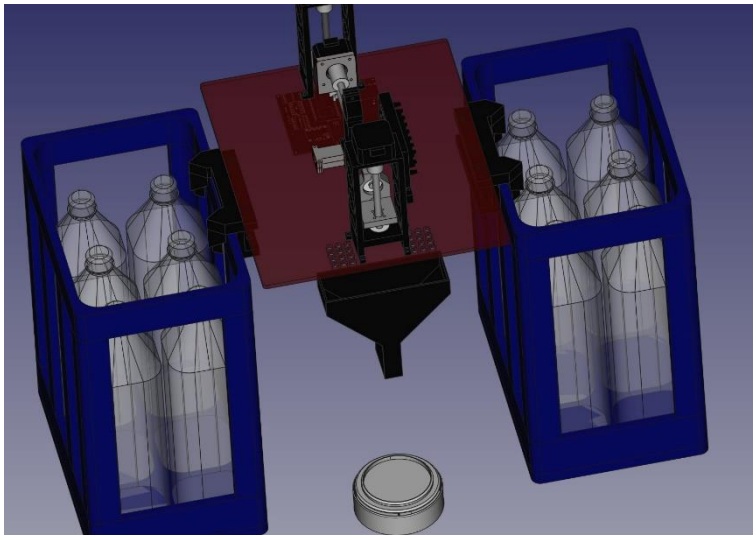


*Abbildung 11: Design 2*

verwendet werden. Nachdem ich Modelle für diese drei Möglichkeiten erstellt hatte, habe ich mir die Vor- und Nachteile der einzelnen Ideen aufgeschrieben und gegenübergestellt. Das erste Design hatte die meisten Nachteile. Denn während bei diesem Design die Schläuche den kürzesten Weg haben, ist das Rohrsystem darunter schwer waschbar und würde wahrscheinlich auch schwer abzunehmen sein. Außerdem würde das Rohrsystem eine lange Latenzzeit aufweisen, da zwischen dem Start der Flüssigkeitsförderung und der ersten Flüssigkeit im Glas eine gewisse Zeit vergeht, wodurch Kalkulationen, wann genau die Pumpe aufhören muss zu pumpen, schwierig geworden wären. Zudem entstehen durch den 3D-Druck Rillen, in welchen sich Flüssigkeit ablagern könnte. Auch auf die Lebensmittelechtheit muss geachtet werden. Die zweite Konstruktion hat dadurch, dass diese keine so langen Rohre verwendet, da alles relativ nah an einander ist, den Vorteil, dass es keine lange Latenzzeit gibt. Der Trichter wäre außerdem relativ klein, was einen Vorteil bei der Reinigung mit sich bringt. Ungeachtet dessen verbleibt jedoch das Problem mit der Ablagerung der Flüssigkeiten in den Rillen und der Lebensmittelechtheit. So blieb für mich nur noch die dritte Option übrig, dessen einziger Nachteil es noch war, dass alles noch kleiner gemacht werden müsste, jedoch gab es dafür kein Problem mit der Lebensmittelechtheit und Ablagerungen zwischen den Rillen. Glücklicherweise kam hier aber die Förderung ins Spiel, so dass der Nachteil komplett ausgeräumt wurde. Mit Hilfe dieser konnte größeres Acrylglas, als das bereits in der Schule vorhandene, gekauft werden. Dadurch war es nicht mehr nötig alles noch kleiner gestalten zu müssen. Diese sollen mit dem Laser von AtomStack präzise zugeschnitten werden.

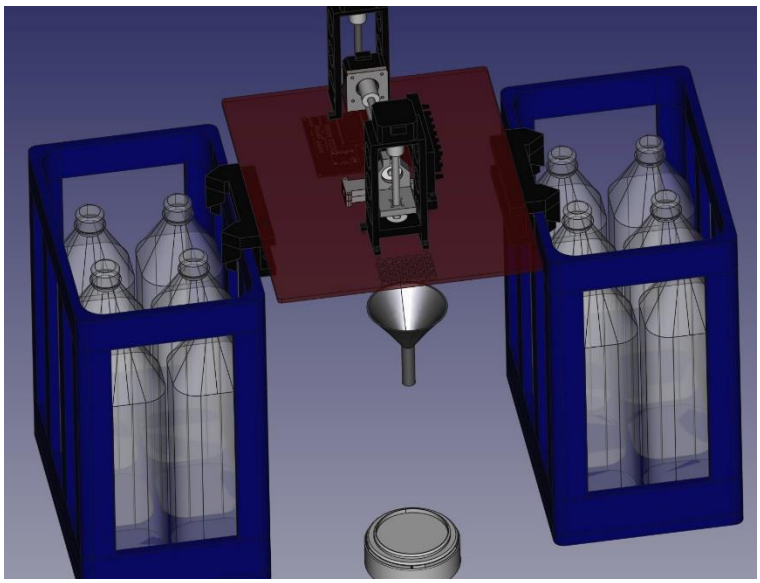


*Abbildung 13: Design 3*

### 4.5.2. Luftschlauchführung

Die Luftschlauchführung ist an sich zwar kein so großes Problem, wie die Führung der Flüssigkeitsschläuche, jedoch sollte auch die Luftschlauchführung nicht in einem Gewirr an Schläuchen enden. Während es wichtig war, dass bei den Flüssigkeitsschläuchen die Schläuche gerade durch beide Platten nach unten geführt werden würden, damit diese auch einfach entfernbar sind, um diese, wenn nötig, durchspülen zu können, ist dies bei den Luftschläuchen nicht nötig, da diese nur Luft und keine Flüssigkeiten transportieren. Aus diesem Grund wurde sich dafür entschieden die Luftschläuche, von dem Kasten der dem Verteiler gegenüber liegt, auf dessen Seite in die untere Ebene zum Verteiler hin zu führen und diese dann erst dort wieder auf die obere Ebene zu führen.

## 4.6. Gehäuse

Ein Gehäuse um den Barkeeper drum herum hat als Ziele die Lebensdauer zu erhöhen (wie z.B. durch Schutz vor Flüssigkeiten), jegliche Verletzungsrisiken zu vermeiden (wie z.B. Hautquetschungen) und zusätzlich das Design verkaufsfördernd zu gestalten. Da die Technik des Barkeepers gesehen werden soll, wird dieses Gehäuse mit dem aus der Förderung angefragten Acrylglas gebaut werden. Um hier die richtigen Maße und präzise Schnitte für das Gehäuse verwenden zu können, soll auch hier der Lasercutter von AtomStack eingesetzt werden.