



UNIVERSITÄT ZU LÜBECK

BACHELORPROJEKT

Nova Spot Micro

Autoren: Lukas Bruns, Yasin Kirschstein
Betreuer: Prof. Dr. rer. nat. habil. Floris Ernst
Institut für Robotik und Kognitive Systeme

Einführung

Vorbild Boston Dynamics Spot

Der in diesem Projekt gebaute Roboter wurde inspiriert von dem Boston Dynamics Roboter „Spot“, welcher vorrangig zur Erkundung, Überwachung und Sicherheit eingesetzt werden kann. So verfügt dieser beispielsweise über Remote-Site-Zugriff und ist in der Lage sowohl visuelle, akustische als auch thermale Aufnahmen zu machen und ist gleichzeitig in der Lage an schwer zu erreichende Positionen zu gelangen, welche potenziell gefährlich für Menschen wären. Zudem kann der Spot mit einem 6-DOF Greifarm ausgestattet werden, welcher es dem Roboter erlaubt Dinge in seiner Umgebung zu greifen und tragen, um so noch mehr Einfluss auf seine Umgebung zu nehmen.

Grundlage

Das initiale 3D-Design für unser Projekt wurde von Deok-yeon Kim entwickelt und auf Thingiverse.com zur Verfügung gestellt. Seitdem wurden viele Änderungen am 3D-Design selbst vorgenommen und zusätzlich Dokumentation für Elektronik und Arduino Code von Chris Locke zur Verfügung gestellt. Dieser hat seine Arbeit auf einer eigens für dieses Projekt erstellten [Homepage](#) öffentlich zugänglich gemacht und es wurden zudem ein GitHub, Discord-Server, sowie viele YouTube-Videos zu verschiedensten Themen aufgenommen. Obwohl die Dokumentation ausführlich ist, kann es manchmal schwierig sein, nicht den Überblick zu verlieren, da es viele verschiedene Versionen des Projekts gab. Zudem sind die YouTube-Videos meistens zu veralteten Versionen des Roboters. Zwar wurde eine aktuelle Videoserie aufgenommen, diese ist derzeit aber noch nicht beendet worden. Da das letzte Video bereits vor elf Monaten erschien, ist es durchaus möglich, dass diese auch nicht mehr zu Ende geführt wird.

Einordnung des Projekts

Das Projekt lässt sich in den Teilbereich der mobilen Robotik einordnen und benötigt eine Einarbeitung in verschiedenste Bereiche der Robotik. So werden auf Hardware-Seite sowohl Kenntnisse über verschiedenste Elektronik Bauteile und ihre Interaktion miteinander als auch, Schaltpläne, 3D-Druck oder Löten gefordert. Auf Software-Seite muss sich vor allem in Arduino und dessen Funktionsweise eingearbeitet werden. Dies kam in diesem Projekt besonders zum Vorschein, da wir das Original von zwei Mikrocontrollern, namentlich Teensy 4.0 und Arduino Nano, auf einen Arduino Nano RP2040 vereinfacht haben. Dazu aber in einem späteren Abschnitt mehr.

Organisation und Ablauf

Da dieses Projekt bereits im letzten Semester von Studierenden der Universität durchgeführt wurde, waren sowohl große Teile der Elektronik als auch ein Teil der zu druckenden 3D-Teile vorhanden. Daher war der erste Schritt von unserem Projekt eine Bestandsaufnahme über bereits vorhandene Elektronik und 3D-Teile durchzuführen und fehlende Elektronik nachzubestellen bzw. fehlende 3D-Teile neu zu drucken. Da die 3D-Teile größtenteils Mängel aufgewiesen haben, wurde ein Großteil dieser neu gedruckt. Zudem waren wir sehr darauf bedacht alle 3D-Teile in einem möglichst makellosen Zustand zu haben, um dadurch beim späteren Zusammenbau des Roboters keine Probleme zu haben.

Motivation und Ziel dieses Projekts

Ziel dieses Projekts war der Bau eines weiteren „Nova Spot Micro“ Roboters, um damit weitere Forschung zu betreiben. So können mit diesem Roboter beispielsweise Bachelorarbeiten oder weitere Projekte, welche beispielsweise einen Roboterarm am Roboter anbringen, durchgeführt werden. Unsere Motivation dieses Projekt zu bearbeiten, stammte größtenteils daher, tiefere Einblicke in vor allem den Hardware-Teil eines Roboteraufbaus zu erlangen. Dieser wurde unserer Meinung nach bisher im Studium größtenteils nicht behandelt, welches bei einem Studium an einer Universität auch erwartbar ist. Zudem deckt das Projekt ein breites Spektrum von vielen verschiedenen Bereichen der Robotik ab und half vor allem bei der Themenfindung unserer Bachelorarbeit.

Aufbau dieses Projektberichts

Da, wie bereits vorher schon einmal angesprochen wurde, dieses Projekt bereits von Studenten der Universität durchgeführt wurde, werden in diesem Bericht nur die Unterschiede zum vorherigen Projekt bzw. zum Original Roboter dargestellt. Für eine umfangreiche Beschreibung der einzelnen Teile des Roboteraufbaus verweisen wir [auf folgenden Bericht](#).

3D Druck und Vorbereitung der Einzelteile

Der 3D-Druck der fehlenden Teile erfolgte in der Werkstatt des Instituts und verlief ohne Probleme. Das eigentliche Problem dieser Phase war das Vorbereiten der Teile für den späteren Zusammenbau. So mussten bei so gut wie allen Teilen Stützmaterialien entfernt werden, da dies es sonst unmöglich gemacht hätte, die einzelnen Teile des Roboters zusammenzufügen. Das größte Problem in dieser Phase war der Einbau von Muttern, Servohörnern und Lagern, da diese praktisch nie in die dafür vorgesehenen Aussparungen, an den einzelnen Teilen gepasst haben. Dies wurde entweder mittels Erhitzens der einzelnen Bauteile gelöst, welche dann in die entsprechenden Aussparungen gedrückt wurden oder indem mit einem Dremel die Löcher ausgebessert wurden bzw. die Ränder von den einzelnen Teilen abgeschliffen wurden. Die gesamte Druckzeit lässt sich auf über 100 Stunden abschätzen, da aber bereits einige Teile vorhanden waren, mussten wir nur ungefähr 60 Stunden drucken.



Abbildung 1: Bearbeitete 3D-gedruckte Teile der Beine



Abbildung 2: Bearbeitete 3D-gedruckte Teile des Torsos

Zusammenbau des Roboters

Der Aufbau des Roboters lässt sich in Frame und Cover Teile einteilen. In der Originalversion wurden die Frameteile in Schwarz und die Coverteile in Gelb gedruckt, welches ein Einfaches auseinanderhalten ermöglicht hat. Da wir die Drucker im Institut genutzt haben und nicht immer die genutzten PLA-Rollen tauschen wollten, haben die Teile verschiedene Farben und stellen keinen klaren Anhaltspunkt zur auseinander Haltung dieser dar. Zudem wurden für die Tibia Gelenke, anders als im vorherigen Projekt, 35Kg starke Servomotoren verbaut, welche bessere Bewegungen des Roboters zur Folge haben.

Elektronik Modifikation

Entstehung der Idee

Die Idee nur einen Mikrocontroller zu nutzen, entstand bei der Durchführung des vorherigen Projekts. Dort wurde gemerkt, dass auf Bauteile wie den MP3-Player, Audio-Speaker und verschiedenste LED-Anzeigen verzichtet werden konnte, da der Roboter primär für Forschungszwecke genutzt werden soll und diese dafür unerheblich sind. Dieses Wegstreichen von Bauteilen hatte eine große Verkürzung des Programmcodes zur Folge. Dadurch war es folglich möglich den kompletten Programmcode nur auf den Arduino zu laden, welches vorher nicht möglich war. Zudem hat der Arduino Nano allein genug Pinouts, um alle wesentlichen Sensoren und Aktoren anzuschließen. Der Vorteil dieser Neugestaltung ist, dass die komplizierte Aufteilung der Funktionalitäten des Roboters gestrichen werden kann. Dies hat zum einen eine Vereinfachung des Codes zur Folge, aber ist auch für später an diesem Roboter durchgeführte Projekte benutzerfreundlicher. So lassen sich beispielsweise spätere Erweiterungen einfacher integrieren.

Für Änderung nötige Schritte

Um diese Modifikation zu ermöglichen, wurde im ersten Schritt untersucht, welche der im Original verbauten Elektronikkomponenten in unserem Roboter verbaut werden müssen. Obwohl von der vorherigen Gruppe noch ein eigens für den Roboter designtes PCB-Board vorhanden war, konnte dieses nicht genutzt werden, da viele Connections zum Teensy 4.0 geführt haben, welchen wir in diesem Projekt nicht nutzen werden. Daher wurde als erster Schritt mittels der Software „Fritzing“ ein eigener Netzplan erstellt, an dem wir uns dann im späteren Verlauf orientieren konnten.

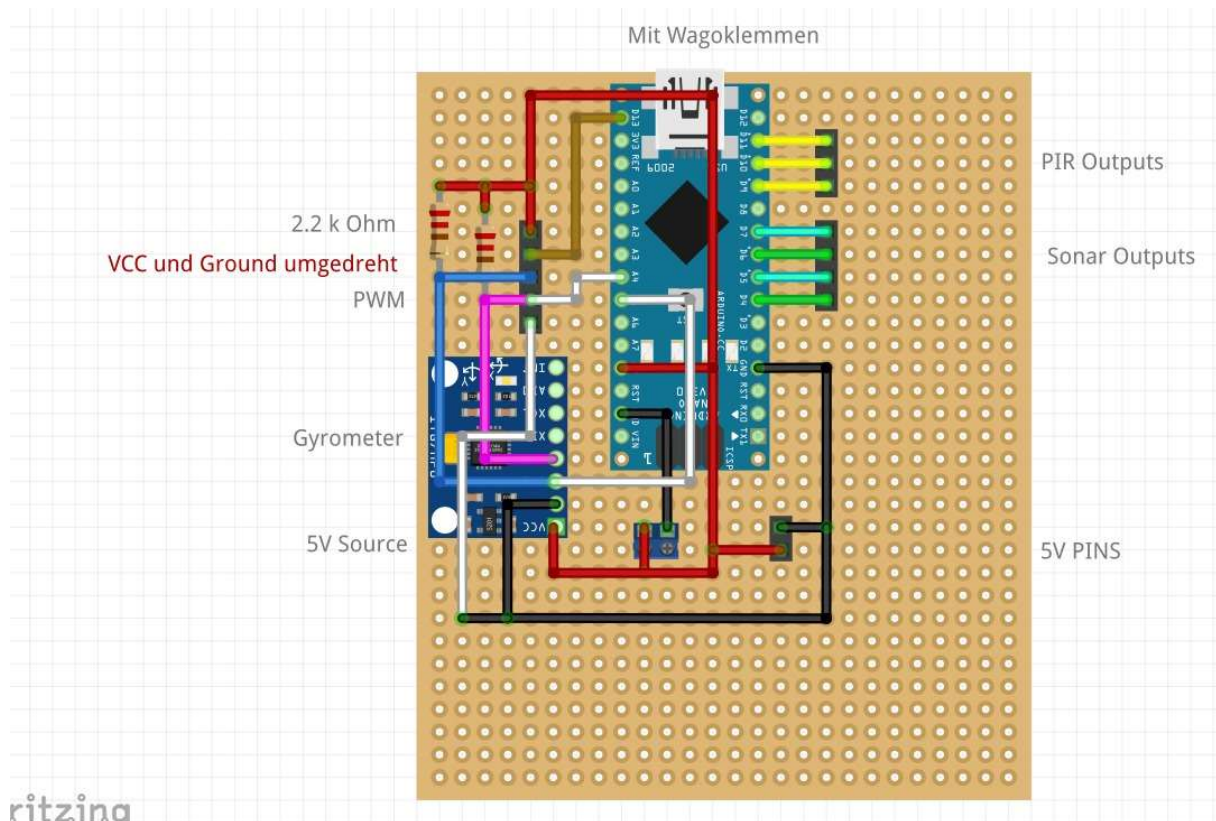


Abbildung 3: Netzplan des PCB-Boards, erstellt in der Software Fritzing

Aufbauend auf dieser Vorlage wurde dann das PCB-Board in echt gelötet. Der Ablauf dessen wird später genauer behandelt.

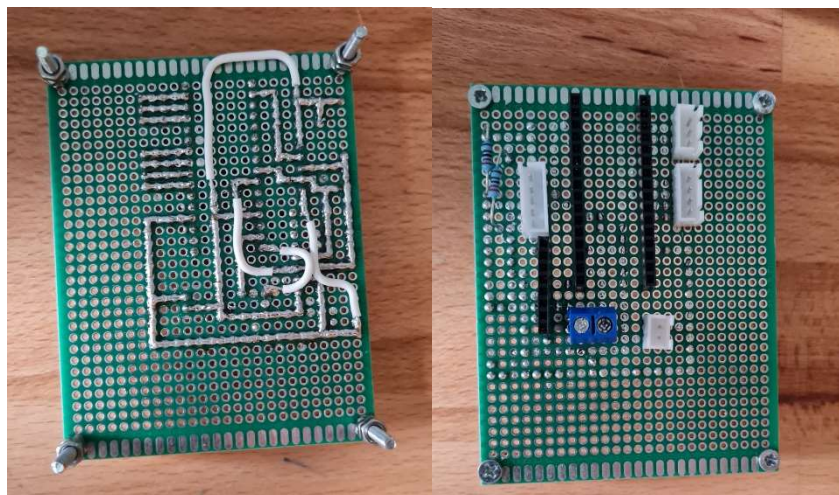


Abbildung 4: Vorder- und Hinterseite des gelöteten PCB-Boards

Überblick über die verbauten Elektronikkomponenten PCB

Auf dem PCB wurden verschiedenste Elektronikkomponenten eingebaut und angeschlossen. Zum einen der Arduino selbst, welcher mittels einfacher Steckplätze angebracht und somit auch leicht ausgetauscht werden kann. Aber auch der PWM Controller PCA9865, welcher für eine einheitliche Ansteuerung der verbauten Servomotoren genutzt wird. Dieser ermöglicht einen Anschluss von bis zu 16 Motoren, wo von in diesem Projekt nur 12 genutzt wurden. Zudem ist die MPU6050 verbaut und wird als Beschleunigungssensor und Gyroskop genutzt. Für Objekterkennung von beispielsweise Wänden und Gegenständen sind zwei Ultraschallsensoren am Kopf des Roboters angebracht und am

Board angeschlossen. Abschließend sind noch drei PIR-Sensoren verbaut, deren Sichtweite auf 120 Grad mittels 3D gedruckten Kappen reduziert wird, damit jeder Sensor nur Bewegungen in einer Richtung detektieren kann. Der PWM-Controller und die MPU nutzen beide I2C als seriellen Datenbus, um mittels ihrer Adressen mit dem Arduino zu kommunizieren. Da mehrere Elemente an dem I2C Bus angeschlossen wurden haben wir jeweils 2.2k Ohm Pull-up Widerstände an die SDA und SCL-Verbindungen und die 5V Leiterbahn angeschlossen. Dies verbessert die Signalqualität auf dem I2C-Bus und ist notwendig, um sicherzustellen, dass das Signal auf den Busleitungen SDA und SCL auf einem High-Pegel gehalten wird.

Restliche Elektronik

Die Stromkonvertierung ist gleich der des vorherigen Projekts. Es wurde eine 2200 mAh Batterie genutzt, welche 11V bereitstellt und mittels des ersten variablen Buck-Converters auf 6.8V reduziert wird. Dieser Wert entspricht dem maximalen Volt-Wert für die einzelnen Servomotoren. Mit einem zweiten Buck-Converter wird der 6.8V starke Strom dann auf 5V reduziert, um dann direkt an den 5V Eingang des Arduino Nano angeschlossen zu werden und diesen mit Strom zu versorgen. Zudem besitzt der Roboter einen An/Aus Schalter für die komplette Stromzufuhr und einen Wippschalter nur für den PWM-Controller, um gegebenenfalls gefährliche Bewegungen des Roboters zu stoppen, ohne dabei die komplette Stromversorgung zu unterbrechen.



Abbildung 5: Elektronische Einzelteile für die Aktorik der Robotik



Abbildung 6: Verkabelungen und die Batterie für das Anschließen der Elektronik

Anbringen der Elektronik an PCB-Board

Da nur ein einzelnes Board kreiert werden musste und gleichzeitig mögliche Änderungen am Layout des Boards berücksichtigt wurden, haben wir die JST und Pin-Verbindungen auf eine doppelseitige Steckplatine gelötet. Dabei wurden die einzelnen Leitungen mittels Lötzinns und Kabel nach dem vorgegebenen Schaltplan gelötet. Beim Layout des Boards wurde darauf geachtet, dieses so effizient und kompakt wie möglich zu gestalten. Dies spart zum einen Löt Aufwand und es wird Platz gespart, welches wichtig ist, da für die Elektronik des Roboters nur eine relativ kleine Aussparung in dessen Mitte vorgesehen ist. Zudem musste darauf geachtet werden, dass der USB-Anschluss des Arduino, vor dem dafür vorgesehen Loch im Torso des Roboters liegt, da es sonst nach Anbringen der Cover-Parts nicht mehr möglich gewesen wäre, diesen mittels Kabel anzuschließen. Die MPU wurde mittig platziert, um ein möglichst akkurates Bild über die Beschleunigung und Lage des Roboters zu erhalten. Die Anschlüsse für die Ultraschall - und PIR-Sensoren wurden auf die rechte Seite des Roboters gelegt, da dies beim späteren Anbringen der Kabel, die am einfachsten zu erreichende Position darstellt. Für beide Sensoren wurden die Verbindungen für die Daten mittels JST-Connections umgesetzt und die Verbindungen für VCC und Ground jeweils mittels Wago Klemmen entsprechend angeschlossen. Die Ultraschallsensoren haben gegenüber den PIR-Sensoren einen zusätzlichen Anschluss für Trigger, welcher ebenfalls mittels JST-Connections umgesetzt wurde. Anstatt des Terminals wurden Wago Klemmen genutzt, um zum einen den Platz im Roboter effektiver auszunutzen und da wir so nicht extra noch Screw Terminals kaufen mussten.

Um die einzelnen Teile im Inneren des Roboters fest unterzubringen, wurden Löcher in die Mitte des Roboters gebohrt, an denen wir dann die einzelnen Elektronikteile mit Schrauben und Muttern angebracht haben. Damit die Elektronik und speziell die einzelnen Lötverbindungen nicht direkt auf dem Plastik anliegen, haben wir Muttern von der anderen Seite an die Schrauben gedreht. Die Kabel wurden dann abschließend so gut es ging in die noch freien Hohlräume zwischen Mitte und Vorder-Hinterteil gedrückt, um ein Einfaches anbringen der Cover-Parts oben und unten zu ermöglichen.

Besonderheiten des genutzten Arduinos

Als alleiniges Board wurde eine besondere Version des Arduino Nano auf Basis des 32-Bit Mikrocontroller RP2040 verwendet. Dieser basiert auf dem Dual-Core ARM Cortex-M0+ Design und wurde von der Raspberry Pi Foundation selbst entwickelt und 2021 erstmals vorgestellt. Das Besondere an diesem Arduino Board ist, dass es sowohl Bluetooth als auch Wi-Fi unterstützt und somit ohne Kabel kontrolliert werden kann. Des Weiteren besitzt das Board eigene Sensoren in der Form eines omnidirektionalen Mikrofons, Temperatursensor und eine sechs Achsen IMU. Zudem lässt sich das Board mittels C++ und Python programmieren und ermöglicht sogar Machine Learning mittels TinyML oder TensorFlow Lite. In diesem Projekt wurde dieses Board hauptsächlich gewählt, da es ermöglicht den Roboter mittels ROS anzusteuern. Vorteilhaft ist, dass diese Version die gleichen Pinouts wie ein normaler Arduino Nano hat und dementsprechend keine Änderungen in dieser Hinsicht getroffen werden mussten.

Programm Modifikationen

Im Original sowie im Projekt der vorherigen Gruppe wurden, wie schon mehrfach erwähnt, zwei Mikrocontroller für die Ansteuerung des Roboters genutzt. Dabei wurde der Arduino Nano als Slave und der Teensy als Master eingesetzt. Da wir in unserem Projekt nur den Arduino Nano nutzen können wir auf diese, aus unserer Sicht unnötig komplizierte, Aufteilung verzichten und den gesamten Programmcode auf dem Arduino Nano laufen lassen. Um dies tatsächlich umzusetzen, mussten einige Änderungen an dem vorhandenen Code vorgenommen werden. Als Ausgangspunkt wurde dabei der Code der vorherigen Gruppe genutzt, da dieser schon für uns relevante Kürzungen des Originalcodes hat. Da der Programmcode des Arduino vorher im Großen und Ganzen nur aus dem Auslesen der Ultraschallsensoren bestand, wurden die für uns relevanten Variablen und Methoden kopiert und der

Rest verworfen. Diese Funktionalität wurde dann in den entsprechenden Stellen des vorherigen Teensy Codes eingefügt. Da für alle Funktionen des Roboters am Anfang des Codes Variablen angelegt wurden, welche bestimmte Verhalten an bzw. aus schalten, konnten wir die Variable zur Aktivierung des Slaves (also des Arduinos) einfach deaktivieren und mussten gar nicht erst weiteren Programmcode löschen. Nach diesem Muster haben wir auch die Funktionalitäten der von uns nicht genutzten Bauteilen, wie bspw. MP3-Player oder Audio-Speaker, ausgeschaltet. So könnten in einem späteren Schritt ganz einfach die Bauteile angeschlossen werden und es müssten nur die entsprechende Variable am Anfang des Codes aufangesetzt werden. An den genutzten Bibliotheken hat sich in diesem Schritt noch nichts geändert.

Bedienung des Roboters

Die zweite große Änderung, die wir in unserem Projekt vorgenommen haben, war die Änderung der Ansteuerung des Roboters. Klassisch wird der Roboter mittels eines Custom entworfenen Controllers oder wahlweise auch einem PS2-Controller bedient. So stehen dann bestimmte Tastenkombinationen für unterschiedliche vorgefertigte Bewegungen, welche der Roboter ausführen kann, zur Verfügung. Durch die Begrenzung an Tasten sind auch die Tastenkombinationen und somit Bewegungen und Einstellmöglichkeiten begrenzt, dieses wird mit unserer Variante behoben. Wir haben uns dazu entschieden, den Roboter über Wi-Fi mit dem Computer zu verbinden. Umgesetzt wurde dies, indem das Arduino Board als ein Zugangspunkt genutzt wird und wir dann einen Webserver auf einer spezifischen IP-Adresse hosten. Wenn man sich dann beispielsweise mit einem Computer zu diesem Zugangspunkt verbindet und die IP-Adresse in einem Browser aufruft, wird eine Anfrage an den Server gesendet. Der Server antwortet dann mit einer Gruppe von HTML-Befehlen, in unserem Fall verschiedenen Buttons für die Ansteuerung der einzelnen Funktionen, welche im Browser angezeigt werden. Die einzelnen Schritte zur Verwendung dieser Funktion werden später behandelt. Durch die Nutzung von Wi-Fi wird eine grafische Benutzeroberfläche geschaffen, welche eine einfache und intuitive Ansteuerung ermöglicht. Zudem ist es einfach neue Bewegungen hinzuzufügen und der Workflow wird verbessert, da alles an einem Computer umgesetzt werden kann und der Extra-Controller gespart wird. Um dies umzusetzen, mussten viele verschiedene Änderungen am Arduino-Code vorgenommen werden.

Änderungen am Programmcode

Um den Roboter mittels Wi-Fi anzusteuern, wurden verschiedene Änderungen am Programmcode vorgenommen. Zum einen wurden zwei neue Variablen hinzugefügt, welche den Namen des Netzwerks, sowie das Passwort repräsentieren. Um dann die unterschiedlichen Wi-Fi-Funktionen nutzen zu können, musste zusätzlich die Arduino Bibliothek „Wi-Fi-NINA“ installiert werden. Die einzelnen Schritte zum Aufsetzen der Wi-Fi-Verbindungen werden wir hier nicht erklären und verweisen stattdessen auf [diesen Arduino Artikel](#), welcher beispielhaft das Aufsetzen einer Wi-Fi Verbindung demonstriert und an welchem wir uns auch orientiert haben. Die eigentliche Ansteuerung geschieht mit Buttons, welche so modifiziert wurden, dass sie durch Klicken die URL der Seite verändern. Diese Veränderung wird dann von uns überprüft und entsprechend dem geklickten Button werden unterschiedliche Funktionalitäten umgesetzt. Die einzelnen Reaktionen auf die Buttons haben wir dabei nicht selbst programmiert, da dies nur Code-Duplikationen zur Folge gehabt hätte. Dies ist damit zu begründen, dass bereits eine Funktion „serial-commands“ existiert, welche eigentlich zur textuellen Ansteuerung mittels des Seriellen Monitors von Arduino gedacht ist. So werden der Funktion verschiedenen Befehle wie „stop“ oder „march“ textuell übergeben und mittels if-Abfragen die richtige Funktion ausgeführt. Wir rufen diese Funktion also einfach mit den Buttons entsprechenden textuellen Eingaben auf und mussten so nicht nochmal den eigentlich selben Code bei der Wi-Fi-Ansteuerung hinzufügen. Genauso ist es auch möglich den Roboter nur mit den „serial-commands“ anzusteuern, man muss dafür dann aber in Kauf nehmen, dass der Roboter an einem Kabel angeschlossen sein muss.

Anleitung zur Wi-Fi-Verbindung

Im ersten Schritt muss der Programmcode, ganz klassisch mittels Kabel, auf den Arduino Nano geladen werden. Daraufhin muss der Serielle Monitor geöffnet werden. Dies führt dazu, dass der Zugangspunkt kreiert und die IP-Adresse des Boards ausgegeben wird. Mit dem entsprechenden Netzwerk kann sich nun klassisch verbunden werden. Es heißt „NovaNova“ und das Passwort zum Einwählen lautet „12345678“. Dies sind beides Variablen und können nach Belieben geändert werden, es ist nur zu beachten, dass das Passwort und der Netzwerkname mindestens acht Zeichen lang sein müssen. Wenn man sich nun in das Netzwerk eingeloggt hat, kann man die IP-Adresse „192.168.4.1“ im Browser aufrufen, womit sich dann die Website zur Ansteuerung öffnet. Anfangs ist der Button, um alle Bewegungen des Roboters zu stoppen. Darunter kann die Geschwindigkeit der Servomotoren eingestellt werden. Darunter sind die verschiedenen Bewegungen, welche der Roboter ausführen kann. Abschließend sind unten die Sensoren, welche mit einem Klick ein –bzw. ausgeschaltet werden können.

Spätere Änderungen

Da wir nach dem kompletten Zusammenbau des Roboters und einem kurzen Test der Funktionalität auf Probleme stießen und die Servomotoren unvorhersehbare Bewegungen ausführten, mussten wir die Elektronik und damit insbesondere unsere selbst gebaute Platine noch einmal umbauen. Dabei haben wir die Stromversorgung des Nano RP2040 nun über die 3,3V realisiert und die vorher verbauten Sensoren in Form von PIR und Ultraschall nicht mehr angeschlossen.

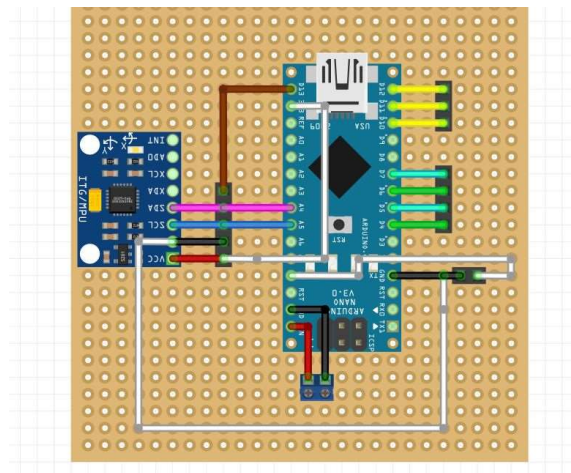


Abbildung 7: Netzplan des vereinfachten neuen PCB-Boards, erstellt in der Software Fritzing

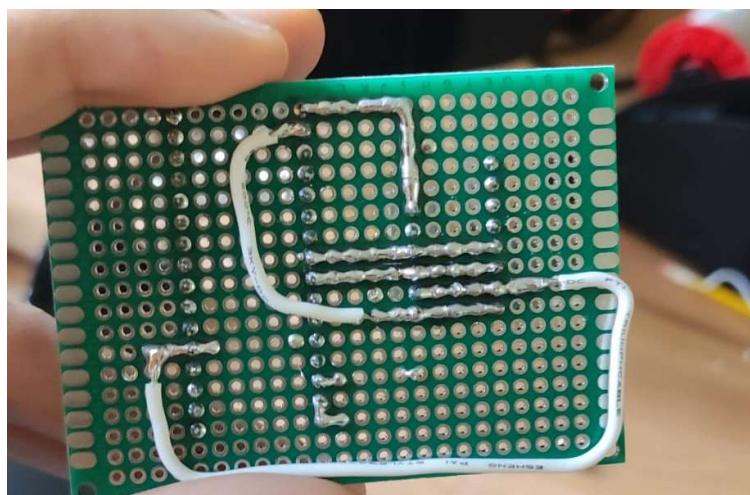


Abbildung 8: Realisierung neuen PCB-Boards

Zudem ist ein neuer Buck-Converter verbaut worden und es wurden dickere Kabel genutzt, um einen größeren Stromfluss zu gewährleisten. Sollten die Ultraschall- und PIR-Sensoren später doch noch eingebaut werden, sollte dieser von uns erstellte Netzplan verwendet werden können.

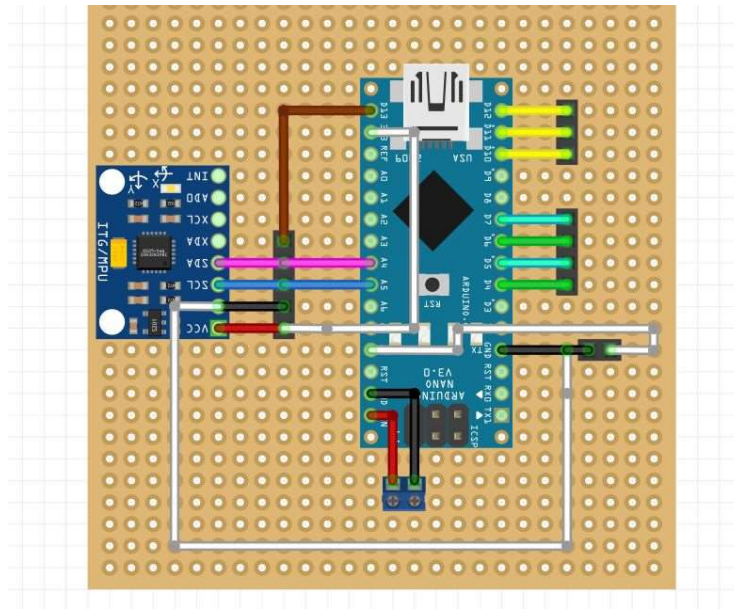


Abbildung 9: Eine mögliche Erweiterung des verwendeten PCB-Boards, erstellt in der Software Fritzing

Fazit und Ausblick

Als Endresultat hat unsere Modifikation einen deutlich vereinfachten Schaltplan und auch elektronische Verkabelung, welche es gleichzeitig einfach ermöglicht Veränderungen vorzunehmen, da beispielsweise auf dem PCB-Board noch Platz ist. Auch der Aufbau des Programmcodes hat sich vereinfacht, da jetzt der gesamte Code nur noch auf dem Arduino läuft. Dadurch ist es beispielsweise einfacher, diesen zu warten oder zu erweitern. Durch den leistungsstärkeren Arduino Nano ist es zudem möglich den Roboter mit Wi-Fi kabellos anzusteuern, wodurch die Ansteuerung mittels PS2-Controller obsolet wird. Die Wi-Fi Ansteuerung ist dabei grafisch über die entsprechende Website umgesetzt und somit einfach zu nutzen. Des Weiteren könnten praktisch beliebig viele unterschiedliche Funktionen hinzugefügt werden und es gibt keine Einschränkung durch die Anzahl an Tasten, wie beim PS2-Controller. Möchte man einfach schnell, sich in der Entwicklung befindliche Funktionen testen, bietet es sich zudem an dies über den Seriellen Monitor und den dort zur Verfügung stehenden „serial-commands“ umzusetzen.

Eine mögliche spätere Arbeit, welche von dem vorherigen Projekt noch nicht aufgegriffen wurde, ist den Roboter, zum Beispiel mittels Reinforcement Learning, laufen lernen zu lassen. Dies könnte man beispielsweise mittels des „DayDreamer“ Algorithmus implementieren und ist inspiriert von diesem [Video](#). Des Weiteren könnte man dem Roboter genau wie dem Roboter von Boston Dynamics ein größeres Repertoire an Sensoren zur Verfügung stellen. Da der jetzige Roboter für die Umwelt intern

nur eine MPU hat und nicht die im Spot enthaltenen Ultraschallsensoren und PIR-Sensoren verwendet, könnte man beispielsweise die auf dem Arduino Nano verbauten Sensoren einbinden.