# Lidar

Lidar steht für „Light detection and ranging“ und ist im Grunde ein Laser Messgerät, welches in mehre Richtung messen kann. Es gibt viele verschiede Lidar, zum Beispiel im neuen IPhone ist auch ein Lidar der nach vorne mit vielen Messpunkten misst. In unsrem fall haben wir ein Lidar der 360 Grad im 2D Bereich messen kann und dieser Lidar Sensor kann mit einer Genauigkeit von 0.15 – 12m Messen mit einer Geschwindigkeit von 6Hz und innerhalb der 6Hz kann man bis zu 2000 Punkte bekommen. Die Daten werden als Polar Koordinaten wieder gegeben.



# Motorsteuerung

Bei der Motorsteuerung mussten, wir uns als erste überlegen was für Motoren wir verwenden wollen. Als ersten haben wir uns für normale DC-Motoren entschieden. Das hat leider nicht funktioniert, da DC-Motoren nicht immer gleich schnell drehen und dadurch der Roboter nicht Geradeaus gefahren ist. Deswegen haben wir uns nach anderen Motoren erkundigt und haben uns schlussendlich für Schrittmotoren entschieden. Die Schrittmotoren habe einen sehr wichtigen Vorteil gegen über DC-Motoren, man kann sie auf Grad genau kontrollieren, sodass alle Motoren gleich schnell fahren können. Ein weiter Grund war das ich diese Motoren noch Zuhause hatte aus einem alten 3D-Drucker.

Als nächstes haben wir uns darüber Gedanken gemacht, wie man die Motoren am besten ansteure kann. Dies wurde durch ein Arduino und ein erweiterungsboard (ramps 1.4) realisiert. Auf dem Board sind vier Schrittmotortreiber (A4988) die den jeweiligen Motor kontrollieren. Da die Schrittmotoren und das erweiterungsboard mit einer Spannung von 12 V arbeiten, aber die beiden Akkus jeweils 7.4 V ausgeben, sind die Akkus in Reihe geschaltet und geben eine Spannung von 14.8V aus. Deswegen ist ein Spannung Wandler zwischen den beiden Akkus und dem erweiterungsboard geschaltet welcher die Spannung auf genau 12V runter regelt.

Auf dem Arduino läuft ein Programm, welches ein Serielles Signal empfängt und in die jeweilige Motor Bewegung umwandelt. Über die Serielle Verbindung werden 15 Byte gesendet:

* 2 Byte: Start(0x3C) und End(0x3E) Byte
* 8 Byte: 2 Byte pro Motor für Stepps (0 - 65,535)
* 4 Byte: Speed pro Motor (0 - 255)
* 1 Byte: Richtung (ersten 4 Bits)

Die Schrittmotortreiber funktionieren so, dass sie immer, wenn sie ein Impuls bekommen, den Motor einen Schritt machen lassen. In der ersten Version des Programmes wurden die Motoren in der normalen „main loop“ angesteuert. Das hat nicht richtig funktioniert, da das Programm zu langsam war oder von der Seriellen Übertragung unterbrochen wurde. Das Probleme wurde dadurch gelöst, dass ein Timer Interrupt genutzt wurde. Dieser wird alle 250ms ausgeführt. Der Timer Interrupt steuert nacheinander die Motoren an, da ein Arduino nicht mehrere Operationen gleichzeitig ausführen kann, aber das wird so schnell hintereinander ausgeführt, dass man keine Verzögerung zwischen der Ansteuerung merkt.

Das Serielle Signal wird von einem Raspberry Pi übertragen. Auf dem Raspberry Pi läuft ein TCP Server, der übers Internet Signale von der App anfängt und diese dann an den Arduino weiterleitet.

# 3D-Model und Drucken

Das 3D Model wurde mit dem Programm Fusion 360 erstellt. Als erstes habe wir ein Prototype erstellt, um alles testen zu können. Danach wurde das Finale 3D Model erstellt mit Verbesserungen. Das ganze Model wurde mit einem 3D Drucker ausgedruckt. Da der Roboter größer ist, als der 3D Drucker ist, musste der Roboter in vier Teile aufgeteilt werden und nacheinander ausgedruckt werden. Dazu kommt noch der Deckel, der auch in vier Teile aufgeteilt werden musste.

Da der Raspberry Pi und die Treiber der Schrittmotoren sehr viel Hitze prodozieren mussten wir im 3D-Model Lüfter einplanen. Schluss endlich haben wir uns für vier Lüfter entschieden. Damit ein Luftstrom entsteht, haben wir uns für zwei Lüfter, die rein pusten und zwei Lüfter, die rausziehen entschieden.

Ein Paar Teile wie der Lidar [1] oder die Lüfterhalter [2] habe ich nicht selbst konstruiert. Bei diesen 3D-Modelen habe ich vorgefertigtes 3D-Model von der Seite „thingiverse“ genommen.

Ebenso musste ein Adapter konstruiert werden, der die Motoren mit dem Rad verbinden. Da dieser Teil sehr klein ist und die Motoren nahezu eine Runde Oberfläche haben verschleißen die Teile sehr schnell und es kann passieren das der Motoren sich drehen, ohne die Räder zu drehen.



# Lidar Mapping V3

In der dritten Version von dem Lidar Mapping haben wir versucht einen sehr Simplen weg zu gehen. Bei der zweiten Version haben wir das Problem, das wir die Drehung nicht rausbekommen haben, deswegen habe wir versucht als erstes die drehen zu errechnen und danach die Verschiebung.

Die Drehung wird so ausgerechnet das ein Fehler ausgerechnet wird für alle Winkel mit den Dazugehörigen längen und das wird immer wieder gemacht, bis alle Winkel um 359 Grad weitergeschoben wurden. Das wird immer wieder abgespeichert mit dem dazu gehörigen Fehler und dem Winkel. Danach wird der kleinste Fehler rausgesucht und der dazugehörige Winkel ist die Drehung. Das funktioniert gut, solange man den Lidar nicht mehrere Zentimeter verschiebt. Die Genauigkeit bei einer Drehung auf der Stelle liegt bei ±1 Grad.

Danach wird die Verschiebung ausgerechnet. Als erstes werden die Polar-Koordinaten in xy-Kordinaten umgewandelt. Nach dem die Karte richtig gedreht ist müsste rein theoretisch die Deltas zwischen den Punkten mit dem gleichen Winkel die Verschiebung sein. Das kommt nicht immer hin, da es punkte gibt, die nicht gefunden wurden oder hin und her springen. Im erstes versuch wurde jeder punkt mit jeden vergleichen und dann das Delta in Liste gespeichert. Wenn genau das gleiche Delta wieder gefunden wird, wird eine Zahl hochgezählt. Die Zahl mit dem höchsten Wert ist die Verschiebung. Das hat leider nicht funktioniert, da auch punkte vergleicht wurde die nichts miteinander zu tun haben oder ziemlich weit voneinander entfernt sind. Deswegen wird das Koordinatensystem in 4 Bereiche aufgeteilt. In diese 4 Bereiche wird dann wieder von jedem punkt jedem andren Punkt das Delta ausgerechnet und weggespeichert. Danach werden von allen 4 Bereiche die Listen zusammengesetzt und das Delta mit der Höchsten Wert ist die Verschiebung. Das funktioniert gut mit einer Genauigkeit von ±1 cm.

Umso weiter man den Lidar vom Ursprung entfernt um so ungenauer werden die Werte deswegen werden alle 3 cm neu Referenz daten genommen. Mit dieser Methode kann gut eine Karte erstellen. Die größten zwei Problem sind zu einem der Lidar selbst da er nicht schnell genug daten liefern kann und zum anderen das der Fehler, der entsteht immer größer wird um so mehr man den Lidar von seiner Start Position entfernt. Das resultiert darin das die gescannten Objekte oder Wende immer dicker werden.

# Quellen:

[1] <https://www.thingiverse.com/thing:1486705>

[2] <https://www.thingiverse.com/thing:4827203>