Dokumentation Team 2

**1.1 Motorisierung:**

Die Motorisierung unseres Roboters beinhaltet zwei Achsen (Eine vordere und eine hintere), welche jeweils mit einem Motor und zwei Rädern bestückt sind.

Der hintere Motor hat den Nutzen den Roboter anzutreiben (Antriebsmotor), währenddessen der vordere Motor dafür sorgt, dass der Roboter in der Spanne von 1° bis 360° je nach Situation richtig lenkt (Lenkungsmotor).

(Genaue Werte und Bilder zum Motortyp sind im Anhang und in der technischen Zeichnung hinterlegt).

**1.2 Energie und Sensoren**

Als Energiequelle wird bei unserem Roboter der Lego Spike Hub verwendet. Dieser besitzt sechs individuelle Anschlüsse, welche wir auch alle ausnutzen. Zwei werden für die Motoren verwendet, zwei andere für Distanzsensoren und die letzten zwei Anschlüsse werden für zwei Farbsensoren verwendet.

Die Distanzsensoren sind am vorderen seitlichen Part des Roboters angebracht unmittelbar über den Vorderreifen. Sie haben die Aufgabe den Roboter in der Spur zu halten und sich mithilfe der Wände auf dem Spielfeld zu orientieren, indem unser Programm – Code einen gewissen Abstand vorrausetzt, welcher dann eingehalten wird um nicht eine der Wände zu fahren.

Aber die Distanzsensoren haben auch noch einen Nutzen, was das Hindernisrennen betrifft. Wenn unser Roboter an einem Hindernis vorbeifährt und die Farbsensoren dieses nicht erkennen reagieren die Distanzsensoren dementsprechend. Falls diese nämlich einen Abstand von 15cm oder weniger erkennen, geht der Roboter in eine Prozedur über um das Hindernis zu umfahren und danach wieder mittig auf die Fahrbahn einzuscheren. Ein großer Teil unserer Hindernisüberwindung ist nämlich möglichst immer mittig zu fahren, also auf der Höhe der Startzonen.

Die Farbsensoren, welche sich vorne am Roboter befinden und beide gerade aus schauen sind dazu da um Hindernisse zu erkennen welche sich frontal vor dem Roboter befinden. Sie sind mit der vorderen Achse also gleichzeitig auch der Lenkachse verbunden wodurch sie immer in Fahrtrichtung schauen. Wir verwenden zwei, weil ein Farbsensor nicht genug wäre um die Breite des Roboter abzudecken und so würde er die Hindernisse nicht erkennen und dementsprechend gegen sie fahren.

(Genaue Werte und Bilder zur Energiequelle und den Sensoren sind im Anhang und der Technischen Zeichnung hinterlegt).

Ein Bild, das Text, Screenshot, Rechteck, Diagramm enthält.

Automatisch generierte BeschreibungTechnische Zeichnung des Roboters:

**1.3 Hindernisse**

Wie bei unseren Sensoren bereits erläutert wurde nutzt der Roboter zum Erkennen und Umfahren von Hindernissen zwei Distanzsensoren, welche sich an den vorderen Seiten über den Vorderreifen befinden und zwei Farbsensoren welche sich am vorderen Teil des Fahrzeugs befinden und mit der vorderen Achse verbunden sind. Unser Roboter kann mit Hilfe dieser Sensoren auf mehrere verschiedene Weise Hindernisse erkennen und umfahren.

Die erste Möglichkeit beinhaltet die vorne angebrachten Farbsensoren. Wenn diese ein Hindernis erkennen schauen sie sich zuerst die Farbe an um zu wissen an welcher Seite er vorbei fahren muss. Darauf folgend misst er den Seitenabstand auf beiden Distanzsensoren um zu wissen aus welchem Winkel er das Hindernis anfährt. Erst dann fährt der Roboter um das Hindernis herum, lenkt dann aber nachdem er das Hindernis umfahren hat so ein, wie es der vorher gemessene Seitenabstand gemessen hat, ein und fährt dann weiter.

Eine weitere Möglichkeit ist das Erkennen von Hindernissen mit den Distanzsensoren, wenn der Roboter seitlich an einem vorbei fährt. In diesem Fall setzt der Roboter ein Stück zurück, lenkt ein Stück Richtung Hindernis ein, schaut es sich mit den Farbsensoren an, setzt dann wieder ein Stück zurück und lenkt abschließend in die Richtung ein in welcher er das Hindernis umfahren muss. Nach dem umfahren soll der Roboter wieder relativ mittig enden und geht wieder in die Schleife über zu beiden Seiten einen gewissen Abstand zu halten sodass er mittig fährt und das nächste Hindernis erkennen kann.

Eine letzte Möglichkeit funktioniert so, dass der Roboter in jeder Kurve die er fährt eine Stoppuhr hochlaufen lässt solange er einlenkt. Diese Stoppuhr wird sobald der Roboter aus der Kurve fährt auf Null zurückgesetzt. Wenn ein beliebiger Sensor, während die Stoppuhr läuft (Also in der Kurve), ein Hindernis erkennt korrigiert der Roboter seine Fahrbahn indem er rückwärts fährt und sich wieder gerade auf die Fahrbahn begibt um dann das Hindernis hinter der Kurve ganz normal zu erkennen und zu umfahren wie vorher bereits erläutert.

**1.4 Fotos:**

Fotos zum Gesamtaufbau des Roboters aus verschiedenen Perspektiven sind auf Github zu finden.

**1.5 Videos:**

Videos wie der Roboter das Eröffnungsrennen bewältigt sowie das Umfahren von Hindernissen sind auf Github zu finden.

**1.6 Anhang**:

Daten zum Motor:

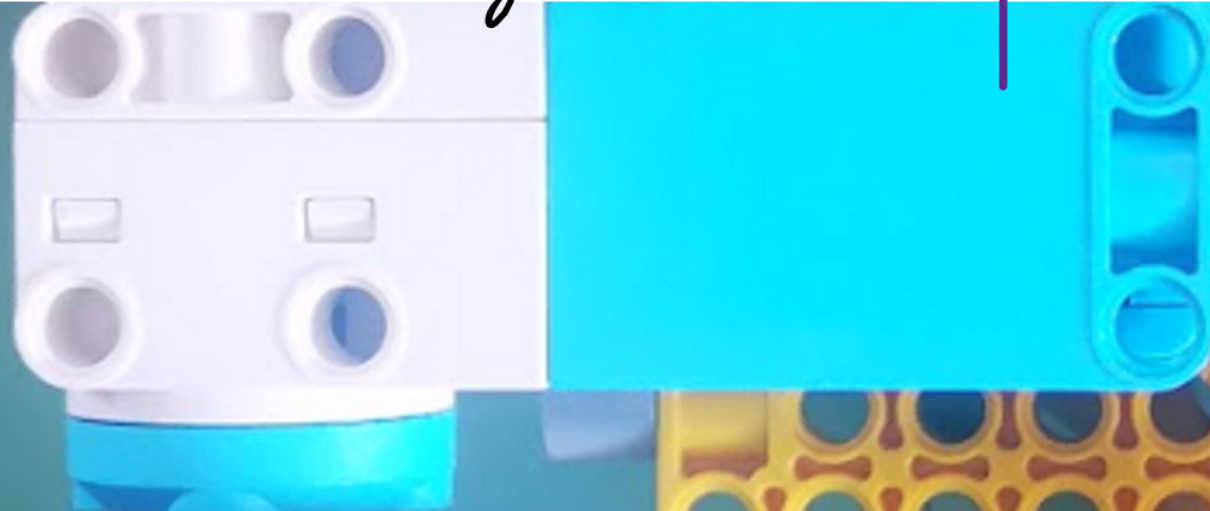
Motor (Antrieb):

* Spannungsbereich: 3,3 bis 6 Volt
* Drehmoment: 1,8 Ncm
* Geschwindigkeit: 85 Umin (+/- 15%)
* Leistungsaufnahme: 170 mA (+/- 15%)



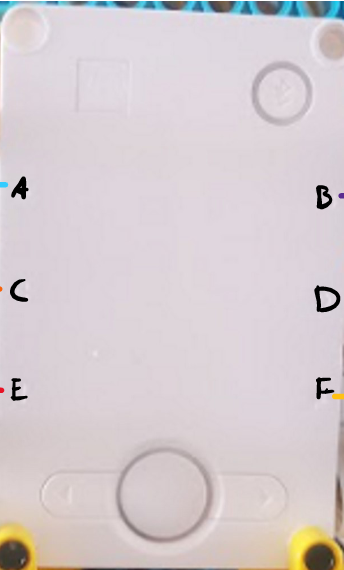
Motor (Lenkung):

* Spannungsbereich: 3,3 bis 6 Volt
* Drehmoment: 1,8 Ncm
* Geschwindigkeit: 85 Umin (+/- 15%)
* Leistungsaufnahme: 170 mA (+/- 15%)



Daten zur Energiequelle und Sensoren:

Energiequelle:

* Lego Spike Hub (Leistung : 5 Volt)
* Akku (Lithium – Ionen – Akku) (7,3 Volt und 2100 mAh)

Lego Spike Hub

Sensoren:

* 2x Farbsensor (Erfassungsrate 1 kHz, 8 Farben erkennbar, misst Intensität von reflektiertem weißen Licht, Optimaler Abstand: 16mm)

Linker Farbsensor Rechter Farbsensor

* 2x Distanzsensor (Messbereich: 1 bis 200 cm, Messgenauigkeit: +/- 1Sek, Eintrittswinkel: +/- 35°, Austrittsauflösung: 1mm, Frequenz: 100 hz )

Ein Bild, das Screenshot, Text, Design, Lautsprecher enthält.

Automatisch generierte BeschreibungEin Bild, das Text, Screenshot, Lautsprecher, Design enthält.

Automatisch generierte BeschreibungLinker Distanzsensor Rechter Distanzsensor

**1.7 Entwicklungsprozess:**

24.1.2023

Wir haben, nachdem wir uns in der letzten Stunde mit den Regeln und Vorgaben vertraut gemacht haben, heute angefangen unseren Roboter umzubauen.

Wir haben als erstes den Motor, an welchen nur der Bewegungssensor angeschlossen ist, anders platziert, indem wir ihn hochkant aufgestellt haben. Dies soll bewirken, dass sich der Sensor mithilfe eines Drehmechanismus kontinuierlich nach rechts und links dreht und so im Weg stehende Hindernisse erkennt und dementsprechend reagiert.

Das nächste Problem ist die Reichweite, die der Sensor beim Drehen abdeckt. Wir versuchen nun diese zu erhöhen, da er sonst nicht alle Hindernisse erkennt welche sich etwas frontal etwas seitlicher befinden. Unser erster Lösungsansatz ist die Drehachse an einem Legostein zu befestigen, welcher so die Reichweite erhöht um die er zurückgezogen und vorgeschoben wird.

31.1.2023

Unser erster Schritt heute ist die Achsen umzubauen, damit sie dem Regelkatalog entsprechen. Nach dem anbringen der neuen Vorderachse, muss nun die Hinterachse auf dieselbe Höhe gebracht werden, da der Sensor sonst zu weit nach oben schaut und die Hindernisse welche frontal kommen nicht erkennt. Wir haben nun die hintere Achse noch tiefer gelegt, damit der Sensor eine Neigung nach unten bekommt, wodurch er wieder alles sehen kann. Wir müssen die vordere Achse noch stabilisieren, da sie unter dem Gewicht des Roboters etwas einknickt.

7.2.2023:

Heute müssen wir die vordere Achse stabilisieren, da wir eine in den Regeln nicht erlaubte Kugel als Stabilisator verwendet haben. Wir haben es nun geschafft.

7.3.2023:

Am heutigen Tag werden wir an der Optik unseres Roboters arbeiten und versuchen ihn möglichst zu verkleinern, sodass er mobil ist und besser durch den Parkour fahren kann.

Eine größere Veränderung haben wir den Sensor, welcher vorher an einem kleinem Turm vorne dran mit einem individuellen Motor befestigt war, etwas tiefer gebaut was zu einem Gewichtsverlust geführt hat und zu einer niedrigeren Lenkbelastung. Durch diese Änderung hat der Sensor zwar seine Bewegungsfähigkeit verloren sich nach links und rechts zu drehen, aber wir haben beschlossen, dass er auch in festem Zustand genug Reichweite hat. Außerdem haben wir den Antriebsmotor vom hinteren Teil des Fahrzeugs, an den unteren Teil platziert um eine bessere Fahrdynamik zu erzielen. Zusammenfassend haben wir die Größe des Fahrzeugs um einiges reduziert und unnötige Teile und Funktionen entfernt.

18.4.23

Der Bau des Roboters ist nun bis auf weiteres aufgeschlossen und der Fokus liegt auf dem Programmieren des Fahrzeugs.

Wir werden wenn nötig Veränderungen vornehmen wenn wir in der Lage sind die ersten Fahrtests durch zu führen mit der dementsprechenden Unterlage.

2.5.23

Als erstes beschäftigen wir uns heute mit dem Code des Farbsensors, welcher eine wichtige Rolle im Projekt einnimmt. Nach einigen Code-Fehlern haben wir es geschafft, dass unser Roboter fährt, Farben erkennt und auch für einen kurzen Moment anhält. Als wir diesen Code fertig hatten, haben wir die ersten Fahrtests gemacht und uns ist aufgefallen, dass der Sensor zu hoch angebracht ist, weshalb wir ihn mit einer zusätzlichen Lego-Konstellation tiefer anbringen mussten, was auch sofort funktioniert hat. Am Ende haben wir noch kurz ein paar Fahrtests gemacht um zu schauen wie der Roboter die Kurven nimmt.

9.5.23

Durch Komplikationen mit einem Update der Lego-Spike App sind wir gezwungen statt Python mit den Textblöcken von Lego-Spike zu arbeiten. Unser heutiges Ziel ist es, dass der Roboter eine volle Runde über die Matte fahren kann, ausgenommen der Hindernisse welche wir erst später in Betracht ziehen.

10.5.23

Da die Lenkung zu viel Spiel hat und wir so nach einer Kurve keine gerade Strecke fahren können, müssen wir die Lenkung mehr befestigen.

15.5.23

Da die Lenkung trotz dessen immer noch zu viel Spiel hat wollen wir den Roboter anderswertig gerade fahren lassen, indem wir an vorerst einer Seite einen Infrarotsensor anbringen, welcher je nach zu viel oder zu wenig Abstand zur Wand dementsprechend gegenlenkt und so in der Spur bleibt. Das Grundprinzip funktioniert soweit aber wir wollen die Distanz von 12 cm noch etwas erhöhen, damit der Roboter nicht zu schnell in der Drehung an der Wand vorbeischaut. Die Freude ist sehr groß da wir den Roboter so hinbekommen haben, dass er kontinuierlich an Wänden entlang fahren kann, Mithilfe des Infrarotsensors.

16.5.23

Wir beginnen den Tag indem wir nun auf der anderen Seite die gleiche Infrarotsensor - Konstellation anbringen und so in beide Richtungen eine Erkennung haben. Außerdem haben wir vorne 2 Farbsensoren angebracht welche schon für den Hindernis - Teil gedacht sind.

23.5.23

Heute testen wir unseren Roboter das erste Mal auf der gesamten Matte. Er schafft es mehrere Runden um die Mitte zu fahren, aber scheitert falls er zu weit weg von der dieser startet (mehr als 35 cm).

30.5.23

Heute treffen wir uns das erste Mal außerhalb der Schule um am Roboter zu bauen. Wir haben erstmal das Spielfeod aufgebaut, ein wenig kleiner als das Original .

Da der Roboter das Eröffnungsrennen in ungefähr 7 bis 8 Fällen von 10, wollen wir mehr Konstanz rein bringen. Der Versuch die Ultraschallsensoren etwas weiter hinten am Roboter zu montieren hat leider nicht funktioniert. Außerdem haben wir die Einlenkung von 30 Grad auf 25 Grad hinabgestuft, was auch keine positive Veränderung gezeigt hat. Eine weiterer Versuch ist die Einstellung zu entfernen, dass wenn links und rechts der Abstand gleich ist, dass dann keine Lenkung vorgenommen wird. Diese Änderung nehmen wir vor, da er bei Kurven manchmal einfach geradeaus gefahren ist.

Nach einem Test erwies sich diese Umstellung nicht als Lösung für das ursprüngliche Problem, weshalb wir sie wieder zurück setzen.

Der nächste Versuch ist während einer 90° Kurve eine 0,5 sekündige Pause einzubauen damit er nicht geradeaus fährt. Diese Dauer war zu kurz, weshalb wir sie auf 2,5 Sekunden erhöhen was mit der Geschwindigkeit ungefähr eine 90° Kurve ergeben sollte. Da Probleme mit dem Spielfeld entstanden sind. Welche auf die schiefen Wände zurückzuführen sind, bauen wir es nun um.

Wir haben beschlossen, nachdem alles vorherige nicht funktioniert hat, das Programm komplett umzuschreiben und das ursprüngliche zu ignorieren. In diesem ist die Idee vorerst geradeaus fahren und dies solange bist einer der Ultraschallsensoren einen sehr viel größeren Abstand aufweist als der andere. Von diesem Moment an weiß er in welche Richtung er sich bewegen muss und orientiert sich an der inneren Wand und versucht einen Abstand von 25 cm einzuhalten.

2.6.23

Heute haben wir nochmal alles umgeworfen und ein komplett neues Programm geschrieben. Wir haben als erstes die Vorderachse verkleinert, damit der Roboter besser durch die Hindernisse kommt. Außerdem haben wir die Position der Sensoren deutlich tiefer gelegt. Dies hat den Grund, dass sich nun die Sensoren parallel mit den Reifen drehen.

Zum Programm:

Der Roboter fährt zuerst gerade aus los und überprüft dabei ob der Abstand links oder rechts größer ist als der andere. Falls dies der Fall ist biegt er in die jeweilige Richtung ab und orientiert sich ab dann mit einem vorgegebenen Abstand von 42 cm an der inneren Wand des Spielfelds.

Während er die Runden durchfährt überprüfen die Farbsensoren ob sich rote oder grüne Hindernisse im Weg befinden. Ist dem so, umfährt das Auto das Hindernis in vorgegebener Richtung.

14.6.23

Nach unserem Wettbewerb in Rüsselsheim haben wir uns als Ziel gesetzt einen großen Teil der Kategorien komplett umzusetzen was das Eröffnungsrennen mit Stoppen beinhaltet und ein fehlerfreies Hindernisrennen, bei welchem wir das Stoppen aber vorerst hinten anschieben, da es an sich schon sehr schwierig ist.

Die erste bedeutende Veränderung welche wir vorgenommen haben ist einen kompletten Umbau des Roboters. Dieser beinhaltet eine deutliche Verkleinerung welche Länge und Breite mit ein schließt. Die Idee dahinter war den Roboter kleiner und somit automatisch lenkfähiger und wendiger zu gestalten, damit dieser vor allem leichter um Hindernisse fahren kann. Außerdem hatten wir die Höhe bis jetzt kaum ausgenutzt was wir nun etwas mehr getan haben. Ein anderer Umbau umschließt die Distanz Sensoren, welche nun direkt über den Reifen sitzen und so deutlich weiter nach vorne verschoben wurden. So erhält unser Roboter eine bessere Reaktionszeit mit Blick auf den eigentlichen Rumpf.

Diese Umbauten haben natürlich dazu geführt, dass wir beide Rennen und dessen Programme komplett neu programmieren mussten.

Beginnend mit dem Eröffnungsrennen haben wir uns wieder als Idee genommen uns an der inneren Wand zu orientieren und das mit einem bestimmten Abstand und wenn dieser Abstand sehr groß wird soll er in diese Richtung einlenken und so die Kurve fahren. Danach orientiert er sich wieder an der inneren Wand bis zur nächsten Kurve. Diese Programmierung ging relativ zügig, sodass wir uns danach auf das Stoppen konzentriert haben. Dies brachte einige Probleme mit sich. Als erstes mussten wir natürlich die Idee finden auf welche Weise er stoppen sollte. Die erste Idee war einen Farbsensor welcher nach unten schaut und dort Linien zählt und nach einer bestimmten Anzahl stoppt. Das war aber leider nicht umsetzbar, da wir zwei Farbsensoren zur Hinderniserkennung brauchen und insgesamt nur sechs Anschlüsse haben. Die zweite Idee war eine Variable. Welche mit jeder Kurve größer wird und dann stoppt, wenn das Limit erreicht wird. Diese Idee haben wir auch ausprobiert, aber leider wurde diese Variable in nur einer Kurve sehr schnell extrem groß, da die „Wiederhole Fortlaufend – Schleife“ dazu führte dass es extrem schnell wiederholt wurde. Danach ist uns beim stöbern in den Programmmöglichkeiten eine Stoppuhr ins Auge gefallen, welche sich als sehr passend erwies. Unsere Umsetzung lief so, dass wir den Timer der Stoppuhr in jeder Kurve laufen lassen, da diese immer gleich sind und nicht mit dem zufälligen Spielfeld Aufbau variieren. Da aus einem unerkenntlichen Grund der Roboter für die eine Richtung eine andere Zeit in Anspruch nimmt als für die andere, ist uns die Idee gekommen, dass der Roboter die Zeit in der ersten Kurve setzt. Erläutert setzt er ein Maximum für die Stoppuhr je nach dem in welche Richtung er einen sehr großen Abstand erkennt.

Beim Hindernisrennen hatten wir seit dem ersten Wettbewerb eine klare Strategie welche wir verfolgt haben. Diese umfasst, dass die zwei vorhandenen Farbsensoren immer von vorne auf das Hindernis blicken können und der Roboter dementsprechend einlenken kann um das Hindernis zu umfahren. Uns ist aber relativ früh aufgefallen, dass der Roboter nicht gerade von vorne auf jedes Hindernis zufahren kann. Deshalb ist uns nach einiger Überlegungszeit die Idee gekommen die Distanzsensoren für dieses Problem zu verwenden. Wie schon erwähnt werden diese bezüglich der Hindernisse aktiv, falls ein Seitenabstand von 15 cm oder weniger auftritt. Der Roboter wird zurückgesetzt damit wir wieder in die Situation kommen wo die Farbsensoren vor dem Hindernis stehen.

15.6.23

Diese beiden Methoden haben wir versucht so gut wie es geht überall unterzubringen. Aber unser Hauptproblem ist, dass es einfach zu viele Situationen gibt in welcher diese zwei Methoden nicht funktionieren. Kombiniert mit dem Problem, dass unser Programmtext laut unserer App zu lang ist mussten wir lange überlegen wie wir diese Schwierigkeit lösen. Als Knackpunkt bei dieser Lösung haben wir die Funktion entdeckt wie wir diese individuellen Situationen alle einzeln programmieren können, denn wir haben das Länge – Problem mithilfe einer Funktion entdeckt, welche uns erlaubt nur einen einzigen Programmblock zu verwenden um die „Prozedur“ aufzurufen, indem wir die eigentliche Methode benennen und einfach neben das Hauptprogramm schreiben und so sehr viel Platz sparen. Auf diese Weise können wir jede individuelle Situation, welche er nicht automatisch löst, einzeln programmieren, was uns sehr weiter hilft. Darunter fallen zum Beispiel die Situationen, wenn ein Hindernis sofort hinter einer Kurve steht, wo der Anfahrwinkel des Roboters auf das Hindernis vom normalen abweicht.

Zum Ende hin werden wir nun alles was wir für die eine Richtung geschafft haben auch für die andere Fahrtrichtung versuchen umzusetzen.