**Die Alltagshilfe**

Ein Projekt von Vincent Schmandt und Fabian Specht

*vom Stefan - George – Gymnasium*

Inhaltsverzeichnis:

Unsere Idee:

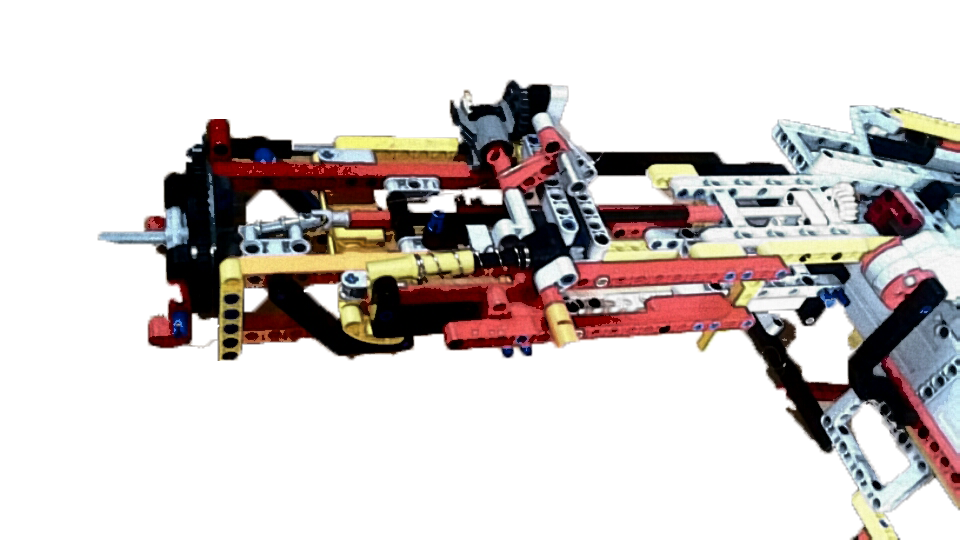
Da wir beide schon an dem Wettbewerb „Jugend forscht/ Schüler experimentieren“ teilgenommen hatten und es uns sehr viel Spaß gemacht hatte, wollten wir erneut teilnehmen. Diesmal kamen wir auf die Idee, eine Einkaufshilfe für körperlich eingeschränkte Menschen zu konstruieren, der ihnen ihre Einkäufe nach Hause tragen kann. Anfangs hielten wir es für keine besonders große Herausforderung, aber mit der Zeit kamen immer mehr größere Probleme auf uns beide zu, welche wir zu bewältigen hatten um unser Projekt fertig zu stellen.

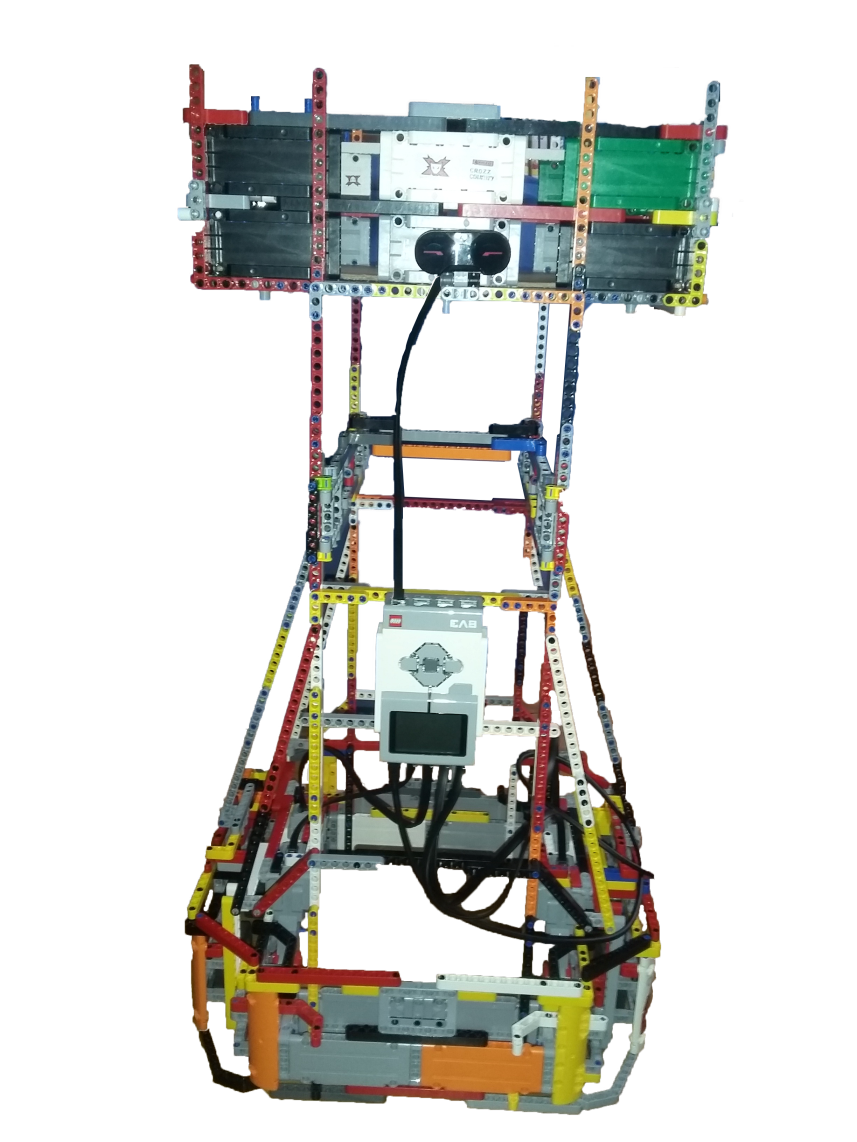
Der Bau:

Als erstes machten wir uns daran, ein solides Grundgerüst für den Roboter zu schaffen. Die Idee dazu war folgende: Wir hatten vor, an unterster Stelle zwei Raupenketten mit jeweils zwei Federungen zu befestigen. Dies war keine schlechte Basis für unsere damaligen Vorstellungen, allerdings kam uns nach einer Weile die Idee, dass der Roboter auch Treppen steigen können sollte.

Daher bauten ihn um und nutzen nun vier Arme, die auf die zwei Außenseiten des Roboters verteilt wurden. Diese sollten hoch und runter klappbar sein, sodass sie den Roboter auf die Stufen stemmen können. Außerdem sollten sich an den Enden der Arme Räder befinden welche dann, als Ersatz für die Ketten, zur Fortbewegung dienen sollten. Die Arme sollten jeweils von einem Motor betrieben werden, welcher seine Kraft über eine Flaschenzugähnliche Konstruktion aus Nylonfäden daran weiter geben sollte.

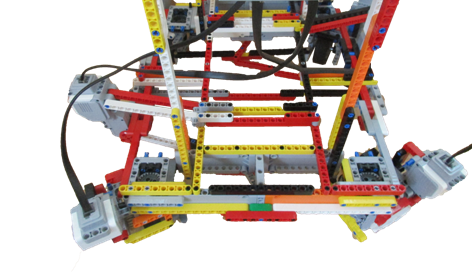
Hierzu ein Bild:



Dies funktionierte allerdings nicht. Denn wir hatten bei der Planung einen entscheidenden Faktor vergessen: Die Motoren, sowie auch anderen Bauteile waren nicht dafür geschaffen solche Lasten zu heben und schafften es daher nicht denn Roboter höher als wenige Zentimeter anzuheben. Der Einbau eines Getriebes war aufgrund der Geringen Belastbarkeit auch nicht möglich. Letztendlich gaben wir die Idee des Treppensteigens auf und widmeten uns dem eigentlichem Hauptbestandteil des Projekts, welcher das Folgen einer Person und das transportieren vom Gegenständen ist.

Wir nahmen den Rahmen, der schon seit Beginn des Projekts vorlag, bauten die Arme wieder ab und befestigten vier unabhängig voneinander angetriebene Räder, direkt, ohne Federung oder ähnlichem, am Gestell. Auf dem Fahrgestell befestigten wir in einer gut erreichbaren Höhe eine Transportplattform, in welcher zum Beispiel die Einkäufe transportiert werden sollten.

Da sich dieser Ansatz allerdings als sehr schlecht zu lenken erwies bauten wir das Grundgerüst schließlich doch noch einmal grundlegend um und bauten eine mechanische Lenkung mit Getrieben, wie man sie von Offroadern kennt, ein.

Dies klappte dann soweit gut.

Die Programmierung

Anfangs arbeiteten wir um den Roboter zum Fahren zu bringen mit der grafischen LEGO-EV3-Software. Das ging auch solang gut, bis zu dem Zeitpunkt, ab dem wir kompliziertere Sachen wie Multithreading oder eine Verbindung zu einem Smartphone, um dem Roboter darüber Sprachbefehle zu übermitteln, einbringen wollten.

Wir stiegen auf LeJOS, ein inoffizielles, auf einer JavaVM basierendes Betriebssystem für den EV3 um, wodurch wir den EV3 nun mit klassischen Java-Programmen steuern konnten. Unser Wissen dazu hatten wir uns aus dem Internet durch zahlreiche Tutorials und langes Googeln angeeignet.

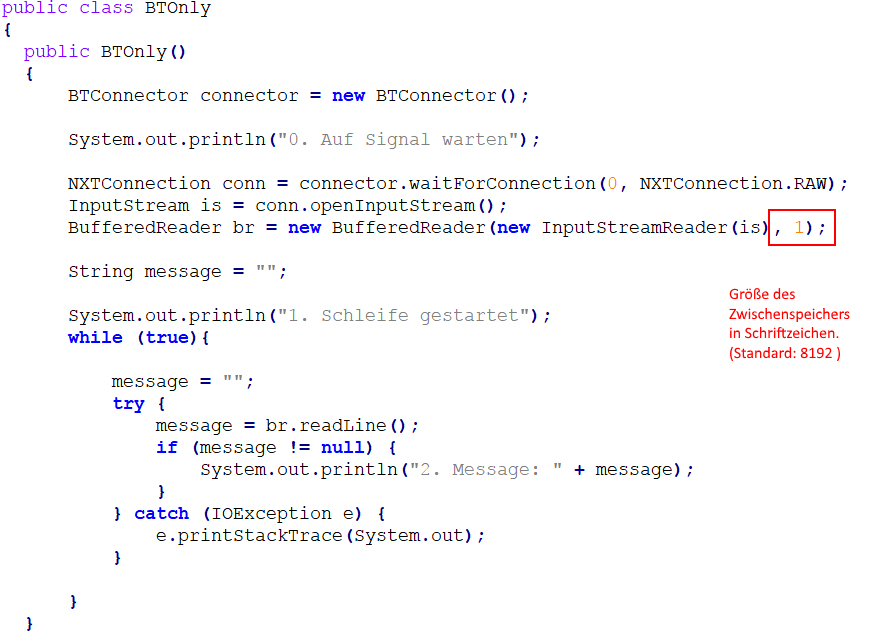
Doch auch das war nicht einfach, denn zum einen hatten wir dadurch eben keinen direkten Ansprechpartner, wie zum Beispiel einen Lehrer, den wir im Zweifelsfall einfach mal hätten ansprechen können, sondern waren auf uns alleine gestellt.

So programmierten wir dann zunächst eine simple Fahrweise basierend zum einen auf der relativen Position zum Infrarotsender, welchen der Besitzer des Roboters bei sich trägt, und den Werten der restlichen Sensoren am Roboter, wie unter anderem mehreren Ultraschallsensoren zum Erkennen von Hindernissen oder Treppen. Dieses Konzept funktionierte nach einigen (auf Seite 3 und 4 behandelten) Verbesserungen und Umbauten am Roboter auch recht gut.

Nun wollten wir Zusatzfunktionen wie die Sprachsteuerung via Smartphone einbinden, da es vor allem für ältere Menschen, welche die Zielgruppe unseres Projekts sind, von Vorteil ist einfach „Stopp“ oder andere Befehle sagen zu können, anstatt jedes Mal die Fernbedienung zu holen um den Roboter anzuhalten oder dessen Ladeklappe zu öffnen. Dazu brauchten wir zuerst eine App, welche im Hintergrund auf dem Handy läuft und sobald sie einen bestimmten Befehl erkennt dem Roboter Bescheid gibt. Da wir allerdings in der Lage waren, dies alleine umzusetzen, holten wir uns Hilfe von einem Informatikstudenten, Lukas Justen. Mit seiner Unterstützung schafften wir es dann nach einiger Zeit, eine Spracherkennung zu Programmieren.

Doch das war erst der einfachere Teil. Nun mussten wir es noch schaffen, dem Roboter mitzuteilen, dass ein bestimmter Sprachbefehl eingegangen ist. Wir entschlossen uns, das ganze über Bluetooth zu machen, da dies bei sowohl dem Handy, als auch dem EV3 bereits standardmäßig vorhanden ist. Dies schien zuerst kein Problem, da eine Kommunikation per Bluetooth in der heutigen Zeit nichts Ungewöhnliches mehr ist und es dazu genug Material im Internet gibt. Allerdings nicht für eine so ungewöhnliche Verbindung wie die zwischen einem Smartphone und einem EV3. Folglich mussten wir improvisieren: Wir versuchten Projekte zum Datenaustausch zwischen Raspberry Pi´s und Handys so anzupassen, dass wir sie für unsere Zwecke nutzen konnten. Das klappte allerdings nicht. Wir vertieften also unser Wissen über Java und schrieben schließlich unser eigenes Script. Als dies auch nicht funktionierte, holten wir uns erneut Hilfe von Lukas. Aber auch er konnte zuerst keinen Fehler in unserem Code feststellen. Es dauerte gut zwei Monate, bis wir schließlich den Fehler fanden. Der Fehler bestand darin, dass wir bei der Software des EV3s den Zwischenspeicher des „BufferedReaders“ (liest Bytes, welche z.B. per Bluetooth empfangen werden, aus) nicht definiert hatten und dieser daher so groß war, dass die empfangenen Nachrichten erst ausgegeben worden wären, wenn diese standardmäßig viel zu große Größe, erreicht worden wäre, was allerdings bei unseren sehr kleinen Nachrichten nie eingetroffen wäre.

Hier unser Code dazu:



Als wir dann diesen Wert hinzugefügt hatten, funktionierte die Kommunikation einwandfrei. Das einzige was noch zu tun war, war die Umsetzung der Befehle.

Doch zuerst überarbeiteten wir noch einmal grundsätzlich unser komplettes Programm und lagerten das Empfangen der Nachrichten auf einen zweiten Thread. Außerdem basierte unser neues Programm zum Auswerten der Sensor- und Positionswerte nun im Gegensatz zu unserem ersten Programm nicht mehr auf verschiedenen Bereichen aus denen die Infrarotstrahlen kommen, sondern auf einer Formel zum errechnen passender Geschwindigkeits- sowie Richtungswerte. Dadurch (und durch den Einbau einer Lenkung sowie eines Getriebes) wurde das Fahren deutlich präziser, schneller und flüssiger.

Allerdings ergab sich auch hierbei ein kleines Problem. Da die Lego-Motoren erneut deutlich zu schwach waren, waren wir gezwungen diesen Mangel auszugleichen. Diesmal nutzten wir die Hebelkraft aus um die Leistung der Motoren zu steigern. Nach dieser Veränderung schlossen wir diesen Part ab und widmeten uns dem hinzufügen weiterer nützlicher Features. Unsere erste Idee war es in die ohnehin bereits vorhandene App eine Einkaufsliste mit Sprachsteuerung zur einfachen Bedienung zu integrieren. Dies war an Sich absolut kein Problem außer, dass wir einige Probleme dabei hatten die App auf mehrere Threads aufzuteilen um eine gute Performanz zu ermöglichen.