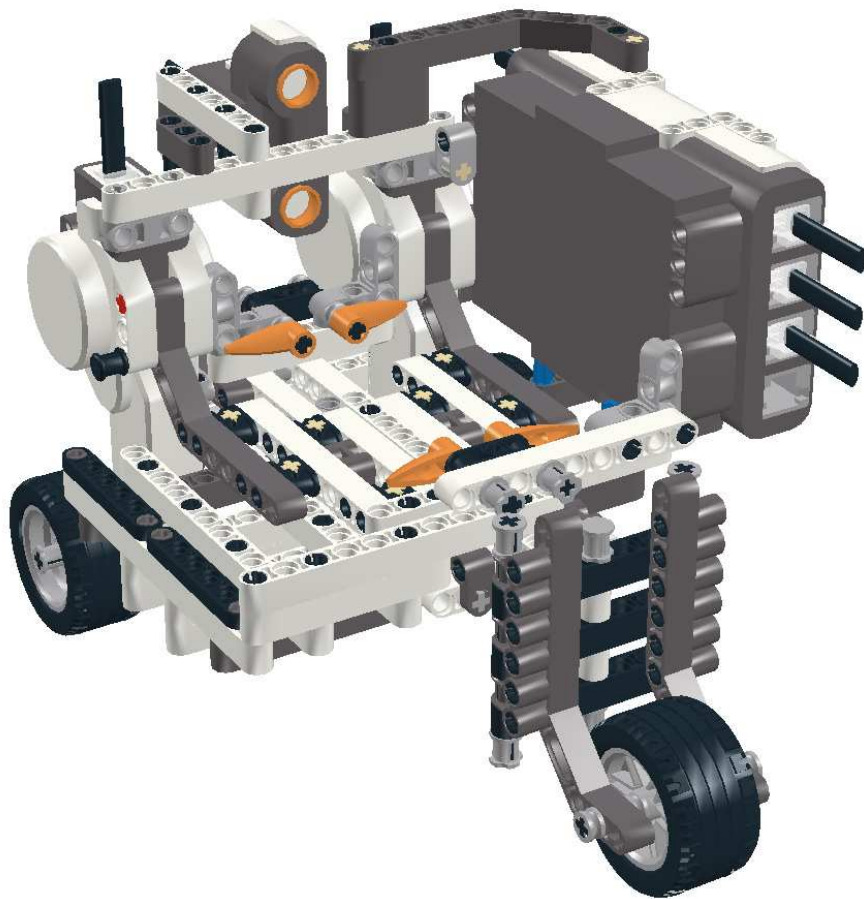


Getränkeautomat

The new generation of barkeeping

Adrian Steiner, Jost Furrer, Manfred Arnold,
Nicolas Stocker, Sandro Arnold, Simon Gisler



Inhaltsverzeichnis

Einleitung	2
Projektbeschreibung	3
Funktionsweise	3
Konstruktionen	3
Programmierung	3
Beschreibung Teilstationen	4
Bestelleinheit	4
Glaslager	5
Mischfahrzeug	6
Schieber oben	7
Lift	8
Schieber unten	9
Fahrzeug	9
Mixer	13
Teamarbeit und Organisation	14
Teamorganisation	14
Teamarbeit	14
Projektplanung	15
Öffentlichkeitsarbeit	15
Schlusswort	16
Erkenntnisse	16
Anhang	17

Einleitung

Sehr geschätzte Leserinnen und Leser. Die aufregendste und interessanteste Zeit der gesamten Berufsmaturitätszeit ist sicherlich die Projektwoche. In der Projektwoche konnten wir zum ersten Mal ein grösseres Projekt selber durchführen. Dass wir ein Legoprojekt machen wollten, war schnell klar. Durch die Teilnahme mit unserem Projekt am Wettbewerb «Robot Team Challenge» kann die geleistete Arbeit doppelt genutzt werden.

Wie Sie schon im Titel unseres Projektes herauslesen können, handelt es sich bei unserem Projekt um einen automatischen Getränkeautomaten. Wie wir auf diese Idee gekommen sind, ist schnell erklärt. Sobald die Pausenglocke in unserem Schulhaus klingelt, bilden sich immer lange, nervtötende und andauernde Warteschlangen vor der Cafeteria. So drängte sich uns der Gedanke auf, das Personal etwas zu entlasten und die Wartezeit zu verkürzen, was natürlich auch in unserem Interesse ist. Wir setzten uns zusammen, brachten unsere Ideen auf Papier und entschieden uns für den besagten Automaten.



Adrian erstellt ein Robolab-Programm...

In der Projektwoche wurden die Konstruktionen erbaut und die Software aus zeitlichen Gründen mit Robolab¹ erstellt. Die Kommunikation über Bluetooth funktioniert bei Robolab nur zwischen zwei NXTs. Somit mussten Informationen über Sensoren von einem NXT zum anderen NXT übergeben werden. Dadurch war der «Glasausstoss» eingeschränkt. Der Getränkeautomat funktionierte trotzdem einwandfrei.

Für die Teilnahme am Wettbewerb wollten wir die einzelnen Konstruktionen noch verfeinern und die Software vollständig auf NXC² umschreiben. Vor allem die Umstellung auf eine textbasierende Programmiersprache war für uns eine grosse Herausforderung.

¹ einfache, graphische Programmiersprache (basiert auf LabView)

² NXC, Abkürzung für Not eXactly C

Projektbeschreibung

Funktionsweise



Unser Getränkeautomat ist in 8 Teilstationen unterteilt. Sie heißen Bestelleinheit (Wahl), Glaslager (Glaslager), Mischfahrzeug (Keeper), Schieber oben (S-Oben), Lift (Lift), Fahrzeug (Wagen), Schieber unten (S-Unten) und Mixer (Mixer). Die Bezeichnungen in den Klammern sind die dazugehörigen Programm-Namen.

Das Getränk wird an der Bestelleinheit ausgewählt. Über Bluetooth übergibt die Bestelleinheit den Startbefehl an das Glaslager und die Getränkeinformation an das Mischfahrzeug weiter. Ein Glas wird dann mit Hilfe eines Schiebers auf das Mischfahrzeug aufgeladen. Je nach gewähltem Getränk fährt das Mischfahrzeug unter die gewünschten Abfüllsysteme. Sind alle Flüssigkeiten eingefüllt, fährt das Fahrzeug zum Lift. Ein weiterer Schieber schiebt das gefüllte Glas in den Lift, der das Getränk nach unten befördert. Unten angekommen, wird es auf das Fahrzeug umgeladen. Dieses transportiert das Glas zum Mixer, der das Getränk kräftig durchrührt. Dann fährt das Fahrzeug zurück zur Bestelleinheit. Dort angekommen, kann die Bestellerin oder der Besteller sein Getränk genießen.

Konstruktionen

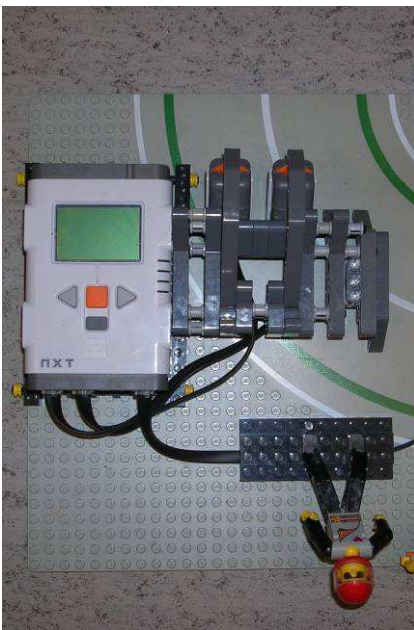
Die Konstruktionen sind uns von Beginn an sehr leicht gefallen, da alle aus unserem Team einen technischen Beruf gelernt haben. Wir können alle auf mindestens 4 Jahre Berufserfahrung zurückgreifen. Schon oft mussten wir technische Probleme innert kürzester Zeit und mit wenigen Hilfsmitteln lösen, was uns bestimmt auch bei dieser Arbeit zugute kam. Als Kinder waren wir alle begeisterte Legobauer...

Programmierung

Mehr Probleme bereitete uns die Programmierung. Nur Adrian Steiner konnte auf seine Programmiererfahrung zurückgreifen. Er kannte jedoch nur Programmiersprachen aus dem SPS-Bereich. Daher kam Simon Gisler noch zusätzlich in die Programmierabteilung. Gemeinsam haben sie sich der Herausforderung gestellt und ein Kommunikationskonzept ausgearbeitet. Das Konzept ist im Anhang aufgeführt. Entstanden ist ein Zusammenwirken von 2 Bluetooth-Gruppen. Die erste Einheit besteht aus der Bestelleinheit, welche als Master agiert. Sie hat 3 Slaves unter sich: Das Glaslager, die Mischstation und den oberen Schieber. Die zweite Einheit hingegen beinhaltet nur 3 NXTs: Den Lift als Master, den unteren Schieber und das Fahrzeug als Slaves. Die Schnittstelle zwischen beiden Gruppen ist ein Lichtsensor beim Lift, der dem oberen Schieber «mitteilt», dass der Lift bereit ist. Somit können wir über die Bluetooth-Verbindungen ein optimales Zusammenwirken der verschiedenen Stationen erreichen.

Beschreibung Teilstationen

Bestelleinheit



Funktion

Der Kunde hat die Möglichkeit aus mehreren vorgegebenen Kombinationen ein Getränk auszuwählen. Die Getränkenummer ist auf dem Display ersichtlich. Die Auswahl eines Getränkes wird mit Schalter 1 gesteuert. Sobald das gewünschte Getränk im Display erscheint, wird die Bestellung mit dem Schalter 2 ausgeführt. Über Bluetooth wird das Glaslager gestartet und die Getränkeinformation an das Mischfahrzeug gesendet. Sobald der obere Glasschieber meldet, dass sich das Glas im Lift befindet, sendet die Bestelleinheit den Rückfahrbefehl an das Mischfahrzeug.

Konstruktion

Die Bestelleinheit besteht aus zwei Berührungssensoren und einem NXT-Baustein. Die Sensoren sind zur einfacheren Betätigung schräg angebracht.

Software

Die einfache Erweiterung eines neuen Getränkes war die Hauptanforderung an die Software. Mit Hilfe eines Zustandsdiagramms wurden die verschiedenen Zustände aufgezeichnet und danach mit NXC programmiert. Die Umsetzung in die Programmiersprache war dank der Case-Anweisung sehr einfach. Hier zeigten sich bereits am Anfang die Vorteile einer textbasierenden Sprache gegenüber von Robolab. Das Zustandsdiagramm und eine mögliche Ausbauparallel sind im Anhang aufgeführt.

Schwierigkeiten

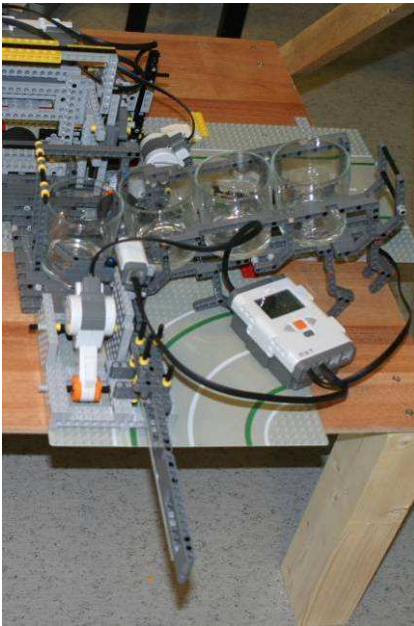
Das Prellen der Schalter musste abgefangen werden. Ohne Wartezeit ratterte die Auswahl im Display unkontrollierbar durch.

Die Kommunikation über Bluetooth bereitete am Anfang grosse Probleme. Leider können die Slaves nur über den Master miteinander kommunizieren. Konkret bedeutete dies, dass der obere Glasschieber über Bluetooth der Bestelleinheit meldet, dass er das Glas in den Lift geschoben hat. Nun kann die Bestelleinheit der Mischstation den Rückfahrbefehl erteilen.

Das Senden über Bluetooth vom Master zu einem Slave war kein Problem. Doch die Kommunikation von einem Slave zum Master war eine Knacknuss. Beim Austesten funktionierte es zwischen einem Slave und einem Master problemlos. Sobald jedoch nur eine zusätzliche Bluetooth Verbindung zu einem zweiten NXT aufgebaut wurde, funktionierte die Software nicht mehr. Das Problem lag beim Kommunikationsprinzip zwischen einem Slave und einem Master: Ein Slave schreibt eine Antwort einfach in seine Ausgangs-Mailbox, die mit der Eingangs-Mailbox des Masters übereinstimmen muss. Die Information bleibt also beim Slave und wird nicht gesendet! Erst wenn der Master ein «MessageRead» zum Slave sendet, erhält der Master die gewünschte Antwort des Slaves.

Bei der Bestelleinheit wurden in einer ersten Variante zweimal kurz nacheinander die Befehle SendRemoteBool (an Keeper und Schieber) aufgerufen. Beim zweiten Aufruf stürzte die Software immer ab. Lange war uns die Ursache nicht klar, da praktisch die identische Software bei der Liftsteuerung keine Probleme verursachte. In der Zwischenzeit ist uns die Ursache bekannt. Durch den zweiten Aufruf wurde die vorgängige Bluetooth-Schreib-Anforderung unterbrochen, was einen Absturz verursachte. Eine sichere Bluetooth-Schreib-Anforderung kann durch die Abfrage des Bluetooth-Status erreicht werden. Es wird solange gewartet, bis BluetoothStatus ein NO_ERR zurückgibt. Diese Status-Abfrage haben wir nun generell vor allen SendRemote-Befehlen eingefügt.

Glaslager



Funktion

Im Glaslager sind leere Gläser hintereinander aufgereiht. Durch einen Schieber wird das erste Glas auf das Mischfahrzeug geschoben. Sobald der Schieber wieder eingefahren ist, wird mit einem Förderband das nächste Glas bereitgestellt.

Konstruktion

Das Glaslager sollte einfach und mit möglichst wenigen Bausteinen konstruiert werden. Eine reibungslose Funktion stand jedoch im Vordergrund. Zu Beginn waren wir uns nicht ganz einig, wie wir das Problem mit dem Nachrutschen des nächsten Glases lösen sollten. Eine Möglichkeit war einen Gummizug einzusetzen, um so von hinten eine Kraft auf die Gläser zu geben. Wir erkannten jedoch, dass die Kraft nicht genug gross ist, wenn nur noch wenige Gläser im Lager vorhanden sind. Deshalb entschlossen wir uns, das Lager schräg angewinkelt zu bauen. Wenn nun das vorderste Glas weggeschoben wird, rutscht automatisch, dank der Schwerkraft, das nächste nach. Diese Lösung funktionierte zuerst einwandfrei. Mit der Zeit ergaben sich aber Probleme (klebrige Rutschfläche, etc.). Als Schlusslösung wurde ein Förderband eingebaut. Es funktioniert, ist aber sicher keine optimale Lösung. Doch die Platzverhältnisse waren in der Zwischenzeit vorgegeben. Eine optimalere Lösung hätte weitere Anpassungsarbeiten erfordert. Der Schieber besitzt an der unteren Kante eine Zahnschiene. Das vom Motor angetriebene Zahnrad kann nun den Schieber nach vorne und hinten bewegen und das Glas somit zur gewünschten Zeit in das Mischfahrzeug befördern.

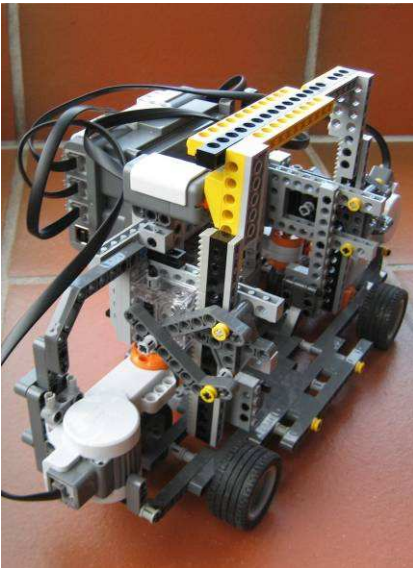
Software

Sobald das Glaslager die gesendete Drinknummer von der Bestelleinheit empfängt, befördert der Schieber ein Glas in die Mischstation. Danach fährt der Schieber wieder fast ganz ein. Das letzte Stück des Schiebers wird langsamer und damit sanft bis zum Endschalter zurückgezogen. Dann wird der Motor des Förderbandes kurz eingeschaltet, damit ein neues Glas nachrutschen kann. Das Flussdiagramm ist im Anhang aufgeführt.

Schwierigkeiten

Die Probleme mit dem Nachschieben der Gläser wurden oben unter *Konstruktion* aufgeführt. Eine mögliche realisierbare Lösung wäre ein zusätzlicher Glasschieber, der die Gläser von links nachschiebt.

Mischfahrzeug

*Funktion*

Das Mischfahrzeug ist verantwortlich für das Abfüllen der verschiedenen Getränke. Die Getränkeinformation wird über Bluetooth von der Bestellstation zum Mischfahrzeug übertragen. Sobald ein Glas aus dem Glaslager auf dem Mischfahrzeug ist, startet das Fahrzeug. Es fährt innerhalb einer Führung zu den verschiedenen Getränkeflaschen. Unter den für das Getränk benötigten Flaschen hält es an und fährt die Querstange über dem gelagerten Glas nach oben aus. Die Querstange betätigt ein Verschlussystem unter der Flasche, worauf die Flüssigkeit durch das Schläuchchen ins Glas fließt.

Konstruktion

Zu Beginn der Arbeit standen wir vor einer grossen Herausforderung. Wie ist es möglich mit Lego einfach und genau die richtige Getränkemenge in ein Glas abzufüllen? Wir suchten diverse Ideen für einen geeigneten Mechanismus. Der Mechanismus sollte eine einfache Bauweise besitzen und vom Mischfahrzeug betätigt werden können. Ansonsten hätte man bei jeder Flasche einen Motor benötigt und diesen jeweils separat ansteuern müssen. Die erste Idee war ein spezieller Flaschenverschluss. Wir machten ein Loch in den Deckel und dichteten es mit einer Kugel ab, welche mit einer Feder verbunden war. Das Mischfahrzeug hätte nun mit der ausfahrbaren Querstange die Feder zusammenpressen und die Kugel dadurch leicht angehoben und so den Getränkefluss ermöglicht. Die Flüssigkeit wäre somit ins Glas geflossen. Jedoch war dieses Verfahren zu aufwändig in der Konstruktion und hygienisch nicht das, was wir suchten. Wir entschlossen uns dann, diesen Teil mit Schläuchen zu realisieren. Die Flaschen mussten nun nicht mehr direkt über der Abfüllstation platziert werden. In den Schraubverschluss der Flaschen wurde ein Loch gebohrt. Ein weiteres Loch für die Luftzufuhr wurde in den Flaschenboden gebohrt. Durch dieses Loch wird die Strömungsgeschwindigkeit begrenzt und gleichzeitig das Nachfüllen erleichtert. Durch das Loch im Deckel wird ein Schläuchlein gesteckt, das von der Flasche auf ein Knick-System geführt wird.

Das Mischfahrzeug fährt bei Bedarf die Querstange aus, diese betätigt den Mechanismus, um den Knick im Schläuchlein zu lösen. Die Flüssigkeit strömt anschliessend sanft ins Glas.

Software

Wie bei der Bestelleinheit war auch hier die einfache Erweiterung eines neuen Getränkes die Hauptanforderung an die Software. Wiederum drängte sich der Einsatz einer Case-Anweisung auf. Zusätzlich wurde die Subroutine *Abfahren* geschrieben. Damit kann nun ein zusätzliches Getränk einfach implementiert werden.

Die Subroutine *Abfahren* kann mit Hilfe der Parameter t_1 bis t_5 gesteuert werden. Es stehen theoretisch 5 verschiedene Getränke zum Mixen zur Auswahl. Wenn zum Beispiel der Parameter $t_1 = 0$ ist, wird das erste Getränk nicht benötigt. Ist der Parameter $t > 0$, wird bei der entsprechenden Einfüllstelle gestoppt und für die Zeitdauer t das entsprechende Getränk eingefüllt. Das Flussdiagramm ist im Anhang aufgeführt.

Schwierigkeiten

Bei den verwendeten Schläuchen war das Material entscheidend. Erst als Manfred die Silikon-Schläuche mitbrachte, gelang es, die Schläuche so abzuknicken, dass keine Flüssigkeit mehr nachtropft. Nach einigen Tests wurde der Doppelknick-Mechanismus entwickelt.

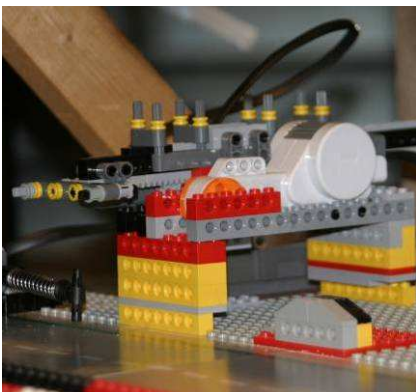
Das Prellen der Schalter musste auch hier abgefangen werden. Doch aufgrund der gemachten Erfahrungen bei der Bestelleinheit, war dieses Problem schnell behoben.

Die Kommunikation über Bluetooth war kein eigentliches Problem. Der Kepper muss nur Befehle von der Bestelleinheit empfangen.

Grosse Probleme bereitete das synchrone Ausfahren der Querstange. Damit genügend Kraft vorhanden ist, wurden zwei Motoren für das Aus- bzw. Einfahren der Querstange eingesetzt. Speziell an der Konstruktion war, dass ein Motor vorwärts und der andere Motor retour fahren musste und dies noch synchron. Mit Robolab war dieses Problem einfach zu lösen. Ein Motor wird als Master eingesetzt und der zweite Motor folgt als Slave dem zweiten Motor mit umgekehrter Richtung. Auch mit NXC gibt es einen entsprechenden Befehl. Erste Versuche verliefen hoffnungsvoll. Doch für den Endausbau wollten wir natürlich die Querstange möglichst schnell ausfahren. Und siehe da, mit kleinen Geschwindigkeiten wurde die Stange synchron bewegt aber bei grossen Geschwindigkeiten war es vorbei mit der Synchronisation. Nach dem Ausprobieren diverser Varianten, wurde ein Versuchsaufbau mit zwei Motoren, die in die gleiche Richtung drehen, ausprobiert. Tatsächlich bewegten sich die Motoren nun auch bei voller Geschwindigkeit synchron. Diese Erkenntnis war beruhigend, weil wieder ein Lösungsweg offen stand. Beim Vorgänger des NXTs konnte die Drehrichtung des Motors durch Umstecken des Anschlusskabels komfortabel geändert werden. Diese Möglichkeit bietet der NXT-Baustein nicht mehr. So musste die Konstruktion mechanisch so abgeändert werden, dass beide Motoren die gleiche Bewegungsrichtung haben. Chris Rogers (Entwickler von Robolab) freute sich natürlich, dass Robolab in diesem Bereich stärker ist als NXC...

Bei längeren Einsätzen muss die Querstange ab und zu wieder ausgerichtet werden. Mit Bricx Command Center können die Motoren direkt angesteuert werden. Damit konnten wir die Querstange bequem ausrichten. Doch komischerweise drehte danach der eine Motor beim ersten Aufruf für kurze Zeit in die falsche Richtung. So war die Ausrichtung natürlich sinnlos. Diesen Effekt haben wir lange nicht bemerkt. Weil dieser Effekt wirklich nur beim ersten Aufruf der Motoren auftrat, konnten wir einen Softwarefehler ausschliessen. Wenn die Software nach der Ausrichtung gestoppt und neu gestartet wird, funktioniert die Ausrichtung einwandfrei!

Schieber oben



Funktion

Der Schieber oben wartet, bis der Keeper (über einen Endschalter) und der Lift (über einen Lichtsensor) bereit sind. Danach wird das Glas in den Lift geschoben. Sobald der Schieber wieder zurück ist, wird dies der Bestelleinheit gemeldet. Somit kann die Bestelleinheit den Keeper zurückrufen. Es kann vorkommen, dass der Lift auf das Fahrzeug warten muss. Deshalb muss der Schieber über einen Lichtsensor die Position des Liftes kennen.

Konstruktion

Insgesamt bauten wir zwei Schieber, einen oben und einen hinten, um die Übergaben zu ermöglichen. Um es uns nicht zu einfach zu machen, haben wir die zwei Schieber unterschiedlich konstruiert. Bei dem längeren, oberen Schieber war die Stabilität ein Problem, er drohte abzuknicken. Mit einer langen Führung konnten wir dieses Problem beheben.

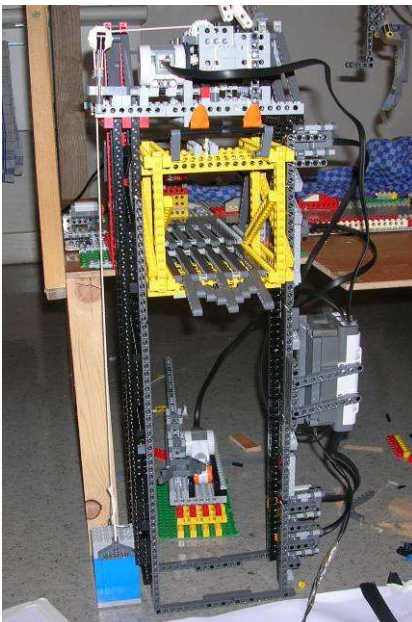
Software

Der obere Schieber wartet bis der Keeper über einen Endschalter und der Lift über einen Lichtsensor bereit sind. Danach wird der Schieber vollständig ausgefahren. Die Ausfahrdistanz wird über den Drehwinkel des eingebauten Rotationssensors eingestellt. Die Einfahrdistanz ist etwas kürzer gewählt. Dank dieser Massnahme kann der Schieber mit einer kleineren Geschwindigkeit sanft bis zum Endschalter eingezogen werden. Dadurch startet das nächste Ausfahren wieder aus einer definierten Position. Damit der «Getränkeausstoss» möglichst hoch ist, wird nach der verkürzten Einfahrt der Bestelleinheit eine Meldung gesendet. So kann die Bestelleinheit als Master dem Keeper die Bewilligung für die Rückfahrt senden. Sobald der Keeper-Endschalter meldet, dass der Keeper aus der Endposition weggefahren ist, startet der Vorgang wieder von vorne. Das Flussdiagramm ist im Anhang aufgeführt.

Schwierigkeiten

Die Position des Liftes wurde zu Beginn mit einem Original NXT-Berührungssensor erfasst. Der Sensor hat den Lift aber durch Reibung beeinträchtigt. Kurzerhand haben wir einen alten RCX-Berührungssensor über ein Adapterkabel eingebaut. Die alten Schalter sind viel kleiner und benötigen kleinere Schaltwege. Als weiterer Vorteil konnte der obere Lift-Endschalter für den Lift selber und für den oberen Schieber genutzt werden. Doch leider rutscht der Lift nach dem Motorstopp wieder ganz wenig nach unten. Damit war ab und zu das Bereitschaftssignal für den oberen Schieber nicht korrekt. So haben wir das Bereitschaftssignal berührungslos über einen Lichtsensor erfasst. Beim Starten der Software muss der Lift oben sein. Damit kann der Referenzwert für den Lichtsensor gespeichert werden.

Lift



Funktion

Der Lift überwindet die Höhe von der Mischstation zum Fahrzeug. Das Glas wird vom oberen Schieber in den Lift geschoben. Unten wird das Glas über den zweiten Glasschieber, über eine bewegliche Rampe, auf das Fahrzeug geschoben.

Konstruktion

Es war nicht einfach, die erwünschte Stabilität zu erreichen. Schon im Vorfeld entschieden wir uns für eine Fachwerkkonstruktion. Wir wollten die ganze Kabine von oben bis unten führen. Dies konnte wegen Mangel an Legosteinen leider nicht realisiert werden. Die Kabine wird nun beidseitig von einer Legoschiene, die durch eine Fachwerkkonstruktion stabilisiert wird, geführt. Um die Kräfte und Drehmomente ins Gleichgewicht zu bringen, verankerten wir die Konstruktion mittels Auslegern nach hinten auf den Tisch, damit das Glas problemlos in den Lift rein- und rausgeschoben werden kann. Die Kabine hängt an einem Seil, welches oben über eine Rolle auf die Seite geleitet wird und wieder nach unten zieht. Am Ende des Kabels hängt ein Gegengewicht, welches die entstehenden Gewichtskräfte ausgleicht. Der Antrieb erfolgt über eine angetriebene Welle, die das Seil aufwickelt. Damit genügend Reibung entsteht, haben wir die Welle mit einem rauen Klebeband umwickelt.

Software

Der Lift ist der zweite Master in unserem Projekt. Sobald der Ultraschallsensor ein Glas erkannt hat, wird der Lift zum Zwischenhalt gefahren. Dieser Zwischenhalt ist notwendig, weil zuerst kontrolliert werden muss, ob

das Fahrzeug bereits auf den Lift wartet. Die Kontrolle wird über Bluetooth realisiert. Wenn das Fahrzeug bereit ist, fährt der Lift mit kleinerer Geschwindigkeit bis zum unteren Endschalter. Über Bluetooth wird der untere Glasschieber aktiviert. Danach fährt der Lift nach oben bis zum oberen Endschalter. Kurz vor dem Endschalter (über Rotationssensor) wird die Geschwindigkeit des Liftmotors verkleinert, damit der Lift sanft bis zum Endschalter hochgezogen wird. Das Flussdiagramm ist im Anhang aufgeführt.

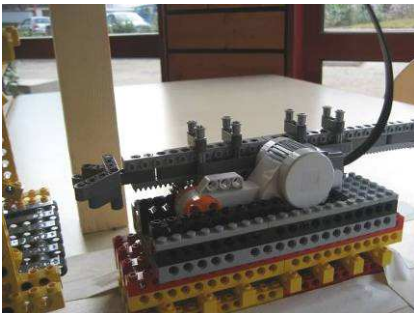
Schwierigkeiten

Zu Beginn der Testphase verwickelte sich das Seil oft auf der Welle. Dies konnte jedoch durch eine präzisere Führung recht schnell behoben werden. Die Original NXT-Berührungssensoren wurden zuerst auf der Seite des Liftschachtes angebracht. Diese Sensoren haben den Lifttransport durch Reibung beeinträchtigt. Durch den Ersatz mit alten RCX-Sensoren und das Anbringen unterhalb bzw. oberhalb des Liftes konnte dieses Problem beseitigt werden.

Dank den gemachten Bluetooth-Erfahrungen bei der Bestelleinheit, hatten wir bei der Kommunikation keine Probleme.

Weil die Ultraschallmessung nicht immer präzise funktioniert, müssen zwei korrekte Messungen nacheinander erfolgen, damit die Funktion «AufGlas-Warten» «wahr» zurückliefert.

Schieber unten



Funktion

Der Schieber unten wartet auf das Startsignal vom Lift. Sobald das Signal eintrifft, wird das Glas vom Lift auf das Fahrzeug geschoben.

Konstruktion

Siehe unter Schieber oben!

Software

Der untere Schieber wartet auf das Startsignal des Liftes, welches über Bluetooth übertragen wird. Das Aus- bzw. das Einfahren des unteren Schiebers werden analog zum oberen Schieber realisiert. Das Flussdiagramm ist im Anhang aufgeführt.

Schwierigkeiten

Es sind keine Schwierigkeiten aufgetreten.

Fahrzeug

Funktion

Das Fahrzeug transportiert das Getränk zur Bestellerin bzw. zum Besteller. Auf diesem Weg wird es bei einem Zwischenstopp nochmals kräftig von einem Mixer durchgerührt. Die Strecke, welche zurückgelegt werden muss, ist mit einer schwarzen Linie, die Stoppstellen durch einen schwarzen Querstrich, definiert.

Konstruktion

Aller guten Dinge sind drei vier! Wir haben vier Anläufe benötigt, bis wir uns auf die aktuelle Konstruktion einigen konnten.



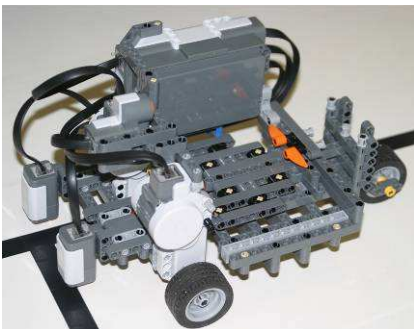
So sah das Fahrzeug nach dem 3. Versuch aus...

1. Versuch

Zuerst konstruierten wir die Hinterachse des Fahrzeuges und bauten das Differentialgetriebe ein. Dazu gehörte ein Antriebsmotor, welcher vor der Achse Platz fand. Dieser Teil zog keine grossen Probleme mit sich, also bauten wir den zweiten Motor, welcher die Lenkung übernehmen sollte, direkt vor den Antriebsmotor. Beide Motoren wurden quer über die Breite des Wagens verbaut. Das Schwierigste war die Aufhängung der vorderen Achse. Die Bedingungen waren: Die Achse muss lenkbar sein und eine gute Stabilität aufweisen. Beides gestaltete sich schwierig und war hart zu knacken, doch kam eine taugliche Konstruktion zustande. Sie war nicht nur stabil, sondern sah auch noch gut aus. Jetzt zeigte sich noch einmal ein nervenaufreibendes Problem. Die Lenkstange sollte mit dem Antrieb verbunden werden. Die Lösung mit Zahnrädern war nicht realisierbar, weil der Platz fehlte und sich das Spiel zwischen den Zahnrädern addierte. Schlussendlich war es eine Viertelumdrehung zu gross und machte etwa die Hälfte der Lenkbewegung aus. Also entschieden wir uns für einen Kettenantrieb. Das Spiel wurde dadurch wesentlich verringert, weil Zahnräder eingespart wurden. Auch diese Variante nutzte das Platzangebot vollständig aus. Weil die Bauzeit viel Zeit in Anspruch nahm, war das Programm bereits vom Programmierer bereitgestellt und wir konnten sofort mit den ersten Tests loslegen. Schnell stellte sich heraus, dass die Lenkung den Zweck schlichtweg nicht erfüllt. Der Kettenantrieb, welcher auf zwei Zahnrädern lief, verklemmt sich mit diesen. Die Ursachen waren eine zu lange Kette, welche schon auf die minimale Länge gekürzt wurde und die Zahnräder, die von der Grösse an der unteren Grenze lagen, sodass die Kettenglieder sich nicht schön darauf abrollen konnten. Ärgerlicherweise musste der ganze vordere Teil neu konstruiert werden.

2. Versuch

Die Suche nach der Lösung für das Lenkproblem ging von vorne los. Schnell war aber klar, dass die vordere Aufhängung aus nur einem Rad bestehen wird, welches mit einem Motor angesteuert und in die gewünschte Richtung gelenkt werden kann. Nach Abschluss der Konstruktion war wieder eine Testphase an der Reihe. Der Erfolg liess aber immer noch auf sich warten. Es war jedoch ersichtlich, dass sich das Prinzip bewähren wird. Wegen einigen Konstruktionsfehlern, Mangel an Stabilität und der Fehlplatzierung der Lichtsensoren musste noch einmal ein Neuaufbau gestartet werden. Die Konstruktion ging somit in eine dritte Phase.



Fahrzeug nach dem 4. Versuch.
Aller guten Dinge sind eben vier!

3. Versuch

Mit der nötigen Erfahrung und dem Wissen, wie es nicht funktioniert, sollte es nun mit dem dritten Anlauf klappen. Aller guten Dinge sind drei! Nach Fertigstellung der mechanischen Arbeit wurde die Testphase gestartet und nach kleinen Veränderungen erfolgreich abgeschlossen. Jetzt verbauten wir noch die Ladefläche für das Transportgut und die NXT wurde noch fest integriert. Anschliessend war die Konstruktion des Fahrzeuges abgeschlossen und es wurden keine grossen Veränderungen mehr vorgenommen. Es wurde noch die Beleuchtung für das Getränk eingebaut, welche den Coolnessfaktor noch um zwei Sterne erhöht...

4. Versuch

Bisher musste die Anlage nur einige wenige Getränke mixen. Als erste Bewährungsprobe stand die LAP-Ausstellung vor der Türe. Während mehrerer Stunden musste sich die Anlage im Dauerbetrieb bewähren. Schnell einmal wurde klar, dass das Fahrzeug immer noch Probleme bereitete. Die Kräfte auf die Konstruktion waren so gross, dass die Legoverbin-

dungen immer wieder gelöst wurden. Als erste Massnahme wurde die Konstruktion verstärkt. Danach wurde die Anfahrsgeschwindigkeit stufenweise erhöht und später der Antrieb noch umgebaut. Damit konnten die zwei Tage LAP-Ausstellung einigermaßen überbrückt werden. Doch für einen reibungslosen Betrieb musste ein neues Fahrzeug her.

Die Erfahrungen der vorgängigen Konstruktionen waren nicht umsonst. Alle Abmessungen und die meisten Tücken waren nun bekannt. Bei der neuen Konstruktion verzichteten wir generell auf die Noppen-Verbindungen. Beim Ultraschallsensor war klar, dass wir ihn um 90 Grad abgewinkelt einbauen wollen. Die Steuerung erfolgt über zwei Frontmotoren, das Linienfahren wird mit einem Lichtsensor (weiss-hellgrau-dunkel) und das Stoppen mit einem zweiten Lichtsensor erledigt. Die Konstruktion wurde mit dem Lego Designer erfasst und ist im Anhang angefügt.

Software

Natürlich bedingten die mechanischen Anpassungen oft auch eine Softwareänderung. Die häufigen Anpassungen haben dafür gesorgt, dass das Fahrzeug schlussendlich eine ausgeklügelte Software besitzt. Automatische Anpassung der Lichtsensoren bei jedem neuen Getränk, Stromsparmodus (Lichtsensoren bei Stopps ausgeschaltet), Glaserkennung berührungslos über Ultraschall und ein PD-Regler (Quelle: <http://spurt.uni-rostock.de/lego.pdf>) zum Linienfolgen sind hier die Schlagworte!

Beim alten Fahrzeug mussten vor dem Start die Lichtsensoren manuell geeicht werden. Dazu mussten die Sensoren abwechselnd auf die helle bzw. dunkle Eichfläche gehalten werden. Mittels der orangenen NXT-Taste wurde der entsprechende Sensorwert abgespeichert. Durch den Langzeittest bei der LAP-Ausstellung wurde klar, dass Lichtänderungen nur mit einer Software, die sich vor jeder Fahrt automatisch neu eicht, in den Griff zu bekommen sind. Deshalb werden die beiden Lichtsensoren automatisch bei voller Geschwindigkeit geeicht (Prinzip, siehe links). Dieser Vorgang wiederholt sich bei jedem Durchgang. Damit können langsame Lichtschwankungen abgefangen werden. Danach fährt das Fahrzeug zum Lift. Hier wird gewartet bis das Glas per Ultraschall erkannt wird. Die Erfahrungen bei der Ultraschallmessung beim Lift kommen auch hier zum Einsatz. Es müssen wieder zwei korrekte Messungen nacheinander erfolgen.

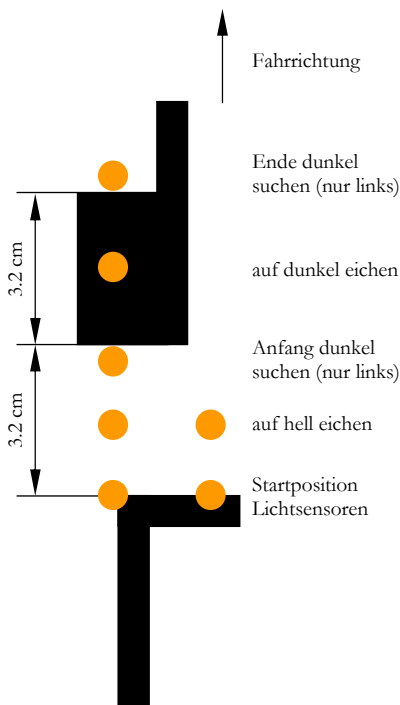
Danach transportiert das Fahrzeug das Glas zum Mixer und zurück zur Bestellereinheit. Dort wird gewartet bis per Ultraschall erkannt wird, dass das Glas entfernt wurde. Nach dem erneuten Eichen fährt das Fahrzeug zurück zum Lift und der Vorgang beginnt von vorne.

Das Linienfolgen wird mit einem Sensor realisiert. Beim Eichen werden die Werte für «hell» und «dunkel» erfasst. Der Sollwert wird aus dem Mittelwert $(\text{hell} + \text{dunkel})/2$ berechnet. Das Fahrzeug fährt sozusagen auf «grau», bzw. am Rand der Linie. Mit Hilfe eines PD-Reglers folgt das Fahrzeug präzise dem Linienrand. Das Grundprinzip ist im Anhang erklärt.

Der zweite Lichtsensor zählt die Querstriche und somit kennt die Software die genaue Position des Fahrzeugs. Das Fahrzeug kann nicht gut an ein Netzgerät angeschlossen werden. Deshalb werden die Lichtsensoren bei Nichtgebrauch ausgeschaltet, damit wird die Einsatzdauer des Akkus wesentlich verlängert. Der Ultraschallsensor wird direkt von der Akkuspannung ohne Stabilisierung versorgt. Deshalb wird pro Durchgang noch zusätzlich die Akkuspannung überprüft. Bei weniger als 7.2 Volt wird das Fahrzeug gestoppt. Der NXT hat zwar bereits eine Akkuüberwachung eingebaut, diese setzt aber bei einer tieferen Spannung ein. Das Zustandsdiagramm bzw. die Flussdiagramme sind im Anhang aufgeführt.

Schwierigkeiten

Die Probleme bei der Konstruktion und mit der abgewinkelten Ultraschallmessung sind bereits oben erwähnt.



Die neue Software funktionierte bei den ersten Tests einwandfrei. Doch plötzlich zeigte sich beim Eichen ein Fehler. Zuerst dachten wir uns nichts dabei, denn beim nächsten Neustart funktionierte das Fahrzeug wieder einwandfrei. Doch sobald mehrere Gläser gemixt wurden, tauchte der Fehler beim Eichen wieder auf. Mit der Zeit bemerkten wir, dass der Eichfehler systematisch nach fünf gemixten Gläsern (also nach ca. 5 Minuten...) auftrat. Unsere Vermutung war nun, dass das Zurückstellen des Rotationssensors nicht korrekt verlief. Nachdem die Software so angepasst wurde, dass die Werte des Rotationssensors auf dem Display erschienen, war die Fehlerursache klar. Tatsächlich war der Anfangswert nach dem Reset nicht auf Null zurückgestellt.

Ursprüngliche Variante

```
sub eichen(int &hell,int &dunkel,int &referenz)
{
  int anfang,int v=100;

  SetSensorLight(IN_2);
  SetSensorLight(IN_3);
  ResetRotationCount(OUT_A);
  anfang=MotorRotationCount(OUT_A);      // startet nicht bei Null
  OnFwdSync(OUT_AC,v,0);
  until((MotorRotationCount(OUT_A)-anfang)>45);
  "
```

Versuchsvariante

```
sub eichen(int &hell,int &dunkel,int &referenz)
{
  int anfang,int v=100;

  ResetRotationCount(OUT_A);
  SetSensorLight(IN_2);
  SetSensorLight(IN_3);
  anfang=MotorRotationCount(OUT_A);      // startet korrekt bei Null
  OnFwdSync(OUT_AC,v,0);
  until((MotorRotationCount(OUT_A)-anfang)>45);
  "
```

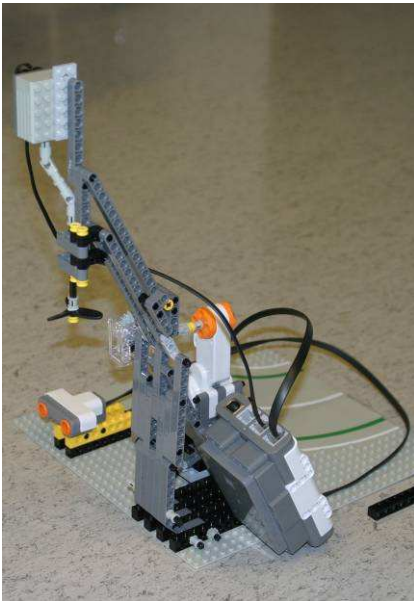
Eine E-Mail-Nachfrage bei John C. Hansen, dem Entwickler von NXC, bestätigte das Problem. So konnten wir auch dieses Problem lösen.

Schlusslösung

```
sub eichen(int &hell,int &dunkel,int &referenz)
{
  int anfang,int v=100;
  bool wiederholen=true;

  while(wiederholen)
  {
    ResetRotationCount(OUT_A);
    Wait(5);
    anfang=MotorRotationCount(OUT_A);
    if (anfang==0)
    {
      wiederholen=false;
    }
  }
  SetSensorLight(IN_2);
  SetSensorLight(IN_3);
  OnFwdSync(OUT_AC,v,0);
  until((MotorRotationCount(OUT_A)-anfang)>45);
  "
```

Mixer



Funktion

Der Mixer mixt das Getränk kräftig durch.

Konstruktion

Beim ersten Fahrzeug bereitete die Positionierung des Mixers Probleme. Der Halt des Fahrzeuges war infolge der Steuerung mit relativ grossen Toleranzen verbunden. Weil das neue Fahrzeug dank der Regelung sehr präzise und mit viel kleineren Toleranzen der Linie folgt, bereitet die Positionierung des Mixers keine Probleme mehr.

Beim Ultraschallsensor war klar, dass wir ihn um 90 Grad abgewinkelt einbauen müssen.

Software

Sobald das Glas über Ultraschall erkannt wird, startet der Mixer. Bei der Distanzmessung konnten wir auf die bewährte Lösung beim Lift bzw. beim Fahrzeug zurückgreifen. Es müssen zwei korrekte Messungen nacheinander erfolgen, damit die Funktion «AufGlasWarten» «wahr» zurückliefert. Das Flussdiagramm ist im Anhang aufgeführt.

Schwierigkeiten

Es sind keine Schwierigkeiten aufgetreten.

Teamarbeit und Organisation

Teamorganisation



hinten von links: Simon, Jost, Adrian
vorne von links: Manfred, Sandro, Nicolas

Adrian

Adrian hatte schon einige Erfahrungen mit ähnlichen Programmiersprachen, aber auch er musste sich zuerst in die Sprache einarbeiten, um die teils komplizierten Programme zu schreiben.

Jost

Jost steckte einen grossen Teil seiner Zeit in die Öffentlichkeitsarbeit und verbrachte Stunden über Stunden am PC. Er setzte sich intensiv mit dem Word auseinander und hat der ganzen Arbeit ein schönes Design verliehen.

Manfred

Manfred entwickelte das ausgeklügelte Dosiersystem für die Getränkebefüllung und half auch beim Erstellen der Website mit. Der gelernte Elektromonteur bewies bei diesem Projekt auch seine Fähigkeiten als Tischler, denn er zimmerte die Flaschenaufhängung.

Nicolas

Nicolas verbrachte viel Zeit mit der Konstruktion vom Lift und Schieber. Er leistete auch einen grossen Beitrag zur Website.

Sandro

Sandro konstruierte die beiden Fahrzeuge, die Führungsschiene sowie das Glaslager. Er konnte wichtige Erfahrungen aus seinem Beruf einbringen.

Simon

Simon setzte sich mit der für ihn neuen Programmiersprache NXC auseinander und konnte sich gut einarbeiten. Er arbeitete erfolgreich und alles funktionierte schlussendlich reibungslos.

Teamarbeit

Die Teamarbeit war sicher eine unserer Stärken. Wichtige Entscheidungen oder die Lösung eines Problems wurden oft gemeinsam getroffen. Durch das gemeinsame Besprechen kamen verschiedene Meinungen zusammen, dabei wurden meistens optimale Lösungen gefunden. Gerade bei schwierigen Problemen war der Teamgeist die antreibende Kraft. Das Wissen, dass andere mit Rat und Tat mithelfen, beflügelt ungemein. Es müssen nicht alle programmieren, doch mitdenken bei Problemen können alle, die Umsetzung kann dann wieder getrost dem Spezialisten überlassen werden. Dieser Austausch untereinander war sicherlich eine wertvolle Erfahrung.

Projektplanung



Gute Planung gibt Sicherheit...

Öffentlichkeitsarbeit



Schlusspräsentation anlässlich der Projektwoche 08

Wenn eine Arbeit zu erledigen war, wurde gehandelt und nicht lange diskutiert. Diese Handlungsfähigkeit ist sicher darauf zurückzuführen, dass alle Teammitglieder bereits eine Lehre erfolgreich absolviert haben.

Für die Projektwoche Ende Februar mussten wir ein Grob- und ein Feinkonzept erstellen. Das Grobkonzept half die anstehenden Arbeiten abzuschätzen. Was muss in der Vorbereitungsphase noch getan werden? Reicht die Zeit überhaupt, um das Projekt zu realisieren? Wo werden Probleme auftreten? Schon bald wurde klar, dass die eigentliche Projektwoche nur ein Teil der gesamten Arbeit ist.

Beim Feinkonzept wurden die Aufgaben und Arbeiten auf die Teammitglieder verteilt und der Tagesablauf bestimmt. Ziele wurden festgelegt und mögliche Gründe, die zum Scheitern des Projekts führen könnten, überdacht. Diese Konzepte wurden für die Beurteilung unserer Arbeit von den beteiligten Fachlehrpersonen geprüft.

Uns war klar, dass wir für die Teilnahme am Wettbewerb «Robot Team Challenge», das Projekt stark überarbeiten müssen. Ein grosses Problem war die zur Verfügung stehende Schulzeit. Im Gegensatz zur Projektwoche konnte unser Informatiklehrer nur 16 Lektionen für die Überarbeitung zur Verfügung stellen. Weiter standen Mitte Juni die Berufsmaturitätsprüfungen und die Präsentation des Getränkeautomaten an der LAP-Feier vor der Türe. Durch das Zusammenlegen der 16 Lektionen auf vier Freitagnachmittage konnte ein sinnvolles Arbeiten ermöglicht werden. Aus diesen Freitagnachmittagen wurden aber oft Freitagabende...

Nach der Berufsmaturität trennen sich unsere gemeinsamen Wege. Die meisten Teammitglieder arbeiten bis zum Schulanfang der Fachhochschulen. Somit war die Organisation der restlichen Arbeiten ein mühsames Unterfangen. Gemeinsame Termine wurden mit Hilfe von Doodle gefunden. Insgesamt wurden ca. 450 Stunden (inklusive Projektwoche) investiert.

Momentan ist noch nicht ganz klar, welche Teammitglieder für die Präsentation am 29. August zur Verfügung stehen. Doch auch dieses Problem werden wir sicher noch lösen...

Die Öffentlichkeitsarbeit ist ein sehr interessanter Teil dieses Wettbewerbs. Einerseits verfassten wir eine schriftliche Arbeit über unser Projekt und kreierten eine Website (<http://www.fraengg.ch/tbm6>). Per Mundzumundpropaganda verbreiteten wir die Information über die Teilnahme an diesem Wettbewerb und verteilten die Websiteadresse mit Visitenkarten. An der LAP-Ausstellung der Berufsfachschule Uri präsentierten wir unseren Getränkeautomate. Mehrere hundert Personen konnten unser Projekt live anschauen und ein Getränk geniessen. An zwei Abenden haben wir ungefähr 300 Getränke mit dem Automat gemixt. Zusätzlich engagierten wir einen Pressefachmann, der einen Zeitungsbericht verfasste, welcher im Urner Wochenblatt erschien. Als Highlight ist sicher die Einladung von Vance Carter (Lego Education Schweiz) für die Worlddidac 2008 Ende Oktober zu werten! Hier eine Zusammenstellung unserer Tätigkeiten:

- Zeitungsartikel über Projektwoche, Urner Zeitung, 29. Februar 2008
- Zeitungsartikel über Wettbewerb, Urner Wochenblatt, 21. Juni 2008
- Website <http://www.fraengg.ch/tbm6>
- Film auf Youtube <http://www.youtube.com/watch?v=qfqplo6BZ4I>
- Präsentation an der LAP-Ausstellung, 19. und 20. Juni 2008
- Präsentation an der Worlddidac in Basel, 29. bis 31. Oktober 2008

Erkenntnisse



...diese Kästen müssen verbaut werden!



Wer sortiert die Kästen am Schluss wieder?

Schlusswort

Ein solch umfangreiches Projekt ist nie fertig. Aus der intensiven Testphase bei der LAP-Ausstellung sind zahlreiche Verbesserungen hervorgegangen. Doch leider bringen scheinbar kleine Änderungen wieder grosse Folgearbeiten mit sich. So wurden aus gedachten Minuten schnell einmal einige Stunden. Der Aufwand hat sich aber gelohnt, der Automat funktioniert zuverlässig und über längere Zeit wartungsfrei.

Die anfänglich geplante Erweiterung mit einem Programmiermodus ist nicht aus dem Papierstadium herausgekommen. Aus zeitlichen Gründen konnte diese tolle Idee nicht umgesetzt werden. Die Grundidee ist jedoch im Anhang als Zustandsdiagramm aufgeführt.

Trotzdem sind wir stolz auf unseren Getränkeautomaten. Wir können schöne Konstruktionsdetails und elegante Softwarelösungen vorweisen.

In den vergangenen Wochen haben wir viele schöne Erlebnisse und Bereicherungen erfahren können. Sei es bei technischen Problemen, welche es zu beseitigen galt oder aber auch in zwischenmenschlichen Belangen, wie der Förderung des Teamgeistes. Unser Team harmonisierte perfekt!

Wir werden uns sicher noch oft an diese Zeit erinnern!

Anhang

Prozessanalyse

- Kommunikationskonzept
- Zustandsdiagramm: Wahl
- Zustandsdiagramm: Wahl mit Programmiermodus (mögl. Erweiterung)
- Flussdiagramm: Glaslager
- Flussdiagramm: Keeper
- Flussdiagramm: Schieber oben
- Flussdiagramm: Lift
- Flussdiagramm: Schieber unten
- Zustandsdiagramm und Flussdiagramme: Wagen
- Flussdiagramm: Mixer
- Funktionsweise Linienfolger

Softwarecodes

- Wahl
- Glaslager
- Keeper
- S-Oben
- Lift
- S-Unten
- Wagen
- Mixer

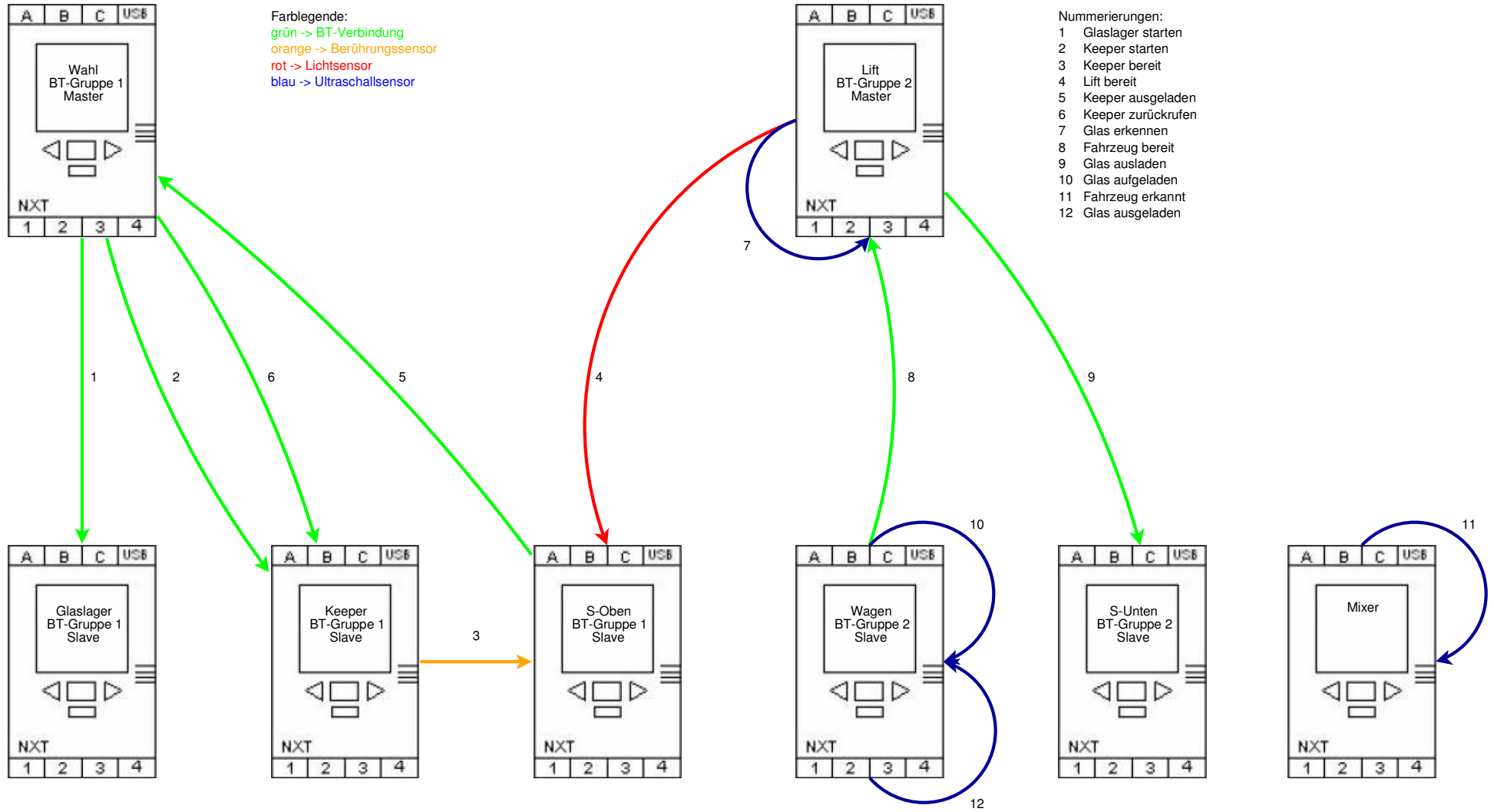
Dokumente Öffentlichkeitsarbeit

- Zeitungsartikel über Projektwoche, Urner Zeitung, 29. Februar 2008
- Zeitungsartikel über Wettbewerb, Urner Wochenblatt, 21. Juni 2008
- Einladung zur Worlddidac (E-Mail von Sarah Carter)
- Bilder Projektwoche <http://bsuri.educanet2.ch/projektwochen2/bm2008>
- Website <http://www.fraengg.ch/tbm6>

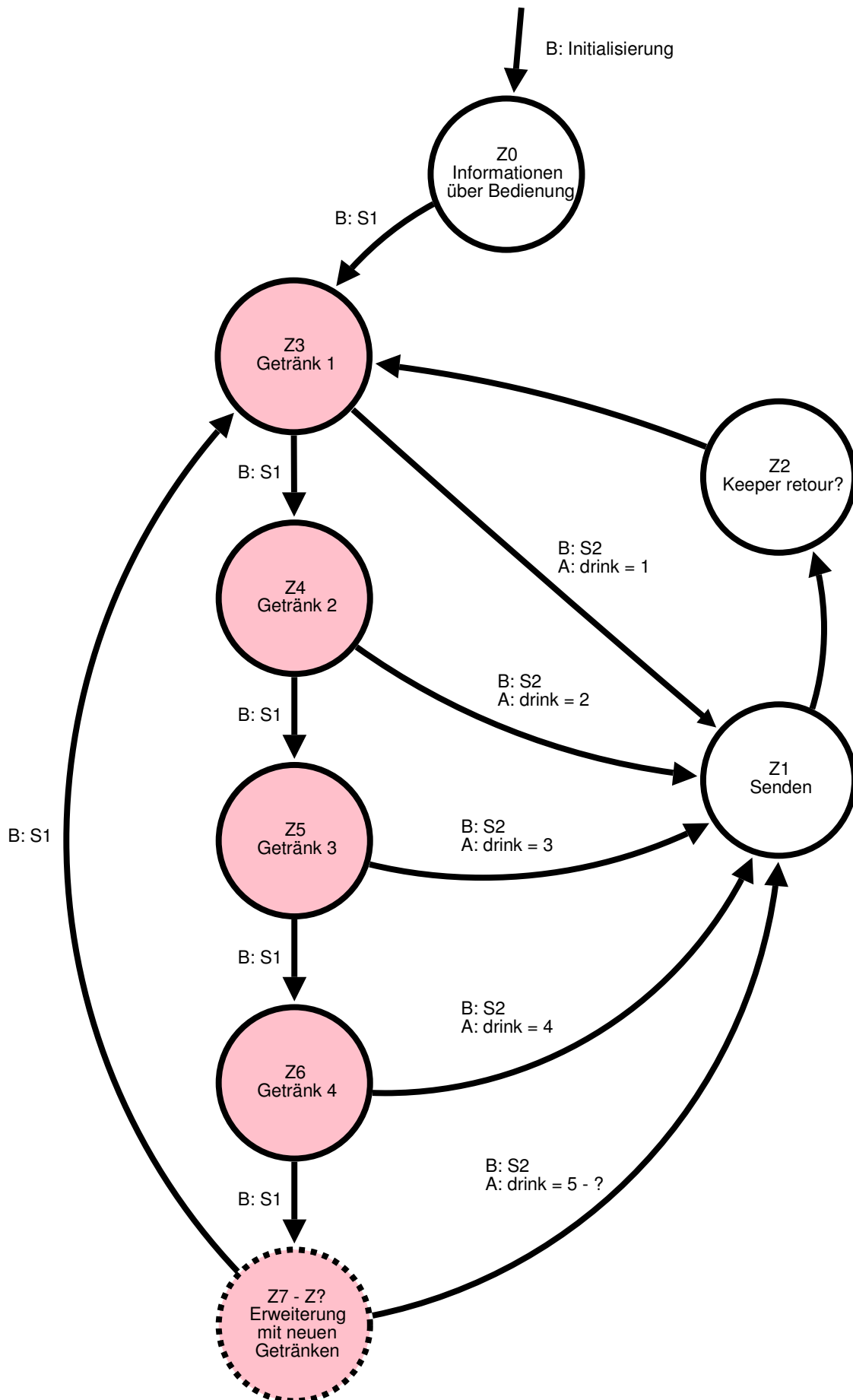
CAD-Daten

- Bauanleitung Wagen

Kommunikationskonzept



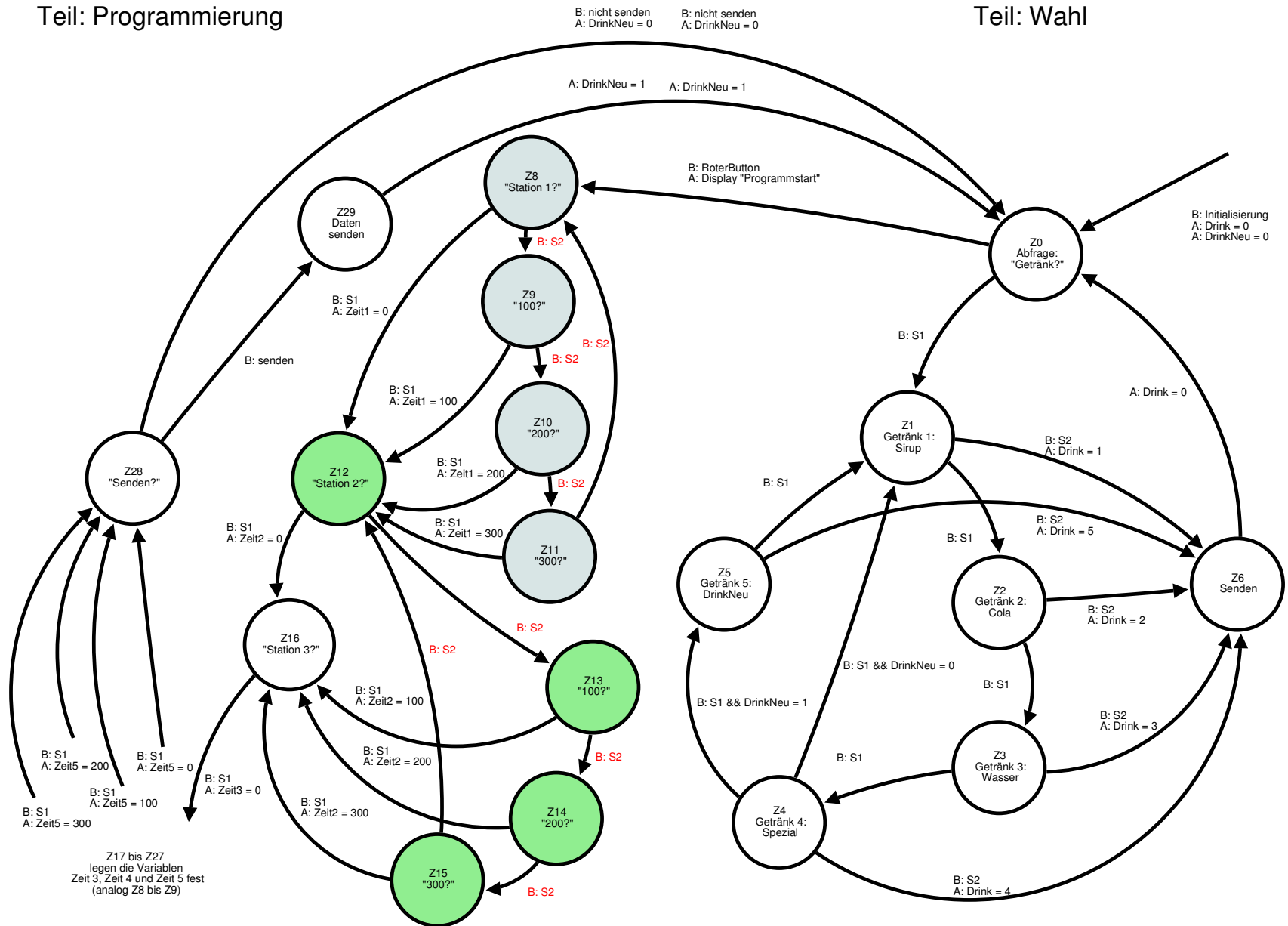
Zustandsdiagramm: Wahl (BT-Gruppe 1)



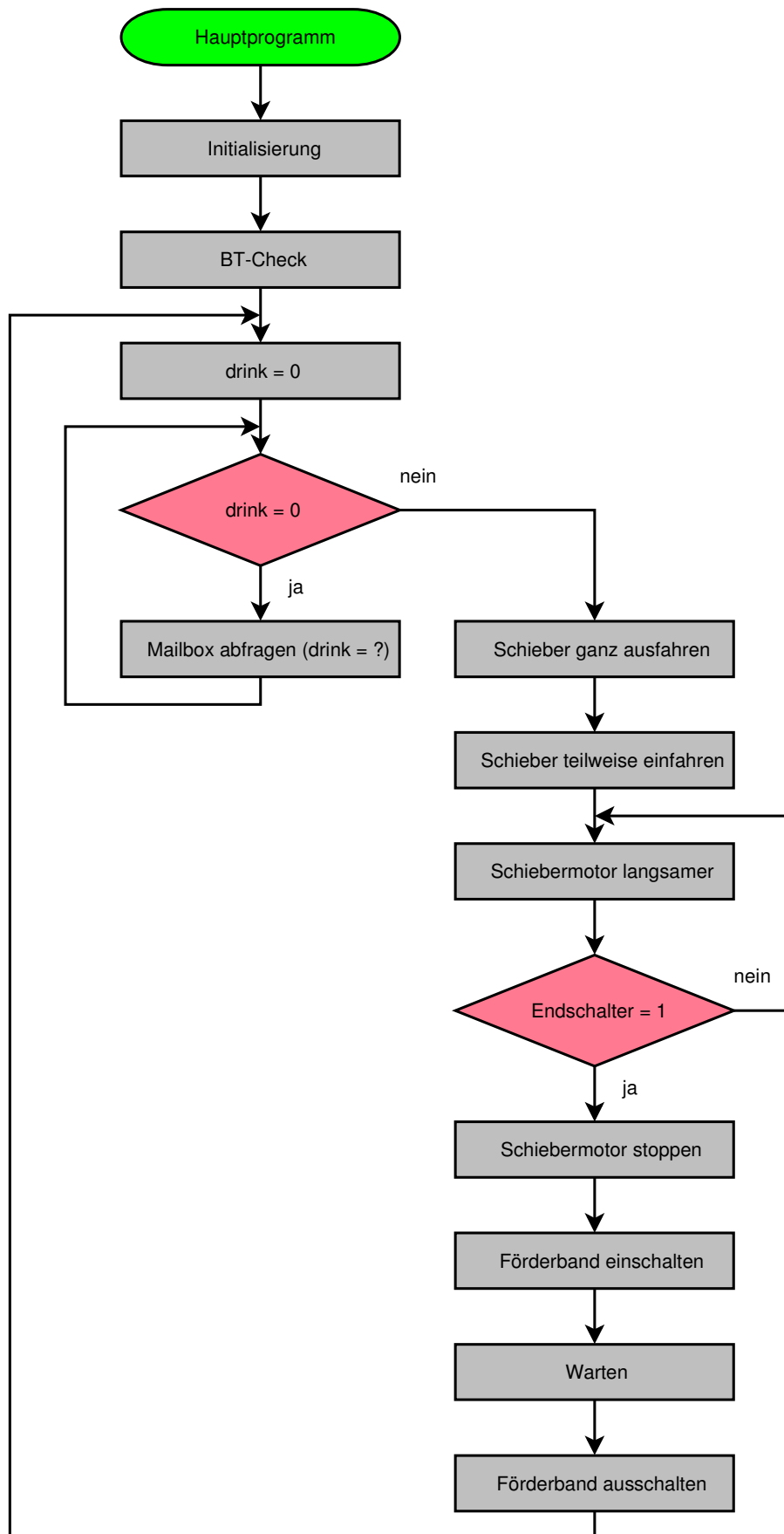
Zustandsdiagramm: Wahl mit Programmiermodus

Teil: Programmierung

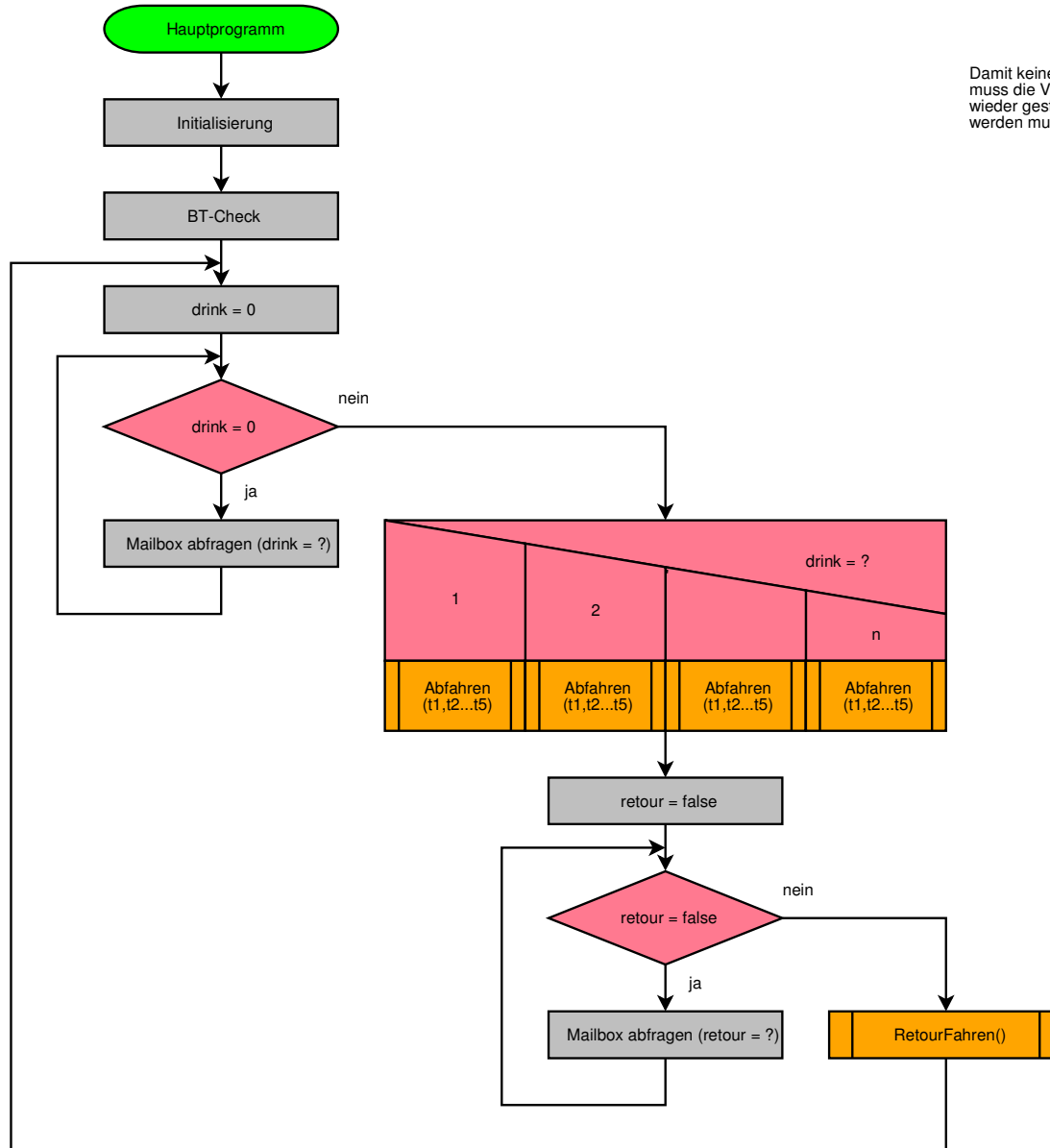
Teil: Wahl



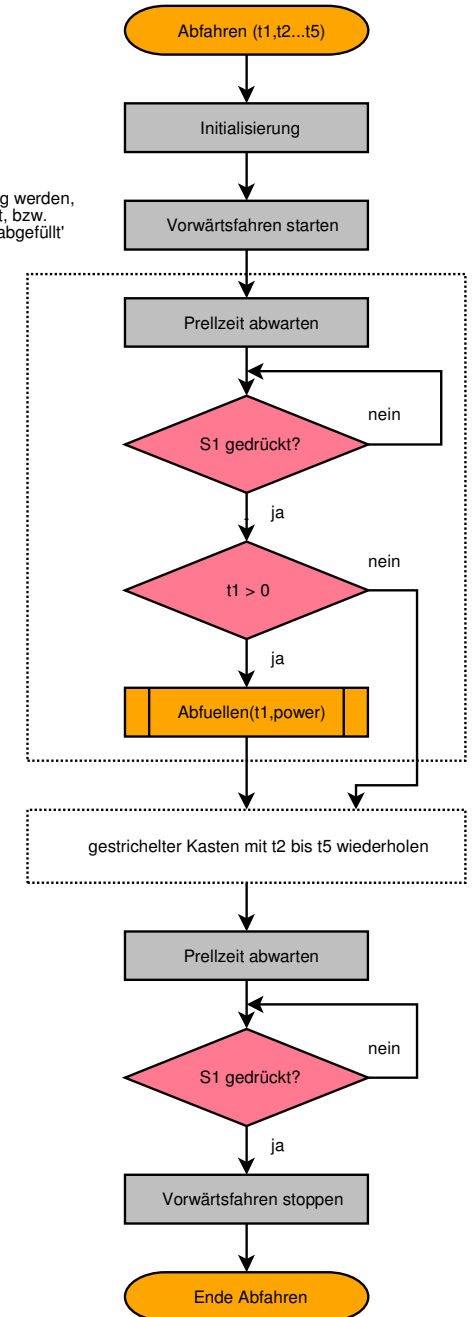
Flussdiagramm: Glaslager (BT-Gruppe 1)



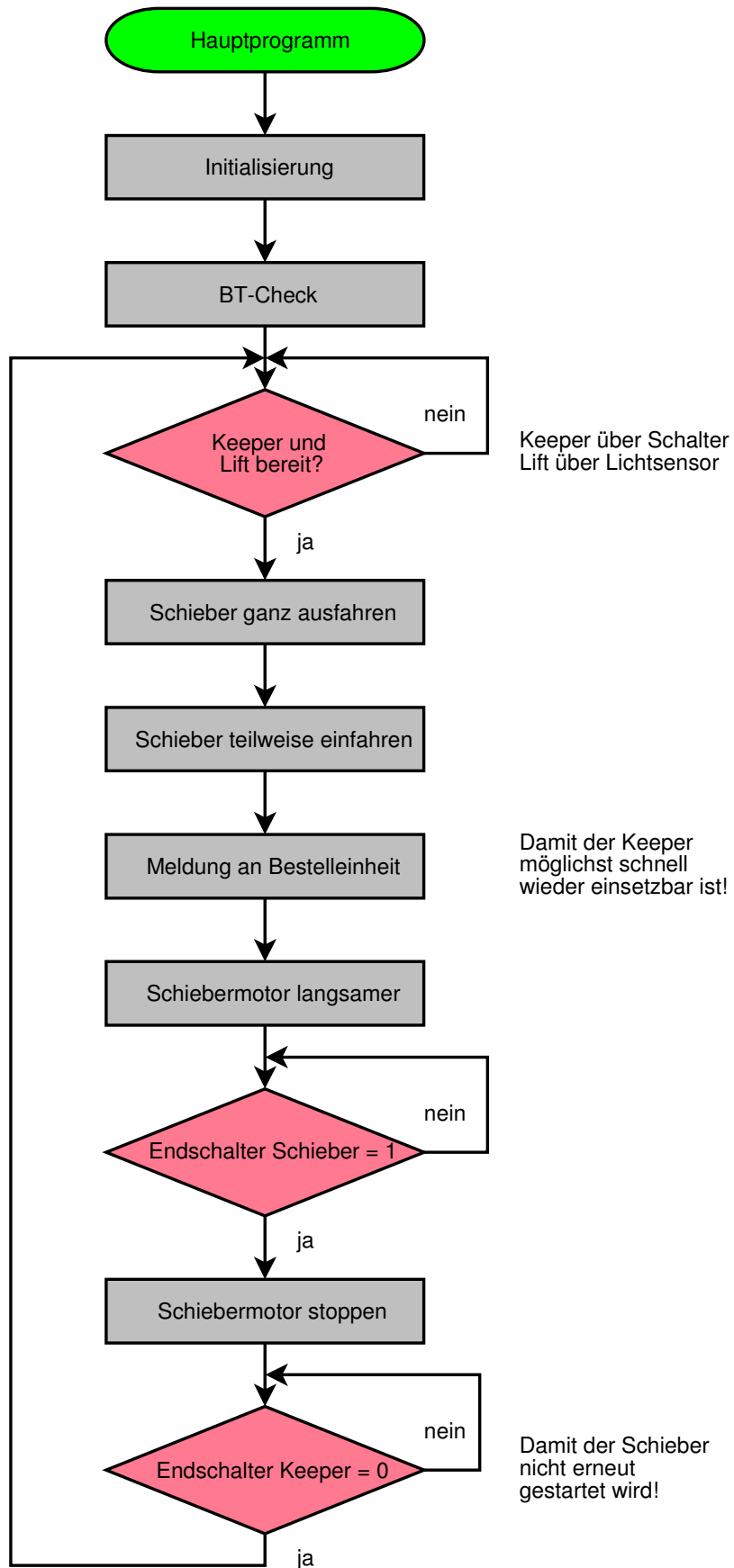
Flussdiagramm: Keeper (BT-Gruppe 1)



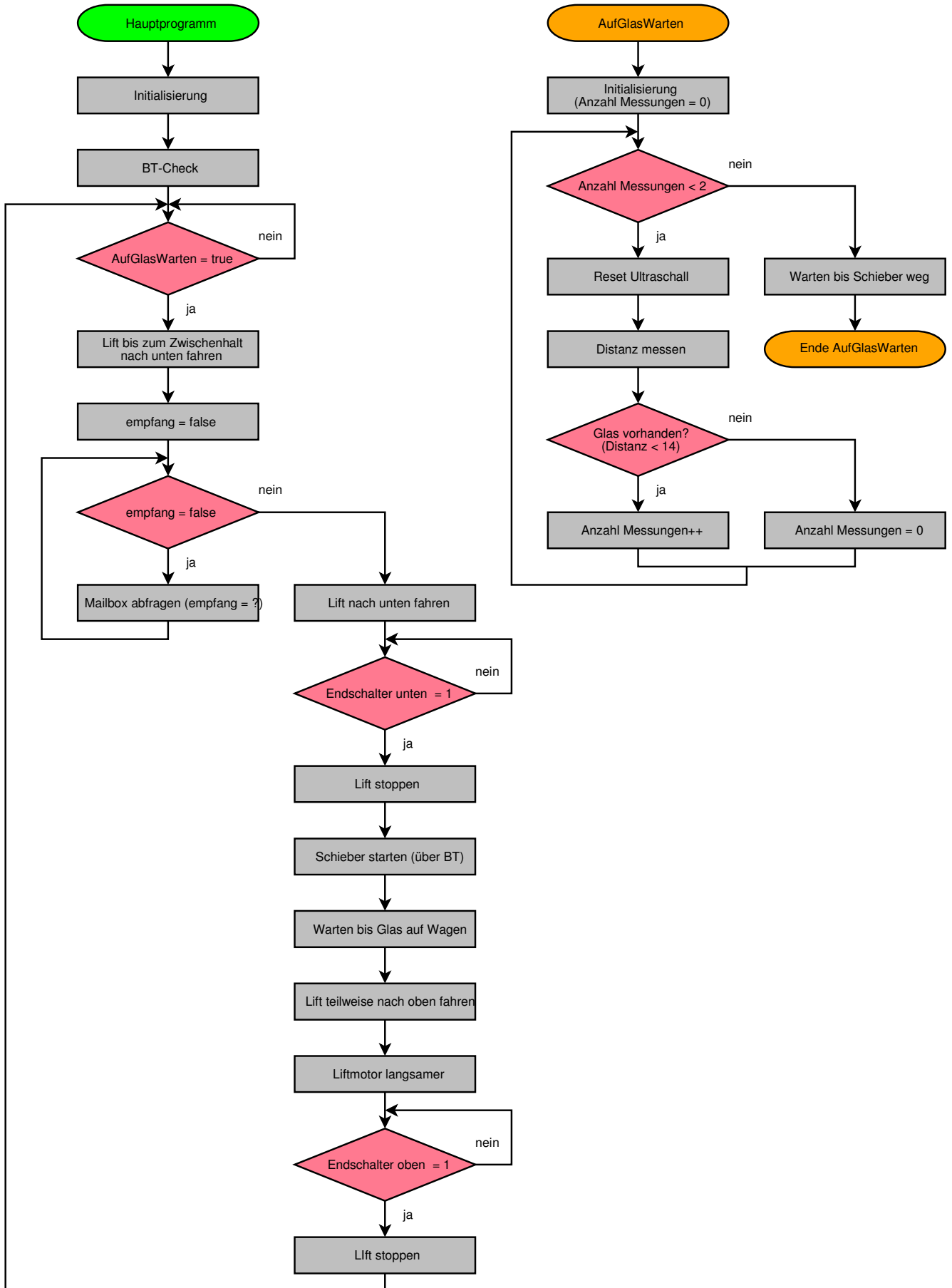
Damit keine unschönen Halts bei t = 0 notwendig werden, muss die Vorwärtsfahrt unter Abfuellen gestoppt, bzw. wieder gestartet werden. Nur wenn tatsächlich 'abgefüllt' werden muss, ist auch ein Halt notwendig!



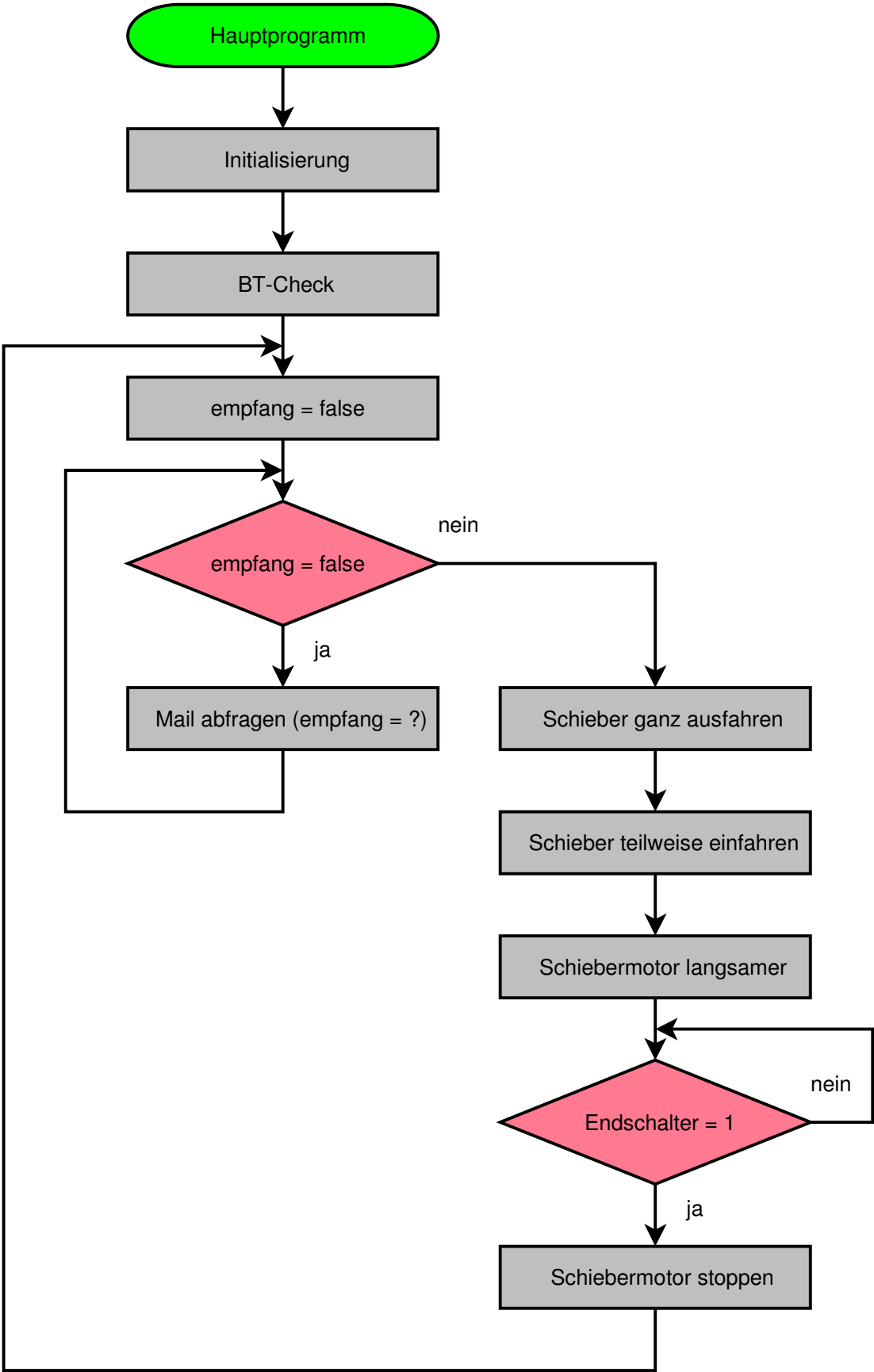
Flussdiagramm: S-Oben (BT-Gruppe 1)



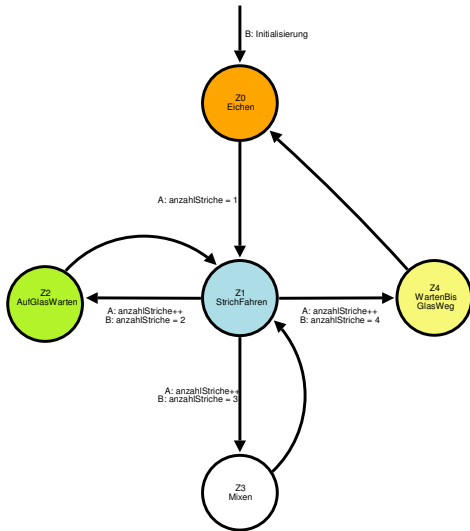
Flussdiagramm: Lift (BT-Gruppe 2)



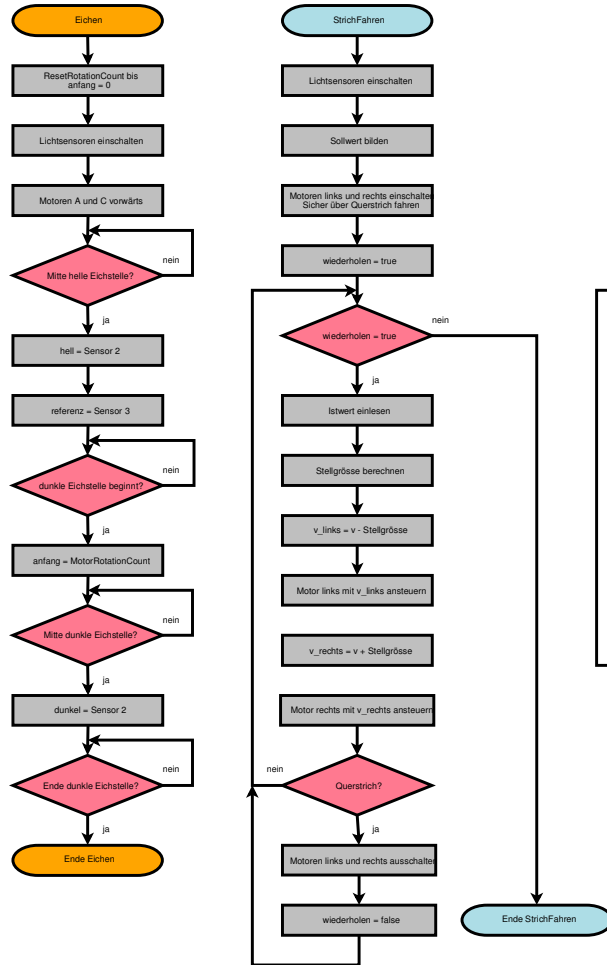
Flussdiagramm: S-Unten (BT-Gruppe 2)



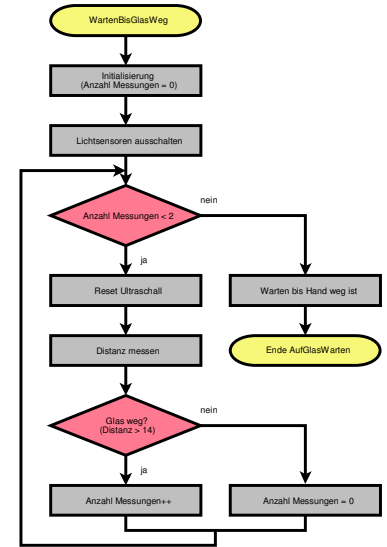
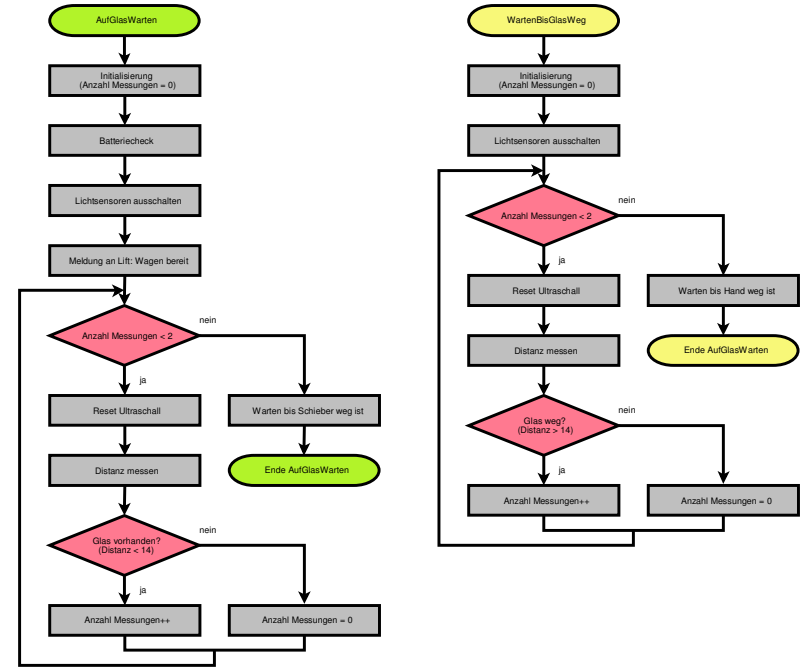
Zustandsdiagramm: Wagen (BT-Gruppe 2)



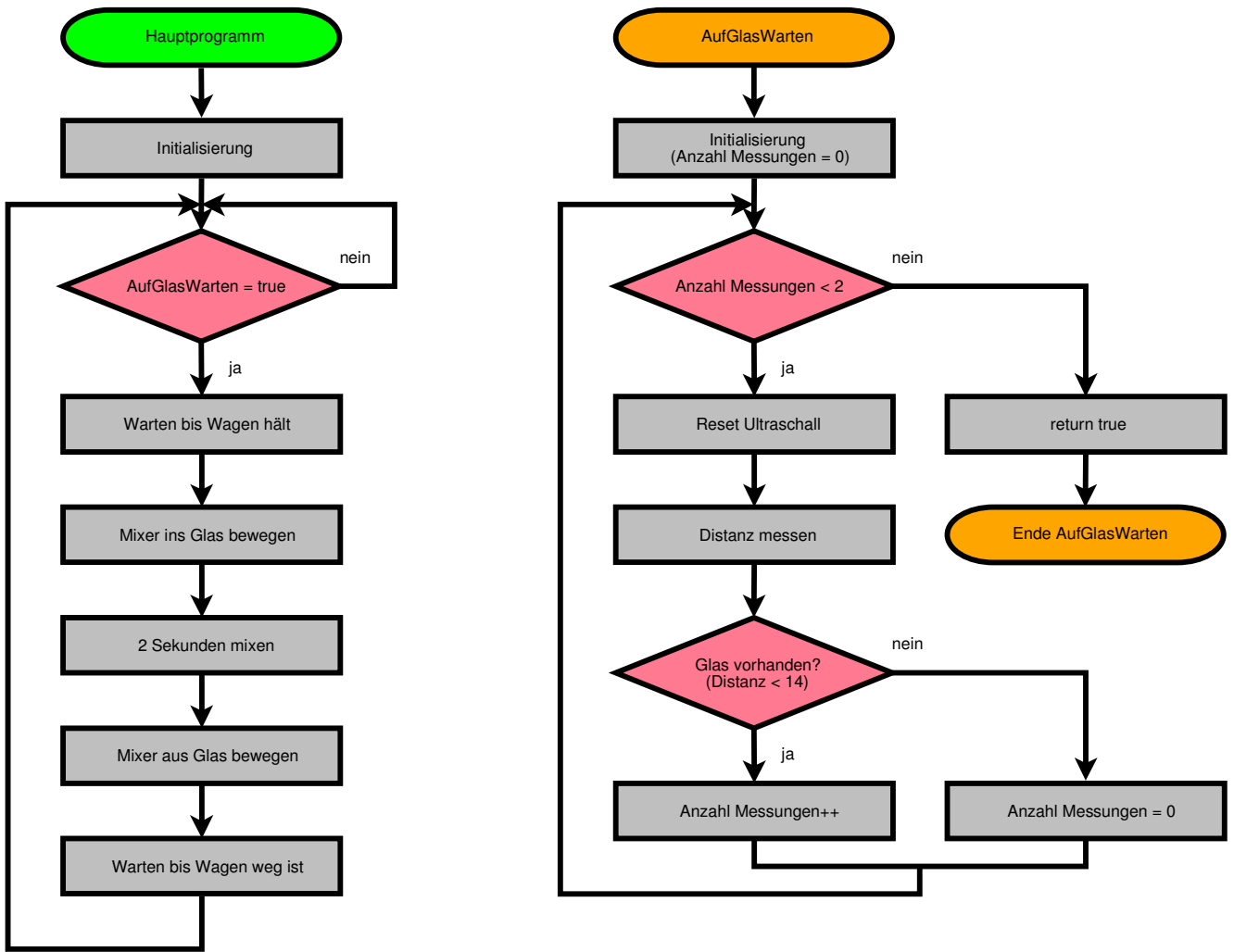
Flussdiagramme: Wagen (Teil 1)



Flussdiagramme: Wagen (Teil 2)



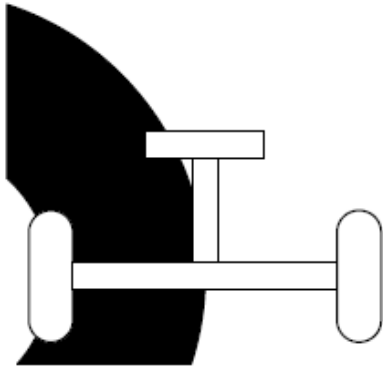
Flussdiagramm: Mixer



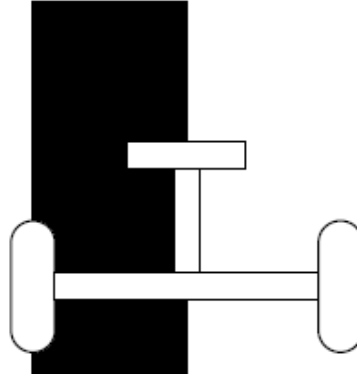
Funktionsweise Linienfolger

Grundprinzip: fährt auf «Grau»

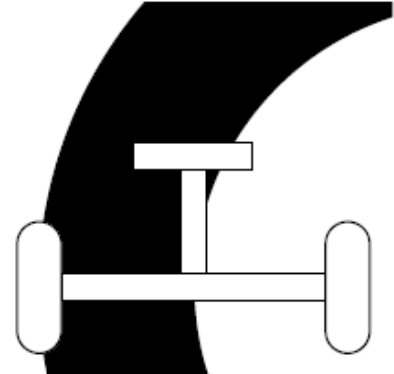
linkes Rad langsamer
rechtes Rad schneller



beide Räder gleich schnell



linkes Rad schneller
rechtes Rad langsamer



Einfluss P-Anteil

Die Abweichung (**Sollwert – Istwert**) wird proportional verstärkt. Wenn eine grosse Abweichung vorliegt, wird stark korrigiert. Ist die Abweichung klein, wird nur wenig korrigiert.

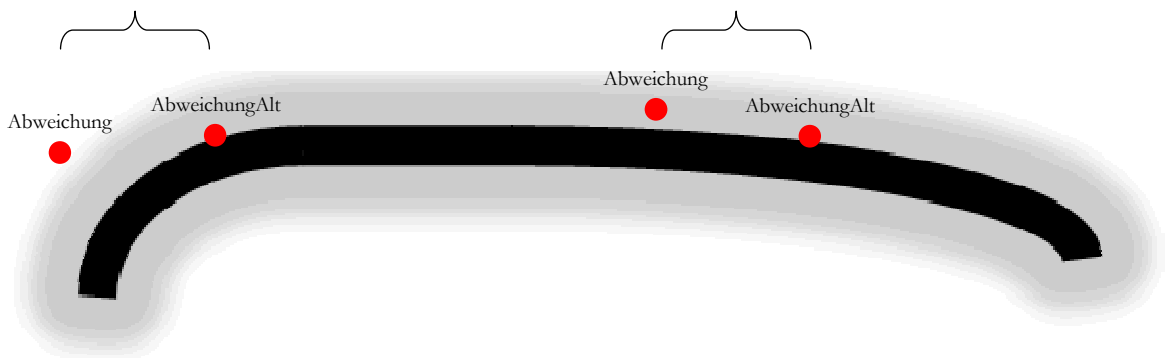
Das Fahrzeug kann bereits der Spur folgen. Die Bewegung ist aber noch ruckartig, weil es erst gebremst wird, wenn die Abweichung zu gross ist.

Einfluss D-Anteil

Der D-Anteil macht den Regler schnell und lässt ihn vorausschauend arbeiten. Eine kurzzeitige grosse Abweichungsänderung (**Abweichung – AbweichungAlt**) wird mit einem Faktor K_d verstärkt und korrigiert sofort das Fahrzeug.

grosse Abweichungsänderung
(grosser Lichtunterschied)

kleine Abweichungsänderung
(kleiner Lichtunterschied)



```

/*
Programmname:      Wahl.nxc
Programmiersprache: NXC
Brick Type:       NXT
Funktion:          Bestelleinheit, dank der 'case-Struktur' problemlos erweiterbar.
                  Die Getränkeummer ist auf dem Display ersichtlich. Die Auswahl
                  eines Getränkes wird mit Schalter 1 gesteuert. Sobald das ge-
                  wünschte Getränk im Display erscheint, wird die Bestellung mit dem
                  Schalter 2 ausgeführt. Über Bluetooth wird das Glaslager gestartet
                  und die Getränkeinformation an den Keeper gesendet. Sobald der
                  obere Glasschieber meldet, dass sich das Glas im Lift befindet,
                  sendet die Bestelleinheit den Rückfahrbefehl an den Keeper.

Bluetooth-Gruppe: 1 (Bestelleinheit -> Master, Glaslager, Keeper, S-Oben -> Slaves)
Autor:            Adrian Steiner / Simon Gisler
Erstellt:        27. März 2008
Geändert:        27. Juni 2008
Hinweis:         keine Sonderzeichen verwenden (ä, ö, ü, etc.)
*/

//-----

sub BTCheck(int conn)
{
  if (!BluetoothStatus(conn)==NO_ERR)
  {
    TextOut(0,LCD_LINE1,"BT-Fehler",true);
    PlayTone(5000,1000); // Warnton
    Wait(4000);
    Stop(true);
  }
}

//-----

int DrinkWahl(int drink)
{
  /*
  Setzt die Zustandsvariable abhängig vom gewählten Getränk und
  vom Zustand Drinkwahl bzw. Bestellung (S1 bzw. S2 gedrückt).
  Bei Erweiterungen muss MAXDRINK angepasst werden!
  */

  #define MAXDRINK 12

  int zustand,prellZeit=400;
  string drinkStr;

  Wait(prellZeit);
  drinkStr=NumToStr(drink);
  TextOut(0,LCD_LINE1,StrCat("Drink ",drinkStr," -> S2"),true);
  until(SENSOR_1==1 || SENSOR_2==1);
  if (SENSOR_1==1) // Drinkwahl
  {
    if (drink==MAXDRINK)
    {
      zustand=3; // wieder beim Drink 1 anfangen
    }
    else
    {
      zustand=drink+3;
    }
  }
  else // S2 -> Bestellung
  {
    zustand=1;
  }
  return zustand;
}

//-----

#define BT_CONN_GLASLAGER 1
#define BT_CONN_KEEPER 2
#define BT_CONN_SCHIEBER 3
#define OUTBOX_GLASLAGER 1
#define OUTBOX_KEEPER 1
#define INBOX_SCHIEBER 1

```

```

task main()
{
    int zustand,drink,anzahl=0;
    bool empfang;
    string meldung;

    SetSensor(S1,SENSOR_TOUCH);
    SetSensor(S2,SENSOR_TOUCH);
    zustand=0;
    while (true)
    {
        switch(zustand)
        {
            case 0: // Initialisierung und Informationen
                TextOut(0,LCD_LINE1,"Informationen:");
                TextOut(0,LCD_LINE3,"Drinkwahl -> S1");
                TextOut(0,LCD_LINE4,"Bestellung -> S2");
                TextOut(0,LCD_LINE6,"weiter mit -> S1");
                until(SENSOR_1==1);
                zustand=3;
                break;
            case 1: // Glaslager starten und Getränkewunsch an Keeper senden
                BTCheck(BT_CONN_GLASLAGER);
                BTCheck(BT_CONN_KEEPER);
                BTCheck(BT_CONN_SCHIEBER);
                until (BluetoothStatus(BT_CONN_GLASLAGER)==NO_ERR); // warten bis BT frei ist
                SendRemoteNumber(BT_CONN_GLASLAGER,OUTBOX_GLASLAGER,drink);
                anzahl++;
                meldung=NumToStr(anzahl);
                TextOut(0,LCD_LINE1,"Warten bis",true);
                TextOut(0,LCD_LINE2,"Keeper retour!");
                TextOut(0,LCD_LINE5,StrCat("Anzahl = ",meldung));
                Wait(4000); // warten bis Glas auf Keeper geschoben ist
                until (BluetoothStatus(BT_CONN_KEEPER)==NO_ERR); // warten bis BT frei ist
                SendRemoteNumber(BT_CONN_KEEPER,OUTBOX_KEEPER,drink);
                zustand=2;
                break;
            case 2: // warten -> bis Schieber meldet -> Keeper abgefahren
                empfang=false;
                while (!empfang)
                {
                    ReceiveRemoteBool(INBOX_SCHIEBER,true,empfang); // inkl. 'MessageRead'
                }
                until (BluetoothStatus(BT_CONN_KEEPER)==NO_ERR); // warten bis BT frei ist
                SendRemoteBool(BT_CONN_KEEPER,OUTBOX_KEEPER,true);
                zustand=3;
                Wait(5000); // Zeit, die der Keeper für die Rückfahrt braucht
                break;
            case 3: // Start bei Drink 1, bei Erweiterungen MAXDRINK anpassen!
                drink=1;
                zustand=DrinkWahl(drink);
                break;
            case 4:
                drink=2;
                zustand=DrinkWahl(drink);
                break;
            case 5:
                drink=3;
                zustand=DrinkWahl(drink);
                break;
            case 6:
                drink=4;
                zustand=DrinkWahl(drink);
                break;
            case 7:
                drink=5;
                zustand=DrinkWahl(drink);
                break;
            case 8:
                drink=6;
                zustand=DrinkWahl(drink);
                break;
            case 9:
                drink=7;
                zustand=DrinkWahl(drink);
                break;
            case 10:
                drink=8;

```

```
        zustand=DrinkWahl(drink);
        break;
    case 11:
        drink=9;
        zustand=DrinkWahl(drink);
        break;
    case 12:
        drink=10;
        zustand=DrinkWahl(drink);
        break;
    case 13:
        drink=11;
        zustand=DrinkWahl(drink);
        break;
    case 14:
        drink=12;
        zustand=DrinkWahl(drink);
        break;
    }
}
```



```

/*
  Programmname:      Glaslager.nxc
  Programmiersprache: NXC
  Brick Type:       NXT
  Funktion:         Das Glaslager schiebt ein Glas auf den Barkepper.
                   Sobald der Schieber wieder eingefahren ist, wird mit einem
                   Förderband das nächste Glas bereitgestellt.

  Bluetooth-Gruppe: 1 (Bestelleinheit -> Master, Glaslager, Keeper, S-Oben -> Slaves)
  Autor:            Adrian Steiner / Simon Gisler
  Erstellt:        27. März 2008
  Geändert:        30. Juli 2008
  Hinweis:         keine Sonderzeichen verwenden (ä, ö, ü, etc.)
*/

//-----

sub BTCheck(int conn)
{
  if (!BluetoothStatus(conn)==NO_ERR)
  {
    TextOut(0,LCD_LINE1,"BT-Fehler",true);
    PlayTone(5000,1000); // Warnton
    Wait(4000);
    Stop(true);
  }
}

//-----

#define BT_CONN_BESTELLEINHEIT 0
#define INBOX_BESTELLEINHEIT 1

task main()
{
  int drink,geschwindigkeit=100,distanz=1990;

  SetSensor(S2,SENSOR_TOUCH);
  BTCheck(BT_CONN_BESTELLEINHEIT);
  while(true)
  {
    TextOut(0,LCD_LINE1,"Empfangsbereit",true);
    drink=0;
    while(drink==0)
    {
      ReceiveRemoteNumber(INBOX_BESTELLEINHEIT,true,drink)
    }
    TextOut(0,LCD_LINE1,"Warten...",true);
    RotateMotor(OUT_A,-geschwindigkeit,distanz); // Glas auf Keeper schieben
    RotateMotor(OUT_A,geschwindigkeit,distanz-200); // Schieber zurückziehen
    OnFwd(OUT_A,geschwindigkeit-40); // Schieber langsam zurückziehen
    until(SENSOR_2==1); // Endschalter -> Ausgangsposition
    Off(OUT_A);
    OnFwd(OUT_B,geschwindigkeit-70); // Glastransport auf dem Förderband
    Wait(1000);
    Off(OUT_B);
  }
}

```

```

/*
  Programmname:      Keeper.nxc
  Programmiersprache: NXC
  Brick Type:       NXT
  Funktion:         Abfüllfahrzeug, dank der 'case-Struktur' problemlos erweiterbar!
                   Es fährt zu den verschiedenen Getränkeflaschen. Unter den für das
                   Getränk benötigten Flaschen hält es an und fährt die Querstange
                   über dem gelagerten Glas nach oben aus. Die Querstange betätigt
                   ein Verschlussystem unter der Flasche, worauf die Flüssigkeit
                   durch das Schläuchchen ins Glas fließt.

  Bluetooth-Gruppe: 1 (Bestelleinheit -> Master, Glaslager, Keeper, S-Oben -> Slaves)
  Autor:            Adrian Steiner / Simon Gisler
  Erstellt:        27. März 2008
  Geändert:        19. Mai 2008
  Hinweis:         keine Sonderzeichen verwenden (ä, ö, ü, etc.)
*/

```

```

//-----

```

```

sub BTCheck(int conn)
{
  if (!BluetoothStatus(conn)==NO_ERR)
  {
    TextOut(0,LCD_LINE1,"BT-Fehler",true);
    PlayTone(5000,1000); // Warnton
    Wait(4000);
    Stop(true);
  }
}

```

```

//-----

```

```

sub Abfuellen(int zeit,int geschwindigkeit)
{
  int powerB=100,distanz=5400;

  Off(OUT_A); // Wagen zuerst stoppen
  SetSensor(S2,SENSOR_TOUCH);
  RotateMotorEx(OUT_BC,-powerB,distanz,0,true,true); // Querstange ausfahren
  Wait(zeit); // Abfüllzeit abwarten
  OnFwdSync(OUT_BC,powerB,0); // Querstange einfahren
  until (SENSOR_2==1); // Endschalter -> Ausgangsposition
  Off(OUT_BC);
  OnFwd(OUT_A,geschwindigkeit); // Wagen wieder vorwärtsbewegen
}

```

```

//-----

```

```

sub Abfahren(int t1,int t2,int t3,int t4,int t5)
{
  int powerA=90,prellZeit=340;

  SetSensor(S1,SENSOR_TOUCH);
  OnFwd(OUT_A,powerA);
  Wait(prellZeit); // damit weggefahren werden kann, Wagenstopp unter 'Abfuellen'
  until(SENSOR_1==1); // 1. Halt
  if (t1>0)
  {
    Abfuellen(t1,powerA);
  }
  Wait(prellZeit);
  until(SENSOR_1==1); // 2. Halt
  if (t2>0)
  {
    Abfuellen(t2,powerA);
  }
  Wait(prellZeit);
  until(SENSOR_1==1); // 3. Halt
  if (t3>0)
  {
    Abfuellen(t3,powerA);
  }
  Wait(prellZeit);
  until(SENSOR_1==1); // 4. Halt
  if (t4>0)
  {
    Abfuellen(t4,powerA);
  }
}

```

```

Wait(prellZeit);
until(SENSOR_1==1); // 5. Halt
powerA=70;
OnFwd(OUT_A,powerA);
if (t5>0)
{
    Abfuellen(t5,powerA);
}
Wait(prellZeit);
until(SENSOR_1==1); // zur Ausladeposition (Lift) fahren
Off(OUT_A);
}

```

//-----

```

sub RetourFahren()
{
    int zaehler,powerA=90,prellZeit=340;

    SetSensor(S1,SENSOR_TOUCH);
    for (zaehler=0;zaehler<6;zaehler++)
    {
        if (zaehler==5) powerA=70;
        OnRev(OUT_A,powerA);
        Wait(prellZeit);
        until(SENSOR_1==1);
    }
    Off(OUT_A);
}

```

//-----

```

#define BT_CONN_BESTELLEINHEIT 0
#define INBOX_BESTELLEINHEIT 1

```

```

task main()
{
    int drink;
    bool retour;

    BTCheck(BT_CONN_BESTELLEINHEIT);
    TextOut(0,LCD_LINE1,"Empfangsbereit",true);
    while(true)
    {
        drink=0;
        while(drink==0)
        {
            ReceiveRemoteNumber(INBOX_BESTELLEINHEIT,true,drink)
        }
        switch(drink)
        {
            case 1:
                Abfahren(0,13000,0,0,16000);
                break;
            case 2:
                Abfahren(0,0,13000,0,8000);
                break;
            case 3:
                Abfahren(0,0,0,13000,8000);
                break;
            case 4:
                Abfahren(0,7000,0,7000,0);
                break;
            case 5:
                Abfahren(7000,7000,0,0,0);
                break;
            case 6:
                Abfahren(0,7000,7000,0,0);
                break;
            case 7:
                Abfahren(7000,0,0,7000,0);
                break;
            case 8:
                Abfahren(16000,0,0,0,0);
                break;
            case 9:
                Abfahren(0,0,0,16000,0);
                break;
        }
    }
}

```

```
    case 10:
        Abfahren(0,0,16000,0,0);
        break;
    case 11:
        Abfahren(0,16000,0,0,0);
        break;
    case 12:
        Abfahren(2000,2000,2000,2000,8000);
        break;
    default:
        break;
}
retour=false;
while(!retour)
{
    ReceiveRemoteBool(INBOX_BESTELLEINHEIT,true,retour)
}
RetourFahren();
}
```

```

/*
  Programmname:      S-Oben.nxc
  Programmiersprache: NXC
  Brick Type:       NXT
  Funktion:         Glasschieber oben wartet, bis der Keeper (über Endschalter) und der
                   Lift (über Lichtsensor) bereit sind. Danach wird das Glas in den
                   Lift geschoben. Sobald der Schieber wieder zurück ist, wird dies
                   der Bestelleinheit gemeldet. Somit kann die Bestelleinheit den
                   Keeper zurückrufen.

  Bluetooth-Gruppe: 1 (Bestelleinheit -> Master, Glaslager, Keeper, S-Oben -> Slaves)
  Autor:            Adrian Steiner / Simon Gisler
  Erstellt:         27. März 2008
  Geändert:         02. August 2008
  Hinweis:          keine Sonderzeichen verwenden (ä, ö, ü, etc.)
*/

//-----

sub BTCheck(int conn)
{
  if (!BluetoothStatus(conn)==NO_ERR)
  {
    TextOut(0,LCD_LINE1,"BT-Fehler",true);
    PlayTone(5000,1000); // Warnton
    Wait(4000);
    Stop(true);
  }
}

//-----

#define BT_CONN_BESTELLEINHEIT 0
#define OUTBOX_SCHIEBER 1

bool LiftBereit(int dunkel)
{
  int offset=4;
  if (SENSOR_3>=dunkel+offset || SENSOR_3<=dunkel-offset)
  {
    return false;
  }
  else
  {
    return true;
  }
}

//-----

task main()
{
  int geschwindigkeit=100,distanz=2850,dunkel;

  SetSensor(S1,SENSOR_TOUCH);
  SetSensor(S2,SENSOR_TOUCH);
  SetSensor(S4,SENSOR_TOUCH);
  SetSensorLight(IN_3);
  BTCheck(BT_CONN_BESTELLEINHEIT);
  while (SENSOR_4==0) // Lift muss zum Eichen oben sein!
  {
    TextOut(0,LCD_LINE1,"Lift > nach oben!");
    PlayTone(3000,100); // Warnton
    Wait(200); // damit Warnton nicht zu stark nervt...
  }
  dunkel=Sensor(IN_3);
  ResetScreen();
  while(true)
  {
    until (SENSOR_1==1 && LiftBereit(dunkel)); // Kepper und Lift bereit?
    RotateMotor(OUT_A,geschwindigkeit,distanz); // Glas in Lift schieben
    RotateMotor(OUT_A,-geschwindigkeit,distanz-200); // Schieber zurückziehen
    SendResponseBool(OUTBOX_SCHIEBER,true); // Meldung an Bestelleinheit
    OnRev(OUT_A,geschwindigkeit-50); // Schieber langsam zurückziehen
    until (SENSOR_2==1); // Endschalter Schieber
    Off(OUT_A); // Endschalter Schieber
    until (SENSOR_1==0); // Warten bis Keeper abgefahren
  }
}

```

```

/*
  Programmname:      Lift.nxc
  Programmiersprache: NXC
  Brick Type:       NXT
  Funktion:         Sobald der Ultraschallsensor ein Glas erkannt hat, wird der Lift
                   zum Zwischenhalt gefahren. Dieser Zwischenhalt ist notwendig, weil
                   zuerst kontrolliert werden muss, ob der Wagen bereits auf den Lift
                   wartet. Die Kontrolle wird über Bluetooth realisiert. Wenn der
                   Wagen bereit ist, fährt der Lift bis zum unteren Endschalter.
                   Über Bluetooth wird der untere Glasschieber aktiviert. Danach fährt
                   der Lift nach oben bis zum oberen Endschalter.

  Bluetooth-Gruppe: 2 (Lift -> Master, Wagen, S-Unten -> Slaves)
  Autor:            Adrian Steiner / Simon Gisler
  Erstellt:        27. März 2008
  Geändert:        06. Juni 2008
  Hinweis:         keine Sonderzeichen verwenden (ä, ö, ü, etc.)
*/

//-----

sub BTCheck(int conn)
{
  if (!BluetoothStatus (conn)==NO_ERR)
  {
    TextOut (0,LCD_LINE1, "BT-Fehler",true);
    PlayTone (5000,1000); // Warnton
    Wait (4000);
    Stop (true);
  }
}

//-----

bool AufGlasWarten()
{
  int distanzGlas,anzahlMessungen=0,maxMessungen=2;
  string meldung;

  while (anzahlMessungen<maxMessungen)
  {
    ResetSensor (IN_4);
    distanzGlas=SensorUS (IN_4);
    meldung=NumToStr (distanzGlas);
    TextOut (0,LCD_LINE1, "auf Glas warten",true);
    TextOut (0,LCD_LINE2, StrCat ("Distanz = ",meldung));
    if (distanzGlas<14)
    {
      anzahlMessungen=anzahlMessungen+1;
    }
    else
    {
      anzahlMessungen=0;
    }
  }
  Wait (2000); // bis Schieber weg ist
  return true;
}

//-----

#define BT_CONN_WAGEN 1
#define BT_CONN_SCHIEBER 2
#define OUTBOX_SCHIEBER 1
#define INBOX_WAGEN 1

task main()
{
  int geschwindigkeit=100;
  bool empfang;

  SetSensor (S1, SENSOR_TOUCH);
  SetSensor (S2, SENSOR_TOUCH);
  SetSensorLowspeed (IN_4);
  BTCheck (BT_CONN_WAGEN);
  BTCheck (BT_CONN_SCHIEBER);
  while (true)
  {
    if (AufGlasWarten())

```

```

{
  TextOut(0,LCD_LINE1,"zum Zwischenhalt",true);
  RotateMotor(OUT_A,-geschwindigkeit,6300); // Lift -> Zwischenstation
  empfang=false; // Wagen bereit?
  while(!empfang)
  {
    ReceiveRemoteBool(INBOX_WAGEN,true,empfang); // inkl. 'MessageRead'
  }
  TextOut(0,LCD_LINE1,"zum unteren Ende",true);
  OnRev(OUT_A,geschwindigkeit-20); // Lift nach unten fahren
  until(SENSOR_1==1); // Endschalter unten
  Off(OUT_A); // Lift stoppen
  TextOut(0,LCD_LINE1,"Schieber",true);
  until (BluetoothStatus(BT_CONN_SCHIEBER)==NO_ERR); // warten bis BT frei ist
  SendRemoteBool(BT_CONN_SCHIEBER,OUTBOX_SCHIEBER,true); // Schieber starten
  Wait(6200); // bis Glas auf Wagen
  TextOut(0,LCD_LINE1,"zum oberen Ende",true);
  RotateMotor(OUT_A,geschwindigkeit,10000); // Lift teilweise -> oben
  OnFwd(OUT_A,geschwindigkeit-20);
  until(SENSOR_2==1); // Endschalter oben
  Off(OUT_A); // Lift stoppen
}
}
}

```

```

/*
  Programmname:      S-Unten.nxc
  Programmiersprache: NXC
  Brick Type:       NXT
  Funktion:         Glasschieber unten, wartet auf das Startsignal vom Lift.
                   Danach wird das Glas auf den Wagen geschoben.
  Bluetooth-Gruppe: 2 (Lift -> Master, Wagen, S-Unten -> Slaves)
  Autor:           Adrian Steiner / Simon Gisler
  Erstellt:       08. Juni 2008
  Geändert:
  Hinweis:       keine Sonderzeichen verwenden (ä, ö, ü, etc.)
*/

//-----

sub BTCheck(int conn)
{
  if (!BluetoothStatus(conn)==NO_ERR)
  {
    TextOut(0,LCD_LINE1,"BT-Fehler",true);
    PlayTone(5000,1000); // Warnton
    Wait(4000);
    Stop(true);
  }
}

//-----

#define BT_CONN_LIFT 0
#define INBOX_LIFT 1

task main()
{
  int geschwindigkeit=100,distanz=1620;
  bool empfang;

  SetSensor(S1,SENSOR_TOUCH);
  BTCheck(BT_CONN_LIFT);
  while(true)
  {
    empfang=false;
    while(!empfang)
    {
      ReceiveRemoteBool(INBOX_LIFT,true,empfang); // warten auf ok von Lift
    }
    RotateMotor(OUT_A,geschwindigkeit,distanz); // Glas auf Wagen schieben
    RotateMotor(OUT_A,-geschwindigkeit,distanz-200); // Schieber zurückziehen
    OnRev(OUT_A,geschwindigkeit-50); // Schieber langsam zurückziehen
    until(SENSOR_1==1); // Endschalter -> Ausgangsposition
    Off(OUT_A);
  }
}

```



```

/*
Programmname:      Wagen.nxc
Programmiersprache: NXC
Brick Type:       NXT
Funktion:          Zuerst werden die beiden Sensoren automatisch geeicht. Dieser
                  Vorgang wiederholt sich bei jedem Durchgang. Damit können langsame
                  Lichtschwankungen abgefangen werden. Danach fährt der Wagen zum
                  Lift. Dort wird gewartet bis das Glas per Ultraschall erkannt wird.
                  Danach transportiert der Wagen das Glas zum Mixer und danach zurück
                  zur Bestelleinheit. Dort wird gewartet bis per Ultraschall erkannt
                  wird, dass das Glas entfernt wurde. Nach dem erneuten Eichen fährt
                  der Wagen zurück zum Lift und der Vorgang beginnt von vorne.
                  Das Linienfolgen wird mit einem Sensor realisiert. Beim Eichen
                  werden die Werte für 'hell' und 'dunkel' erfasst. Der Sollwert wird
                  aus dem Mittelwert ->(hell + dunkel)/2 berechnet. Der Wagen fährt
                  sozusagen auf 'grau', bzw. am Rand der Linie. Mit Hilfe eines PD-
                  Reglers folgt der Wagen präzise dem Linienrand. Der zweite Licht-
                  sensor zählt die Querstriche und somit kennt die Software die
                  genaue Position des Wagens.
                  Der Wagen kann nicht an ein Netzgerät angeschlossen werden. Deshalb
                  werden die Lichtsensoren bei Nichtgebrauch ausgeschaltet, damit
                  wird die Einsatzdauer des Akkus wesentlich verlängert.
                  Weil die Ultraschallmessung nicht immer präzise funktioniert, müssen
                  zwei korrekte Messungen nacheinander erfolgen, damit die Subroutine
                  "AufGlas-Warten" "wahr" zurückliefert. Der Ultraschallsensor wird
                  direkt von der Akkuspannung ohne Stabilisierung versorgt. Deshalb
                  wird pro Durchgang noch zusätzlich die Akkuspannung überprüft. Bei
                  weniger als 7.2 Volt wird der Wagen gestoppt. Der NXT hat zwar
                  bereits eine Akkuüberwachung eingebaut, diese setzt aber erst bei
                  einer tieferen Spannung ein.
Autor:            Adrian Steiner / Simon Gisler
Erstellt:        27. März 2008
Geändert:        05. Juli 2008 (an neue Konstruktion angepasst, PD-Regler)
Hinweis:        keine Sonderzeichen verwenden (ä, ö, ü, etc.)
*/

```

```

//-----

```

```

sub BTCheck(int conn)
{
  if (!BluetoothStatus(conn)==NO_ERR)
  {
    TextOut(0,LCD_LINE1,"BT-Fehler",true);
    PlayTone(5000,1000); // Warnton
    Wait(4000);
    Stop(true);
  }
}

```

```

//-----

```

```

sub BatterieCheck()
{
  int spannung;
  string wert;

  spannung=BatteryLevel();
  if (spannung<7200)
  {
    wert=NumToStr(spannung);
    TextOut(0,LCD_LINE1,StrCat("U in mV = ",wert),true);
    TextOut(0,LCD_LINE2,"Batterie laden");
    PlayTone(5000,1000); // Warnton
    Wait(4000);
    Stop(true);
  }
  ResetScreen();
}

```

```

//-----

```

```

sub Eichen(int &hell,int &dunkel,int &referenz)
{
  int anfang,v=100,offset=8;
  bool wiederholen=true;

  while(wiederholen)
  {

```

```

ResetRotationCount (OUT_A);
Wait (5); // Tipp von John C. Hansen
anfang=MotorRotationCount (OUT_A);
if (anfang==0)
{
    wiederholen=false;
}
}
SetSensorLight (IN_2); // Lichtsensoren einschalten
SetSensorLight (IN_3);
OnFwdSync (OUT_AC, v, 0); // möglichst geradeaus fahren
until ((MotorRotationCount (OUT_A)-anfang)>45); // auf helle Eichstelle fahren
hell=Sensor (IN_2); // Sensorwert 'hell' für Randfahren
referenz=Sensor (IN_3); // Referenzwert für Querstrich hell
until (Sensor (IN_2)<hell-offset); // fahren bis dunkle Eichstelle beginnt
anfang=MotorRotationCount (OUT_A);
until ((MotorRotationCount (OUT_A)-anfang)>15); // auf dunkle Eichstelle fahren
dunkel=Sensor (IN_2); // Sensorwert 'dunkel' für Randfahren
until (Sensor (IN_2)>dunkel+offset); // fahren bis dunkle Eichstelle endet
}

//-----
sub StrichFahren(int hell,int dunkel,int referenz)
{
    #define Kp 4
    #define Kd 60
    #define max_v 100
    #define v 100

    int abweichung=0,abweichungAlt=0,istWert,sollWert,stellGroesse,links,rechts;
    bool wiederholen=true;

    SetSensorLight (IN_2); // Lichtsensoren einschalten
    SetSensorLight (IN_3);
    sollWert=(hell+dunkel)/2;
    OnFwdSync (OUT_AC, v, 0); // sicher über Querstrich fahren
    Wait (200);
    while (wiederholen)
    {
        istWert=Sensor (IN_2);
        abweichung=sollWert-istWert;
        stellGroesse=Kp*abweichung+Kd*(abweichung-abweichungAlt);
        abweichungAlt=abweichung;
        links=v-stellGroesse;
        if (links>max_v) links=max_v; // Geschwindigkeit links begrenzen
        if (links<-max_v) links=-max_v;
        OnFwd (OUT_C, links);
        rechts=v+stellGroesse;
        if (rechts>max_v) rechts=max_v; // Geschwindigkeit rechts begrenzen
        if (rechts<-max_v) rechts=-max_v;
        OnFwd (OUT_A, rechts);
        if (abs ((Sensor (IN_3)-referenz))>15) // wenn Querstrich erkannt
        {
            Off (OUT_AC);
            wiederholen=false;
        }
    }
}

//-----
sub AufGlasWarten()
{
    #define OUTBOX_WAGEN 1

    int distanzGlas,anzahlMessungen=0,maxMessungen=2;
    string meldung;

    BatterieCheck ();
    SetSensorLowSpeed (IN_1);
    SetSensor (IN_2, SENSOR_TYPE_NONE); // Lichtsensoren ausschalten
    SetSensor (IN_3, SENSOR_TYPE_NONE);
    SendResponseBool (OUTBOX_WAGEN, true); // ok, Lift kann Glasschieber aktivieren
    while (anzahlMessungen<maxMessungen)
    {
        ResetSensor (IN_1);
        distanzGlas=SensorUS (IN_1);
    }
}

```

```

meldung=NumToStr(distanzGlas);
TextOut(0,LCD_LINE1,"auf Glas warten",true);
TextOut(0,LCD_LINE2,StrCat("Distanz = ",meldung));
if(distanzGlas<14)
{
    anzahlMessungen=anzahlMessungen+1;
}
else
{
    anzahlMessungen=0;
}
}
Wait(8700); // bis Schieber weg ist
ResetScreen();
}

//-----

sub WartenBisGlasWeg()
{
    int distanzGlas,anzahlMessungen=0,maxMessungen=2;
    string meldung;

    SetSensorLowspeed(IN_1);
    SetSensor(IN_2,SENSOR_TYPE_NONE); // Lichtsensoren ausschalten
    SetSensor(IN_3,SENSOR_TYPE_NONE);
    while(anzahlMessungen<maxMessungen)
    {
        ResetSensor(IN_1);
        distanzGlas=SensorUS(IN_1);
        meldung=NumToStr(distanzGlas);
        TextOut(0,LCD_LINE1,"bis Glas weg",true);
        TextOut(0,LCD_LINE2,StrCat("Distanz = ",meldung));
        PlayTone(3000,100); // Warnton
        Wait(200); // damit Warnton nicht zu stark nervt...
        if(distanzGlas>14)
        {
            anzahlMessungen=anzahlMessungen+1;
        }
        else
        {
            anzahlMessungen=0;
        }
    }
    Wait(500); // bis Glas (Hand) weg ist
    ResetScreen();
}

//-----

#define BT_CONN_LIFT 0

task main()
{
    int hell,dunkel,referenz,zustand=0,anzahlStriche;

    BTCheck(BT_CONN_LIFT);
    while(true)
    {
        switch(zustand)
        {
            case 0:
                Eichen(hell,dunkel,referenz);
                anzahlStriche=1;
                zustand=1;
                break;
            case 1:
                StrichFahren(hell,dunkel,referenz);
                anzahlStriche=anzahlStriche+1;
                zustand=anzahlStriche;
                break;
            case 2: // beim Lift
                AufGlasWarten();
                zustand=1;
                break;
            case 3: // beim Mixer
                SetSensor(IN_2,SENSOR_TYPE_NONE); // Lichtsensoren ausschalten
                SetSensor(IN_3,SENSOR_TYPE_NONE);
        }
    }
}

```

```
TextOut(0,LCD_LINE1,"Mixer",true);
Wait(6500);
ResetScreen();
zustand=1;
break;
case 4: // bei der Bestelleinheit
WartenBisGlasWeg();
zustand=0;
break;
}
}
}
```

```

/*
  Programmname:      Mixer.nxc
  Programmiersprache: NXC
  Brick Type:       NXT
  Funktion:         Sobald das Glas über Ultraschall erkannt wird, startet der Mixer.
                   Weil die Ultraschallmessung nicht immer präzise funktioniert, müssen
                   zwei korrekte Messungen nacheinander erfolgen, damit die Subroutine
                   'AufGlasWarten' 'wahr' zurückliefert.

  Bluetooth-Gruppe: keine
  Autor:            Adrian Steiner / Simon Gisler
  Erstellt:         19. Juni 2008
  Geändert:
  Hinweis:          keine Sonderzeichen verwenden (ä, ö, ü, etc.)
*/

//-----

bool AufGlasWarten()
{
  int  distanzGlas,anzahlMessungen=0,maxMessungen=2;
  string meldung;

  while(anzahlMessungen<maxMessungen)
  {
    ResetSensor(IN_4);
    distanzGlas=SensorUS(IN_4);
    meldung=NumToStr(distanzGlas);
    TextOut(0,LCD_LINE1,"auf Glas warten",true);
    TextOut(0,LCD_LINE2,StrCat("Distanz = ",meldung));
    if (distanzGlas<14)
    {
      anzahlMessungen=anzahlMessungen+1;
    }
    else
    {
      anzahlMessungen=0;
    }
  }
  TextOut(0,LCD_LINE1,"mischen...",true);
  return true;
}

//-----

task main()
{
  int geschwindigkeit=100,distanz=750;

  SetSensorLowSpeed(IN_4);
  while(true)
  {
    if (AufGlasWarten())
    {
      Wait(1000); // warten bis Wagen hält
      RotateMotor(OUT_A,-geschwindigkeit,distanz); // Mixer ins Glas bewegen
      OnFwd(OUT_B,geschwindigkeit); // mixen starten
      Wait(2000);
      Off(OUT_B); // mixen stoppen
      RotateMotor(OUT_A,geschwindigkeit,distanz);
      Wait(4000); // warten bis Wagen weggefahren ist
    }
  }
}

```

theater(uri)

Jacques Brels Musik lebt auf

MZ. Nicolas Turicchia war schwer beeindruckt, als er das erste Mal aus der Juke-Box in der kleinen Dorfbar im Wallis, wo er aufgewachsen ist, die Stimme von Jacques Brel hörte. «Ich war wie vom Blitz getroffen», erinnert er sich. Der Traum, seinen Worten auch eine Stimme zu geben, hat Turicchia fasziniert. Jetzt, aus 22-jähriger Distanz heraus, hat er sich entschieden, den Traum umzusetzen. Was daraus entstanden ist, kann das Publikum am kommenden Donnerstag im theater(uri) hören.

Der Liederabend mit Nicolas Turicchia (Gesang) und Miguel Sesma (Klavier) entstand aus Anlass des 30. Todesjahres des belgischen Liedermachers und Komponisten. Die Themen von Jacques Brels Lieder sind sehr unterschiedlich. Sie erkunden Liebe, Gesellschaft und spirituelle Aspekte. Seine Werke überschreiten die Stilrichtungen. Brel kann unterhaltsam, aber auch tiefgründig ernst sein.

HINWEIS

► Der Liederabend mit Nicolas Turicchia (Gesang) und Miguel Sesma (Klavier) im theater(uri) am kommenden Donnerstag, 6. März, beginnt um 19 Uhr. Es wird eine Türkollekte aufgenommen. ◀

LESERBRIEF

Ziel: Flüelen 2020 mit null Geleisen

Zur Standesinitiative «Verlagerung des alpenquerenden Güterverkehrs auf die Schiene»

Mit grosser Mehrheit hat der Landrat in der Februar-Session die Motion Braunwalder erheblich erklärt. Der Regierungsrat steht auch dahinter.

Auch ich habe mich stark gemacht für die Standesinitiative: Das Volk hat sich entschieden, den alpenquerenden Güterverkehr zu grossen Teilen auf die Schiene zu verlagern. Dafür investieren wir Milliardenbeträge in hochmoderne Tunnelsysteme. Die Strasse soll möglichst dem privaten Verkehr und dem lokalen Gütertransport zur Verfügung stehen. Bahn und Strassen sollen sich intelligent ergänzen zu einem integrierten Verkehrssystem mit dem Ziel, den Verkehr fliessen zu lassen.

Damit das geschehen kann, müssen aber gleichzeitig auch die Zulaufstrecken ausgebaut werden. Dies wird in Punkt 2 der Motion gefordert. Im Klartext heisst das, dass jetzt der Aentunnel gebaut werden muss. Damit wird der Weg frei für eine Umfahrung von Flüelen. Ohne Umfahrung wird Flüelen 24 Stunden Bahnlärm ausgesetzt werden. Das werden auch die Lärmschutzwände nicht schlucken. Flüelen 2020 mit null Geleisen. Das ist ein Ziel, für das ich mich zu 100 Prozent einsetze.

THOMAS ARNOLD, LANDRAT, FLÜELEN

URNER ZEITUNG

IMPRESSUM

Herausgeberin: Neue Luzerner Zeitung AG, Maihofstrasse 76, Luzern, Erwin Bachmann, Delegierter des Verwaltungsrates, E-Mail: erwin.bachmann@medien.ch

Verlag: Jürg Weber, Geschäfts- und Verlagsleiter; Ueli Kaltener, Leserkontakt; Edi Lindegger, Anzeigenmarkt; Ombudsmann: Dr. Andreas Z'Graggen

Redaktion: Neue Urner Zeitung: Bruno Arnold (bar), Redaktionsleiter; Markus Zwysig (MZ), stv. Redaktionsleiter; Angel Sanchez (san); Sven Aregger (ars); redaktionelle Mitarbeiter: Philipp Arnold (pha); Urs Hanhart (uha), Reporter; Manuela Kalbermaten (mak)

Redaktionsleitung Neue Luzerner Zeitung und Regionalausgaben: Chefredaktor: Thomas Bornhauser (TbB); Stv. Chefredaktor/überregionale Ressorts: Stefan Ragaz (az); Regionale Ressorts: Stefan Inderbitzin (sti); Sport: Marco von Ah (mva), Gruppe Gesellschaft und Kultur: Arno Renggli (are)

Ressortleiter: Nachrichten: Dominik Buholzer (bu); Sven Gallinelli (sg); Politik: Raphael Prinz (rp); Markt/Wirtschaft: Inge Staub (ist); Sport: Marco von Ah (mva); Gesellschaft und Kultur: Arno Renggli (are); Zentralschweiz: Cyril Aregger (ca); Leben: Hans Graber (hag); Dossier: Iwona Meyer-Swielik (im); Montag: Manuel Gübeli (mag); APERO: Thomas Huwyler (huw); Zisch: Robert Bachmann (bac); Fotos: Eveline Bachmann (ev)

Adressen und Telefonnummern: 6460 Altdorf, Höfligasse 3. Redaktion: Telefon 041 874 21 60, Fax 041 874 21 61, E-Mail: redaktion@neue-uz.ch. Hauptredaktion: Telefon 041 429 51 51, Fax 041 429 51 81, E-Mail: redaktion@neue-uz.ch

Abonnemente und Zustelldienst: Telefon 041 871 10 20, Fax 041 429 53 83, E-Mail: abo@neue-uz.ch. Billetvorverkauf: Telefon 041 874 21 60 (60 Rp./Min.)

Technische Herstellung: Neue Luzerner Zeitung AG, 6002 Luzern, Maihofstrasse 76, Postfach, Telefon 041 429 52 52, Fax 041 429 52 89

Anzeigen: Gamma Druck + Verlag AG, Medien-Atelier, Höfligasse 3, 6460 Altdorf, Telefon 041 874 21 50, Fax 041 874 21 58. Technischer Kundendienst Anzeigen: Telefon 041 227 56 56

Für Todesanzeigen an Sonn- und Feiertagen: Fax 041 429 53 91. Abonnementspreis: 12 Monate Fr. 356.–/6 Monate Fr. 186.– (inkl. 2,4% MWST)

Die irgendwie gearbeitete Verwertung von in diesem Titel abgedruckten Anzeigen oder Teilen davon, insbesondere durch Einspeisung in einen Online-Dienst, durch dazu nicht autorisierte Dritte ist untersagt. Jeder Verstoß wird gerichtlich verfolgt.

Berufsmaturitätsschule Uri

Lego-Roboter lernt das Laufen

Frauenbewegung, italienische Einwanderer in Uri und ein Lego-Barkeeper: Das sind die Themen, welche die Berufsmaturitätsschüler in ihrer Projektwoche untersuchten.

red. Die technische und gesundheitlichsoziale Berufsmaturitätsschule Uri führt jedes Jahr mit ihren Schülern eine Projektwoche durch. Die Studenten wählen selbstständig ein Thema, welches sie erforschen, erarbeiten und präsentieren.

Sobald die Berufsmaturitätsklasse die Projektaufgabe erhalten hatte, befassten sich alle Gruppen intensiv mit der Wahl ihres Themas. In der Vorbereitungsphase war jedes Team damit beschäftigt, Informationen und Bilder zu sammeln und geeignete Hilfspersonen ausfindig zu machen. Dies war ein wichtiger Teil, um in der Projektwoche nicht unnötig viel Zeit zu verlieren.

Eine spannende Zeitreise

Die erste Zweiergruppe begab sich auf die Spuren des Gotthardtunnelbaus. Intensiv befassten sie sich mit den geschichtlichen und sozialen Aspekten der damaligen Zeit. Dabei stützten sie sich auf die Leitfrage: «Was machte den Kanton Uri um 1882 für Italiener attraktiv?»

Die Familie Cavaletti aus Göschenen, Nachfahren eines italienischen Mineurs, lieferte den Jugendlichen wertvolle Erinnerungen. Eine solche Familie ausfindig zu machen gestaltete sich jedoch schwieriger als erwartet. Am Schluss kristallisierte sich deutlich heraus, dass die Arbeitssuche der damaligen Hauptgrund für die Einwanderung war. Ausserdem gewann die Gruppe einen umfassenden Einblick in die Lebensumstände der italienischen Arbeiter.

Frauen sollen an die Macht

Eine weitere Zweiergruppe setzte sich mit dem Thema «Die Frau im Wandel der Zeit» auseinander. Zu den Vorbereitungen gehörten eine Materialsammlung, eine Feldforschung, in der die Gruppe die Wahrnehmung gegenüber der Frau aufzeigen wollte und zwei Interviews mit Urner Frauen. Die Jugendlichen konnten während dieser Projektarbeit viele Erfahrungen sammeln: Dazu gehören selbstständiges Arbeiten und Recherchieren, das Erstellen einer optimalen Zeiteinteilung, die

Arbeit im Team sowie die Fähigkeit, Interviews zu führen. Die Hauptkenntnis der Projektwoche liegt darin, dass die Frau dem Mann gesetzlich gleichgestellt ist, aber dies lange nicht in allen Bereichen so durchgeführt wird. Zum Schluss konnte die Gruppe ihre Ergebnisse in einer Präsentation aufzeigen.

Roboter als Barkeeper

Nach intensiver Vorarbeit war das Ziel der dritten siebenköpfigen Gruppe, innert einer Woche einen funktionsfähigen Lego-Barkeeper zu bauen. Geboren wurde diese aussergewöhnliche Idee während eines Aufenthaltes in einer Altdorfer Stammbaar. Die Jugendlichen begannen zahlreiche Legoteile zu verbauen und die Steuerung zu programmieren. Die Arbeit verlief reibungslos, wodurch schon am Mittwochabend der erste Drink bestellt

werden konnte. In den folgenden Tagen wurden dann die Feinabstimmungen erledigt. Am Freitag durfte die Gruppe ihre Arbeit mit Stolz der Öffentlichkeit präsentieren, welche sich ausgesprochen interessiert am selbstständig arbeitenden Barkeeper zeigte.

Vorbereitung für die Zukunft

Nachdem das Getränk bestellt ist, mischt der Roboter bei einer Abfüllstation die Zutaten des Drinks selbstständig. Anschliessend wird er einem Lift übergeben, der ihn nach unten befördert und ihn einem weiteren Fahrzeug übergibt. Dieses intelligente, sensor-gesteuerte Fahrzeug bringt das Getränk zurück zum Besteller. Das Ziel während dieser Projektwoche war es nicht nur, den Roboter auf die Beine zu stellen, sondern auch das Arbeiten in einer Gruppe näher kennen zu lernen. Es war eine gute Möglichkeit für die jungen

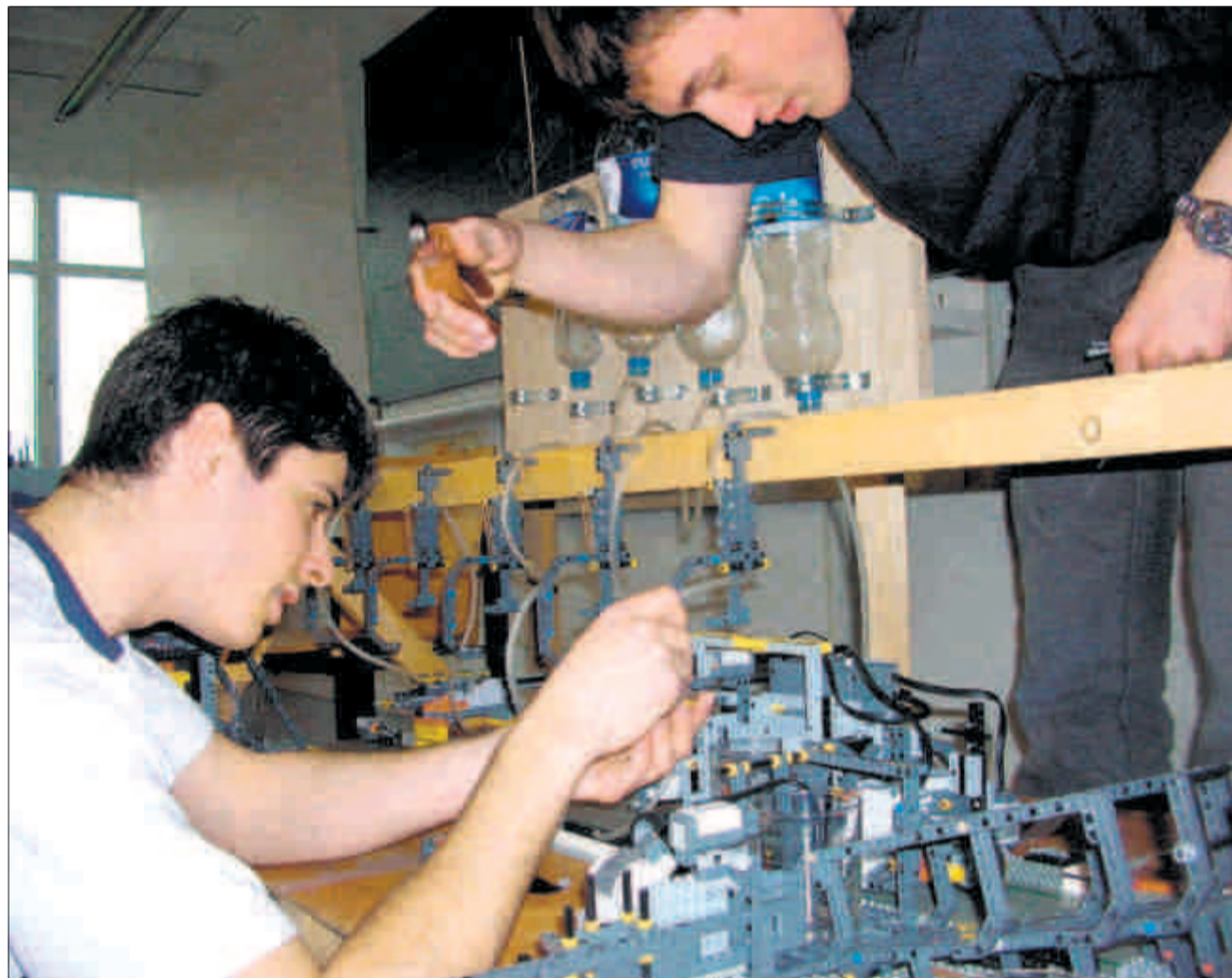
EXPRESS

- Die Berufsmaturitätsschule Uri führte mit den Schülern eine Projektwoche durch.
- In drei Gruppen erforschten und erarbeiteten die Schüler ihr selbst gewähltes Thema.
- Am Ende der Woche wurden die Arbeiten der Öffentlichkeit präsentiert.

Leute, selbstständig ein Projekt zu realisieren, um sie auf den künftigen Werdegang vorzubereiten.

HINWEIS

► Video und Bilder des Barkeeper-Roboters gibt es im Internet unter <http://www.fraengg.ch>. ◀



Jost Furrer (links) und Manfred Arnold bei der Optimierung des Abfüllsystems, die dem Roboter die anschliessende Arbeit als Barkeeper erleichtert.

BILD PD

LESERBRIEFE

Soll Schattdorferne unter Wasser gesetzt werden?

Zum Artikel «Blöchliger schlägt eine Flussbrücke vor» in der Neuen UZ vom 21. Februar

Landrat Oskar Blöchliger (SVP, Altdorf) und 38 Mitunterzeichner schlagen in einem Postulat vor, den Schächen bei der Ruag über die Strasse, Eisenbahn und Nationalstrasse hinwegzuführen.

Um dieses Vorhaben zu ermöglichen, soll das Gefälle des Schächen fast auf die Hälfte reduziert werden. «Geschiebesammler sind im Schächen wegen der viel zu grossen Geschiebemenge eindeutig nachteilig», argumentiert der Postulant laut Zeitungsbericht. Wie sinnvoll diese Aussage ist, mag jeder selber entscheiden.

Was stellt sich Noch-Landrat Blöchliger vor? Einen wild reissenden Schächen mit möglichst vielen Baumstämmen und donnerndem Geschiebe hoch über der Nationalstrasse und einen beeindruckenden Wasserfall in die Reuss bei Attinghausen?

Ich kann diesem Postulat überhaupt nichts abgewinnen. Der Schächen würde in seinem Hochbett über der Landschaft die Reuss nie und nimmer erreichen. Die Baumstämme und das gewaltige Gesteinsmaterial, das bei der Katastrophe 1977 bis zu über 2 Kubik-

meter grosse Felsbrocken enthielt, würde das gesamte Volumen des Schächenkanals füllen, da keine Kiessammler mehr Holz und Gesteinsmaterial zurückhielten. Mit Sicherheit würde der Ruag-Areal das neue Bachbett verlassen. Schlimmstenfalls würde er sich bereits im Raume Fussballplatz Schattdorf austoben.

Alles weist darauf hin, dass sich meteorologische Extremereignisse in nächster Zeit häufen werden. Eine intensive Regenperiode, die zwei oder sogar drei Tage anhält, würde bei der

Neugestaltung des Schächen nach dem Vorschlag Blöchlingers ohne Zweifel in eine Katastrophe münden.

Auch die Erhöhung der Reussdämme, wie sie von Blöchliger auch vorgeschlagen wird, erscheint mir unsinnig, hat man die Dämme doch erst vor kurzem neu gestaltet. Überdies müsste die Nationalstrasse verlegt werden, wertvolles Grünland ginge beiderseits der Reuss verloren. Die Idee überzeugt nicht. Man muss sie wohl als Ausfluss des Bemühens deuten, vor den Wahlen ein bisschen Aufmerksamkeit zu erregen: Angaben zur Kostenfolge fehlen jedenfalls. Ich warte gespannt auf die Bewertung des Projekts durch die verantwortlichen Stellen.

MAX ROTHENFLUH, SCHATTDORF

«Schlimmstenfalls würde sich der Schächen bereits im Raume Fussballplatz Schattdorf austoben.»

MAX ROTHENFLUH,
SCHATTDORF

Beatrice Bünter: initiativ und immer offen für Neues

Zu den Landratswahlen vom 6. April in der Gemeinde Altdorf

Wahlen heisst auch die Zukunft mitgestalten. Speziell in der heutigen Zeit benötigen wir Politikerinnen und Politiker, die sich den echten Problemen annehmen. Jugendprobleme, Perspektivlosigkeit von Jugendlichen in unserer globalisierten Welt sowie die Klimapolitik. Das sind drei zentrale Themen, die uns in den nächsten Jahren nachhaltig beschäftigen werden. Mit Beatrice Bünter steht eine Kandidatin zur Wahl, die nicht nur über

Jugendförderung spricht, sondern diese auch aktiv betreibt. Beim FC Altdorf stand sie mit anderen seit Jahren in der Verantwortung bei der Jugendförderung.

Sie ist initiativ, immer für Neues offen, unterstützt und kann anderen sehr gut zuhören. Chancengleichheit in Schule, Ausbildung, wie auch im Beruf sind Voraussetzungen für das künftige Leben. Genügend Lehrstellen statt

Leerstellen heisst ihr Leitsatz. Ebenso wichtig ist ihr ein gut funktionierender öffentlicher Verkehr mit einem dem Markt angepassten Angebot.

«Abonnemente und Einzelbillette sind im Kanton Uri, im Verhältnis zu anderen Orten, eher zu teuer.»

BRUNO GISLER, ALTDORF

Ein weiteres zentrales Anliegen ist ihr, dass auch im Kanton Uri der Tarifverbund eine Chance hat. Abonnemente und Einzelbillette sind im Kanton Uri, im Verhältnis zu anderen Orten, eher zu teuer. Zumindest für die Abonnemente müsste ein Verbund

möglich sein. Dies würde erst noch Familien mit Kindern finanziell entlasten und dank dem Umsteigeeffekt – Erfahrung aus anderen Gegenden mit Tarifverbänden – einen Beitrag zu weniger CO₂-Ausstoss leisten.

Mit Beatrice Bünter als Landrätin bin ich überzeugt, dass diese wichtigen Themen eine engagierte und lösungsorientierte Verfechterin erhalten. Deshalb gebe ich ihr die Stimme auf der SP-Liste der Gemeinde Altdorf.

BRUNO GISLER,
WELTIGASSE 8, ALTDORF

Vom Geissenhirt in Realp zum Ständerat

Bauen | Altständerat Oswald Ziegler feiert heute seinen 75. Geburtstag

Dem beschwerlichen Schulweg von Bauen ins Kollegi nach Altdorf folgte eine erfolgreiche Karriere als Anwalt, Politiker und Militär.

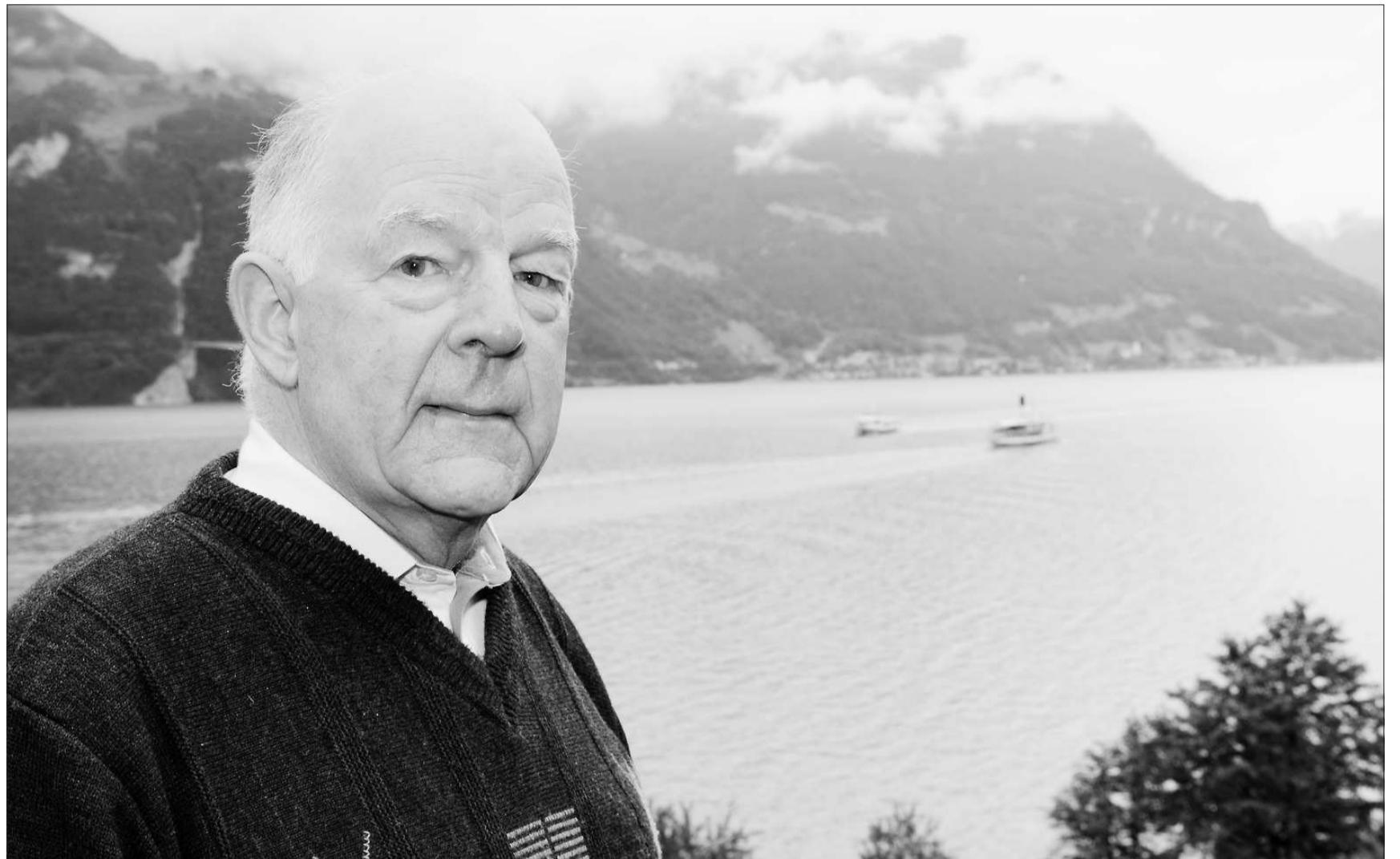
Fast so tief wie der Urnersee sind die Wurzeln von Oswald Ziegler in Bauen. Er kam am 21. Juni 1933 als sechstes von 14 Kindern, 5 Brüdern und 8 Schwestern, als Sohn des Johann und der Josefa Ziegler-Jauch auf die Welt. Er ist rund 15 Minuten oberhalb des Dorfes auf einem Bauernhof aufgewachsen. Schon als Primarschüler wurde er während der Sommermonate nach Realp als Geissbub geschickt. Dort hielt der Bauer den jungen Knecht im Herbst so lange zurück, dass er einen Monat zu spät in die Schule Bauen einrückte, sodass sein Vater eine Busse bezahlen musste, die grösser war als der Hirtlohn.

Mit dem Velo nach Altdorf

In Realp lernte er für sein späteres Leben Selbstbewusstsein, eigenständiges Verhalten, Verantwortung. Nach der obligatorischen Primarschule wollte er unbedingt ins Gymnasium nach Altdorf. Der beschwerliche Schulweg wurde zu Fuss, mit dem Schiff und dem Fahrrad bewältigt. Nach dem Studienabschluss der Rechtswissenschaften an der Uni Fribourg machte der frisch lizenzierte die obligaten Praktika beim Landgericht Uri und beim Rechtsanwalt und Notar Franz Muheim in Altdorf. Mit seinem Kollegen Franz Muheim und später mit dessen Sohn Franz Xaver Muheim, führte er bis 2000 eine gemeinsame Anwaltspraxis. 1965 heiratete er die junge Lehrerin Wilfrieda Wipfli. Die beiden Töchter Patricia und Carla traten später in die beruflichen Fussstapfen der Mutter und wurden auch Lehrerinnen.

Vielfältige politische Karriere

Bereits als Student und später als Anwalt interessierte ihn die Politik. Nach 10-jähriger Amtszeit als Gemeinde-



Der Jubilar und seine tägliche Aussicht: Als Kollegischüler in Altdorf ist Oswald Ziegler früher jeden Tag mit dem Schiff gefahren.

FOTO: DANIEL REGLI

schreiber der Gemeinde Bauen diente er der Gemeinde Bauen während acht Jahren als Gemeindepräsident. Spannend erlebte er die zwölf Jahre im Urner Landrat, dem er 1984/85 vorstand. Acht Jahre als Ständerat und Mitglied vieler Kommissionen öffneten den Blick in die Bundes- und Weltpolitik, besonders als Präsident der Sicherheitspolitischen Kommission und als Mitglied der Finanzkommission. Es würde den Rahmen sprengen, hier alle seine Mandate im Kanton Uri aufzuzählen. Für drei davon

setzte er aber viel Herzblut ein. Der öffentliche Verkehr war ihm stets ein grosses Anliegen. Unter seinem 31-jährigen Präsidium entwickelte sich die Auto AG Uri von einem kleinen Betrieb zu einem grossen Unternehmen mit grosser kantonaler Bedeutung. Ebenso freute er sich als Präsident der Treib-Seelisberg-Bahn an deren Entwicklung. Als Präsident der Urner Kantonalbank war er am Aufschwung der Bank wesentlich beteiligt und war Präsident der Baukommission des Bankneubaues.

Heimat liegt ihm am Herzen

Trotz Rückzug aus aktiver Politik und Beruf ist das Interesse daran nie verloren gegangen. Tägliche Zeitungslektüre und Tagesschau halten ihn über globale, kantonale und kommunale Politik auf dem Laufenden. Seine Heimatgemeinde und deren Probleme liegen ihm natürlich sehr am Herzen – seit Jahren sind der Dorfladen und die Post verwaist, zwei Gastrobetriebe stehen zum Verkauf, die Schule wird nach Seedorf ausgelagert, die Schuldenlast drückt, die Suche nach Behördenmitgliedern scheint

aussichtslos – grosse Probleme einer kleinen Randgemeinde. Patentlösungen kann er keine aufzeigen, wohnt aber trotzdem gerne in seinem schönen Heimort. Er erfreut sich an der wunderbaren Aussicht auf Berge und See, wandert gerne mit seiner Frau oder begleitet sie als «Mappenschlepper» zu ihren Orgeldiensten in Bauen und Silenen. Im Sinne von Oswald Ziegler wünschen wir seiner Heimat- und Wohn-gemeinde einen neuen Aufschwung. Ihm gönnen wir noch viele glückliche Jahre in Bauen. (UW)

Mit Begeisterung und Akribie am Werk

Berufsmatura | Urner machen national am Robot Team Challenge mit

Sechs Berufsmaturanden haben rund 350 Freizeitstunden für ein ganz besonderes Vorhaben investiert, einen speziellen Getränkeautomaten.

Ruedi Ammann

Eigentlich hätte dieses Jahr an der traditionellen Ausstellung der Lehrabschlussarbeiten – präsentiert am Donnerstag und Freitag dieser Woche – die Coiffeursparte zum besonderen Aushängeschild werden sollen. Doch weil dieses Jahr schweizweit generell dem Jahr der Informatik das spezielle Augenmerk gilt, sahen sich die Verantwortlichen der Berufsschule dazu veranlasst, die «Figaros» auf 2009 zu verfrachten.

Abwechslung im Berufsalltag

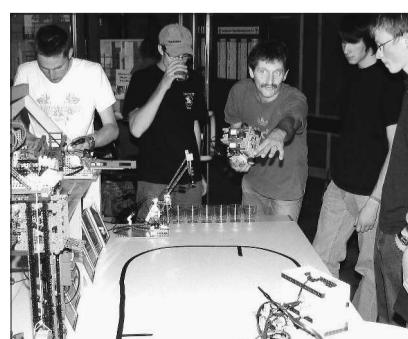
431 Urner Lernende schliessen in diesen Wochen einen ganz besonderen Lebensabschnitt ab. Unter diese Zahl fallen auch die Berufsmaturanden. Sechs junge Männer der technischen Sparte hatten sich zusätzlich zu den obligaten Zielsetzungen eine Teilnahme an einem schweizweit lancierten Projekt, dem Robot Team Challenge, aufs Banner geschrieben. Zwar war der Wettbewerb eigentlich für Mittelschülerinnen und Mittelschüler ausgeschrieben worden, doch flatterte das entsprechende «Placet» bald auch ins Berufsschulhaus. «Wir wollten ganz einfach etwas tun, das Abwechslung in den Alltag bringt», erklärt



Das «Jung-Ingenieure-Sextett»: (hinten, von links) Simon Gisler, Jost Furrer und Adrian Steiner; (vorne, von links) Manfred Arnold, Sandro Arnold und Nicolas Stocker.

FOTOS: RUEDI AMMANN

Manfred Arnold, Attinghausen, anlässlich der Vorstellung der Arbeit im Ausstellungsraum des kantonalen Berufsschulhauses gegenüber dem «Urner Wochenblatt». Das rührige Sextett setzte eine Idee zum Thema «Getränkeautomat» (Lego-Mindstorms) in die Tat um. Bevor allerdings zur Ausführung geschritten werden konnte, wurde das Vorhaben von einer Zürcher Jury abgesegnnet. Die Realisierung lief im April dieses Jahres an. Bisher haben die sechs Berufsmaturanden, unter aktiver Beihilfe von Fachlehrer Franz Philipp, in die inte-



Fachlehrer Franz Philipp (Mitte) hat das Projekt «Getränkeautomat» mit Herzblut unterstützt.

ressante Arbeit rund 350 Stunden investiert. In einem ersten Teil wurde die Erstellung einer Homepage, in einem zweiten die Ausführung realisiert. Nun verbleibt noch die recht zeitaufwendige Aufgabe, ein sogenanntes Script (also einen Projektbeschreibung) zu verfassen.

Entscheid fällt in Zürich

Die Projektarbeit der Urner Berufsmaturanden muss bis Ende August der Jury in Zürich eingereicht werden. Diese wird dann bewertet. «Für uns ist die Bewertung zweitrangig. Wichtig für unsere Gruppe ist es ganz einfach, im Programmieren und Konstruieren zusätzliche Erfahrungen zu sammeln», betont Manfred Arnold, nicht ohne zu erwähnen, dass die Umsetzung des Vorhabens den Gruppenzusammenhalt enorm gestärkt hat. Die Preisvergabe findet am Donnerstag, 11. September, in Zürich statt.

Fachlehrer Franz Philipp würdigte an der Ausstellung die Arbeit seiner Schützlinge: «Die Berufsschüler haben sehr gut zusammengearbeitet, und das sowohl während der Schul- als auch während der Freizeit. Dazu galt es für alle, sich auch mit den eigentlichen anstrengenden Diplomprüfungsarbeiten zu befassen.» Zwei Details dazu: Die Arbeitsgruppe der Berufsschule Uri erhielt als Beihilfe für den Bau ihres Getränkeautomaten einen Lego-Robotik-Baukasten im Wert von 400 Franken. Am Wettbewerb beteiligen sich 30 Teams aus zahlreichen Kantonen.

Truppenübung im Kanton Uri

Die Volltruppenübung des Gebirgsinfanteriebataillons 29 wird kommende Woche in den Kantonen Uri und Schwyz viel Verkehr generieren. Die Übung, bei der das ganze Bataillon mit rund 1200 Soldaten im Einsatz steht, beginnt am Montag, 23. Juni, und endet am Mittwoch, 25. Juni, mit der Fahnenabgabe auf dem Hauptplatz in Schwyz. Da sich der Hauptteil der Übung im Kanton Uri abspielen wird, jedoch vier der sechs Kompanien im Kanton Schwyz ihre Unterkunft haben, wird insbesondere die Axenstrasse am Montag in Richtung Flüelen und am Mittwochnachmittag in der Gegenrichtung viel Verkehr durch die Radschützenpanzer und anderen Militärfahrzeuge erleben, heisst es in einer Medienmitteilung. Das Geb Inf Bat 29 entschuldigt sich schon im Vorfeld für eventuelle Störungen und Wartezeiten und dankt für die Rücksichtnahme. (UW)

KANTON URI

Pensionierung

Ende Juni wird Franz Senn, Amt für Betrieb Nationalstrassen, nach 27 ½ Jahren vorzeitig aus der Kantonsverwaltung austreten. Der Regierungsrat dankt dem Mitarbeiter für seine langjährige Arbeit im Dienste des Kantons Uri und wünscht ihm für die Zukunft alles Gute. (UW)

Franz Philipp

Von: Sarah Carter/ EducaTec AG [sarah.carter@bluewin.ch]
Gesendet: Dienstag, 22. Juli 2008 11:40
An: 'Franz Philipp'
Betreff: NXT Barkeeper
Kennzeichnung: Zur Nachverfolgung
Kennzeichnungsstatus: Rot

Guten Tag Herr Philipp

Ich schreibe betreffend den NXT Barkeeper ihrer Schüler, den wir gerne an der Worlddidac ausstellen würden.

Was sind seine Dimensionen?

Und was brauchen Sie, um ihn am Stand aufzustellen?

Vielen Danke für die Informationen!

Freundliche Grüsse,
Sarah Carter

EducaTec AG

Kanzleigasse 2 - Altes Schulhaus

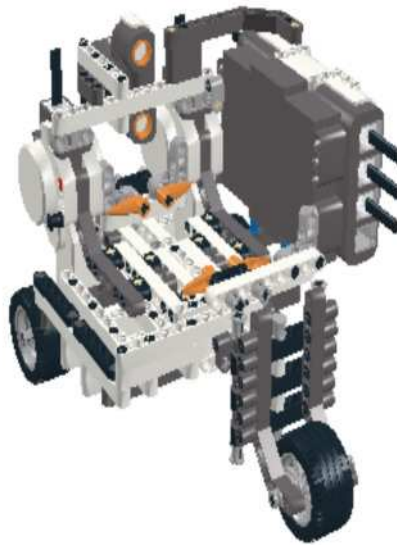
CH-5312 Döttingen

Schweiz - Suisse - Switzerland

<http://www.educatec.ch> / sarah.carter@educatec.ch

Tel: +41 56 245 8161 - Fax: +41 56 245 8163

Model Name: **Wagen**
Number of Bricks: **249**



Step 1

1 x



Step 2

Step 3



Step 4



Step 5



Step 6



Step 7



Step 8



Step 9



Step 10

1 x



Step 11

1 x



1 x



1 x



Step 12



Step 13

2 x



Step 14



Step 15



Step 16



Step 17

2 x



Step 18



Step 19

1 x



2 x



2 x



Step 20



Step 21

2 x



Step 22

1 x



2 x



Step 23



Step 24



Step 25



Step 26



Step 27

1 x



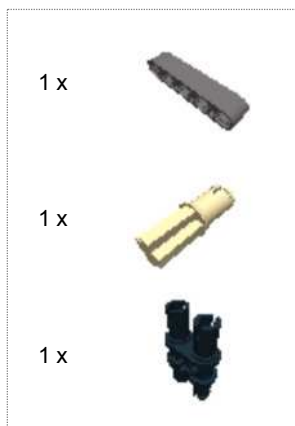
Step 28



Step 29



Step 30



Step 31



Step 32



Step 33



Step 34

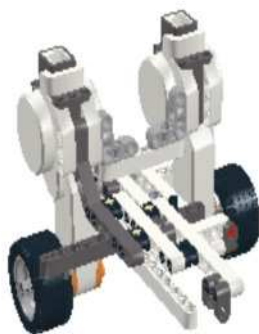


1 x



Step 35

1 x



Step 36

1 x



2 x



1 x



1 x



Step 37

1 x



1 x



Step 38



Step 39

1 x



1 x



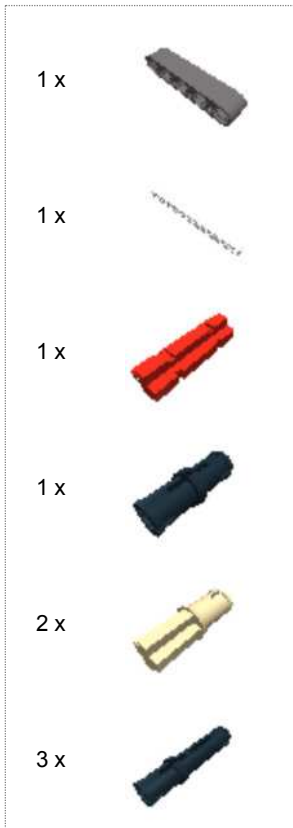
2 x



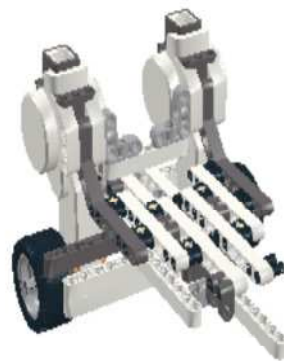
Step 40



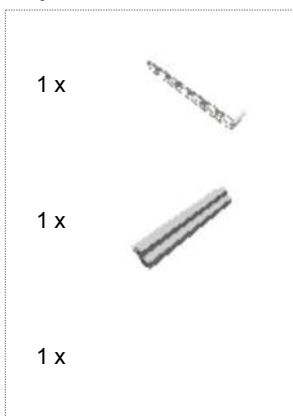
Step 41



Step 42

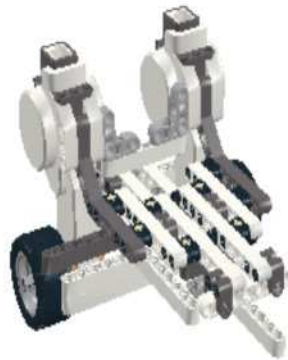


Step 43

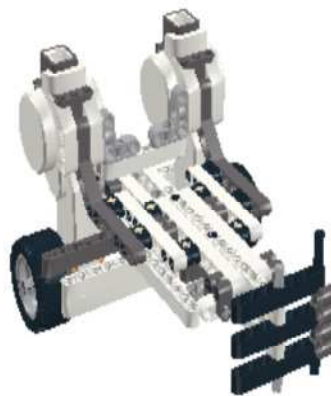
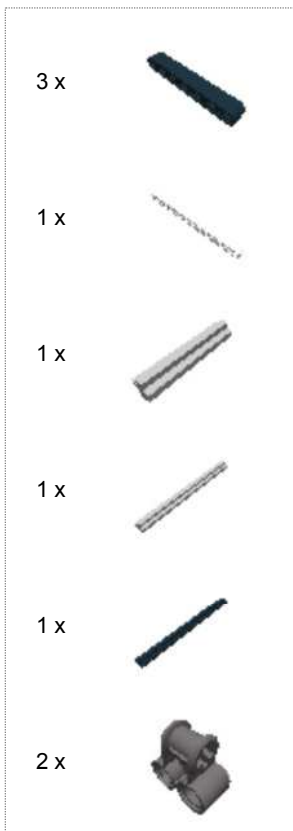




Step 44

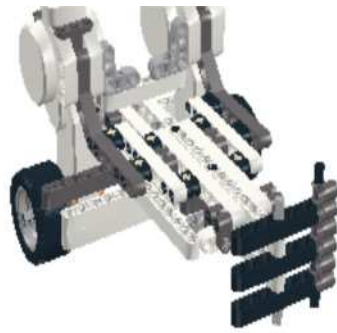


Step 45

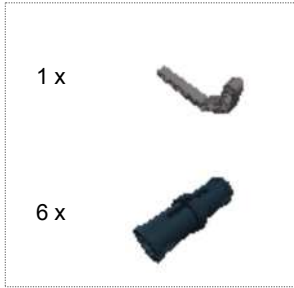


Step 46

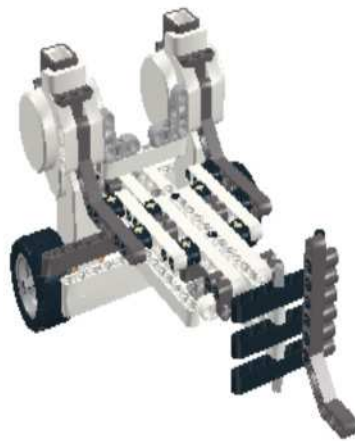




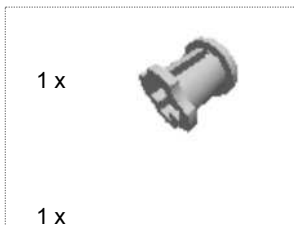
Step 47

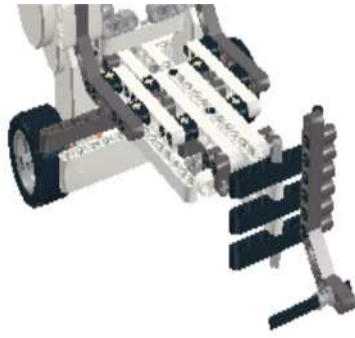


Step 48



Step 49

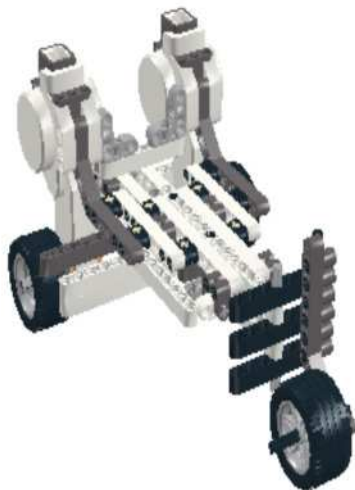




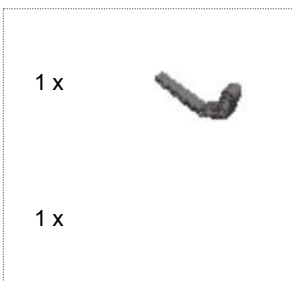
Step 50

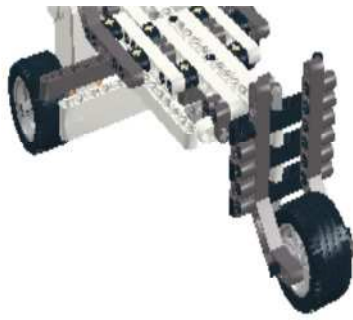


Step 51

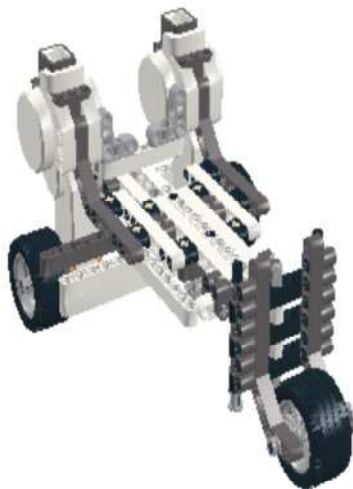


Step 52

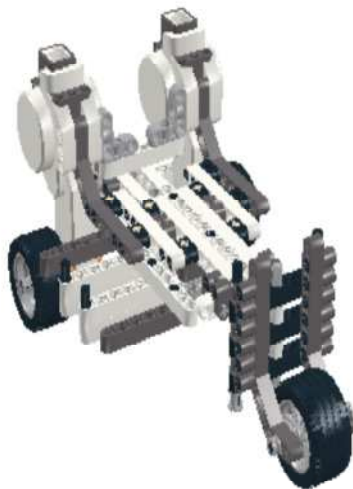




Step 53

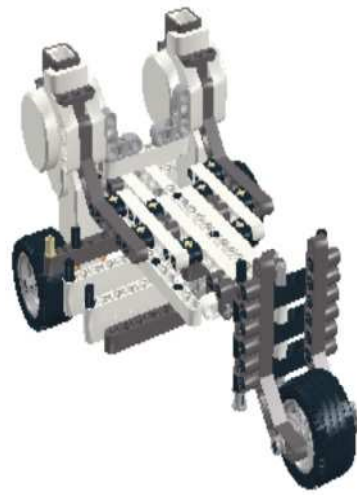


Step 54

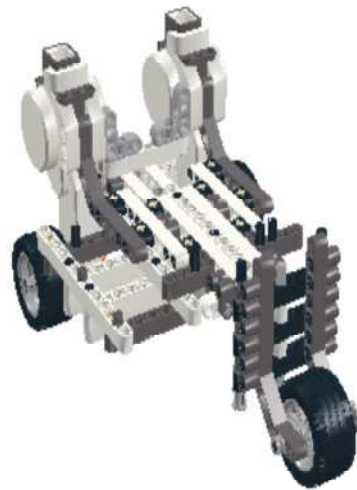


Step 55

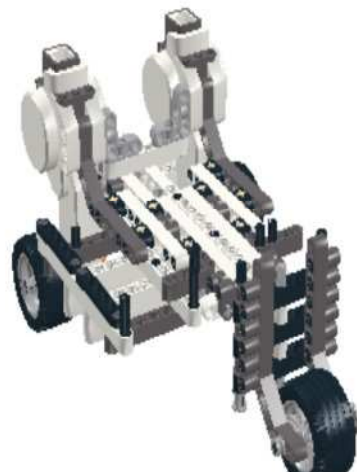




Step 56



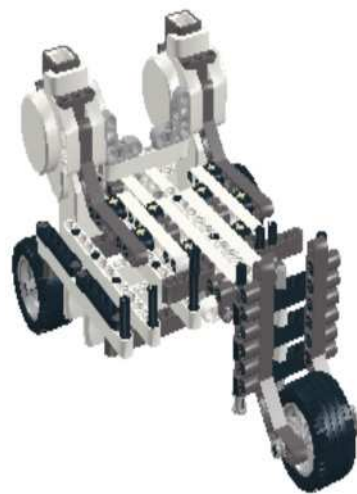
Step 57



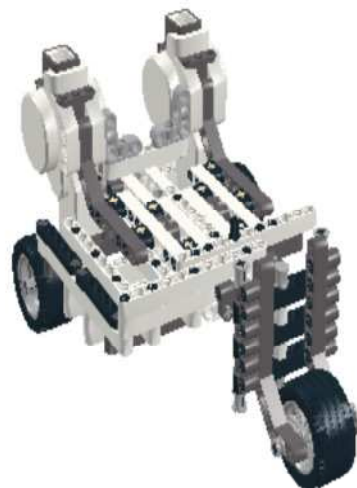
Step 58



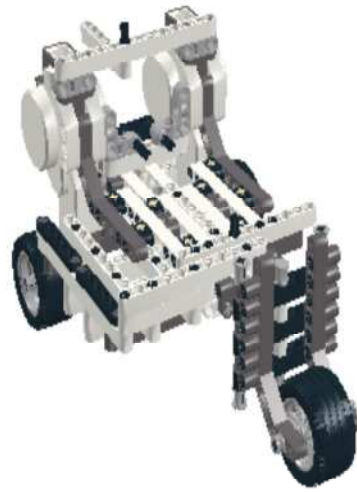
Step 59



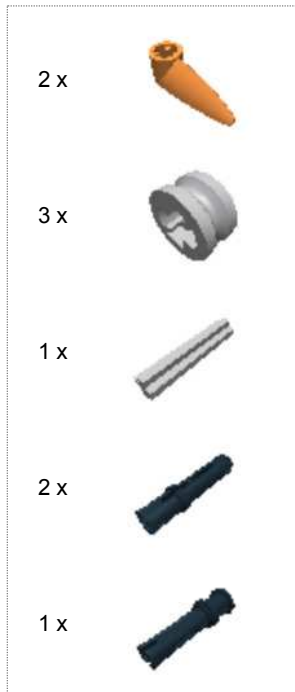
Step 60



Step 61



Step 62



Step 63





Step 64



Step 65



Step 66





Step 67

1 x



Step 68

1 x



2 x



Step 69





Step 70



Step 71



Step 72

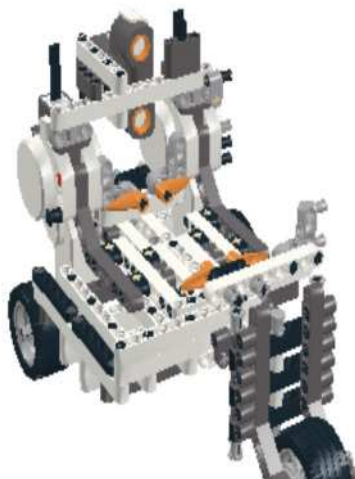




Step 73

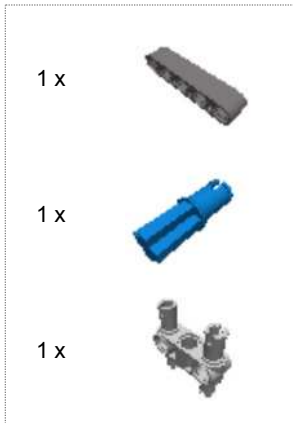


Step 74

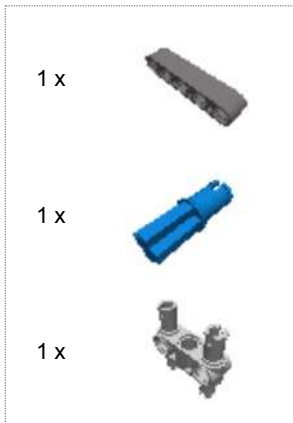




Step 75



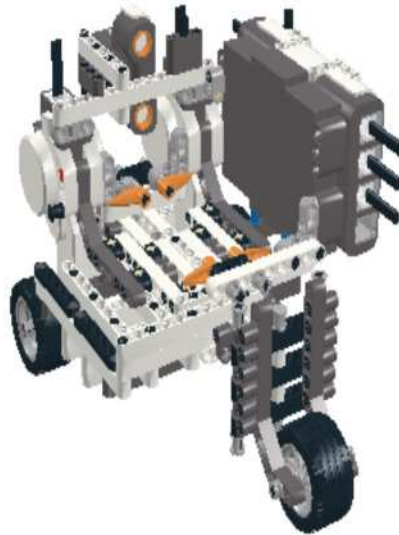
Step 76



Step 77

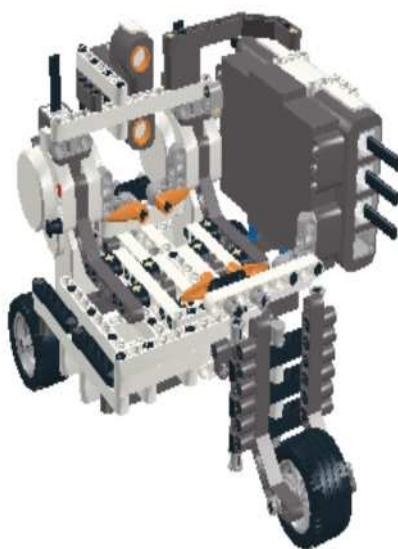


Step 78

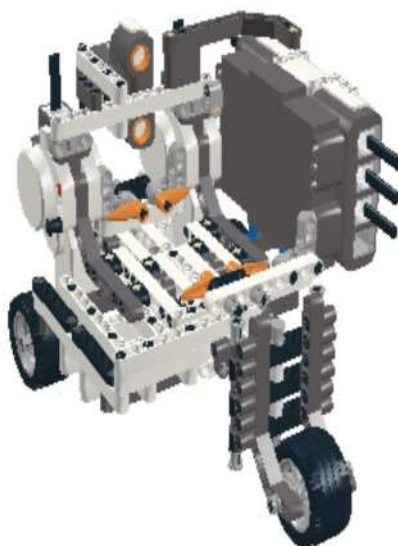
Step 79

Step 80

Step 81



Step 82



[LEGO Digital designer website](#)

© 2007 The LEGO Group. All rights reserved.
Use of the LEGO Digital Designer html print tool, signifies your agreement to the terms of use.