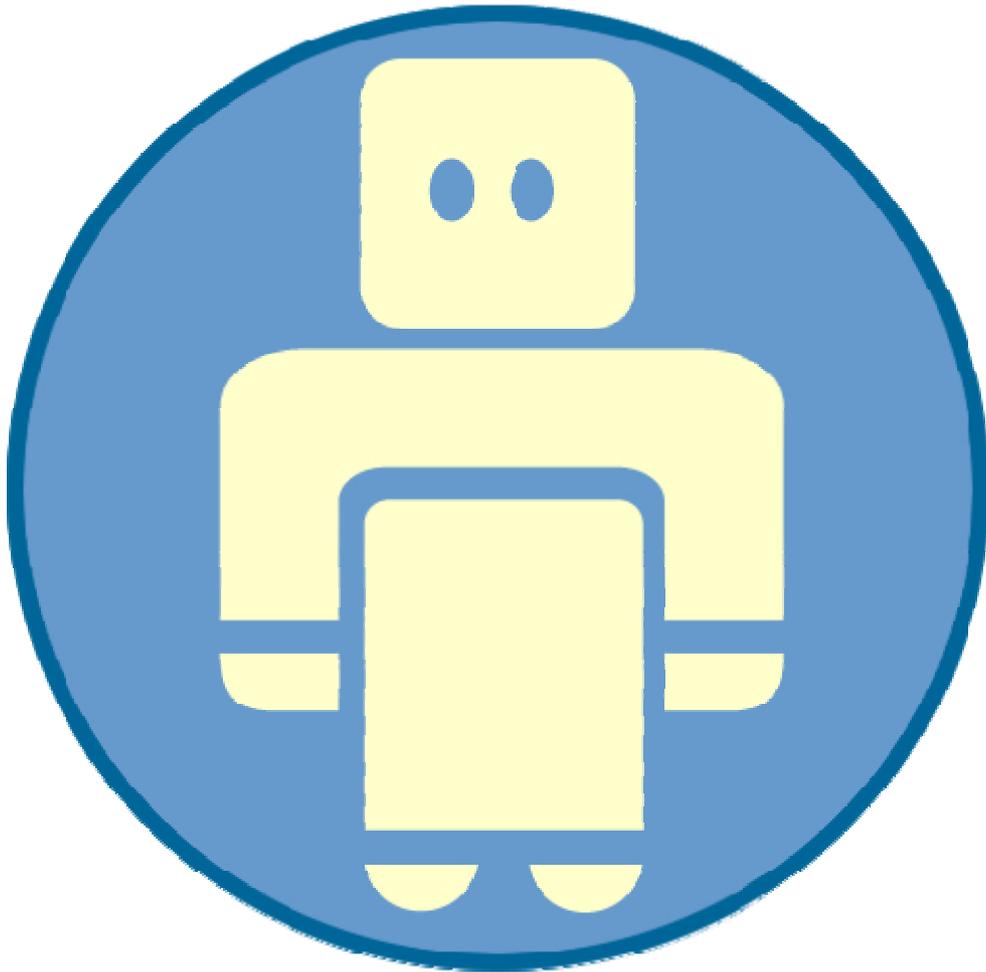


Robotik



Interview mit dem Robotik-Team

Der Robotik Kurs der Science Academy 2005, ein Kurs voller begeisterter Schüler und Schülerinnen aus ganzen Baden Württemberg, beschäftigte sich während der 2 Wochen intensiv mit Bauen von Programmieren von Robotern. Nun konnten wir alle Teilnehmer zu einem Interview einladen.

Hier ein Auszug aus dem aktuellen *Robotics and Education Journal* (übersetzt von Jessica Jing Guo).

RaEJ: Herzlich Willkommen, danke, dass ihr euch Zeit nehmt. Also, stellt euch doch mal vor.

Kurs Robotik: Unser Team besteht aus elf Teilnehmer: Ronja Berg, Annkatrin Staiber, Georges Blattner, Andreas Gut, Philipp Gutknecht, Florian Kim Dang Huynh, Finn Lanzendorfer, Carsten Plasberg, Thomas Nutz, Arne Schauf und Max Wolf.

RaEJ: Wie alt seid ihr?

Ronja: Wir sind zwischen 14 und 16 Jahre alt.

Philipp: Ist ja auch eine Mittelstufen-Akademie. *lach*

RaEJ: Warum habt ihr den Robotik Kurs ausgewählt?

Finn: Ich habe selbst oft mit LEGO gespielt!!

Georges: Ich wollte mich mal allgemein über Programmieren informieren.

Max: Hm...gute Frage...na ja, weil es der beste Kurs von Allen ist!

(Kurze Lachpause, einstimmiges Nicken.)

Florian: Roboter werden sicherlich in der Zukunft eine große Rolle spielen. Deshalb ist es interessant auch selber einen Roboter zu bauen und alles mal auszuprobieren.

RaEJ: Was macht ihr denn so in eurer Freizeit?

Annkatrin: Ich spiele sehr gerne Geige. Aber ich denke, dass wir alle ohne unseren PC nicht leben könnten.

Carsten: Absolut unvorstellbar! Außerdem lese ich oft und viel. Segeln und Radfahren macht mir natürlich auch Spaß.

Thomas: Viele von spielen auch im Verein Fußball, Andi und ich zum Beispiel.

Arne: Ich spiele Volleyball.

RaEJ: Was wären die drei Dinge, die ihr auf eine einsame Insel mitnehmen würdet?

Philipp: Eine Hängematte wäre praktisch, kann man sich schön entspannen. Und was zu essen und trinken.

Thomas: Ich würde ein Boot zum Abhauen mitnehmen und ein Funkgerät.

Finn: Ohne mein Taschenmesser würde ich nirgendwohin gehen.

RaEJ: Könntet ihr euch in kurzen Worten beschreiben?

Georges: Andi ist unser super Konstrukteur, der baut den besten Roboter, da kann ihn keiner überbieten. Und Max kann unglaublich schnell tippen und hat sogar einen Preis als SuperSpammer bekommen, weil er in unserem Forum die meisten Beiträge hatte (ca.1200).

Andreas: Ja, aber du machst ständig Werbung für Nike (er trägt fast nur Nike Sachen).

Finn: Und Philipp ist der Computerfreak unter uns. Man bekommt ihn einfach nicht mehr vom Bildschirm weg.

Ronja: Alle sind super lieb, wir waren echt eine coole Gruppe.

Annkatrin: Stimmt, hilfsbereit und freundliche Leute haben wir in der Gruppe.

Phillip: Und Carsten ist eine Kichererbse.

RaEJ: Warum denn das?

Thomas: Na ja, wenn er mal anfängt zu lachen, hört er gar nicht mehr auf!!

(Carsten lacht, wiedermal...)

Max: Und natürlich dürfen wir nicht unsere Leiter und Schülermentorinnen nicht vergessen. **Matthias Taulien** und **Georg Wilke** leiten zum dritten Mal den Robotik Kurs in der Science Academy. Sie sind beide in den Hector Seminar tätig und Lehrer an Gymnasien.

Jessica Jing Guo und **Rebecca Zelt**, die letztes Jahr ebenfalls Teilnehmerinnen waren, unterstützten die Arbeit in den 2 Wochen.

Thomas: Es waren zwei erfolgreiche Wochen. Die Atmosphäre war einfach einmalig. Die Leiter waren stets hilfsbereit und freundlich. Mit ihre Unterstützung haben wir viel geschafft.

RaEJ: Ich bedanke mich für dieses tolle Interview.

Nun, liebe Leser, werden wir Ihnen einen Einblick in das Kursleben verschaffen und die Ergebnisse präsentieren.

Das Team



Der Wochenablauf

Von Georges Blattner & Max Wolf

26.8.05

Freitag, der 26. August 2005, unser erster Tag zurück in Adelsheim. Dieses wird uns nun 2 Wochen lang als Zuhause dienen. Schon heute beginnen auch unsere Kurstreffen. Zuerst ganz harmlos, mit Tischeschleppen und Nachbauen von Robotern nach Anleitung. Zwar wollen wir eigentlich gleich eigene Roboter entwickeln, zu Übungszwecken ist es jedoch besser, wenn alle mit demselben Roboter arbeiten. Außerdem lernen wir unsere Schülermentorinnen Rebecca Zelt und Jessica Guo kennen.

27.8.05

An unserem ersten, ganz „normalen“ Tag in Adelsheim, finden natürlich, wie an jedem anderen Tag auch, zwei Kurstreffen statt. Zur Auffrischung unserer Javakenntnisse befassen wir uns wieder einmal mit den Grundlagen von Java.

Dabei teilen wir uns zum ersten Mal in verschiedene Gruppen auf um effektiver Arbeiten zu können. Da die Fischertechnik-Grundprogramme wegen eines Versehens Fehler aufweisen, ist auch die Gruppe, die sich ursprünglich mit Fischertechnik befassen wollte, dazu gezwungen, mit Lego zu arbeiten.

28.8.05

An unserem 2. Tag beginnen wir mit etwas schwierigeren Übungsaufgaben. So versuchen wir z.B. einen Roboter dazu zu bewegen einer schwarzen Linie zu folgen. Hierbei benutzen wir zum ersten Mal auch Sensoren, wie Tastsensoren oder Lichtsensoren und bauen eigene Roboter, also keine Roboter nach Anleitung. Am späten Nachmittag haben wir auch diese Übungsaufgabe endlich geschafft: Juhu!

29.8.05

Wir kennen uns nun schon mit den einzelnen Sensoren und auch mit fortgeschrittenen Programmier-techniken aus. Diese werden im Kapitel *Java* genauer erläutert. Jetzt können wir ein neues Projekt beginnen: den *Rotation-Navigator*. Diese *Java-Klasse* errechnet aus den Werten zweier Rotationssensoren, dem Radabstand und dem Raddurchmesser die aktuelle Position des Roboters und überträgt diese in ein Koordinatensystem. Mit dessen Hilfe können Positionen gespeichert und auch wieder abgerufen/angefahren werden. Langsam beginnt auch das vorher erwähnte Fischertechnik-Grundprogramm, an welchem

Herr Taulien und Herr Wilke schreiben, eine feste Form anzunehmen.



Die ersten Schritte zum Roboter

30.8.05

Das Fernsehen ist da! Diese Tatsache, von einigen sehr, von den anderen weniger freudig aufgenommen, behindert uns leider bei unserer Arbeit: so müssen wir z.B. auf Grund schlechter Lichtverhältnisse ein und denselben, nicht funktionierenden Roboter 10 mal fahren lassen oder wir kommen mit einem „imaginären“ Problem zu Herrn Taulien, der uns daraufhin die Sachlage erklärt. Gott-sei-Dank interessiert sich das Fernsehen nur für die Roboter und nicht für deren Programmierung, so dass man im Nebenraum weitgehend

ungestört arbeiten kann. Leider müssen wir erfahren, dass sich zwei bestimmte Programmkomponenten, der *RotationNavigator* und die *Behaviors* nicht miteinander kombinieren lassen.

Zur Erklärung: *Behaviors* sind ebenfalls *Java-Klassen*, die den Ablauf der einzelnen Aktionen des Roboters regeln.



Jean Claude

1.9.05

Obwohl das Fernsehen noch nicht abgereist ist, lässt es uns an diesem Tag weitestgehend in Frieden. Außerdem hat Herr Wilke eine Lösung gefunden, wie wir den *RotationNavigator* doch zusammen mit den *Behaviors* einsetzen können. Auch bei der Programmierung der Roboter kommen wir gut voran. Nachdem wir feststellen müssen, dass der *RotationNavigator* nicht mit Kettenantrieb funktioniert, bauen wir die Lego Roboter um. Auch bei der Kommunikation zwischen den Lego-Robotern machen wir große Fortschritte.

2.9.05

An diesem Morgen arbeiten wir erst einmal wieder an den Robotern und verbessern die Programme. Den restlichen Tag bereiten wir noch die Präsentation für die bevorstehende Rotation in den von uns aufgestellten, kursinternen Gruppen vor.

3.9.05

Nun ist es soweit, der Rotationstag ist gekommen. Die Gruppen der einzelnen Kurse, werden in einen Zeitplan eingeteilt und halten jeweils einen Vortrag. Wer nicht gerade dabei ist, einen Vortrag zu halten, kann sich über die Fortschritte der anderen Kurse informieren. Im Großen und Ganzen werden die Vorträge, trotz einiger Pannen, sehr gut vom Publikum

aufgenommen. Nach dem Stress am Vormittag, können wir uns dann nachmittags bei den Highland-Games und abends beim Bergfest austoben.

4.9.05

Nach dem ereignisreichen Vortag ist es an der Zeit sich wieder voll und ganz auf das gesetzte Ziel zu konzentrieren und die Roboter funktionsstüchtig zu machen.

Große Probleme macht immer noch die Tatsache, dass der Roboter nicht an die vorgesehene Position fährt. Dies ist auf Reibung und Messungenauigkeiten zurückzuführen. Aufgrund dessen lassen dich die Aufgaben (Dosen suchen, Dosen sammeln, Dosen greifen) der einzelnen Roboter nicht miteinander kombinieren.

5.9.05

An diesem Tag gehen wir nach Heidelberg. Wir besuchen das deutsche Krebsforschungszentrum und anschließend ABB. Dort bekommen wir Industrieroboter zu sehen und erhalten einen Einblick in das Produktsortiment der Weltfirma.

6.9.05

Nur noch ein Tag ist übrig, dann ist Präsentationstag und es gibt dementsprechend viel zu tun. Bis zur letzten Minute arbeiten wir fieber

haft daran, die einzelnen Arbeitsschritte der Roboter miteinander zu verbinden. Außerdem üben wir wieder einmal die Präsentation und fertigen Plakate an. Am Abend findet dann schließlich das Abschlusskonzert statt.

7.9.05



Der erste Test

Heute ist der Tag, auf den alle Kurse hingearbeitet haben. Wie bei der Rotation sind wir in Gruppen eingeteilt, die jeweils zu anderen Zeiten ihre Vorträge halten. Gegen 14 Uhr treffen die ersten Eltern ein und die Nervosität steigt in Anbetracht der näherrückenden Präsentation. Um 15 Uhr geht es los: Die Vorträge sind gelungen, auch wenn sich manche Roboter bei der Vorführung selbstständig machen. Auch die Vorträge der anderen Kurse sind interessant und es bleibt genügend Zeit diese zu besuchen. Nach der Präsentation und

dem Essen mit den Eltern versammeln sich alle zum Abschlussabend. Dieser beinhaltet ein paar Gesangsdarbietungen, eine Theateraufführung, Reden und Danksagungen sowie einen abschließenden Syrtaki.

8.9.05

Der letzte Tag der Science-Academy bricht an. Wir räumen auf und säubern die Zimmer, bevor wir uns dann alle unter Tränen verabschieden und schließlich schweren Herzens nach Hause fahren.

Nun werden Sie sicher mit einer Menge Begriffen konfrontiert worden sein, die wir Ihnen auf den nächsten Seiten näher bringen möchten.

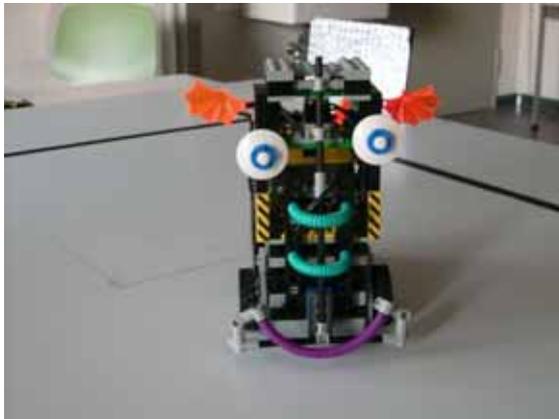


Wo steckt der Fehler?

Unser Ziel

Von Ronja Berg

Am Anfang hatten wir lauter verrückte Ideen, von Robotern die wir bauen wollten: Vorhang hinaufklettern, Schreibtisch aufräumen... Doch schon nach kurzer Zeit, als wir einige Erfahrungen mit dem Bauen und Programmieren gesammelt hatten, wussten wir was einigermaßen umsetzbar war. Wir haben uns in drei Gruppen aufgeteilt. Zwei von ihnen haben Roboter aus Lego und die dritte einen Fischertechnikroboter gebaut. Wir haben uns viel vorgenommen und ein Großteil davon auch umsetzen können.



Suchroboter Jean-Claude

Jean Claude, der erste Roboter, sucht auf einem Feld, das durch eine schwarze Linie markiert ist, nach Tonnen und merkt sich deren Position. Dies geschieht dadurch, dass die Rotationssensoren die Umdrehungen der Räder zählen, kann der RCX ausrechnen, wo sich der Roboter in einem Koordinatensystem befindet.

Immer wenn er an der schwarzen Linie angekommen ist (dies merkt er durch seine Lichtsensoren), wendet er und sucht etwas versetzt weiter. Das Ende ist markiert durch ein aufgeklebtes Stück Alufolie. Dieses erkennt er ebenfalls durch seine Lichtsensoren und es ist für ihn das Zeichen, dass er wieder zu seinem Anfangspunkt zurückfahren soll.

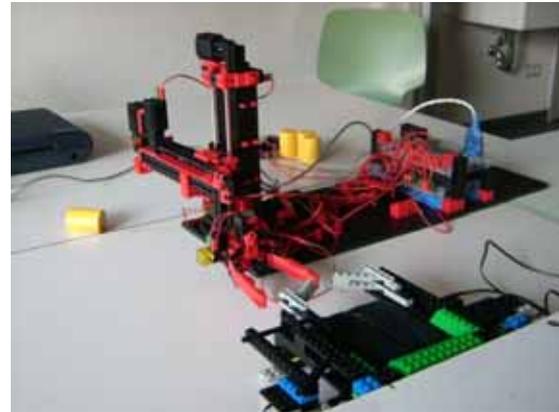
Die erste große Schwierigkeit bestand darin, dass Jean Claude mit Sammy, dem 2. Roboter, kommunizieren sollte um ihm die Koordinate der Tonne zu übermitteln. Dazu haben wir an seinem Anfangspunkt die mit einer Alufolie überzogen ist. Jean Claude sendet die Daten der Tonne per Infrarot gegen die Wand, und von dort werden sie zu dem zweiten Roboter, Sammy reflektiert. Da dieser aber nicht genau auf der gleichen Stelle steht, wie Jean Claude, hat er ein anderes Koordinatensys

tem. Das heißt, wir mussten sein Koordinatensystem etwas versetzen, damit er auch wirklich zu der Koordinate fährt, wo sich die Tonne befindet.



Sammelroboter Sammy

Sammy fährt zu der ihm übermittelten und versetzten Koordinate und schließt seine Greifarme, sobald der Drucksensor von der Tonne gedrückt wurde. Er fährt dann zu einer festgelegten Stelle, an der ein Fließband ist, stellt die Tonne dort ab und fährt dann zur nächsten Tonne. Das Fließband befördert die Tonne so, dass sie aufrecht ankommt. Eigentlich sollte sie hier einen Drucksensor auslösen, der den Fischertechnik-Roboter Fischi Deluxe in Kraft setzt, aber dazu hat uns wieder die Zeit nicht ganz gereicht. Also haben wir das manuell gemacht. Der Fischi Deluxe ist eine stationärer Roboter, der aus einem Greifarm besteht.



Fischi Deluxe: Unser Fischertechnik-Roboter

Der Greifarm von Fischi Deluxe greift die Tonne, dreht sich mit ihr und sollte sie eigentlich in ein Regal stellen. Wir haben dann aber beschlossen, dass es reicht, wenn er sie über einer Tesafilmrolle fallen lässt.

Wir hätten das ganze zwar viel einfacher haben können, z. B. indem der Roboter, der die Platte absucht, eine gefundene Tonne gleich zu dem Fischertechnikroboter fährt, aber wir wollten mit der Kommunikation per Infrarot zwischen den beiden Robotern eine Herausforderung haben. Und ich finde wir haben sie für die kurzen 2 Wochen, die wir nur Zeit hatten wirklich gut gelöst. Auf die Umsetzung der geschilderten Aufgaben wir im Zusammenhang mit der Programmierung im nächsten Kapitel auf Grundlagenebene erneut eingegangen.

Java

Von Thomas Nutz

Ein Roboter muss sich zwei elementaren Aufgaben widmen: Zum einen muss er eine funktionierende Maschine, bei uns aus Lego und Fischertechnik, bauen. Zum anderen braucht das Gerät noch eine Abfolge von Anweisungen, die sog. Choreographie. Auch muss ihm gesagt werden, wie es auf Veränderungen der Umgebung, die es ja mithilfe der Sensoren wahrnimmt, zu reagieren hat, um eine Aufgabe wunschgemäß zu erfüllen. Beide Aspekte sind Inhalt des Programms, dessen Erstellung wohl die meiste Zeit der Arbeit in Anspruch nimmt. Für unseren Kurs würde ich die Programmierung auch als eigentlichen Lerninhalt bezeichnen, da uns das Bauen von Lego- und Fischertechnik- Apparaten nicht an unsere intellektuellen Grenzen führen sollte.

Unser Kurs benutzte die Programmiersprache Java. Diese Programmiersprache weist für uns eine Reihe von Vorteilen gegenüber anderen Programmiersprachen wie z.B. die vom Kurs Pinball verwendete Sprache C++ auf:

Java basiert auf dem „Write once, Run anywhere“ Konzept. Dieses Konzept beinhaltet, dass der vom Programmierer verfasste Quelltext zuerst in einen sog. Bytecode übersetzt wird und dann von jedem Betriebssystem,

dass über eine Java-Umgebung verfügt, ausgeführt werden. Dadurch lassen sich mit Java unterschiedlichste Aufgaben bewältigen, z.B. auch die Programmierung von Handys etc.

Java ist sehr modern, d.h. nach den neuesten Erkenntnissen der Informatik aufgebaut. Dabei greift die Sprache auf bewährte Eigenschaften anderer Programmiersprachen zurück und verbindet diese mit einem hohen Maß an Sicherheit.

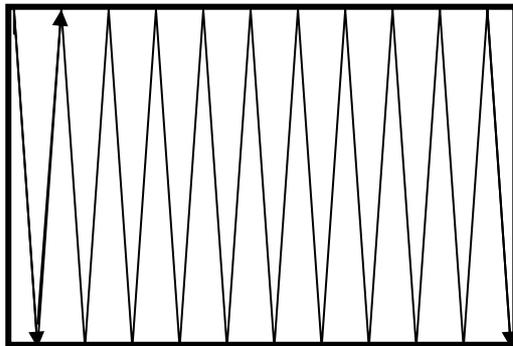
Java ist von ihrem Aufbau gesehen „die“ objektorientierte Programmiersprache.

„Objektorientierte Programmierung (OOP)“ ist ein wichtiges Konzept, das in der modernen Informatik eine tonangebende Rolle spielt. Es beinhaltet u.a. folgende Merkmale:

Programmteile sind „gekapselt“. Kapselung beschreibt die Art und Weise, wie auf Objekte zugegriffen werden kann. Wichtig ist dabei, dass man den Zugriff schon innerhalb des Objektes durch Signalwörter steuern kann und das sowohl der Programmierer als auch das zugreifende Programm über die Schnittstelle nur Informationen über die Leistung des Objekts, nicht aber über seine Funktionsweise bekommt.

Vererbung: Diese elementare Eigenschaft eines objektorientierten Programms, die ein Programm vielfältig wiederverwertbar macht und somit Quelltext spart, wird am anschaulichsten, wenn man sich aus dem Alltag ein Beispiel zu Hilfe nimmt: Unser Ausgangsprogramm sei eine Klasse „Pflanze“. Diese Klasse hat die Methoden (Vorgehensweisen) „Wachsen“ und „Verbreiten“. Sie benötigt auch einige Angaben wie Verbreitungsart oder Wuchshöhe, an denen Pflanzen unterschieden werden. Wollen wir nun eine Klasse „Baum“ schreiben, so können wir einfach „Baum“ von „Pflanze“ ableiten und haben damit schon die elementaren Eigenschaften und Vorgehensweisen. Um aus der Pflanze nun einen Baum werden zu lassen, fügt man nur noch Methoden wie „Blätter im Herbst abwerfen“ oder „Wurzeln bis ins Grundwasser treiben“ hinzu. An Eigenschaften wären noch z.B. Blätterform, Wuchsgeschwindigkeit oder Vorkommen prak-

tisch. Man sieht, nur noch wenig Quelltext unterscheidet „Baum“ von „Kartoffel“ oder „Seegrass“, da beide auf „Pflanze“ aufbauen. Selbstverständlich beinhaltet objektorientierte Programmierung noch weit mehr Eigenheiten und Merkmale, doch empfand ich diese beiden Eigenschaften für unsere Programmierarbeit am grundlegendsten und faszinierendsten. Doch damit wären wir schon relativ weit in die große Welt des Programmierens eingedrungen, im Folgenden möchte ich noch etwas näher auf die konkreten Aufgaben und erarbeiteten Lösungen unseres Kurses eingehen, exemplarisch ist dabei das Programm des Suchroboters genauer erläutert, da es eigentlich die interessantesten Aspekte der Robotik mit Java beinhaltet.



Weg des Sammelroboters



Hier war ein Stück Aluminiumfolie angebracht

Herausforderungen unserer Programmierarbeit

Von Thomas Nutz

Unser Ziel war, drei Roboter zu bauen, die auf einem mit schwarzen Linien umrandeten Feld Tonnen suchen, transportieren und verräumen können.

Für die erste Aufgabe, Tonnen suchen, musste also ein Programm erstellt werden, das einen Roboter dazu bringt, das Feld systematisch abzusuchen. Wir bedienten uns eines Lichtsensors, bei dessen Meldung „Schwarze Linie“ der Roboter um ca. 170° wendete und in dieser Richtung weiter fuhr. Dadurch hatten wir eine systematische Abarbeitung des gesamten Feldes erreicht. Der Weg des Roboters sah also ca. wie in Abb. 1 zu sehen ist aus.

Wenn er in der letzten Ecke des Feldes angekommen war, so traf er statt auf eine schwarze Linie auf ein Stück Aluminiumfolie, die er anhand ihrer großen Reflektionsfähigkeit für Licht erkannte. Im Programm war das Erkennen der Aluminiumfolie mit der Anweisung verbunden, zum Ausgangspunkt zurückzukehren und dem zweiten Roboter mithilfe der Infrarotschnittstelle die Koordinaten der gefundenen Tonnen mitzuteilen. Eine Tonne erkannte er durch einen Drucksensor. Wurde dieser Drucksensor

betätigt, so wurde eine Methode aufgerufen, die den Roboter um die Tonne herumfahren und dann seinen Weg fortsetzen ließ. Um die absolute Position der Tonne auf dem Feld beschreiben und dem anderen Roboter mitteilen zu können, benötigten alle Roboter ein einheitliches Koordinatensystem, mit dem sich jeder beliebige Punkt innerhalb des Feldes definieren ließ.

Dies war, vom Programm aus gesehen, eine der größten Herausforderungen für uns.

Glücklicherweise beinhaltet das Lego-Roboter Softwarepaket bereits das Programm Rotation Navigator, das mithilfe des Rotationssensors die Achsumdrehungen zählt und auf diese Weise das Anfahren von Koordinaten ermöglicht. Damit wären zwei der Herausforderungen unserer Programmierarbeit erläutert, doch muss klar sein, dass damit noch lange kein Roboter seinen Dienst tut. Hiermit wären zwei der Herausforderungen unserer Programmierarbeit erläutert, doch muss klar sein, dass damit noch lange kein Roboter seinen Dienst tut.

Doch damit genug von unserem Suchroboter. Nun wollten wir ja nicht nur ein Gefährt über ein Feld ruckeln und dabei Tonnen suchen lassen. Unser Ziel waren drei zusammenarbeitende Roboter, die die Tonnen am Ende in ein Gefäß neben dem Feld befördern sollten. Beim Transport der Tonnen wären wir auch

schon bei unserem zweiten Roboter mit der Aufgabe, die vom Suchroboter ermittelten Koordinaten anzufahren und die Tonnen abzuholen, um sie dem Fischertechnik-Roboter zu bringen.

Hierbei war als erstes das Problem der Datenübermittlung via Infrarotschnittstelle von Lego-Modul auf Lego-Modul zu bewältigen. Unsere zwei Vorgängerkurse aus den letzten Jahren hatten dieses Problem stets umgangen, indem sie nur einen Roboter, der sämtliche Aufgaben übernahm, bauten. In der Phase der Zielfindung beschlossen wir jedoch, als Pioniere dieses Gebiet in Angriff zu nehmen. Technisch ließ sich dieses Problem der Datenübermittlung schließlich doch relativ einfach lösen.

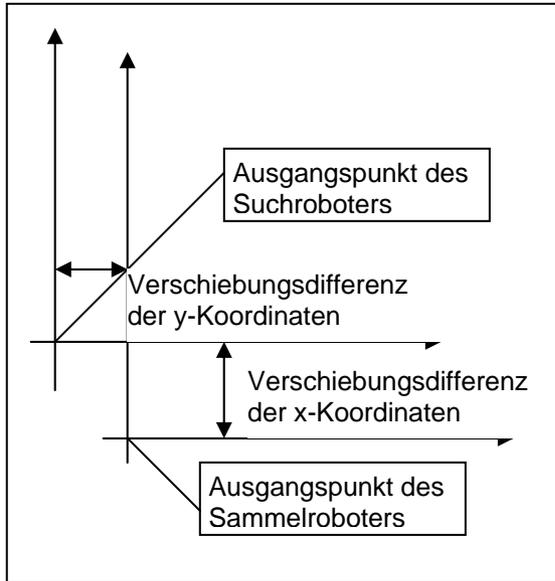
Größere Probleme bereitete uns, dass beide Roboter über ein und dasselbe Koordinatensystem verfügen mussten, da schließlich die Information des Suchroboters „Auf Punkt (485/390) befindet sich eine Tonne“ beim Sammelroboter die Anweisung „Fahre zu Punkt (485/390) und hole die Tonne“ aufrief. Ging nun der Sammelroboter von einem anderen (0/0) Punkt als der Suchroboter aus, so war auch der anzufahrende Punkt verschoben. Diese Verschiebung musste im Programm eines Roboters berücksichtigt werden. Dieses Programm zog bzw. fügte dann die Differenz der beiden Koordinatensysteme ab bzw. hinzu. Somit mussten wir uns auch noch mit dem

mathematischen Problem einer Koordinatensystems-Verschiebung auseinandersetzen.

Nachdem schließlich der Suchroboter das Feld abgefahren und die Tonnenpositionen ermittelt hatte, teilte er dem Sammelroboter die Koordinaten der Tonnen mit. Dieser übertrug dann diese Koordinaten in sein verschobenes Koordinatensystem und fuhr die Positionen an. In seinem Programm war das Erreichen eines Koordinatenpunktes mit der Anweisung verbunden, die Tonne aufzunehmen. Nun fuhr er mit ihr zum Förderband des Fischertechnikroboters, wo er einen Drucksensor des Förderbandes berührte. Wurde dieser Drucksensor betätigt, so lief das Förderband an und beförderte die Tonne in die Zange des Fischertechnik-Roboters, der die Ankunft der Tonne wiederum mit einem Drucksensor bemerkte. Seine Anweisungen zur weiteren Verarbeitung der Tonne wurden durch das Signal „gedrückt“ des Drucksensors aufgerufen.

Man sieht, auch wenn die Aufgabe unserer Roboter scheinbar leicht aussieht, so verlangt sie doch ein perfektes Zusammenspiel von vielen Komponenten, bei denen schon sogenannte „einfache“ Probleme wie das Übermitteln von einem x- und einem y-Wert zu größeren Herausforderungen werden können.

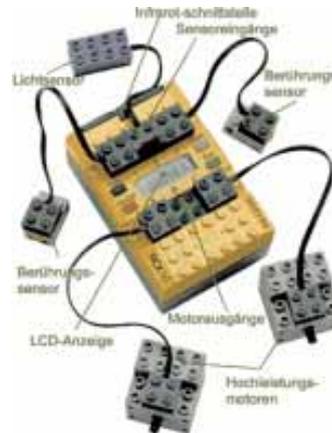
Über ähnliche Probleme und deren Lösung im Bereich der Hardware informieren die nächsten Abschnitte.



Verschiebung der Koordinatensysteme

LEGO

Von Ronja Berg



LEGO ist ein dänisches Spielzeugunternehmen, das sicher jedes Kind kennt. Dabei handelt es sich um ein Bausystem, bei dem verschiedene Bauteile wie Kunststoff-Klötzchen

oder Zahnräder zusammengesteckt werden, um damit Modelle zu konstruieren. Auch für unseren Roboter haben wir das System ausgewählt. Das Herzstück des LEGO Roboters ist der RCX, welcher hier als ein Miniprozessor dient, das sowohl die Programme speichert auch sie abspielt. Wird ihm einmal das fertige Programm per Infrarot mit Hilfe eines Towers, der eine Kabelverbindung zum Computer hat, gesendet, kann er es so oft ausführen, bis ein neues Programm gesendet

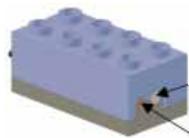


wird, oder die Batterien leer sind. Somit sind unsere Lego Roboter mobil. Der RCX hat

drei Ausgänge für Motoren, die mit A, B und C bezeichnet werden. Die Motoren besitzen jeweils eine Achse, auf die man z. B. Räder oder Zahnräder, die mit den Rädern verbunden sind, anstecken kann. Drei Eingänge für Sensoren hat der RCX auch. Diese werden mit 1, 2 und 3 bezeichnet. Uns standen 3 verschiedene Sensortypen zur Verfügung: Drucksensoren, Lichtsensoren und Rotationssensoren. Der Drucksensor hat vorne einen gelben "Knopf", der als Berührungskontakt dient. In das Programm kann man dann einen



Befehl schreiben, den der Roboter ausführen soll, wenn er gegen einen Gegenstand gefahren ist, also wenn der Sensor gedrückt ist. Der Roboter könnte sich dann zum Beispiel drehen und in die entgegengesetzte Richtung weiterfahren. Der Lichtsensor dagegen kann



Farben unterscheiden, indem er einen Lichtstrahl aussendet und die Reflexionswerte misst. Er besitzt also einen Sender und einen Empfänger. Wird das Licht an einer hellen Farbe wie z. B. weiß reflektiert, kommt mehr Licht an, als bei einer dunklen Farbe, wie Schwarz. Der Rotationssensor hat eine Achse,



die über Zahnräder mit dem Rad verbunden werden kann. Da das Programm von uns die genauen Werte der Achsenbreite, Rad-durchmesser und Übersetzungsverhältnis wusste, kann der Rotationssensor die Umdrehungen zählen und diese dann dem RCX senden. Darauf wertet er die Ergebnisse aus und überträgt sie auf ein virtuelles Koordinatensystem, sodass der Roboter weiß, wo er sich gerade befindet. Die Motoren und Sensoren sind über spezielle Kabel, an deren Enden sich jeweils ein Legostein befindet, mit dem RCX verbunden.

Bau des Legoroboters

Von Andreas Gut

Als erstes brauchte unser Roboter ein Grundgerüst geben. Nachdem wir dieses nach mehreren Fehlversuchen konstruiert hatten, konnten wir den Antrieb einbauen. Unsere erste Idee war, den Roboter mit zwei Raupen, über vier Motoren anzutreiben, aber nach den ersten Tests merkten wir, dass der Roboter, der mit einfachen Test-Programmen ausgestattet war, sich in den Kurven fast willkürlich drehte.



Kursalltag: Sortieren

Es war nicht möglich, den Roboter so zu programmieren, dass er sich in einer Kurve um genau 90° drehte. Auf geraden Strecken war das kein Problem, aber sobald der Roboter wenden sollte, wurde er ungenau und drehte sich mal mehr und mal weniger. Das lag an den starken Reibungskräften, die auf die Rauhen wirkten. Also musste eine neue Idee her. Wir haben uns, da es ja nicht viele andere Möglichkeiten mehr gab, dafür entschieden auf Räder umzustellen. Der Roboter wurde mit zwei Rädern ausgestattet, die jeweils von

einem Motor angetrieben wurden. Das hatte den Vorteil, dass geringe Reibungskräfte wirkten. Wir haben an den hinteren Teil unseres Roboters ein Spornrad gebaut, das sich um 360° drehen ließ. Der Roboter stand auf

drei Punkten auf. Darum kippte er sehr leicht. Außerdem musste das Spornrad am Anfang immer gerade hingestellt werden. Dieses Problem haben wir mit Rutschern gelöst, von denen wir aber nicht nur einen, sondern zwei angebracht haben, damit der Roboter nicht mehr umfällt. Jetzt war der Antrieb im Großen und Ganzen fertig.

Der Roboter sollte aber nicht nur fahren und wenden können, sondern auch Informationen aus seiner Umgebung aufnehmen. Wir haben zwei Rotationssensoren, einen Licht- und einen Tastsensor eingebaut. Die zwei Rotationssensoren waren für das virtuelle Koordinatensystem nötig, in dem sich der Roboter bewegte. Der Tastsensor war dafür zuständig, ein Signal an den RCX weiterzugeben, falls der Roboter auf eine Tonne trifft. Der Roboter hat sich die Position gemerkt, die durch die Rotationssensoren bestimmt wurde. Der Lichtsensor begrenzte den Spielraum des Roboters. An den Kanten der Platte haben wir schwarze Klebestreifen angebracht, an denen der Roboter gewendet hat.

Allgemein war das Bauen mit Lego manchmal ziemlich anstrengend. Man hat gerade etwas fertig gebaut und dann bröseln alles wieder auseinander. Aber insgesamt hat es viel Spaß gemacht einen Roboter mit Lego zu bauen.

Programmierung von Lego Robotern

Von Arne Schauf

Wir haben die Lego-Roboter in der Programmiersprache Java programmiert. Dabei gab es einige Probleme. Zum Beispiel mussten wir uns erst einmal mit der Programmiersprache vertraut machen, denn fast niemand wusste vorher irgendetwas über Java.

Während wir die Grundzüge von Java kennen lernten verursachten wir Fehlermeldungen, die der Editor *JCreator* ausgab als wir das geschriebene „Programm“ kompilierten (kompilieren d. h., von der Sprache die Menschen verstehen, übersetzen, in eine Sprache, die der Computer versteht). Zunächst waren es nur kleine Fehler, wie Semikolon oder Klammer vergessen. Diese Fehler lernten wir recht schnell selbst zu lösen. Als die Programme dann komplizierter wurden gab es auch sehr komplizierte Fehler, bei den wir dann Herr Taulien oder Herr Wilke fragen mussten.

Für das Programm des Sammler-Roboters verwendeten wir sogenannte Behavior-Klassen. *Behaviors* (engl. Behavior = Verhalten, Benehmen), sind bestimmte Verhaltensmuster die der Roboter annehmen soll, wenn etwas bestimmtes passiert, z.B. wenn er gegen eine Wand fährt oder er eine dunkle Linie überfährt. Der Roboter registriert dies mit Hilfe

der Sensoren. Wenn ein bestimmtes Ereignis eintritt wird das entsprechende Behavior aufgerufen, welches dann bestimmte Aktionen durchführt, z.B. zurück zum Null-Punkt fahren. Außerdem verwendeten wir ein virtuelles Koordinatensystem für unseren Roboter. Er startete immer am Punkt 0/0 des Koordinatensystems. Der Roboter war mit Rotationssensoren ausgestattet, damit er immer „wusste“ wo er gerade ist (er misst mit Rotationssensoren die zurückgelegte Strecke, und berechnet dann, wo er sich gerade befindet und in welchem Winkel er sich befindet). Doch es gab auch Probleme beim

Verwenden des Koordinatensystems, da z.B. manchmal die Rotationssensoren nicht funktioniert haben, aber das war nur der Anfang, denn es gab meistens immer irgendeinen Fehler (entweder im Programm, oder im Bau). Als wir dann gegen Ende die Verständigung zwischen den Robotern versuchten mussten wir auch das Koordinatensystem umsetzen. Bei der Verständigung mussten wir viel experimentieren, da auch die Leiter die Verständigung zwischen zwei RCX (Steuerungsbausteine („Computer“) der LEGO-Roboter) dies noch nie gemacht hatten.

Neben Lego, über das Sie nun bestens informiert sind, stand uns Fischertechnik als zweites Hardwaresystem zur Verfügung.

Fischertechnik

Sensoren

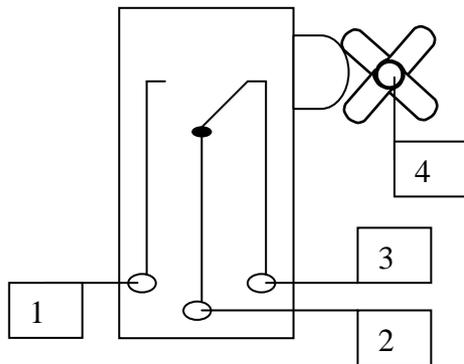
Von Carsten Plasberg

Ein Sensor ist ein Gerät, welches Umweltinformationen in für einen Computer verständliche Informationen umwandelt.

Bei Fischertechnik haben wir nur den Drucksensor verwendet.

Dieser Drucksensor ist ein einfacher Schalter, der, wenn er betätigt wird, die Verbindung zwischen 2 und 3 unterbricht (Öffner). Dafür verbindet er aber die Kontakte 1 und 2 (Schließer).

Je nach dem ob eine Verbindung zwischen



Das Schema des Tasters

den Kontakten besteht kann man feststellen ob der Taster gedrückt oder losgelassen ist,

wenn man weiß, ob der Taster ein Öffner oder Schließer ist.

Über ein spezielles Zahnrad (4), das den Taster viermal pro Umdrehung betätigt kann man die Umdrehungen einer Achse messen. Dadurch kann man den Taster zu einem Rotationssensor "aufrüsten".

Mit der Kombination von einem Taster und einem Rotationssensor kann die Entfernung eines Objektes zum Nullpunkt am Anschlagtaster bestimmt werden. Hierzu wird das Objekt an den Anschlag gefahren, der damit als Nullpunkt definiert wird. Von diesem Punkt kann man die Umdrehungen des Antriebs zählen und diese in cm oder einen Winkel umrechnen. Dieses Prinzip haben wir auch bei unserem Roboter angewandt, da wir mehrere Achsen hatten, die bestimmte Positionen anfahren sollten, die von dem Benutzer frei festgelegt werden.

Fischertechnikmotoren

Von Carsten Plasberg

Es gibt zwei verschiedene Fischertechnikmotoren. Einen kleinen, leichten und dafür auch etwas schwachen und einen großen und starken Typ. Dem kleinen Motor steht zur Kraftübertragung eine Schnecke zur

Verfügung, auf die zur Benutzung ein kleines Getriebe aufgeschoben wird. Der große Motor hat ein internes Getriebe und ein Zahnrad, von dem wir die Kraft übertragen können.



Ein Fischertechnik-Motor

Das Interface

Von Philipp Gutknecht

Das Fischertechnik Interface ist eine Schnittstelle zwischen Fischertechnik Bauteilen und dem Steuerungscomputer.

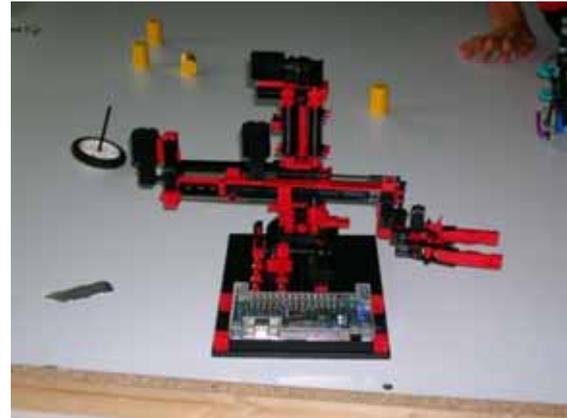
Es ist ständig mit dem Computer verbunden um die Ein- und Ausgänge anzusteuern zu können. An dieses Interface können bis zu 8 Motoren bzw. bis zu 18 Sensoren angeschlossen werden.

Wir haben den Steuercomputer in der Sprache Java programmiert.

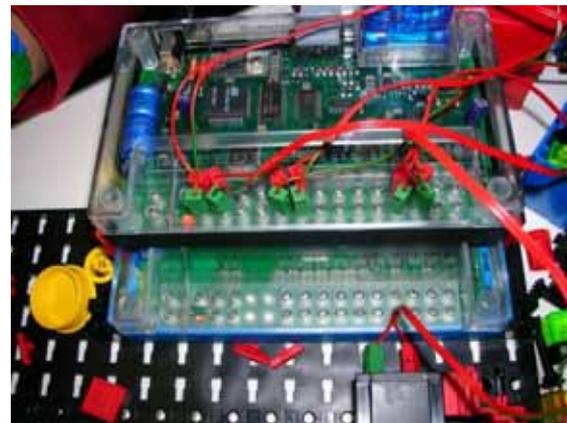
Uns stand eine Grundbibliothek in Java mit verschiedenen Klassen zur Verfügung, die

Herr Wilke, Herr Taulien und Herr Richter noch modifiziert haben.

Aufgrund dieser schon geleisteten Vorarbeit sind um einen Motor einzuschalten nur noch wenige simple Befehle erforderlich.



Fische Deluxe



Ein Fischertechnik-Interface

Motoren:

```
1 import ftinterface.*;
2 import ftinterface.device.*;
3 import ftinterface.event.*;
4
5 public class MotorTester {
6     private FtMotor motor;
7     private FtInterface fti;
8
9     public static void main() {
10         fti = new FtInterface(FtConstants.COM4, FtConstants.EXT_MODULE_CONNECTED);
11         motor = new FtMotor(fti, FtConstants.M1);
12         motor.left();
13         motor.right();
14         motor.stop();
15     }
16 }
```

Erläuterung:

Zeile 1 bis 3: Hier werden die vordefinierten Pakete eingebunden.

Zeile 6: Deklariert eine Variable `motor` vom Typ `FtMotor`

Zeile 7: Deklariert eine Variable `fti` vom Typ `FtInterface`

Zeile 11: Erstellt im Objekt `fti` eine Instanz d.h. eine unabhängige Kopie der Klasse `FtInterface`, mit zwei speziellen Übergabewerten in den Klammern, z.B. den Anschluss am PC.

Zeile 12: Erstellt eine Instanz von `FtMotor`.

Zeile 14: Startet den Motor im Linkslauf.

Zeile 15: Schaltet den Motor auf Rechtslauf um.

Zeile 16: Stoppt den Motor.

Tastensoren

Um einen Tastsensor abzufragen stehen „Abhörer“ sog. Listener zur Verfügung, die bei Statusänderung des Tasters Anweisungen ausführen.

```

1 import ftinterface.*;
2 import ftinterface.device.*;
3 import ftinterface.event.*;
4
5 public class TasterTester{
6     private FtPushButton schalter
7     private FtInterface fti;
8
9     public static void main(){
10        fti = new FtInterface(FtConstants.COM4, FtConstants.EXT_MODULE_CONNECTED);
11        schalter = new FtPushButton(fti, FtConstants.E3, FtConstants.PUSH_CLOSE)
12        schalter.addDigitalInportListener(new sListener())
13
14        private class sListener implements FtDigitalInportListner{
15
16        public void inPortStateClosed(FtInterfaceEvent, e){
17            i++
18        }
19        public void inPortStateOpened(FtInterfaceEvent, e){
20
21        }
22 }

```

Erläuterung:

- Zeile 6: Deklariert eine Variable schalter vom Typ FtPushButton
- Zeile 13: Erstellt eine Instanz von FtPushButton
- Zeile 14: Fügt dem Schalter einen Listener namens sListener hinzu.
- Zeile 16: Klasse des Listeners basierend „implements“ auf der Klasse FtDigitalInPortListener

Zeile 18 bis 20:

Hier reagiert das Programm auf das Ereignis (=event) inPortStateClosed d.h. Schalter ist geschlossen und erhöht den Wert der Variablen i um 1 (i++). Dies ist ein einfacher Schaltimpulszähler.

Zeile 21 bis 23:

Beim Ereignis inPortStateOpened d.h. Schalter geöffnet passiert nichts.



Fischi Deluxe im Bau

Das Programm zum Roboter:

Von Philipp Gutknecht

Das Programm ist genau so strukturiert, wie der Roboter gebaut ist:

Aus den vorgegebenen Klassen `FtMotor`, `FtPushButton` und `FtInterface` schreiben wir die Klassen:

- `FtZähler`, die die Impulse eines Schalters zählt.
- `MotorMitAnschlag`, die einen Motor beim Anschlag an einen von zwei Drucksensoren stoppt.

Daraus schreiben wir dann die Klasse `FtAchse`, die einen `MotorMitAnschlag` und einen `FtZähler` beinhaltet.

Dazu kommt noch eine Klasse für die Zange und eine Klasse, die die grafische Oberfläche des Programms darstellt.

Die Hauptklasse `FtRoboter` beinhaltet nun 3 Achsen vom Typ `FtAchse`, eine Zange vom Typ `FtZange` und eine GUI (=graphical user interface = Grafische Benutzerschnittstelle).

Fischi Deluxe:

Von Finn Lanzendorfer

Der Fischertechnikroboter, den wir gebaut haben, hat drei Achsen und eine Greifzange. Die drei Achsen sind:

- Den Greifarm vor und zurück fahren(x-Achse)
- hoch und runter. (y-Achse)
- den Roboter drehen(z-Achse)

Die Greifzange wird durch einen Motor bewegt, der am anderen Ende des Greifarmes sitzt. Die Zange ist mit ihm durch eine drehbare Stange verbunden. Auf dem Greifarm sitzt der Motor für die x-Achse. Der Motor für die Drehung ist fest mit der Basisplatte verbunden und dreht den Roboter mit einer Schnecke. Der Motor für die Auf- und Abwärtsbewegung sitzt oberhalb des Fahrbereichs.

ches der y-Achse. Diese Achse ist eine Metallstange mit einer Schnecke, die auf die Stange gesteckt ist. An jeder Achse und am Greifarm befindet sich jeweils ein Drehsensor, um die Impulse zu zählen, damit man weiß, wie weit die jeweilige Achse sich bewegt hat. Außerdem gibt es an jedem Ende jeder Achse einen Tastsensor, der dann gedrückt ist, wenn die entsprechende Achse am Anschlag ist. Der Arm an sich ist außen am Roboter angebaut, ist aber im Gleichgewicht. Das Interface steht etwas abseits, damit der Roboter sich „frei“ bewegen kann und sich nicht im Kabelgewirr verfängt, da an das Interface alle Kabel angeschlossen sind. Es sind immerhin 4 Motoren, 4 Drehsensoren und 7 Tastsensoren.

Exkursionstag

Beim Konzern ABB

Von Florian Huynh

Nach einer langen Bahn und Busfahrt vom Krebsforschungszentrum zum Forschungszentrum der Firma ABB in Ladenburg, erwartete uns der Leiter der Robotik-Abteilung vor

Ort, um uns durch das Forschungszentrum zu führen.

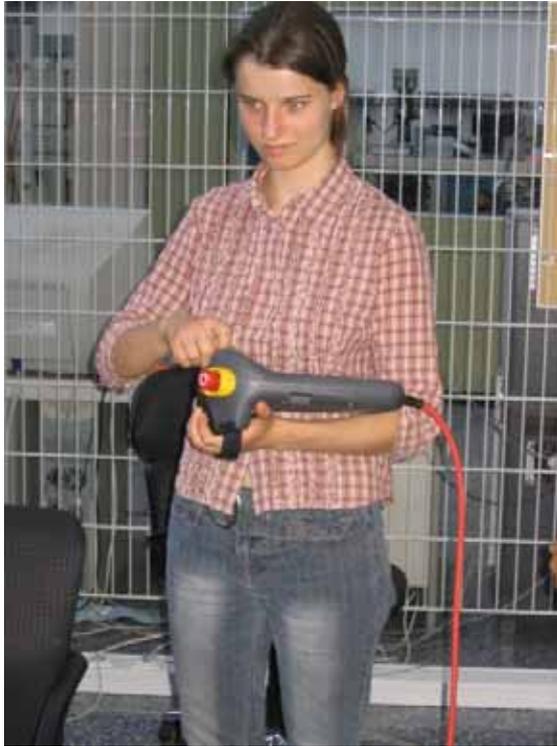
Anfangs wurden wir in eine Halle geführt, in der viele Roboter für verschiedene Aufgaben aufgestellt waren. Diese Roboter waren hinter Käfigen aufgestellt, da sie vielleicht eine Fehlprogrammierung haben oder blitzartige Bewegungen vollführen könnten. In dieser Roboterhalle waren einige Mitarbeiter bei Forschungen, doch zwei von ihnen erklärten sich bereit uns die einzelnen Roboter genauer vorzustellen.

Wir teilten uns in zwei Gruppen.

Eine Angestellte führte die eine Gruppe zu einem Roboter, der speziell zum Schreiben konzipiert war. Sie erklärte uns zunächst die Zusammenarbeit von Steuerung und Roboter. Der Schreibroboter besaß sechs Achsen um sich zu bewegen. Die Fernsteuerung war durch ein Kabel mit dem Computer verbunden, dieser hatte wiederum mit dem Roboter einen Kontakt. Anschließend durfte jeder von uns den Roboter per Fernsteuerung lenken.

Am Ende gab es noch eine Vorstellung, bei der der Roboter selbstständig einen ihm vorgegebenen Namen schrieb. Ein Angestellter erklärte der anderen Gruppe die Konstruktion und Aufgaben zweier Industrieroboter.

Der erste kleinere Roboter war so gebaut, dass er Objekte mit seinem Greifarm packen und zu einem anderen Roboter tragen konnte.



Beim Steuern eines „echten“ Roboters

Der zweite Roboter war ein wenig größer als der erstere, er hatte anders als beim ersten keinen Greifer, sondern einen Schweißgerät als Manipulator.

Theoretisch könnten diese beiden universellen, programmierbaren Maschinen in einer Fabrik zusammen arbeiten.

Da wir so eifrig Fragen gestellt und so viel Interesse gezeigt hatten, schlug unser eigent



Für die meisten von uns der erste „echte“ Roboter

licher Leiter nach der Besichtigung vor, im Hauptgebäude uns den Konzern ABB in einer Präsentation vorzustellen. Eigentlich sollte er uns einen Vortrag von ca. 10min. halten, am Ende war es dann aber knapp eine Stunde! Am Ende verabschiedeten wir uns von allen Mitarbeitern und machten uns wieder auf den Weg zum vereinbarten Treffpunkt in Heidelberg.

Persönlicher Rückblick auf die Akademiezeit

Von Thomas Nutz

Mein persönlichen Rückblick hat die Aufgabe, einen Eindruck der Atmosphäre und aller durchlebten Stimmungsschwankungen mit allen Hochs und Tiefs des Kurses zu vermitteln.

Dazu sind zuerst einige Erklärungen zum Konzept und der Planung unserer Kursarbeit nötig. Bei diesem Punkt ist bemerkenswert, dass wir, als wir uns am Fr, dem 26.8., zum ersten Mal in einem Gruppentreffen zusammenfanden, noch kaum eine Vorstellung hatten, auf was die Eltern sich bei der Abschlusspräsentation gefasst machen sollten. Die Zielsetzung sollte aus der Schar der Teilnehmer heraus und nicht von der Kursleitung her erfolgen. Die Absicht dahinter dürfte einleuchten: Im Idealfall sollte sich innerhalb der 13 Tage eine Arbeitsgemeinschaft unter dem Dach eines gemeinsam inspirierten Zieles zusammenfinden, in dem jeder einzelne seine eigenen Ideen wiedererkennt. Die Folge war eine generell hohe Motivation und die Vermeidung von desinteressierten Außenseiterrollen. Dieses auf eine Veranstaltung wie diese Ferienakademie zugeschnittene Konzept wurde uns schon bei

dem Einführungswochenende mit allen Voraussetzungen wie Verhaltensregeln und der erstrebenswerten Arbeitseinstellung vorgestellt.

Für unseren Kurs kann man aus meiner Sicht durchaus von einem erfolgreichen Konzept sprechen. Wir waren alle an den Arbeiten des Kurses beteiligt und dennoch konnte jeder das tun, was ihn am meisten interessierte und wo er seine Begabungen einsetzen konnte. Eine wichtige Grundlage für eine funktionierende Kursarbeit war die gelungene Gruppenbildung, bei der sich keine Gruppe isolierte und in ihren Aufgabenbereich zurückzog. War beispielsweise ein Mitglied der Suchrobotergruppe gerade ohne Aufgabe, so war es kein Problem, sich einfach an einen anderen Gruppentisch zu setzen und sich dort einzubringen.

Nach diesen Grundlagenerläuterungen werde ich im Folgenden auf den Stimmungsverlauf eingehen. Wie schon bemerkt waren die ersten Tage stark geprägt von Einarbeitung in die Themenbereiche, der Aneignung erster Programmiererfahrungen und vor allem natürlich

der Frage: „Was wollen wir den Eltern an der Abschlusspräsentation nun eigentlich zeigen?“ Praktisch sah diese Einarbeitungszeit so aus, dass wir uns zum einen vor den Laptops in das Programmieren von Roboter-Beispielprogrammen vertieften und zum anderen in die beiden Hardware-Systeme Lego und Fischertechnik einarbeiten, bis uns nach ca. 90 min. in einer PowerPointpräsentation eine neue Programmieraufgabe gestellt wurde. Diese Gesamtkursvorträge sorgten für einen strukturierten Arbeitsalltag, bei denen uns zum einen die Grundlagen für die späteren Programmierarbeiten vermittelt wurden und zum anderen die Zielfindung im Gange war. Mir persönlich erschienen die ersten 3-4 Tage als am anstrengendsten im Vergleich mit anderen Zeitabschnitten, da ich oft nicht mit den gestellten Aufgaben zurechtkam und mich noch nicht so selbstbewusst fühlte, einen Kursleiter um Hilfe zu bitten. Allmählich begannen sich Interessen zu zeigen und Gruppenkonstellationen zu bilden. Diese Phase des allgemeinen Vertrautmachens mit den Themenbereichen wurde abgeschlossen mit der entgeltigen Einteilung der Teilnehmer in drei Gruppen, wobei jede Gruppe einen Roboter zu entwerfen hatte. Zur Mitte und dem Ende der ersten Akademiewoche hin begannen sich also die Aufgaben zu verzweigen, und mittler

weile hatte auch jeder seine eigenen Interessen und Begabungen innerhalb der Gruppe eingeordnet, wodurch sich Spezialisierungen ergaben. Dies machte die Gesamttreffen des Kurses immer unbedeutender, sie wurden teilweise ersetzt durch Beratungen von zwei bis drei Teilnehmern mit einem Kursleiter, wobei dann auf die spezielleren Fragen eingegangen werden konnte. Hier lernte ich zu ersten Mal die Notwendigkeit von persönlicher Förderung kennen, wie sie ja oft in unserem Schulsystem als unzureichend empfunden wird. Der Zeit von Montag bis Donnerstag der ersten Woche würde ich also die hauptsächliche Entstehung der Roboter zuordnen. Zum Ende der ersten Woche rückte die anstehende Rotation immer mehr ins Blickfeld. All



mählich mussten wir uns angesichts der Vorbereitung auf die Rotation klarere Gedanken über unser konkretes Vorhaben machen, wobei manch kühner Wunschtraum der Anfangszeit der Realität und der verbleibenden Zeit zum Opfer fiel. Bei der Vorbereitung der Präsentation wurde dann auch die Frage immer wichtiger, wie man das Erarbeitete veranschaulichen und Außenstehenden näher bringen sollte, was uns zum Zusammenfassen des bisherigen Lerninhaltes zwang. Auch der Gedanke an die Zeit nach der Akademie, in der man sich ja vielleicht noch einmal mit dem Programmieren in Java zu beschäftigen plante, wurde wichtiger. In diesem Sinne mussten die Leiter über allgemeinere Gebiete von Java und die Grundlagen des objektorientierten Programmierens Auskunft geben, alles wurde sorgsam notiert, um nach der Akademie nicht mit leeren Händen dazustehen. Doch nicht nur von den Lerninhalten sollte etwas hängen bleiben: Man sammelte Unterschriften, tauschte Erinnerungsstücke aus und machte Photos.

Diese Zeit des Sammelns und Zusammentragens wurde gefolgt von dem wohl bewegtesten Abschnitt der Akademie: Der Exkursion nach Heidelberg. Für unseren Kurs beinhaltete diese Exkursion im Wesentlichen einen Besuch des Forschungszentrums der Firma ABB in Ladenburg, wo wir endlich „echte“ Roboter

zu Gesicht bekamen. Ich denke, dass wohl jeder Teilnehmer in diesem Zusammenhang von einem gelungenen Ausflug sprechen würde: Wir fragten den Leitern des Forschungszentrums Löcher in den Bauch und so wurden aus den eingeplanten 2 h gute 3 Stunden. Vorausgegangen war dieser Besichtigung der Besuch einer Vorlesung in der Uni Heidelberg inklusive echtem „Vorlesungsfeeling“, was wohl auch niemandem so schnell in Vergessenheit geraten wird. Das abschließende Abendessen im Ruderclub Heidelberg vor einer äußerst bewegten Busfahrt hat wohl bei allen Akademieteilnehmern bleibende Erinnerungen vor allem an gute Kameraden hinterlassen.

Als wir uns dann also am Dienstagmorgen im Kurstreffen wiederfanden, mussten wir uns der Tatsache stellen, dass wir noch einen einzigen Tag bis zur großen Abschlusspräsentation übrig hatten. Wie Herr Taulien hin und wieder zu sagen pflegte: Schicht am Schacht! Wir programmierten und bauten, was das Zeug hielt, und bis kurz vor dem Mittagessen des Mittwochs waren wir noch mit letzten Verbesserungen beschäftigt, wobei sich ein Teil des Kurses der Vorträge, dem Vortragsmaterial und der Powerpointpräsentation annahm. Erst in diesem Endspurt wurden die drei Roboter zusammen getestet, wo noch unerwartete Probleme auftraten, was die

Arbeitsatmosphäre weiter unter Spannung setzte.

Die Abschlusspräsentation wurde somit mit einem Gemisch aus Angst, Stolz und Wehmut erwartet, verlief dann jedoch für jeden Beteiligten trotz kleiner Zwischenfälle erfreulich. Bei den Vorträgen ist zu erwähnen, dass wir als einer von wenigen Kursen einen Vortrag auf Englisch arrangierten, um auch unseren chinesischen Gästen einen Einblick in unsere Kursarbeit zu ermöglichen. Mit den zwei Austauschschülern aus China kam unser Kurs übrigens geradezu ideal zurecht, ein Junge trug sogar zusammen mit der Englisch-Vortragsgruppe einen Teil der Präsentation vor.

Die Integration in unserem Kurs hatte funktioniert.

Diese letzten drei Tage der Akademie schienen davongelaufen zu sein, trotz der Versuche, sich wenigstens einige Erinnerungen an diese sicher für jeden von uns prägende Zeit festzuhalten. Ich denke, dass keinem Kursteilnehmer die Frage „Wo ist nur die Zeit geblieben, auf die ich Wochen und Monate gewartet habe?“ in diesen Tagen fremd gewesen ist.

Im Nachhinein betrachtet bleibt mir der Eindruck einer äußerst anregenden und aufgeregten Zeit, die wohl für jeden Akademieteilnehmer etwas neues war, und 13 Tagen, die ich schon wegen der gewonnenen Freunde nicht vergessen werde.

