

**Seminar Agentensysteme:
Roboter – Vom Webstuhl zum Androiden**

Wintersemester 2005/2006

Dozent:

Dr. Dietmar Lammers

Sascha Hauke

haukes@math.uni-muenster.de

Zusammenfassung

ROBOTER nehmen eine immer größere Rolle in unserem Leben ein. Zunächst beschränkte sich ihr „Lebensraum“ auf Fabrikhallen und Labore, mit der immer schneller werdenden technischen Entwicklung der letzten Jahrzehnte erobern Roboter jedoch immer mehr Nischen für sich. Waren sie lange Zeit Objekte, die man aus Fernsehreportagen oder Science-Fiction Büchern kannte, so häufen sich heute die Berührungspunkte mit Robotern im menschlichen Alltag. Sei es als Helfer in Industrie, Gesundheitswesen oder gar im Haushalt, Roboter werden immer präsenter – sogar als Spielzeug. Der folgende Artikel soll in seiner kurzen Form einen Überblick darüber geben, wie sich die Roboter in groben Zügen entwickelt haben, und einen kleinen Ausblick darauf geben, wohin diese Entwicklung führen mag.

Inhaltsverzeichnis

| | |
|---------------------------------------------------------------|-----------|
| Inhaltsverzeichnis | I |
| Abbildungsverzeichnis | II |
| Tabellenverzeichnis | II |
| Listings | II |
| 1 Ursprung und Definition: Was ist ein Roboter? | 1 |
| 2 Vom Webstuhl zum Marsrover | 2 |
| 2.1 Die Anfänge - Das 18. und 19. Jahrhundert | 2 |
| 2.2 Die technische Revolution – das 20. Jahrhundert | 3 |
| 3 Asuro | 5 |
| 3.1 Aufbau und Leistung | 5 |
| 3.2 Software und Programmierung | 6 |
| 4 Vom Marsrover zum Androiden | 8 |
| 4.1 Ausblick | 8 |
| A Roboterklassifikation | 10 |
| B Informationen zu <i>Asuro</i> | 11 |

Abbildungsverzeichnis

| | | |
|-----|------------------------------------------------------------------------|----|
| 2.1 | Spinning Jenny, ca. 1764 | 2 |
| 2.2 | Atanasoffs Computer, 1942 | 3 |
| 2.3 | <i>Unimat</i> im Einsatz, 1961 | 3 |
| 2.4 | Mars Exploration Rover (links), Vorgänger Sojourner (rechts) | 4 |
| 3.1 | Roboterbausatz <i>Asuro</i> | 5 |
| 4.1 | Kawada Industries <i>HRP-2</i> | 8 |
| 4.2 | RobotCubs <i>iCub</i> | 9 |
| B.1 | <i>Asuro</i> – Schematischer Aufbau | 12 |

Tabellenverzeichnis

| | | |
|-----|------------------------------------------------|----|
| B.1 | <i>Asuro</i> – Hardwareeigenschaften | 11 |
|-----|------------------------------------------------|----|

Listings

| | | |
|-----|----------------------------------------------------|----|
| 3.1 | <i>Asuro</i> – Initialisierung | 6 |
| B.1 | <i>Asuro</i> – Code für Linienverfolgung | 12 |

Kapitel 1

Ursprung und Definition: Was ist ein Roboter?

ROBOTER – wenn dieses Wort fällt, weiß heute jeder, was damit gemeint ist. Oder so könnte man zumindest vermuten – doch tatsächlich ist gar nicht so sehr klar, was sich hinter diesem Wort eigentlich verbirgt. Die Vorstellungen eines jeden einzelnen, wie denn nun eine solche Maschine auszusehen hat, sind von einer Vielzahl von Faktoren bestimmt. Am Anfang soll daher eine genauere Betrachtung dessen stehen, was wir uns heute – als Gesellschaft – unter Robotern vorstellen.

Der Begriff des Roboters leitet sich aus dem slawischen Wort „robota“ ab, das ursprünglich soviel bedeutete wie „Frohdienst“, heute aber auch in der Bedeutung von „Arbeit“ oder „Maloche“ gebraucht wird.

Entscheidend für die Prägung dieses Begriffs war der tschechische Autor Karel Capek, der mit seinem 1921 erschienen Roman „Rossum’s Universal Robots“ für die Popularisierung des Names sorgte. In seinem Buch beschreibt Capek die Schaffung von perfekten Androiden – Maschinen in Menschengestalt – durch den Wissenschaftler Rossum, um den Menschen alle schweren und gefährlichen Arbeiten abzunehmen. Die Roboter lehnen sich jedoch gegen dieses Sklavendasein auf und töten das gesamte menschliche Leben. Sicherlich hat diese Geschichte unser Verhältnis zu Robotern ge-

prägt; das Thema ist auch heute noch ein Prototyp der Science-Fiction Literatur. Sie führte jedoch auch zu einer schnellen Aufnahme des Wortes Roboter in den allgemeinen Sprachgebrauch.

Die Definition, was aber einen Roboter im eigentlichen Sinne ausmachen soll, ist auch heute bei weitem noch nicht einheitlich. Insbesondere existiert (noch) eine große Diskrepanz zwischen den Robotern aus Literatur/Film und tatsächlich existierenden Maschinen. Betrachtet man einige Definitionen, wie sie in Wörterbüchern oder Industriestandards gegeben werden, so sieht man dies sehr deutlich. Der Fremdwörterduden etwa beschreibt einen Roboter als „äußerlich wie ein Mensch gestaltete Apparatur, die manuelle Funktionen eines Menschen ausführen kann“, während andererseits laut VDI ein Roboter „ein universell einsetzbarer Bewegungsautomat mit mehreren Achsen“ ist, „dessen Bewegungen hinsichtlich Folge und Wegen bzw. Winkeln frei programmierbar und ggf. sensorgeführt“ ist.

Eine genauere Klassifizierung von Industrierobotern (exemplarisch die der Japanese Industrial Robot Association) findet sich in Anhang A auf Seite 10 – um jedoch eine einheitliche Basisdefinition zu haben, sei im folgenden ein Roboter einfach eine Maschine, die autonom eine bestimmte Aufgabe erfüllt.

Kapitel 2

Vom Webstuhl zum Marsrover

2.1 Die Anfänge - Das 18. und 19. Jahrhundert

MASCHINEN, die ihre Aufgaben weitgehend autonom erfüllen, existieren seit den Anfängen der industriellen Revolution gegen Ende des 18. Jahrhunderts. Sicher sind dies noch keine Roboter, sie stellen jedoch eine wichtige Vorstufe in der Entwicklung moderner mechanischer und elektronischer Geräte dar. Mit der Erfindung der Dampfmaschine Anfang des 18. Jahrhunderts, und insbesondere durch die Verbesserung ihres Wirkungsgrads durch die Entdeckungen von James Watt in der zweiten Hälfte dieses Jahrhunderts, stand eine zuverlässige Methode zur Verfügung, um mechanische Energie zu erzeugen, ohne auf die Elemente der Natur oder Muskelkraft angewiesen zu sein. Damit setzte eine rasante Entwicklung der Fertigungstechnik ein und es entstanden erste Automaten, die selbständig Arbeiten erledigen konnten. Die prominentesten und wohl auch die wichtigsten Beispiele dafür waren die sogenannte *Spinning Jenny*, eine automatische Spinnmaschine, und der *Power Loom*, der erste automatische Webstuhl, 1785 von Edmond Cartwright erfunden.

Der *Power Loom* wurde durch eine Art hölzerner Lochkarte gesteuert und somit war das von ihm erzeugte Webmuster sogar variierbar – sein „Programmablauf“ war sozusagen veränderbar. Diese Maschinen führ-



Abbildung 2.1: Spinning Jenny, ca. 1764

ten zu einer starken Rationalisierung des Arbeitsprozesses in der Textilherstellung und fanden dadurch breiten Zuspruch und Verbreitung, allerdings stießen sie auch aus dem gleichen Grund auf offene, zum Teil mit Gewalt verbundene, gesellschaftliche Ablehnung.

Im Laufe des folgenden Jahrhunderts verschärfte sich dieser Umbruch in den Fertigungsmethoden und Automaten und Maschinen ersetzten den Menschen in zunehmenden Maß als Erbringer von handwerklicher Arbeit und Leistung. Die industrielle Gesellschaft war geboren, und damit die Voraussetzungen die zur Entwicklung der Robotik führen sollten.

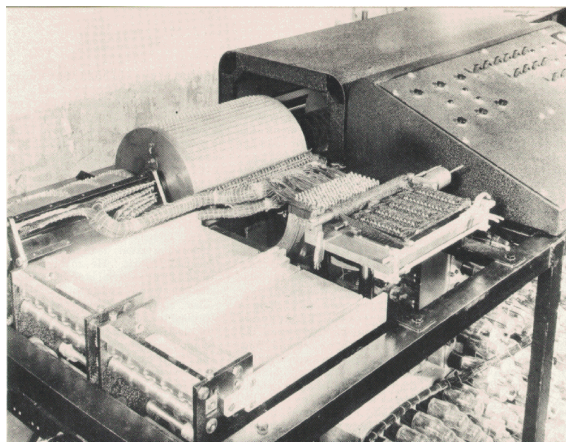


Abbildung 2.2: Atanasoffs Computer, 1942

2.2 Die technische Revolution – das 20. Jahrhundert

OBWOHL die Fortschritte des 18. und 19. Jahrhunderts eine wichtige Basis für die Entwicklung vom, aus heutiger Sicht, relativ primitiven mechanischen Gerät hin zu wirklich autonomen Einheiten darstellten, brachte erst die Einführung der Elektronik Anfang/Mitte des 20. Jahrhunderts die entscheidenden Impulse.

Weitreichende Elektrifizierung stellte eine universell verfügbare und einfach zu transportierende Energieform zur Verfügung, und die Erfindung von elektronischen Rechenmaschinen, Computern, in Europa und Amerika, erlaubte es durch die Kombination von Mechanik und Elektronik, Maschinen immer komplexere Arbeitsgänge beizubringen, d.h. Ihre Autonomie zu steigern.

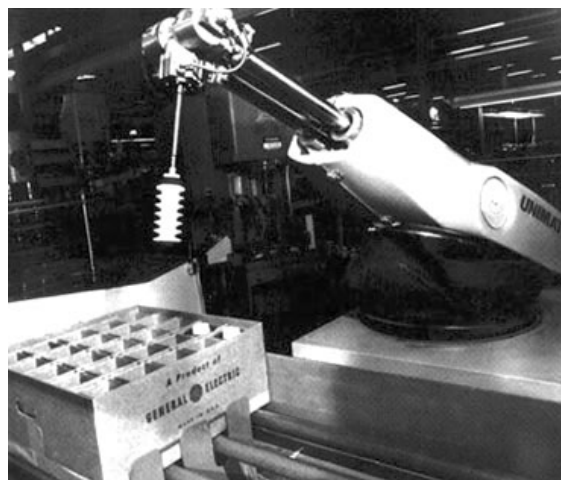
Obwohl die ersten Computer, wie etwa der *ENIAC* und Atanasoffs Computer in den USA, oder Zuses *Z-Reihe* in Deutschland, noch recht groß und rechenschwach waren, setzte nach dem 2. Weltkrieg eine immer rascher werdende technische Entwicklung ein, die es bereits 1961 ermöglichte den ersten Roboter in der industriellen Fertigung ein-

zusetzen. Dieser erste im Einsatz befindliche Roboter, in den 1950er Jahren von George Devol und Joseph Engelberger, den beiden Gründern der ersten Robotikfirma *Unimation*, entwickelt (Patent 1954, Firmengründung von *Unimation* 1956), versah unter dem Namen *Unimat* seinen Dienst in der Autofertigung bei General Motors.

Das Patent für den *Unimat* spezifizierte ihn als „a more or less general purpose machine that has universal application to a vast diversity of applications where cyclic control is desired“ [eine multifunktionale Maschine, die universell in einer Vielzahl von Anwendungen zum Einsatz kommen kann, bei denen zyklische Kontrollabläufe gewünscht sind].

In den 1960er Jahren waren die Vereinigten Staaten führend in der Entwicklung, Produktion und dem Einsatz von Industrierobotern, wobei der Einsatz vor allem in der Automobilindustrie erfolgte. Der erste Industrieroboter Japans wurde ab 1967 eingesetzt, der erste Deutschlands ab 1970 bei Daimler-Benz.

Ab den 1970er Jahren drangen japanische Konzernkonglomerate massiv in den Industrierobotermarkt ein und übernahmen die Führungsrolle – heute werden in Japan allein ca. 400.000 Industrieroboter eingesetzt, fast doppelt so viele wie in Deutschland und

Abbildung 2.3: *Unimat* im Einsatz, 1961

den Vereinigten Staaten zusammen, die im Robotereinsatz auf Platz zwei und drei rangieren.

Während Unimat in seiner Beweglichkeit auf nur 3 Achsen beschränkt war, erfolgte 1969 durch die Stanford University ein weiterer Schritt in der (Industrie-)Roboterentwicklung, die Einführung des auch heute noch gebräuchlichen, modernen 6-Achsen Artikulatorarms. Die dadurch erlangte größere Beweglichkeit weitete das Einsatzfeld dieser Maschinen auf Gebiete wie Bogenschweissen oder Montageaufgaben aus. Fortschritte in dieser Kombinationsform von Elektronik und Mechanik – der Mechatronik – ermöglichten die Konstruktion immer komplexerer Roboterteile, wie etwa Manipulatorarmen.

Aus der Kombination der durch die immer mehr voranschreitende Miniaturisierung und Leistungssteigerung in der Microprozessorentwicklung und der Verbesserungen in der Mechatronik ergibt sich nun die Möglichkeit Maschinen zu entwickeln, die sich wirklich autonom verhalten, sowohl im physischen Sinn in der Möglichkeit der eigenständigen Bewegungen und Interaktion mit der Umwelt, als auch in ihrem Steuerprogrammablauf. Autonome Automaten, die selbstständig handeln und entscheiden, sind besonders für militärische und wissenschaftliche Anwendungen interessant. In der Forschung liegt ihr Einsatzgebiet zum Beispiel in der Erkundung der Tiefsee oder des Welt-raums – Bereiche, die Aufgrund der herrschenden Eigenschaften (zum Beispiel den Umgebungsdruck, Strahlung oder große Entfernungen), wenn überhaupt nur unter größten technischen Anstrengungen von Menschen erforscht werden können und in denen eine direkte Fernsteuerung unpraktisch ist. Ein prominentes Beispiel sind die beiden NASA Mars Exploration Rover *Spirit* und *Opportunity*, die 2003 bzw. 2004 auf dem roten Planeten landeten, und ihr Vorgänger, *Sojourner*.

Doch obwohl Militär und Forschung zu



Abbildung 2.4: Mars Exploration Rover (links), Vorgänger Sojourner (rechts)

den Gebieten gehören, in denen das meiste Geld in die Entwicklung von autonomen Robotern fließt, erschliessen sie sich in den letzten Jahren auch andere Bereiche, in denen sie dem Menschen immer präsenter werden: Als Serviceroboter im Haushalt (zum Beispiel als automatischer Staubsauger), als Pflegeroboter im medizinischen Sektor oder als High-Tech Spielzeug. Im Consumernereich existieren heute Lösungen für praktisch jeden Geldbeutel, angefangen von Roboternbausätzen, die bereits für einige Euro erworben werden können, bis hin zum komplexen Haustierersatz, wie ihn Sonys *AIBO* darstellt. Inzwischen existieren sogar Maschinen, die humanoid sind, wie ein Mensch gehen und greifen können, Gesichter oder Stimmen erkennen. Noch sind deren Kapazitäten recht beschränkt, doch auch auf diesem Gebiet war die Entwicklung im letzten Jahrzehnt rasant und wird es wohl auch bleiben. Vielleicht wird es sogar möglich sein Roboter zu bauen, die zumindest intelligent erscheinen - das allerdings erfordert neue Methoden in Programmierung und Design, mit den klassischen Verfahren ist dies nicht zu erreichen.

Kapitel 3

Asuro

NACH dem vorangegangenen kurzen Abriss über Ursprung und Entwicklung der Roboter, soll nun ein Roboter etwas genauer vorgestellt werden. Dabei handelt es sich um den vom *Deutschen Zentrum für Luft und Raumfahrt* (DLR) entworfenen und von der Firma Arexx vertriebenen Roboterbausatz *Asuro*. Wegen seines geringen Preises von lediglich 50 Euro ist er gut für den Einsatz an Schulen und Universitäten geeignet, besonders um Schülern und Studenten einen ersten Einblick in die Methoden der Roboterprogrammierung zu geben.

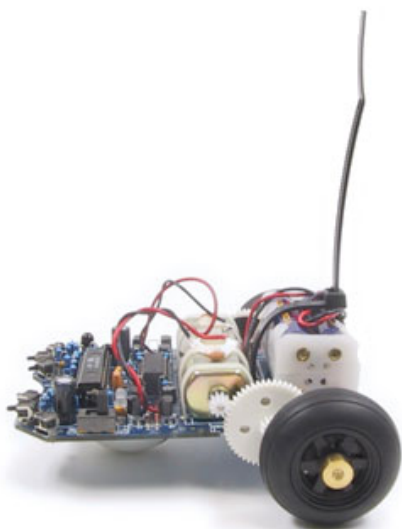


Abbildung 3.1: Roboterbausatz *Asuro*

3.1 Aufbau und Leistung

TROTZ der auf den ersten Blick wenig leistungsfähigen Hardware (siehe Anhang B, Tabelle B.1, Seite 11), steht dieser Roboterbausatz anderen, weitaus teureren Roboterbausätzen nicht nach, denn die Einschränkungen bei Steuerungselektronik und Sensorik betreffen Roboter im allgemeinen. Zum einen liegt dies – in der hier betrachteten Größeklasse von Maschinen – an dem mangelnden physischen Platzangebot für schnellere Prozessoren und größeren Speicher, die auch einen höheren Energiebedarf besitzen und somit größere Batterien nach sich ziehen, zum anderen daran, daß generell für diese einfachen Bausätze einfach keine höheren Rechenleistungen erforderlich sind. Lediglich der kleine Programmspeicher des Prozessors, der sich aufgrund der CPU-Architektur nicht erweitern läßt, setzt starke Grenzen beim Programmumfang.

Das Herzstück des Roboters stellt ein vollintegrierter RISC-Microcontroller des Typs ATmega8L dar, der ansonsten vor allem in einfachen Steuergeräten, etwa für die Klimakontrolle, eingesetzt wird. An diesen Prozessor sind zehn Sensoren angebunden, mit denen *Asuros* Programmablauf beeinflusst werden kann und mit denen er seine Umwelt wahrnehmen kann. Neben sechs einzeln ansprechbaren Drucksensoren (Tastern), stehen zwei Photorezeptoren, etwa zur Linienverfolgung, und zwei Odometriesenso-

ren in Form von Reflexlichtschranken zur Ermittlung der zurückgelegten Entfernung zur Verfügung. Desweiteren existiert auch eine Infrarotschnittstelle, über die der Roboter programmiert und gesteuert werden kann, die jedoch auch zur Übermittlung von Daten vom Roboter zu einem anderen IR-Empfänger verwendet werden kann.

Um mit seiner Umwelt zu interagieren, hat *Asuro* Kontrolle über zwei Elektromotoren, die ihn Antreiben und ihm zu flinken Bewegungen verhelfen; jeder Motor treibt jeweils ein Rad an und kann unabhängig angesteuert werden. Für die Statusübermittlung stehen dem kleinen Automaten eine zweifarbige Status-LED und zwei einfarbige LEDs auf Höhe der Räder bereit, hinzu kommt noch der IR-Transceiver, mit dem auch komplexere Daten ausgegeben werden können.

Mittlerweile gibt es sogar eine Erweiterungsplatine, die es erlaubt mit eigenen Schaltungen und Sensoren an diesem Roboter zu experimentieren. Dies ermöglicht es etwa einen Ultraschallsensor anzubringen, oder ein LCD-Display, sodaß zur Ausgabe von Nachrichten nicht auf die IR-Schnittstelle zurückgegriffen werden muß.

Damit sind für *Asuro* die Voraussetzungen gegeben auf seine Umgebung wahrzunehmen (etwa ein Hindernis zu bemerken), auf sie zu reagieren (das Hindernis zu umfahren) und somit den Programmablauf den Gegebenheiten der Umwelt anzupassen. Somit fällt er nach der Kategorisierung der JIRA (Anhang A) bereits in die Klasse der „intelligent robots“.

3.2 Software und Programmierung

HARDWARE allein ist aber nur eine Hälfte dessen, was einem Roboter die Möglichkeit zu autonomen Handeln verleiht – es ist nötig diese Hardware auch programmieren zu können. Hierfür wird bei *Asuro* standardmäßig gewöhnlicher *C*-Code verwendet,

der mit einem speziellen Compiler für Atmels AVR Embedded Prozessoren in, für den ATmega8L lesbaren, Maschinencode umgewandelt wird. Mit ein wenig Programmiererfahrung in *C* ist es schnell möglich seine ersten Programme für den kleinen Roboter zu schreiben, insbesondere da auch das Benutzerhandbuch bereits eine schnelle Einführung in die Roboterprogrammierung bietet.

Um dem Programmierer eine einfache Möglichkeit zu geben schnell leichtverständliche Programme zu erstellen, existiert bereits eine Programm-Bibliothek, die Funktionen zum Ansprechen der Motoren/ LEDs und Auslesen der von den Sensoren ermittelten Werte bereitstellt. Hierzu sei auf das Handbuch verwiesen, das neben der Entwicklungssoftware für *Asuro* auf www.arexx.com zum Download steht.

In Anhang B auf Seite 12 findet sich ein kleines Beispiel, das den prinzipiellen Aufbau und Ablauf eines Programms für *Asuro* im Besonderen, aber auch für Roboter im Allgemeinen, veranschaulichen soll. Es handelt sich um die Implementierung einer Linienverfolgung, die die Photorezeptoren nutzt, um sich an einer Kontrastkante zu orientieren und an dieser entlangzufahren.

```
#include "asuro.h"

int main(void) {
    Init();
    return 0;
}
```

Listing 3.1: *Asuro* – Initialisierung

Anstelle der ansonsten bei *C*-Programmen üblichen Einbindung des `stdio`-Headers wird hier die Datei `asuro.h` hinter der `#include`-Anweisung angegeben. Damit werden die Funktionen zum Ansprechen der Sensor-, Motor- und Signalisierungshardware des Roboters zur Verfügung gestellt. Daneben muß ein Programm für *Asuro* in der `main`-Prozedur

zwingend zunächst die Hardware per `Init()` in den Ausgangszustand versetzen, bevor andere Befehle ausgeführt werden. Ein sehr einfaches Programm, das lediglich die Hardware initialisiert, sähe aus wie in Listing 3.1.

Eine weitere Auffälligkeit ist die Hauptprogrammierschleife des Linienverfolgungsprogramms – sie terminiert nicht. Bei einem normalen Computerprogramm sind Endlosschleifen auf den ersten Blick unerwünscht; betrachtet man jedoch die Read-Eval-Print-Schleife, etwa in SCHEME, so stellt man fest, daß es, wenn es um Interaktion mit einer Umgebung geht, durchaus nicht ungewöhnlich ist, wenn keine Terminierung möglich ist (abgesehen einmal vom Abschalten des Geräts). Schleifen stellen eine wichtige Kontrollstruktur dar, die in ihrer Bedeutung in der Roboterprogrammierung sogar noch wichtiger ist als bei normalen, deterministischen Computerprogrammen.

In der Schleife wird zur Linienverfolgung ein vom Prinzip her einfaches Programm ausgeführt. Der Funktionsaufruf `LineData(daten)` liefert von den beiden Photorezeptoren ermittelten Helligkeitswerte und speichert sie als Integerwert im Array `daten`, wobei `daten[0]` den des linken, `daten[1]` den des rechten Sensors enthält ('0' steht für dunkel, '1023' für hell). Erfährt der rechte Photosensor eine dunklere Fläche, als der linke, so fährt der Robot nach links, gegenteiliges gilt den Umkehrfall. Die Richtungsänderung wird dadurch verursacht, daß der Befehl `MotorSpeed(links, rechts)`, mit dem sich die beiden Elektromotoren steuern lassen, unterschiedliche Geschwindigkeiten übergeben bekommen.

Das Beispiel sollte verdeutlicht haben, daß man mit relativ geringem Programmieraufwand und ein wenig praktischem Denken schnell ein kleines Programm schreiben kann, mit dem sich ein Roboter ansatzweise schon autonom verhält. Hat man erst einmal ein wenig Erfahrung und einige Ideen, so lassen

sich darauf aufbauend leicht anspruchsvollere Aufgaben verwirklichen.

Ist man nicht daran interessiert den Atmel-Prozessor in *C* zu programmieren, so bietet sich als Alternative dazu die *JAVA*-Lösung NanoVM an. Es handelt sich um eine sehr kompakte Virtuelle Maschine, die in das 8k große Flash-ROM des ATmega8L paßt und es so zuläßt *JAVA*-Code auf ihm auszuführen. Allerdings ist es nicht möglich die VM selber über das Infrarot-Modul zu übertragen, da NanoVM die eigentlich Firmware *Asuros* ersetzt. Deshalb ist es notwendig das Flash-ROM der CPU in einem entsprechenden Schreibegerät direkt zu überschreiben – eine Prozedur die schon einigen Aufwand erfordert, allerdings durchaus möglich ist, da der Prozessor gesockelt und nicht gelötet ist. Aufgrund des geringen Preises der Recheneinheit ist es allerdings oft sogar praktischer, direkt eine mit NanoVM beschriebene CPU zu kaufen, und einfach nach Bedarf zu wechseln. Nähere Informationen zu NanoVM findet man unter www.harbaum.org/till/nanovm/.

Obwohl der Roboterbausatz *Asuro* mehr Verständnis von Programmierung, und für den Zusammenbau vor allem Kenntnisse in Elektrotechnik. erfordert, als etwa Legos *Mindstorm*-Reihe, ist er trotzdem, gerade aufgrund seines geringen Preises, ein guter Einstieg, um einen Einblick in die Funktionsweise von Robotern zu erhalten.

Kapitel 4

Vom Marsrover zum Androiden

4.1 Ausblick

BETRACHTET man die Fortschritte, die in den letzten Jahrzehnten auf dem Gebiet der Robotik gemacht worden sind, dann sind diese zweifellos beachtlich – an einen wirklichen Androiden, einen menschenähnlichen Roboter, reichen die Entwicklungen aber bei weitem noch nicht heran.

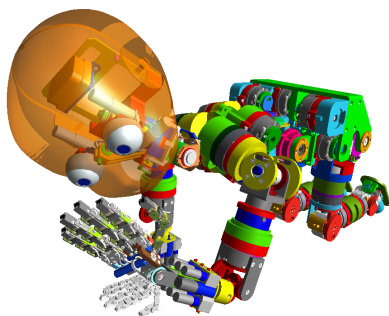
Allein der aufrechte Gang ist lange Zeit eine Herausforderung für die Avantgarde der Roboter gewesen, und erst in den letzten Jahren ist es gelungen mit Maschinen wie Hondas *Asimo* oder Kawadas *HRP-2* diese Hürde zu nehmen. Wirklich autonom sind diese Maschinen jedoch nicht, zu groß ist Ihre Abhängigkeit von der Fernsteuerung durch einen Operator, zuviel der Rechenleistung muß aufgebracht werden, um das Gleichgewicht zu behalten. Dennoch, erste Schritte sind gemacht, *Asimo* erkennt bereits selbständig Gesten, Gesichter und Stimmen, wenn auch nur eine relativ geringe Anzahl, und *HRP-2* ist in der Lage koordiniert mit einem Menschen eine Platte zu tragen.

Von einem mechatronischen Gesichtspunkt her betrachtet, mag man also sagen, daß Roboter wirklich schon eine wichtige Schwelle in Richtung Menschenähnlichkeit überschritten haben – in der Form ihres Äußeren und der Art sich zu bewegen. Auf der anderen Seite steht jedoch die Intelligenz der Maschinen. Klassische KI-Methoden sind sicher nicht in der Lage einem



Abbildung 4.1: Kawada Industries *HRP-2*

Roboter auch nur ansatzweise menschliche Intelligenz zu geben, zu stark ist er an seinen Programmablauf gebunden. Es ist also ein neues Paradigma der Roboterprogrammierung erforderlich. Das RobotCub Consortium (www.robotcub.org) mit seinem iCub könnte hierbei ein erster Schritt sein. iCub ist ein Roboter, der ähnlich einem menschlichen Kind, krabbelnd seine Umgebung erforschen soll. Dabei soll er lernen, da er von vornherein nur mit einigen wenigen Grundfunktionalitäten ausgestattet ist. Das Projekt erhofft sich hieraus Erkenntnisse einerseits über die Verfahren, wie man einen Roboter entwickeln kann, der lernt und intelligent handelt, andererseits aber auch dar-

Abbildung 4.2: RobotCubs *iCub*

über, wie menschliche Kinder lernen. Hierzu setzt es neben internationaler Kooperation – es sind 16 Partnerorganisationen aus der EU, Japan und den USA an RobotCub beteiligt – auch auf eine Open-Source-Plattform, so daß sich eine breite Masse an Forschern, auch wenn sie nicht direkt an dem Projekt beteiligt sind, ihr Fachwissen einfließen lassen können. Das Budget beträgt 8,5 Mio. Euro bei einer Projektlaufzeit von 5 Jahren.

Hauptthese des Projektes ist, daß sich Kognitionsfähigkeiten nicht durch händische Programmierung nachbilden lassen, sondern nur durch einen Entwicklungsprozeß. Dieser kognitive Entwicklungsprozeß verläuft ihrer Ansicht nach emergent im Zusammenspiel mit einem körperlichen.

Erweist sich das Projekt als erfolgreich, wäre der nächste Abschnitt der Robotereentwicklung eingeleitet. Ob aber die Mittel, sowohl finanziell, als auch technisch, die dem Projekt zur Verfügung stehen, um einen solchen Durchbruch zu erzielen, scheinen fraglich. Dennoch ist RobotCub ein Indikator dafür, daß auch in der Anwendung ein Umdenken eingesetzt hat, nämlich, daß es gar nicht so abwegig ist, daß es Verfahren gibt, um Robotern das Lernen beizubringen. Vielleicht wird es damit möglich sein Robotergehilfen und -gefährten zu erschaffen, wie sie heute nur in der Literatur auftauchen.

Anhang A

Roboterklassifikation

Roboterklassen nach JIRA

1. Manual Manipulator:

- kein Programm
 - direkt vom Bediener geführt
-

2. Fixed Sequence Robot:

- wiederholt konstant ein Bewegungsmuster
 - ändern des Bewegungsmusters ist relativ aufwendig
-

3. Variable Sequence Robot:

- wiederholt konstant ein Bewegungsmuster
 - ändern des Bewegungsmusters ist relativ problemlos
-

4. Playback Robot:

- Bewegungsablauf wird einmal durch den Bediener vorgeführt und abgespeichert
 - Bewegungsablauf beliebig wiederholbar
-

5. Numerical Control Robot:

- NC-gesteuerte Maschine
 - Informationen über Bewegungsablauf wird über Taster, Schalter oder Datenträger zahlenmäßig eingegeben
-

6. Intelligent Robot:

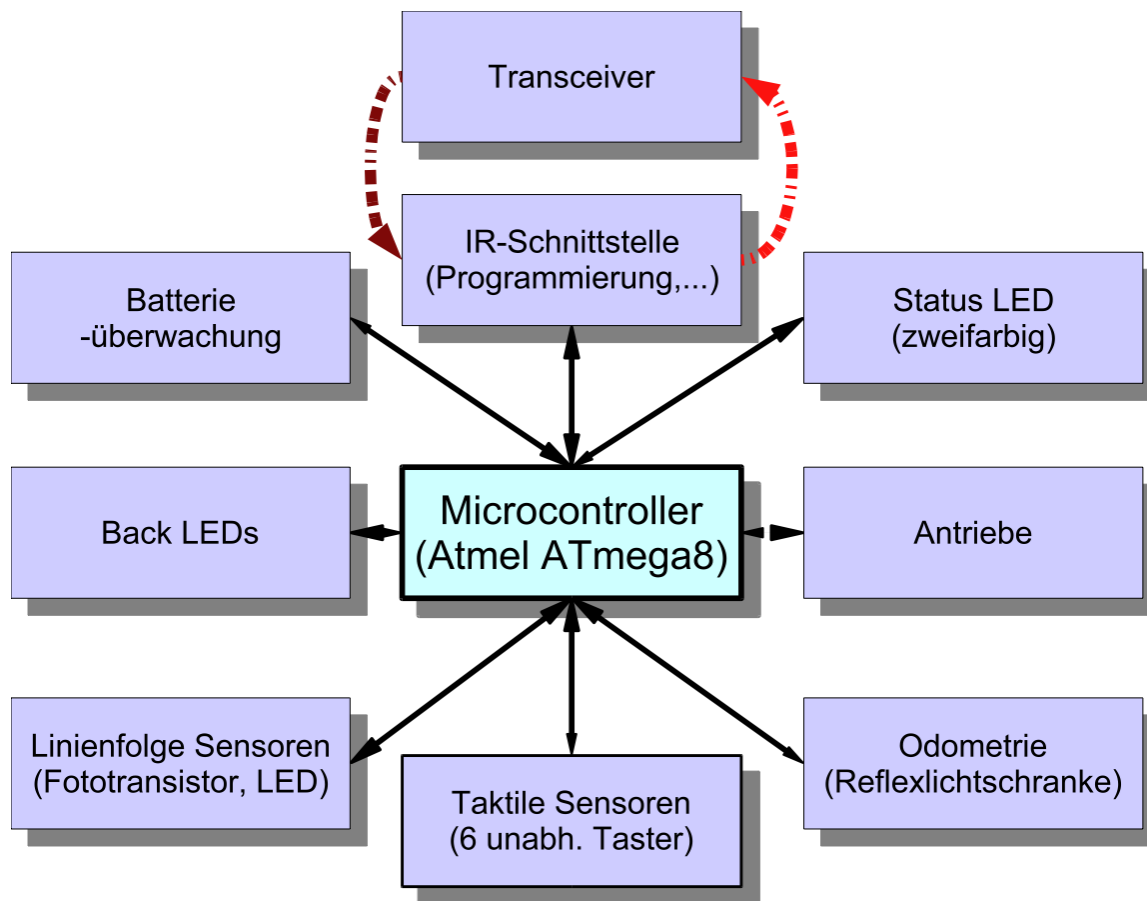
- höchste Roboterklasse
- verschiedene Sensoren zur Erfassung der Umwelt
- selbsttätige Veränderung des Programmablaufs

Anhang B

Informationen zu *Asuro*

| | |
|------------------------------|--------------------------------------------------------------------------|
| CPU | Atmel Atmega8L (8bit) |
| CPU-Takt | 8 Mhz |
| ROM | 8 kB (zzgl. 512 B EEPROM) |
| SRAM (Chip) | 1 kB |
| Bewegung | 2 Motoren, 9V 500 mA |
| Eingabe | 10 Sensoren: – 6 Bumper – 2 Fotorezeptoren – 2 Drehzahlsensoren |
| Sound | 1 Speaker |
| Timer | 2 System-Timer (8 Bit) 1 System-Timer (16 Bit) |
| Batterien Stromversorgung | 4x 1,5V 6V, ACDC |
| Kommunikation | IR-Port (Transceiver) |

Tabelle B.1: *Asuro* – Hardwareeigenschaften

Abbildung B.1: *Asuro* – Schematischer Aufbau

Asuro – Beispielcode

```

#include "asuro.h"

int main(void) {
    #define SPEED 150
    Init();
    FrontLED(ON);

    /* char ergebnis[10]; */

    int links, rechts, ADCoffset, a;
    unsigned int daten[2];

    MotorDir(FWD,FWD);

    for (a = 0; a < 254; a++) {
  
```



```

        LineData(daten);
    }
    LineData(daten);
    ADCoffset = daten[0] - daten[1];
    links = SPEED;
    rechts = SPEED;

/* Hauptprogrammschleife */

    while (1) {
        int b;

        LineData(daten);
        b=daten[0] - daten[1] -ADCoffset;

        /*
        Ausgabe per serieller Schnittstelle
        itoa(b, ergebnis, 10);
        SerWrite(ergebnis, strlen(ergebnis));
        SerWrite("\n\r", 2);
        */

        if (b > 4) {
            StatusLED(RED);
            rechts = rechts - 2 ;
            if (rechts < 1) rechts = 0;
        }
        else if (b < -4) {
            StatusLED(GREEN);
            links = links - 2 ;
            if (links < 1) links = 0;
        }
        else{
            StatusLED(OFF);
            links = SPEED;
            rechts = SPEED;
        }
        MotorSpeed(links , rechts);
    }
return 0;
}

```

Listing B.1: Asuro – Code für Linienverfolgung