

EduKaBot - Aufbau eines edukativen Roboter-Baukastensystems

Tilo Gockel, Oliver Taminé, Pedram Azad, Rüdiger Dillmann

Institut für Rechnerentwurf und Fehlertoleranz (IRF), Universität Karlsruhe,
Technologiefabrik, Haid-und-Neu-Straße 7, D-76131 Karlsruhe

{gockel, tamine, azad, dillmann}@ira.uka.de
<http://wwwiaim.ira.uka.de/~{gockel,...}>

Zusammenfassung. In dieser Veröffentlichung möchten wir unsere Herangehensweise und Spezifizierung für die Auswahl einer low-cost-Robotertechnologie für den Einsatz in Lehrveranstaltungen und Praktika erläutern. Dieses offene Roboter-Baukastensystem soll Studierende im Grundstudium Informatik und Ingenieurwissenschaften an die Technik autonomer, mobiler Roboter heranführen. Bei der Komponentenauswahl stand die Transparenz des Systems in Hinblick auf Hard- und Software im Vordergrund: so werden keine vorgefertigten Blackbox-Module eingesetzt, sondern der Student soll den langsam im Zuge eines Praktikums entstehenden Roboter von Grund auf kennen- und verstehen lernen.

1 Motivation

Der Aufbau eines Roboter-Baukastensystems für die Lehre wurde motiviert durch den Wunsch, ein neues Praktikum an unserem Institut anzubieten. Mit diesem Basispraktikum sollte sowohl der Bereich *Technische Informatik* (Digitaltechnik, Prozessortechnik, Maschinensprache ...) als auch *Grundlagen der Robotik* für Studierende technischer Fachrichtungen im Grundstudium abgedeckt werden. Ein Blick auf den Markt zeigte schnell, dass verfügbare Systeme nicht allen Anforderungen gerecht wurden und so begann die Auswahl eigener Soft- und Hardware-Komponenten und die Erstellung entsprechender begleitender Unterlagen.

Die entstandene Roboter-Architektur hat sich mittlerweile in einem Durchlauf von zehn Versuchen sehr gut bewährt. Das Konzept ist transparent und vom Studierenden gut nachzuvollziehen. Die kleinen Roboter wurden jeweils in 2er Teams von Grund auf selbst aufgebaut und somit wurden auch Meta-Inhalte wie *Teamwork*, *Induktive/Deduktive Fehlersuche*, *Bottom Up-/Top Down-Entwurf* ganz automatisch gefordert, vermittelt und angewandt.

2 Systementwurf

2.1 Spezifikation

Vor der Marktanalyse erfolgte die Auflistung aller gewünschten Systemeigenschaften. Diese seien hier in der nachfolgenden Aufstellung stichwortartig aufgeführt:

- + Unterstützung der Software-Entwicklung mittels frei verfügbarer, leistungsfähiger Tools (ANSI-C-Compiler...)
- + Nutzung einer weit verbreiteten Controller-Technologie (Dokumentation, Support, Langzeitverfügbarkeit)
- + Einfache Reproduzierbarkeit (möglichst weite Verbreitung, Aufbau einer User-Gemeinde unter den Studierenden)
- + Geringer finanzieller Aufwand (dito)
- + Einfache und kostengünstige Antriebstechnik, genauer: Differentialantrieb, kreisförmige Grundfläche der Fahrbasis, optional: Bumper-Glocke
- + Einfache Adaption verschiedener Sensortechnologien

2.2 Marktanalyse

Nach der Spezifikation erfolgte die Suche nach einem kommerziellen Produkt, welches unseren Anforderungen genügen sollte. Die untenstehende Marktanalyse zeigt die in Frage kommenden Systeme; die Auswahl ist bereits auf Radgetriebene autonome, mobile Roboter der unteren Preisklasse beschränkt (Tabelle 1).

Produkt	Controller-Technologie	Standard-Komponenten?	freier Assembler verfügbar	freier C-Compiler verfügbar	in-circuit-prog.	€	Anm.
Khepera /7/	MC68HC11	-	√	√	√	600,00	
BoeBot /4,5,6/	BASIC-Stamp	(√)	-	-	√	250,00	
LEGO Mindstorms /8/	COP8 (?)	-	-	(√)	√	250,00	
Fischer-technik Mobile Robots /9/	?	-	-	-	√	250,00	
Palm Pilot Robot Kit /10/	Palm Pilot Dragonball	-	-	-	√	315,00	Benötigt Palm Pilot
Evolution /11/	PC-basiert	-	-	-	√	500,00	Benötigt Laptop
Rug Warrior /1,12/	MC68HC11	-	√	(√)	√	580,00	Benutzt Interactive C (MIT)
Robot Kit, Myke Predko /13/	BASIC-Stamp	-	√	-	-	60,00	BASIC Stamp nicht inkl.
LynxMotion Carpet Rover /15/	BASIC-Stamp	(√)	-	-	√	150,00	

Tabelle 1: Marktübersicht zu edukativen Robotersystemen

Der Tabelle ist leicht zu entnehmen, dass keiner der dort aufgeführten Roboter alle Voraussetzungen gemäß unserer Systemspezifikation mitbringt. Weiterhin fällt auf, dass die verfügbaren Roboterbausätze zum größeren Teil sehr kostspielig sind und dies von vorneherein eine weite Verbreitung in Form einer User-Gemeinde unter den Studierenden verbietet.

Entsprechend ergab sich die Notwendigkeit, ein eigenes System zusammenzustellen. Die ausgewählten Hard- und Softwarekomponenten hierzu sind im nachfolgenden Kapitel *Komponentenauswahl* aufgeführt.

2.3 Komponentenauswahl

Die endgültige Komponentenauswahl für den Roboter-Systembaukasten ist inspiriert durch bereits auf dem Markt erhältliche Produkte, bei der Kombination der einzelnen Module gehen wir aber eigene Wege.

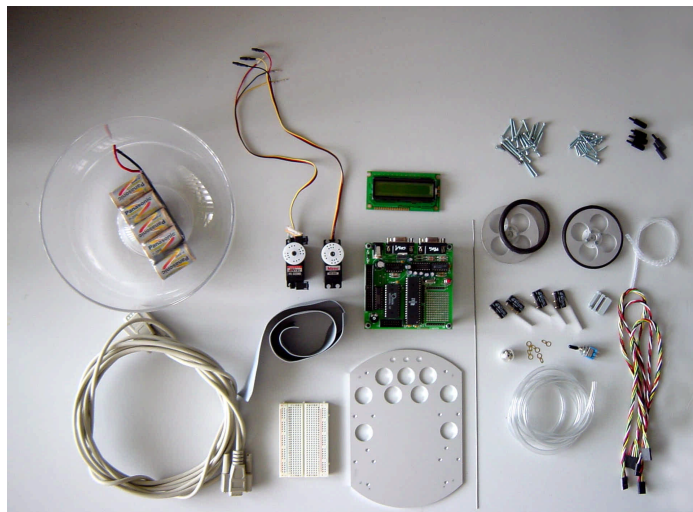


Abbildung 1: Die Hardware des Roboter-Baukastensystems

Hardware (vgl. Abbildung 1)

- 8051-kompatibler Controller mit FLASH-Programmspeicher und externem RAM auf Miniatur-Evaluationsboard. In-Circuit-programmierbar, serielle und LCD-Schnittstelle, Port-Pins herausgeführt (aktuell: Atmel 89S8252 auf Elektor-Board, /19,18/, ersatzweise auch anderes Board: /17,22/).
- Differentialantrieb mittels zweier modifizierter Modellbau-Servomotoren (LowCost-Standard-Typen, Hitec HS-300)
- NiCd-Akku-Stromversorgung, 6V, 2Ah
- Bumper-Glocke ähnlich Rug Warrior, aber mit anderer Aufhängung /1,12/
- Sensorische Erweiterungen optional (IR-Abstandssensoren, Odometrie per Optical Mouse-Chip...), auf Steckbrett aufzubauen. /1,2,3/
Teilweise Anschluss per externem seriellem A/DC (TLC549, /24/)

Software

- C-Compiler: sdcc (small device c compiler), frei verfügbar für Windows und Linux /20/
- Assembler, Linker: dito (dem Compiler unterlagert, aber auch einzeln nutzbar)
- Download-Utility: AtmelISP-Flasher, frei verfügbar /21/
- Bibliotheken (eigene Implementierung, /29/): LCD, Timer, Servo-Ansteuerung, RS232-Kommunikation, Delay-Routinen, A/DC, Mathematische Routinen mit CORDIC-Algorithmus, State Machines zur Umsetzung von Subsumptionsnetzwerken ...

Diese Zusammenstellung trägt nun allen Punkten unserer Systemspezifikation Rechnung: Der gewählte Compiler ist auch im Quelltext frei verfügbar, ist aber dennoch vergleichsweise leistungsfähig und wird bis dato auch gut gepflegt. Die 8051-Mikrocontroller-Technologie ist wohl die am weitesten verbreitete uC-Architektur der Welt, dennoch ist diese Technologie keineswegs veraltet und findet aktuell auch in leistungsfähigen DSPs Anwendung./29/ Einfache Reproduzierbarkeit ist durch die begleitenden Unterlagen gegeben, welche auch eine vollständige Teileliste mit Bezugsquelle, sowie - für Räder und Basis - CAD-Pläne für einen Nachbau enthält.

Der finanzielle Aufwand ist in der vorliegenden Version relativ hoch, da Räder und Basis in kleiner Stückzahl gefräst wurden. /23/ Den Praktikumsunterlagen liegt aber auch ein Anhang mit Ratschlägen bei, wie ein Nachbau wesentlich kostengünstiger realisiert werden kann.

Die einfache Adaption unterschiedlichster Sensortechnologien ist durch die freie Wahl des Sensor-Interfaces gegeben. So können beispielsweise Sensoren mit digitalem, seriellem Ausgang (Sharp-Entfernungssensor GP2D02 u. a., /3/) direkt mittels Port-Pins bedient werden, andere Sensoren, wie beispielsweise lichtempfindliche Widerstände, können über den gewählten A/D-Wandler TLC549 Anschluss finden. /24/

3 Aufbau des Roboter-Baukastensystems

3.1 Hardware

Das Zusammenspiel der einzelnen Hardware-Komponenten und eine genauere Erklärung hierzu werden in Abbildung 2 gezeigt. Hier ist auch gut zu sehen, dass die gewählte Bauform und Baugröße eine elegante Unterbringung der Komponenten zulässt. Dies ist nicht selbstverständlich, so hat beispielsweise bei dem Miniaturroboter Khepera /7/ der Anwender immer auch mit der Miniaturisierung der Komponenten zu kämpfen.

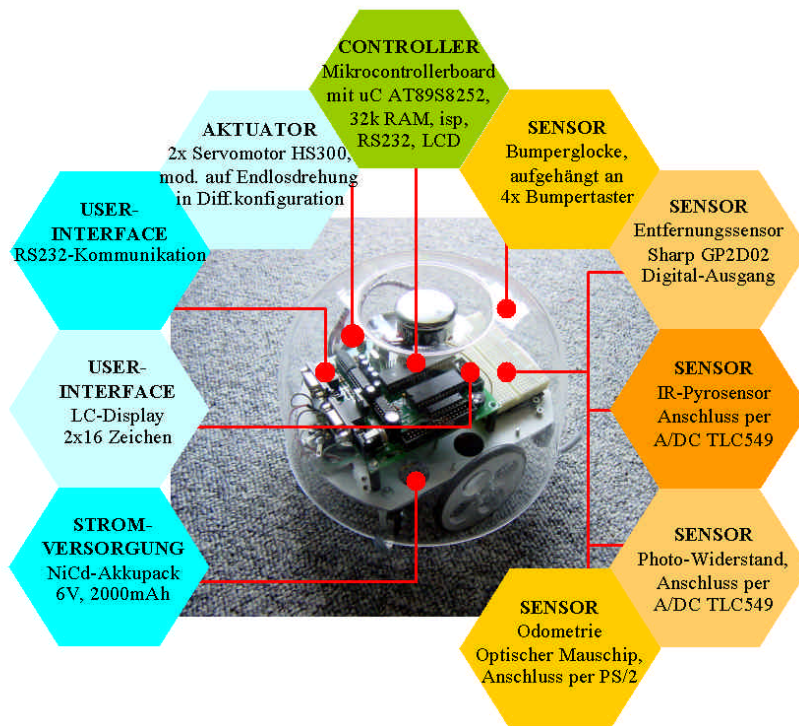


Abbildung 2: Zusammenspiel der Hardware-Komponenten

3.2 Software

Leistungsfähige freie Entwicklungswerkzeuge stehen für die Plattformen Windows und Linux zur Verfügung. Der Entwicklungsprozess ist für den Anwender vollkommen transparent, der Ablauf ist in Abbildung 3 dargestellt. /20,21/

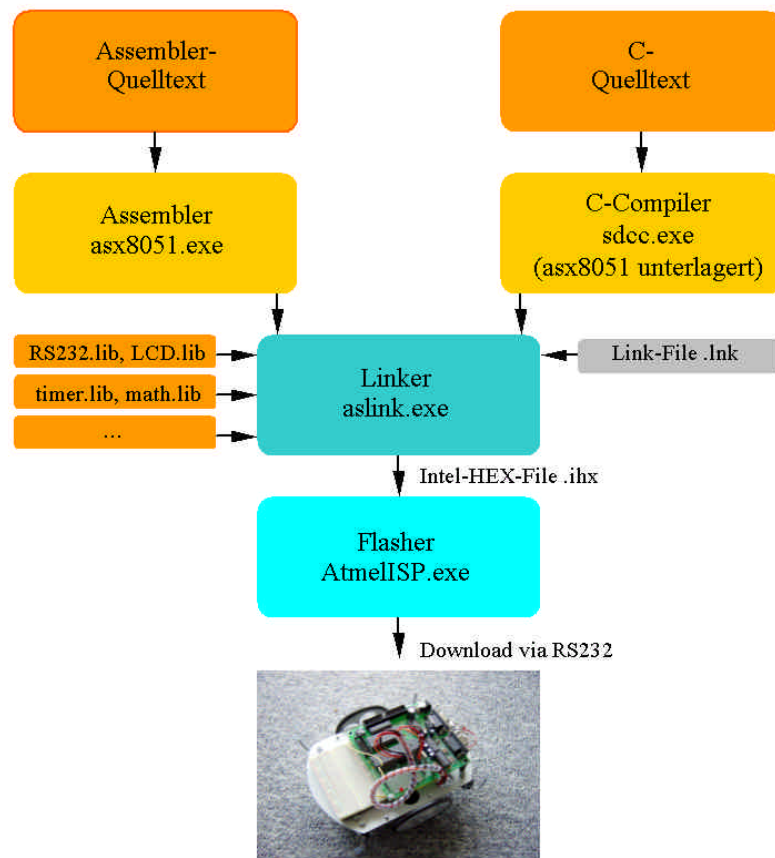


Abbildung 3: Der Software-Entwicklungsprozess

Die Bibliotheken zur LCD-Ansteuerung, zur Motorenansteuerung, zur Timer-Programmierung usw. werden von den Studierenden im Rahmen des Praktikums selbstständig erstellt, als Hilfestellung hierzu wird Literatur zum 8051 zur Verfügung gestellt. /29/

Die begleitenden Unterlagen basieren auf erprobten Beispiel-Implementierungen der Funktionen durch die Praktikumsbetreuer und bieten entsprechend hierzu eine gute Unterstützung. Besonders hervorzuheben ist hierbei für Versuch 9 und 10 die Basis für ein Reflex-basiertes Verhalten des Roboters mittels Rodney Brooks' Subsumption Architecture (Beispiel: siehe Abbildung 4). /25,26,27/

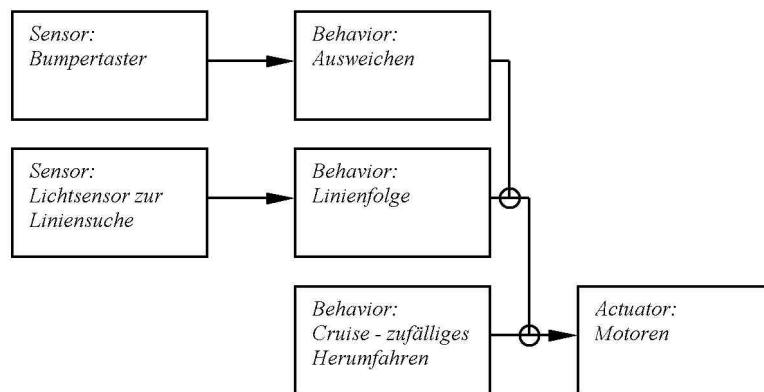


Abbildung 4: Beispiel für ein Subsumptions-Netzwerk /25, 26/

Für eine leichte Umsetzung dieser Software-Architektur wurde hierfür ein übergeordneter Automat für das Scheduling und ein weiterer (Moore-)Automat für die Bumper-Sensoren implementiert. /28/

4 Praktikumsinhalte

Die Aufteilung der Praktikumsinhalte auf zehn Versuchstermine ist in Tabelle 2 wiedergegeben.

Es fällt auf, dass, unserem ganzheitlichen Ansatz entsprechend, ein großer Teil der Zeit auf den Aufbau des uC-Boards, Implementierung der Basis-Funktionalität usw. verwandt wird.

Inhalte der Praktikumstermine (jeweils vierstündig)											
1	2	3	4	5	6	7	8	9	10	11	12
Bestückung des uC-Boards, Inbetriebnahme, Fehlersuche =====											
Installation der Entwicklungsumgebung, Lösung kleinerer Aufgaben in Assembler und C =====											
Erstellung der Bibliotheken für Ein- und Ausgabe (RS232, Timer, Servomotoren), Einbindung der Sensoren =====											
Montage des Roboters, Integration des uC-Boards, Variationen =====											
Implementierung verschiedener Ablaufsteuerungen auf dem Roboter (Labyrinth, Linienfolger...) =====											
Übergang von Ablaufsteuerung zu State Machine (quasi-parallele Verarbeitung, Subsumption) =====											

Tabelle 2: Praktikumsinhalte

Die Praktikumssteilnehmer haben zu dieser Zeit also noch keinen Roboter zur Verfügung. In der abschließenden anonymen Befragung der Teilnehmer (vgl. Kapitel 5) wurde dies aber keineswegs als demotivierend beschrieben.

5 Ergebnisse

Als Ergebnis steht mittlerweile ein leistungsfähiges System zur Verfügung, welches sich in einem Basispraktikum mit zehn Versuchsterminen und zehn Studierenden sehr gut bewährt hat. Nach Abschluss des Praktikums wurde eine anonyme Studentenbefragung durchgeführt, welche bei den relevanten Fragen „Hat mir das Praktikum Spaß gemacht?“ und „Wieviel Wissenswertes habe ich in diesem Praktikum gelernt?“ im Mittel ca. 94% , bzw. 92% geliefert hat.

Die Vermittlung des kompletten Prozesses von der Platinenbestückung, über mechanische Arbeiten bis hin zur Implementierung komplexer Zustandsautomaten zur Robotersteuerung wurde in der Befragung der Studenten als besonders positiv herausgestellt.

Weiterhin wurden zwei organisierte Wettrennen mit Preisvergabe (Hindernisparcours, vgl. Abbildung, und Linienfolge mit Ausweichbehandlung) zum Abschluss des 8., bzw. 10. Versuchstermins als besonders unterhaltsam und motivierend gelobt (Abbildung 5).



Abbildung 5: Ein Wettlauf durch einen Hindernis-Parcours

Auch wir als Betreuer des Praktikums haben subjektiv den Eindruck, dass die Studenten im Laufe der Versuchsreihe sich wesentlich verbessert haben und dies gerade darauf zurückzuführen ist, dass der jeweilige Roboter als individuelles Produkt angesehen wurde.

Aktuell laufen Arbeiten, auf dem 8051 ein kleines, schlankes Echtzeitbetriebssystem zu implementieren. Hiermit wird Multi-Threading zur einfacheren Umsetzung von Subsumptionsnetzwerken ermöglicht werden. Auch bei dieser Implementierung wird besonderes Gewicht auf einer leichten Vermittelbarkeit gelegt, der Lernende wird dieses RTOS nicht als Black Box vorgesetzt bekommen, sondern wir werden im Rahmen des nächsten Praktikumdurchlaufes diese Software zuerst motivieren und dann Stück für Stück im Quelltext vermitteln.

Die bisher vorliegenden Praktikumsunterlagen stehen für Interessenten als .pdf-Dateien unter der nachfolgenden URL zur Verfügung:

**<http://www.iain.ira.uka.de/users/tamine/bp/>
User: MobRob, Passwort: rob!98mob**

6 Literaturverzeichnis

- [1] *Mobile Roboter, von der Idee...*, Jones und Flynn, Addison-Wesley, München, 1996
- [2] *Robots, Androids and Animatrons*, John Iovine, McGrawHill-Verlag, NY, 2002
- [3] <http://www.acroname.com>
- [4] http://www.parallax.com/html_pages/robotics/boebot/boebot.asp
- [5] <http://users.etown.edu/w/wunderjt/BOEbot/robot/>
- [6] <http://www.baeg.engr.uark.edu/Academics/BAEG1012/boe-bots/rob.pdf>
- [7] <http://www.k-team.com/robots/khepera/>
- [8] <http://mindstorms.lego.com>
- [9] <http://www.fischertechnik.de>
- [10] <http://www-2.cs.cmu.edu/~pprk/>
- [11] <http://www.evolution.com/>
- [12] http://www.robotbooks.com/rug_warrior.htm
- [13] <http://engineeringbookshelf.tripod.com/isbn0071387870.htm>
- [14] <http://www.onrobo.com/reviews/Kits/Intermediate/isbn0071387870/>
- [15] <http://www.lynxmotion.com>
- [16] http://www.fh-muenchen.de/home/fb/fb06/professoren/parzhuber/Project_Start/Project_Overview/Home_pr_stud.htm
- [17] <http://www.modul-bus.de/mbnews/mbnews01/flash52.htm>
- [18] <http://www.atmel.com/products/8051/>
- [19] *Elektor, Zeitschrift für Elektronik und Computertechnik*, Ausgabe 12/2001 ff.
- [20] <http://sdcc.sourceforge.net>
- [21] <http://www.modul-bus.de/mbnews/mbnews02/isp.htm>
- [22] <http://home.arcor.de/mikronetz/>
- [23] <http://www.schaeffer-apparatebau.de>
- [24] <http://www.weltregierung.de/misc/docs/datasheets/tlc549.pdf>
- [25] *Menschmaschinen*, Rodney Brooks, Campus-Verlag Frankfurt / New York, 2002; S. 48-62, S. 260 ff
- [26] <http://www.ai.mit.edu/people/brooks/papers/AIM-864.pdf>
- [27] <http://www.cis.tugraz.at/igi/STIB/WS98/Bergmann/einleitung.htm>
- [28] http://www.ti.informatik.uni-bonn.de/html/ti1-w01_4_470-493.pdf
- [29] *Das Mikrocontroller Kochbuch MCS51*, Andreas Roth, mitp-Verlag, Bonn, 2002