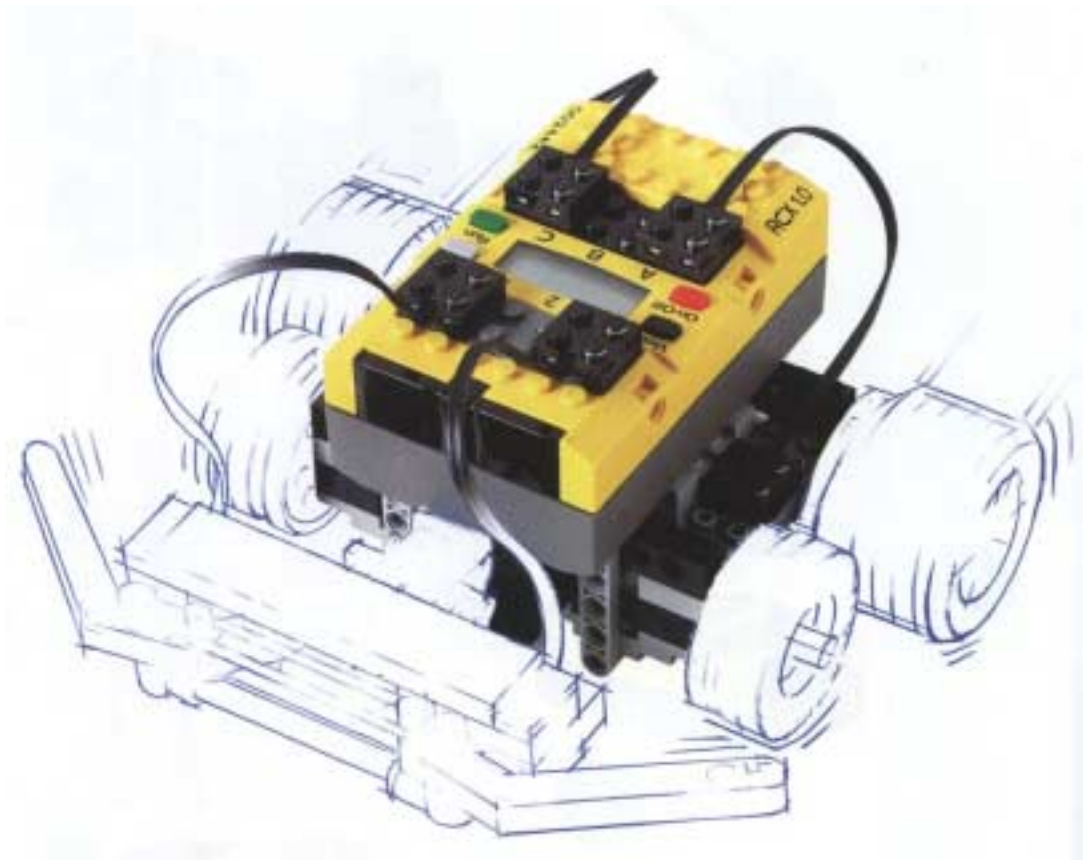


Robotik und Sensorik

Darstellungsschwerpunkt: Selbständige Entwicklung
„unscharfer“ Algorithmen zur räumlichen Orientierung
(unter Verwendung des LEGO-Mindstorms-Systems)

(Informatik / Basiskurs / Gymnasium)

Schriftliche Prüfungsarbeit zur zweiten Staatsprüfung
für das Amt des Studienrats



vorgelegt von

Michael Abend

Studienreferendar

1. Schulpraktisches Seminar im Bezirk Steglitz (S)

Berlin, den 11. September 2001

Gliederung	Seite
1. Einleitung	3
2. Robotik und Sensorik im Informatikunterricht	6
2.1 Verankerung des Themas im Rahmenplan	6
2.2 Auswahl und Bewertung von Unterrichtsmaterialien	7
2.2.1 Das LEGO-Mindstorms-System	8
2.2.2 Programmierung mit NQC (Not Quite C)	9
2.3 Algorithmen	11
2.3.1 ‚Scharfe‘ und ‚unscharfe‘ Algorithmen	11
2.3.2 Geometrische Programme	12
2.3.3 Der Regelkreis	13
3. Methoden zur Verbesserung der Selbständigkeit der Schüler	15
3.1 Handlungsorientierung und Qualifikationserwerb	16
3.2 Partnerarbeit	18
3.2.1 Abstimmung und Arbeitsverteilung	18
3.2.2 Ergebnispräsentation	20
3.3 Schüler entwickeln Programme	21
3.3.1 Programmentwurf mit Struktogrammen	22
3.3.2 Systembeherrschung	23
3.3.3 Kodierung	24
3.3.4 Testen und Programmänderung	25
4. Unterrichtsvoraussetzungen	26
4.1 Allgemeine Unterrichtsvoraussetzungen	26
4.2 Spezielle Unterrichtsvoraussetzungen	26
4.2.1 Der Unterricht im ersten Jahr Informatik	26
4.2.2 Die Lerngruppe	27
4.2.3 Materielle Ausstattung	29
5. Planung der Unterrichtsreihe	30
5.1 Reduktion und Schwerpunktbildung	32
5.2 Ziele des Unterrichts	33
5.3 Synopse	36
6. Darstellung und Analyse des Unterrichts	37
6.1 Begründung zur Auswahl der Stunden	37
6.2 Analyse	37
6.2.1 Der Programmentwurf	38
6.2.2 Testen und Verbessern	39
6.2.3 Präsentation	40
7. Gesamtreflexion	40
7.1 Einsichten	40
7.2 Motivation	41
7.3 Partnerarbeit	41
7.4 Methodentraining	42
7.5 Schülerprodukte	44
7.6 Was noch zu tun wäre	45
8. Literaturverzeichnis	46
9. Anhang	48

1. Einleitung

Seit sich Menschen mit Fragen zu ihrer eigenen Entstehung beschäftigen, tauchen immer wieder Ideen über die Erschaffung künstlichen Lebens (KL) in Literatur und Film¹ oder in Wissenschaft und Technik auf. *„KL ist der Versuch, den uralten Traum der Menschen zu verwirklichen, das Phänomen Leben vollständig zu beherrschen und die Erzeugnisse der eigenen Schöpfungskraft in autonome Wesen zu verwandeln.“* BACHMANN (S.102). Mechanisch gesteuerte Figuren waren die Vorläufer jener Geräte, die heute mittels moderner Computertechnik funktionieren und als ‚Roboter‘ bezeichnet werden.

„Der Begriff Roboter stammt von dem tschechischen Wort ‚robota‘, das so viel wie „Zwangsarbeit“ bedeutet. Er wurde zuerst 1921 in dem Stück R.U.R. (Rossum’s Universal Robots, deutscher Titel: Werstands Universal-Roboter) des tschechischen Romanschriftstellers und Dramatikers KAREL CAPEK verwendet.“ (ENCARTA ENZYKLOPÄDIE 1999). Nach KNUDSEN, NOGA (S.2) *„ist ein Roboter eine Maschine mit programmierbarem Verhalten“*, die aus den Komponenten Steuerungsinstanz, Körper bzw. Chassis, Aktoren, Sensoren und Energiequelle besteht.

Der Einsatz der Robotertechnik geht mittlerweile über Montageroboter in der Automobilindustrie weit hinaus. So finden sich in der Medizintechnik, in der Luft- und Raumfahrtindustrie² oder der Bautechnik Anwendungen für Roboter. Sogar in manches Kinderzimmer haben durch Sensoren, Motoren und Mikroprozessor

¹ „Golem (hebräisch: ungeformte Masse), der jüdischen Legende und Mystik zufolge ein künstlicher Mensch ohne Stimme, dem durch eine magische Spruchformel aus dem *Buch der Schöpfung (Sefer Jezira)* Leben eingehaucht werden kann.“ (ENCARTA Enzyklopädie)

Thematisiert wird das Thema ‚künstliches Leben‘ auch in M. SHELLEYS *Frankenstein* oder in S. SPIELBERGS neuem Film ‚KI‘.

² Ein berühmtes Beispiel hierzu war der Miniroboter ‚Sojourner‘ der US-Raumsonde ‚Pathfinder‘, der Bilder von der Marsoberfläche zur Erde funkte (vgl. ALTENBURG, ALTENBURG, S. 10).

gesteuerte Spielzeuge (z.B. die ‚Furby‘-Figuren³) ihren Einzug gehalten.

Warum es Menschen faszinierend finden, scheinbar lebendige Wesen zu konstruieren, liegt nach Ansicht von PAPERT (S.195) daran, dass *„Technologie als Medium für die Darstellung von Verhaltensweisen dient, die man bei sich selbst und anderen beobachten kann.“* So beschreiben DECHAU, C.-P. und PSIHOYOS, L. (S.47), wie Kleinstroboter nach tierischen Vorbildern entwickelt wurden *„..., indem sie in ihren ‚Insekten-Robotern‘ Sensoren und Aktoren stets direkt miteinander koppelten.“*

Um aber Verhaltensweisen nachbilden zu können, ist zuerst ein Analyseschritt notwendig. Es muss sowohl eine physische Komponente, welche Bewegung oder Aktion ein Roboter ausführen soll, als auch eine kognitive Komponente, das Sammeln von Informationen und deren Auswertung in Form von Entscheidungen, berücksichtigt werden.

Für Schüler stellt die Beschäftigung mit Robotersystemen neben der reinen informatischen Komponente einen fächerübergreifenden Lerngegenstand dar, da man das Regelkreisprinzip oder Rückkopplungsmechanismen auch in biologischen oder physikalischen Sachverhalten wiederfindet.

PAPERT (vgl. S.196) vergleicht den Vorgang der Konzeptionierung eines Roboters mit dem kreativen Akt beim Malen eines Bildes und sieht einen großen Unterschied zum herkömmlichen lehrgangsartig organisierten Unterrichtsstil in den Naturwissenschaften. Auch wenn in der vorliegenden Unterrichtsreihe aus Kostengründen lediglich ein einziges bereits vorgefertigtes Robotermodell eingesetzt werden kann, handelt es sich dennoch im Sinne von JANK, MEYER (vgl. S.355f.) um ein ganzheitliches Beschäftigen mit dem Unterrichtsgegenstand, bei

³ Diese erschienen 1998 bei Tiger Electronics Ltd; Illinois (USA). Es handelt sich um ‚sprechende‘ Plüschpuppen, welche auf Berührung, Licht und Geräusche reagieren.

dem alle Sinne angesprochen werden, da der Roboter anders als bei Software-Simulationen⁴ physisch präsent ist.

Da die Schüler im Laufe der Entwicklung ihrer kybernetischen Programme wiederholt und unmittelbar mit den realen Ergebnissen ihrer Arbeit in Form von Roboterbewegungen konfrontiert werden, möchte ich die folgende These in dieser Arbeit gesondert prüfen:

Die praktische Erfahrung bei der Umsetzung theoretischer Lösungsansätze unter den realen Bedingungen einer Robotersteuerung verhilft den Schülern zu einem erweiterten Verständnis des Algorithmusbegriffes.

Ich bin mir dabei zwar der Schwierigkeit bewusst, messbare Indikatoren für diesen Erkenntnisprozess bei den Schülern zu bestimmen, aber ein langfristiger Erfolg meines Vorhabens sollte sich spätestens bei den größeren Softwareprojekten der nächsten beiden Schuljahre im Informatikunterricht (und darüber hinaus) zeigen. Ob sich die Idee der ‚unscharfen‘ Algorithmen auf dem von mir gewählten Weg transportieren lässt, soll einerseits durch mitprotokollierte Schüleräußerungen zu belegen sein und andererseits aus den Schülerprodukten ersichtlich werden.

Im folgenden Kapitel wird in einer Sachanalyse der Themenbereich Robotik für den Informatikunterricht passend eingegrenzt und die Auswahl geeigneter Algorithmen begründet werden. Daran schließt sich die Beschreibung der verwendeten Methodik an. Damit folge ich der Darstellung LEHMANN'S (S.204): „*Beim projektorientierten Unterricht gehört die Methode mit zum Inhalt.*“ Im zweiten Teil der Arbeit stelle ich die Planung und Auswertung derjenigen Unterrichtsphasen vor, in denen selbständiges Handeln der Schüler weiter entwickelt werden soll.

⁴ Typische Vertreter dieser simulierten Roboterwelten sind zum Beispiel die ‚LOGO-Turtle‘, ‚Niki‘ oder ‚Karel‘.

2. Robotik und Sensorik im Informatikunterricht

Um die Relevanz des Themas ‚Robotik‘ für die Schule bestimmen zu können, muss zuerst eine Einordnung dieses Teilgebietes innerhalb der Informatik vorgenommen werden. So findet man bei RECHENBERG (S.211) die Robotik im Themenbereich der ‚Angewandten Informatik‘: *„Zwei weitere Arten der Automatisierung, in denen der Rechner eine Rolle spielt, sind die Robotik und die Steuerung und Regelung technischer Prozesse.“* Damit ist gleichsam schon eine Unterordnung der Robotik innerhalb der Automatisierung und eine Abgrenzung zur Regelungstechnik beschrieben.

Einen gewichtigen Grund für die Behandlung in der Schule sehen ABEL, KÜPER in der Anschaulichkeit des Themas und stellen fest: *„Der Industrieroboter als hochtechnisierte computergesteuerte Maschine bietet nach erfolgter Reduktion ein überschaubares technisches System, das in seiner praktischen Bedeutung unmittelbar verständlich ist und dessen gesellschaftliche Relevanz für einen modernen Industriestaat schnell zu erfassen ist.“* (S.37)

2.1 Verankerung des Themas im Rahmenplan

Da sich das Thema Robotik nicht konkret im Berliner Rahmenplan Informatik wiederfindet, möchte ich schon an dieser Stelle die Themenwahl legitimieren. Für das erste Unterrichtsjahr Informatik sieht der Berliner Rahmenplan im Fach Informatik einen Schwerpunkt in der *„Konstruktion von Teilalgorithmen zu Anwendungsfällen“* (SENATSV ERWALTUNG FÜR SCHULE, BERUFSBILDUNG UND SPORT, S.11) mit einem Umfang von 35 Stunden vor. Um den Schülern kein verzerrtes Bild der Informatik zu präsentieren, werden in der Konzeption realitätsnahe und hinreichend komplexe Programmbeispiele gefordert. Diese sollen *„typische Einsatzbereiche repräsentieren“* (S.12) und möglichst *„Themen zur Anwendung des Computers, zu Auswirkungen seines Einsatzes und zur Rechnerorganisation“* (S.13) integrativ behandeln. Weiterhin wird eine Verwendung von Software-Bibliotheken von Anfang an empfohlen. All

diese Forderungen des Rahmenplans entspricht das Thema Robotik und Sensorik in fast idealer Weise.

Drei Lernziele bzw. Lerninhalte, in denen es besonders auf selbständiges Arbeiten ankommt, und die in dieser Unterrichtsreihe eine wichtige Rolle spielen, sollen hier angesprochen werden:

- *Algorithmen in einer programmiersprachen-unabhängigen Form darstellen können;*
- *Umsetzung des Entwurfes in die verwendete Programmiersprache unter Einführung der benötigten Sprachelemente;*
- *Programmtests durchführen und bewerten, sowie eventuell auftretende inhaltliche Fehler des Algorithmus und Notationsfehler beseitigen können.*⁵

2.2 Auswahl und Bewertung von Unterrichtsmaterialien

Ein praxisorientiertes und auf selbständiges Handeln der Schüler angelegtes Unterrichtsvorhaben zum Thema Robotik verlangt nach einem praktikablen Robotermodell. In der Zeitschrift LOGIN wurden in zwei Schwerpunktausgaben (5-6/88 und 3/95) verschiedene Robotermodelle bzw. Baukastensysteme für Unterricht vorgeschlagen. Ein dicht am Industrieroboter orientiertes Fertigmodell ‚Teach-Robot‘ der Fa. Kalms GmbH stellte einen Greifarm dar, der mittels serieller Schnittstelle mit maschinennahen Befehlen zu programmieren war⁶. Die Computing Baukästen von Fischertechnik, die schon Ende der 80er Jahre erhältlich waren, stellten ein flexibleres Konzept zur Verfügung. Neben vorgeschlagenen Modellen, wie z.B. Ampelsteuerung, Solarzellen-Nachführung, Waschmaschine oder ‚Turtle‘ konnten auch eigene Konstruktionen oder Veränderungen gemacht werden. Die ‚Turtle‘⁷ verkörperte einen mobilen Roboter, welcher jedoch, wie alle anderen Modelle auch, über ein Interface per

⁵ Zit. Senatsverwaltung für Schule, Berufsbildung und Sport, S.16f.

⁶ Vgl. ABEL, KÜPER, S.38f.

⁷ Vgl. CVK FISCHERTECHNIK, S.9

Kabel mit einem Rechner zu verbinden war und somit leider in seiner Mobilität eingeschränkt blieb. Die Programmierung der Modelle erfolgte mit Basic oder Turbo-Pascal, welches an vielen Schulen als Programmiersprache im Informatikunterricht eingesetzt wird. BOEGNER u.a. (S.46) berichten aber auch hier von Unzulänglichkeiten: *„Die Schüler setzen ihr Pascal-Programm in ein Basic Programm um und binden dieses in ein vorhandenes Rahmenprogramm ein.“*

Ein ähnliches Baukastenkonzept mit einem parallelen angesteuertem Interface bot zur gleichen Zeit LEGO mit den Lego-Lines auf dem Markt an. Es verfügte über eine spezielle sprachenunabhängige Programmierumgebung⁸. Diese wurde in der Folge um eine Programmierumgebung für Turbo-Pascal ergänzt, welche laut SCHRÖDER (S.30) auf das *„Einsatzziel der Informatikunterricht in der Sekundarstufe II ...“* abzielt.

2.2.1 Das LEGO-Mindstorms-System

Eine Weiterentwicklung des Baukastensystems von LEGO fand in den 90er Jahren am Massachusetts Institut of Technologie unter Federführung von Seymour Papert statt. Ziel der Arbeitsgemeinschaft ‚Erkenntnistheorie und Lernforschung‘ war die Entwicklung eines autonomen Roboterkonzepts. Dazu wurde ein Minirechner inklusive Speicher, Display, sowie Ein- und Ausgängen in einem Gehäuse integriert. Dieser ‚RCX‘ (Robotic Command Explorer) genannte Baustein ist über eine Infrarotschnittstelle vom PC aus zu programmieren.

LEGO liefert seit 1998 den Baukasten ‚Mindstorms RIS‘ für Kinder ab 11 Jahren aus. Dieser enthält ca. 700 Lego-Bauteile der Technic Serie, den ‚RCX‘, zwei Motoren, zwei Berührungssensoren, einen Helligkeitssensor, eine Infrarotschnittstelle, ein Handbuch und eine grafische Programmiersoftware ‚RCX Code‘.

⁸ Vgl. THODE, S.91ff., LEGO TC Newsletter Nr.1/88



Abbildung I: Helligkeitssensor, Berührungssensor und Motorenbaustein des Mindstorms-Baukasten.

Alternativ wurden Schulbaukästen mit einer anderen ebenfalls grafischen Software Namens ‚Robolab‘ für den ‚RCX‘ angeboten⁹.

Die Auswahl dieses Baukastensystems gründet sich somit auf folgende Vorteile:

- Die Lego-Bausteine sind vielen Kindern und Jugendlichen bekannt;
- die Baukästen und somit die Modelle sind flexibel erweiterbar;
- man kann mit Hilfe des ‚RCX‘ autonome Roboter konstruieren;
- es gibt aufgrund der öffentlich zugänglichen Schnittstellenbeschreibung alternative Programmierumgebungen für den ‚RCX‘ und
- über das LEGO-Mindstorms-System sind wegen seines großen Verkaufserfolges zahlreiche Publikationen in Buchform und im Internet erschienen.

2.2.2 Programmierung mit NQC (Not Quite C)

Die Programmierschnittstelle des ‚RCX‘ wurde von der LEGO-Entwicklungsgruppe im Internet veröffentlicht. Dies führte zu einer ganzen Reihe inzwischen im Internet verfügbarer Programmiersprachen für das LEGO-System.

„The LEGO Group hat das Robotics Invention System für Leute geschaffen, die noch nie programmiert haben. Für diese Zielgruppe ist RCX Code ein sanfter Einstieg in die Programmierung mobiler Roboter.“¹⁰ Da die Schüler am Ende des ersten Unterrichtsjahres aber bereits Programmiererfahrung mit der prozeduralen und objektorientierten Programmiersprache ‚Python‘ gemacht haben, liegt die Verwendung einer vergleichbaren Sprache nahe. Eine Tabelle auf

⁹ Vgl. LEGO [2000], S.10f.

¹⁰ Zit. KNUDSEN, NOGA, S.55

Seite 254 des Buches von KNUDSEN/Noga zeigt die verfügbaren Sprachen und Entwicklungsumgebungen sowie deren Fundstellen im Internet.

Eine davon ist NQC (**Not Quite C**), die an die Programmiersprache ‚C‘ angelehnt ist und Erweiterungen zur RCX-Programmierung enthält.

Zwei große Vorteile von NQC sind erstens die gute System-Dokumentation von Dave BAUM, dem Autor der Sprache selbst¹¹ und zweitens die einfach zu bedienende und funktionale Windows-Entwicklungsumgebung¹² von Marc Overmars.

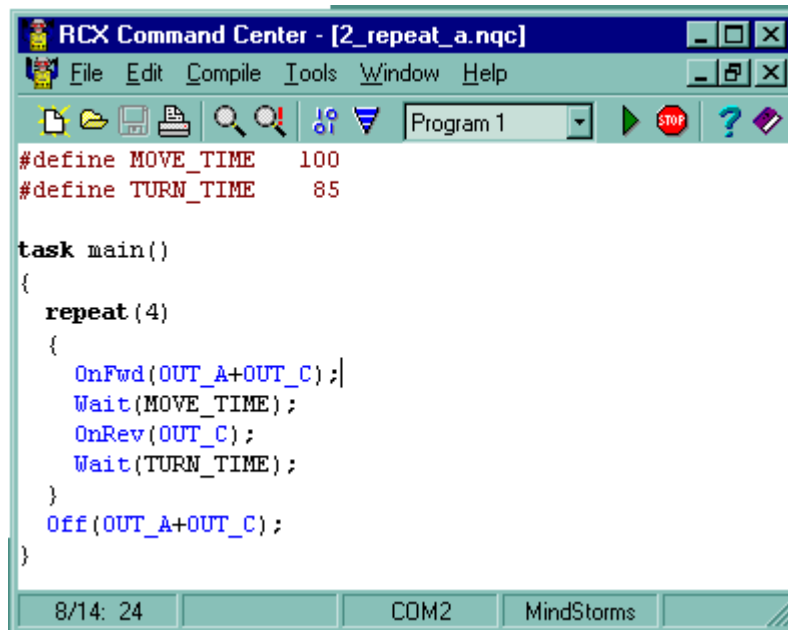


Abbildung II: Haupt-Fenster des RCX-Command-Center mit einem Beispielprogramm für ein ‚geometrisches‘ Roboterprogramm in NQC.

Sehr positiv für die Programmentwicklung durch Schüler erweist sich der automatische Fettdruck von Schlüsselwörtern der NQC-Syntax und die farbliche Hervorhebung von Bibliotheksfunktionen (blau) und Kommentaren (rot).

¹¹ Vgl. BAUM, D: NQC Programmieranleitung.

¹² Von der Universität Utrecht: <http://www.cs.uu.nl/people/markov/lego/RCXCommandCenter> Version 3.1.

2.3 Algorithmen

Nachdem einiges zur eingesetzten Hard- und Software der Unterrichtsreihe gesagt wurde, nun zu den zu erstellenden Schülerprodukten, den Programmen. Ihre Grundlage sind Algorithmen, welche auch losgelöst vom Robotermodell betrachtet werden können.

Eine Definition des Begriffes soll die wesentlichen Aspekte verdeutlichen: *„In der Informatik versteht man unter einem Algorithmus eine endliche Anzahl von eindeutigen und ausführbaren Anweisungen. In den Bearbeitungsschritten werden einzelne (...) einfache Operationen ausgeführt. Unter gleichen Voraussetzungen und Bedingungen führt ein Algorithmus stets zum gleichen Ergebnis.“*¹³

Da es im Folgenden um jene Verfahren geht, welche mobile Roboter durch die Ausführung einer Abfolge von Einzelschritten bewegen, sollen zwei unterschiedliche Lösungsstrategien dargelegt werden. Beide Konzepte werden in der Unterrichtsreihe berücksichtigt.

Zur Veranschaulichung der Algorithmen sollen Struktogramme verwendet werden, welche die Schüler im ersten Unterrichtsjahr als Arbeitsmethode kennen gelernt haben.

2.3.1 ‚Scharfe‘ und ‚unscharfe‘ Algorithmen

Die Unterscheidung der beiden Ansätze ‚scharfe‘ und ‚unscharfe‘ Algorithmen stellt den erkenntnistheoretischen Kern dieses Unterrichtsvorhabens dar. PAPERT (S.198) wird zwar in der deutschen Fassung seines Werks mit den Begriffen ‚genau‘ und ‚ungenau‘ übersetzt. Ich glaube jedoch, dass die in der Überschrift gewählten Adjektive besser den Unterschied beschreiben, um den es geht: *„Herkömmliche Erkenntnistheorie ist eine Erkenntnistheorie der Genauigkeit: Wissen wird danach beurteilt, ob es genau ist, und als*

¹³ Hahn, Dzewas: Informatik für die Sekundarstufe II; Band 2; Westermann; Braunschweig 1991; S.8.

minderwertig betrachtet, wenn es ungenau ist. Kybernetik schafft eine Erkenntnistheorie der ‚beherrschten Ungenauigkeit‘.“¹⁴

Die Grundlagen dieses Theorieansatzes stammen vom Begründer der Kybernetik, Norbert Wiener, und Vertretern anderer Wissenschaftsfelder, wie z.B. dem Anthropologen G.Bateson, welcher das Rückkopplungsprinzip auf die Erklärung menschlichen Verhaltens anwendete.

Die Verschiedenartigkeit beider Ansätze soll an Beispielen zur Robotersteuerung verdeutlicht werden.

2.3.2 Geometrische Programme

Die zu lösende Aufgabe besteht darin, den Roboter ein quadratisches Hindernis umfahren zu lassen, wobei die Startposition des Roboters festgelegt ist. Ein naheliegender Ansatz könnte folgendermaßen aussehen:

1. Das Hindernis abmessen (z.B. 100 Längeneinheiten)
2. Probiere das Programm:

wiederhole 4 mal (fahre 100 einheiten vor, drehe 90 grad nach rechts)

Dieser Versuch kann funktionieren, aber: *„Zuerst lässt sich feststellen, dass dies ein Fall ist, bei dem man Gefahr läuft, schrecklich unrecht zu haben, wenn man zu sehr genau recht hat. Das Programm ... lässt keinen Spielraum für Fehler....In der Praxis würde es beinahe sicher versagen.“*, denn: *„Fehler sind ein universelles Merkmal der Welt,...“¹⁵*.

Kleinste Ursachen, wie z.B. eine schräge Unterlage oder schwächer werdende Batterien des Roboters könnten dazu führen, dass er sich nicht mehr exakt um 90 Grad dreht und in der Folge mit dem Hindernis kollidiert.

Für die Schüler wird die erste praktische Aufgabe lauten, ein solches Programm zu entwerfen und die Fehleranfälligkeit sowie ihre Ursachen zu identifizieren. Denn Programme, die Roboter lediglich am Bildschirm simulieren, „unterschlagen“ reale Fehlerquellen.

¹⁴ Zit. PAPERT, S.198

¹⁵ Zit. PAPERT, S.200

2.3.3 Der Regelkreis

Die Basis von Steuerungs- und Regelungsvorgängen in Natur und Technik ist das Modell des Regelkreises. Es beschreibt ein sich selbst regulierendes System, welches durch negative Rückkopplung Ein- und Ausgänge verknüpft. Die Rückkopplung wird deswegen als ‚negativ‘ bezeichnet, weil einem unerwünschten Istwert am Eingang Steuerungsmaßnahmen des Reglers am Ausgang über ein Stellglied entgegengewirkt, um den Sollwert wieder zu erreichen. Die Eingänge des Reglers werden über Messfühler bzw. Sensoren mit Informationen versorgt. Das LEGO-Robotermodell verfügt über einen Helligkeitsfühler und einen Berührungssensor. Die Ausgänge enthalten entsprechend den im Regler getroffenen Entscheidungen Informationen für die Stellglieder, welche im Fall des LEGO-Roboters durch zwei Motoren repräsentiert werden. Der ‚RCX‘ Baustein und das in ihm enthaltene Programm übernimmt die Funktion des Reglers. Ein auf diesem Prinzip beruhender Algorithmus für den Regelkreis könnte dann so aussehen:

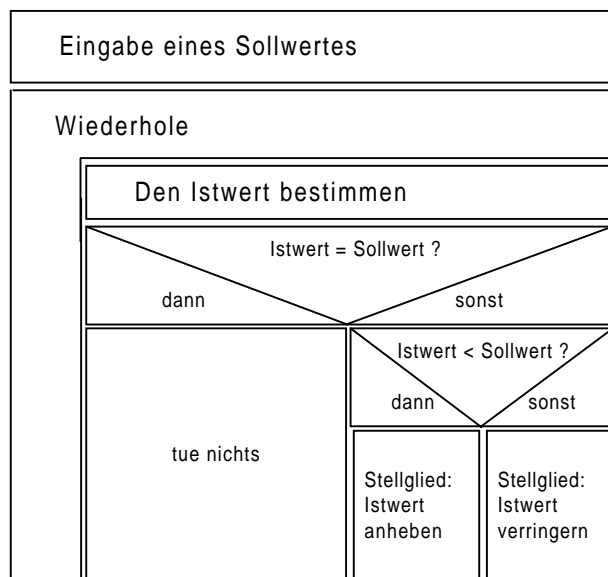


Abbildung III: Struktogramm für den allgemeinen Regelkreisalgorithmus

Ein mit dem Rückkopplungsprinzip arbeitender alternativer Algorithmus für das Hindernis-Problem hätte als Sollwert ‚Sensor

unberührt' vorgegeben. Falls dies zutrifft, fährt der Roboter geradeaus, falls nicht (Zustand ‚Sensor berührt‘), dreht er sich ein Stück z.B. nach links¹⁶, bis der Sollzustand wieder eintritt. *„Die auffälligste Eigenschaft dieses Programms ist seine Unbestimmtheit in bezug auf Größe und Form der Schachtel.“*¹⁷ Beide Werte tauchen im Programm nicht auf. *„Der kybernetische Ansatz ist robust und sehr allgemein: Er wird bei fast allen Gegenständen, die umgangen werden müssen, immer funktionieren.“*¹⁸ Außerdem entspricht dieses Vorgehen auch eher dem Verhalten eines Menschen, wenn dieser mit verbundenen Augen vor ein vergleichbares Problem gestellt werden würde.

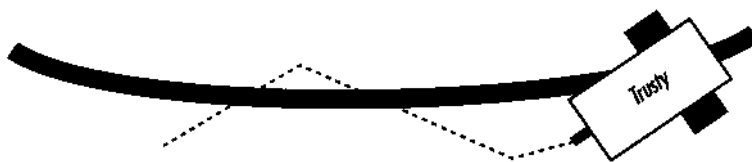


Abbildung IV: Schematische Darstellung eines möglichen Fahrweges von ‚Trusty‘ (hier der Name des mobilen Roboters) von oben aus gesehen (aus: KNUDSEN/NOGA, S.46).

Als anspruchvollstes Problem der Reihe wird ein ebenfalls nach dem Prinzip des Regelkreises arbeitender Algorithmus erarbeitet, bei dem der mobile Roboter mit Hilfe seines Helligkeitssensors einer dunklen ungeradlinigen Markierung auf einer Tischoberfläche folgen soll. Es ergeben sich dabei zwei neue Probleme:

- Der Helligkeitssensor besitzt nicht nur zwei Zustände (berührt / nicht berührt) wie der Berührungssensor, sondern liefert diskrete Helligkeitswerte;
- der Roboter besitzt keine Information darüber, ob er sich links oder rechts von der Markierung befindet, wenn er diese verlassen hat.

Für das erste Problem genügt es, den Sensor durch die Verwendung entsprechender Konstanten zu eichen.¹⁹ Das zweite ist deutlich

¹⁶ Die beiden Fahrtmotoren für den links- und rechtsseitigen Antrieb des Roboters übernehmen die Aufgabe des Stellgliedes.

¹⁷ Zit. PAPERT, S.201

¹⁸ Zit. PAPERT, S.202.

¹⁹ Das könnte in NQC z.B. so aussehen: **#define Schwelle_dunkel 35**

schwieriger zu lösen. In einer Variablen muss die Richtung der letzten Kurskorrektur gespeichert werden. Wenn sich der Roboter vorher nach links bewegt hat, so muss er wahrscheinlich beim nächsten Korrekturversuch nach rechts korrigieren, um die Markierung wieder zu finden²⁰.

Ein nach geometrischer Vorgehensweise ohne Helligkeitssensor arbeitender Algorithmus wäre für dieses Problem nur schwer realisierbar, wahrscheinlich sehr fehleranfällig und zudem nicht universell verwendbar, da er nur für eine vorher ausgemessene Markierung funktionieren würde. Die Messungen wären sehr aufwendig und die Genauigkeit der Startposition ausschlaggebend für eine der Markierung exakt folgenden Fahrt des Roboters.

3. Methoden zur Verbesserung der Selbständigkeit der Schüler

Der Begriff ‚Selbständigkeit‘ ist in aller Munde. STRUCK (S. 151) stellt hierzu fest: *„Selbständigkeit ist mittlerweile eine derart wichtige Schlüsselqualifikation, dass sie in neuen Schulgesetzen, in Lernzielkatalogen und in erzieherischen Richtlinien eigentlich immer an erster Stelle genannt wird.“* Denn sie sei *„...nicht nur eine Schlüsselqualifikation, die in Arbeits- und Wirtschaftswelt, in Studium und Beruf sowie in der Freizeit nützt, sondern sie zahlt sich bereits beim Lernen in der Schule in Form größerer Lerneffekte aus.“*²¹ Als Teilbereiche der allgemeinen Eigenschaft ‚Selbständigkeit‘ nennt STRUCK die Methoden:

- Sich gezielt Informationen beschaffen zu können;
- anhand dieser Informationen selbständig urteilen zu können;
- begründete Entscheidungen zu treffen;
- selbständig gemäß der eigenen Entscheidungen zu handeln.

Da sich jedoch die Beherrschung dieser Methoden bei den Schülern nicht von alleine einstellt, fordern JANK, MEYER (S.371): *„Wenn die*

²⁰ Vgl. KNUDSEN, NOGA, S.45f.

²¹ Zit. Struck, S.151.

Erziehung zur Selbständigkeit ein übergeordnetes Ziel der Schule ist, dann muß sie im Unterricht geübt werden.“ Um diese Methoden zu üben, sollte man „...den Schüler als handelndes Subjekt im Unterricht akzeptieren und fördern...“²². Denn: „In dem Maße, wie sich sein Methodenrepertoire erweitert und festigt, wächst auch seine Selbststeuerungs- und Selbstbestimmungsfähigkeit – und damit seine Mündigkeit.“²³

Im folgenden Kapitel soll erläutert werden, wieso sich gerade die Vermittlung der oben beschriebenen Algorithmen mit einem Robotermodell als Training für selbständiges Lernen eignet und wie sich ein Lernfortschritt in dieser Richtung erkennen läßt.

3.1 Handlungsorientierung und Qualifikationserwerb

Im Laufe der Unterrichtsreihe erstellen die Schüler ein Handlungsprodukt, das ‚Roboter-Programm‘, am Rechner. Dieses ist aber zunächst nicht konkret fassbar. Es handelt sich lediglich um eine Beschreibung zur Lösung eines Problems in formaler, programmiersprachlicher Kodierung. Durch die Übertragung auf das Modell und die anschließende Testphase wird das Programm real, seine Auswirkungen sind materiell sicht- und spürbar. Ein Argument für die Wichtigkeit dieser Eigenschaft liefern JANK, MEYER (S.372): *Handlungsorientierter Unterricht ist notwendig, um dem allmählichen Verschwinden der Wirklichkeit aus dem Lernprozeß zu begegnen.“*

Die bei der schrittweisen Erstellung des Programmes von den Schülern gemachten Erfahrungen im Umgang mit spezifischen Arbeitsweisen der Softwareentwicklung sollen zudem ihre Methodenkompetenz verbessern. Denn „*die Methodenkompetenzen und –defizite der Schülerinnen können im Handlungsorientierten Unterricht bewusst gemacht und gezielt weiterentwickelt bzw. ausgeglichen werden.*“²⁴

²² Zit. KLIPPert, S.27.

²³ Zit. KLIPPert, S.27.

²⁴ Zit. JANK, MEYER, S.369.

Gerade die 'unscharfen' Algorithmen eignen sich dazu, erst einmal eine Lösung zu entwickeln, die noch Fehler enthalten kann, welche in einer anschließenden Testphase selbständig entdeckt und verbessert werden. Hierbei übernimmt nicht der Lehrer, sondern das Robotermodell die Korrekturfunktion. Für eine gegenteilige Vorgehensweise stellt STRUCK (S.155) fest: *„Die Erkundungsfreude und die Handlungsmotivation von Schülern werden nämlich erheblich gebremst, wenn sie beim Lernen keine Fehler machen dürfen, wenn sie ständig Angst vor roter Tinte, schlechten Noten, bösen Blicken, drohenden Gebärden und Strafen haben müssen.“* PAPERT äußert sich bezogen auf Informatik ebenfalls gleicher Weise: *„Besser als Noten finde ich Arbeitsproben - selber erschaffene Computerprogramme ... ein Nachweis von mathematischen und künstlerischen Fähigkeiten.“*²⁵ Ein engagiertes Experimentierverhalten von sonst eher zurückhaltend agierenden Schülern wäre also als Fortschritt beim selbständigen und entdeckenden Lernen zu bewerten.

Den Qualifikationserwerb, welcher bei der Durchführung eines Robotik-Projekts erlangt werden kann, beschreiben ALTENBURG, ALTENBURG (S.14): *„Doch ein Student ..., der während des Studiums, und sei es aus Lego-Teilen, einen Miniroboter baut und zum Funktionieren bringt, hat gegenüber denen, die den unter Umständen besseren Abschluss vorweisen können, unschätzbare Vorteile bei der Bewertung praxisrelevanter Aufgabenstellungen.“*

Ein speziell auf diesen Bereich der Informatik ausgerichteten Studiengang ‚Systemtechnik und technische Kybernetik‘ gibt es seit dem Jahr 2000 an der Universität Magdeburg²⁶ und auch in Berlin wird an der Freien Universität in ähnlicher Weise geforscht und ausgebildet.²⁷ Ein dem Könnenstand der Basiskursschüler angepasstes Unterrichtsvorhaben kann somit auch zukünftige Bildungsentscheidungen vorbereiten helfen.

²⁵ Zit. PAPERT, S.[2000]; Interview mit Michael West; Internet: www.brueckenbauer.ch/papert.html

²⁶ vgl.: <http://www.mpi-magdeburg.mpg.de>

²⁷ vgl.: <http://www.inf.fu-berlin.de/~robocup>

Schließlich sollte noch erwähnt werden, dass die Durchführung eines Kurzprojektes zum Ende des ersten Unterrichtsjahres bereits methodische Voraussetzungen schafft, welche für das große ‚Software-Projekt‘ zu Beginn des dritten Unterrichtsjahres Informatik benötigt werden²⁸.

3.2 Partnerarbeit

Für die Wahl der Partnerarbeit in der Informatik und speziell in dieser Reihe als günstige Organisationsform gibt es drei Argumente:

- Es findet in den Erarbeitungsphasen eine intensive Schüler-Schüler Interaktion statt. Die hochgelobte Schlüsselqualifikation ‚Teamfähigkeit‘ kann dabei weiterentwickelt werden.²⁹
- Einige Arbeitsaufgaben können von einem Schüler alleine gar nicht gelöst werden.³⁰ Es muss Arbeitsteilung praktiziert werden.³¹
- Bei mehr als zwei Schülern an einem PC lenkt sich erfahrungsgemäß mindestens einer anderweitig ab, da er nichts sieht, Verständnisprobleme hat oder nicht in die Arbeit eingebunden wird. Die Gefahr, dass dieser Schüler wiederum andere Schüler ablenkt, ist groß.

Fortschritte in der Selbständigkeit der Schüler können bei der Partnerarbeit dann beobachtet werden, wenn es ihnen gelingt, ohne Lehrerhilfen eigene Entscheidungen zu treffen, konzentriert am Unterrichtsgegenstand zu arbeiten und schließlich ein Ergebnis vorweisen zu können.

3.2.1 Abstimmung und Arbeitsverteilung

Die sozialen Anforderungen an die Schüler innerhalb von Arbeitsgruppen sind gegenseitige Abstimmung, also z.B. wessen

²⁸ vgl. SENATSVERWALTUNG FÜR SCHULE, BERUFSBILDUNG UND SPORT, S.33ff.

²⁹ Vgl. STRUCK S.161ff.

³⁰ Ein Schüler bedient z.B. die NQC-Entwicklungsumgebung am PC, der Partner setzt gleichzeitig den mobilen Roboter in Gang und beobachtet dessen Verhalten.

³¹ vgl. LEHMANN S.205

Lösungsansatz realisiert wird, und die Aufteilung der anfallenden Arbeiten. Idealerweise sollten die Partner darauf achten, dass z.B. jeder zu gleichen Teilen Eingaben auf der Tastatur des PCs machen darf.

Zum Teil gilt es, durch die Bildung von Zweier-Teams, Tendenzen entgegenzuwirken, die gerade die ‚Computerkids‘ im Wahlfach Informatik meist schon verinnerlicht haben. POSTMAN deutet auf diesen Zusammenhang hin, wenn er schreibt (S.64): *„Trotzdem hat der Computer ... die machtvolle Tendenz, die persönliche Autonomie zu verstärken und individuelle Problemlösungen zu begünstigen, und die meisten Beispiele und Illustrationen – Papert zum Trotz – zeigen Kinder, die alleine arbeiten.“*

Einen möglichen positiven Aspekt der Partnerarbeit beschreibt STRUCK (S.162): *„Kinder lernen dabei immer zweierlei zugleich, nämlich den richtigen Weg zur Lösung der Aufgabe bzw. wie das Ergebnis lautet einerseits und wie man am besten mit den Teamgefährten zurechtkommt andererseits.“*

Während der Reihe tauschen mehrere schriftliche Aufgaben auf, die alle Schüler gleichermaßen trotz Partnerarbeit erledigen müssen, um die Sicherung der Arbeitsergebnisse zu gewährleisten:

- Erstellung von Struktogrammen
- Protokollierung der Tests
- Ausdruck der fertigen Programme

Aber in der Implementationsphase bedürfen die Schüler hinsichtlich des einzuschlagenden Lösungsweg einer gegenseitigen Abstimmung, bei der unter Umständen ein Partner nachgeben muss. Die Abstimmung erfordert dadurch natürlich auch gute Argumente für die eigene Vorgehensweise.

Der Lehrer kann in dieser Unterrichtsphase zwar beobachten, wie sich die Teams vor dem Rechner verhalten, Rückschlüsse hinsichtlich der individuellen Anteile am Schülerprodukt sind aber kaum möglich. Wenn z.B. kein Rollenwechsel an der Tastatur des PC stattfindet, zeugt das nicht zwingend davon, dass Schüler gedanklich nicht

mitarbeiten. Es kann aber ein Indiz dafür sein, wer in der Gruppe die Führungsrolle übernommen hat.

In der Testphase muss das Zweier-Team sich mit den anderen Gruppen darüber abstimmen, wer als nächster den RCX verwenden darf. In evtl. Wartezeiten ist hier auch Gelegenheit, von den Fehlern oder guten Ideen der anderen Paare zu lernen.

Das Experiment an sich erfordert arbeitsteilig je einen Schüler am Steuer-PC und den anderen am Roboter. Durch Anweisungen wie z.B. „Jetzt kannst Du den RCX starten!“, wird ein Experiment koordiniert.

Selbständiges und arbeitsteiliges Verhalten wäre einerseits in einer intensiven Kommunikation zwischen den Schülern zu erkennen und andererseits in der Erledigung auch „unbeliebter“ Aufgaben, wie der Protokollführung, zu beobachten.

3.2.2 Ergebnispräsentation

Die Präsentation der Arbeitsergebnisse entspricht in dieser Reihe der Auswertungsphase nach JANK, MEYER³² und hat somit zugleich eine Dokumentationsfunktion.

Es ist zu erwarten, dass sich die Schülerprodukte, die Steuerungsprogramme für den mobilen Roboter, zwar ähneln, jedoch nicht gleichen. Eine Präsentation des eigenen Programms ist in doppelter Hinsicht vorgesehen. Zum einen führen die Paare ihre letzte Programmversion als Roboterexperiment vor. Zum anderen wird der Programmcode projiziert und mit eigenen Worten erläutert. Ein Schwerpunkt soll dabei auf der Erklärung gelöster Teilprobleme liegen. Eine Festlegung auf die beste oder einzig richtige Musterlösung wird der Lehrer an dieser Stelle bewusst vermeiden. Gelungene Ergebnisse sollte der Lehrer aber loben und auf noch vorhandene Unzulänglichkeiten hinweisen.

³² Vgl. Jank, Meyer, S.368

Bei der Präsentation üben und ergänzen sich die Partner im freien Vortrag vor der Gruppe und stellen sich möglichen Nachfragen. Dieses Vorgehen trägt somit schon einen propädeutischen Charakter. In diesem Anforderungsbereich selbständigen Handelns kann aus den von den Schülern gemachten Äußerungen ihre Identifikation mit dem gemeinsamen Arbeitsergebnis deutlich werden. Dabei spielt sowohl der eigene Redeanteil als auch die kompetente Verwendung von Fachbegriffen und Fachwissen eine wesentliche Bedeutung.

3.3 Schüler entwickeln Programme

Um die Entwicklung komplexerer Programme beherrschbar zu gestalten, gibt es in der Industrie verschiedene Phasenmodelle hierfür. Im weit verbreiteten sogenannten ‚Wasserfallmodell‘ folgt auf das Ergebnis einer Phase sukzessive die nächste Phase. ROLLKE, SENNHOLZ (S.314) schlagen vor, sich „... *im Informatikunterricht mit einer einfacheren Version...*“ zu begnügen:

- Systemanalyse;
- Systementwurf;
- Systemrealisierung (Implementation);
- Systemtest;
- Systeminstallation.

Für die Durchführung von Kurzprojekten schlägt LEHMANN (S.207) jedoch vor: „*Dabei und auch bei Zeitmangel kann es durchaus sinnvoll sein, die Themenwahlphase bei Vorliegen eines überzeugenden Themas zu überspringen und den Schülern das Thema mit Begründung vorzustellen.*“ Auf die Robotik-Reihe angewandt, bedeutet dies, dass die Bewegungsaufgaben des Roboters vorgegeben werden.

Ein beispielhaftes Robotik-Projekt skizzieren ROLLKE, SENNHOLZ (S.326 f.) und merken an, dass die Systemintegration, also der Einsatz des Roboters über das Unterrichtsprojekt hinaus, entfallen kann.

Diesen Überlegungen folgend, werde ich mich auch bei meiner Reihe auf die drei Phasen Entwurf, Realisierung und Test beschränken.

Der Schwerpunkt meiner Reihe liegt aber eindeutig auf wiederholter Implementation und nachfolgender Testphase. Dieses Vorgehen entspricht der ‚rapid prototyping‘ genannten Methode, bei der zunächst ein ungefährender Prototyp eines Programmes erstellt wird, welcher dann beim Testen optimiert werden kann.

3.3.1 Programmentwurf mit Struktogrammen

Einen Schlüssel zum Gelingen oder Misslingen der Unterrichtsreihe bildet bereits die Entwurfsphase wegen der dort angestellten grundlegenden Überlegungen der Schüler zur Lösung der Aufgabenstellung. Umso wichtiger ist an dieser Stelle eine Methode zur Strukturierung der zunächst ungeordnet vorhandenen Ideen, welche die Schüler im Unterrichtsgespräch äußern. Eine formale Methode bietet dabei das programmiersprachenunabhängige Konzept des ‚Struktogramms‘³³. Das Problem wird bei dieser Methode in kleinere Lösungsschritte eingeteilt. Diese werden dann grafisch durch entsprechende Strukturblöcke dargestellt. Ein Beispiel war bereits in Kapitel 2.3.3 zu sehen.

Es ist für die Schüler besonders günstig, dass die schriftliche Beschreibung der Lösungsschritte innerhalb eines Blocks sowohl umgangssprachlich als auch schon in programmiersprachenähnlichem ‚Pseudocode‘ erfolgen kann. Somit kann der Lehrer die Aufgabenstellung dem Könnensstand der Schüler anpassen.

Die schriftliche Erstellung eines Struktogrammes ist auch gut geeignet, um sie als vorbereitende Hausaufgabe für die praktische

³³ Dieses wird auch nach seinen Entwicklern Nassi-Shneidermann-Diagramm genannt (vgl. Informatik-Duden).

Unterrichtsphase am Rechner zu verwenden, nachdem bereits ein Lösungsansatz in der vorherigen Stunde erarbeitet wurde.

Für den Lehrer bietet ein Schüler-Struktogramm gleichfalls die Möglichkeit, eventuelle ‚Denkfehler‘ zu erkennen, welche sich von reinen syntaktischen Fehlern in der Implementierungsphase unterscheiden.

Durch die Forderung nach einem höheren Präzisionsgrad innerhalb des Struktogrammes, also dem Aufteilen in kleinschrittigere Blöcke, können vorher nicht beachtete Detailprobleme gelöst werden, ohne dass der Lehrer dem Schüler die Lösung vorgeben muss.

Aus den genannten Gründen sehe ich die Verwendung von Struktogrammen für die Robotik-Reihe in dieser Jahrgangsstufe als geeignete Methode an.

3.3.2 Systembeherrschung

Die oben beschriebenen Komponenten Robotermodell, Infrarotschnittstelle, Rechner und Entwicklungsumgebung stellen ein komplexes Experimentierfeld dar. Für ein reibungsloses Zusammenspiel der Komponenten bedarf es für die Schüler genauer Kenntnis der erforderlichen Arbeitsabläufe. Deswegen teilt sich dieser Bereich der Reihe in zwei Phasen auf. Während des ersten beispielhaften Einsatzes des Gesamtsystems durch den Lehrer werden die Komponenten einzeln und im Zusammenspiel demonstriert. Diesem lehrgangsartig organisierten Teil folgt eine Übungsphase, in der die Schüler die einzelnen Arbeitsschritte selbsttätig ausprobieren, indem sie ein vorgefertigtes Beispielprogramm für den RCX ablaufen lassen.

Bereits vor dieser Phase sind die Schüler in Gruppenpaare aufzuteilen. Da jede Arbeitsgruppe zwar einen PC zur Verfügung hat, sich aber alle Schüler ein Experimentalsystem teilen, müssen sie ab diesem Zeitpunkt zusätzlich darauf achten, miteinander zu kooperieren. Der Lehrer kann sich darauf beschränken, lediglich dann

eingzugreifen, falls es in einer Gruppe nicht zu einem Rollentausch³⁴ bei der Bearbeitung der Aufgaben kommen sollte oder Abstimmungsprobleme zwischen den Arbeitsgruppen auftauchen. Die in dieser Phase auftretenden Probleme sowohl im Umgang mit dem Gesamtsystem, als auch zwischen oder innerhalb der Gruppenpaare, können bereits zu diesem Zeitpunkt behoben werden, indem sie mit dem gesamten Kurs im Unterrichtsgespräch thematisiert werden.

3.3.3 Kodierung

Um für die Schüler den Einstieg in die für den RCX-Baustein ausgewählte Programmiersprache ‚NQC‘ zu erleichtern, wird zunächst eine Gegenüberstellung von NQC zu der den Schülern bekannten Programmiersprache ‚Python‘ vorgenommen. Aus zeitlichen Gründen verzichte ich auf eine ebenfalls mögliche Erarbeitung der syntaktischen Regeln von NQC³⁵ durch die Schüler selbst. Der Vergleich erfolgt in Form einer zweiseitigen Tabelle auf einem Arbeitsbogen, bei der sich die einzelnen syntaktischen Elemente für die Kontrollstrukturen in den Programmiersprachen Python und NQC in je einer Zeile gegenüberstehen.

Im zweiten Schritt soll ein einfaches Demonstrationsprogramm in NQC analysiert und dessen realer Ablauf von den Schülern mit eigenen Worten beschrieben werden. Dabei verschaffen sich die Schüler Kenntnis von den speziellen Befehlen in NQC, welche z.B. zur Ansteuerung der Motoren des Roboters vonnöten sind.

Für die selbsttätige Entwicklung eines NQC-Programmes gibt es je nach Könnensstand der Schüler zwei Wege. Sie können das Beispielprogramm modifizieren oder ein gänzlich neues NQC-Programm entwerfen. Bei der Wahl der ersten Möglichkeit entsteht dabei ein Schülerprodukt, das bei der Benennung von Prozedur- und

³⁴ Dieser Rollentausch ist unabdingbar, damit jeder Schüler alle Arbeitsschritte durch eigenes Handeln kennen lernt.

³⁵ Z.B. anhand der NQC Dokumentation (vgl. BAUM).

Variablennamen durchgängig einheitlich gestaltet wäre. Für eine Bewertung der Eigenleistung der Schüler bei der Kodierung ist die Verwendung einer Fremdvorlage jedoch problematisch.

3.3.4 Testen und Programmänderung

Wenn man den Begriff ‚Entwicklung‘ ernst nimmt, so beginnt mit der Methode des Testens nun eine sehr zeitintensive Phase. Je besser der erste Entwurf, der Prototyp, durchdacht war, desto dichter könnte man sich am Ziel befinden. Jedoch bietet die Überprüfung der eigenen Ideen durch die reale Durchführung des Robotik-Experimentes eine nicht zu unterschätzenden Lernmöglichkeit. POSTMAN (S.146) drückt Wirkung entdeckenden Lernens deutlich aus: *„Ich riskiere es, John Deweys berühmtesten Aphorismus zu widersprechen, wenn ich sage, dass wir zwar durchaus beim Tun lernen („learning by doing“), dass wir aber weit mehr lernen, indem wir etwas nicht tun; durch Versuch und Irrtum nämlich, indem wir Fehler machen, sie korrigieren und so weiter.“*

Um nun aber nicht hilflosem und ungerichtetem Herumprobieren der Schüler bei der Suche nach einer praktikablen Lösung Vorschub zu leisten, muss der Prozess des Testens durch eine zielgerichtete Methode flankiert werden. In Form vorbereiteter Testprotokolle, die lediglich eine leere zweispaltige Tabelle enthalten, werden die Arbeitsgruppen aufgefordert, ihre Testsbeobachtungen und Programmänderungen zu formulieren und gleichzeitig damit zu dokumentieren. Eine permanente Kontrolle aller Arbeitsgruppen auf ihrem Lösungsweg durch den Lehrer kann somit entfallen. Das Testprotokoll offenbart jederzeit die Optimierungsrichtung der Programme, welche die Zweier-Teams gewählt haben. Für die spätere Präsentation des Arbeitsergebnisses liefert diese Methode als Gedächtnisstütze ebenfalls eine wichtige Voraussetzung.

4. Unterrichtsvoraussetzungen

4.1 Allgemeine Unterrichtsvoraussetzungen

Es handelt sich bei dem Basiskurs um das erste Unterrichtsjahr Informatik für die Schüler. Ich habe den Unterricht seit Schuljahresbeginn selbständig erteilt. Die Stunden liegen am Dienstag in der 6.Stunde und am Mittwoch als Doppelstunde in den ersten beiden Stunden. Durch den langen Zeitraum vom Mittwoch bis zum darauf folgenden Dienstag muss in der Dienstagsstunde immer etwas mehr Raum für eine Wiederholung bzw. den Themeneinstieg berücksichtigt werden. Als Stunden mit Schülerarbeit am Rechner eignen sich die Doppelstunden am Mittwoch besser. Ich konnte hier auch eine bessere Konzentrationsfähigkeit der Schüler feststellen. Umfangreichere Hausaufgaben sind ebenfalls aufgrund des Stundenplans günstiger vom Mittwoch zur darauffolgenden Woche einzuplanen.

4.2 Spezielle Unterrichtsvoraussetzungen

4.2.1 Der Unterricht im ersten Jahr Informatik

Ein zu Beginn des Schuljahres ausgegebener Fragebogen ergab ein großes Vorwissen aller Schüler im Umgang mit dem PC. Zudem wählten fünf von elf Schülern das Thema Robotik als einen ihrer Interessenschwerpunkte im Fach Informatik aus einer Liste mit zehn Themengebieten aus.

So verknüpfte ich im ersten Halbjahr geschichtliche Aspekte der Informatik mit einfachen Programmen in der Sprache ‚Python‘.³⁶ Einen weiteren Schwerpunkt bildete eine Methodik zum strukturierten Problemlösen. Algorithmen wurden zuerst umgangssprachlich formuliert und später grafisch mittels Struktogramm visualisiert.

Zur Förderung der selbständigen Erarbeitung von informatischen Themen musste jeder Schüler ein ca. 10 minütiges Kurzreferat halten, wozu er vorab Informationsmaterialien erhielt. Die Kriterien für die

³⁶ Ein Programm konnte z.B. Texteingaben in das Morse-Alphabet umwandeln.

Gestaltung der Referate wurden im Unterrichtsgespräch entwickelt. Weiterhin wurden in Theoriestunden an einem Laptop-PC Stundenprotokolle von abwechselnden Schülern gefertigt.

In der Algorithmik beherrschen die Schüler die Kontrollstrukturen ‚Sequenz‘, ‚bedingte Anweisung‘ sowie die ‚vorprüfende Schleife‘. Sie wissen, was Variablen und Prozeduren sind und verwenden diese in ihren Programmen.

Ein komplexeres Python-Programm, welches Bibliotheksfunktionen benutzt und mit Wert- und Variablenparameterübergabe arbeitete, ging der hier behandelten Unterrichtreihe voraus. Über den Leistungsstand der Schüler gab die Halbjahresklausur am 6.6.01 Auskunft. Der Notendurchschnitt der Klausur war mit 3,6 deutlich schlechter als bei der vorigen Halbjahresklausur, die einen Schnitt von 2,8 hatte. Die Gründe für die Verschlechterung liegen z.T. daran, dass Schüler, die Informatik in der Oberstufe nicht weiterführen wollen, nicht mehr richtig mitarbeiten. Einen zweiten Faktor machen Defizite bei der Verwendung programmiersprachlicher Hilfsmittel, wie z.B. Struktogramme, aus.

4.2.2 Die Lerngruppe

Zu den 11 Schülern zu Schuljahresbeginn, unter denen nur eine Schülerin ist, kamen zum Halbjahr zwei Schüler neu hinzu. Einer hatte als Austauschschüler keinerlei Informatikkenntnisse und hat den Kurs nach drei Monaten bereits wieder verlassen. Der zweite Schüler MF. kam von einer anderen Schule und hat sich als leistungsstarker Schüler bereits gut in die Gruppe integriert. Seine Unterrichtbeiträge werden von seinen Mitschülern anerkannt. Er zeigt jedoch offen seinen Unmut, wenn Undiszipliniertheiten, insbesondere der leistungsschwächeren Schüler, ihn stören.

Die Unterrichtsatmosphäre im Informatikkurs empfinde ich trotzdem als überwiegend konzentriert und interessiert. Selbst die leistungsschwächeren Schüler O. und SF. beteiligen sich durch Wortbeiträge regelmäßig und geben sich bei Programmieraufgaben

Mühe. Lediglich der zurückhaltende K. muss häufig zur Mitarbeit aufgefordert werden. Er fehlte auch in den letzten Stunden häufiger, da er wahrscheinlich genauso wie O. und M. bereits über seine Nichtversetzung informiert war.

Das einzige Mädchen im Kurs fehlte zu Beginn des Halbjahres fast vier Wochen und schaffte es nicht, den Unterrichtsstoff in kurzer Zeit nachzuholen. Ein schlechtes Klausurergebnis versuchte sie seitdem durch verstärktes Engagement im Unterricht wettzumachen. Sie arbeitete in der Vergangenheit bei Partnerarbeitsphasen zunächst mit O. und M. zusammen; Spannungen führten aber dazu, dass sie teilweise alleine arbeiten wollte. Mit den anderen Jungen im Kurs scheint ein unverkrampfteres Arbeiten möglich zu sein.

Bei der Erarbeitung neuer Unterrichtsinhalte sind die Schüler SV., F., N., P. und MF. tragende Säulen im Unterrichtsgespräch.

Problematisch erwies sich die unstete Mitarbeit und das häufigere Fehlen von SF., was sich auch in nicht erbrachten oder mangelhaften Arbeitsleistungen niederschlug. Seine immer größer werdenden Wissenslücken konnte er immer weniger in der mündlichen Mitarbeit verbergen.

Die Schüler des Kurses sind mit der Arbeitsform ‚Partnerarbeit‘ gut vertraut, wobei die anfangs gebildeten Zweiergruppen überwiegend bestehen blieben. Da sowohl leistungsstarke als auch leistungsschwache Schüler untereinander Teams bildeten, wurden einige Paare neu zusammengesetzt, was zu spürbaren Widerständen führte. Für die gegenseitige Akzeptanz der Schüler untereinander war diese Steuerungsmaßnahme jedoch langfristig positiv zu beurteilen.

Des Weiteren wurde der freie Kurzvortrag geübt, sei es allein oder in der Gruppe, bei Schülerreferaten und Ergebnispräsentationen zum Ende einer Unterrichtsreihe. Es ließen sich schon deutliche Verbesserungen in der freien Rede vor der Gruppe bei den Schülern beobachten. Als mediale Hilfsmittel verwendeten die Schüler sowohl den Overhead-Projektor als auch den Lehrerrechner, auf dem erstellte Programme mittels eines ‚Beamers‘ projiziert werden können.

In der Benutzung menügesteuerter Programme im Windows Betriebssystem, mit dem die Schulrechner ausgestattet sind, zeigen alle Schüler bereits einen routinierten Umgang. Hilfestellungen, wenn überhaupt notwendig, geben sie sich größtenteils gegenseitig.

Die selbständige Erarbeitung neuer Sachverhalte anhand schriftlicher Unterlagen erbrachte teilweise nicht die gewünschten Ergebnisse, so dass ich lehrgangsartige Unterrichtsphasen in die Planung einbauen musste. Schriftliche Hausaufgaben wurden ebenfalls von den leistungsschwächeren Schülern (K., O. und SF.) nicht zufriedenstellend für die Vor- bzw. Nachbereitung des Unterrichts genutzt.

4.2.3 Materielle Ausstattung

Die Planung der Unterrichtsreihe stützt sich auf die stete Verfügbarkeit eines Unterrichtsraumes, welcher sowohl für normalen Unterricht als auch für Rechnerarbeit geeignet ist. Die Anzahl von 10 Schülerrechnern und einem Lehrerrechner würde in diesem Kurs sogar die Einzelarbeit am Computer ermöglichen, was aber aus pädagogischen Gründen, wie z.B. gegenseitiges Helfen oder Abstimmen über Lösungsentscheidungen gar nicht erwünscht ist.

Eine unbedingte Voraussetzung für ein auf selbständiges experimentelles Arbeiten angelegtes Unterrichtsprojekt war die Anschaffung wenigstens eines Roboterbaukastens und die vorherige ausgiebige Erprobung seiner Möglichkeiten und eventuellen Schwachstellen. Für die Durchführung der Schülerexperimente bot sich die Bereitstellung des mobilen Robotermodells auf einem zentralen und für alle Schüler einsehbaren Tisch an. Denn zu dieser Experimentierfläche hatten alle Schüler freien Zugang und sie musste sich auch für die Präsentationen der fertigen Programme eignen. Zu diesem Tisch musste die PC-Infrarotschnittstelle per Kabel heranreichen.

Um die von den Schülern erstellten NQC-Programme von den Schülerrechnern auf den Lehrerrechner zu übertragen, ist

idealerweise ein Rechnernetz zu verwenden. Beim Ausfall oder dem Fehlen eines solchen Netzes könnten die Programme aber auch auf einer Diskette zwischen den Rechnern transportiert werden.

Da der RCX-Baustein über fünf Programmspeicherplätze verfügt, ist die gleichzeitige Arbeit mit einem Modell auf fünf Arbeitsgruppen begrenzt.

5. Planung der Unterrichtsreihe

Da eine Umfrage vor der Unterrichtsreihe ergab, dass die Schüler mit Ausnahme von Science-Fiction-Filmen, kaum eine Berührung mit dem Thema Robotik gehabt haben, erscheint ein Einstieg mit einem Dokumentarfilm über Industrieroboter naheliegend. Um darin auftauchende Fachbegriffe festhalten und später klassifizieren zu können, wird ein entsprechend gestalteter Arbeitsbogen ausgegeben. Dieser gesellschaftsbezogene Aspekt einerseits und Begriffsklärungen andererseits legen die Grundlage für spätere Unterrichtsgespräche.

In einer lehrgangsartig organisierten zweiten Phase der Reihe soll das fächerübergreifend relevante Regelkreisprinzip behandelt und auf die Robotik angewandt werden. Dazu sind in Gruppenarbeit zwei Texte aus nicht-informatischen Themenbereichen³⁷ zu analysieren und ein Regelkreisschaubild um die jeweiligen Begriffe zu ergänzen. Anschließend sollen die Begriffe aus der Robotik im gleichen Schaubild im gelenkten Unterrichtsgespräch auf einer OH-Folie eingetragen werden. Die allgemeinen Prinzipien

- der Rückkopplung,
- der sich regelmäßig wiederholenden Erfassung von Messwerten
- und des Einflusses von Störgrößen,

³⁷ Die beiden Texte finden sich unter:

- „Die Wasseruhr des Ktesibios „
www.techkyb.uni-stuttgart.de/wasistkybernetik/geschichte/wasseruhr.php
- „Temperaturregulation bei körperlicher Aktivität „
In: deMarees,H., Mester,J.: Sportphysiologie II; Diesterweg-Verlag; Frankfurt(M); 1990; S.63.

welche das Regelkreismodell kennzeichnen, sollen durch den Vergleich der drei entstandenen Schaubilder herausgearbeitet werden.

Die nun folgende Arbeit mit dem realen Robotermodell verlässt für kurze Zeit die Regelkreisthematik, um die Eigenschaften der NQC-Entwicklungsumgebung und des Roboters kennen- und beherrschen zu lernen. Die Sensoren werden dabei zunächst nicht behandelt. Die Schüler erhalten eine Aufgabe, bei der es hauptsächlich um die Verwendung von Steuerungsbefehlen und die Anwendung der richtigen NQC-Syntax geht: Der mobile Roboter soll ein vorgegebenes Rechteck auf einer Tischoberfläche umfahren.

Durch die Einführung des Berührungssensors kommt man anschließend zur Regelkreisthematik zurück. Als Hausaufgabe ist ein Struktogramm für eine kybernetische Lösung des Hindernis-Problems zu entwickeln. Im Unterricht folgt die experimentelle Umsetzung in Partnerarbeit und eine anschließende Präsentation, nach der sich alle Schüler wieder auf dem gleichen Kenntnissniveau befinden sollen.

Die letzte Phase der Reihe sieht eine Wiederholung der Arbeitsabläufe, wie sie für das Hindernis-Problem beschrieben wurden, vor. Die Verwendung der Arbeitsmethoden durch die Schüler soll der Übung und Vertiefung des Gelernten dienen.

Mittels des im Kapitel ‚Algorithmen‘ beschriebenen höheren Schwierigkeitsgrades beim Problem ‚Linienfolger‘ ist es bei dieser Aufgabe möglich, als Differenzierungsmaßnahme einzelnen Paaren bereits Teile der Lösung vorzugeben.

Da die Reihe vor den Sommerferien endet, ist unter anderem aus Motivationsgründen für die letzte Unterrichtseinheit die Lesung der Kurzgeschichte ‚Vernunft‘ von I.ASIMOV geplant, welche besonders den Aspekt der künstlichen Intelligenz beinhaltet. Sie bietet vor dem Hintergrund der vorher im Umgang mit dem Robotermodell selbst gemachten Erfahrungen eine Diskussionsgrundlage für eine eher philosophische Behandlung des Themas künstliche Intelligenz / Robotik.

Unabhängig vom letzten Teil der Reihe sollen die Schüler zu Hause einen Kurzaufsatz verfassen, in dem sie kritisch zur Unterrichtsreihe und zum Thema Robotik Stellung nehmen sollen.

5.1 Reduktion und Schwerpunktbildung

Um das äußerst komplexe Themenfeld Robotik für den Informatikunterricht beherrschbar zu machen, gibt es meiner Ansicht nach drei Kernaspekte, welche schulgemäß reduziert werden können:

- Die Funktionsweise bzw. die Konstruktion eines Robotermodells;
- die Lösung kybernetischer Fragestellungen mit geeigneter Software;
- die gesellschaftliche Relevanz und der KI-Aspekt.

Möchte man sich eher mit der elektromechanischen Funktionsweise von Robotern beschäftigen, so sollte man Antriebstechnik und Getriebemechanismen behandeln. Diese Vorgehensweise mündet in der Konstruktion eigener Robotermodelle und die dabei möglichen Lerneffekte sind nicht zu verleugnen³⁸. Die dafür notwendige materielle und zeitliche Ausstattung legt aber nahe, solche Vorhaben in eine Arbeitsgemeinschaft zu verlegen, die eine kleinere Gruppenstärke hat und über längere Zeitenzyklen als den 45-Minutentakt einer Unterrichtsstunde verfügt. Es resultiert somit die Vorgabe eines bereits fertigen Modells, dessen elektromechanische Funktionsweise zwar erklärt werden muss, aber keinen Erkenntnisschwerpunkt darstellt.

Auf der Seite der Programmierung wäre es ideal, wenn die sonst im Unterricht verwendete Programmiersprache beibehalten werden könnte und nur um die Bibliotheksfunktionen zur Robotersteuerung zu ergänzen wäre. Die in den Login-Artikeln und den Schulbüchern³⁹ verwendeten Ansätze sind allesamt noch sehr hardwarenah und teilweise kryptisch. Durch die hohen Verkaufszahlen des LEGO-Mindstorms-Systems und die Möglichkeit, Ideen, Programme oder

³⁸ vgl. LEGO[2001] S.16f. und LEGO[2002] S.10f.

³⁹ vgl. besonders SCHWARZE, H., HOLZGREFE, H.-W (S.28 ff.) und THODE, R. (S.91ff.).

Baupläne im Internet auszutauschen, löst sich das Softwareproblem ‚fast von allein‘. Es bleibt die Auswahl einer praktikablen Programmiersprache zu treffen⁴⁰. In dieser Reihe kann durch die Auswahl der Entwicklungsumgebung für NQC den Schülern eine entsprechend Könnensstand und Alter passende Software verwendet werden. In den beiden ausgewählten kybernetischen Fragestellungen⁴¹ benötigen die Schüler keine Kontrollstrukturen, die sie nicht schon in der ihnen bekannten Programmiersprache Python verwendet haben.

Die Aspekte gesellschaftliche Relevanz der Robotik und künstliche Intelligenz bieten jeder für sich genug Stoff für eine eigene Unterrichtsreihe. Sie sollen in dieser Planung lediglich einen Rahmen bilden, indem sie Neugierde wecken und gleichzeitig Bezüge vom eigenen Roboterexperiment im Kleinen zu Themen der Robotik außerhalb der Schule herstellen helfen.

5.2 Ziele des Unterrichts

Entsprechend den in den vorigen Kapiteln gemachten Unterscheidungen möchte ich auch die mit der Unterrichtsreihe angepeilten Lernziele in den Bereichen ...

- Kompetenzerwerb im Themenfeld Robotik (**I**=Informatik),
- Methodenkompetenzen (**M**=Methoden) und
- soziale Kompetenzen (**S**=sozial)

ansiedeln. Innerhalb der von mir schwerpunktmäßig beobachteten Phasen Softwareentwurf, Implementierung und Test möchte ich folgende Teilziele erreichen:

In der Entwurfsphase:

- Die Schüler erwerben Kenntnisse zu den technischen Eigenschaften des verwendeten Robotermodells. (I)

⁴⁰ siehe KNUDSEN, NOGA (S.254)

⁴¹ siehe Kapitel 2.3 Algorithmen

- Sie können das allgemein gültige Regelkreisprinzip auf eine Robotersteuerung anwenden. (I)
- Sie stellen den Regelkreisalgorithmus in einer programmiersprachenunabhängigen Form, z.B. als Struktogramm dar. (M)
- Sie unterstützen eigene Entwurfsentscheidungen gegenüber Mitschülern im Unterrichtsgespräch durch geeignete Argumente. (S)

In der Implementationsphase:

- Die Schüler verwenden die Programmierbefehle und die Kontrollstrukturen für die Robotersteuerung in syntaktisch richtiger Weise. (I)
- Sie erstellen strukturierte und lesbare Programme, die ein Abbild ihrer schriftlichen Vorüberlegungen sind. (I+M)
- Sie verwenden selbständig das Hilfesystem der Entwicklungsumgebung bzw. eigene Unterlagen bei der Klärung von Problemen. (M)
- Sie integrieren Vorschläge ihres Arbeitspartners in das gemeinsam erstellte Programm. (M+S)
- Sie tauschen in der Partnerarbeit ihre Rollen. (S)

In der Testphase:

- Die Schüler erkennen die Notwendigkeit, Testergebnisse genau zu protokollieren. (I+M)
- Sie verbessern gezielt ihre Programme, indem sie auf ihre Testaufzeichnungen, Struktogramme oder Hilfesysteme zurückgreifen. (M)
- Sie bestimmen gemeinsam den Moment, in dem sie mit ihrem Programm zufrieden sind und es in Hinblick auf die Aufgabenstellung als fertig ansehen. (M+S)

Das Erreichen der Lernziele soll in zweierlei Hinsicht überprüft werden. Die auf die Informatik bezogenen Ziele können anhand der Güte der Roboterprogramme, welche die Schüler erstellt haben,

erfolgen. Eine positive Entwicklung der Schüler beim selbständigen Handeln und im sozialen Verhalten könnte lediglich im zukünftigen unterrichtlichen Geschehen festgestellt werden.

5.3 Synopse

Stunde	Thema	Material	U-Form	Groblernziel	Didaktischer Kurzkommentar
1 13.6.01 9:00-9:45	Robotik in der Arbeitswelt	Film I ⁴² Arbeitsbogen 1	-----	Die Sch. klassifizieren Begriffsfelder der Robotik und tragen diese in AB ein (HA).	Die Filmanalyse stellt einen praxisbezogenen Einstieg in die Reihe dar.
2 26.6.01 13:00-13:45	Der Regelkreis	Film II ⁴³	gUG Tafel	Die Sch. reflektieren die Merkmale des Regelkreises.	Der Lehrfilm zeigt sehr anschaulich, wie ein Rechner eine Dampfmaschine steuert.
3+4 27.6.01 8:00-9:45	Rückkopplung als Algorithmus	Texte zum Regelkreis, Arbeitsbogen 2 OH-Folien	Stillarbeit GA	Die Sch. erkennen die übergeordneten Prinzipien des Regelkreises.	Die Sch. entwickeln ein Struktogramm für den Regelkreis als nachbereitende HA.
5 3.7.01 13:00-13:45	Umgang mit NQC und der Roboter-Experimentierumgebung ⁴⁴	Roboter-Modell, PC, Arbeitsbogen 3	LV Stillarbeit PA	Die Sch. erproben die Durchführung eines beispielhaften Robotikexperiments.	Es werden lehrgangsartig die Voraussetzungen für die experimentelle Arbeit geschaffen.
6+7 4.7.01 8:00-9:45	Einführung von Sensoren Entwicklung und Präsentation des Programmes „Hindernis“	Roboter-Modell, PC Beamer	gUg LV PA SV	Die Sch. erarbeiten die Einbindung von Sensorabfragen in ihr Roboterprogramm.	Eine Erweiterung des Beispielprogramms aus der vorigen Stunde führt zu einer einfachen Lösung.
8 10.7.01 13:00-13:45	Lösungsstrategien für das Programm „Linienfolger“	Tafel	gUg PA	Die Sch. entwickeln einen Algorithmus für das ‚Linienfolger‘-Problem.	Der Algorithmus wird zunächst umgangssprachlich formuliert, dann als Struktogramm formalisiert.
9+10 11.7.01 8:00-9:45	Entwicklung und Test des Programmes „Linienfolger“	Roboter-Modell, PC Arbeitsbogen 4/5	PA	Die Sch. leiten aus den protokollierten Beobachtungen Entscheidungen zur Programmänderung ab.	Von den Schülern wird eine partnerschaftliche Abstimmung sowie disziplinierte Protokollierung verlangt. HA: Vorbereitung der Präsentation
11 17.7.01 13:00-13:45	Ergebnispräsentation und Besprechung	Roboter-Modell, PC, Beamer	SV gUg	Die Sch. erläutern ihr Roboterprogramm und begründen ihre Vorgehensweise.	Die Sch. müssen sich vor dem Kurs zu ihrem Werk erklären bzw. anderen Sch. zuhören.
12+13 18.7.01 8:00-9:45	Text: „Vernunft“ von I. Asimov mit anschl. Diskussion	---	fUg	Die Sch. vergleichen fiktive Roboterdarstellungen mit den eigenen Erfahrungen.	Die Einordnung des eigenen Wissens über Robotik ermöglicht eine fundierte Bewertung fiktiver Thesen.

⁴² Film I: „Registrieren/Reagieren – Regeln und Steuern“; BRD 1991; 30 min.

⁴³ Film II: „Wie der Computer Dampf macht“; BRD 1988; 30 min.

⁴⁴ In dieser Stunde waren Hr. Witten (Fachseminarleiter) und Herr Herbst (Schulleiter) anwesend.

6. Darstellung und Analyse des Unterrichts

6.1 Begründung zur Auswahl der Stunden

Die im Darstellungsschwerpunkt formulierte „selbständige Entwicklung ‚unscharfer‘ Algorithmen zur räumlichen Orientierung“ deutet schon an, dass nur diejenigen Stunden der Unterrichtsreihe einer intensiven Analyse unterzogen werden müssen, in denen die Schüler tatsächlich experimentell gearbeitet haben. Die Kenntnisse von Begriffen und wesentlichen Prinzipien der Robotik sowie die Fertigkeiten in der Bedienung des Experimentalsystems stellen die notwendigen Vorbedingungen sicher, ohne die kein Experiment hätte gelingen können.

Die in der Synopse grau unterlegten Zeilen stellen somit die zu untersuchenden Stunden dar. Da in diesen insgesamt sechs Unterrichtsstunden zwei im Schwierigkeitsgrad aufeinander aufbauende Programme entwickelt werden sollten, stellt die Doppelstunde vom 4.7.01 eine Art Generalprobe für das in den Stunden vom 10.7. bis zum 17.7. zu erstellende Programm ‚Linienfolger‘ dar. Sie ist in ihrem Phasenaufbau genauso konzipiert wie die folgenden vier Unterrichtsstunden im Ganzen. Bei am 4.7. auftauchenden Problemen sollten Planungsänderungen für die folgenden Stunden möglich bleiben.

6.2 Analyse

Für die Analyse des Unterrichts möchte ich der obigen Überlegung folgen und hauptsächlich die Stunde vom 4.7. jeweils in den einzelnen Phasen der Programmentwicklung vom Entwurf bis zur Präsentation darstellen. Hierbei sollen die fachlichen, methodischen und sozialen Aspekte, der Gliederung meiner Lernziele folgend, erörtert werden. Konsequenzen für die zweite Aufgabenstellung werden im Anschluss der jeweiligen Phasen diskutiert und mögliche Alternativen aufgezeigt.

6.2.1 Der Programmentwurf

Als Einstiegsimpuls für die Stunde am 4.7. erschien mir eine Aussage von N. aus der vorigen Stunde⁴⁵ geeignet. Bei dem Versuch den Roboter wieder in seine Ausgangsposition zurückzusteuern, sagte er: „*Unter realen Bedingungen kann man keine Perfektion erreichen.*“ Dieses Zitat stand zu Stundenbeginn an der Tafel. Das folgende Unterrichtsgespräch wurde von mir in die Richtung gelenkt, welche Änderungen am Roboter vorzunehmen wären, um dennoch eine perfekte Steuerung erreichen zu können. Es wurden mehrere Vorschläge gemacht:

- „Eine komplett ebene Tischplatte verwenden.“ (SF.)
- „Die Ausgangsposition in Variablen merken.“ (M.)
- „Man muss Orientierungspunkte anbringen.“ (SV.)
- „Der Roboter müsste die Umgebung erkunden können.“ (V.)

Der erste Vorschlag, die Umgebung dem Roboter anzupassen, wurde verworfen und die drei übrigen Ideen durch die Vorstellung des Berührungssensors in einem 5-minütigen Lehrervortrag ergänzt. Dabei wurde die Verwendung der Sensorbefehle in NQC demonstriert.

Wie das Problem ‚Hindernis umfahren‘ nun anders gelöst werden könnte, fiel N. ein: „*Man muss immerzu gucken, welchen Wert der Sensor hat, und den Roboter dann drehen.*“ Der Transfer zu dem Regelkreisstruktogramm, den ich erhofft hatte, gelangt den Schülern aber nicht. Der Arbeitsauftrag lautete deswegen, dieses Struktogramm in Partnerarbeit an das aktuelle Problem anzupassen. Die Bekanntheit der Struktogramm-Methode führte zu schnellen, aber dennoch richtigen Lösungen. Der Grad der Selbständigkeit ist in dieser Entwurfsmethode bereits soweit fortgeschritten, dass es Hilfestellungen seitens der Lehrkraft selten bedurfte.

Durch diese Vorarbeiten gestaltete sich die Implementationsphase, also die Umsetzung des Struktogrammes in ein NQC-Programm sehr unproblematisch. Nachfragen hinsichtlich Unklarheiten in der NQC-

⁴⁵ In dieser Stunde wurde eine geometrische Lösung des Problems ‚Hindernis umfahren‘ programmiert.

Syntax, die jetzt vermehrt auftauchten, beantwortete ich durchgängig mit dem Verweis auf Arbeitsbogen 3 bzw. die Hilfefunktion der NQC-Entwicklungsumgebung. Auf der Suche nach Antworten auf ihre Fragen arbeiteten die Schüler kooperativ und fanden in fast allen Fällen eine Lösung.

Die hinzugewonnene Selbständigkeit der Schüler bei der Nutzung unterschiedlicher Informationsquellen zahlte sich bei der Entwicklung des zweiten Programms ‚Linienfolger‘ aus, wodurch die Lehrerhilfe weiter reduziert werden konnte⁴⁶.

6.2.2 Testen und Verbessern

Die Notwendigkeit zur disziplinierten Protokollierung wurde für das einfachere Programm ‚Hindernis umfahren‘ von den Schülern nicht eingesehen. Die Protokolle ähnelten Schmierzetteln und wurden als lästige Pflichtaufgabe angesehen. Für das zweite Programm konzipierte ich deshalb einen Arbeitsbogen für das Protokoll und forderte die Schüler auf, in verständlichen Sätzen ihre Beobachtungen zu notieren.

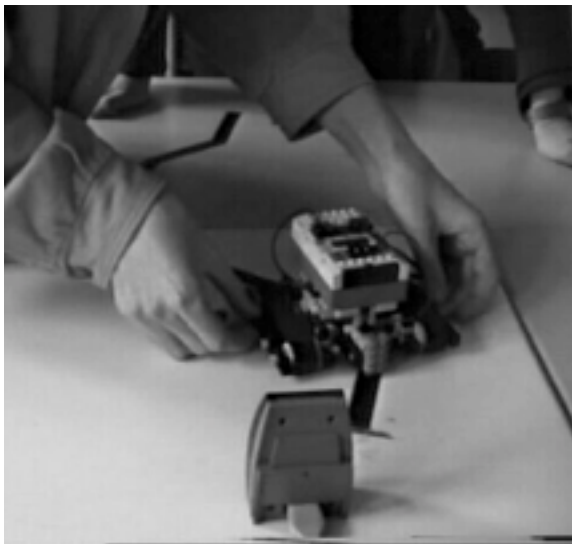


Abbildung V: Schüler beim Testen des ‚Linienfolger‘ Programms. Im Vordergrund ist die Infrarotschnittstelle zu sehen.

Die wachsende Komplexität des ‚Linienfolger‘-Problems und ausbleibende schnelle Erfolge verbesserten die Einsicht der Schüler in

⁴⁶ vgl. dazu: KLIPPERT, S.27

die Protokollführung. Trotzdem waren noch große Defizite in der Formulierung der Beobachtungen vorhanden.⁴⁷ Ich würde deswegen für diese Methode alternativ noch mehr Zeit einplanen und die Protokolle auch in die Leistungsbewertung einbeziehen. Die Schüler haben es scheinbar verinnerlicht, nur Sorgfalt und Mühe in bewertete Arbeiten zu investieren.

6.2.3 Präsentation

Im freien Vortrag offenbarten sich bei allen Schülern mehr oder weniger starke Defizite in der Formulierung und in der Begründung von Entwurfsentscheidungen zu ihren Programme. Vor einer erneuten Präsentationsaufgabe müsste auf jeden Fall eine Unterrichtsstunde für die schriftliche Vorbereitung des Vortrages reserviert werden.

Die Wichtigkeit einer gelungenen Präsentation wurde den Schülern auch nicht deutlich genug. Die Vorstellung, dass das Programm ja fertig sei, lies die Präsentation eine Pflichtübung werden. Denn bereits durch die ausgiebigen Testphasen kannten die Schüler die Programme der anderen Teams. Als günstiger hätte sich vielleicht eine Präsentation z.B. vor den Schülern des anderen Basiskurses Informatik ausgewirkt.

7. Gesamtreflexion

7.1 Einsichten

Ich bin mit der Hoffnung in dieses Unterrichtsvorhaben gestartet, den Schülern meines Kurses einen tieferen Einblick in die Probleme zu geben, welche sich eröffnen, wenn theoretisch entworfene Programme mit der Realität konfrontiert werden. Ob dies mit dem ausgewählten LEGO-Mindstorms-System in Kombination mit der NQC-Programmiersprache gelungen ist, ist der erste Frage, die ich durch Auszüge aus den Kurzaufsätzen⁴⁸ der Schüler mit ‚ja‘ beantworten kann. Äußerungen, wie: „*Dadurch wurde mir das erste Mal wirklich*

⁴⁷ Siehe Anhang C) Testprotokoll für das ‚Linienfolger‘-Programm von T. und V.

⁴⁸ Die Form eines Aufsatzes erschien mir geeigneter, um authentische Aussagen zu erhalten, als eine Umfrage per Fragebogen.

klar, wie kompliziert das wirklich ist.“ (SV.), „Da der Tisch oder auch andere Einflussfaktoren vorlagen, konnte man nie sicher sein, dass auch alles so funktionierte, wie man es programmiert hatte.“ (F.) oder: „Aber genau das ist es, was Informatik ausmacht. Nämlich selbst auf Lösungen kommen, eigene Dinge kreieren und Vorhandenes zu optimieren.“ (T.) belegen dies.

7.2 Motivation

Wie war es nun um die Motivation der Schüler nach der Semesterklausur und vor den Sommerferien bestellt? Sie und die Arbeitsatmosphäre im Unterricht übertrafen meine Erwartungen erheblich. Von der ersten bis zur letzten Stunde der Robotikreihe konnte ich keine Ablenkung mit anderen Dingen, wie z.B. Versuche unerlaubt im Internet zu surfen, registrieren. Dem füge ich einige Zitate aus den Kurzaufsätzen unterstützend hinzu:

„Noch einmal? Auf jeden Fall, denn ich habe gemerkt, wie viel Spaß das macht (man macht etwas und kann gleich die Resultate sehen).“ (V.)

„Im Großen und Ganzen hat mir die Unterrichtsreihe Robotik/Roboter sehr viel Spaß gemacht und mich auch angeregt, mehr über Roboter zu erfahren.“ (F.)

Auch die Fehlzeiten der Schüler belegen ihr starkes Interesse an der Thematik. Lediglich K., der bereits über seine Nichtversetzung im Bilde war, fehlte an vier Terminen und P. war an drei Terminen krank.

7.3 Partnerarbeit

Das leidige Thema Fehlzeiten leitet direkt über zur Hauptorganisationsform ‚Partnerarbeit‘, bei der aufgrund der Fehlzeiten von fünf Paaren auf drei Paare und eine Dreiergruppe umgestellt werden musste. Die von der Umstellung betroffenen Paare kamen in Schwierigkeiten, ihre Programme in der zur Verfügung stehenden Zeit fertigzustellen. *„Was ich mir gewünscht hätte, wäre etwas mehr Zeit gewesen,...“ (SV.).* Dem gegenüber stehen Aussagen

der Schüler, welche die Partnerarbeit als Arbeitsform geeignet erscheinen lassen: *“...teilten wir uns in Arbeitsgruppen auf, was eine richtig gute Arbeitsatmosphäre schaffte und einen zur Teamarbeit mehr oder weniger zwang, was vor allem meiner Meinung nach in jedem modernen Job verlangt wird.”* (F.). Und im Unterricht gab z.B. V. ihre Freude gegenüber dem Arbeitspartner Ausdruck: *„Juhu, wenigstens haben wir es schon mal bis hierhin geschafft.“*

Während der gemeinsamen Arbeit am PC erklärte der ansonsten oft genervt scheinende MF. seinen Partnern O. und M. geduldig die von ihm geplanten nächsten Arbeitsschritte. In diesem Team, sowie beim Paar SV. und SF. trafen die leistungstärkeren Schüler die wichtige Entwurfsentscheidungen alleine. In der Experimentierphase war aber durchgängig eine Beteiligung aller Schüler zu beobachten.

Die Testphase offenbarte eine Schwäche der Planung, denn es *„...wurde um den RCX doch etwas eng, so dass man nicht immer in Ruhe alles ausprobieren konnte.“* (SV.) Die erste Veränderung in der Unterrichtsgestaltung war die Aufgabe, dass die Schüler eine syntaktisch fehlerfreie Version ihres Programms auf ihrem Rechner compilieren (d.h. übersetzen) lassen sollten, bevor sie mit dem eigentlichen Experiment begannen. Somit konnten die Wartezeiten etwas verkürzt werden. Ein weiterer Baukasten oder die Bildung größerer Gruppen wären weitere mögliche Alternativen. Wobei in größeren Gruppen ein verstärkt arbeitsteiliges Vorgehen unterstützt werden müsste, um alle Schüler gleichzeitig zu beschäftigen.

7.4 Methodentraining

Das erhoffte Ziel, durch konsequente Protokollführung schneller und effizienter die Programme zu verbessern und Fehler zu finden, konnte leider nur ansatzweise erreicht werden. Die für das Problem ‚Hindernis umfahren‘ geforderten Protokolle hatten keinerlei Struktur und waren, wenn überhaupt geführt, bessere Schmierzettel. Trotzdem wurde ich weiterhin als Lehrer aufgefordert, die von den Schülern gemachten Fehler zu identifizieren. Die Einsicht in die Notwendigkeit der

Versuchprotokolle als neue Unterrichtsmethode fehlte. *„Was leider ein bisschen genervt hat, war, dass man seine Änderungen am Programm immer dokumentieren musste, was dann aber im Endeffekt doch nicht so gestört hat.“* (T.). Ironisch merkt zum gleichen Problem T. an: *„Oh, wir haben die Klammer in der 23. Zeile vergessen. Klar, dass das nicht funktionieren konnte.“*

Aus diesen Erfahrungen heraus konzipierte ich den Arbeitsbogen 4 neu als Arbeitsbogen 5, indem eine Tabelle äußere Form und erwarteten Inhalt der zu protokollierenden Tests für das ‚Linienfolger‘-Problem vorgab. Trotzdem⁴⁹ hatten die Protokolle noch nicht die gewünschte Gestalt und ihr Nutzen war kleiner als erwartet. In einer ähnlichen praxisorientierten Reihe würde ich die Schüler erst wieder an die Computer lassen, wenn die Protokolleinträge sorgfältig erfolgt sind. Außerdem wage ich die Prognose, dass es bei einer wiederholten Anwendung der Methode durch Einsicht in den Nutzen dieses Verfahrens zu besseren Ergebnissen kommt.

Diesen Effekt konnte ich bei der im ersten Unterrichtsjahr schon häufiger angewandten Methode der Problemzerlegung mittels Struktogrammen beobachten. Sie wird von den Schülern als Hilfsmittel anerkannt, ohne das größere Probleme kaum zu bewältigen sind. Dadurch konnte die Entwurfsphase reibungslos in der geplanten Zeit durchlaufen werden.

Schließlich noch einige Bemerkungen zur Präsentation der Programme. Da ein Konkurrenzdruck unter den Schülern ohnehin latent vorhanden zu sein scheint, habe ich auf einen wettbewerbsähnlichen Charakter der Präsentation bewusst verzichtet. So zollten bei der Vorführung der Programme auch Schüler der anderen Teams Beifall für ein funktionierendes Programm. Da ich alle Programme vor dieser Vorführung eingesammelt und getestet hatte, konnte ich auch zwei Gruppen, deren Programme noch kleine Fehler

⁴⁹ Vgl. Anhang C)

enthielten, durch Korrekturhinweise einen erfolgreichen Abschluss ihrer Arbeit ermöglichen.

Die Probleme in der Präsentation lagen, wie bereits erwähnt, in der fehlenden gemeinsamen Vorbereitung der Paare auf diesen Schülervortrag. Eine schriftliche Konzeption des Vortrags könnte zudem vorher mit dem Lehrer abgesprochen werden.

7.5 Schülerprodukte

Als ein Ergebnis dieses Unterrichtsvorhabens hatten die Arbeitspaare schließlich jeweils zwei fertige NQC-Programme zur Robotersteuerung erstellt.⁵⁰ Bei allen Schülerprodukten war deutlich die strukturierte Vorgehensweise bei der Programmgestaltung erkennbar, wenn auch teilweise die Wahl von Variablen- oder Prozedurnamen noch nicht dem Problem angemessen war.

F. schreibt in seinem Aufsatz, dass sich *„...die Schwierigkeit der Programme sich immer etwas gesteigert hat und (sie) so gut aufeinander aufbauten.“* Während das Programm ‚Hindernis umfahren‘ von allen Paaren mittels einer einzigen strukturiert programmierten Prozedur zufriedenstellend gelöst wurde, führten die Schwierigkeiten, mehrere Prozeduren (in NQC ‚tasks‘ genannt) im zweiten Programm geeignet zu koordinieren, zu so großen Schwierigkeiten, dass nur zwei Gruppen ein lauffähiges Programm erstellen konnten. Ich hatte diese Schwierigkeiten antizipiert und den Schülern Teile des Algorithmus vorgegeben, denke jedoch, dass es besser gewesen wäre, Teile der NQC-Lösung vorzugeben. Auch wenn man dadurch einen Teil der selbständigen Programmentwicklung in Frage gestellt hätte. Für eine arbeitsteilige Lösung der Programmieraufgabe durch alle Gruppen erscheint der Leistungsstand der Schüler noch nicht auszureichen.

Ein einfacheres Beispiel für die Verwendung des Helligkeitssensors wäre z.B. das ‚Finden einer Lichtquelle durch den Roboter‘ gewesen. Das ‚Linienfolger‘-Problem würde sich ebenfalls einfacher gestalten,

⁵⁰ Siehe Anhang A) und B): zwei ausgewählte NQC-Programme der Schüler.

wenn zwei Helligkeitssensoren verwendet würden, da dann die Richtung der Abweichung von der Markierung sofort ermittelt werden könnte.

7.6. Was noch zu tun wäre

Das Interesse am Thema Robotik war so stark, dass einige Schüler noch mehr Stoff behandelt sehen wollten. MF. formuliert seine Wünsche: *“Zum Beispiel hätten wir auf das automatische Haus eingehen können,...wir haben uns zu sehr mit dem LEGO-Roboter beschäftigt.“*

Und V. *„hätte gern selbst mal so einen Roboter gebaut“*. Den elektromechanischen Aspekt empfand SV. unterrepräsentiert: *„Was mir wirklich gefehlt hat, war das Eingehen auf die Funktionsweise der Sensoren...“*. Man kann also nicht immer alle Wünsche aller Schüler im Unterricht erfüllen. Dennoch meine ich, dass mit der Reihe ‚Robotik‘ Interessen bei den Schülern geweckt wurden und dass ein Teil der zukünftigen Lebenswelt für sie transparenter geworden ist.

F. liefert mir in seinem Kurzaufsatz ein hoffnungsvolles Schlusswort: *„Was mir dieses Thema gebracht hat, wird sich zeigen, da ich mich damit noch außerschulisch beschäftigen werde, denke ich, dass es mir auf jeden Fall einen Stoß gegeben hat.“*

8. Literaturverzeichnis

ABEL, J., KÜPER, K.: Roboter zum Anfassen – Eine Unterrichtsreihe zum Fach Technik in der Realschule; In: Login 3/95; S. 37-42.

ALTENBURG, J., ALTENBURG, U.; Mobile Roboter – Vom einfachen Experiment zur Künstlichen Intelligenz; München; 1999.

ASIMOV, I.; Vernunft; In[Asimov, I.]: Robotervisionen; Bastei-Verlag; Bergisch Gladbach 1992.

BACHMANN, K.; Künstliches Leben – Die Wesen aus dem Morgenland. In: GEO 3/98; S.94-112.

BAUM, D.; NQC Programmieranleitung; Version 2.0 rev 2;
<http://www.enteract.com/~dbaum/nqc>

BOEGNER, S, BÜCHS, K.-D., KÜHNE, P.; Die Roboter kommen – Der Weg ins Paradies oder zum Arbeitsamt? In: Login 5-6/88; S.44-49.

CVK FISCHERTECHNIK ; Computing Schulprogramm; Cornelsen Experimenta; Berlin 1989.

DECHAU, C.-P., PSIHOYOS, L.; Die Käfer kommen – Mikro Roboter; In: GEO 5/91; S. 42-56.

DEKER, U.; Künstliche Intelligenz; Aus der Reihe: Enzyklopädie der Informationsverarbeitung; IBM Deutschland; Stuttgart 1989.

DUDEN INFORMATIK – Ein Sachlexikon für Studium und Praxis;
2.Auflage; Dudenverlag; Mannheim 1993.

ENCARTA ENZYKLOPÄDIE; Microsoft; 1999.

GINTHUM, S.; LEGO Dacta Unterrichtsmedien & Lernkonzepte – Lernen für die Zukunft; Technik-LPE GmbH; Eberbach 2001.

HAHN, DZEWAS; Informatik für die Sekundarstufe II; Band 2;
Westermann; Braunschweig 1991.

JACOBS, J.; Didaktische Gesichtspunkte bei der Behandlung des Themas ‚Industrieroboter‘; In: Login 5-6/88; S. 23-28.

JANK, W., MEYER, H.; Didaktische Modelle; Cornelsen-Verlag; Frankfurt 1991.

KNUDSEN, B.K., NOGA,M.; Das Inoffizielle Handbuch für LEGO Mindstorms Roboter; O'Reilly-Verlag; Köln 2000.

KOERBER, B.; Automatisierung – Mobile Roboter in Industrie, Wirtschaft und Unterricht; In: Login 1998/5; S.10.

LEGO [2001]; Lego Dacta – Learning Concepts; (Informationsbroschüre, Bezug über: www.lego.com/dacta)

LEGO [2000]; Lego Dacta Unterrichtsmedien & Lernkonzepte – Lernen für die Zukunft; Technik LPE; Eberbach 2000.

LEGO [1998]; Lego - Mindstorms – Robotics Invention System 1.5 – Constructopedia; (Bau- und Bedienungsanleitung, Bestandteil des Baukastens).

LEGO [1988]; Lego TC Newsletter Nr.1/88: Unterrichtsvorschläge zur Informationstechnischen Grundbildung; In: Login 5-6/88; S.105-108.

LEHMANN, E.; Projektarbeit im Informatikunterricht; Teubner-verlag; Stuttgart 1985.

LOUIS, D.; LEGO Mindstorms: Roboter programmieren lernen; X-Games-Verlag; München 1999.

MARTIN, W.; Industrieroboter – Aufbau, Einsatz und Auswirkungen; In: Login 5-6/88; S. 15-22.

PAPERT, S.[1994]; Revolution des Lernens – Kinder, Computer, Schule in einer digitalen Welt; Heise-Verlag; Hannover 1994.

PAPERT, S.[2000]; Interview mit Michael West; Internet: www.brueckenbauer.ch/papert.html

POSTMAN, N.; Keine Götter mehr – Das Ende der Erziehung; dtv; München 1997.

RECHENBERG, P.; Was ist Informatik? Eine allgemeinverständliche Einführung; 3.Auflage; Hanser-Verlag; München 2000.

ROLLKE, K.H., SENNHOLZ, K.; Grund- und Leistungskurs Informatik; Cornelsen-Verlag; Düsseldorf 1992.

SCHRÖDER, U.S.; Prozesssteuerung im Informatikunterricht; In: Login 8/88; S.29-32.

SCHWARZE,H., HOLZGREFE, H.-W.; Computereinsatz beim Regeln und Steuern; Teubner-verlag; Stuttgart 1987.

SENATSVORWALTUNG FÜR SCHULE, BERUFSBILDUNG UND SPORT:
Vorläufiger Rahmenplan für Unterricht und Erziehung in der
Berliner Schule – Gymnasiale Oberstufe – Fach Informatik;
Berlin 1993.

STRUCK, P.; Netzwerk Schule – Wie Kinder mit dem Computer das
Lernen lernen; Hanser-Verlag; München 1998.

THODE, R.; Steuern und Regeln mit dem Computer – Eine Einführung
in LEGO-LINES; In: Login Sonderheft ITG 9/89; S.91-94.

8. Anhang

A) NQC-Programm ‚Hindernis umfahren‘ von MF. und SV.

```
//Autor: MF., SV.  
//Datum 4.7.2001  
  
task main()  
{  
    SetSensor(SENSOR_1, SENSOR_TOUCH);  
  
    SetPower(OUT_C, OUT_FULL);  
    SetPower(OUT_A, 2);  
    OnRev(OUT_A + OUT_C);  
    if (SENSOR_1 != 1){OnFor(OUT_A + OUT_C, 5);  
    }  
    else  
    {while (1 == 1)  
    {  
        while (SENSOR_1 == 0) {  
            SetPower(OUT_A,1);  
            OnFor(OUT_A + OUT_C,20);  
        }  
        while (SENSOR_1 == 1){  
            Float(OUT_C);  
            OnFor(OUT_A,2);  
        }  
    }  
    }  
}
```


B) NQC-Programm ‚Linienfolger‘ von N. und F.

```
//Autoren: N., F.
//Version: 1.0.5 Sternzeit:2001 der 11.07
#define links 1
#define rechts 2
int turn;
int richtung;

task main(){
    richtung = rechts;
    SetSensor(SENSOR_2,SENSOR_LIGHT);
    SetPower(OUT_A+OUT_C,OUT_FULL);
    start fahrt;
    while (true){
        if (SENSOR_2 > 36 && SENSOR_2 < 50){
            stop fahrt;
            start korrektur;
        }
        if (SENSOR_2 < 35){
            stop korrektur;
            start fahrt;
        }
        if (SENSOR_2 >50){
            Off(OUT_A+OUT_C);
            StopAllTasks();
        }
        Wait(30);           // ergänzt von M.Aabend
    }
}

task fahrt(){
    while (true){
        OnFwd(OUT_A+OUT_C);
    }
}

task korrektur(){
    turn=25;
    drehen();
}

sub drehen(){
    if (richtung==rechts){
        Rev(OUT_C);
        Fwd(OUT_A);
        Wait(turn);
        richtung=links;
        turn= turn + 10;
    }
    else {
        Rev(OUT_A);
        Fwd(OUT_C);
        Wait(turn);
        richtung=rechts;
        turn= turn + 10;
    }
}
```

Jack & Taz

Informatik-Basiskurs

Umgangssprachlicher Algorithmus für ein Programm „Liniengänger“:

- 1. • Wenn ich über der Linie bin, gehe geradeaus.
- 2. • Wenn ich nicht über der Linie bin {
 - Drehe in die umgekehrte Richtung als die, in der ich mich letztmal gedreht habe.
 - Wenn ich die Linie immer noch nicht finden kann, dann drehe noch weiter, aber wieder zurück. }
 - Fahre solange, bis deine Hand ansetzt und ~~senke~~ ^{senke} Werte für p_{acc} ^{see (Schwingeren)}
 - Drehe und suche Wert und fahre ihn weiter (wenn nicht, dann 2. Schritt)
 - Stoppe bei bestimmten Wert!

Idee: Algorithmus	Test: Beobachtung / Fehler
	<ul style="list-style-type: none"> - fährt nur geradeaus - hebt sich nur 10cm für ein wenig aus - danach wird noch hineingedrückt, strömt da gas und gas nicht "in Tank")
<ul style="list-style-type: none"> - auf der schraube keine gerade aus fahren (if stat. werte) 	<ul style="list-style-type: none"> - fährt zwar, aber dreht sich nicht nach rechts
<ul style="list-style-type: none"> - 2. "off" mit in die "unstable" - schleife - last "hell" wird immer und wird gestartet! (statt fast "sub") 	<ul style="list-style-type: none"> - fährt jetzt immer rückwärts!
<ul style="list-style-type: none"> - neue task "hellruck" - fährt immer noch rückwärts - Motoren gehen nicht ganz aus (nicht alle Motoren) - dreht nur in eine Richtung 	<ul style="list-style-type: none"> - immer noch nicht - wenn es immer noch auf dem schrauben streifen ist, startet er "hellruck" (28 Zeile zugezogen)! - Klammer zurück - In Zeile 5, Programm aufend

Senior	30	40	60
Time	135	135	135

ackwerk
~~1-2 Jahre~~ aber
gut / schlecht, / ~~Wetter~~

- wait (waggen) zum Drehen
weggenommen!
- "Tosge" genommen! (verwendet)