

Didaktische Entwürfe für den Kompetenzbereich Industrielle Informationstechnik

DIPLOMARBEIT

zur Erlangung des akademischen Grades

Magister der Sozial- und Wirtschaftswissenschaften

im Rahmen des Studiums

Informatikmanagement

eingereicht von

Dipl.-Ing. (FH) Markus Brunner

Matrikelnummer 0827574

an der
Fakultät für Informatik der Technischen Universität Wien

Betreuung
Betreuerin: Ass.Prof. Dr.techn. Monika Di Angelo

Wien, 15.03.2012

(Unterschrift Verfasser)

(Unterschrift Betreuerin)

Abstract

Educational standard and competence orientation are two key topics in the current educational debate. The declared objectives in this process are a sustainable increase of quality and a better comparability of educational attainment. The Federal Ministry of Education, Arts and Culture holds responsibility for the practical implementation in the vocational school system. As part of the project „Bildungsstandards in der Berufsbildung“ („standard of education in vocational education“), competence models, descriptors and didactic examples - not least for the field of IT in a technical high school (HTL) - were compiled. The implementation of this project implies a paradigm shift of the vocational school system, which requires a new pedagogic foundation.

This thesis primarily concentrates on the construction of a didactical master plan, that meets the requirements of educational standards in general and competence orientation in particular. The focus lies on the competence sector industrial information technology (INIT) of an IT-HTL. The implementation is carried out on the example „HTL Krems“. The foundation of this thesis is an extensive literature research. As a second step, the initial situation, basic requirements and the outcome will be interpreted with the method „analysis“. As a last and final step, the gained results are transferred to the teaching concepts.

A proposal for implementation of these educational standards in the competence sector INIT can be considered the central conclusion. The specification is based on selected teaching sessions for microcontroller technique. The application of student orientated teaching methods, which is bound to certain basic requirements, is represented. During the training sessions, procedures to promote the intrinsic motivation and the creativity in general are applied. In general, it shows that the competence sector INIT conveys remarkably demanding learning outcomes, whose implementation on multiple levels is challenging.

Keywords: computer science education, didactics, educational standards, microcontroller technique, embedded systems, technical high school

Kurzfassung

Bildungsstandards und Kompetenzorientierung sind zwei zentrale Begriffe der gegenwärtigen Bildungsdiskussion. Hierbei können eine nachhaltige Qualitätssteigerung der Ausbildungsgänge sowie eine bessere Vergleichbarkeit von Bildungsabschlüssen als die erklärten Ziele angesehen werden. Für die praktische Umsetzung im Bereich des berufsbildenden Schulwesens zeichnet sich das Bundesministerium für Unterricht, Kunst und Kultur verantwortlich. Im Rahmen des Projektes „Bildungsstandards in der Berufsbildung“ wurden Kompetenzmodelle, Deskriptoren und prototypische Unterrichtsbeispiele - nicht zuletzt auch für den Fachbereich Informationstechnologie (IT) einer Höheren Technischen Lehranstalt (HTL) - erarbeitet. Mit der Umsetzung dieses Projektes kommt ein Paradigmenwechsel auf das berufsbildende Schulwesen zu, der eine neue pädagogische Grundlage erfordert.

Diese Arbeit geht primär der Frage nach, wie sich ein didaktisches Gesamtkonzept gestalten muss, das den Anforderungen der Bildungsstandards im Allgemeinen und der Kompetenzorientierung im Besonderen gerecht wird. Im Fokus steht hierbei der Kompetenzbereich Industrielle Informationstechnik (INIT) einer IT-HTL. Die Durchführung erfolgt am Beispiel der HTL-Krems. Die Grundlage der Arbeit bildet eine umfassende Literaturrecherche. Im zweiten Schritt werden durch den Einsatz der Methode „Analyse“ die Ausgangssituation sowie die Rahmenbedingungen untersucht und die Erkenntnisse interpretiert. Letztendlich werden im dritten Schritt die gewonnen Ergebnisse in die Unterrichtskonzeption überführt.

Als zentrales Ergebnis kann ein Vorschlag zur Umsetzung der Bildungsstandards im Kompetenzbereich INIT angesehen werden. Die Konkretisierung erfolgt anhand ausgewählter Unterrichtseinheiten für die Mikrocontrollertechnik. Der Einsatz von schülerzentrierten Unterrichtsmethoden, der an bestimmte Rahmenbedingungen geknüpft ist, wird aufgezeigt. Bei den Übungseinheiten werden Maßnahmen zur Förderung der intrinsischen Motivation und der Kreativität im Allgemeinen implementiert. Generell zeigt sich, dass der Kompetenzbereich INIT überaus anspruchsvolle Lernziele formuliert, deren Umsetzung auf mehreren Ebenen herausfordernd ist.

Schlagworte: Informatikunterricht, Didaktik, Bildungsstandards, Mikrocontrollertechnik, Embedded Systems, Höhere Technische Lehranstalt

Eidstattliche Erklärung

Dipl.-Ing.(FH) Markus Brunner, 3500 Krems an der Donau, Langenloiserstraße 13/15

„Hiermit erkläre ich, dass ich diese Arbeit selbständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit - einschließlich Tabellen, Karten und Abbildungen -, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quellen als Entlehnung kenntlich gemacht habe.“

Krems, 15.3.2012

Markus Brunner

Danksagung

Ich bin dankbar, nicht weil es vorteilhaft ist, sondern weil es mir Freude macht.

Lucius Annaeus Seneca

An dieser Stelle möchte ich mich bei allen bedanken, die mir mit Rat und Tat und vor allem mit motivierenden Worten beim Verfassen dieser Diplomarbeit zur Seite gestanden haben.

Besonderer Dank gilt AV Prof. Dipl.-Ing. Anton Hauleitner, HTL Krems, für die Bereitstellung von Informationen und das kompetente Feedback.

Ebenso bedanke ich mich bei Ass.Prof. Dipl.-Ing. Dr.techn. Monika Di Angelo, TU Wien, für die wertvollen und konstruktiven Beiträge.

Ein herzliches Dankeschön geht an meine Lebensgefährtin, Ingrid, die mir während des Studiums in vielerlei Hinsicht eine große Unterstützung war.

Inhaltsverzeichnis

Abstract	i
Kurzfassung	ii
Eidstattliche Erklärung	iii
Danksagung	iv
1 Einleitung	1
1.1 Ausgangslage	1
1.2 Ziele	2
1.3 Vorgehensweise und Aufbau der Arbeit	2
2 Theoretische Grundlagen	4
2.1 Pädagogisch-psychologischer Blickwinkel	4
2.1.1 Lernformen im Überblick	5
2.1.2 Lerntheorien - Konstruktivismus	6
2.1.3 Aspekte der Lernpsychologie	7
2.2 Didaktik im Kontext der Informatik	8
2.3 Berufsbildende Schulen in Österreich	9
2.3.1 Technische, gewerbliche und kunstgewerbliche Schulen	9
2.3.2 Einordnung der berufsbildenden mittleren und höheren Schulen	11

3	Bildungsstandards und Kompetenzorientierung	13
3.1	Ziele und Aufgaben	13
3.2	Kompetenz und Kompetenzorientierung	14
3.3	Kompetenzmodell	15
3.4	Deskriptoren	17
4	Unterricht im Kontext der Kompetenzorientierung	19
4.1	Definition von Unterricht	19
4.2	Merkmale eines kompetenzorientierten Unterrichts	20
4.3	Unterrichtsplanung	22
4.3.1	Bestandteile der Unterrichtsplanung	22
4.3.2	Vorgehensweise bei der Unterrichtsplanung	23
4.4	Unterrichtsdurchführung	23
4.4.1	Grundlegende Formen handlungsorientierter Methoden	26
4.4.2	Spezifische Vertreter handlungsorientierter Methoden	28
4.5	Leistungsfeststellung und -beurteilung	35
4.5.1	Lernvertrag	35
4.5.2	Lerntagebuch	36
4.5.3	Portfolio	37
5	Analyse der Ausgangssituation	38
5.1	Höhere Technische Lehranstalt für Informationstechnologie	38
5.1.1	Ziele und Inhalte des Kompetenzbereichs Industrielle Informations- technik	41
5.1.2	Allgemeine Bildungsziele	42
5.1.3	Didaktische Grundsätze	42
5.1.4	Bildungsstandards und prototypische Beispiele	43
5.2	HTBL Krems - Abteilung für Informationstechnologie	45
5.2.1	Implementierung der Industriellen Informationstechnik	46
5.2.2	Interdisziplinärer Unterricht - Potenziale und Möglichkeiten	47
5.2.3	Konkretisierung der Erwartungen an die Vorbildung	52
5.3	Embedded Systems als Lehrgegenstand	55
5.3.1	Kursimplementierungen	56
5.3.2	Motivations- und Kreativitätsförderung	62
5.4	Synthese der Anforderungen an die Unterrichtskonzeption	65
5.4.1	Anforderungscluster A: Pädagogik und Lernpsychologie	66

5.4.2	Anforderungscluster B: Kompetenzorientiertes Unterrichten	66
5.4.3	Anforderungscluster C: Gesetzliche und institutionelle Vorgaben . . .	67
5.4.4	Anforderungscluster D: Embedded System als Unterrichtsgegenstand	68
6	Unterrichtskonzeption	71
6.1	Überblick und grundlegende Annahmen	71
6.1.1	Auswahl und Strukturierung der Lehrinhalte	72
6.1.2	Durchführung und Zielsetzung	74
6.2	Theorieeinheiten	75
6.2.1	T1 - Einführung und thematische Einordnung	76
6.2.2	T2 - Der Mikrocontroller	79
6.2.3	T3 - Leistungsklassen und Auswahlkriterien	84
6.2.4	T4 - Relevante Baugruppen und deren Bedeutung	87
6.3	Übungseinheiten	91
6.3.1	UE1 - Embedded Software Entwicklung	92
6.3.2	UE2 - How to use a port	97
6.3.3	UE3 - Anbindung peripherer Komponenten	105
6.3.4	UE4 - E/A-Baugruppen	108
6.3.5	UE5 - Finalisierung Gruppenprojekt 2	114
6.4	Projekte	114
6.5	Übungskatalog	116
6.6	Leistungsbeurteilung	116
7	Zusammenfassung und Diskussion	118
7.1	Zusammenfassung	118
7.2	Diskussion	119
A	Anhang	122
A.1	Kompetenzmodell Informationstechnologie	123
A.2	Programmierung des Arduino Ethernet-Boards	124
A.3	Übungsblatt Bitmanipulation	128
A.4	Ergänzende Unterrichtsmaterialien	129
	Literatur	133

Abkürzungsverzeichnis

ADC	Analog Digital Converter
AHS	Allgemeinbildende Höhere Schule
ALU	Arithmetic Logic Unit
BMHS	Berufsbildende Mittlere und Höhere Schulen
BMUKK	Bundesministerium für Unterricht, Kunst und Kultur
C#	Programmiersprache C-Sharp
CAN	Controller Area Network
CPU	Central Processing Unit
EEPROM	Electrically Erasable Programmable Read-only Memory
EK	Evaluation Kit
ETH	Eidgenössische Technische Hochschule Zürich
EU	Europäische Union
EQR	Europäischer Qualifikationsrahmen
DiP	Durchführung informationstechnischer Projekte
HTBL	Höhere Technische Bundeslehranstalt
GINF	Grundlagen der Informatik
HW	Hardware
HKUST	Hong Kong University of Science and Technology
IC	Integrated Circuit
IDE	Integrated Development Environment
IMST	Innovationen Machen Schulen Top
INIT	Industrielle Informationstechnik
Ing	Ingenieur/Ingenieurin
ISCED	International Standard Classification of Education

ISR	Interrupt Service Routine
IT	Informationstechnologie
IP	Internet Protocol
ISO	Internationale Organisation für Normung
KTH	Königlich Technische Hochschule
LBVO	Leistungsbeurteilungsverordnung
LCD	Liquid Crystal Display
LED	Light Emitting Diode
MSR	Mess-, Steuer- und Regeltechnik
NQR	Nationaler Qualifikationsrahmen
OOP	Objektorientierte Programmierung
OSI	Open Systems Interconnection
PBL	Problem-Based Learning
PC	Personal Computer
PM	Projektmanagement
PPuP	Prozessdatenverarbeitung, -visualisierung u. -kommunikation
PWM	Pulsweitenmodulation
QM	Qualitätsmanagement
RTOS	Real Time Operating System
RWT	Rheinisch-Westfälische Technische Hochschule
SEP	Softwareentwicklungsprozess
SPI	Serial Peripheral Interface
SPS	Speicherprogrammierbare Steuerung
SRAM	Static Random Access Memory
SuB	Schnittstellen und Bibliotheken
SW	Software
TCP	Transmission Control Protocol
TU	Technische Universität
μC	Mikrocontroller
UML	Unified Modeling Language
USART	Universal Asynchronous Receiver Transmitter
ZbE	Zeitgeberbasierte Einheiten

1.1 Ausgangslage

Österreichs Schulsystem befindet sich im Wandel: Dies zeigt sich an Maßnahmen, die bereits umgesetzt wurden [1], beziehungsweise deren Umsetzung kurz bevor steht [2]. Der vermeintliche Wandel wird jedoch nicht immer als solcher wahrgenommen und die getätigten Maßnahmen entsprechen nicht immer zur Gänze den Vorstellungen [3, 4, 5]. Nicht zuletzt deshalb birgt das Thema „Bildung der Zukunft“ eine gewisse gesellschaftspolitische Brisanz in sich und wird weiterhin im Diskussionsmittelpunkt stehen. Wie auch immer sich das zukünftige Bildungssystem - insofern man überhaupt von einem Endzustand sprechen kann - im Detail gestalten wird, eine konkrete Maßnahme hat der angesprochene Wandel bereits mit sich gebracht: die Einführung von Bildungsstandards und der damit verbundenen Kompetenzorientierung.

Seit einigen Jahren arbeiten die entsprechende Sektion des Bundesministeriums für Unterricht, Kunst und Kultur (BMUKK) an der Entwicklung von Bildungsstandards im berufsbildenden Schulwesen [6]. Das umfasst auch den Bereich der Höheren Technischen Lehranstalten (HTL). Mit dem Bundesgesetzblatt BGBl. II Nr. 300/2011 [7] wurde die rechtliche Grundlage für Bildungsstandards im HTL-Bereich geschaffen, indem über sechs Fachrichtungen - inklusive der Fachrichtung Informationstechnologie (IT) - ein neuer, kompetenzorientierter Lehrplan verordnet wurde. Nachdem die HTL-Krems [8] unter anderem über eine IT-Abteilung verfügt, wird seit Beginn des Schuljahres 2011/2012 gemäß des neuen Lehrplans gelehrt - zumindest in den gegenwärtig ersten und zweiten Klassen. Die anderen Schulstufen sind von dieser Umstellung nicht mehr betroffen.

Bildungsstandards können als konkrete und verbindliche Erwartungen an die zu erbringenden Lernergebnisse angesehen werden. Sie legen jene Kompetenzen fest, die vonseiten der SchülerInnen bis zu einem definierten Zeitpunkt zu erlangen sind. Als Grundlage hierfür dient ein sogenanntes Kompetenzmodell, das unterschiedliche Handlungskompetenzen für die jeweiligen Lehrinhalte formuliert. Es ist jedoch nicht ausschließlich nur die Entwicklung der entsprechenden Fach- und Methodenkompetenz ein erklärtes Ziel, ebenso befinden sich die Sozial- und Personalkompetenz im Fokus der Ausbildung.

Es wird deutlich, dass mit der Verordnung des neuen Lehrplans zwei zentrale Änderungen eintreten - einerseits neue Lehrinhalte und andererseits die Kompetenzorientierung. Die Adaptierung der Lehrinhalte ist wohl der Dynamik des Fachbereichs IT geschuldet. So fanden beispielsweise Lehrinhalte Berücksichtigung, die der überaus bedeutenden Disziplin „Embedded Systems“ (dt. eingebettete Systeme) zuzuordnen sind. Der hierfür definierte, neue Kompetenzbereich lautet „Industrielle Informationstechnik“ (INIT), der sich auch im Fokus dieser Arbeit befindet. In Bezug auf die Kompetenzorientierung bleibt anzumerken, dass ein Paradigmenwechsel im berufsbildenden Schulwesen erfolgt, der Herausforderungen auf mehreren Ebene mit sich bringt.

1.2 Ziele

In den Klassenzimmern stellt nach wie vor der sogenannte Frontalunterricht die bestimmende Unterrichtsmethode dar. Das zentrale Charakteristikum dieser Methode ist, dass die SchülerInnen die passive Rolle des Informationsempfängers einnehmen, indem sie den Ausführungen der Lehrkraft folgen. Das steht im Widerspruch zu den einleitend formulierten Merkmalen der Kompetenzorientierung. Ziel dieser Arbeit ist es, didaktische Konzepte für den neuen Kompetenzbereich INIT zu entwickeln, die den Anforderungen der Kompetenzorientierung gerecht werden - also neben der Fach- und Methodenkompetenz auch die Aspekte der Personal- und Sozialkompetenz berücksichtigen. Nachdem die HTL-Krems stets dem Puls der Zeit folgt und somit großes Interesse am Mitwirken bekundet hat, erfolgt die Durchführung dieser Arbeit am Beispiel der IT-Abteilung der HTL-Krems. Insgesamt betrachtet ergeben sich somit folgende zentrale Fragestellungen:

- F1:** Wie muss Unterricht gestaltet sein, um die Bildungsstandards des Kompetenzbereichs Industrielle Informationstechnik zu erfüllen?
- F2:** Wie kann die individuelle Lernzielerreichung auf fachlicher Ebene sichergestellt werden, ohne die Entwicklung personaler und sozialer Kompetenzen zu vernachlässigen?
- F3:** Welche Bedeutung wird der schulspezifischen Implementierung des Lehrplans (beim Einsatz handlungsorientierter Methoden) zuteil?
- F4:** Wie muss der Unterricht gestaltet sein, damit Kreativität und intrinsische Motivation entstehen können?

1.3 Vorgehensweise und Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich vorrangig in einen Grundlagen-, Analyse- und Konzeptionsteil. Zur Erarbeitung der Grundlagen wird die Methode Literaturrecherche angewendet. Der Schwerpunkt richtet sich hierbei auf jene Themen, die für die weitere Arbeit von zentraler Bedeutung sind. Darauf aufbauend erfolgt durch den Einsatz der Methode „Analyse“ eine systematische Untersuchung und Interpretation der Ausgangssituation mit dem Ziel, konkrete Anforderungen an die Unterrichtskonzeption zu definieren. Den Abschluss bildet

der Konzeptionsteil mit der Entwicklung eines didaktischen Gesamtkonzepts und der Konkretisierung ausgewählter Unterrichtssequenzen. Nachfolgend werden die einzelnen Kapitel überblicksartig vorgestellt:

Kapitel 1: Einleitung

Kapitel 2: Theoretische Grundlagen Dieser Abschnitt behandelt die Grundformen des Lernens und zeigt hierbei besondere Merkmale auf. Mit dem Konstruktivismus wird eine Lerntheorie vorgestellt, die vor allem in Bezug auf die Kompetenzorientierung eine zentrale Rolle einnimmt. Dass der Schultyp HTL eine internationale Sonderstellung einnimmt, wird ebenso aufgezeigt wie dessen unterschiedliche Ausbildungsformen.

Kapitel 3: Bildungsstandards und Kompetenzorientierung Einleitend wird festgehalten, dass Bildungsstandards als konkrete und verbindliche Erwartungen an die zu erbringenden Lernergebnisse angesehen werden können. Dieser Abschnitt erläutert diesbezügliche Hintergründe. Anhand des IT-Kompetenzmodells wird die Funktionsweise von Bildungsstandards und der zugrundeliegenden Kompetenzorientierung dargelegt.

Kapitel 4: Unterricht im Kontext der Kompetenzorientierung Nachdem die Entwicklung von Unterrichtssequenzen ein Ziel dieser Arbeit ist, kommt den Hauptbausteinen dieses Prozesses - also der Frage nach dem „Wie?“ - eine große Relevanz zu. Im Detail zählen hierzu die Unterrichtsplanung und -durchführung sowie die Leistungsbeurteilung. Dieser Abschnitt behandelt die einzelnen Bausteine im Kontext der Kompetenzorientierung und zeigt hierbei wesentliche Herausforderungen als auch Möglichkeiten hinsichtlich deren Gestaltung auf.

Kapitel 5: Analyse der Ausgangssituation Grundsätzlich ist Unterricht eine organisierte Form der Interaktion zwischen SchülerInnen und Lehrkraft. Um diese Organisation zu schaffen, gilt es eine Fülle an Einflussfaktoren zu berücksichtigen. Die Erhebung und Bewertung dieser Faktoren ist Bestandteil dieses Abschnitts. Hierzu zählen beispielsweise die Erhebung jener Kompetenzen, welche die SchülerInnen im Kompetenzbereich INIT erlangen sollen und der Anforderungen, die aufgrund der schulspezifischen Implementierung des Kompetenzbereichs zu berücksichtigen sind. Um die didaktische Frage „Was soll unterrichtet werden?“ zu beantworten, wird in diesem Abschnitt der State of the Art bezüglich „Embedded Systems als Unterrichtsgegenstand“ untersucht. Als Ergebnis dieses Kapitels stehen konkrete Anforderungen an die Unterrichtskonzeption zur Verfügung.

Kapitel 6: Unterrichtskonzeption Die Formulierung eines Gesamtkonzepts ist das Ziel dieses Abschnitts. Hierbei werden einerseits grundlegende Annahmen hinsichtlich Strukturierung und Durchführung ausgewählter Unterrichtssequenzen getroffen und diese in weiterer Folge auch erarbeitet. Als Grundlage dienen die Erkenntnisse der vorangegangenen Kapitel. Das Gesamtkonzept umfasst neben den Theorie- und Übungseinheiten auch Projekte sowie einen Übungskatalog und die Leistungsbeurteilung.

Kapitel 7: Zusammenfassung und Diskussion

Kapitel 2

Theoretische Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen, die für das weitere Verständnis der Arbeit von Bedeutung sind, behandelt. Einleitend erfolgt eine pädagogisch-psychologische Vorbetrachtung mit dem Ziel, die Mannigfaltigkeit des Lernens zu verdeutlichen und mit dem Konstruktivismus eine bedeutende Lerntheorie vorzustellen. Nachdem diese Arbeit die Entwicklung didaktischer Konzepte für eine berufsbildende höhere Schule, Fachbereich Informationstechnologie (IT), verfolgt, wird die Didaktik als Fachwissenschaft behandelt. Den Kapitelabschluss bildet eine überblicksartige Darstellung des berufsbildenden Schulwesens in Österreich. Hierbei wird der Frage „Wie charakterisieren sich die technisch-gewerblichen Schulen und welche Bedeutung wird diesem Schultyp im österreichischen Schulwesen zuteil?“ nachgegangen werden

2.1 Pädagogisch-psychologischer Blickwinkel

Laut dem Schulorganisationsgesetz [9] liegt die primäre Aufgabe der Institution Schule in der Vermittlung von Wissen und Können mit dem Ziel, die Jugend für das (Berufs-)Leben vorzubereiten. Um das zu bewerkstelligen, wird im Rahmen eines Unterrichts (siehe Kapitel 4.1) institutionelles Lehren und Lernen vollzogen. Jedoch beschränkt sich „Lernen“ nicht nur auf die Schule; Lernen ist vielmehr ein zentraler Vorgang im täglichen Leben eines Menschen, der im Alltag (bereits vor als auch nach dem Besuch der Institution Schule) stattfindet. Die Literatur stellt zahlreiche Definitionen des Begriffs Lernen zu Verfügung. Schröder definiert Lernen in [10] wie folgt:

Lernen bewirkt eine relativ dauerhafte Verhaltensänderung aufgrund von Erfahrungen.

Diese Erfahrungen, die auf der Aufnahme, der Verarbeitung und der Speicherung von Wissen beruhen, können durch unterschiedliche Lernformen gemacht werden. Nachfolgende Unterkapitel liefern eine zusammenfassende Darstellung dieser Lernformen und praxisrelevanten Lerntheorien.

2.1.1 Lernformen im Überblick

Abbildung 2.1 zeigt eine mögliche Gruppierung von Lernformen. Schröder versteht darunter Variationsmöglichkeiten des Lernens, die in Abhängigkeit unterschiedlicher Lernfaktoren auftreten können.

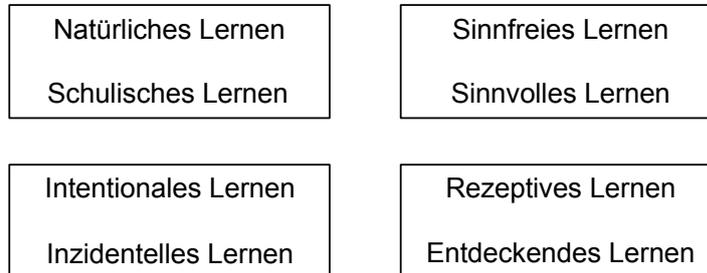


Abbildung 2.1: Lernformen (Eigendarstellung nach [10])

Natürliches Lernen ist jene Form des Lernens, die jeder Mensch von Kindheitstagen anwendet. Dies geschieht in der Regel aus Eigenmotivation. Hilfe wird nur bei Unüberwindbarkeit einer „Barriere“ in Anspruch genommen. Klassische Merkmale dieser Lernform sind beispielsweise hohe Eigenaktivität oder andauernde Konzentration. Demgegenüber steht das schulische Lernen (im Rahmen eines Unterrichts). Bei dieser Lernform findet das Lernen weniger freiwillig statt, sondern wird gefordert. Noch dazu werden die Lernziele (unabhängig von den Bedürfnissen einzelner) vorgegeben. Ziel der Schule muss sein, das schulische Lernen dem natürlichen möglichst anzunähern.

Weitere Lernformen stellen sinnfreies sowie sinnvolles Lernen dar. Unter sinnfreiem Lernen versteht man das klassische Auswendiglernen. Hierbei werden Inhalte nicht unbedingt verstanden, sondern einfach nur reproduziert. Im Gegensatz dazu steht das sinnvolle Lernen. Hierbei ist es das Ziel, Zusammenhänge zu erkennen und Einsicht zu gewinnen. Das sogenannte „Problem-basierte Lernen“ (siehe Kapitel 4.4.2) kann beispielsweise dem sinnvollen Lernen zugeordnet werden.

Gilt es vonseiten der Lernenden konkrete Lernziele zu erbringen, spricht man vom intentionalen Lernen. Wobei es in diesem Zusammenhang nicht von Bedeutung ist, wer den Lernauftrag erteilt. Das kann einerseits die Lehrkraft aber andererseits auch der Lernende sein, indem dieser beabsichtigt, etwas Bestimmtes zu lernen. Demgegenüber erfolgt beim inzidentellen Lernen das Lernen ohne konkreten Auftrag. Das Lernen erfolgt nebenher, ohne dass ein Lernziel bewusst vereinbart wird. Erwähnenswert ist in diesem Zusammenhang, dass vor allem schlechte Lernende - also jene Personen, die im Sinne des intentionalen Lernens die definierten Ziele nicht oder nur unzureichend erfüllen - besonders gut im Bereich des inzidentellen Lernens abschneiden. Also dort beachtenswerte Ergebnisse liefern, wo sie nicht verlangt werden. Schröder führt dieses Verhalten auf eine innere Ablehnung gegenüber Arbeitsaufträgen, die von der Lehrkraft erteilt werden, zurück.

Rezeptives beziehungsweise entdeckendes Lernen unterscheidet sich grundsätzlich in der Haltung des Lernenden. Beim rezeptiven Lernen übernehmen die Lernenden die Inhalte gemäß der didaktischen Aufbereitung der Lehrkraft. Es bleibt wenig Spielraum zur selbstständigen Konstruktion kongnitiver Strukturen. Im Gegensatz dazu erfolgt beim entdeckenden Lernen eine schöpferische Tätigkeit bei den Lernenden. Die Lehrkraft gibt einen Rahmen

vor, in welchem sich die Lernenden ausprobieren und somit Einsichten und Erkenntnisse gewinnen können.

2.1.2 Lerntheorien - Konstruktivismus

Die Annahme, Wissen lässt sich von der Lehrkraft (beziehungsweise von einem Lehrbuch) an Lernende einfach weiterreichen, ist überaus kritisch zu betrachten. Duit kommt diesbezüglich in [11] zu dem Schluss, dass das grundsätzlich „nicht möglich“ ist. Die Begründung liegt für Duit in der fehlenden Bedeutung der Sinnesdaten, die Lernende von der Lehrkraft empfangen. Wenn eine Lehrkraft ein Signal aussendet, indem sie beispielsweise einen Satz im Rahmen eines Vortrages von sich gibt, so hat dieser eine bestimmte Bedeutung in der Vorstellung der Lehrkraft. Die Lernenden verfügen jedoch (noch) nicht über diese Vorstellung. Daher sind sie gezwungen, diesen Satz gemäß ihrer Vorstellungskraft zu interpretieren - diesem also eine überaus subjektive Bedeutung zu verleihen. Oftmals deckt sich diese jedoch nicht mit jener der Lehrkraft. In der Praxis zeigt sich das daran, wenn die Lehrkraft nach erfolgter Darbietung neuer Inhalte eine Frage stellt, die es von den Lernenden zu beantworten gilt. Die Antworten entsprechen oft nicht oder nur in geringem Maße den Erwartungen der Lehrkraft, da sie der Antwort aufgrund der eigenen Vorstellung ebenfalls eine andere Bedeutung beimisst. Dieser Umstand kann zu erheblichen Irritationen führen.

Beim zuvor skizzierten einfachen Wissenstransport wird den Lernenden überwiegend eine passive Rolle (Wissensempfänger) zuteil. Im Gegensatz dazu steht die Theorie des Konstruktivismus, der in der heutigen Lehr- und Lernforschung eine zentrale Rolle einnimmt. Vereinfacht ausgedrückt verfolgt die konstruktivistische Sicht des Lernens, dass sich Lernende - basierend auf bereits vorhandenen Vorstellungen - ihr neues Wissen selbst konstruieren. Duit reduziert den Konstruktivismus auf folgende Kernaussage:

Jeder ist seines Wissens Schmied.

Humbert fasst die konstruktivistische Sicht auf Lehr- und Lernprozesse in [12] (basierend auf Glasersfeld [13]) folgendermaßen zusammen:

- *Erwerb von Fertigkeiten, d.h. von Handlungsmustern ist klar von der aktiven Konstruktion begrifflicher Netzwerke, also vom Verstehen zu unterscheiden.*
- *Methodische Hilfsmittel des Auswendiglernens und des Wiederholens im Training behalten ihren Wert, es wäre jedoch naiv zu erwarten, dass sie auch das Verstehen befördern.*
- *Die verbale Erklärung eines Problems führt nicht zum Verstehen.*
- *Die Lehrerin verfügt über ein adäquates Modell des begrifflichen Netzwerkes, innerhalb dessen die Schülerin neue Elemente einzupassen versucht.*
- *Lernen ist das Produkt von Selbstorganisation.*

Die Prozessmerkmale des konstruktivistischen Lernens werden durch ein Exzerpt aus Hubwieser [14] dargelegt:

- Aktive Beteiligung, Motivation und Interesse sind Grundvoraussetzungen. Nur so ist Lernen möglich.
- Lernen ist nur möglich, wenn Lernende Steuerungs- und Kontrollprozesse übernehmen.
- Um zu lernen, müssen individuelle Erfahrungen und eigene Interpretationen zentraler Bestandteil sein.
- Lernen muss situativ - also in spezifischen Kontexten - erfolgen.
- Lernen bedeutet auch Interaktion in sozialen Formen.

2.1.3 Aspekte der Lernpsychologie

Unterrichten heißt, mit Menschen zu interagieren (vgl. Kapitel 4.1). Menschen, die je nach Alterstufe und persönlichem Reifegrad unterschiedlicher nicht sein könnten. Dieser Umstand stellt eine große Herausforderung für die Lehrenden bei der täglichen Arbeit im Klassenzimmer dar. Forschungen im Bereich der Lernpsychologie haben interessante Erkenntnisse zu Tage gebracht, deren Kenntnis so manche Situationen im Klassenzimmer begreiflicher erscheinen lässt, oder eine effizientere Gestaltung des Unterrichts ermöglicht. Schröder [10] führt in diesem Zusammenhang folgende Punkte an:

Das sogenannte *Ebbinghaus¹-Gesetz* besagt, dass bereits bei einer geringfügigen Vermehrung des zu lernenden Stoffs die Anzahl der erforderlichen Wiederholungen, um die Inhalte zu verarbeiten, wesentlich mehr zunimmt. Dieses Faktum ist beispielsweise bei Hausübungen durchaus von Bedeutung.

Aus Sicht der Lehrkraft mag es oft unverstündlich erscheinen, dass ein Großteil erst kürzlich vorgestellter Inhalte relativ schnell wieder in Vergessenheit gerät. Was auf den ersten Blick auch nicht nachvollziehbar ist, wurden die Inhalte ja auch erst vor ein paar Tagen oder gar Stunden behandelt. Zurückzuführen ist das auf die sogenannte „Vergessenskurve“. Zu Beginn fällt die Kurve steil ab; dies bedeutet, dass der Verlust des Gelernten in der ersten Zeit am Größten ist. Danach läuft sie flach aus - das noch behaltene Wissen nimmt nur mehr geringfügig ab. Negativen Einfluss auf die Kurve übt beispielsweise Reizüberflutung aus. Dieser Aspekt ist bei der Aufbereitung von Unterrichtsinhalten sowie bei der Auswahl von Unterrichtsmedien von Bedeutung.

Ebenfalls erwiesen ist, dass die ersten und abschließenden Bestandteile einer Lernstrecke besonders schnell aufgenommen werden und überaus gut präsent bleiben. Problematischer sieht es mit den Bestandteilen dazwischen aus, diese Inhalte werden relativ leicht vergessen (siehe Vergessenskurve). Diesen Umstand kann man sich im Unterricht dahingehend zunutze machen, indem eine zusammenfassende Darstellung zentraler Inhalte entweder zu Beginn oder am Ende einer Lernstrecke platziert wird.

Damit der Aufbau neuer Wissensstrukturen möglich ist, müssen neue Lehrinhalte zu- und eingeordnet werden können. Inhalte, die nicht zuordenbar sind, werden nur schwer aufgenommen. Neben fehlender Zuordenbarkeit von Inhalten kann es auch dann zur erschwerten

¹Hermann Ebbinghaus (1850-1909)

Aufnahme kommen, wenn gleiche Inhalte in zeitlicher Überlagerung in verschiedenen Gegenständen zu lernen sind. Für die Praxis ist weiters von Bedeutung, dass, wenn es neue Lehrinhalte zu vermitteln gilt, dies unter Berücksichtigung mehrerer Aspekte stattfindet: Hierzu zählt beispielsweise der Einsatz von „bildhafter Vorstellung“ sowie die Einteilung des Stoffs in größere Einheiten, hierarchische Strukturen und Schemata.

2.2 Didaktik im Kontext der Informatik

Die Literatur bietet eine Vielzahl an Definitionsversuchen für „Didaktik“. Einer etymologischen Betrachtung nach findet der Begriff seinen Ursprung in „lehren“ oder „unterrichten“ [12]. Demnach kann die Didaktik als jener (wissenschaftliche) Fachbereich angesehen werden, der sich mit der Theorie des Unterrichts sowie des Lehrens beschäftigt. Hubwieser [14] führt in diesem Zusammenhang vier Theorien an.

Hierzu zählt der bildungstheoretische Ansatz, der im Wesentlichen die Allgemeinbildung verfolgt - es sollen nur Themen erfasst werden, die genereller Bedeutung für die Zukunft sind. Die Frage der Methodik rückt bei diesem Ansatz in den Hintergrund. Im Gegensatz zum bildungstheoretischen Ansatz inkludiert der lerntheoretische Aspekte der Methodik - die Frage nach dem „Wie?“ ist relevant. Daher wird dieser Ansatz auch als „Theorie des Lehrens und Lernens“ verstanden. Basierend auf der Informationstheorie von Shannon [15] wurde der informationstheoretisch-kybernetische Ansatz gebildet. Dieser verwendet einen Regelkreis als Grundlage und reduziert die Didaktik primär auf die Methodik. Der Auswahl des Lehrstoffs kommt hierbei eine untergeordnete Rolle zu. Der letzte Ansatz, die kommunikative Didaktik, sieht den Unterricht als kommunikativen und bildungsbezogenen Prozess, mit zwei zentralen Aspekten: der Inhalts- und der Beziehungsdimension.

Betrachtet man die Didaktik im Kontext der Informatik gilt es im ersten Schritt die Bezugswissenschaft - also die Informatik - zu definieren. Wie bereits bei der Didaktik gibt es auch bei der Informatik eine Fülle an Definitionen, die vor allem der jeweiligen Sichtweise sowie dem jeweiligen State of the Art geschuldet sind. Eine einfache und schülerorientierte Sichtweise liefert Humbert in [12], die lautet: „Informatik = Information + Automatik“. Also die Wissenschaft von der automatischen Verarbeitung von Information/Daten.

Betrachtet man die Geschichte der Informatik, findet man häufig eine Gliederung nach Generationen von Informatiksystemen. Als Grundlage für diese Gliederung dienen in der Regel die technischen Realisierungsvarianten - frei nach dem Motto „Vom Abakus bis zu den integrierten Schaltkreisen“. Auf Programmebene könnte die Gliederung von der Lochkarte bis hin zur objektorientierten Programmierung lauten. Diese Entwicklungsschritte brachten eine Fülle an Paradigmen hervor, die, wie im Falle der Objektorientierung, auch heute noch Gültigkeit haben, aber in der Regel eine geringe Halbwertszeit aufweisen [12].

Ein weiterer Punkt, den es in diesem Zusammenhang zu betrachten gilt, ist die Unterscheidung zwischen Konzept- und Produktwissen in der Informatik [16]. Konzeptwissen umfasst grundlegende Zusammenhänge und weist eine längerfristige Gültigkeitsdauer auf. Hingegen handelt es sich beim Produktwissen um Kenntnisse und Fähigkeiten, die sich auf die An- und Verwendung eines konkreten Produktes beziehen. In Anbetracht der Dynamik, welcher der Fachbereich Informatik unterliegt, erscheint Konzeptwissen ansich als überaus bedeutsam.

2.3 Berufsbildende Schulen in Österreich

Das österreichische Schulsystem bietet vor allem im Bereich der berufsbildenden Schulen ein breites fachliches Spektrum. Einen überaus bekannten und gefragten Vertreter des berufsbildenden Schulwesens stellen die technischen, gewerblichen und kunstgewerblichen Schulen - auch als Höhere Technische Lehranstalten (HTL) bekannt - dar. Laut der Internetplattform *HTL - Bildung mit Zukunft* [17] stehen 75 technische, gewerbliche und kunstgewerbliche Schulen, deren Bildungsangebot rund 15 Fachrichtungen - von Bautechnik bis Wirtschaftsingenieurwesen - umfasst, für SchülerInnen zur Verfügung. Weitere Vertreter dieses Schultyps sind laut dem Bundesministerium für Unterricht, Kunst und Kultur (BMUKK) [18] beispielsweise kaufmännische Schulen sowie Schulen für Mode und Bekleidungstechnik, Tourismus, Sozialberufe oder Kindergartenpädagogik.

Wie die Bezeichnung des jeweiligen Schultyps bereits zum Ausdruck bringt, liegt besonderes Augenmerk auf der beruflichen Erstausbildung, welche die SchülerInnen zur Ausübung eines ihrer Ausbildung entsprechenden Berufs befähigen soll. Obwohl die Berufsausbildung im Mittelpunkt steht, kommt die Allgemeinbildung - ob im Bereich der Naturwissenschaften oder der Sprachen - nicht zu kurz [18]. Somit erlangen die AbsolventenInnen ein Kompetenzprofil, das sie einerseits als zukünftige ArbeitnehmerInnen für die industrielle und gewerbliche Wirtschaft interessant macht, andererseits aber auch für den tertiären Bildungsbereich entsprechend vorbereitet. Nicht zuletzt deswegen fanden vor allem die höheren berufsbildenden Schulen großen Anklang bei AbsolventInnen der achten Pflichtschulstufe. Das spiegelt sich in der Entwicklung der Schülerzahl wider: So konnten beispielsweise die höheren berufsbildenden Schulen im Zeitraum 1990 bis 2010 einen Schülerzuwachs von rund 38 Prozent verzeichnen. Im Vergleich dazu nahm im selben Zeitraum die Schüleranzahl an der AHS-Oberstufe um rund 10 Prozent ab [18]. Da für die weitere Arbeit ausschließlich das technische, gewerbliche und kunstgewerbliche Schulwesen von Bedeutung ist, wird dieses nachfolgend im Detail betrachtet.

2.3.1 Technische, gewerbliche und kunstgewerbliche Schulen

Das Schulorganisationsgesetz [9] unterteilt die technischen, gewerblichen und kunstgewerblichen Schulen in berufsbildende mittlere und höhere Schulen. Den berufsbildenden mittleren Schulen sind

- die technischen, gewerblichen und kunstgewerblichen Fachschulen sowie
- etwaige Sonderformen der technischen, gewerblichen und kunstgewerblichen Fachschulen, wie zum Beispiel Meister-, Werkmeister- und Bauhandwerkerschulen,

zuzuordnen. Der Bereich der berufsbildenden höheren Schulen setzt sich wiederum aus

- den höheren technischen und gewerblichen (einschließlich kunstgewerblichen) Lehranstalten (HTL) sowie
- etwaigen Sonderformen der technischen, gewerblichen und kunstgewerblichen Lehranstalten, wie zum Beispiel Aufbaulehrgänge oder Kollegs,

zusammen. Das Qualitätsmanagement-Handbuch *HTL Q-Sys* [19] verwendet für den gesamten Bereich der berufsbildenden mittleren und höheren Schulen sowie deren jeweilige Sonderformen die Bezeichnung „technisches Schulwesen“. Um die Lesbarkeit dieser Arbeit zu optimieren, wird demzufolge ebenfalls diese Bezeichnung verwendet. Weiters, insofern nicht explizit darauf hingewiesen wird, bezieht sich die Abkürzung BMHS (Berufsbildende mittlere und höhere Schulen) ausschließlich auf die technischen Vertreter dieses Schultyps.

Berufsbildende mittlere Schulen

Bei den mittleren berufsbildenden Schulen - auch Fachschulen genannt -, die in der Regel drei oder vier Jahre (9. bis 11. oder 12. Schulstufe) in Anspruch nehmen, kommt der fachbezogenen Praxis eine große Bedeutung zu. Folglich umfasst die Ausbildung Pflichtpraktika im Ausmaß von vier Wochen. Darüber hinaus inkludiert die Ausbildung üblicherweise ein 12-wöchiges Betriebspraktikum, das im letzten Schuljahr zu absolvieren ist. Die Fachschule wird mit einer sogenannten Abschlussprüfung abgeschlossen und soll, wie bei der höheren Schule auch, die berufliche Qualifikation zum Ausdruck bringen. Mit dem Erwerb des Abschlusszeugnisses erhalten die SchülerInnen einen unmittelbaren Zugang zu bestimmten Berufen. FachschülerInnen können mit entsprechenden Zusatzprüfungen ebenfalls die Hochschulreife erlangen. Zur Gruppe der Fachschulen zählen laut BMUKK auch Sonderformen wie beispielsweise Meister- und Werkmeisterschulen oder Vorbereitungslehrgänge [18].

Berufsbildende höhere Schulen

Berufsbildende höhere Schulen dauern in der Regel fünf Jahre (9. bis 13. Schulstufe) und schließen mit einer sogenannten Reife- und Diplomprüfung ab. Teil dieser Reife- und Diplomprüfung kann eine Diplomarbeit sein, die ein Thema aus dem Fachbereich behandeln muss [18]. Die Ausbildung erfordert ebenfalls Pflichtpraktika, die es im Ausmaß von acht Wochen in einem facheinschlägigen Betrieb zu absolvieren gilt. Mit dem Abschluss erlangen die SchülerInnen einerseits die Hochschulreife, andererseits werden mit dem Diplom die beruflichen Qualifikationen im Fachgebiet bescheinigt, wodurch die Ausübung gehobener Berufe ermöglicht wird. AbsolventInnen einer höheren Schule wird nach erfolgreichem Abschluss und einer dreijährigen facheinschlägigen beruflichen Praxis die Standesbezeichnung „Ingenieur/Ingenieurin“ verliehen [18].

Allgemeine Bildungsziele und -inhalte

Wie einleitend bereits erwähnt und der Informationsbroschüre [18] für berufsbildende Schulen zu entnehmen ist, können beide Schulzweige - ob Fachschule oder höhere Schule - auf eine fundierte Allgemeinbildung verweisen. Ebenso ist es das Ziel, den SchülerInnen neben einer hochwertigen Methodenkompetenz eine entsprechende Sozialkompetenz zu vermitteln. Die geforderte Praxisnähe wird durch Übungen in Laboratorien, bei welchen das theoretische Wissen in die Praxis überführt und vertieft werden kann, sichergestellt. Somit erwerben die SchülerInnen Fähigkeiten, die sie in Kombination mit den Pflichtpraktika auf einen möglichst reibungslosen Einstieg in das Berufsleben vorbereiten sollen. Da heutzutage ein

betriebswirtschaftliches und/oder rechtliches Basiswissen nahezu unumgänglich ist, finden auch diese Inhalte Berücksichtigung in der Ausbildung.

2.3.2 Einordnung der berufsbildenden mittleren und höheren Schulen

Beschränkt man die Einordnung der BMHS auf die nationale Ebene, bedarf es keiner ausführlichen Erörterung: dieser Schultyp ist tief im österreichischen Bildungssystem verankert und hat große Tradition. Wie bereits festgehalten, haben AbsolventInnen einer höheren berufsbildenden Schule Zugang zum tertiären Bildungsbereich. Demnach entscheiden sich laut einer Erhebung des „Instituts für Bildungswissenschaften“ [20] rund die Hälfte der etwa 8.000 HTL-AbsolventInnen für ein Studium an einer Universität oder Fachhochschule. Diese Quote inkludiert auch jene, die parallel zum Studium eine Erwerbstätigkeit aufnehmen. Jene AbsolventInnen, die sich für den Eintritt in den Arbeitsmarkt entscheiden, stoßen bei der Arbeitgeberseite auf eine breite Akzeptanz [21, 22, 23]; was einerseits auf das hohe Ausbildungsniveau, andererseits aber auch auf die ausgeprägten Erfahrungswerte hinsichtlich der Fähigkeiten von AbsolventInnen zurückzuführen ist.

Erstreckt man die Einordnung auf internationale Ebene, was aufgrund der Globalisierung im Bildungswesen und der damit verbundenen Mobilität von AbsolventInnen zunehmend erforderlich ist, wird deutlich, dass vor allem die höheren berufsbildenden Schulen im internationalen Vergleich schwer einzuordnen sind. Wie das Institut für Bildungsforschung und Wirtschaft im Rahmen einer Studie [23] festgestellt hat, liegt dies einerseits an der hohen Diversität der Bildungssysteme sowie andererseits daran, dass die meisten Länder vergleichbare Ausbildungen im tertiären Bereich angesiedelt haben. Im Detail bedeutet das, dass der HTL-Abschluss dem Level 4A (siehe Tabelle 2.1) des sechsstufigen Klassifikationssystems ISCED (*International Standard Classification of Education*) [24] zuzuordnen ist. Hingegen werden in anderen Ländern (die im Vergleich zu Österreich eine höhere Akademikerquote aufweisen) vergleichbare Abschlüsse dem Level 5B (berufsorientierter Tertiärbereich) zugeordnet. Lediglich BHS-Kollegs, deren Besuch eine allgemeine Hochschulreife voraussetzt, oder Werkmeisterschulen finden im Level 5B Berücksichtigung, jedoch nicht die Tagesform einer HTL. Laut Schneeberger [25] liegt das im Wesentlichen an der mangelhaften Differenzierung im tertiären Bildungsbereich Österreichs, der keine Unterscheidung zwischen berufsorientierter (5B) und akademischer (5A) Hochschulausbildung erlaubt. Im Rahmen einer möglichen Neu-Strukturierung des österreichischen Schulwesens wäre somit eine höherwertige Einordnung des HTL-Abschlusses überaus angebracht.

Gegenwärtig liegt der Schluss nahe, dass HTL-AbsolventInnen im internationalen Vergleich unter ihrem Wert geschlagen werden. Abhilfe - zumindest auf Ebene der Europäischen Union - soll der so genannte *Europäische Qualifikationsrahmen* (EQR) [26] schaffen. Hierbei handelt es sich um ein Übersetzungsinstrument, das den Vergleich verschiedener Qualifikationssysteme ermöglichen soll. Der EQR definiert für Lernergebnisse die Kategorien „Kenntnisse“, „Fähigkeiten“ und „Kompetenz“ in acht Leistungsniveaus. Basierend auf dem EQR können die einzelnen Mitgliedsstaaten die nationalen Qualifikationen bewerten und in weiterer Folge in einem *Nationalen Qualifikationsrahmen* (NQR) [27] formulieren. Ziel ist es, dass zukünftig alle Staaten der Europäischen Union über einen NQR verfügen, wodurch die

Tabelle 2.1: Einordnung österreichischer Bildungsgänge gemäß den ISCED-Bildungslevels (Eigendarstellung nach [25])

Level	Beschreibung	Bildungsgang
0	Vorschulische Bildung	Kindergarten, Vorschulstufe
1	Primarschulbildung	Volksschule, Sonderschule
2	Sekundarbereich I	Hauptschule, AHS-Unterstufe
3 A	Sekundarbereich II	AHS-Oberstufe
3 B	Sekundarbereich II	Lehre (Duale Ausbildung), Mittlere berufsbildende Schulen (Fachschulen)
4 A	Postsekundärer, aber nicht-tertiärer Bereich	Höhere berufsbildende Schulen, Aufbaulehrgänge
4 B	Postsekundärer, aber nicht-tertiärer Bereich	Schulen für Gesundheits- und Krankenpflege
5 A	Tertiärbereich, erste Stufe, mind. 3-jährige Ausbildung	Bakkalaureats-, Diplom- o. Magisterstudium an einer FH o. Uni
5 B	Tertiärbereich, erste Stufe, mind. 2-jährige Ausbildung	Meister- u. Werkmeisterausbildung, Kollegs, Berufspädagogische Akademien
6	Tertiärbereich, zweite Stufe	Doktoratsstudium

Transparenz und somit die Vergleichbarkeit von Bildungsabschlüssen beziehungsweise von Bildungssystemen wesentlich vereinfacht werden soll.

Kapitel 3

Bildungsstandards und Kompetenzorientierung

Wie das Kapitel 2.3.2 (Einordnung der berufsbildenden mittleren und höheren Schulen) bereits aufzeigt, besteht die Notwendigkeit der Vergleichbarkeit von Bildungssystemen. Hierfür wurde mit dem Europäischen Qualifikationsrahmen (EQR) bereits ein Instrument entwickelt, das die gewünschte Transparenz auf Ebene der Europäischen Union (EU) schaffen soll. Parallel zu diesen Bemühungen arbeitet das Bundesministerium für Unterricht, Kunst und Kultur (BMUKK) seit etwa 2005 an der Entwicklung von Bildungsstandards im berufsbildenden Schulwesen [6]. In Anlehnung an den europäischen Qualitätsgedanken verfolgt das Projekt das Ziel, einerseits die Qualität der Ausbildungsgänge nachhaltig zu verbessern sowie andererseits die Vergleichbarkeit von Bildungsabschlüssen auf nationaler Ebene bestmöglich zu fördern. Mit der Einführung von Bildungsstandards und der Kompetenzorientierung kommt auf die Schulen ein Paradigmenwechsel zu, der Herausforderungen auf mehreren Ebenen mit sich bringt.

Der erste Teil dieses Kapitels widmet sich den Zielen und Aufgaben des Projektes „Bildungsstandards“. Darauf aufbauend werden der Grundgedanke der Kompetenzorientierung sowie das Kompetenzmodell der Höheren Technischen Lehranstalt (HTL) für Informationstechnologie (IT) und dessen Deskriptoren erörtert.

3.1 Ziele und Aufgaben

Für die Entwicklung der Bildungsstandards zeichnet sich die Sektion II (Berufsbildendes Schulwesen, Erwachsenenbildung und Schulsport) des BMUKK verantwortlich. Detaillierten Aufschluss über die Projektinhalte liefert das Projekthandbuch „Bildungsstandards in der Berufsbildung“ [28]. Diesem ist zu entnehmen, dass die Bezeichnung Bildungsstandard grundsätzlich als Überbegriff für Kompetenzmodelle, Deskriptoren und prototypische Unterrichtsbeispiele zu verstehen ist. Die grundlegenden Ziele der Bildungsstandards sowie die Aufgaben der einzelnen Bausteine werden nachfolgend im Detail erläutert.

Bildungsstandards sind als konkrete und vor allem verbindliche Zielvorstellungen für eine Fachrichtung, wie beispielsweise IT, zu verstehen. Wesentlich hierbei ist, dass seitens der Lernenden ein kumulierter Kompetenzerwerb über sämtliche Schulstufen eines Bildungsganges erfolgt, sodass am Ende der Ausbildung das definierte Kompetenzniveau im jeweiligen Kompetenzbereich erreicht wird. Das zu erreichende Kompetenzniveau wird mittels sogenannter Deskriptoren (für den jeweiligen Kompetenzbereich) festgelegt. Die Summe dieser Deskriptoren ergibt wiederum das zuvor erwähnte Kompetenzmodell.

Um die Dimension eines Deskriptors im Detail zu verdeutlichen, dienen prototypische Unterrichtsbeispiele. Diese wurden von den fachrichtungsspezifischen Arbeitsgruppen des BMUKK entwickelt und für die Fachrichtung IT in [29] formuliert. Ausgehend von den Deskriptoren, beziehungsweise von den prototypischen Unterrichtsbeispielen gilt es von den Lehrenden konkrete Unterrichtsbeispiele/-sequenzen zu entwickeln.

Als Basis für die Bildungsstandards dient der jeweilige Lehrplan mit seinen Bildungszielen, didaktischen Grundsätzen sowie Lehrstoffvorgaben [28]; wobei laut der Initiative IMST des BMUKK [30] Lehrplan und Bildungsstandards durchaus in Wechselwirkung zueinander stehen können - und zwar dann, wenn aufgrund des rasanten technologischen Fortschritts (im Bereich der IT) eine Adaptierung der Bildungsstandards erforderlich wird.

3.2 Kompetenz und Kompetenzorientierung

Nachdem die Kompetenzorientierung der zentrale Bestandteil der Bildungsstandards ist, erfolgt im ersten Schritt eine Definition des Begriffs „Kompetenz“. Die Projektgruppe des BMUKK greift hierfür auf Weinert [31] zurück, der Kompetenz wie folgt definiert:

Unter Kompetenzen versteht man die bei Individuen verfügbaren oder durch sie erlernbaren kognitiven Fähigkeiten, um bestimmte Probleme zu lösen sowie die damit verbundenen motivationalen, volitionalen und sozialen Bereitschaften und Fähigkeiten, um Problemlösungen in variablen Situationen erfolgreich und verantwortungsvoll nutzen zu können.

Wie diese Definition bereits zum Ausdruck bringt, handelt es sich beim Begriff Kompetenz um einen überaus vielschichtigen Begriff, der eine differenzierte Betrachtung und somit eine Unterteilung in mehrere unterschiedliche Komponenten erfordert. Laut Weinert zählen hierzu neben

- der Fach- und der Methodenkompetenz,
- die Sozialkompetenz sowie
- die personale Kompetenz.

Die Fachkompetenz besteht im Wesentlichen aus speziellem Wissen über einen Fachbereich (z.B. dessen Fakten, Theorien und Konzepte). Die Methodenkompetenz gibt darüber Auskunft, inwieweit SchülerInnen imstande sind, Anwendungsmöglichkeiten für erworbene Kompetenzen zu erkennen und diese in weiterer Folge auch anzuwenden. Hierzu zählt beispielsweise die Fähigkeit, neues Wissen selbstständig zu erwerben. Im Projekthandbuch [28] werden die Fach- und Methodenkompetenz als kognitive Fähigkeiten zusammengefasst.

Fähigkeiten, die vor allem die zwischenmenschliche oder persönliche Ebene betreffen (z.B. Kommunikation und Interaktion mit Mitmenschen oder die Fähigkeit, eigene Wissensdefizite zu erkennen), werden durch die Sozial- und Personalkompetenz zum Ausdruck gebracht [28].

Zusammenfassend kann festgehalten werden, dass mit der Kompetenzorientierung eine Richtung eingeschlagen wird, die weg von der reinen Wissensvermittlung hin zur Vermittlung von Fähigkeiten führt und ein selbstständiges Handeln und Tun im Kontext der Problemlösung ermöglichen sollen. Diese Neuausrichtung bringt, wie einleitend bereits erwähnt, mehrere Herausforderungen mit sich. Hierzu zählen beispielsweise neben der Unterrichtsdurchführung die Leistungsfeststellung und -beurteilung, da eine einfache Wissensabprüfung den Anforderungen der Kompetenzorientierung nicht mehr gerecht wird. Es bedarf hierbei einer Vorgehensweise, die die Vielschichtigkeit einer „Kompetenz“ entsprechend berücksichtigt. Weiterführende Inhalte sind dem Kapitel 4, Kompetenzorientiertes Unterrichten, zu entnehmen.

3.3 Kompetenzmodell

Das Kompetenzmodell¹ definiert die zu erwartenden Lernergebnisse, die es vonseiten der SchülerInnen zu erbringen gilt. Wie Abbildung 3.1 verdeutlicht, besteht ein Kompetenzmodell aus zwei Dimensionen: der Inhalts- und der Handlungsdimension. Die Inhaltsdimension definiert jene Inhalte, die es im Fachbereich IT zu vermitteln gilt. Hierzu zählen beispielsweise „Softwareentwicklung“, „Informationstechnische Projekte“ oder „Systemtechnik“, die als Kompetenzfelder bezeichnet werden. Jedes dieser sechs Kompetenzfelder unterteilt sich wiederum in mehrere Kompetenzbereiche. Der für diese Arbeit relevante Kompetenzbereich „Industrielle Informationstechnik“ ist im Kompetenzfeld Systemtechnik (siehe Abbildung 3.1, 4) enthalten. Im Detail setzt sich das Kompetenzfeld Systemtechnik aus folgenden Kompetenzbereichen zusammen:

- Elektrotechnik und Elektronik für IT
- Grundlagen der Informatik
- Betriebssysteme
- Industrielle Informationstechnik
- Systemintegration und Infrastruktur
- Dezentrale Systeme

Bei den Kompetenzfeldern Systemtechnik, Medientechnik und Netzwerktechnik (siehe Abbildung 3.1, graue Markierung) handelt es sich um HTL-spezifische Spezialisierungen, die je nach gewähltem Ausbildungsschwerpunkt schulautonom Berücksichtigung finden. Diese ergänzen die fachtheoretische und -praktische Basisausbildung, die unabhängig vom gewählten

¹Da diese Arbeit die Entwicklung didaktischer Konzepte für die Fachrichtung IT zum Ziel hat, erfolgt die Erläuterung des Kompetenzmodells am Beispiel der Fachrichtung IT [29].

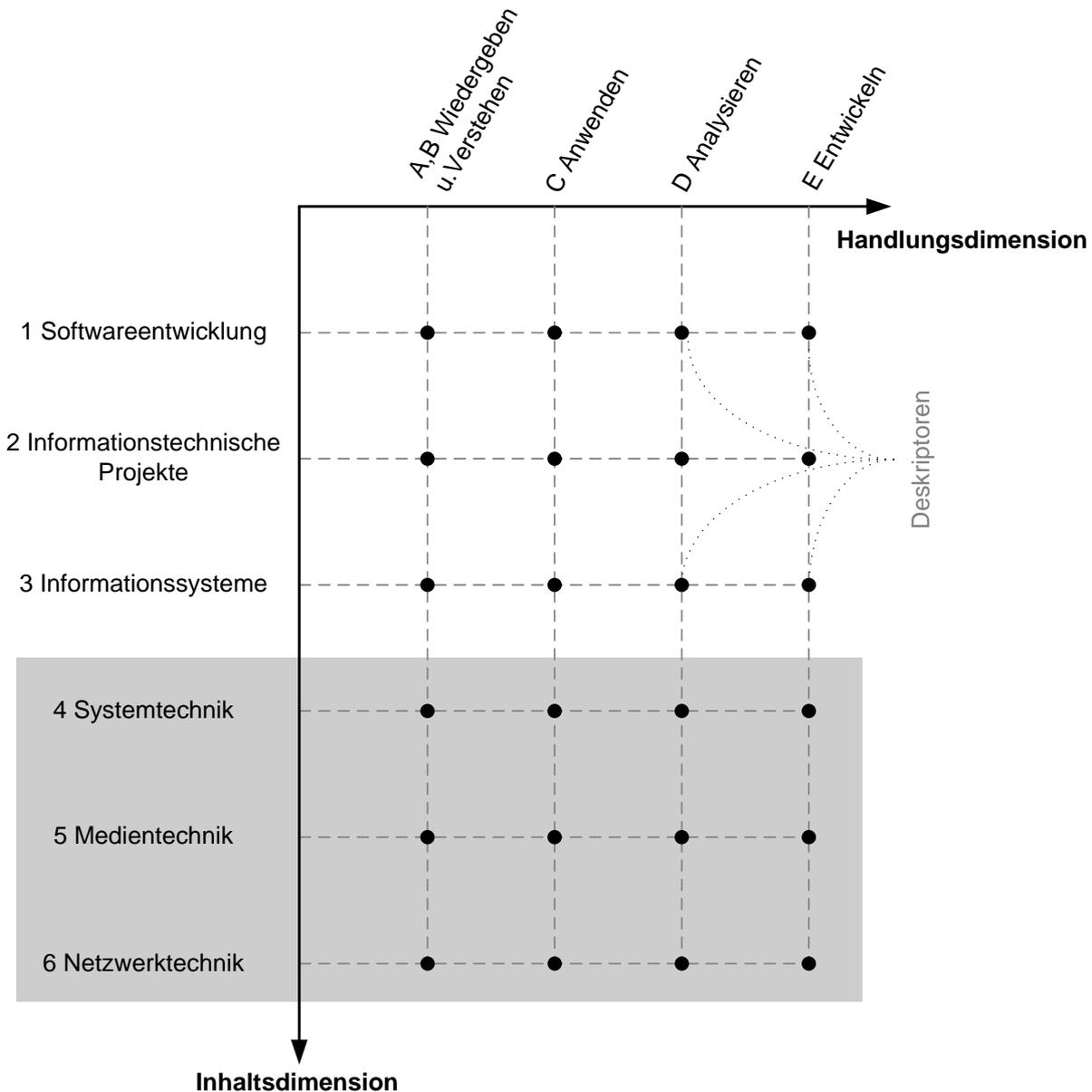


Abbildung 3.1: Kompetenzmodell der HTL-Fachrichtung Informationstechnologie (Eigendarstellung nach [29])

Ausbildungsschwerpunkt an jeder IT-HTL einheitlich ist. Eine detaillierte Darstellung aller Kompetenzfelder ist im Anhang, Abbildung A.1, enthalten.

Die Handlungsdimension weist die zu erbringende kognitive Leistung auf (vgl. Kapitel 3.2, Kompetenz und Kompetenzorientierung). Grundlage hierfür bilden die theoretischen Arbeiten von Benjamin S. Bloom [32] im Bereich der Lerntheorie. Bloom entwickelte ein Klassifikationsschema, mit welchem Lernziele und somit der kognitive Prozess eingeordnet werden können. Hierfür hat Bloom mit „Wissen“, „Verstehen“, „Anwenden“, „Analyse“, „Synthese“ und „Beurteilung“ sechs Kategorien in aufsteigender Reihenfolge festgelegt. Die fachrichtungsspezifische Arbeitsgruppe des BMUKK hat dieses Modell in überarbeiteter Form übernommen, indem beispielsweise die ersten beiden Kategorien (Wiedergeben und Verstehen) zusammengefasst wurden [33]. Nachfolgend werden die in Abbildung 3.1 angeführten Handlungsebenen inhaltlich kurz vorgestellt. Grundlage hierfür bilden einerseits das Pro-

jekthandbuch „Bildungsstandards in der Berufsbildung“ [28] und die aktuelle Version der Fachrichtungsstandards „Informationstechnologie“ [29] sowie andererseits Reinbacher in [34].

Wiedergeben und Verstehen: Wie die Bezeichnung dieser Handlungsebene bereits zum Ausdruck bringt, soll die Kompetenz insofern entwickelt sein, dass grundlegendes Fachwissen im Bereich der Informationstechnologie verstanden und mit entsprechendem Fachvokabular wiedergegeben werden kann. Ebenfalls sollen Zusammenhänge erkannt und Gelerntes auf einen entsprechenden Sachverhalt übertragen werden können.

Anwenden: Die Kategorie Anwenden erfordert das Vorhandensein grundlegender Wissensstrukturen (aus der vorhergehenden Kategorie), die in Zusammenhang mit neuem Wissen das Abarbeiten vorgegebener Arbeitsschritte ermöglicht. Um die Problemlösung zu bewältigen, muss ein eigenständiger Wissenstransfer erfolgen.

Analysieren: Die Handlungsebene Analysieren erfordert die Fähigkeit, vorhandenes Wissen neu zu strukturieren sowie gegebenenfalls eigene Kriterien daraus zu entwickeln.

Entwickeln: Entwickeln stellt die letzte und somit anspruchsvollste Stufe der vier Handlungsebenen dar. Hierbei muss die Kompetenz soweit ausgeprägt sein, dass SchülerInnen imstande sind, eigenständig etwas Neues zu entwickeln.

Ergänzend gilt es festzuhalten, dass Kompetenzmodelle im Allgemeinen eine personale und soziale Dimension enthalten (vgl. Kapitel 3.2), die jedoch im Rahmen dieser Arbeit nicht näher erläutert werden. Weiterführende Informationen sind in [28], [29] sowie [33] enthalten.

3.4 Deskriptoren

Die Schnittpunkte (siehe Abbildung 3.1, farblich hinterlegte Kreise) der jeweiligen Inhalts- und Handlungsdimension werden als Deskriptoren bezeichnet. Die Aufgabe eines Deskriptors liegt darin, für jeden dieser Schnittpunkte die Anforderung in ergebnisorientierter Form zu definieren, die es vonseiten der SchülerInnen zu erbringen gilt. Um einen Deskriptor einem dezidierten Kompetenzfeld zuordnen zu können, entwickelte das BMUKK in [33] eine entsprechende Codierungs-Syntax, die sich wie folgt gestaltet:

<Fachrichtung><Schwerpunkt> - <Inhalt> . <Nummerierung> - <Handlung>

Für die Informationstechnologie (IT) lautet der Fachrichtungscode in jedem Fall „IT“. Wie in Kapitel 5.1 ausführlich erläutert, sieht der IT-Lehrplan mit Systemtechnik, Medientechnik und Netzwerktechnik drei Ausbildungsschwerpunkte vor, die schulautonom festgelegt werden. Demnach sieht die Codierungsvorschrift den Buchstaben „S“ für Deskriptoren des Schwerpunkts Systemtechnik, „M“ für Deskriptoren des Schwerpunkts Medientechnik und „N“ für Deskriptoren des Schwerpunkts Netzwerktechnik vor, der dem Fachrichtungscode angefügt wird. Bezieht sich ein Deskriptor auf die Basisausbildung, erfolgt keine Angabe im Schwerpunktfeld. Das Feld „Inhalt“ gibt Auskunft über das Kompetenzfeld, welchem ein Deskriptor zuzuordnen ist. Folgende Inhalte sind möglich: Netzwerktechnik = 1, Medientechnik = 2, Systemtechnik = 3, Informationssysteme = 4 und Softwareentwicklung =

5 (vgl. Abbildung 3.1, Inhaltsdimension). Das Feld „Nummerierung“ dient zur fortlaufenden Nummerierung der Deskriptoren. Ergänzend zum Feld „Schwerpunkt“ gibt dieses Auskunft, ob der Deskriptor der Basisausbildung oder einem Schwerpunkt zuzuordnen ist. Weist ein Deskriptor eine Nummer zwischen 1 und 100 auf, handelt es sich um einen Deskriptor der Basisausbildung beziehungsweise zwischen 101 und 199 um einen Schwerpunktthemen-Deskriptor. Im letzten Feld „Handlung“ ist eine Kategorie der Handlungsdimension angegeben (vgl. Abbildung 3.1); wobei der Buchstabe „A“ der Handlungsebene Wiedergeben und der Buchstabe „E“ der Handlungsebene Entwickeln entspricht¹. Um die Anwendung der vorgestellten Codierungsvorschrift zu demonstrieren, werden auszugsweise einige Deskriptoren aus dem bestehenden Entwurf [33] vorgestellt.

- Bsp. 1:** IT-5.1-B Ich kann die wichtigsten skalaren Datentypen einer höheren Programmiersprache erklären.
- Bsp. 2:** IT-4.13-C Ich kann Daten in Tabellen einfügen, Daten verändern und Daten löschen.
- Bsp. 3:** IT-3.23-D Ich kann Betriebssysteme beurteilen und für bestimmte Zwecke das geeignete auswählen.

Gemäß der Codierungsvorschrift kann der Beispiel-Deskriptor 1 dem Kompetenzfeld Softwareentwicklung zugeordnet werden. Von den SchülerInnen wird verlangt, dass sie die geforderten Inhalte verstehen und wiedergeben können. Der Deskriptor weist die laufende Nummer 1 auf. Das Beispiel 2 mit der laufenden Nummer 13 erfordert die Kompetenz Anwenden aus dem Kompetenzfeld Informationssysteme. Dem Kompetenzfeld Systemtechnik kann Beispiel 3 zugewiesen werden. Das verlangte Kompetenzniveau ist Analysieren. Die laufende Nummer lautet 23.

Insgesamt betrachtet erscheint die gegenwärtige Situation so, dass die wesentlichen Schritte bei der Entwicklung der IT-Bildungsstandards vollzogen wurden, Details jedoch noch ausstehen. Ein Indikator hierfür ist, dass es sich bei den in [33] formulierten Deskriptoren für die Basisausbildung um Entwürfe handelt. Inwieweit die Diskrepanz, dass das Kompetenzmodell in [33] fünf jedoch das in [29] sechs Inhaltsdimensionen aufweist, ebenfalls auf den Dokumentenstatus „Entwurf“ zurückzuführen ist, wird die Zukunft zeigen. Überhaupt erscheint es als zielführend, die Dokumente [29] und [33] zu synthetisieren. Für eine erfolgreiche Umsetzung der Bildungsstandards ist die Vervollständigung der in [29] angeführten prototypischen Unterrichtsbeispiele ebenfalls von großer Bedeutung. Dass bei den Fachrichtungsstandards in [29] die in [33] entwickelte und in dieser Arbeit vorgestellte Codierungssyntax nicht angewendet wurde, erscheint im Gegensatz dazu als unproblematisch.

Wie auch Reinbacher in seiner Arbeit [34] festhält, wird sich zeigen, ob mit der Kompetenzorientierung der entscheidende Schritt zu einer Qualitätssteigerung gelingt, oder ausschließlich der Vergleich von Bildungsgängen erleichtert wird. Voraussetzung hierfür ist in jedem Fall ein für die Fachrichtung IT durchgängiges Kompetenzmodell sowie Deskriptoren, die dem State of the Art der IT gerecht werden.

¹Da die ersten beiden Handlungsebenen zusammengefasst wurden, beginnt die Bezeichnung der Deskriptoren mit dem Buchstaben B (siehe Kapitel 3.3).

Kapitel 4

Unterricht im Kontext der Kompetenzorientierung

Bildungsstandards definieren die zu erzielenden Lernergebnisse in kompetenzorientierter Form (vgl. Kapitel 3). Damit Lernende die geforderten Kompetenzen möglichst effektiv erwerben können, sind entsprechende Unterrichtsmethoden erforderlich, die über die Vermittlung von Faktenwissen im Rahmen eines Frontalunterrichts hinausgehen. Im Grundlagenpapier „Kompetenzorientiertes Unterrichten“ [35] des Bundesministeriums für Unterricht, Kunst und Kultur (BMUKK) werden mit

- der Unterrichtsplanung und
- der Unterrichtsdurchführung sowie
- der Leistungsfeststellung und -beurteilung

die Hauptbausteine für einen erfolgreichen kompetenzorientierten Unterricht identifiziert. An dieser Stelle ist jedoch festzuhalten, dass diese drei Handlungsbereiche an sich nichts Neues darstellen, denn eine adäquate Unterrichtsplanung und -durchführung ist auch beim zuvor erwähnten Frontalunterricht unabdingbar. Der kompetenzorientierte Unterricht verlangt jedoch eine Neugestaltung der zugrundeliegenden Prinzipien oder Methoden.

Die nachfolgende Darstellung behandelt das Thema Unterricht sowohl im Allgemein als auch im Kontext der Kompetenzorientierung. Hierbei erfolgt einleitend die Definition von Unterricht. Darauf aufbauend werden die Merkmale des kompetenzorientierten Unterrichts dargelegt und die zuvor angeführten Hauptbausteine (in Bezug auf die Kompetenzorientierung) behandelt.

4.1 Definition von Unterricht

Hinsichtlich der Definition des Begriffs Unterricht bietet die Fachliteratur eine Reihe von Möglichkeiten. Hubwieser definiert Unterricht in [14] wie folgt:

Unterricht ist ein hochkomplexer Prozess mit zahlreichen, auch zyklischen Wechselwirkungen, bei dem Lehrende und Lernende unter gewissen gesellschaftlichen, bürokratischen und materiellen Vorgaben und Rahmenbedingungen im Hinblick auf eine bestimmte Zielsetzung interagieren. Langfristig hat dieser Prozess wiederum Auswirkungen auf die gesamte Gesellschaft und die von ihr formulierten Vorgaben.

Eine vereinfachte Form der Definition stellt Schröder in [10] bereit:

Unterricht ist organisierte Interaktion von Lehren und Lernen.

Wie den vorgestellten Definitionen zu entnehmen ist, kommt der Interaktion von Lehrenden und Lernenden eine große Bedeutung zu. Die von Hubwieser angeführte Komplexität lässt sich beispielsweise auf die große Anzahl der beteiligten Akteure mit unterschiedlichen Bedürfnissen, oder aber auch auf das „hierarchische Gefälle im SchülerIn-LehrerIn-Verhältnis“ und den damit einhergehenden Herausforderungen zurückführen. Infolgedessen erfordert das Unterrichten eine adäquate Vorgehensweise und Methodik, sodass dieser soziale Interaktionsprozess möglichst zielführend gestaltet werden kann.

4.2 Merkmale eines kompetenzorientierten Unterrichts

Nachdem die Kompetenzorientierung die Entwicklung von Handlungskompetenzen als zentrales Ziel verfolgt, werden in diesem Abschnitt konkrete Maßnahmen zur Zielerreichung vorgestellt. Aufgrund der sich bietenden Vielfalt erfolgt die Darstellung auszugsweise und stellt keinen Anspruch auf Vollständigkeit dar.

Ein grundlegendes Merkmal des kompetenzorientierten Unterrichts ist die Lernendenzentrierung. Hierbei stehen die Lernenden im Mittelpunkt und nicht der Gegenstand (vgl. Gegenstandszentrierung¹). Im Detail bedeutet das, Lernende werden von ihrer passiven Rolle als InformationsempfängerInnen, wie es beispielsweise beim klassischen Frontalunterricht der Fall ist, in eine aktive Rolle des Handelns und Tuns versetzt. Nur so kann der Aufbau der in Kapitel 3.2 geforderten Fach- und Methoden- aber auch Sozialkompetenz erfolgen [35].

In Bezug auf die Praxis lässt sich diese Vorgabe durch die Idee des selbstgesteuerten Lernens umsetzen. Hierbei tritt die Lehrkraft in den Hintergrund und übernimmt die Rolle des Coaches. Die Lernenden hingegen sind angehalten, durch eigenständiges Tun und Handeln die jeweiligen Kompetenzen zu entwickeln. Die Lehrkraft ist dahingehend gefordert, dass die erforderlichen Rahmenbedingungen erfüllt sind. Hierzu zählen beispielsweise brauchbare Arbeitsunterlagen sowie eine ansprechende Arbeitsumgebung. Sind nicht zuletzt diese Voraussetzungen gegeben, kann sich ein möglichst hoher Grad an Eigenmotivation bei den Lernenden entwickeln. In diesem Zusammenhang gilt es außerdem zu erwähnen, dass die Motivation bei den Lernenden aufgrund ihrer persönlichen Neugierde - also intrinsisch -, zumal sich eine Aufgabenstellung als überaus spannend und herausfordernd anmutet, entsteht und

¹Die Gegenstandszentrierung charakterisiert sich im Wesentlichen dahingehend, dass Lehrende die Inhalte eines Gegenstandes didaktisch aufbereiten und über den Weg der direkten Instruktion (z.B. Frontalunterricht) an die Lernenden weiterreichen [35].

nicht von den Lehrenden beispielsweise verbal eingefordert wird. Um das zu bewerkstelligen, empfiehlt sich zum Beispiel das sogenannte „Problem-Based Learning“ (PBL). Hierbei findet die Aufgabenstellung ihren Ursprung in realen Situationen oder Problemen des täglichen Lebens, die es durch das Bearbeiten der Aufgabenstellung zu lösen gilt. Diese Konfrontation weckt die Aufmerksamkeit der Lernenden. Damit die Motivation aufrechterhalten bleibt, ist es wichtig, dass der Schwierigkeitsgrad der Aufgabenstellung mit den Fähigkeiten der Lernenden übereinstimmt. Andernfalls käme es zu motivationalen Störungen. Eine präventive Gegenmaßnahme, um mögliche motivationale Störungen zu vermeiden, stellt die Differenzierung im Schwierigkeitsgrad der Aufgabenstellung dar [10]. Potenzielle Unterrichtsverfahren und -methoden reichen von Gruppenpuzzle über Fallstudien, Projektarbeiten oder sogenannten Übungsfirmlen bis hin zum „Lernen durch Lehren“. Hierbei bereiten Lernende Lehrinhalte didaktisch auf und präsentieren diese. Eine ausführliche Darstellung der zuvor angeführten Unterrichtsverfahren und -methoden erfolgt in Kapitel 4.4.

Der Gedanke, der zuvor erwähnten inhaltlichen Differenzierung, findet sich bei der Individualisierung wieder. Mit der Individualisierung wird der Heterogenität einer Klasse Rechnung getragen. Lernende unterscheiden sich beispielsweise hinsichtlich ihrer Interessen oder Begabungen und somit auch bei der Lerngeschwindigkeit. Die Individualisierung verfolgt das Ziel, auf die persönlichen Bedürfnisse bestmöglich einzugehen. Eine konkrete Maßnahme sind beispielsweise unterschiedliche Lernziele mit entsprechenden Übungs- und Hausaufgaben. Somit können Lernende die Inhalte entsprechend ihrer Lerngeschwindigkeit aufnehmen und verarbeiten. Was jedoch nicht bedeuten soll, dass unterschiedliche Leistungen mit gleichem Maßstab beurteilt werden - diese Maßnahme erfordert eine transparente, differenzierte Beurteilung. Darüber hinaus ist das Erreichen einer entsprechenden Grundkompetenz für einen positiven Abschluss erforderlich [35].

Im Gegensatz zur Vermittlung von reinem Faktenwissen kommt der Interdisziplinarität bei der Kompetenzorientierung eine zentrale Bedeutung zu: nachhaltiger Kompetenzaufbau erfolgt vor allem auch durch das gegenstandsübergreifende Vernetzen von Inhalten. Dies erfordert eine inhaltliche und zeitliche Abstimmung mit Lehrenden relevanter Gegenstände beziehungsweise Kompetenzbereiche innerhalb eines Ausbildungsgangs. Im Grundlagenpapier „Kompetenzorientiertes Unterrichten“ [35] werden außerdem noch weitere Kriterien angeführt, die einem erfolgreichen kompetenzorientierten Unterricht zugrunde liegen. Hierzu zählen:

- Strukturierter Unterricht mit eindeutigen Lernzielen
- Methodenvielfalt: Lern- und Arbeitsformen variabel gestalten
- Beurteilungsfreien Raum für ein variationsreiches Üben schaffen
- Erfahrung von Kompetenzzuwachs ermöglichen
- Motivierendes Umfeld schaffen (Interesse am Lernfortschritt zeigen etc.)
- Kritikfähigkeit entwickeln
- Zeit zum Lernen lassen

Nachdem die wesentlichen Merkmale des kompetenzorientierten Unterrichts aus der Sicht der Lernenden behandelt wurden, lässt sich konstatieren, dass kompetenzorientiertes Lehren aufgrund seiner Komplexität einerseits die entsprechende Bereitschaft, andererseits aber auch das entsprechende Know-how der Lehrenden voraussetzt. Das reicht von der grundsätzlichen Akzeptanz bis zur erforderlichen kognitiven Basiskompetenz, auf welcher der Unterricht aufbaut. Vor allem die Akzeptanz kann als zentrales Erfolgskriterium angesehen werden, denn ohne entsprechender Motivation und der Bereitschaft, neue Wege zu beschreiten, ist die Etablierung der Kompetenzorientierung kaum zu bewerkstelligen.

4.3 Unterrichtsplanung

Erfolgreicher Unterricht erfordert eine entsprechende Planung. Humbert [12] identifiziert für diesen Planungsprozess vier zentrale Bereiche, die im Zuge der Unterrichtsplanung zu berücksichtigen sind (siehe Abbildung 4.1). Die Substanz der Bereiche „Ziele“ und „Inhalte“

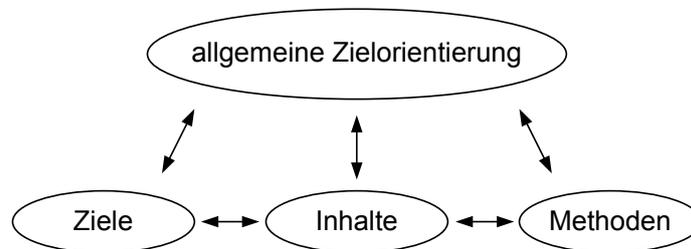


Abbildung 4.1: Bestandteile der Unterrichtsplanung und deren Zusammenwirken [12]

lässt sich mit der Frage „Was soll ich unterrichten?“ erarbeiten. Hinsichtlich der „Methoden“ gilt es die Frage „Wie soll ich unterrichten?“ zu beantworten. Die Rahmenbedingungen, die hierbei zu berücksichtigen sind, können der „allgemeinen Zielsetzung“ zugeordnet werden.

4.3.1 Bestandteile der Unterrichtsplanung

Hinsichtlich der Ziele und Inhalte geben der Lehrplan und die Bildungsstandards die Richtung vor. In der Verantwortung des Lehrenden liegt deren Konkretisierung; was vor allem die Detailauswahl der Lehrinhalte betrifft. Je nach Dynamik des Fachgebietes sind aktuelle Entwicklungen zu berücksichtigen, deren Gegenwarts- und Zukunftsbedeutungen zu analysieren und gegebenenfalls in den Unterricht zu integrieren. Die Frage der Unterrichtsmethode stellt wohl - vor allem im Kontext der Kompetenzorientierung - die größte Herausforderung dar. Die in Kapitel 4.2 angeführte und im Grundlagenpapier der Kompetenzorientierung geforderte Methodenvielfalt setzt umfangreiche Kenntnisse im Bereich der Unterrichtsmethodik bei den Lehrenden voraus. Unterrichtsmethoden, wie zum Beispiel Gruppenpuzzle, erfordern mitunter ein bestimmtes Zeitbudget, dem die übliche 50-Minuten-Taktung unter Umständen nicht gerecht wird [36]. Ebenfalls zu berücksichtigen ist, dass handlungsorientierte Unterrichtsmethoden gegebenenfalls bestimmte Unterrichtsmedien oder ein entsprechendes Raum-Setting erfordern.

Um den Gedanken des interdisziplinären Kompetenzerwerbs in die Praxis umzusetzen, ist - je nach Grad der gewünschten Interdisziplinarität - die Kenntnis der Lehrinhalte anderer

Kompetenzbereiche unumgänglich. Vorausgesetzt, dass diese Bedingung erfüllt ist, können die Lehrinhalte thematisch als auch zeitlich im Kreise der Lehrenden abgestimmt und bei der Planung des Unterrichts berücksichtigt werden.

Mit der Individualisierung soll auf die individuellen Bedürfnisse von Lernenden eingegangen werden (vgl. Kapitel 4.2). Dass diese Maßnahme in jedem Fall eine große Herausforderung darstellt, ist naheliegend. Vor allem wenn man die durchschnittliche Klassengröße einer Berufsbildenden Höheren Schule (BHS) in Betracht zieht, welche bei 25 SchülerInnen liegt [37]. Unter diesen Voraussetzungen ist zu klären, auf welcher Ebene die Individualisierung grundsätzlich erfolgen soll: auf (Klein-)Gruppenebene oder Schülerbene. Je nach Individualisierungsgrad gilt es die Planung des Unterrichts zu gestalten. In diesem Zusammenhang erscheinen auch die Merkmale „Erfahrung von Kompetenzzuwachs“ und „Zeit zum Lernen lassen“ (vgl. Kapitel 4.2) von Bedeutung, da in beiden Fällen eine Verallgemeinerung nicht zielführend wäre. Daher sind sie bei der Planung von individualisierungsspezifischen Maßnahmen ebenfalls entsprechend zu berücksichtigen.

4.3.2 Vorgehensweise bei der Unterrichtsplanung

Um den Ablauf der Unterrichtsplanung zu strukturieren, empfiehlt sich der Einsatz entsprechender Vorgehensmodelle. Ein mögliches Modell stellt das in Abbildung 4.2 illustrierte Modell nach Hartmann [16] dar. Obwohl in dieser Planungsgrundlage bis auf den Punkt „Infrastruktur, Hilfsmittel - womit?“ keine besonderen fachdidaktischen Elemente berücksichtigt werden, ist diese aufgrund des pragmatischen Ansatzes vor allem für Neulehrer geeignet [12]. Wie der Planungsgrundlage zu entnehmen ist, lässt sich der Planungsprozess in vier Phasen einteilen (siehe Markierungen 1 bis 4). In Bezug auf die zuvor hervorgehobenen Merkmale des kompetenzorientierten Unterrichts kommt vor allem den Phasen 2 und 3 eine gewisse Bedeutung zu, da zum Einen die eingesetzten Unterrichtsmethoden und -techniken sowie zum Anderen die Unterrichtsdurchführung zu planen sind. Und wie Kapitel 4.2 weiters zeigt, bringt die Kompetenzorientierung vor allem in diesen Bereichen große Veränderungen mit sich.

Um die Inhalte der Phase 1 zu konkretisieren, sollte ein so genannter Diagnosecheck Bestandteil der Unterrichtsplanung sein. Mit diesem lässt sich der Kompetenz-Ist-Stand zu Beginn des gemeinsamen Unterrichts erfassen. Je nach Ergebnis dieses Checks kann gegebenenfalls eine (geringfügige) Adaptierung der zu erzielenden Lernergebnisse erfolgen. Darüber hinaus lassen sich erste Aussagen über das Leistungsvermögen einzelner SchülerInnen treffen, wodurch die Grundlage zur Individualisierung gebildet wird.

Den Abschluss der Unterrichtsplanung sollte eine Jahresplanung mit Meilensteinen bilden, die überblicksartig die Inhalte sowie Grobziele des Kompetenzbereichs aufschlüsselt und in zeitlicher Abfolge darstellt.

4.4 Unterrichtsdurchführung

Die Unterrichtsdurchführung war und ist wohl jener Bereich, der die größten Herausforderungen für alle Beteiligten - ob Lehrende oder Lernende - mit sich bringt. Nicht zuletzt

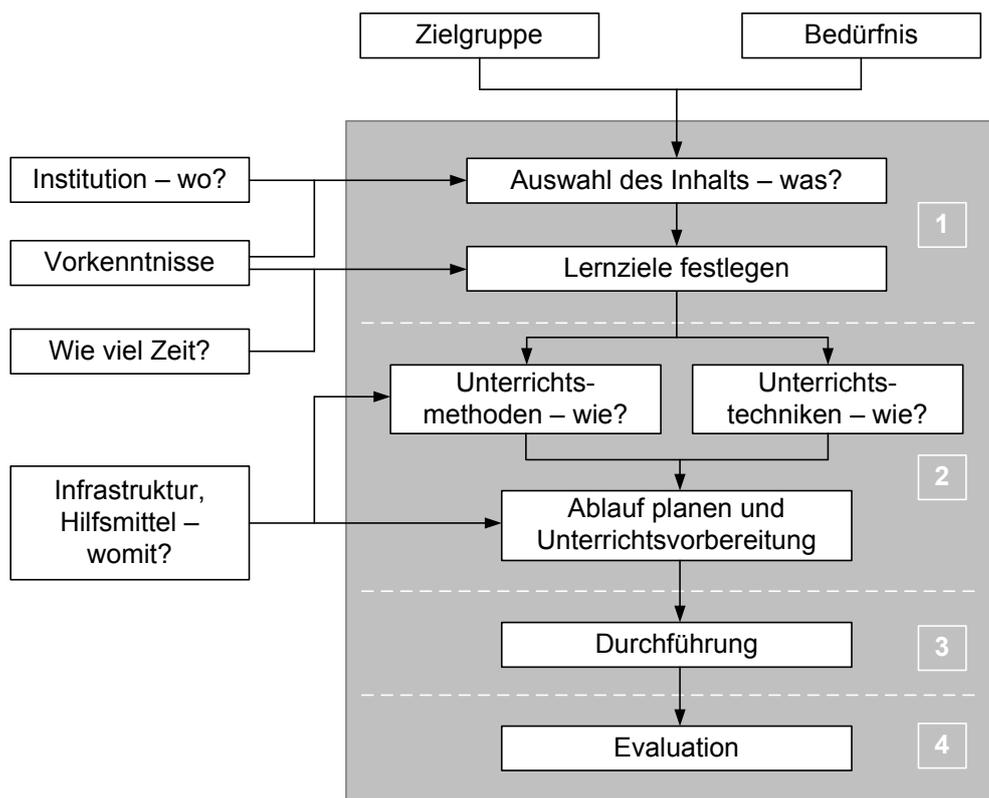


Abbildung 4.2: Unterrichtsplanung nach Hartmann [16]

aufgrund des bevorstehenden Paradigmenwechsels, der diesen Umstand eher noch verstärkt. Erfordert doch die gewünschte Methodenvielfalt neben entsprechender Methodenkompetenz bei den Lehrenden auch Kompetenzen bei den Lernenden. Dies deshalb, da nahezu alle handlungsorientierten Methoden ein hohes Maß an Kollaboration und Interaktion mit anderen Lernenden erfordern. Folglich ist ein Mindestmaß an sozialer Kompetenz bei den Lernenden unerlässlich. Nachfolgend werden einige Unterrichtsmethoden im Detail vorgestellt. Im Fokus stehen neben den klassischen handlungsorientierten Methoden, wie Einzel- oder Gruppenarbeit, auch jene, die eine spezifische Form der Handlungsorientierung aufweisen. Dass der Frontalunterricht nicht unbedingt zu diesen Methoden zählt und somit in diesem Zusammenhang keine bedeutende Position einnimmt, wird in den vorangegangenen Kapiteln bereits mehrmals zum Ausdruck gebracht. Einleitend soll daher der Frage, warum der Frontalunterricht den Anforderungen der Kompetenzorientierung nicht gerecht wird, nachgegangen werden.

Realität ist, dass der Frontalunterricht die beherrschende Unterrichtsmethode in Österreichs Klassenzimmern darstellt. Wie Abbildung 4.3 illustriert, nimmt die Lehrkraft bei dieser Unterrichtsform eine zentrale Rolle ein, indem die Lernenden (in der Regel reihenweise) der Lehrkraft gegenüber sitzen. Als Unterrichtsmedien werden beispielsweise Sprache, Tafel oder Folien verwendet. Die Lehrkraft lenkt und steuert den Lernprozess. Die Lernenden finden sich in der Rolle des Zuhörers wieder. Deren primäre Aufgabe ist es, den Ausführungen der Lehrkraft zu folgen. Diese Konstellation führt zwangsläufig zu einer Lehrkraftzentrierung. Aufgrund der ausgeprägten Lehrkraftzentrierung kann diese den Unterricht überaus gut planen. Die Wahrscheinlichkeit, dass unvorhersehbare Ereignisse eintreten, ist äußerst gering. Das führt wiederum dazu, dass die Lehrkraft ein Gefühl der Sicherheit verspürt. Unter diesen

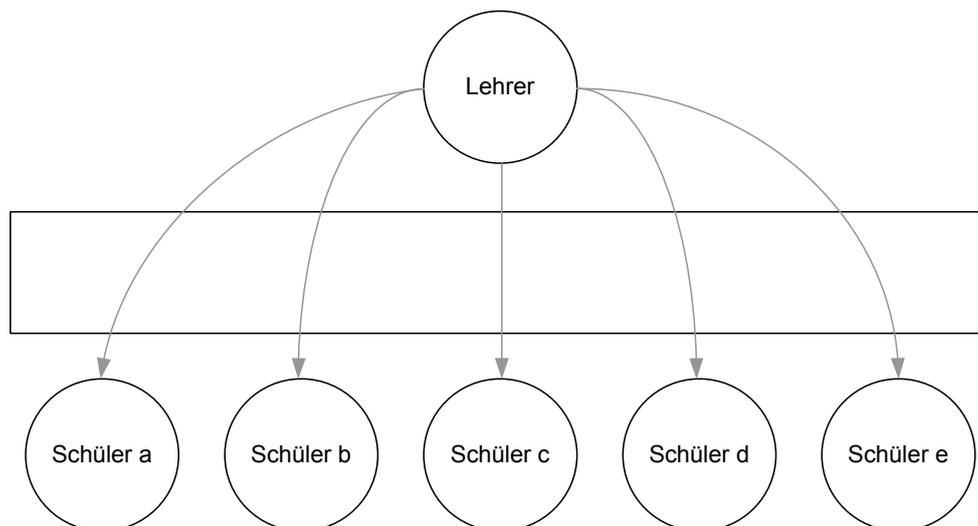


Abbildung 4.3: Raum-Setting beim Frontalunterricht [34]

Umständen kann dem Frontalunterricht eine gewisse Attraktivität nicht abgesprochen werden, denn (einfache) Inhalte lassen sich somit relativ schnell und effektiv vermitteln. Diese und gewiss auch historische Umstände [38] sind wohl für die bereits erwähnte Präsenz des Frontalunterrichts verantwortlich. Und laut Reich [36] lassen sich mit dem Frontalunterricht durchaus ansprechende Lernergebnisse erzielen. Vor allem dann, wenn es eine Lehrkraft versteht, über den Weg der direkten Instruktion Wissen zu vermitteln, indem Hintergründe und Zusammenhänge auf verständliche und nachvollziehbare Weise dargelegt werden.

Problematischer wird es hingegen, wenn fertige Wissenssysteme präsentiert werden [38]. Als Konsequenz sind die Lernenden gezwungen, die Inhalte gemäß der didaktischen Aufbereitung der Lehrkraft zu übernehmen, die sie im günstigsten Fall „eins zu eins“ reproduzieren können. Also das, was umgangssprachlich als „stupidies Auswendiglernen“ bezeichnet wird. Die Konstellation des Frontalunterrichts verlangt es, dass alle Lernenden den Ausführungen der Lehrkraft gleichermaßen folgen können. Das erfordert - abgesehen von der persönlichen kognitiven Fähigkeit - eine überaus disziplinierte Haltung aller Lernenden; was keine Selbstverständlichkeit ist. Folglich muss diese Haltung von der Lehrkraft gegebenenfalls wiederholend, oder durch autoritäres Verhalten eingefordert werden. Dass diese extrinsische Motivationsform nicht unbedingt zielführend ist, ist naheliegend.

Es wird deutlich, dass diese Rahmenbedingungen in keiner Weise den Anforderungen der Kompetenzorientierung gerecht werden; es erfolgt weder ein eigenständiger Wissenserwerb im konstruktivistischen Sinne, noch wird die Selbstständigkeit oder Klassengemeinschaft gefördert. Außerdem findet die Entwicklung sozialer Kompetenzen nahezu keine Berücksichtigung. Insgesamt betrachtet lässt sich festhalten, dass Unterrichtseinheiten durchaus (kurze) frontale Phasen beinhalten können, um beispielsweise in ein Thema einzuführen, neues Material vorzustellen oder (Evaluierungs-)Ergebnisse zu präsentieren [38]. Jedoch sollte dies zeitlich stark beschränkt sein und in jedem Fall in Kombination mit handlungsorientierten Methoden erfolgen.

4.4.1 Grundlegende Formen handlungsorientierter Methoden

Abgesehen von den diversen Ausprägungen des Frontalunterrichts (z.B. Fragen-entwickelnde Methode) lassen sich drei zentrale Unterrichtsmethoden - Reich [39] bezeichnet diese als „klassische Methoden“ - unterscheiden:

- Einzelarbeit
- Partnerarbeit
- Gruppenarbeit

Nachfolgend werden diese überblicksartig vorgestellt. Insofern nicht explizit darauf hingewiesen wird, bildet der Methodenpool der Universität Köln [39] die Grundlage zur Ausarbeitung.

Einzelarbeit

Bei allen drei Vertretern handelt es sich um überaus bekannte und erprobte Methoden. Die Einzelarbeit wird grundsätzlich im Bereich des individuellen Lernens eingesetzt. Vor allem dann, wenn es um die Wiederholung und somit Festigung bekannter Inhalte geht. In der Regel erfolgt dies in schriftlicher Form. Eine Alternative stellt beispielsweise die mündliche Artikulation dar. In jedem Fall ist es wichtig, dass es sich um einen konstruktiven Prozess und nicht um ein einfaches Nachmachen von etwas Bekanntem handelt. Bewerkstelligen lässt sich das, indem die Einzelarbeit entweder die Überführung des vorhandenen Wissens in eine andere Darstellungsform oder eine Transformation auf eine vergleichbare Aufgabe erfordert. Je nach Bedarf kann der Aufgabenstellung eine kurze Einführungsphase vorausgehen, die beispielsweise mit einem gemeinsamen Brainstorming bewältigt werden kann. In diesem Zusammenhang ist es von Bedeutung, dass es nach Abschluss der Einführungsphase allen Lernenden möglich ist, die Einzelarbeit zu bewältigen. Es ist nicht zielführend, wenn die Einführungsphase in Form eines Frontalvortrags erfolgt, dem nur eine Gruppe Lernender folgen kann und somit für die darauf folgende Einzelarbeit gerüstet ist. Die Konsequenz wäre, dass jene, die nicht über das vorausgesetzte (Hintergrund-)Wissen verfügen, motivationale Störungen erleiden oder gänzlich resignieren würden.

Grundsätzlich sollte die Durchführung in den Unterricht eingebettet sein. Somit ist sichergestellt, dass die Lehrkraft unterstützend eingreifen kann. Ein Lernender sollte nicht zum Einzelkämpfer verkommen. Die Einzelergebnisse sollten wiederum in die Lernergebnisse der Gruppe zurückfließen und nicht isoliert für sich stehen. Nachdem individuelle Lernphasen entsprechend Zeit benötigen, gilt es, ein entsprechendes Zeitbudget zu berücksichtigen. Da sich Einzelarbeit gut mit anderen Methoden kombinieren lässt, sollte sie - insofern sie nicht ohnehin schon impliziter Bestandteil einer spezifischen handlungsorientierten Methode ist (siehe Kapitel 4.4.2) - fester Bestandteil des Unterrichts sein.

Partnerarbeit

Eine weitere Form des Lernens stellt die Partnerarbeit dar. Hierbei handelt es sich ebenfalls um eine schülerzentrierte Methode, bei welcher zwei Lernende gemeinsam an einer Aufgabenstellung arbeiten. Da die gemeinschaftliche Lösungsfindung das primäre Ziel ist, wird durch das kooperative Lernen neben der Methoden- vor allem die Entwicklung der Sozialkompetenz gefördert. Dies ist im Wesentlichen darauf zurückzuführen, dass der Lösungsfindungsprozess eine aktive Auseinandersetzung mit einem Thema und in weiterer Folge mit der Sichtweise und der Auffassung des Lernpartners/der Lernpartnerin erfordert. Damit dieser Prozess von den Lernenden zielführend bewerkstelligt werden kann, ist eine entsprechende Kommunikations- und Diskussionskultur erforderlich. Um die hierfür erforderlichen Kompetenzen zu erwerben, stellt die Partnerarbeit ein gutes Instrument dar.

Damit die Durchführung möglichst von Erfolg gekrönt ist, gilt es, einige Punkte zu berücksichtigen. Hierzu zählt die Themenwahl. Das Thema sollte grundsätzlich „interessant“ sein und vor allem mehrere Meinungen und Sichtweisen ermöglichen, sodass Potenzial für eine Diskussion gegeben ist. Die Einleitung oder Hinführung zum Thema könnte zum Beispiel mit einer provokanten Frage erfolgen. Bei der Paarbildung ist Heterogenität (nach Geschlecht, Leistung oder Vorwissen) gewünscht. Die Ergebnissicherung und -rückführung kann beispielsweise mit kurzen Lernendenpräsentationen erfolgen.

Gruppenarbeit

Die Methode Gruppenarbeit ist überaus bekannt und findet inzwischen auch breite Anwendung in der (berufsbildenden) Unterrichtspraxis. Zurückzuführen ist das auf den Umstand, dass beispielsweise die Teamfähigkeit mittlerweile eine Grundvoraussetzung für nahezu jede Arbeitsstelle ist und dementsprechend in der Ausbildung Berücksichtigung finden muss. Neben der Förderung der Sozialkompetenz ist die Gruppenarbeit aber auch bestens dafür geeignet, fachliche und methodische sowie kommunikative Kompetenzen möglichst effektiv zu vermitteln. Sei es um ein bekanntes Thema zu vertiefen, ein neues zu erschließen oder bestimmte Inhalte zu wiederholen.

Bei der Gruppenarbeit an sich handelt es sich - wie bei der Partnerarbeit - um eine schülerzentrierte Unterrichtsmethode, die das selbstständige Arbeiten beziehungsweise Lernen in den Fokus rückt. Da in der Regel mehrere Personen im Team zusammenarbeiten, eignet sich diese Methode besonders für schwierige Aufgaben, die gegebenenfalls ein gewisse Kreativität oder Qualität in der Problemlösung erfordern. Durch die Gruppendiskussion - vorausgesetzt, dass diese nach bestimmten Regeln abläuft - entstehen neue Sichtweisen und Perspektiven, die in kumulierter Form zur Problemlösung führen. Dieser Prozess kann auch dazu beitragen, dass mögliche Wissenslücken bei Lernenden geschlossen werden, welche im Laufe des Regelunterrichts entstanden sind.

Damit die positiven Aspekte der Gruppenarbeit auch zum Tragen kommen, sind einige Punkte zu berücksichtigen. Hierzu zählt unter anderem die Gruppengröße, die im Bereich von drei bis maximal fünf Mitgliedern liegen sollte. Wie in der Berufswelt, gilt auch beim schulischen Lernen: Sollen mehrere Personen im Team zusammenarbeiten, bedarf es einer guten Planung vor und einer entsprechenden Organisation während der Gruppenarbeit. In

der Berufswelt obliegt diese Aufgabe zum Beispiel dem/der ProjektmanagerIn, in der Schule der Lehrkraft.

Es bleibt anzumerken, dass mit der Gruppenarbeit ein überaus vielseitiges Instrument zur Verfügung steht, das positive Effekte auf mehreren Ebenen mit sich bringen kann. Beispielsweise bauen die Lernenden Beziehungen zu ihren KollegInnen auf. Aufgrund der gemeinsamen Bearbeitung einer Aufgabenstellung lernen sie die Eigenheiten und Herausforderungen gruppenspezifischer Prozesse kennen. Die Zusammenarbeit in einer Gruppe wirkt grundsätzlich motivierend. Außerdem muss jedes Gruppenmitglied Verantwortung übernehmen, was wiederum den Zusammenhalt fördert. Ebenso sind die Lernenden mit der didaktischen Aufbereitung erzielter Ergebnisse konfrontiert, was die diesbezügliche Kompetenzentwicklung positiv beeinflusst. Bei Gruppenarbeiten besteht jedoch auch die Gefahr, dass ein Gruppenmitglied unbeteiligt bleibt und zum sogenannten Trittbrettfahrer wird. Oder dass eine Person diskriminiert wird, indem ihr die ganze Arbeit übertragen wird. Ebenso nicht zielführend ist, wenn ein/eine überaus begabte(r) Lerner/Lernerin sämtliche Arbeit an sich reißt, und den anderen keine Chance zur Mitarbeit ermöglicht. Derartige negative Erfahrungen gilt es auf Seiten der Lernenden zu vermeiden.

Im nachfolgenden Kapitel 4.4.2 werden spezifische Vertreter vorgestellt, die auf dem Grundgedanken der Gruppenarbeit basieren.

4.4.2 Spezifische Vertreter handlungsorientierter Methoden

Aufbauend auf den in Kapitel 4.4.1 vorgestellten Unterrichtsmethoden behandelt dieser Abschnitt spezifische, handlungsorientierte Methoden. Reich bezeichnet diese in [39] als „große Methoden“. Vielen dieser großen Methoden ist gemein, dass sie auf den Ideen der Partner- oder Gruppenarbeit aufbauen, diese jedoch in spezifischer Form adaptieren oder ergänzen. Es werden ausschließlich jene Methoden berücksichtigt, die für den weiteren Verlauf der Arbeit von Bedeutung sind.

Problem Based Learning

Wie die in Kapitel 3.2 erwähnte Definition von „Kompetenz“ verlangt, sollen Lernende fachspezifische Probleme lösen können. Dass die geforderte Problemlösungskompetenz elementar ist, wird spätestens beim Eintritt in die Berufswelt deutlich: viele Tätigkeiten - vor allem im Bereich der Technik - haben ihren Ursprung in konkreten Problemstellungen. Damit Lernende die entsprechenden (Lösungs-)Kompetenzen erlangen, empfiehlt sich das so genannte Problem-Based Learning (PBL).

Wie die Bezeichnung dieser Methode bereits zum Ausdruck bringt, findet das individuelle Lernen aufgrund von Erfahrungen statt, die beim Lösen einer möglichst authentischen Problemstellung (engl. *real-world problem*) gesammelt werden. Hierbei wird eine aktive Rolle von den Lernenden erwartet; bestehendes Wissen gilt es anzuwenden, fehlendes ist in der Gruppe zu generieren. Daraus wird deutlich, dass die Entwicklung konkreter Handlungskompetenzen gefördert wird. Das betrifft einerseits die Sach-, Methoden- aber auch Medienkompetenz, die beispielsweise beim selbstständigen Wissenserwerb (z.B. durch Literaturrecherche im Internet) gefördert wird. Da PBL in der Regel auf der Sozialform Gruppenarbeit (vgl. Kapitel

4.4.1) basiert, wird durch die Kollaboration andererseits die Entwicklung sozialer Kompetenzen gefördert. Eine zusammenfassende Darstellung der Ziele von PBL liefert Barrows in [40]:

- Aufbau einer Wissensbasis (im Sachgebiet)
- Entwicklung von Fähigkeiten zur Problemlösung und selbstgesteuertem Lernen
- Entwicklung von Teamfähigkeit
- Steigerung der Lernmotivation

Die Problemstellung ist Ausgangspunkt für die Lernenden und somit zentraler Bestandteil dieser Unterrichtsmethode. Folglich gilt es, einige Aspekte bei der Problemdefinition zu berücksichtigen. Dutch definiert hierfür in [41] fünf charakteristische Merkmale:

1. Die Problemstellung sollte das Interesse der Lernenden wecken und die Motivation zur Bearbeitung fördern. Um das zu erzielen, muss die Problemstellung, wie bereits erwähnt, möglichst authentisch sein. Dadurch ist es den Lernenden möglich, einen Bezug herzustellen.
2. Ein Problem sollte mehrere Lösungsansätze/-wege zulassen. Das lässt sich am besten durch Entscheidungen, die von der Gruppe im Laufe des Problemlösungsprozesses zu treffen sind, realisieren. In diesem Zusammenhang ist es wichtig, dass es keine „richtige“ Lösung gibt.
3. Die Problemstellung sollte „open-ended“ und durchaus kontrovers sein. Dies fördert beispielsweise die gruppeninterne Kommunikation und kann wiederum für die Entwicklung der sozialen als auch persönlichen Kompetenz zuträglich sein.
4. Ebenfalls von Bedeutung ist, dass die Problemstellung eine entsprechende Komplexität aufweist. Die Problemstellung sollte daher über den aktuellen Wissensstand hinausgehen, oder zumindest eine Neuordnung des aktuellen Wissens erfordern.
5. Die inhaltliche Zielsetzung des Gegenstands/Kompetenzbereichs sollte in die Problemstellung eingearbeitet sein.

Da PBL in der Regel auf Gruppenarbeit basiert, gelten die entsprechenden Rahmenbedingungen (siehe Kapitel 4.4.1, Gruppenarbeit). Hinsichtlich der Durchführung PBL-basierter Einheiten empfiehlt sich der sogenannte „Siebensprung“. Laut Reich [42] handelt es sich hierbei um ein bekanntes Modell, das auch unter „McMaster Modell“ bekannt ist. Wie die Bezeichnung bereits zum Ausdruck bringt und der Abbildung 4.4 zu entnehmen ist, durchläuft der Lernprozess sieben Schritte.

- **Phase 1 - Problemanalyse:** Im ersten Schritt klärt die Gruppe, ob alle die Begriffe kennen und gleich verstehen. Im zweiten Schritt ist es das Ziel, das Problem zu identifizieren und zu definieren. Danach wird mit dem Schritt 3 das Problem analysiert. Hierbei wird das problemspezifische Vorwissen der Gruppe zusammengetragen und in

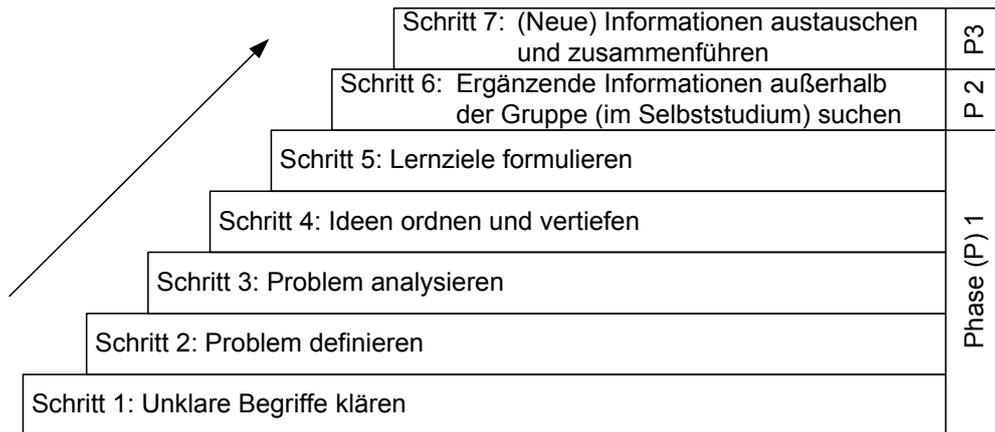


Abbildung 4.4: Ablauf eines PBL-Zyklus (Eigendarstellung nach [43] u. [44])

Form eines Brainstormings werde Lösungsvorschläge gesammelt und schriftlich festgehalten. Eine Diskussion findet (noch) nicht statt. Im Schritt 4 werden die gesammelten Ideen geordnet, mit Überbegriffen versehen und vertieft. Dies geschieht im Rahmen einer intensiven Gruppendiskussion. Es gilt Prioritäten zu setzen. Aufkommende Fragen, die (noch) nicht beantwortet werden können, sind schriftlich festzuhalten. Im letzten Schritt dieser Phase, Schritt 5, sind die Lernziele (und Hypothesen) zu formulieren. Die Lernenden entscheiden, welche Fragen aus Schritt 4 hinsichtlich der Problemlösung von Relevanz und somit einer weiterführenden Betrachtung zu unterziehen sind.

- **Phase 2 - Problembearbeitung:** Schritt 6 konzentriert sich auf die Informationsbeschaffung. Hierbei ist es das Ziel, dass sich die Lernenden absprechen, wer welche Medien, Quellen etc. recherchiert. Dies kann laut Moust et al. [43] auch in Einzel- oder Partnerarbeit erfolgen, indem sich die Gruppe teilt.
- **Phase 3 - Problemlösung:** Nach der Informationsbeschaffung (Schritt 6) werden die Ergebnisse in der Gruppe diskutiert, kritisch hinterfragt und gegebenenfalls ergänzt. Es soll gemeinsam reflektiert werden, was in Bezug auf die formulierten Lernziele gelernt wurde. Lernende unterrichten sich in dieser Phase gegenseitig und gleichen ihre Überlegungen ab.

Hinsichtlich der Beurteilung bei PBL führt Reich die sogenannte Evaluation an [42]. Evaluation bedeutet in diesem Zusammenhang Feedback aus unterschiedlichen Perspektiven zu erteilen. Es haben sich vier Formen bewährt: die Selbstevaluation, die Peer-Evaluation, die formative und die summative Evaluation.

Dass diese Methode ihre Vorteile hat, ist evident. Hmelo-Silver hält in [45] fest, dass PBL grundsätzlich die Fähigkeit des selbstgesteuerten Lernens - was vor allem über die institutionelle Ausbildung hinaus von großem Wert ist - oder die Fähigkeit des reflektiven sowie flexiblen Denkens durchaus fördert. Die intensive Kollaboration trägt wiederum zur nachhaltigen Bildung von sozialen und personalen Kompetenzen bei. Erfahrungsberichte zeigen, dass PBL überaus gewinnbringend im Bereich der (universitären) Informatikausbildung eingesetzt werden kann: ob im Bereich von Tutorials [46] oder umfassenden Programmierkursen [47]. Anzumerken bleibt, dass - vor allem bei PBL-basierten Lehrplänen - darauf zu achten

ist, dass bei der Vermittlung von Problemlösungskompetenzen allgemein gültige Konzepte, Ideen etc. und nicht ausschließlich disziplinspezifische Ansätze vermittelt werden [48].

Projektarbeit

Mit der Projektarbeit wird eine weitere Methode vorgestellt, die vor allem im Bereich der Informatik von großer Bedeutung ist. Hartmann [16] hält diesbezüglich fest, dass elementare Strategien und Methoden erst beim Lösen komplexer Probleme bei der Entwicklung „großer“ (Software-)Systeme vermittelt werden können. So lässt sich beispielsweise die Bedeutung der Dokumentation ausschließlich im Rahmen größerer Softwareentwicklungsprojekte plausibel verdeutlichen. Werden im Unterricht jedoch nur kleine Programme geschrieben, geht die Bedeutung der Dokumentation unter Umständen nicht klar hervor. Ein weiteres Argument für den Projektunterricht stellt der Umstand dar, dass reale Informatikprojekte in der Regel projektbasiert abgewickelt werden. Folglich ist es von Bedeutung, dass Lernende bereits erste - wenn auch konstruierte - Erfahrungen mit den Eigenheiten und Herausforderungen der Projektarbeit sammeln. Jedoch gilt es an dieser Stelle anzumerken, dass die Methode Projektarbeit nicht mit der Bearbeitung realer Projekte (aus dem Bereich der Informationstechnologie) gleichzusetzen ist. Realen Projekten liegt ein Auftraggeber/-nehmersverhältnis zugrunde, bei welchem der Auftraggeber im Detail die Projektziele definiert und Rahmenbedingungen anführt. Dieses Faktum steht klar im Widerspruch zur Methode Projektarbeit, bei der beispielsweise die Zieldefinition den Lernenden obliegt. Werden also „reale“ IT-Projekte im Unterricht umgesetzt, dann handelt es sich eher um projektartigen Unterricht, der von der Methode Projektarbeit zu unterscheiden ist [49].

Wie jedoch bei realen Projekten planen die Lernenden selbst, wie sie die Aufgabenstellung angehen. Sie übernehmen, wie bereits erwähnt, die Zielsetzung und legen mit einem Zeitplan fest, wer was bis wann zu erledigen hat. Somit werden die Lernenden beispielsweise mit den Herausforderungen der Aufwandsabschätzung und in weiterer Folge mit der Vergabe einzelner Zeitbudgets konfrontiert. Damit der Projekterfolg eintritt, müssen Meilensteine vereinbart werden, welche einzelne Teilergebnisse beschreiben und terminisieren. Die Teilergebnisse gilt es wiederum zu evaluieren und schlussendlich zu einem Ganzen zusammenzufügen.

Kommt die Projektmethode zum Einsatz, ist es wichtig, allen Lernenden die Möglichkeit zu bieten, in die Rolle des Projektleiters/der Projektleiterin zu schlüpfen. Um die Motivation zu fördern, weckt das Projekt idealerweise die Neugierde der Lernenden und entwickelt im günstigsten Fall eine Eigendynamik. Hartmann verweist in punkto Durchführung auf das siebenstufige Modell von Frey [50]:

1. **Projektinitiative:** Das Ziel der ersten Stufe ist das Darlegen der Aufgabenstellung oder der Idee. Hierbei gilt: die Projektinitiative ist offen zu führen. Einfach nur ein klar definiertes Problem zu lösen, ist nicht das Ziel.
2. **Eingrenzung:** Stufe zwei setzt sich mit der Begutachtung und Diskussion der generierten Ideen und möglichen Fragen, die im Zuge des Denkprozesses aufkommen, auseinander. Der Lehrkraft obliegt es, gemeinsam mit Lernenden Ideen hinsichtlich deren Umsetzung zu validieren. Hierbei spielen vor allem die Faktoren Zeit, Ressourcen oder Vorkenntnisse eine zentrale Rolle. Unter Umständen müssen die Projektziele

eingegrenzt und Schwerpunkte, die im Rahmen der Projektarbeit zu verfolgen sind, gesetzt werden.

3. **Projektplan:** Nachdem die Zieldefinition erfolgt ist, gilt es festzulegen, wer welche Tätigkeiten bis wann zu erledigen hat. Ebenso ist zu klären, wie die Zusammenarbeit (z.B. Informationsaustausch in der Gruppe) erfolgen soll. Ergebnis dieser Stufe ist ein Arbeitsplan, der diesbezügliche Aspekte zusammengefasst darstellt.
4. **Ausführung:** Ist alles geplant, kann die Ausführung erfolgen. Die Lernenden führen die geplanten Aktivitäten aus und vergleichen, ob die erzielten Ergebnisse der Zielsetzung im Projektplan entsprechen. Gegebenenfalls muss der Projektplan adaptiert werden.
5. **Abschluss:** Die Phase Abschluss widmet sich primär der Ergebnispräsentation. Ein gemeinsamer Rückblick schließt das Projekt ab.

Als integraler Bestandteil des Modells sind die Stufen **Fixpunkte** (vgl. Meilensteine) und **Metainteraktion** zu verstehen. Meilensteine stellen sicher, dass ein Informationsaustausch zu definierten Zeitpunkten erfolgt. Das ist vor allem bei Projekten, die über einen größeren Zeitraum durchgeführt werden, wichtig. Die Stufe Metainteraktion zielt auf die Entwicklung sozialer Kompetenzen, oder Kompetenzen im Bereich des Projektmanagements ab. Es erfolgt ein laufender Austausch im Projektteam über den Projektverlauf. Hierbei sind nach Frey [50] Fragestellungen wie „Sind wir auf dem richtigen Weg?“, „Gibt es Schwachstellen?“ oder „Funktioniert die Kommunikation in der Gruppe?“ zu diskutieren.

Es bleibt anzumerken, dass der projektbasierte Unterricht - von der Methode Projektarbeit bis zum projektartigen Unterricht - ein hohes Maß an Handlungsorientierung aufweist und somit den Anforderungen des kompetenzorientierten Unterrichts (siehe Kapitel 4.2) gerecht wird. Die Methode Projektarbeit bietet vor allem bei der Zieldefinition oder Projektabwicklung mehr Spielraum als es bei realen IT-Projekten der Fall wäre. Dadurch kann Raum für eigene kreative Lösungen entstehen, was für hohes Identifikationspotential und somit Motivation bei der Projektdurchführung sorgt. Reale IT-Projekte weisen hingegen Charakteristika auf, deren Kenntnis vor allem in Hinblick auf die berufliche Tätigkeit der Lernenden von zentraler Bedeutung sind. Hierzu zählt beispielsweise die Beziehung Auftraggeber und -nehmer. Je nach Blickwinkel werden unterschiedliche Projektinteressen verfolgt. Dies als Lernender zu erfahren, ist durchaus von Bedeutung für die berufliche Zukunft. Ebenso können die oft ambitionierten zeitlichen Anforderungen oder die technischen Rahmenbedingungen realer IT-Projekte, die es bei der Projektplanung und -durchführung zu berücksichtigen gilt, einen wertvollen Beitrag zum Qualifikationsprofil der Lernenden leisten.

Für weiterführende Information sowie konkrete Beispiele wird auf die Literatur Frey [50], Hartmann [16] oder Reich [49] verwiesen.

Gruppenpuzzle

Die Methode Gruppenpuzzle kann, wie der Name bereits zum Ausdruck bringt, der Sozialform Gruppenarbeit (siehe Kapitel 4.4.1) zugeordnet werden. Folglich haben die Regeln und Rahmenbedingungen der Gruppenarbeit Gültigkeit. Das Besondere dieser Methode ist der Rollenwechsel: die Lernenden nehmen auch die Rolle des Lehrenden ein. Im Detail bedeutet

das, dass mehrere Lernende ein Teilgebiet eines Themas in den so genannten Expertengruppen didaktisch aufbereiten und dieses in weiterer Folge unterrichten. Hierfür werden sogenannte Unterrichtsgruppen gebildet.

Zu Beginn gilt es die Expertengruppen zu bilden (siehe Abbildung 4.5, Expertenrunde). Hartmann schlägt Gruppengrößen von drei bis maximal sechs Lernenden vor [16]. Bei der Zusammenstellung der Expertengruppen ist darauf zu achten, dass dies leistungsheterogen - in der Regel gelingt das per Los - erfolgt [49].

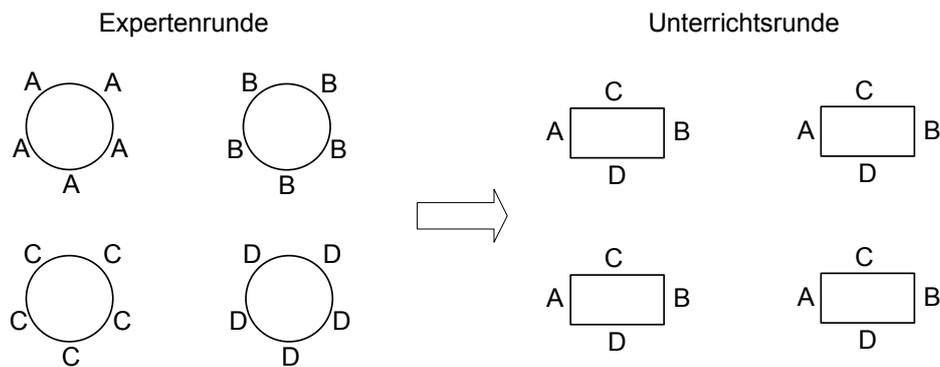


Abbildung 4.5: Aufteilung in Experten- und Unterrichtsgruppe [16]

Wie bereits erwähnt, bearbeitet jede dieser Expertengruppen einen Teilbereich des Themas; wobei das erste Lesen zwar bereits in der Expertengruppe aber noch als Einzelarbeit erfolgt. Wichtig ist, dass das Thema überschaubar ist und sich in nahezu gleich große Blöcke aufteilen lässt. Nachdem eine erste Sichtung der Materialien erfolgt ist, gilt es abzuklären, ob die angehenden ExpertInnen alle Begriffe verstehen. Ziel ist, dass alle Lernenden zu wirklichen ExpertInnen werden. Schlussendlich müssen sie ja zu einem späteren Zeitpunkt die Rolle des Lehrenden einnehmen und das erfordert entsprechendes Know-how. Um diesen Prozess zu fördern, sind Kontrollfragen, die eine Selbstüberprüfung ermöglichen, überaus zielführend. Die Lernenden können somit relativ einfach überprüfen, ob sie bereits über das erforderliche Expertenwissen verfügen oder nicht. Ergänzend dazu ist es Aufgabe der Lehrkraft, durch stilles Beiwohnen der Diskussion in den Expertengruppen, festzustellen, ob noch (partielle) Wissenslücken bestehen. Je nach Stoffumfang variiert die Dauer der Expertenrunde. Danach sind die Expertengruppen aufzulösen - die Lernenden bilden die Unterrichtsgruppen, die ebenfalls vorher zu definieren sind.

Eine Unterrichtsgruppe (siehe Abbildung 4.5, Unterrichtsgruppe) besteht aus zumindest einem Lernenden aus jeder Expertengruppe. Bei vier Teilbereichen besteht somit eine Gruppe der Unterrichtsgruppe aus mindestens vier Lernenden. Nachdem die neue Gruppeneinteilung erfolgt ist, unterrichten die ExpertInnen der Reihe nach die Gruppe.

Ergänzend zu den bereits formulierten Vorteilen der Gruppenarbeit (siehe Kapitel 4.4.1), erwerben die Lernenden didaktische Kompetenzen, da sie Inhalte aufbereiten und vermitteln müssen. Weiters hält Reich in [49] fest, dass weitere Kompetenzen und vor allem auch inhaltliche Erfahrungen gesammelt werden. Da die Lernenden ein Thema selbstständig erarbeiten müssen, sind sie dazu angehalten, Relevantes aus Texten zu extrahieren. Nachteilig kann sich der Umstand auswirken, dass nicht jeder den gesamten Stoff durcharbeitet. Je nach Kompetenz des/der jeweiligen Experten/Expertin werden Inhalte manchmal besser und manchmal weniger gut übermittelt.

Leitprogramm

Im Kapitel 4.2 ist mit der Individualisierung ein zentrales Merkmal des kompetenzorientierten Unterrichts angeführt. Eine Methode, die dieses Thema adressiert, ist das sogenannte Leitprogramm. Vereinfacht dargestellt, handelt es sich bei einem Leitprogramm um eine schriftliche Anleitung mit „Kochrezept-Charakter“, die den Lernenden Schritt für Schritt anleitet und zu einem klaren Lernziel führt [51]. Im Detail wird der zu vermittelnde Stoff in einzelne Pakete aufgeteilt und mit einer ausführlichen Arbeitsanleitung für das Selbststudium versehen. Die Arbeitsanleitung bereitet den Stoff in leicht verständlicher Form auf, enthält in der Regel Zusatzhilfen und sorgt mittels Kontrollfragen für die Wissenssicherung. Hartmann empfiehlt nach jedem Paket eine von der Lehrkraft gesteuerte Lernstandskontrolle [16]. Erst wenn der/die Lernende den Stoff versteht, kann das nächste Lernpaket in Angriff genommen werden. Der eingangs erwähnten Individualisierung wird dadurch Rechnung getragen, indem ein Leitprogramm einerseits über ein Fundamentum und andererseits über ein Additum verfügt (siehe Abbildung 4.6). Das Fundamentum beinhaltet jene Lehrinhalte, die

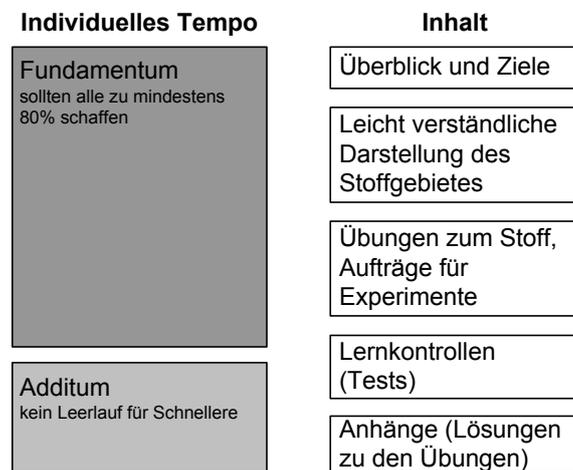


Abbildung 4.6: Aufbau einer Leitprogrammseinheit [16]

von allen Lernenden - Hartmann definiert in diesem Zusammenhang mindestens 80 Prozent - erfolgreich absolviert werden müssen. Da davon auszugehen ist, dass einige Lernende das Fundamentum schneller absolvieren, stellt das Additum weiterführende Aufgaben bereit. Damit diese auch in Angriff genommen werden, sind entsprechende Reize zu setzen. Unumgänglich ist, dass die Aufgaben des Additum über eine gewisse Attraktivität verfügen und einen Mehrwert nach sich ziehen. Ebenso erscheint eine adäquate Belohnung als durchaus zielführend. Andernfalls kann die Motivation, das Additum zu absolvieren, erheblich darunter leiden. Hinsichtlich der Integration des Additums bieten sich grundsätzlich zwei Möglichkeiten an: Entweder als ergänzender Bestandteil, indem zum Beispiel jede Übungsaufgabe in einen Pflicht- und einen freiwilligen Zusatzteil unterteilt wird oder als abschließendes Element am Ende eines Kapitels.

Insgesamt betrachtet lässt sich konstatieren, dass mit der Methode Leitprogramm der Heterogenität und den damit einhergehenden unterschiedlichen Lerntempi der Lernenden entgegengewirkt werden kann. Nachdem die Lernenden ein Leitprogramm größtenteils im Selbststudium durcharbeiten, wird eine gewisse Kompetenz und Eigenverantwortung vorausgesetzt. Und dass die Entwicklung sozialer Kompetenzen ebenso nicht das Ziel dieser Methode

ist, liegt nahe - vor allem wenn sie isoliert eingesetzt wird. Nicht zu vergessen ist der enorme Arbeitsaufwand für die Lehrkraft, den die Erstellung und Wartung eines Leitprogramms mit sich bringt.

Weiterführende Informationen bezüglich der Methode Leitprogramm sind in [52] oder [51] enthalten. Für praktische Leitprogrammbeispiele aus den Bereichen Informatik oder Mathematik bieten sich [53] und [54] sowie [55] der ETH Zürich an.

4.5 Leistungsfeststellung und -beurteilung

Wie in der Einleitung dieses Kapitels bereits erwähnt, erfordert die Kompetenzorientierung einen Paradigmenwechsel. Ein Teilbereich, der in hohem Maße davon betroffen ist, ist die Leistungsfeststellung und -beurteilung. Im Grundlagenpapier zum kompetenzorientierten Unterrichten [35] wird explizit darauf hingewiesen, dass die aktuelle Leistungsbeurteilungsverordnung (LBVO)[56] den Anforderungen der Kompetenzorientierung nicht gerecht wird. So finden beispielsweise ein fächerübergreifendes Arbeiten oder Gruppenarbeiten keine Berücksichtigung, obwohl sie im Rahmen des kompetenzorientierten Unterrichts zentrale Elemente darstellen (vgl. Kapitel 4.4).

Diese Umstände erfordern neben den gesetzlichen Rahmenbedingungen auch ein Umdenken bei den Lehrenden. In der derzeitigen Unterrichtspraxis umfasst die Leistungsfeststellung hauptsächlich reine „Wissensfragen“ mit dem Ziel, auswendig Gelerntes zu reproduzieren. Aufgrund von Zeitdruck oder aus anderen organisatorischen Gründen erfolgt dies in der Regel in kollektiver und schriftlicher Form - was nicht zwingend negativ sein muss. Jedoch kann mit Wissensfragen nur bedingt eine Handlungskompetenz erfasst werden. Ebenso findet der persönliche Lernfortschritt keine Berücksichtigung. Daher kommen die Autoren des Grundlagenpapiers zu dem Schluss, dass kompetenzorientierte Leistungsfeststellung eher einem individuellen Prozess (vgl. mündliche Leistungsfeststellung) gleichkommen sollte. Nachfolgend werden ausgewählte Methoden und Instrumente zur kompetenzorientierten Leistungsfeststellung überblicksartig vorgestellt, die einerseits die personenzentrierte Sichtweise sowie andererseits den kumulativen Lernzuwachs berücksichtigen.

4.5.1 Lernvertrag

Die grundlegende Idee beim Lernvertrag ist, wie die Bezeichnung bereits erahnen lässt, dass ein Vertrag zwischen Lehrkraft und dem/der Lernenden abgeschlossen wird. Ein Schriftstück hält konkrete Lernziele und Maßnahmen, wie diese erreicht werden können sowie den Zeitraum, der zur Erreichung dieser Ziele zur Verfügung steht, fest. Amrhein-Kreml et al. weist jedoch explizit darauf hin, dass terminliche Abweichungen möglich sein dürfen, dies jedoch in Absprache mit der Lehrkraft zu erfolgen hat [57]. Unumgänglich ist hingegen, dass definierte Lernziele in jedem Fall an vorangegangene anschließen und den aktuellen Prozess begleiten.

Nachdem mit Hilfe der Methode Lernvertrag eine personenbezogene Dokumentation des Lernprozesses erfolgt, wird die individuelle Forderung aber auch Förderung überaus transparent gestaltet. Die positive Konsequenz ist, dass der Lernende Defizite rasch und verständlich erkennen kann, denn schlussendlich wurden vertragliche Vereinbarungen nicht eingehalten.

Weiters wird der Lehrkraft ein Instrument bereitgestellt, das eine gute Argumentationsgrundlage bei der Beurteilung liefert. Nebenbei trägt diese Methode auch zur Förderung der personalen Kompetenz bei, da die Lernenden Eigenverantwortung übernehmen müssen. Für weiterführende Informationen sowie konkrete Anwendungsbeispiele wird auf [57] oder [58] verwiesen.

4.5.2 Lerntagebuch

Personenzentrierung und Transparenz beim Kompetenzerwerbsprozess sind ausgeprägte Merkmale der Methode Lerntagebuch. In Anlehnung an ein herkömmliches Tagebuch halten die Lernenden ihre Lernerfahrungen fest. Sämtliche Ergebnisse, Lernfortschritte oder Lösungsprozesse sowie Fragen und Probleme werden von den Lernenden in ein spezielles Tagebuch eingetragen. Ergänzt werden diese Eintragungen mit Gedanken, Beobachtungen aber auch Gefühlen, die bei den vorangegangenen Lernprozessen aufgekommen sind. Daraus geht hervor: Lerntagebücher enthalten unter Umständen private Informationen. Das heißt, es ist im Vorhinein zu klären, wie mit diesen Inhalten umzugehen ist. Wird im Unterricht die Sozialform Gruppenarbeit eingesetzt, kann ein Tagebuch durchaus auch auf Gruppenebene geführt werden. Damit es zu täglichen Eintragungen kommt, sollte die Durchführung zu einem Fixpunkt im Unterricht erhoben werden. Alternativ könnten die Eintragungen zuhause erfolgen. Das setzt jedoch eine gewisse Reife und Motivation bei den Lernenden voraus.

Mit Hilfe des Lerntagebuchs können somit Stärken und Schwächen - entweder durch Eigenreflexion oder durch diagnostische Arbeit der Lehrkraft - erkannt werden. Aufgrund dieser Rückmeldungen kann gegebenenfalls eine Steuerung des individuellen Lernprozesses oder der Unterrichtsplanung erfolgen. Wie beim Lernvertrag, müssen die Lernenden Eigenverantwortung übernehmen. Aufgrund der detaillierten Aufzeichnungen kann jederzeit eine Bewertung des Lernerfolges im Rahmen eines Gespräches zwischen Lehrkraft und Lernenden/Lernender erfolgen. Amrhein-Kreml [57] führt folgende Vorschläge für Leitfragen an:

- Was habe ich gemacht?
- Was habe ich dabei erfahren und gelernt?
- Wie ist es mir hierbei gegangen?
- Wie ist das zu bewerten (Gesamteindruck)?
- Welche Erwartungen, Hoffnungen, Wünsche habe ich?
- Das nächste Mal werde ich...
- u.v.m.

Hinsichtlich des Aufzeichnungsumfangs sollten im Vorhinein Vereinbarungen getroffen werden, sodass sich der beiderseitige Aufwand - schlussendlich müssen die Inhalte von der Lehrkraft gelesen werden - in einem bewältigbaren Rahmen bewegt. Weiterführende Informationen und Beispiele sind in [57] oder [59] enthalten.

4.5.3 Portfolio

Die Portfolio-Methode stellt wohl einen der bekanntesten Vertreter dar. Nicht zuletzt deshalb, weil Portfolios in vielen anderen Bereichen breite Anwendung finden. So werden beispielsweise Wertpapiere in sogenannten Aktienportfolios verwaltet. Analog dazu können auch Lernergebnisse in einer „Mappe“ verwaltet werden. Laut [57] liegt der primäre Einsatzzweck bei der Dokumentation von Arbeitsprozessen. Ziel ist es, die besten und/oder repräsentativsten Arbeitsergebnisse zu sammeln, wodurch die Beobachtung des Lernfortschritts und in weiterer Folge die Reflexion darüber ermöglicht wird. Die Lernenden können die eigene Kompetenzentwicklung (jederzeit zurück-)verfolgen.

Bezüglich der konkreten Implementierung im Unterricht stellt die Literatur unterschiedliche Ansätze bereit. Im Grundlagenpapier „Kompetenzorientiertes Unterrichten“ [35] des BMUKK wird beispielsweise zwischen drei Portfoliotypen unterschieden: Arbeits-, Beurteilungs- und Präsentationsportfolio. Mit dem Arbeitsportfolio sollen die Stärken und Schwächen der Lernenden beschrieben werden, wodurch es einen diagnostischen Charakter aufweist. Das Beurteilungsportfolio soll zeigen, was der/die Lernende gelernt hat. Es dient auch als Grundlage von Lernentwicklungsgesprächen. Die tatsächliche Beurteilung erfolgt basierend auf dem Präsentationsportfolio. Laut Amrhein-Kreml [57] können mögliche Beurteilungskriterien sein:

- *Form und Gestaltung*
- *Umfang, Komplexität und Vollständigkeit*
- *Sprachliche Ausdrucksweise*
- *Korrektheit der Inhalte*
- *Ausmaß der Eigeninitiative*
- *Umfang der Reflexion*

Wichtig erscheint, dass die Ziele der Methode und die Erwartungen, die mit den Ergebnissen verknüpft sind, im Vorfeld geklärt werden. Hierfür bietet sich beispielsweise die zuvor erläuterte Methode Lernvertrag an.

Kapitel 5

Analyse der Ausgangssituation

Um die Grundlage für die anschließende Unterrichtskonzeption zu schaffen, analysiert dieser Abschnitt die Ausgangssituation mit dem Ziel, konkrete Anforderungen zu definieren. Hierbei werden jene Rahmenbedingungen in Betracht gezogen, die einerseits vom Gesetzgeber durch den Lehrplan und die Bildungsstandards sowie andererseits durch die HTL-Krems aufgrund der schulspezifischen Implementierung des Kompetenzbereichs „Industrielle Informationstechnik“ (INIT) vorgegeben sind. Einen weiteren Schwerpunkt stellt die Erhebung des State of the Art hinsichtlich des Fachbereichs „Embedded Systems“ als Lehrgegenstand an (Hoch-)Schulen dar. Die gewonnen Erkenntnisse werden schlussendlich in konkrete Anforderungen an die Unterrichtskonzeption überführt.

5.1 Höhere Technische Lehranstalt für Informationstechnologie

Mit dem Bundesgesetzblatt BGBl. II Nr. 300/2011 [7] erfolgte die Verordnung neuer Lehrpläne über sechs HTL-Fachrichtungen. Ebenfalls davon betroffen ist die Fachrichtung Informationstechnologie (IT). Ausschlaggebend hierfür ist einerseits der in Kapitel 3 angeführte Paradigmenwechsel in der Lehre, der neue pädagogische Grundlagen erfordert sowie andererseits der technologische Fortschritt, dem die neuen Lehrpläne ebenfalls Rechnung tragen sollen [60]. In Bezug auf die Fachrichtung IT bedeutet das, dass die ersten Jahrgänge ab dem Jahr 2011 sowie die ersten Jahrgänge des Schuljahres 2010/11 in den neuen kompetenzorientierten Lehrplan [61] wechseln. Die verbleibenden Jahrgänge 2009/10, 2008/09 und 2007/08 - also die gegenwärtigen dritten, vierten und fünften Klassen - werden weiterhin gemäß dem „alten“ Lehrplan [62] unterrichtet.

Nachdem diese Arbeit die Entwicklung didaktischer Konzepte für den Kompetenzbereich INIT des neuen Lehrplans verfolgt, bildet dieser die Grundlage für die weitere Ausführung. Folglich bezieht sich die Bezeichnung „(neuer) Lehrplan“ - insofern nichts Gegenteiliges angeführt ist - auf den neuen, kompetenzorientierten Lehrplan [61].

Die Aufgabe eines Lehrplans liegt im Wesentlichen darin, die Lehrziele und -inhalte einer Fachrichtung (von den allgemeinbildenden Pflichtgegenständen bis zu den Pflichtgegenständen der Fachtheorie und -praxis) festzulegen. Darüber hinaus werden auch noch allgemeine Bildungsziele, die beispielsweise die personale als auch soziale Ausbildung der Lernenden betreffen, beschrieben.

Wie einleitend erwähnt, stellte der Paradigmenwechsel - hin zur Kompetenzorientierung - einen zentralen Beweggrund für die Überarbeitung des Lehrplans dar. Vergleicht man nun von diesem Hintergrund ausgehend den neuen Lehrplan mit der Vorgängerversion, ist auf den ersten Blick ersichtlich, dass sich die Lehrinhalte in sogenannte Kompetenzfelder (vgl. Gegenstände) unterteilen; welche sich wiederum in mehrere Kompetenzbereiche aufschlüsseln. Nachfolgende Abbildung 5.1 zeigt die Stundentafel des neuen Lehrplans, welche die Kompetenzfelder/Gegenstände überblicksartig darstellt. Dieser kann beispielsweise das Ausmaß der Wochenstunden eines Kompetenzfelds/Gegenstandes im jeweiligen Jahrgang entnommen werden. Die jeweiligen Werte in den Klammern legen die Anzahl der Übungsstunden, die davon im Laboratorium abzuhalten sind, fest.

I. STUNDENTAFEL							
(Gesamtstundenzahl und Stundenausmaß der einzelnen Unterrichtsgegenstände)							
Pflichtgegenstände	Wochenstunden					Summe	Lehrverpflichtungsgruppe
	Jahrgang						
	I.	II.	III.	IV.	V.		
A. Allgemeine Pflichtgegenstände							
1. Religion	2	2	2	2	2	10	(III)
2. Deutsch	3	2	2	2	2	11	(I)
3. Englisch	2	2	2	2	2	10	(I)
4. Geografie, Geschichte und Politische Bildung	2	2	2	2	-	8	III
5. Wirtschaft und Recht	-	-	-	3	2	5	II bzw. III
6. Bewegung und Sport	2	2	2	1	1	8	(IVa)
7. Angewandte Mathematik	4	3	3	2	2	14	(I)
8. Naturwissenschaften	3	2	2	2	-	9	II
B. Fachtheorie und Fachpraxis							
1. Softwareentwicklung	3(2)	3(2)	3(2)	3(2)	2(2)	14	I
2. Informationstechnische Projekte	-	2	5(1)	6(4)	6(4)	19	II
3. Informationssysteme	-	-	3(1)	5(2)	4(2)	12	I
4. Systemtechnik	4(2)	6	5(3)	6(4)	10(6)	31	I
5. Medientechnik	2(2)	2(1)	2(2)	-	-	6	I
6. Netzwerktechnik	2	2(1)	3(2)	1	-	8	I
7. Computerpraktikum	4	4	-	-	-	8	IVa
Verbindliche Übung							
Soziale und personale Kompetenz	1(1)	1(1)	-	-	-	2	III
Gesamtwochenstundenzahl	34	35	36	37	33	175	

Abbildung 5.1: Stundentafel Lehrplan IT [61]

Wie der Abbildung 5.1 außerdem noch zu entnehmen ist, stellt die „Systemtechnik“ das Kompetenzfeld mit den meisten Wochenstunden im Laufe der Ausbildung dar. Im Detail beläuft sich die Stundenanzahl auf 31 - zumindest in der Vorgabe des Lehrplans, denn aufgrund schulautonomer Änderungen können diese Werte geringfügig variieren. Dem Kom-

petenzfeld Systemtechnik sind, wie bereits in Kapitel 3.3 festgehalten, neben der INIT die Kompetenzbereiche

- Elektronik und Elektrotechnik für IT,
- Grundlagen der Informatik,
- Betriebssysteme,
- Systemintegration und Infrastruktur sowie
- Dezentrale Systeme

zuzuordnen. Für diese Bereiche steht in Abhängigkeit des jeweiligen Jahrgangs eine bestimmte Stundenanzahl zur Verfügung; wobei nicht jeder Kompetenzbereich vom 1. bis zum 5. Jahrgang durchgehend Bestandteil der Ausbildung ist. So findet beispielsweise der für diese Arbeit bedeutende Kompetenzbereich INIT ausschließlich im 4. und 5. Jahrgang Berücksichtigung. Das Ausmaß der Wochenstunden eines Kompetenzbereichs ist hingegen nicht festgelegt - diese Entscheidung ist wiederum schulautonom zu treffen. Da jedoch die Kompetenzbereiche „Elektronik und Elektrotechnik für IT“, „Grundlagen der Informatik“ und „Betriebssysteme“ in den ersten drei Jahrgängen angesiedelt sind, verbleiben neben der INIT mit „Systemintegration und Infrastruktur“ sowie „Dezentrale Systeme“ nur zwei Kompetenzbereiche, mit welchen das Zeitbudget von 6 Stunden im 4. beziehungsweise 10 Stunden im 5. Jahrgang zu teilen ist (siehe Abbildung 5.1, Systemtechnik). Wie diese Aufteilung erfolgt, liegt im Ermessen der Schule.

Ein weiteres Merkmal, das dem Paradigmenwechsel zugeordnet werden kann, betrifft die Formulierungen der Bildungs- und Lehraufgaben. Diese sollen exemplarisch am Beispiel des Kompetenzbereichs „Elektronik und Elektrotechnik für IT“, der ebenfalls dem Kompetenzfeld Systemtechnik angehört, vorgestellt werden:

Bildungs- und Lehraufgabe:

Die Studierenden

- können Schaltungen mit Gleichspannungsquellen und Wechselspannungsquellen analysieren und berechnen;
- können das Betriebsverhalten von Analog-Digital- sowie Digital-Analog-Wandlern erklären und geeignete Typen auswählen;
- ...

Die Formulierung „Die Studierenden können...“ fordert eine bestimmte Handlungskompetenz ein. Vergleichbare Formulierungen werden jedoch auch bei der Vorgängerversion des Lehrplans - wenn auch nur teilweise - verwendet. Weitere Merkmale lassen sich nicht identifizieren.

Es bleibt anzumerken, dass der Lehrplan - abgesehen von Überschriften - ein überschaubares Ausmaß an „kompetenzorientierter Substanz“ in sich birgt. Hinweise zur Implementierung der Fachrichtungsstandards (siehe Kapitel 3) werden im Lehrplan ebenfalls nicht angeführt. Unter Umständen ist die fehlende Verzahnung von Lehrplan und Bildungsstandards der Tatsache geschuldet, dass Bildungsstandards somit in kürzeren Intervallen unabhängig vom Lehrplan adaptiert werden können (vgl. Kapitel 3.1, Ziele und Aufgaben).

Nachfolgend werden die Lern- und Bildungsziele des Kompetenzbereichs INIT im Detail erläutert sowie die allgemeinen Bildungsziele und didaktischen Grundsätze überblicksartig vorgestellt. Den Abschluss dieses Abschnitts stellt die Erhebung des Ist-Standes beim Projekt Bildungsstandards dar.

5.1.1 Ziele und Inhalte des Kompetenzbereichs Industrielle Informationstechnik

Die INIT ist, wie zuvor bereits festgehalten, im 4. und 5. Jahrgang zu unterrichten. Der Lehrplan hält bezüglich der Bildungs- und Lehraufgaben sowie für den Lehrstoff folgende Ziele beziehungsweise Inhalte fest:

Bildungs- und Lehraufgabe:

Die Studierenden

- kennen den grundlegenden Aufbau und die Funktionsweise von SPS-Systemen und Mikrocontrollersystemen und können diese zur Lösung technischer Aufgaben in typischen Anwendungen der industriellen Informationstechnik einsetzen;
- kennen den Aufbau typischer industrieller Bussysteme und können die darin eingesetzten Technologien und Übertragungsverfahren einsetzen;
- können die zur Prozessdatenverarbeitung und Prozessvisualisierung in industriellen Prozessen erforderliche IT-Infrastruktur planen und handhaben sowie deren Dokumentation und Überwachung durchführen;
- können ihre Kenntnisse über SPS- und Mikrocontrollertechnik erfolgreich zur Realisierung auch netzwerk- und echtzeitfähiger Systeme im industriellen Umfeld einsetzen sowie geeignete Mechanismen zur Prozesskommunikation in solchen Systemen implementieren.

Lehrstoff:

4. Jahrgang:

Aufbau und Funktionsweise von Systemen der SPS- und Mikrocontrollertechnik, Entwicklung typischer Anwendungen; Industrielle Feldbussysteme; Prozessdatenverarbeitung; Prozessvisualisierung, Prozesskommunikation.

5. Jahrgang:

Vertiefung der SPS- und Mikrocontrollertechnik, Entwicklung und Implementierung von Systemen für spezifische Anforderungen.

Es wird deutlich: Im 4. Jahrgang gilt es die Grundlagen zu legen - die Lernenden sollen die angeführten Themenbereiche im Überblick erfahren. Im darauf folgenden Jahr erfolgt eine thematische Vertiefung mit dem Schwerpunkt SPS- und Mikrocontrollertechnik-basierte Applikationsentwicklung.

5.1.2 Allgemeine Bildungsziele

Die allgemeinbildenden Kompetenzbereiche sowie die allgemeinen Bildungsziele sind in der Anlage [63] des Lehrplans angeführt. In dieser ist festgehalten, dass gemäß dem doppelten Bildungsauftrag einerseits die Erlangung der Hochschulreife sowie andererseits die Befähigung zur Ausübung eines gehobenen Berufes die zentralen Ziele der Ausbildung sind (vgl. Kapitel 2.3). In diesem Zusammenhang sollen entsprechende Fach-, Methoden-, Fremdsprachen- sowie soziale und personale Kompetenzen im Rahmen der Pflichtgegenstände - zu denen auch die technischen Kompetenzfelder zählen - erlangt werden.

5.1.3 Didaktische Grundsätze

Neben den allgemeinen Bildungszielen sind auch didaktische Grundsätze in der Anlage [63] des Lehrplans niedergeschrieben. Vorgegeben ist, dass sich die Unterrichtsplanung an den allgemeinen Bildungszielen (siehe Kapitel 5.1.2) sowie an den Bildungs- und Lehraufgaben - im Kontext dieser Arbeit sind das die Bildungs- und Lehraufgaben des Kompetenzbereichs INIT (siehe Kapitel 5.1.1) - zu orientieren hat. Ebenso gilt es den jeweiligen Stand der Technik oder relevante Entwicklungen im Bereich der Wirtschaft und Gesellschaft bei der Planung zu berücksichtigen.

Die angeführten Bildungs- und Lehraufgaben sind schulstufenübergreifend zu vernetzen; wobei der Lehrkraft bei der Konkretisierung einzelner Teilkompetenzen Spielraum eingeräumt wird. Wie bei der Analyse der Bildungs- und Lehraufgaben des Kompetenzbereichs INIT bereits festgehalten, gilt es, wenn Themenbereiche beispielsweise über zwei Jahrgänge zu behandeln sind, im ersten Jahr die Grundlagen zu legen. Im darauf folgenden Jahr erfolgt dann die thematische Vertiefung durch Zunahme der Komplexität und des Schwierigkeitsgrads. Generell gilt, dass bei der Umsetzung der Bildungs- und Lehraufgaben einer grundlegenden Vertiefung gegenüber oberflächlicher Vielfalt Vorzug zu geben ist.

Zur Förderung der Motivation ist ein praxisnaher Unterricht das Mittel der Wahl, welcher seine Grundlage in realen Problemstellungen findet (Stichwort Problem-based Learning, vgl. Kapitel 4.4.2). Überhaupt ist der Erwerb von Handlungskompetenzen - ob zum Lösen konkreter Probleme und/oder zum selbstverantwortlichen Lernen - ein erklärtes Ziel.

Dementsprechend sind die Unterrichtsmethoden zu wählen. In diesem Zusammenhang werden beispielsweise der Projektunterricht oder Simulationen erwähnt.

Ein zentrales Anliegen ist die individuelle Förderung der Lernenden. Es wird explizit darauf hingewiesen, dass die Möglichkeiten individueller Fördermaßnahmen und differenzierten Unterrichts auszuschöpfen sind. Den Lernenden soll auch die Möglichkeit zur Reflexion geboten werden, indem entsprechende Methoden wie zum Beispiel Portfolios eingesetzt werden.

Besonderes Augenmerk hat auch in der Entwicklung sozialer und personaler Kompetenzen zu liegen. Demnach ist die Lehrkraft dazu angehalten, den (individuellen) Entwicklungsprozess zu begleiten und vor allem zu fördern.

5.1.4 Bildungsstandards und prototypische Beispiele

In Kapitel 3 werden die Ziele, die Aufgaben sowie die Funktionsweise von Bildungsstandards und der damit verbundenen Kompetenzorientierung allgemein erläutert. Das Ziel dieses Abschnitts ist die Diskussion der gegenwärtig aktuellen Version der Bildungsstandards für den Fachbereich Informationstechnologie (IT). Im Fokus steht ausschließlich der für diese Arbeit relevante Kompetenzbereich INIT. Als Grundlage dient die Bildungsstandard-Version „September 2011“ [29].

Die Tabellen 5.1 und 5.2 beschreiben die geforderten Kompetenzen des Kompetenzbereichs INIT mittels Deskriptoren (vgl. Kapitel 3.4). Die Handlungsebenen „Wiedergeben/Verstehen“ und „Anwenden“ sind in der Tabelle 5.1 enthalten. Ergänzend dazu sind die in Kapitel 5.1.1 dargestellten Lehrinhalte des Kompetenzbereichs nochmals angeführt. Tabelle 5.2 zeigt die Anforderungen der Kompetenzebenen „Analysieren“ und „Entwickeln“.

Tabelle 5.1: Deskriptoren für den Kompetenzbereich Industrielle Informationstechnik [29]

Inhalte lt. Lehrplan	Wiedergeben/Verstehen	Anwenden
	Die Schülerinnen und Schüler	Die Schülerinnen und Schüler
- Aufbau und Funktionsweise von Systemen der SPS- und Mikrocontrollertechnik	...kennen den grundlegenden Aufbau und die Funktionsweise von SPS-Systemen und Mikrocontrollersystemen.	...können SPS-Systeme und Mikrocontrollersysteme zur Lösung technischer Aufgaben in typischen Anwendungen der industriellen Informationstechnik einsetzen.
- Entwicklung typischer Anwendungen	...kennen den Aufbau typischer industrieller Bussysteme sowie die darin eingesetzten Technologien und Übertragungsverfahren.	...können die in industriellen Bussystemen eingesetzten Technologien und Übertragungsverfahren einsetzen.
- Industrielle Feldbussysteme		
- Prozessdatenverarbeitung, Prozessvisualisierung, Prozesskommunikation		
- Vertiefung der SPS- und Mikrocontrollertechnik		
- Entwicklung und Implementierung von Systemen für spezifische Anforderungen		

Eine eindeutige Zuordnung der Kompetenzen zu einem Jahrgang findet weder im Lehrplan noch bei den Bildungsstandards statt. Daraus lässt sich schließen, dass die Planung hinsicht-

lich der zeitlichen Abfolge der Kompetenzentwicklung in der Verantwortung der Lehrkraft liegt. Selbstverständlich mit dem Ziel, dass die Lernenden am Ende des Bildungsganges über die definierten Kompetenzen verfügen. Nachdem der Kompetenzbereich INIT im 4. und 5. Jahrgang Bestandteil der Ausbildung ist, stehen hierfür grundsätzlich vier Semester zur Verfügung. Augenscheinlich ist, dass die vom Projektteam „Bildungsstandards“ entworfene und in Kapitel 3.4 vorgestellte Notation zur Bezeichnung der Deskriptoren bisher (noch) keine Berücksichtigung findet.

Tabelle 5.2: Deskriptoren für den Kompetenzbereich Industrielle Informationstechnik (Fortsetzung) [29]

Inhalte lt. Lehrplan	Analysieren	Entwickeln
	Die Schülerinnen und Schüler	Die Schülerinnen und Schüler
Siehe Tabelle 5.1	...können die zur Prozessdatenverarbeitung und Prozessvisualisierung in industriellen Prozessen erforderlichen IT-Infrastrukturen planen und handhaben sowie deren Dokumentation und Überwachung durchführen.	...können SPS- und Mikrocontrollertechnik zur Entwicklung netzwerk- und echtzeitfähiger Systeme im industriellen Umfeld einsetzen und geeignete Mechanismen zur Prozesskommunikation in solchen Systemen implementieren.

Um die Dimension der unterschiedlichen Kompetenzniveaus zu verdeutlichen, dienen sogenannte „Prototypische Unterrichtsbeispiele“ (kurz Beispiele, vgl. Kapitel 3.4). Für den Kompetenzbereich INIT ist allerdings nur ein einziges Beispiel in [29] verfügbar, das sich wie folgt gestaltet:

PC und SPS-Visualisierung einer Produktionsanlage	
Kompetenzfeld:	Systemtechnik
Kompetenzbereich:	Industrielle Informationstechnik
Handlungskompetenz:	E (Entwickeln)
Relevante(r) Deskriptor(en):	Ich kann geeignete Mechanismen zur Prozesskommunikation in SPS- und Mikrocontrollersystemen implementieren.
Themenbereich(e) und Fertigkeit(en):	Industrielle Informationstechnik Prozesskommunikation
Methodisch-didaktische Hinweise:	Einzelarbeit am PC
Hilfsmittel:	Unterrichtsunterlagen
Quelle:	-
Zeitbedarf:	120 Min.

Aufgabenstellung:

In einer Produktionsanlage werden von einer SPS laufend Messwerte erfasst. Einer dieser erfassten Messwerte soll auf einem PC in einer grafischen Oberfläche angezeigt und zyklisch jede halbe Sekunde aktualisiert werden. Für diesen Messwert soll über die grafische Oberfläche auch die Vorgabe eines Sollwerts zur Steuerung der Anlage über die SPS möglich sein. Die Realisierung der Visualisierungsoberfläche soll mit Standard Software erfolgen.

Lösungsvorschlag:

Die Visualisierungsoberfläche wird auf einem Windows-PC erstellt. Für die Kommunikation zwischen dem PC und der SPS wird OPC-Technik (OLE for Process Control) eingesetzt. Der dafür nötige OPC-Server wird üblicherweise vom Hersteller der SPS mitgeliefert und stellt die von der SPS ermittelten Werte über das OPC-Protokoll zur Verfügung. Die Kommunikation ist dabei bidirektional, womit auch die Vorgabe von Sollwerten möglich ist. Eine Realisierungsmöglichkeit des OPC-Clients mit Standardsoftware bietet das Microsoft Office Paket. Die Oberfläche wird in MS-Excel erstellt, die nötige Programmierung erfolgt über die integrierte Programmiersprache Visual Basic for Applications (VBA). Damit über VBA auf den OPC-Server zugegriffen werden kann, muss die entsprechende OPC-Library des SPS-Herstellers eingebunden werden.

Das Beispiel kann dem Lehrstoffbereich „Prozessvisualisierung“ zugeordnet werden und zielt im Detail auf den einzigen Handlungsdeskriptor der Ebene „Entwickeln“ ab. Die Durchführung soll in Form einer Einzelarbeit am PC erfolgen. Ein Lösungsvorschlag ist ebenfalls angeführt (Details, wie z.B. Screenshot der grafischen Oberfläche oder Quellcode, siehe [29]). Für die weiteren Lehrstoffbereiche des Kapitels 5.1.1 sind, wie bereits festgehalten, (noch) keine Beispiele vorhanden.

5.2 HTBL Krems - Abteilung für Informationstechnologie

Nachdem das Kapitel 5.1 den Lehrplan der Fachrichtung Informationstechnologie allgemein behandelt, widmet sich dieser Abschnitt der schulspezifischen Umsetzung des Lehrplans an der HTL Krems [8]. Einleitend werden die HTL Krems im Allgemeinen und die Abteilung Informationstechnologie im Speziellen vorgestellt. Darauf aufbauend wird die Implementierung des Kompetenzbereichs INIT aufgezeigt, um thematische Anknüpfungspunkte hinsichtlich eines interdisziplinären Unterrichts zu identifizieren. Der letzte Abschnitt widmet sich der Konkretisierung der Erwartungen an die Vorbildung der Lernenden.

Die HTL Krems verfügt gegenwärtig über Bildungsgänge in den Fachbereichen Bautechnik und IT. Der Fachbereich Bautechnik setzt sich einerseits aus den höheren Abteilungen für Hoch- und Tiefbau sowie Sanierungstechnik und andererseits aus der Fachschule für Bautechnik zusammen. Ergänzend zu den höheren und dem mittleren Bildungsgängen/-gang wird ein Bautechnik-Kolleg angeboten. Der Fachbereich IT besteht einzig und allein aus der

gleichnamigen höheren Abteilung. Eine IT-Fachschule befindet sich nicht im Bildungsangebot der HTL Krems. Wie der Abbildung 5.2 zu entnehmen ist, verfügt die HTL Krems, Abteilung IT, über eine Expositur in Zwettl (NÖ). Somit kann festgehalten werden, dass die IT-Abteilung über zwei sogenannte Züge an unterschiedlichen Standorten verfügt.

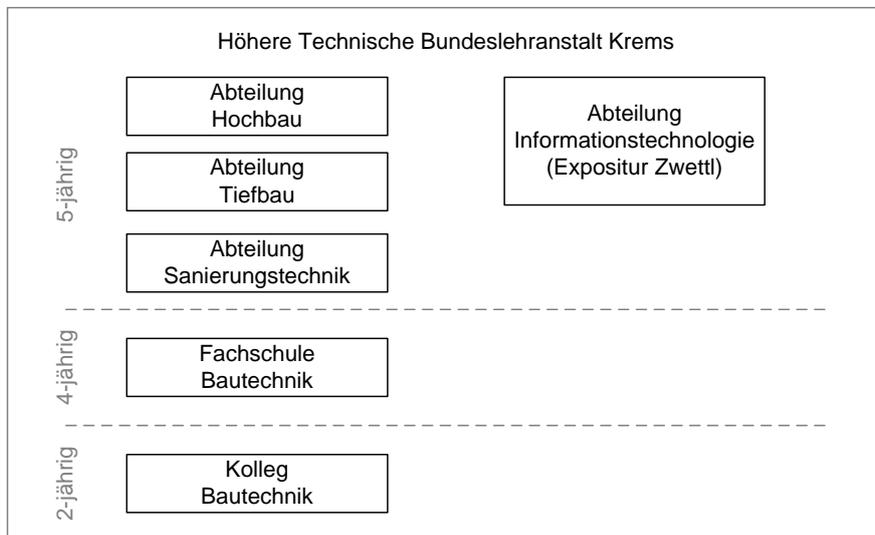


Abbildung 5.2: HTBL-Krems im Überblick

Insgesamt betrachtet, besuchen rund 600 SchülerInnen die HTL Krems; wobei gut ein Drittel der SchülerInnen der Abteilung IT zuzurechnen ist. Um die Voraussetzungen für einen möglichst ansprechenden Unterricht zu schaffen, verfügt die HTL Krems über modern ausgestattete Klassenzimmer und Laboratorien. Beamer, digitale Overhead-Projektoren und Touchscreens gehören zur Standardausstattung der Räumlichkeiten.

5.2.1 Implementierung der Industriellen Informationstechnik

Schulautonome Lehrplanbestimmungen ermöglichen es, den Lehrplan hinsichtlich der Stundenverteilung geringfügig abzuändern. Die Grundlage hierfür bildet die Anlage [63] des Lehrplans. Diese hält in Absatz II fest:

[...] Die Nutzung dieser Freiräume hat auf der Grundlage eines Konzeptes zu erfolgen. Das Konzept hat die Anforderungen des regionalen Umfelds, insbesondere aber die Erfordernisse des Arbeitsmarktes im Bereich der gehobenen Berufe auf technischen, gewerblichen und kunstgewerblichen Gebiet, die Bedürfnisse der Schülerinnen und Schüler, der Schulpartner insgesamt sowie die personellen und materiellen Möglichkeiten des Schulstandortes zu berücksichtigen.

In Anbetracht dieser zugesprochenen Freiräume wurde die Implementierung des Kompetenzbereichs INIT wie folgt festgelegt: Wie der Tabelle 5.3 zu entnehmen ist, wird der Kompetenzbereich im 4. Jahrgang unterrichtet; wobei für die Fachtheorie eine und für die Übung zwei Wochenstunden vorgesehen sind. Gleiches gilt für den 5. Jahrgang. Nachdem dieser Kompetenzbereich im 5. Jahrgang Berücksichtigung findet, ist dieser Gegenstand der Reife- und Diplomprüfung.

Tabelle 5.3: Implementierung des Kompetenzbereichs Industrielle Informationstechnik an der HTL Krems

	Wochenstunden Fachtheorie	Wochenstunden Übung
4. Jahrgang	1	2
5. Jahrgang	1	2

Es wird deutlich, dass der Kompetenzbereich INIT eine durchaus bedeutende Position in der IT-Ausbildung der HTL Krems einnimmt. Dies ist in Anbetracht der Vielschichtigkeit, der Komplexität aber auch der allgemeinen Bedeutung dieses Themenbereichs nachvollziehbar.

5.2.2 Interdisziplinärer Unterricht - Potenziale und Möglichkeiten

Die vorhergehenden Abschnitte dieses Kapitels 5.2 charakterisieren den Kompetenzbereich INIT allgemein und zeigen die schulspezifische Implementierung der HTL Krems auf. In diesem Abschnitt wird dessen Potenzial hinsichtlich eines interdisziplinären Unterrichts untersucht. Grundlage hierfür bilden die Erkenntnisse der Abschnitte 4.2 und 4.3.1.

Interdisziplinärer Unterricht lässt sich im Wesentlichen dadurch beschreiben, dass zur Bearbeitung komplexer Problemstellungen oftmals Beiträge aus mehreren Unterrichtsdisziplinen erforderlich sind [64]. Das bedeutet, die Bearbeitung einer solchen Problemstellung muss arbeitsteilig - in den betroffenen Unterrichtsdisziplinen - erfolgen. Folglich kommen nur jene Kompetenzbereiche in Frage, die wie die INIT entweder im 4. oder 5. Jahrgang angesiedelt sind. Wie der Tabelle 5.4 zu entnehmen ist, sind das grundsätzlich mehrere. Der Übersicht halber wird die Darstellung der ersten drei Jahrgänge ausgespart. Ist somit bei einem Kompetenzbereich keine Markierung in der Spalte des 4. oder 5. Jahrgangs gesetzt, bedeutet das, dass dieser Kompetenzbereich in den unteren Jahrgängen eins, zwei und/oder drei Bestandteil der Ausbildung ist. Für diesbezügliche Detailinformationen wird auf [61] verwiesen.

Tabelle 5.4: Kompetenzfelder und -bereiche der Fachtheorie und -praxis im Überblick

Kompetenzfeld	Kompetenzbereich	Jahrgang	
		4.	5.
Softwareentwicklung	Strukturierte Programmierung		
	Algorithmen und Datenstrukturen		
	Objektorientierte Programmierung		
	Softwareentwicklungsprozess	x	x
	Schnittstellen und Bibliotheken	x	x
	Anwendungsprogrammierung	x	x
Informationstechnische Projekte	Grundlagen des Betriebs		
	Organisation		
	Betriebliche Ziele		
	Projektmanagement	x	x
	Qualitätsmanagement		x
	Durchführung informationstechnischer Projekte	x	x
Informationssysteme	Eigenschaften und Architekturen von Datenbanksystemen		x
	Datenmodelle	x	
	Abfragesprachen	x	
	Datenbankanwendungen	x	
	Administration von Datenbanksystemen	x	x
	Informationssysteme und Contentmanagement	x	x
	Informationsmanagement	x	x
Systemtechnik	Elektrotechnik und Elektronik für Informationstechnologie		
	Grundlagen der Informatik		
	Betriebssysteme		
	Industrielle Informationstechnik	x	x
	Systemintegration und Infrastruktur	x	x
	Dezentrale Systeme	x	x
Medientechnik	Informationsdarstellung im Internet		
	Internet- und Multimedia-Anwendungen		
	Grundlagen der akustischen und visuellen Wahrnehmung		
	Komprimierungsverfahren		
	2D-Animation		
	Medienbearbeitung		
	Multimediahardware		
	Grafische Benutzerschnittstellen und Userinterfacedesign		
Netzwerktechnik	Übertragungsmedien und Netztopologien		
	Schichtenmodelle und Protokolle		
	Netzwerkmanagement	x	
	Switchung und Routing		
	Netzwerksicherheit	x	
Computerpraktikum			

Neben dem Kriterium „Jahrgang“ stellen thematische Berührungspunkte ein weiteres Muss-Kriterium dar. Um diese zu formulieren, werden im ersten Schritt Cluster mit kompetenzbereichsspezifischen Merkmalen der INIT gebildet (siehe Abbildung 5.3). Anhand dieser Merkmale werden dann im zweiten Schritt die Inhalte aller in Frage kommenden Kompetenzbereiche der Tabelle 5.4 untersucht mit dem Ziel, potenzielle Kandidaten für einen interdisziplinären Unterricht zu identifizieren. Als Grundlage für die Analyse dienen zum Einen die Bildungs- und Lehraufgaben [61] sowie zum Anderen die Bildungsstandards [29] der in Frage kommenden Kompetenzbereiche.



Abbildung 5.3: Identifikationskriterien für die Interdisziplinarität

Kapitel 5.2.1 hält fest, dass der Kompetenzbereich INIT ab dem 4. Jahrgang unterrichtet wird. Das bedeutet, die Lernenden kommen zu diesem Zeitpunkt erstmals mit grundlegenden Themen dieses Kompetenzbereichs in Berührung. Das Ziel des interdisziplinären Unterrichts ist jedoch, komplexe Problemstellungen fächerübergreifend zu behandeln, um eben vernetztes Denken zu fördern. Damit Lernende beim Erstkontakt mit der Thematik keine motivationalen Störungen erfahren, indem sie mit komplexen, fächerübergreifenden Aufgabenstellungen konfrontiert und womöglich überfordert werden, ist hinsichtlich der Umsetzung des interdisziplinären Unterrichts umsichtiges Vorgehen erforderlich.

Szenario 1

Potenzial für interdisziplinären Unterricht bringen die Kompetenzbereiche „Projektmanagement“ (PM) und „Qualitätsmanagement“ (QM) im Allgemeinen und „Durchführung informationstechnischer Projekte“ (DiP) im Speziellen mit sich. Wobei es zu berücksichtigen gilt, dass QM ausschließlich im 5. Jahrgang Thema der Ausbildung ist. PM und DiP werden hingegen ebenfalls im 4. und 5. Jahrgang unterrichtet. Dadurch können die Techniken und Methoden des PM und des QM anhand realer Projekte aus der Domäne Industrieautomation oder Gebäudetechnik praktisch erprobt werden. Die eigentliche Umsetzung des Projektes kann selbstständig im Kompetenzbereich DiP erfolgen. Die Kapazitäten des „Projekt Hosting“-Kompetenzbereichs INIT stehen dann für fachspezifische Diskussionen, punktuelle Wissensergänzungen etc. zur Verfügung. Als Durchführungszeitpunkt würde sich der 5. Jahrgang anbieten - nicht zuletzt wegen der Bildungs- und Lehraufgaben des Kompetenzbereichs INIT, die die „Entwicklung und Implementierung von Systemen für spezifische Anforderungen“ im 5. Jahrgang einfordern. Außerdem sollten die Lernenden des 5. Jahrgangs bereits

imstande sein, größere anspruchsvolle Projekte selbstständig zu realisieren. Dadurch können auch erste Erfahrungen gesammelt werden, die jenen aus realen Projekten durchaus nahe kommen. Ferner bleibt anzumerken, dass die „Projektarbeit“ (vgl. Kapitel 4.4.2) - auch wenn diese im Team bewältigt wird - Aspekte der Individualisierung adressiert. Herausfordernd ist der Umstand, dass vier Kompetenzbereiche von diesem Szenario betroffen wären, was erheblichen Organisationsaufwand erfordert - vor allem bei vier verschiedenen Lehrkräften. Alternativ könnte man die Interdisziplinarität auf drei Disziplinen reduzieren, indem man ausschließlich die Werkzeuge aus PM und QM anwendet. Die praktische Umsetzung erfolgt dann in den Übungseinheiten der INIT. Alles in allem würde dieses Szenario einen anderen - praxisnahen - Blickwinkel auf das Projekt- wie auch Qualitätsmanagement ermöglichen.

Szenario 2

Wie im Szenario 1 festgehalten, ist die „Entwicklung und Implementierung von Systemen für spezifische Anforderungen“ zentrales Thema der INIT im 5. Jahrgang. Nachdem es ein übergeordnetes Ziel der Ausbildung ist, den Lernenden ein allgemeines Qualitätsbewusstsein als auch Methodenkompetenz zu vermitteln, gilt das auch für den Kompetenzbereich IN-IT. Schlussendlich sollen die Lernenden befähigt sein, ansprechende „Embedded Software“-Systemen zu entwickeln. Dass diesbezüglicher Handlungsbedarf besteht, zeigt Grechenig et al. in [65] auf:

[...] Faktum ist, dass mehr als 65% der real im Einsatz befindlichen Software aus der Sicht des qualifizierten Ingenieurs heute in einem „erbärmlichen“ Zustand sind. Diese Software ist schwer zu ändern. Sie ist technisch schlecht dokumentiert. Die Entwicklungsdokumente - sofern noch vorhanden - sind nicht konsistent oder unvollständig.

Unter diesem Gesichtspunkt - auch wenn nicht eindeutig hervorgeht, inwieweit das auch Embedded Software-Systeme inkludiert - erscheint ein interdisziplinärer Unterricht, der zur Förderung des entsprechenden Bewusstseins beiträgt, nur von Vorteil. Der hierfür relevante Kompetenzbereich ist „Softwareentwicklungsprozess“ (SEP), welcher vom 1. bis zum 5. Jahrgang Bestandteil des Bildungsganges ist. SEP konzentriert sich einerseits auf Vorgehensmodelle, Entwicklungsmethoden und Entwurfsmuster im Bereich der Softwaretechnik sowie andererseits auf das Testen und die methodische Fehlersuche bei Softwaresystemen. Das Thema Versionsverwaltung ist ebenfalls Inhalt von SEP. Hierbei handelt es sich um Lehrinhalte, deren Behandlung nicht lösgelöst von der INIT erfolgen darf. Als Grundlage kann wiederum eine Projektarbeit aus dem Bereich Industrieautomation oder Gebäudetechnik dienen. In den SEP-Einheiten können die verschiedenen Vorgehensmodelle am Beispiel von Embedded Systems-Softwareprojekten behandelt und deren praktische Implementierung diskutiert werden. Die praktische Umsetzung kann dann, wie beim vorhergehenden Szenario auch, entweder in DiP oder in den Übungseinheiten der INIT erfolgen. Besonderes Augenmerk sollte auf das Testen und die Fehlersuche - insbesondere auf das Debuggen - im Kontext der Mikrocontrollertechnik gerichtet werden. Bei Embedded Software-Systemen gelten aufgrund der Integration externer Systemkomponenten (z.B. Sensoren, Taster etc.) und den damit verbundenen Abhängigkeiten bei der Programmausführung andere Anforderungen bei der

Fehlersuche als das beispielsweise bei der klassischen Web-Entwicklung der Fall ist - vor allem unter Berücksichtigung des Echtzeitaspekts. Die Behandlung dieser spezifischen Anforderungen bietet sich hervorragend für einen interdisziplinären Unterricht mit SEP an.

Szenario 3

Wie das vorhergehende Szenario zeigt, kommt den Systemschnittstellen - von einfacher Port-I/O bis hin zur Implementierung eines TCP/IP-Stacks oder der Anbindung eines Feldbus - im Bereich der Mikrocontrollertechnik eine wichtige Bedeutung zu. Diese müssen gemäß den entsprechenden Methoden definiert und dokumentiert werden. Das sind wiederum Themen, die im hohen Maße mit dem Kompetenzbereich „Schnittstellen und Bibliotheken „ (SuB) korrelieren. So könnte beispielsweise die Definition einer Systemschnittstelle in SuB erfolgen. Die praktische Umsetzung ist dann wiederum in den Übungseinheiten der INIT zu bewerkstelligen.

Szenario 4

Ergänzend ist noch festzuhalten, dass sich das Kompetenzfeld „Angewandte Mathematik“ ebenfalls für einen interdisziplinären Unterricht anbieten würde - beispielsweise um die Fehlerfortpflanzung bei Messaufgaben zu bewältigen. Stichwort Messen: Nachdem es im Bereich der Industrieautomation oder der Gebäudetechnik in der Regel eine physikalische Größe zu erfassen und in weiterer Folge in eine elektrische Größe umzuwandeln gilt, kann die Funktionsweise diverser Sensoren im Kompetenzbereich „Grundlegende physikalische Größen und ihre Messung“ behandelt werden.

Diskussion

Die hier vorgestellten Szenarien sind ausschließlich als Vorschlag zu verstehen. Selbstverständlich sind auch andere Kombinationsmöglichkeiten der identifizierten Kompetenzbereiche möglich. Ein Faktor, der jedoch nicht außer Acht gelassen werden sollte, ist, dass diese Art des Unterrichts eine persönliche Bereitschaft vonseiten der betroffenen Lehrkräfte erfordert. Schlussendlich erzwingt die Interdisziplinarität ein gewisses Maß an Transparenz im Unterrichtsgeschehen, was mitunter ein Problem darstellen könnte. Auch wenn der Lehrplan und die Bildungsstandards durchaus konkrete Erwartungen formulieren, sind die individuellen Vorstellungen der Lehrkräfte hinsichtlich inhaltlicher Schwerpunktsetzung, didaktischer Aufbereitung der interdisziplinären Inhalte sowie deren zeitliche Abhandlung in Einklang zu bringen. Dieser Mehraufwand erfordert eine gewisse Leistungsbereitschaft als auch Flexibilität vonseiten der handelnden Personen. Vor allem dann, wenn beispielsweise der interdisziplinäre Unterricht an verschiedenen Wochentagen abgehalten wird und ein Kompetenzbereich unter Umständen mehrmals hintereinander ausfällt. Damit dabei der Mehrwert dieser Unterrichtsform nicht verloren geht, ist eine intensive Kommunikation hinsichtlich inhaltlicher als auch terminlicher Unstimmigkeiten eine unabkömmliche Voraussetzung für das Gelingen.

5.2.3 Konkretisierung der Erwartungen an die Vorbildung

Der Kompetenzbereich INIT ist laut schulautonomer Anordnung in den Jahrgängen vier und fünf Bestandteil der Ausbildung (vgl. Kapitel 5.2.1). Betrachtet man die Lehrstoffdefinition des Lehrplans (siehe Kapitel 5.1.1), wird deutlich, dass umfassende Themenblöcke, wie zum Beispiel Mikrocontrollertechnik oder Feldebussysteme, zu behandeln sind. Die inhaltliche Bewältigung dieser Bereiche setzt entsprechendes Vorwissen voraus, das jedoch nicht ausdrücklich definiert ist. Es werden zwar mit den (wenigen) Deskriptoren der INIT die zu erwartenden Kompetenzen beschrieben, die es zu erwerben gilt, jedoch nicht jene, die zu Beginn der Ausbildung im Kompetenzbereich INIT Voraussetzung sind; obgleich davon auszugehen ist, dass die Inhalte des Lehrplans und der Bildungsstandards bestens aufeinander abgestimmt sind. Dennoch steht die Frage im Zentrum, welche Themenbereiche als zentraler Bestandteil - sozusagen als die unabdingbaren Grundlagen - der Vorbildung anzusehen sind?

Deren Formulierung erfolgt im ersten Schritt. Im zweiten Schritt wird analysiert, inwieweit die Bildungs- und Lehraufgaben aber auch die Deskriptoren der betroffenen Kompetenzbereiche die Lernenden auf die Herausforderungen der INIT vorbereiten. Schließlich steht zur Umsetzung der ambitionierten Lern- und Bildungsaufgaben nur begrenzt Zeit zur Verfügung - ein ausuferndes „Nachlernen von Grundlagen“ würde die Zielerreichung wesentlich erschweren.

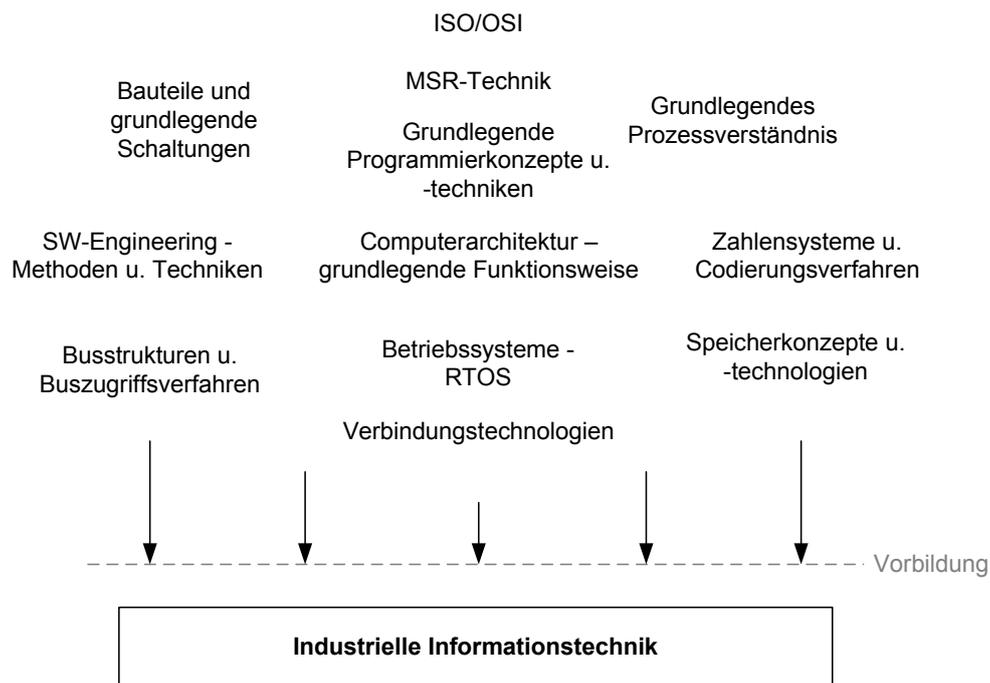


Abbildung 5.4: Vorwissen - relevante Themenbereiche für die Industrielle Informationstechnik

Anhand von Schlagwörtern stellt die Abbildung 5.4 Themen dar, deren Kenntnis als Anforderung an die Vorbildung angesehen werden kann. Selbstverständlich stellt diese Darstellung keinen Anspruch auf Vollständigkeit, es sollen ausschließlich die wichtigsten Themen aufgezeigt werden. Ähnlich verhält es sich mit dem Kompetenzniveau im jeweiligen Thema: Ausgeprägte Handlungskompetenzen sind beispielsweise bei Zahlensystemen unabdingbar. Registerzuweisungen in der Mikrocontrollertechnik erfolgen in der Regel binär oder

hexadezimal durch Bitoperationen. Folglich ist die Anwendung dieser Zahlensysteme eine elementare Voraussetzung und muss daher Bestandteil der Vorbildung sein. Nachdem diese Inhalte im Kompetenzbereich „Grundlagen der Informatik“ (kurz GINF, 1. Jahrgang) vermittelt werden, können diesbezügliche Kompetenzen vorausgesetzt werden. Gleiches gilt für die Zeichenkodierung sowie für die BOOLE'sche Algebra, da diese ebenfalls in GINF in der erforderlichen Tiefe unterrichtet werden. Hardwarenahe Softwareentwicklung erfordert ein Grundverständnis für die Arbeitsweise und den Aufbau eines Mikrocontrollers sowie der (integrierten) Speichertechnologien. Nachdem entsprechende Inhalte im Kompetenzbereich GINF teilweise behandelt werden, kann von einem (Teil-)Vorwissen ausgegangen werden.

Ähnliches gilt für sogenannte *Real Time Operating Systems* (RTOS). Der Kompetenzbereich „Betriebssysteme“ behandelt zwar „Konzepte moderner Betriebssysteme“ oder „Beurteilung der Tauglichkeit von Betriebssystemen für verschiedene Einsatzgebiete“, aber inwieweit hier RTOS inkludiert sind, geht nicht hervor. Demnach müssten RTOS - insbesondere jene Themenbereiche, die sich aufgrund der Anforderungen des Echtzeitverhaltens von herkömmlichen Betriebssystemen unterscheiden (z.B. die Bedeutung von Mutex oder Semaphore bei der Synchronisation) - im Kompetenzbereich INIT behandelt werden. Zielführender wäre hingegen, diese im jeweiligen Fachbereich - zumindest auf theoretischer Ebene - zu behandeln. In der Praxis bedeutet das, dass die Schnittstelle der Kompetenzbereiche Betriebssysteme und INIT zwischen den Lehrenden zu definieren ist - die Frage „Wer entwickelt welche Detailkompetenzen in welchem Ausmaß?“ ist zu klären.

Hardwarenahe Softwareentwicklung erfordert den Einsatz einer höheren Programmiersprache wie beispielsweise „C“. Nachdem im Lehrplan keine konkreten SW-Technologien angeführt sind, gilt es diesbezüglich schulinterne Abstimmungen zu treffen. Gespräche mit dem Abteilungsvorstand¹ haben ergeben, dass im Rahmen des Kompetenzfeldes „Softwareentwicklung“ - also in jenen Kompetenzbereichen, die sich der Programmierausbildung widmen - die Programmiersprache „C-Sharp“ (C#) zum Einsatz kommt. Im Rahmen der Medientechnikausbildung sind die Lernenden beispielsweise mit der Scriptsprache „PHP“ konfrontiert, die wie „C#“ stark an „C“ angelehnt ist. Demnach ist der Schritt zu „(Embedded-)C“ ein bewältigbarer, da die Syntax der angeführten Technologien große Ähnlichkeiten gegenüber „C“ aufweist - auch wenn diesen unterschiedliche Programmierparadigmen zugrunde liegen. Vor allem unter dem Gesichtspunkt, dass der Kompetenzbereich „Strukturierte Programmierung“ grundlegende Programmiertechniken (technologieneutral) behandelt. Somit sind gute Voraussetzungen für einen direkten Einstieg in die Programmierung von Mikrocontroller-Systemen gegeben. Aufgrund des zuvor angeführten Technologieeinsatzes sollte die Bedeutung wesentlicher Werkzeuge (z.B. „Compiler“) als auch Begriffe der Softwareentwicklung allgemein bekannt sein.

Hinsichtlich formaler Dokumentationsmethoden des Software-Engineering sollte zu Beginn des 4. Jahrgangs die Kenntnis ausgewählter (UML-)Verhaltens- und Strukturdiagramme (z.B. Zustands- oder Ablauf-Diagramm) zumindest bekannt sein. Diesbezügliche Lehrinhalte sind jedoch weder im Lehrplan noch in den Bildungsstandard explizit ausgewiesen. Obwohl davon auszugehen ist, dass dieses Thema im Kompetenzbereich „Strukturierte Programmierung“ Berücksichtigung findet, sollte dies in Gesprächen mit der jeweiligen Lehrkraft sichergestellt werden. Grundlegende Begriffe des SW-Entwicklungsprozess (z.B. „Analyse“,

¹Prof. Dipl.-Ing. Anton Hauleitner, Vorstand der Abteilung Informationstechnologie

„Design“, „Implementierung“ und „Test“) sollten ebenfalls bekannt sein. Konkrete Vorgehensmodelle werden hingegen erst ab dem 4. Jahrgang vermittelt (siehe Kapitel 5.2.1 und Kapitel 5.2.2).

Einen weiteren wichtigen Bereich bei der Industrieautomation beziehungsweise Prozessdatenverarbeitung stellt die Mess-, Steuer und Regeltechnik (MSR) dar. Obwohl die Fachbereiche Elektrotechnik und Elektronik Bestandteil der Ausbildung sind (vgl. Kompetenzbereich „Elektrotechnik und Elektronik für Informationstechnologie“), ist nur von geringen Vorkenntnissen im Bereich der MSR auszugehen. Diese umfassen jedoch das Betriebsverhalten eines Analog-Digital- sowie Digital-Analog-Wandlers. Weiters kann einfaches Basiswissen bezüglich der Funktionsweise und Anwendung von Bauelementen (Widerstand, Kondensator, Spule, Diode, Bipolarer Transistor und ausgewählte Integrierten Schaltungen (IC)) vorausgesetzt werden. Die Kenntnis wichtiger Grundschaltungen zur Messung elektrischer und nichtelektrischer Größen sollte ebenfalls gegeben sein. Ähnliches gilt für die Digitaltechnik: Grundbegriffe, logische Verknüpfungen, Schaltungen logischer Glieder sowie die Schaltungsanalyse und -synthese sollten bekannt sein. Nachdem die Ausbildung im Bereich der Elektrotechnik und Elektronik praktische Elemente im Labor inkludiert (1. und 2. Jahrgang, jeweils 2 Wochenstunden), kann von Kompetenzen im Bereich der Handhabung von Bauteilen, Werkzeugen und Materialien ausgegangen werden, die beim Schaltungsaufbau erforderlich sind.

Netzwerktechnik-Themen (siehe Abbildung 5.4, ISO/OSI, Busstrukturen und -zugriffsverfahren, Verbindungstechnologien) werden in der erforderlichen Tiefe im Kompetenzfeld „Netzwerktechnik“ erörtert. Nachdem Feldbusse im Kompetenzbereich INIT vermittelt werden, ist wiederum eine Detailabstimmung mit den handelnden Personen des Kompetenzfelds „Netzwerktechnik“ zielführend. Beispielsweise werden Busstrukturen und -zugriffsverfahren in diesem Kompetenzfeld (1. bis 3. Jahrgang) behandelt; wodurch die Bedeutung dieser Themen in anderen Kompetenzbereichen - wie eben der INIT - bereits hervorgehoben und die elementaren Grundlagen unter Berücksichtigung der Anforderungen der industriellen Informationstechnik vermittelt werden sollten.

Zusammengefasst zeigt sich, dass der Kompetenzbereich INIT überaus vielschichtig ist - viele Themenbereiche der ersten Jahrgänge sind integraler Bestandteil dieses Kompetenzbereichs. An dieser Stelle gilt es ergänzend zum Kapitel 5.2.2 festzuhalten: Obwohl ein interdisziplinärer Unterricht mit dem Kompetenzbereich „Elektrotechnik und Elektronik für Informationstechnologie“ aufgrund zeitlicher Inkongruenz nicht möglich ist (vgl. Tabelle 5.4), würde sich eine zeitlich versetzte Form der fächerübergreifenden Zusammenarbeit anbieten. Begründung hierfür ist, dass, wie bereits erwähnt, im Kompetenzbereich „Elektrotechnik und Elektronik für Informationstechnologie“ einfache Schaltungen dimensioniert und aufgebaut werden. Somit liegt der Schluss nahe, dass dies in Abstimmung mit dem Kompetenzbereich INIT erfolgt. Beispielsweise könnten Schaltungen mit einem Mikrocontroller, der bereits mit dem entsprechenden Programm versehen ist, und Leuchtdioden sowie Taster aufgebaut und getestet werden. Das hat die positive Konsequenz, dass einerseits die erforderlichen Elektronik-Übungen vollzogen und andererseits aber auch erste Erfahrungen mit einem Mikrocontroller gemacht werden (z.B. Datenblatt lesen und die relevanten Anschlüsse identifizieren). Geschieht dies mit dem Hinweis, dass diese Kompetenzen für zukünftige Ausbildungsbereiche von Bedeutung sind, ist eine Förderung der intrinsischen Motivation überaus

wahrscheinlich.

Einen Schritt weiter würde die vollständige Entwicklung eines sogenannten „Evaluation Kits“ gehen. Die Lernenden fertigen gewissermaßen die erforderliche(n) Hardware-Komponente(n) für den Kompetenzbereich INIT - von der BauteilAuswahl (für die periphere Beschaltung des Mikrocontrollers) bis zur Herstellung der Leiterplatte(n). Inwieweit das jedoch noch mit dem Ausbildungsziel der HTL-Fachrichtung IT vereinbar ist, gilt es schulintern abzuklären.

5.3 Embedded Systems als Lehrgegenstand

Dieser Abschnitt hat zum Ziel, den State of the Art im Bereich der Embedded System-Ausbildung zu erheben: „Welche Vorgehensweisen oder Unterrichtsmethoden eignen sich besonders für diese Disziplin?“ oder „Wie kann sich erfolgreicher, kompetenzorientierter Mikrokontrollertechnik-Unterricht gestalten, der auf intrinsischer Motivation beruht?“ sind nur zwei Fragen, denen in diesem Abschnitt nachgegangen wird.

Nachdem sich die Ausarbeitung der didaktischen Konzepte in Kapitel 6 primär auf die Mikrokontrollertechnik konzentriert, beschränkt sich nachfolgende Darstellung des State of the Art ausschließlich auf diesen Themenbereich. Feldbusse oder SPS sowie Prozessdatenverarbeitung, -visualisierung oder -kommunikation, die ebenfalls Bestandteil des Kompetenzbereichs INIT sind, finden keine Berücksichtigung.

Der im Lehrplan angeführte Themenbereich Mikrokontrollertechnik (vgl. Kapitel 5.1.1) wird grundsätzlich im Kontext von Embedded Systems gelehrt - zumindest schlagen das die Autoren jenes Werkes vor, dass die Inhalte der universitären „Computer Engineering“²-Ausbildung (vgl. Technische Informatik) skizziert [67]. In diesem Zusammenhang ist anzumerken, dass Embedded Systems eine überschaubare Historie als eigenständiger Lehrgegenstand aufweist: in der Regel wurden (beziehungsweise werden auch gegenwärtig noch) entsprechenden Inhalte in Lehrveranstaltungen wie „Betriebssysteme“, „Computerarchitekturen“ oder „Elektrotechnik“ behandelt [68]. Grimheden et al. hält diesbezüglich im Jahr 2005 fest [69]:

Even though embedded systems have been designed for more than 30 years, the academic subject of embedded systems is a new, relatively undefined subject, mostly regarded as an interdisciplinary field combining areas such as computer science, automatic control and electrical engineering. The scope of the subject is continuously discussed, with its implications on research and education. [...]

²Grundsätzlich handelt es sich hierbei um einen Vorschlag für den tertiären Bildungsbereich, aber nachdem der Schultyp HTL in inhaltlichen Belangen in den Hochschulbereich hineinragt - vor allem in der Fachtheorie - denn andere Länder haben vergleichbare Ausbildungen im tertiären Bereich angesiedelt (vgl. Kapitel 2.3.2), bildet dieser die Grundlage zur Erhebung des State of the Art. Weiters ist festzuhalten, dass neben den Empfehlungen für „Computer Engineering“ auch welche für „Information Technologie“, die sich grundsätzlich eher mit dem HTL-Fachbereich Informationstechnologie decken, verfügbar sind [66]. Diese umfassen jedoch keine Mikrokontrollertechnik-Inhalte. Hier unterscheidet sich der HTL-Fachbereich Informationstechnologie vom tertiären Vorschlag.

Demnach kann Embedded Systems als durchaus junger - und teilweise auch diffuser - Lehrgegenstand bezeichnet werden. Das spiegelt sich in den unterschiedlichen Bezeichnungen sowie Inhalten von Kursen mit Embedded System-Charakter wider. Mögliche Kursbezeichnungen reichen von „Embedded System“ bis hin zu „Embedded System Design“ und „Embedded System Engineering“ sowie von „Embedded Programming“ bis hin zu „Embedded Software“. Gemäß dieser unterschiedlichen Bezeichnungen variieren auch die Lehrinhalte, die beispielsweise einmal mehr oder weniger die Hardwareaspekte von Embedded Systems berücksichtigen.

5.3.1 Kursimplementierungen

Eine überaus wichtige Frage, die es bei der Konzeption einer Embedded System-Lehrveranstaltung zu beantworten gilt, diskutiert Grimheden in [69]: Sollen Lernende „von allem etwas“ oder „alles von etwas“ können? Umgelegt auf Embedded Systems könnte das wohl bedeuten, dass Lernende unterschiedliche Mikrocontroller-Technologien kennenlernen, indem ein Übungsbeispiel auf mehreren Technologien durchexerziert wird. Das hätte zwar den Vorteil, dass die Lernenden einen Überblick über die verschiedenen Mikrocontroller-Technologien und deren jeweilige Vor- und Nachteile, jedoch keine vertiefende Kenntnisse erfahren. Alternativ bietet es sich an, eine Mikrocontroller-Technologien auszuwählen und den Fokus auf ein tiefgehendes Verständnis der selektierten Mikrocontroller-Plattform zu richten. Die grundlegende Idee dahinter ist, dass Lernende Experten auf einer bestimmten Plattform werden und in weiterer Folge imstande sind, dass Wissen zu verallgemeinern und auf eine andere Mikrocontroller-Plattform zu transferieren. Dieser Ansatz entspricht durchaus auch dem Grundgedanken des Problem-Based Learning (vgl. Kapitel 4.4.2), bei welchem die Lernenden anhand einer konkreten Problemstellung grundlegende Kompetenzen erwerben.

Der von Grimheden beschriebene und an der Königlich Technischen Hochschule (KTH) von Stockholm implementierte Embedded System-Kurs folgt einer allgemeinen Initiative der KTH. Ziel dieser Initiative ist ein Studienplan, der die Studierenden gemäß des „Conceive“, „Design“, „Implement“ und „Operate“ (CDIO)-Modells unterrichtet. Im Detail bedeutet das, bei allen Kursen, bei welchen die Systementwicklung ein zentrale Aufgabe darstellt, werden die Lehrinhalte gemäß des CDIO-Modells gelehrt - wie eben bei Embedded System. Wobei es festzuhalten gilt, dass das CDIO-Modell ausschließlich in der letzten Phase der Ausbildung Berücksichtigung findet. Der Unterricht wird in dieser Phase der Ausbildung gänzlich projekt- und PBL-basiert abgehalten. Die Aufgabenstellungen orientieren sich durchgängig an realen Problemstellungen der Industrie. Unternehmen werden eingeladen, diese und die dahinterliegende Motivation zur Lösung zu präsentieren. Die Projekte werden gemäß des CDIO-Modells organisiert. Neben den fachlichen Aspekten kommt auch der Entwicklung sozialer und personaler Kompetenzen sowie Kompetenzen im Bereich des Projektmanagements eine Bedeutung zu. Beispielsweise werden jedem/jeder Studierenden zwei dezidierte Verantwortungsbereiche zugewiesen: einer das Produkt und einer den Prozess betreffend.

Um einerseits experimentelles Arbeiten und andererseits die Motivation als auch Kommunikation zu fördern wurde das Projekt „The lab in your pocket“ an der KTH ins Leben gerufen, dessen grundlegenden Ideen zusammengefasst dargestellt werden:

1. Jeder Student/jede Studentin hat jederzeit Zugang zu seinem/ihrem eigenen Labor-Equipment.
2. Das Equipment kann ortsungebunden verwendet werden. Einzige Voraussetzung ist ein herkömmlicher PC.
3. Mit dem bereitgestellten Equipment können sämtliche Laborübungen des Kurses bewältigt werden.
4. Das Equipment fördert sogenannten „Open-ended“-Lösungen. Was bedeutet, dass alle Aufgaben-/Problemstellungen eine gewisse Flexibilität innehaben, sodass kreative Lösungen zustande kommen können.

Die Frage, wie ein Embedded System-Kurs grundsätzlich strukturiert sein kann, behandelt Muppala [70]. Der an der Hong Kong University of Science and Technology (HKUST) implementierte Embedded System-Kurs führt diesbezüglich die drei „S“ an: Science, Skills und State of the Art. Science umfasst die theoretischen Grundlagen. Skills bezieht sich auf die praktischen Fähigkeiten, die erforderlich sind, um das theoretische Wissen an realen Beispielen umzusetzen. Unter State of the Art wird das Wissen über aktuelle Technologien (in der Industrie) verstanden. Die Aufteilung der drei Bereiche erfolgt zu gleichen Teilen.

Hartmann et al. führt in diesem Zusammenhang an, dass erfolgreicher Informatikunterricht im Allgemeinen einerseits ausreichend Konzept- aber auch das erforderliche Produktwissen vermitteln muss [16]. Das Konzeptwissen umfasst grundlegende, allgemein gültige Zusammenhänge eines Sachgebietes. Das Produktwissen konzentriert sich hingegen auf die Anwendung eines bestimmten Produktes. Um beispielsweise neue Fakten in größeren Zusammenhängen einordnen zu können, ist wiederum das Konzeptwissen unabdingbar. Im günstigsten Fall wird der Zusammenhang dieser Wissensbereiche im Unterricht hergestellt.

Der zuvor angeführte Kurs an der HKUST adressiert primär die Software-Aspekte von Embedded System; daher auch die Bezeichnung „Embedded Software“. Hardware-Aspekte werden nur in dem Ausmaß behandelt, wie es für die Softwareentwicklung erforderlich ist. Abgehalten wird der Kurs gegen Ende eines Informatik-Studiums. Die Auswahl der eigentlichen Lehrinhalte macht Muppala nicht zuletzt an zwei zentralen Punkten fest: einerseits daran, was StudentInnen der Informatik über Embedded System/Software grundsätzlich wissen sollten und andererseits an deren Vorwissen beziehungsweise den Inhalten vorangegangener Kursen.

Nachdem die StudentInnen in den entsprechenden Kursen bereits ausgeprägte Programmierkenntnisse erworben haben, ist die Darlegung des Unterschieds von der Embedded Software-Entwicklung gegenüber der traditionellen ein Schwerpunkt. Ausschlaggebend hierfür ist der Umstand, dass der Einsatz von Entwicklungsumgebungen, wie zum Beispiel *Microsoft Visual Studio*TM, das grundlegende Verständnis des Kompilierungs-Prozesses nicht unbedingt fördert. Dies ist aber eine Voraussetzung für das Konzept der plattformübergreifenden Entwicklung. Nachdem bei Embedded Systems andere Anforderungen an das Betriebssystem gestellt werden als bei herkömmlichen Computer-Systemen, sieht Muppala RTOS als wichtigen Bestandteil der Embedded Software-Ausbildung an der HKUST. Der Fokus liegt hier viel

mehr auf der Verlässlichkeit oder dem Echtzeitverhalten als auf Dateisystemen oder der virtuellen Speicherverwaltung. Der dritte Schwerpunkt konzentriert sich auf Embedded Software-Engineering. Hierbei werden bereits bekannte Methoden/Techniken (z.B. Wasserfall- oder Spiralenmodell) im Kontext der Embedded-Entwicklung betrachtet. Weiters werden formale Methoden wie UML (*Unified Modeling Language*) im Speziellen und Software-Testen im Besonderen behandelt. Dieser Kurs soll mehr oder weniger einen Überblick über relevante Themen der Embedded Software-Entwicklung geben. Die praktischen Übungen werden ausschließlich projektbasiert im Rahmen von Gruppenarbeiten abgehalten. Die maximale Gruppengröße beläuft sich hierbei auf drei StudentInnen. Den Kursabschluss stellt ein größeres Projekt dar, welches mehrere Wochen in Anspruch nimmt. Die Formulierung der Projektidee obliegt den StudentInnen.

Eine zentrale Erkenntnis, die aus der Durchführung dieses Kurses resultiert, ist, dass StudentInnen zu Beginn des Kurses oftmals dieselben Wissenslücken aufweisen. Beispielsweise sind die allgemeinen Programmierfähigkeiten nicht in dem Ausmaß vorhanden, wie es für einen schnellen Einstieg in die Embedded Software-Entwicklung erforderlich wäre. Probleme gibt es auch beim grundlegenden Verständnis von Computer-Architekturen oder Betriebssystemen. Das sind scheinbar Themen, die von den StudentInnen generell wenig geschätzt werden.

Weitere Kurse, welche ebenfalls die Software-Aspekte von Embedded Systems adressieren, sind beispielsweise an der RWTH Aachen (Deutschland, kurz RWTHA) [71] sowie an der Tampere University of Technology (Finnland, kurz TUT) [72] implementiert.

Der Kurs an der RWTHA ist für Drittsemester der Informatikausbildung konzipiert und verfolgt zwei zentrale Ziele: einerseits sollen die StudentInnen mit der „Embedded Hardware“ (Mikrocontroller) und den damit verbundenen Einschränkungen (z.B. I/O oder Speicher) bei der Entwicklung in Berührung kommen. Andererseits sollen durch den Einsatz entsprechender Unterrichtsmethoden (z.B. Partnerarbeit) soziale Kompetenzen entwickelt werden. Der Kurs erstreckt sich über ein Semester. Im Vergleich zum bereits vorgestellten Kurs an der HKUST erfolgt die Vermittlung der Lehrinhalte nicht anhand einzelner Themenschwerpunkte, die der Reihe nach abgearbeitet werden, sondern anhand eines großen Übungsbeispiels, das sich über die Dauer des Kurses erstreckt und alle relevanten Punkte der Embedded Software-Entwicklung umfasst. Als Beispiel dient die Entwicklung eines Echtzeitbetriebssystems. Hintergrund für die Beispielsauswahl ist, dass aufgrund der damit verbundenen Interdisziplinarität das vernetzte Denken mit der entsprechenden Disziplin gefördert wird. Tabelle 5.5 zeigt die modulare Aufschlüsselung des Projektes.

Tabelle 5.5: Laborübung des Embedded System-Kurses an der RWTHA - Ablauf und Inhalte [71]

Block	Thema
1	Boot Loader
2	Scheduler
3	Memory Management
4	Process Manager
5	External Memory Chip
6	Application Task

Anhand dieser Modularisierung soll beispielsweise die Idee der Wiederverwendung einzelner Programmbausteine verdeutlicht werden. Im Gegensatz zu dem von Grimheden beschriebenen Kurs an der KTH Stockholm (siehe oben) gibt es nur eine begrenzte Anzahl an sogenannten „Evaluation Kits“ (EK), um Programme zu testen. Die Verleihung dieser EK wird vom Institut gesteuert. Als Mikrocontroller-Plattform kommt der ATmega-Controller der AVR-Familie zum Einsatz. Als Entwicklungsumgebung das *AVR Studio* [73]. Nachdem die Programme in C zu schreiben sind, ist ein entsprechender Compiler erforderlich. Hierfür bietet sich der freie *avr-gcc* aus dem Softwarepaket *WinAVR* [74] an. Begründet wird diese Komponentenauswahl einerseits mit der zunehmenden Bedeutung der AVR-Mikrocontroller, andererseits damit, dass sämtliche Entwicklungs-Werkzeuge frei zur Verfügung stehen. Als Ergänzung zum integrierten Simulator des AVR Studios wird auf den ebenfalls freien Simulator *HapSim* [75] verwiesen.

Bei der Konkretisierung der Lehrinhalte wird vor allem das Ziel angeführt, die StudentInnen mit den Herausforderungen des Schreibens von hardwarenaher Software sowie den damit verbundenen Ressourceneinschränkungen bei der Hardware zu vermitteln, da dies als die wesentlichen Punkte in dieser Disziplin angesehen werden. Das impliziert beispielsweise das Konfigurieren des Mikrocontrollers und dessen peripheren Komponenten mit den dafür vorgesehenen Registern.

Die einzelnen Übungsmodule werden von einer Beschreibung begleitet. Diese korrespondiert mit den theoretischen Unterrichtseinheiten insofern, dass ein unmittelbarer Bezug hergestellt wird. Im Detail sind es drei Inhaltsschwerpunkte:

- Ziel des Übungsblocks
- Theorie und Implementierungshilfen
- Wiederholungsfragen zum Üben

Wie Muppala haben auch Stollenwerk et al. an der RWTHA festgestellt, dass die StudentInnen zwar relativ firm im Bereich der Java- oder C#-basierten objektorientierten Programmierung (OOP) sind, jedoch Schwierigkeiten beim Einstieg in die hardwarenahe Programmierung haben.

Der Kurs an der TUT verfolgt ebenfalls - wie bereits angeführt - die Software-Aspekte von Embedded Systems. Ähnlich dem Kurs an der RWTHA kommt der Atmel AVR-Controller mit derselben Begründung zum Einsatz, gleiches gilt für die Programmiersprache C. Der Kurs erstreckt sich ebenfalls über ein Semester (im Informatik-Bachelor). Als Lernziele werden folgende Punkte festgehalten:

Die StudentInnen

1. können Programme für Embedded System mit oder ohne Betriebssystem schreiben.
2. verstehen die Bedeutung von „Cross-compiling“ und wissen, warum es verwendet wird.
3. können herkömmliche periphere Komponenten betreiben.
4. wissen was Interrupts sind und können diese anwenden.

Wie an der RWTHA wird als Methode die Partnerarbeit eingesetzt und es gibt nur eine begrenzte Anzahl von EK. Die Unterrichtsinhalte schlüsseln sich, wie in Tabelle 5.6 dargestellt, folgendermaßen auf:

Tabelle 5.6: Inhalte des Embedded System-Kurs an der TUT [72]

Block	Thema
L1	Introduction to embedded systems
L2	Peripheral interfaces
L3	Memory management
L4	Software engineering work in embedded systems
L5	Concurrency and real-time
L6	Scheduling
L7	Kernels
L8	Peripheral controlling
L9	Lifting the abstraction level
L10	Introduction to Symbian environment
L11	Distributed embedded systems, revision lecture

Ergänzend zu den angeführten Themenblöcken gibt es drei praktische Projekte, die zeitversetzt die jeweiligen Themen behandeln. Bei jedem dieser Übungsprojekte sind vonseiten der Studierenden drei Teile zu erbringen:

1. Software, welche die geforderte Funktionalität realisiert.
2. Präsentation, bei welcher die Ergebnisse vorgestellt werden.
3. Umfangreiche Dokumentation

Eine ausführliche Evaluierung des Kurses hat mehrere Erkenntnisse zutage gebracht. Beispielsweise ist der Einsatz von peripheren Hardware-Komponenten (wie zum Beispiel einem Bluetooth-Modul) teilweise mit Problemen behaftet, da ein Ausfall bei fehlendem Ersatz problematisch ist. Vor allem dann, wenn es nur vier Arbeitsplätze (PCs) mit jeweils einem EK gibt. Eine Alternative wäre der Einsatz von Emulatoren, wodurch mögliche Hardware-Probleme grundsätzlich nicht auftreten würden. Die Autoren weisen jedoch darauf hin, dass das einerseits eben dieser Erfahrungswerte sowie andererseits der Motivation abträglich wäre. Daher wird der Einsatz von EK bevorzugt. Nachdem den drei Übungsprojekten ein Demonstrationsprojekt vorausging und die StudentInnen bereitgestellte Codeteile aus diesem verwenden durften, wurde das ebenfalls als überaus motivierend für die weiteren, schwierigeren Aufgaben empfunden. Die Motivation, bei der Entwicklung auf die Wiederverwendbarkeit von Codeteilen zu achten, wurde mit dem dezidierten Hinweis, dass das nicht nur erlaubt, sondern gewünscht ist, entscheidend gefördert. Den Einsatz vorkonfigurierter EKs (Ann. Mikrocontroller mit z.B. Bluetooth-Modul oder LCD) sehen die Kursverantwortlichen ebenfalls positiv.

Bevor abschließend auf zentrale Fragen der intrinsischen Motivationsförderung eingegangen wird, erfolgt eine zusammenfassende Darstellung einer Fallstudie zum Thema „Mikrocontroller-Ausbildung“ an der Technischen Universität (TU) Wien [76]. Der von Weiss et al. vorgestellte Kurs wird im zweiten Jahr des Studiums der Technischen Informatik abgehalten und behandelt sowohl Hard- als auch Software-Aspekte der Mikrocontrollertechnik. Daher werden

einerseits Vorkenntnisse im Bereich der C-Programmierung sowie andererseits im Bereich der Computer-Architekturen und Elektrotechnik/Elektronik vorausgesetzt. Im Gegensatz zu den bereits vorgestellten Kursen beleuchtet die Fallstudie zum Beispiel auch wirtschaftliche Aspekte. Obwohl nicht im Detail hervorgeht, wie die Punktevergabe erfolgt ist und inwieweit hier auch standortspezifische Einflussfaktoren zum Tragen kommen, liefern die in Abbildung 5.5 dargestellte Bewertung potenzieller Einflussfaktoren eine weitere Grundlage zur Unterrichtskonzeption. Es werden relevante Faktoren der Unterrichtsdurchführung hinsichtlich deren finanzieller, pädagogischer und motivationaler Auswirkung bewertet. Ein „+“ bedeutet einen positiven und ein „-“ einen negativen Einfluss. Die Werte in den Klammern beziehen sich auf einmalige Kosten (Ann. Einrichtungskosten).

	Fin.	Edu.	Mot.	Total
autogradable exams	+++	-	-	1
essay exams	---	+	++	0
just 1 exam	++	-	--	-1
several exams	-	+	++	2
drop result	-		+++	2
flexible grading	-	+	+++	3
voluntary homework	+++	---		0
ungraded homework	++	++	---	1
graded homework	---	+++	+	1
flexible workload	-		++	1
extra work counts	-	+	++	2
open labs	+++	-	++	4
closed labs	--	+++	-	0
supervision	---	++		-1
no supervision	+++	---	--	-2
group work	++	--		0
lectures	--	++	++	2
electronic lectures	[- - -]	++	+	3
no lectures	+++	--	-	0
script	[- -]	+	+	2
suppl. material	[- -]	+	++	3

Abbildung 5.5: Verschiedene Methoden, Maßnahmen etc. und deren Einfluss auf die Bereiche „Finance“, „Education“ und „Motivation“ [76]

Weiss et al. kommt ebenfalls zu dem Schluss, dass der Einsatz realer Hardware aus mehreren Gründen zielführend ist: zum Einen sind es die wertvollen Erfahrungen, die beispielsweise beim Hantieren mit der Hardware gemacht werden. Zum Anderen fördert es die Motivation, um auch eigene Projekte zu beginnen. Als Hardware-Plattform wird, wie bei anderen Kursen auch, die AVR-Familie von Atmel eingesetzt. Hierbei handelt es sich um einen EK, der Matrix-Tastenfeld, LEDs, 7-Segmentanzeige, RS-232 etc. integriert. Für herausfordernde Projekte wird auch noch ein „Motoren-Board“ bereitgestellt.

Der Kurs ist in drei Teilbereiche unterteilt. Der erste Block umfasst einerseits die Vorstellung der eingesetzten Werkzeuge (z.B. Entwicklungsumgebung), andererseits erfolgt eine Einführung in Mikrocontroller-Architekturen, der eingesetzten Hardware und der Assemblersprache. Der zweite Block konzentriert sich primär auf die Verwendung und Programmierung grundlegender Mikrocontroller-Komponenten. Hierzu zählen Ein- und Ausgabeein-

heiten (z.B. serielle und parallele Ein-/Ausgabe oder Wandlung zwischen analogen und digitalen Signalen), zeitgeberbasierte Einheiten (z.B. Zähler) und Unterbrechungssteuerung. Neben den Mikrocontroller-Komponenten werden auch typische Hardwarekomponenten wie 7-Segment-Anzeigen eingesetzt. Aufbauend darauf verfolgt der dritte Block die Entwicklung konkreter Anwendungen. Der zweite und dritte Block basierend auf der Programmiersprache C.

Die Erfahrung hat gezeigt, dass komplexe Aufgabenstellungen die StudentInnen eher dazu verleiten, Programme irgendwie „zum Laufen zu bringen“ als sie „richtig“ zu machen. Daher haben die Kursverantwortlichen der TU Wien eher auf einfachere Aufgabenstellung gesetzt. In diesem Zusammenhang hat sich wiederum herausgestellt, dass Gruppenarbeiten (3 Personen/Gruppe) in einer sogenannten „Closed Lab-Organisationsform“ nicht unbedingt zu besseren Lernergebnissen bei den StudentInnen führen. Zurückzuführen ist das wohl auf die simplen Aufgabenstellungen, die sich weniger für Gruppenarbeiten eignen - vielmehr nimmt sich in der Regel ein Gruppenmitglied der Arbeit an, während die verbleibenden Mitglieder zusehen. Komplexe Aufgaben können hingegen nicht von einer Person im Alleingang erledigt werden. In diesem Fall ist die Gruppenarbeit eher zielführend - ob auf fachlicher oder sozialer Ebene, da grundsätzlich Engagement von allen gefordert ist. Diese Erkenntnis hat dazu geführt, dass die Sozialform Gruppenarbeit und die Organisationsform Closed Lab zumindest in der Anfangsphase des Kurses keine Berücksichtigung mehr finden. Es ist jedoch festzuhalten, dass Closed Labs grundsätzlich eine effektive pädagogische Methode darstellen (vgl. Abbildung 5.5, Spalte Edu.). Zu diesem Schluss kommt auch Kumar in [77]. Jedoch ist diese Organisationsform überaus ressourcenintensiv und bei dem zuvor vorgestellten Kursaufbau nicht unbedingt förderlich. Als Konsequenz daraus werden die Labor-Unterrichtseinheiten viel mehr für Diskussion und zur Beantwortung von Fragen genutzt.

Damit die StudentInnen ihr Wissen festigen als auch vertiefen können, wird zusätzlich noch ein Pool mit Übungsaufgaben bereitgestellt. Die StudentInnen können eine bestimmte Anzahl an Aufgaben daraus wählen, dies erfolgt auf freiwilliger Basis. Für jede absolvierte Zusatzaufgabe gibt es Bonuspunkte, welche in die Beurteilung einfließen.

Die bei den praktischen Prüfungen gestellten Aufgaben belaufen sich auf maximal 20 Codezeilen; wobei in der Regel ein bereitgestelltes „Framework“ zu verwenden ist. Damit eine Auskunft über das erzielte Ergebnis relativ zeitnahe getätigt werden kann, erfolgt eine Computer-gestützte Auswertung der Programme, die es im Rahmen der praktischen Prüfung zu entwickeln galt. Die Theorieprüfung besteht ausschließlich aus so genannten „Ja/Nein-Fragen“. Weiss et al. weisen diesbezüglich auf die positive Resonanz der StudentInnen hin.

5.3.2 Motivations- und Kreativitätsförderung

Apiola et al. [78] gehen der Frage nach, wie eine Kurs beschaffen sein muss, der die Kreativität hinsichtlich neuer Ideen, die intrinsische Motivationsbildung sowie den Konstruktivismus im Allgemeinen (vgl. Kapitel 2.1.2) bestmöglich fördert. Damit intrinsische Motivation entstehen kann, sollten folgende Voraussetzungen gegeben sein:

- *Social interaction that enhances feelings of competence*

- *Optimal challenges, effectance-promoting feedback, and freedom from demeaning evaluations*
- *Supporting internal perceived locus of causality*
- *Providing choice, acknowledge of feelings and opportunities for self-direction*

Die Förderung der intrinsische Motivation basiert also in erster Linie auf der Schaffung externer Rahmenbedingungen, die diesen Prozess in weiterer Folge ermöglichen sollen. Um kreativ zu sein, so die Autoren, müssen drei Bedingungen erfüllt sein:

- Domäne-relevantes Wissen,
- Kreativprozess (Arbeitsweisen wie z.B. „Brainstorming“, „Open Space Workshop“ [79] oder „3+“) und
- intrinsische Motivation.

Dass Kreativität eher in Kollaboration als in Einzelarbeit entsteht, ist ebenfalls ein wesentlicher Faktor. Basierend auf diesen Annahmen und Rahmenbedingungen wurde die in Abbildung 5.6 dargestellte Lernumgebung entwickelt.

Component	Method of support
Competence	Encouraging teamwork, use of creativity enhancing methods, providing effectance promoting feedback
Autonomy	Providing choice and opportunity for self-direction
Relatedness	Promoting social interaction and creative working methods
Domain relevant skills	Requiring good computing skills from all attendees.
Creative processes and working styles	Use of creativity-enhancing methods in course sessions: brainstorming, 3+ and open space workshops
Constructionism	Use of LEGO-mindstorms robots. Lend each student their own robotics-kit.

Abbildung 5.6: Komponenten der Lernumgebung [78]

Um die Lernumgebung zu evaluieren, wurde der Kurs „Robotics Programming Project“ ins Leben gerufen. Die Teilnahme war freiwillig. Ausgeprägte Programmierkenntnisse waren Grundvoraussetzung. Der Kurs wurde in sechs Einheiten abgehalten (siehe Abbildung 5.7). Als Arbeitsplattform diente *Lego Mindstorms*TM [80].

Wie der Abbildung 5.7 zu entnehmen ist, fanden vier Kreativ-Meetings statt. Bei der zweiten Einheit, dem ersten Kreativ-Meeting, kam die Methode „3+“ zum Einsatz. Hierbei werden die StudentInnen in Gruppen eingeteilt und ein Sesselkreis gebildet. Um den Kreativprozess in Gang zu bringen, artikuliert eine Person eine Idee, wie der Roboter konstruiert werden könnte. Der/die nächste StudentIn im Kreis führt im ersten Schritt drei positive

#	Short description
1.	Information and instructions. Borrowing of robotics kits.
2.	Creativity meeting. Short lecture on creativity methods. Divide students into groups. Ideation exercises with the 3+ method.
3.	Creativity-meeting. Divide students into groups. Continue on creative working methods. Prepare posters. Use open-space workshop-method
4.	Creativity-meeting. Generate ideas using creativity methods in one large group. Discuss the phase and ideas of every group's project. Use open space workshop
5.	Creativity-meeting. Generate ideas using creativity methods in one large group. Discuss the phase and ideas of every group's project. Use open space workshop
6.	Final demo. Every group presents their final project.

Abbildung 5.7: Zusammenfassende Darstellung der Kurs-Meetings [78]

Ideen und im zweiten Schritt eine kritische Bemerkung betreffend der Idee des Sitznachbars/der Sitznachbarin an u.s.w. Bei der dritten Runde kam die Methode Open Space zum Einsatz. Die zuvor gebildeten Gruppen präsentierten deren Ideen, mögliche Lösungen und damit verbundene Probleme mit einem Poster; wobei immer nur die Hälfte der jeweiligen Gruppe präsentierte. Die restlichen Gruppenmitglieder begutachteten die Ergebnisse der anderen Gruppen. Das unter anderem Besondere an dieser Arbeitsmethode ist, dass nicht nur das Endergebnis, sondern auch der Prozess der Ideenfindung als auch die Plaunungs- und Entwicklungsphase vorgestellt werden. Alles in allem stellt dieser Ansatz einen in höchstem Maße StudentInnen-zentrierten Ansatz dar, der überaus positiv aufgenommen wurde.

Pollard et al. [81] betrachten das Thema Motivationsförderung aus einem anderen Blickwinkel, indem sie untersuchen, inwieweit die einfachsten Unterrichtsmethoden/-techniken (z.B. aus dem Kindergarten) auf Hochschulebene angewendet werden können. SchülerInnen, die beispielsweise keine Motivation in guten Noten sehen, können unter Umständen durch einen (Gruppen-)Wettbewerb motiviert werden; zum Beispiel im Kontext von Stundenwiederholungen, bei welchen in der Regel eine überschaubare Beteiligung anzutreffen ist. Die Stundenwiederholung könnte somit als Einzel- oder Gruppenquiz stattfinden, indem mehrere Runden absolviert werden, bis der/die SiegerIn feststeht. Allein die Aussicht zu gewinnen, kann motivationsfördernd wirken. Alternativ könnten auch noch „kleine“ Preise in Aussicht gestellt werden. Mögliche Quizzinhalte könnten beispielsweise das Analysieren und Vorhersagen eines Programms oder das Finden von Fehlern in einem Quellcode-Beispiel sein. Pollard et al. weisen weiters darauf hin, dass gute SchülerInnen bei einem Quiz eher motiviert sind, anderen zu helfen, als es sonst der Fall ist. Zurückzuführen ist das auf den Umstand, dass sie ihre eigenen Ziele - eben den Gewinn des Quiz' - erreichen möchten. Schwächere LernerInnen sind unter Umständen wiederum eher motiviert, sich vorzubereiten, da sie ebenfalls einen positiven Beitrag leisten möchten. Als weitere Quiz-Beispiele werden Quellcode-basierte Lückentexte oder das Ordnen von Quellcode-Zeilen angeführt. Neben den einzelnen Stundenwiederholungen bieten sich auch Projekte für einen Wettbewerb an - transparente und

nachvollziehbare Beurteilungskriterien sind jedoch ein Grundvoraussetzung. Vorsicht ist auch bei der Zusammensetzung der Gruppen geboten (siehe Kapitel 4.4.1, Sozialform Gruppenarbeit).

Gemäß dem Motto „Toys: Making the abstract concrete“ schlagen Pollard et al. vor, Objekte (z.B. Spielsachen) als physische Metapher zu verwenden, um abstrakte Konzepte der Informatik zu verdeutlichen. Findet das regelmäßig statt, sind SchülerInnen aufgrund der Vorfreude - „Welche neuen Tricks folgen heute?“ - eher motiviert. Generell gilt: zuerst Konkretes und dann Abstraktes. Eine weitere motivationsfördernde Maßnahme ist, wenn SchülerInnen Themen der Informatik „erleben“, indem sie beispielsweise eine einfache Kontrollstruktur (z.B. Schleife) oder einen komplexen Algorithmus in einem Rollenspiel durchspielen. Somit bekommen die SchülerInnen unmittelbar Feedback, ohne eine Programmiersprache beherrschen zu müssen.

Als Abschluss zum Thema Motivations- und Kreativitätsförderung werden die von Romeike [82] definierten Kriterien kreativen Informatikunterrichts überblicksartig vorgestellt.

Romeike führt in diesem Zusammenhang einerseits Kriterien für die Fach- sowie andererseits für die Schülerdimension an. Im Kontext der Fachdimension sieht Romeike Konzeptwissen als eine elementare Grundvoraussetzung für jede kreative Tätigkeit an. Als weitere Kriterien werden die subjektive Neuheit sowie die Offenheit angesehen. Das heißt, SchülerInnen soll die Möglichkeit geboten werden, ihre Aufgabenstellung selbst mitgestalten zu können. Weiters soll eine gewisse Offenheit hinsichtlich Ergebniserwartung, Bearbeitungsweg und -tiefe gegeben sein. Da die Vorstellungskraft hinsichtlich eigenständiger, kreativer Lösungen nicht vorausgesetzt werden kann, ist gegebenenfalls ein Stimulus - konkrete Ideenregung - erforderlich, sodass die SchülerInnen in weiterer Folge imstande sind, eigene Ideen zu entwickeln.

Für die Schüler-Dimension sind die Relevanz, die Identifikation und Originalität von Bedeutung. Die zu bearbeitenden Inhalte sollten aus der Lebenswelt der SchülerInnen stammen. Nur wenn die Herstellung eines persönlichen Bezuges möglich ist, kann Kreativität entstehen. Gleiches gilt für die Identifikation, wenn die Bearbeitung in einer positiven Eigendynamik enden soll.

Als dritter Punkt ist die Unterrichtsumgebung zu sehen. Ein gutes Klima sowie ordentliche Unterrichtsmittel sind Grundvoraussetzung. Ebenso muss es möglich sein, dass SchülerInnen experimentieren, oder Heuristiken anwenden und die erdachten Lösungsmöglichkeiten testen können. Dass der Faktor Zeit ein wichtiger ist, bedarf keiner weitreichenden Ausführung, unter zeitlichem Druck kann sich keine Kreativität entfalten.

5.4 Synthese der Anforderungen an die Unterrichtskonzeption

In den vorangegangenen Abschnitten dieses Kapitels erfolgt eine Analyse der Ausgangssituation sowie eine Darlegung der Rahmenbedingungen, die durch den Lehrplan beziehungsweise die Bildungsstandards vorgegeben sind. Der letzte Abschnitt dieses Kapitels hat die Synthese der Anforderungen zum Ziel. Hierbei werden die zuvor dargestellten Inhalte hinsichtlich

deren Bedeutung für den nachfolgenden Entwurf der didaktischen Konzepte in Kapitel 6 diskutiert, bewertet und in konkrete Anforderungen überführt, die es zu berücksichtigen gilt.

Aus Gründen der Übersichtlichkeit werden vier Cluster gebildet, welchen die entsprechenden Anforderungen in weiterer Folge zugeordnet werden:

- Cluster A: Pädagogik und Lernpsychologie
- Cluster B: Kompetenzorientiertes Unterrichten
- Cluster C: Gesetzliche und institutionelle Vorgaben
- Cluster D: Embedded System als Unterrichtsgegenstand

Ergänzend ist noch festzuhalten, dass aufgrund der vielseitigen Herausforderungen, die sich für die Lehrkraft im schulischen Alltag ergeben, nicht alle Anforderungen bei jeder Unterrichtseinheit berücksichtigt werden können - was unter Umständen auch gar nicht zielführend ist. Es soll eine Grundlage für den Planungsprozess geschaffen werden, dessen Ergebnis im günstigsten Fall die Summe aller Anforderungen widerspiegelt.

5.4.1 Anforderungscluster A: Pädagogik und Lernpsychologie

In Kapitel 2.1 werden unterschiedliche Lernformen vor- und gegenübergestellt. Ziel muss sein, das schulische Lernen so gut wie möglich dem natürlichen als auch dem sinnvollen und inzidentellem Lernen gleichzusetzen (siehe Tabelle 5.7, A1). Nachdem die ersten und letzten Bestandteile einer Lernstrecke besonders schnell und gut aufgenommen werden, gilt es diesen Umstand zu berücksichtigen (A2). Eine weitere Anforderung stellt der Umstand dar, dass die SchülerInnen neue Inhalte ein- und zuordnen können (A3).

Tabelle 5.7: Anforderungen des Bereichs „Pädagogik und Lernpsychologie“

ID	Inhalt
A1	Aspekte des natürlichen, sinnvollen u. inzidentiellen Lernens berücksichtigen
A2	Unterschiedliches Aufnahmepotenzial bei Lernstrecken zu Nutzen machen
A3	Zuordenbarkeit von Lehrinhalten sicherstellen und eindeutige Lernziele formulieren

5.4.2 Anforderungscluster B: Kompetenzorientiertes Unterrichten

Hinsichtlich des kompetenzorientierten Unterrichtens gilt es eine Reihe von Anforderungen zu berücksichtigen. Eine davon ist die Schülerzentrierung (siehe Tabelle 5.8, B1). Dass bestmöglich auf die individuellen Bedürfnisse der SchülerInnen einzugehen ist, ist ebenfalls

ein Aspekt der Kompetenzorientierung (B2). Jedoch steht diese Anforderung in engem Zusammenhang mit externen Einflussgrößen (z.B. der Klassengröße und dem damit verbundenen Gruppenteiler etc.).

Tabelle 5.8: Anforderungen des Bereichs „Kompetenzorientiertes Unterrichten“

ID	Inhalt
B1	SchülerInnen durch den Einsatz handlungsorientierter Methoden in eine aktive Rolle versetzen
B2	Berücksichtigung des individuellen Leistungsvermögens durch Individualisierungsmaßnahmen
B3	Förderung der intrinsischen Motivationsbildung und Kreativität im Unterricht
B4	Methodenvielfalt beim Unterrichten
B5	Zeit zum Lernen und Reflektieren lassen
B6	Vernetzendes Denken durch Interdisziplinarität im Unterricht fördern

Die Methodenvielfalt (B4) ist ebenfalls wichtig, um für Abwechslung im Unterrichtsalltag zu sorgen. Jedoch ist bei der tatsächlichen Umsetzung Vorsicht angebracht, dass diese durch häufigen Wechsel oder zu komplexen Methoden nicht in den Vordergrund tritt und somit das Wesentliche - nämlich der Kompetenzaufbau - aus dem Fokus rücken. Dass Zeit zum Lernen (B5) - im Sinne von beurteilungsfreien Phasen im Unterricht - wichtig ist, erklärt sich von selbst. Vernetztes Denken (B6) ist, insofern es die Rahmenbedingungen ermöglichen, ebenfalls zu berücksichtigen.

5.4.3 Anforderungscluster C: Gesetzliche und institutionelle Vorgaben

Der Cluster „Gesetzliche und institutionelle Vorgaben“ fasst jene Anforderungen zusammen, die einerseits vom BMUKK durch den Lehrplan und die Bildungsstandards sowie andererseits vonseiten der HTL Krems vorgegeben sind.

Tabelle 5.9: Anforderungen des Bereichs „Gesetzliche und institutionelle Vorgaben“

ID	Inhalt
C1	Zielvorgaben des Kompetenzdeskriptors „Wiedergeben und Verstehen“ für μ C-Technik erfüllen
C2	Zielvorgaben des Kompetenzdeskriptors „Wiedergeben und Verstehen“ für SPS erfüllen
C3	Zielvorgaben des Kompetenzdeskriptors „Wiedergeben und Verstehen“ für Industrielle Feldbusse erfüllen
C4	Zielvorgaben des Kompetenzdeskriptors „Anwenden“ für μ C-Technik erfüllen
C5	Zielvorgaben des Kompetenzdeskriptors „Anwenden“ für SPS erfüllen
C6	Zielvorgaben des Kompetenzdeskriptors „Anwenden“ für Industrielle Feldbusse erfüllen
C7	Zielvorgaben des Kompetenzdeskriptors „Analysieren“ für Prozessdatenverarbeitung u. -visualisierung erfüllen
C8	Zielvorgaben des Kompetenzdeskriptors „Entwickeln“ für μ C-Technik erfüllen
C9	Zielvorgaben des Kompetenzdeskriptors „Entwickeln“ für SPS erfüllen
C10	Entwicklung der Sozial- und Personalkompetenz fördern
C11	Berücksichtigung der Rahmenbedingungen der schulautonomen Implementierung des Kompetenzbereichs Industrielle Informationstechnik
C12	Praxisnaher Unterricht mit Fokus auf vertiefende Kenntnisse einer ausgewählten Technologie mit Schwerpunkt Embedded Software-Entwicklung

Um den Erfüllungsgrad der Kompetenzdeskriptoren besser beurteilen zu können, erfolgt eine Unterteilung in einzelne Themenbereiche: Mikrocontrollertechnik (μ C-Technik) (C1, C4 u. C8), SPS (C2, C5 u. C9), Industrielle Feldbusse (C3 u. C6) und Prozessdatenverarbeitung u. -visualisierung (C7). Die Anforderung C10 bezieht sich im Wesentlichen auf die Inhalte des Abschnitts 5.1.2. C11 bezieht sich auf die (schulspezifische) Implementierung des Kompetenzbereichs INIT. Hierzu zählen die veranschlagten Wochenstunden (siehe Tabelle 5.3), die Problematik unterschiedlicher Schulstandorte etc. Dass bei der Anforderung C12 explizit die Software-Aspekte hervorgehoben werden, lässt sich mit dem Ausbildungsziel der IT-HTL begründen, das generell die Softwareentwicklung in den Vordergrund rückt.

5.4.4 Anforderungscluster D: Embedded System als Unterrichtsgegenstand

Abschließend werden jene Anforderungen definiert, die aus der Erhebung des State of the Art hinsichtlich Embedded Systems als Unterrichtsgegenstand resultieren. Hierbei wird beispielsweise die Problematik aufgezeigt, dass das Vorwissen im Bereich der Programmierung, der Computerarchitektur oder der Betriebssysteme oftmals nicht in jenem Ausmaß vorhanden ist, wie es erforderlich wäre. Nachdem Ähnliches zu erwarten ist, soll diesem Umstand mit Anforderung D1 entgegnet werden (siehe Tabelle 5.10). Die Festlegung der Lehrinhalte soll

den drei „S“ von Muppala (siehe Kapitel 5.3.1) folgen (D2). Dabei soll sich die inhaltliche Ausrichtung nur in groben Zügen an den Themenblöcken sowie am zeitlichen Ablauf der vorgestellten Hochschulkursen orientieren. Schlussendlich stellt die Ausbildung des tertiären Bildungsbereichs andere Ansprüche an die Lernenden als jene der Sekundarstufe II. Daher gilt es inhaltliche Abstriche zu machen, sodass die SchülerInnen einen guten Einstieg finden. Das Ergebnis dieses Prozesses soll die Frage „Auswahl des Inhalts - was?“ (vgl. Abbildung 4.1) beantworten.

Die Erhebung des State of the Art hat weiters gezeigt, dass keiner der vorgestellten Kurse eine Einführung in die Embedded C-Programmierung umfasst; obwohl davon auszugehen ist, dass es für die meisten StudentInnen der Erstkontakt mit diesem Fachbereich und den damit verbundenen Herausforderungen ist. Es scheint, als ob die StudentInnen dazu angehalten sind, das erforderliche Know-how, wie zum Beispiel Port-Maskierungstechniken, im Rahmen einer Vorlesung oder „on demand“ im Selbststudium zu erwerben. Dieses Vorgehen sollte für HochschülerInnen in der Regel bewältigbar sein, für SchülerInnen der Sekundarstufe II eher nicht. Andererseits ist eine starre Einführung in die Syntax einer Programmiersprache - nach dem Motto „zuerst Datentypen, aufbauend darauf einfache Kontrollstrukturen, gefolgt von Schleifen u.s.w.“ - alles andere als fesselnd. Daher muss es das Ziel sein, die Einführung in Embedded-C als integralen Bestandteil in die Übungsblöcke zu verpacken (D3).

Tabelle 5.10: Anforderungen des Bereichs „Embedded System als Unterrichtsgegenstand“

ID	Inhalt
D1	Konsolidierung des erforderlichen Vorwissens
D2	3 „S“ (Science, Skills u. State of the Art) berücksichtigen
D3	Embedded C-Einführung als integralen Bestandteil der Übungen implementieren
D4	Modulare, offene Projektthemen mit Flexibilität beim Lösungsweg und Raum für kreative Ergebnisse
D5	Plagiarismus erschweren und individuelle Lernzielerreichung fördern
D6	Einsatz von frei verfügbaren Entwicklungswerkzeugen (IDE u. Simulator)
D7	Einsatz von Evaluation Kits
D8	<i>Flexible grading</i> - Die Gesamtnote resultiert aus mehreren, flexiblen Beiträgen
D9	(Haus-)Übungsbeispiele mit unterschiedlichen Schwierigkeits- und Erfüllungsgrad

Das Konzept modularer, offener Projekte (vgl. „Lab in your pocket“, Kapitel 5.3.1) erscheint auch im Kontext der HTL-Ausbildung als vielversprechend (D4). Zu berücksichtigen gilt es jedoch die geänderten Rahmenbedingungen im Vergleich zur universitären Ausbildung. Aber alles in allem sollten im 4. Jahrgang einer HTL modulare, offene Projekte unter gewissen Voraussetzungen durchführbar sein. Da der Feststellung der individuellen Lernzielerreichung bei der schulischen Ausbildung eine zentrale Bedeutung zukommt, gilt es entsprechende Sequenzen zu implementieren (D4). Insofern möglich, sollte der Plagiarismus erschwert werden. Die Analyse des State of the Art hat weiters ergeben, dass sich der Einsatz von frei verfügbaren Entwicklungswerkzeugen (D6) und, wenn möglich, sogenannter „Evaluation Kits“

(EK) überaus positiv auf die Motivation auswirkt (D7). Die von Weiss et al. vorgestellte Fallstudie (siehe Kapitel 5.3.1) zeigt, dass eine gewisse Flexibilität bei der Zusammensetzung der Note (D8) als auch bei den (Haus-)Übungsbeispielen (D9) entscheidend zur Motivation beitragen. Um dies zu bewerkstelligen, eignet sich der überaus vielseitige Kompetenzbereich INIT hervorragend.

In den vorangegangenen Kapiteln dieser Arbeit werden die relevanten Grundlagen dargelegt, das Projekt Bildungsstandards vorgestellt, das Unterrichtswesen im Kontext der Kompetenzorientierung beleuchtet und schlussendlich konkrete Anforderungen an die Unterrichtskonzeption erarbeitet. Darauf aufbauend werden in diesem Abschnitt die gewonnenen Erkenntnisse in ein didaktisches Konzept für den Kompetenzbereich „Industrielle Informationstechnik“ (INIT) überführt. Für dieses Ziel erfolgt einleitend eine überblicksartige Darstellung der Entwürfe sowie eine Diskussion grundlegender Annahmen. Aufbauend darauf werden ausgewählte Unterrichtssequenzen (vgl. Unterrichtseinheiten) im Detail erarbeitet.

Insofern nichts Gegenteiliges angeführt ist, bezieht sich die Bezeichnung „Anforderung“ auf die in Kapitel 5.4 definierten Anforderungen an die Unterrichtskonzeption (z.B. Anforderung A1 oder Anforderung B2). Ebenso wird mit der Bezeichnung Kompetenzbereich eine Kurzform für „Kompetenzbereich INIT“ vereinbart.

6.1 Überblick und grundlegende Annahmen

Ausgehend von den Rahmenbedingungen der schulspezifischen Implementierung des Kompetenzbereichs (Anforderung C11) erfolgt die Entwicklung der didaktischen Konzepte. Diese besagen, dass der Kompetenzbereich im 4. und 5. Jahrgang des Bildungsganges zu unterrichten ist. Nachdem aber die Erarbeitung der didaktischen Konzepte für beide Jahrgänge den Rahmen dieser Arbeit deutlich überschreiten würde, wird ausschließlich der 4. Jahrgang in Betracht gezogen. Bei diesem wiederum erfolgt eine Schwerpunktsetzung auf den Themenbereich „Mikrocontrollertechnik“ (μ C-Technik). Begründet wird diese Entscheidung damit, dass die μ C-Technik durchaus als „der“ zentrale Bestandteil des Kompetenzbereichs angesehen werden kann. Das zeigen nicht zuletzt auch die Inhalte der Bildungs- und Lehraufgaben für den Kompetenzbereich. Gemäß dieser Präzisierung erfolgt die weitere Ausarbeitung der didaktischen Konzepte. Hierbei wird zuerst der Frage nach dem „Was?“ und dann der Frage nach dem „Wie?“ nachgegangen (vgl. Kapitel 4.3).

6.1.1 Auswahl und Strukturierung der Lehrinhalte

Hinsichtlich der Konkretisierung der Lehrinhalte werden mehrere Einflussfaktoren in Betracht gezogen: Zum Einen sind das die Ergebnisse der Analyse der Ausgangssituation an der HTL Krems im Allgemeinen und des Vorwissens der SchülerInnen im Besonderen (siehe Kapitel 5.2). Zum Anderen die mit C1 und C4 festgelegten Erwartungen, welchen die Deskriptoren der Niveaus „Wiedergeben/Verstehen“ und „Anwenden“ der μ C-Technik zugrunde liegen. Und zu guter Letzt jene Erkenntnisse, die aus der Erhebung des State of the Art des Embedded System-Unterrichts resultieren (siehe Tabelle 5.10, Anforderungen des Bereichs „Embedded System als Unterrichtsgegenstand“). Basierend darauf werden drei aufeinander aufbauende Themenblöcke für die μ C-Technik gebildet (siehe Abbildung 6.1, Spalte „Themenblock“, B1, B2 u. B3).

Die inhaltliche Zielsetzung des jeweiligen Blocks kann wie folgt zusammengefasst werden:

- **B1:** Soll die SchülerInnen an den Kompetenzbereich heranführen und einen Überblick über die Wirkungsweise elementarer Hard- und Software-Komponenten des Unterrichts vermitteln.
- **B2:** Soll den SchülerInnen die grundlegende Funktionsweise sowie die einfache Anwendung (einer bestimmten Technologie) erschließen.
- **B3:** Soll den SchülerInnen die Funktionsweise und Anwendung ausgewählter μ C-Komponenten zugänglich machen, wie es die Entwicklung typischer Anwendungen¹ erfordert.

Ausschlaggebend für die inhaltliche Gestaltung der angeführten Themenblöcke B1 bis B3 ist die Anforderung C11, welche festlegt, dass die Durchführung des Kompetenzbereichs zweigeteilt - in einem theoretischen sowie praktischen Unterricht - zu erfolgen hat. Im Detail legt die Anforderung fest, dass der Theorieunterricht eine und der Praxisunterricht zwei Unterrichtseinheit(en) umfasst; wobei für eine Unterrichtseinheit 50 Minuten zur Verfügung stehen.

Damit wird auch der Anforderung D2 (drei „S“ - Science, Skills und State of the Art) genüge getan. Science - also die theoretischen Grundlagen - können somit in den Theorieeinheiten behandelt werden. Hierbei zielt die Durchführung vor allem auf die Entwicklung der Kompetenz „Wiedergeben/Verstehen“ (Anforderung C1) ab. Damit die Lernenden über die entsprechenden Skills - also die praktischen Fähigkeiten - verfügen, soll in den Übungseinheiten das theoretische Wissen in die Praxis überführt und vertieft werden, beginnend bei einfachen Übungsbeispielen bis hin zu Beispielen, die schlussendlich C4 gerecht werden. Die Zuordnung „Science in der Theorieeinheit“ und „Skills in der Übungseinheit“ ist jedoch keine strikte - diese soll den SchülerInnen ausschließlich eine Orientierung ermöglichen.

¹Als „typische“ μ C-Anwendung wird im Kontext der Industriellen Informatik eine Anwendung verstanden, die beispielsweise eine physikalische Größe erfasst, diese in eine digitale umwandelt und in Abhängigkeit der Programmlogik eine Aktionen ausführt (z.B. Displayausgabe, Ansteuerung von Aktoren etc.)

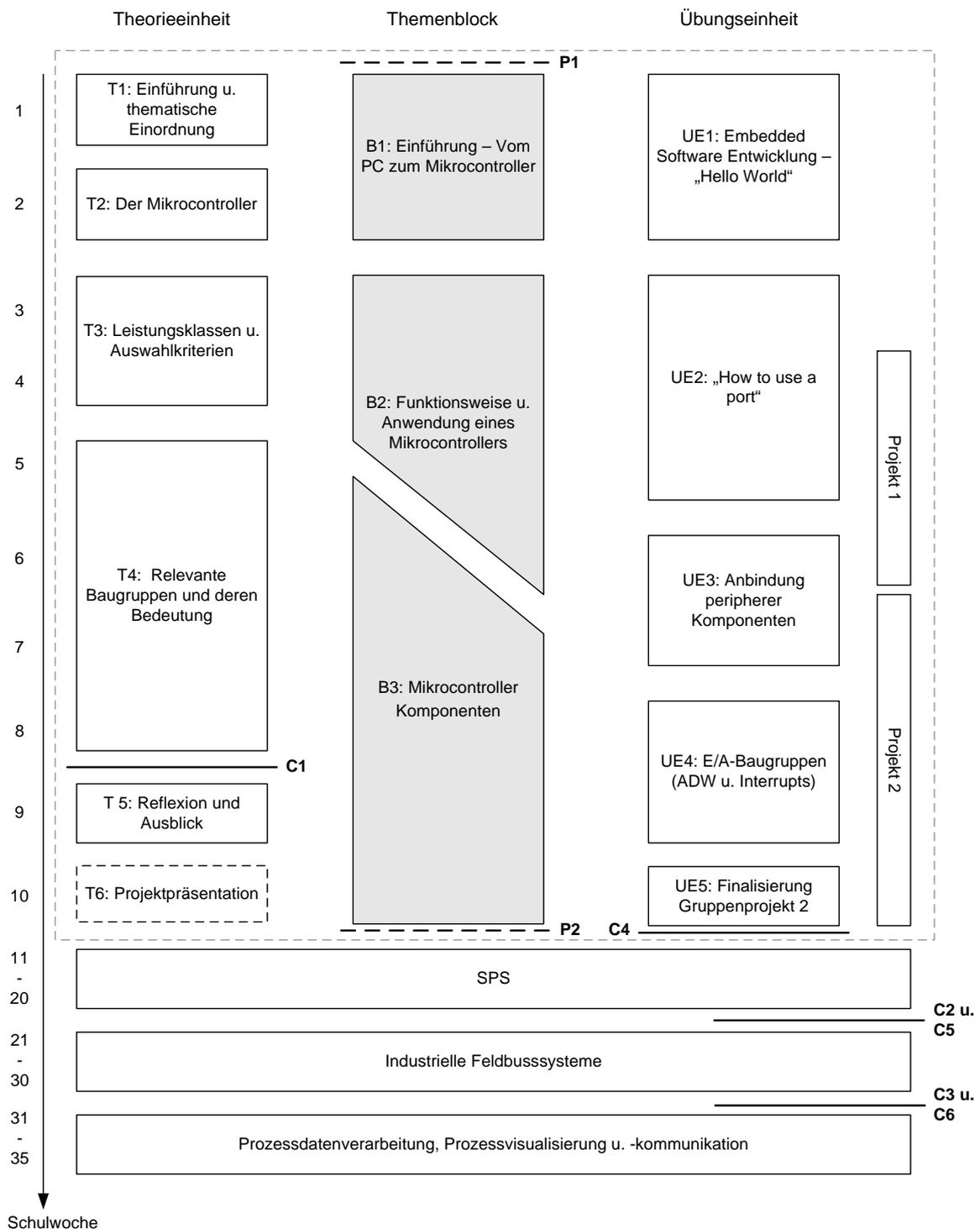


Abbildung 6.1: Übersicht der Unterrichtsplanung für den Kompetenzbereich Industrielle Informationstechnik, 4. Jahrgang einer HTL für Informationstechnologie

State of the Art - also das Wissen über aktuelle Technologien - ist sozusagen ein Querschnittsthema, das sowohl in den Theorie- als auch Übungseinheiten behandelt wird. In diesem Zusammenhang ist jedoch anzumerken, dass es vielmehr das Ziel ist, eine Technologie ganzheitlich zu durchdringen als mehrere oberflächlich zu behandeln (Anforderung C12).

Abschließend gilt es noch anzumerken, dass der grau umrahmte Bereich der Abbildung 6.1 jene Inhalte kennzeichnet, die dem Themenbereich μ C-Technik zuzuordnen sind und infolgedessen im Blickfeld der Ausarbeitung stehen. Die weiteren Themenbereiche des Kompetenzbereichs (SPS, Industrielle Feldbusse sowie Prozessdatenverarbeitung, -visualisierung und -kommunikation (PPuP)) werden nur der Vollständigkeit halber angeführt, um die Jahresplanung zu vollziehen. Diesen kommt zumindest bei der zuvor erwähnten Ausarbeitung keine weitere Bedeutung zu.

6.1.2 Durchführung und Zielsetzung

Allgemeines Ziel ist, dass die Lernenden am Ende des Bildungsganges die Anforderungen C1 bis C9 erfüllen. Es sei nochmals festgehalten, dass C1 bis C6 die Zielvorgaben der Kompetenzdeskriptoren der Kategorien „Wiedergeben/Verstehen“ und „Anwenden“ sowie C7 bis C9 die Zielvorgaben der Kompetenzdeskriptoren der Kategorien „Analysieren“ und „Entwickeln“ enthalten. Somit wäre es naheliegend, dass die ersten zwei Kompetenzniveaus mit den Anforderungen C1 bis C6 im 4. Jahrgang und die verbleibenden zwei mit den Anforderungen C7 bis C9 im 5. Jahrgang erzielt werden. Dagegen spricht jedoch die Vorgabe des Lehrplans. Diese verlangt, dass der Themenblock „Prozessdatenverarbeitung, -visualisierung und -kommunikation“ (PPuP) ausschließlich im 4. Jahrgang zu unterrichten ist (siehe Kapitel 5.1.1, Bildungs- und Lehraufgaben). Jedoch wäre der einzige Kompetenzdeskriptor, der diesem Themenbereich zuzuordnen ist - und zwar jener, der C7 zugrunde liegt -, gemäß den zuvor getätigten Annahmen erst für den 5. Jahrgang vorgesehen, was zu einem Konflikt mit den Vorgaben des Lehrplans führen würde. Inwieweit hier schulautonome Verschiebungen möglich sind, geht nicht hervor.

Um diesen Konflikt zu vermeiden, würde es sich anbieten, C7 neben C1 bis C6 bereits im 4. Jahrgang zu berücksichtigen. Somit wäre den Bestimmungen des Lehrplans genüge getan. Es stellt sich jedoch die Frage, ob das dann nicht zu einer Überforderung der Lernenden führen würde? Denn der Kompetenzdeskriptor, der der Anforderung C7 zugrunde liegt, formuliert überaus anspruchsvolle Inhalte. Alternativ könnte man C4 und C5, die sich auf die μ C- und SPS-Technik im Kompetenzniveau „Anwenden“ fokussieren, in den 5. Jahrgang verschieben. Dies wäre hinsichtlich des Lehrplans gerechtfertigt, erscheint aber bei genauerer Betrachtung als nicht zielführend. Denn die Anforderungen C8 und C9 (vgl. Zielvorgaben des Kompetenzdeskriptors „Entwickeln“) verlangen die „Entwicklung von netzwerk- und echtzeitfähiger Systemen im industriellen Umfeld“. Dieses Kompetenzniveau setzt durchaus ausgeprägtes Know-how im Bereich der SPS- und/oder μ C-Technik voraus. Folglich muss die Erreichung von C4 und C5 aber auch von C6 im Laufe des 4. Jahrgangs bewerkstelligt werden, da andernfalls im 5. Jahrgang zu wenig Zeit für C8 und C9 bliebe. Vor allem wenn man bedenkt, dass der Regelunterricht des 5. Jahrgangs aufgrund der Reife- und Diplomprüfung Ende April endet.

Diesem Gedanken folgend, sollte es das Ziel sein, die Anforderungen C1 bis C6 im ersten

Jahr dieses Kompetenzbereichs umzusetzen (siehe Abbildung 6.1). Das würde bedeuten, die Themenbereiche μ C-Technik, SPS und Industrielle Feldbusse zumindest bis zum Kompetenzniveau „Anwenden“ zu vermitteln. Um den Anforderungen des Lehrplans gerecht zu werden, könnten sich, wie in Abbildung 6.1 dargestellt, die letzten Wochen der PPU_P widmen; jedoch nicht mit dem Endziel, die Anforderung C7 gänzlich zu realisieren. Dies könnte aufgrund zuvor angeführten Bedenken in Kombination mit den verbleibenden Anforderungen C8 und C9 (Kompetenzniveau „Entwickeln“) im 5. Jahrgang, Themenbereich „Entwicklung und Implementierung von Systemen für spezifische Anforderungen“, erfolgen.

Um zur Entwicklung der Sozial- und Personalkompetenz (Anforderung C10) einen Beitrag zu leisten, dienen einerseits ausgewählte Unterrichtssequenzen der Theorie (siehe Kapitel 6.2) sowie andererseits das Projekt 2 (siehe Kapitel 6.4). Projekt 1 ist als Einzelarbeit geplant. Hintergrund liefert Anforderung D5 (Förderung der individuellen Lernzielerreichung). Generell ist vorgesehen, dass die zwei Projekte parallel zum Regelunterricht abgehalten werden - Projekt 1 im Kontext des Kompetenzbereichs und Projekt 2 interdisziplinär mit zum Beispiel dem Kompetenzbereich Projektmanagement. Für eine detaillierte Beschreibung der Projekte siehe Kapitel 6.4. Der abschließende Absatz dieses Abschnitts diskutiert grundlegende Annahmen hinsichtlich der Jahresplanung des Kompetenzbereichs im 4. Jahrgang.

In der Regel kann von 44 Kalenderwochen in einem Schuljahr ausgegangen werden. Abzüglich diverser Ferien zu Weihnachten, Semesterwechsel oder Ostern reduziert sich die Anzahl auf rund 40 Wochen. Werden dann auch noch Schulveranstaltungen (z.B. Schikurs, Exkursionen etc.) oder mögliche Feiertage miteinbezogen - was natürlich vom Unterrichtstag abhängt - ergibt sich eine Nettoanzahl von circa 35 Schulwochen (SW), welche auch als Planungsgrundlage herangezogen wird. Mit dieser konservativen Schätzung sollte auch noch ausreichend Spielraum für Prüfungen, zur Reflexion (Anforderung B5) oder aber auch für Unerwartetes gegeben sein. Zur Durchführung der zuvor definierten Themenblöcke der μ C-Technik wird ein Zeitraum von 10 SW veranschlagt. Dies bedeutet aber nicht, dass die letzte μ C-Technikeinheit zwangsläufig in der SW 10 stattfinden muss - es soll vielmehr aufgezeigt werden, wie sich die Entwicklung der geforderten Kompetenzen innerhalb dieses Zeitraums realisieren lässt. Das verbleibende Zeitbudget wird gemäß den zuvor getätigten Annahmen auf die anderen Themenbereiche SPS, Industrielle Feldbusse und PPU_P aufgeteilt.

6.2 Theorieeinheiten

Dieser Abschnitt fasst die Theorieeinheiten der μ C-Technik zusammen. Es werden insgesamt vier Einheiten im Detail vorgestellt. Die Ausarbeitung dieser Einheiten folgt einem einheitlichen Schema, das sich wie folgt gestaltet:

- Übersicht und Eckdaten
- Aufgabenstellung
- Lösungsvorschläge
- Anmerkungen

Der Bereich „Übersicht und Eckdaten“ fasst Wesentliches der jeweiligen Unterrichtseinheit zusammen. Unter „Aufgabenstellung“ sind jene Inhalte angeführt, mit welchen die SchülerInnen konfrontiert werden. Demnach wird bei der Formulierung der Inhalte ein zielgruppengerechter Schreibstil eingesetzt. Enthält eine Unterrichtseinheit ein sogenanntes Ad-ditum (vgl. Kapitel 4.4.2) oder ist eine Hausübung Bestandteil der Aufgabenstellung, erfolgt ein entsprechender Hinweis. Lösungsvorschläge sind im gleichnamigen Bereich enthalten. Diese sollen jedoch ausschließlich eine Richtung vorgeben, die aufgrund der allgemeinen Zielsetzung des Gesamtkonzeptes erforderlich ist, sodass den definierten Anforderungen genüge getan wird. Erwähnenswertes ist im Punkt „Anmerkung“ angeführt.

Der Theorieunterricht umfasst neben diesen vier Unterrichtseinheiten auch noch die Einheiten „T5 - Reflexion und Ausblick“ (T5) sowie „T6 - Projektpräsentation“ (T6), welche im Folgenden zusammengefasst dargestellt werden.

Die Einheit T5 zielt primär auf eine gemeinsame Reflexion ab, bei welcher die SchülerInnen ihre Erfahrungen aber auch Ideen, die sie im Laufe des Unterrichts sammeln, austauschen. Beispielsweise könnte über weiterführende Projektideen diskutiert werden - also eine gemeinsame Kreativphase. Vor allem die Ideenentwicklung kann der Lerhkraft als Indikator dienen, inwieweit die μ C-Technik ansich oder einzelne Themenbereiche im Speziellen verinnerlicht wurden. Nachdem es parallel zu den Übungseinheiten zwei Projekte zu absolvieren gilt, soll die Einheit T6 dazu dienen, die Ergebnisse des Projektes 2 (siehe Kapitel 6.4) vorzustellen. Dies ist als motivationsfördernde Maßnahme zu verstehen. Da es sich um Gruppenprojekte handelt, sollte das Zeitbudget von 50 Minuten reichen.

Allgemeines Ziel bei der Konzeption der Theorieeinheiten ist es, die zu behandelnden Themenschwerpunkte soweit einer didaktischen Reduktion zu unterziehen, dass die Inhalte unter den gegebenen Rahmenbedingungen behandelt werden können. Nachdem sich einige Unterrichtssequenzen über zwei oder mehrere Wochen erstrecken, ist beispielsweise der Punkt „Aufgabenstellung vorstellen und ins Thema einführen“ (siehe Abschnitt „Übersicht und Eckdaten“, Punkt „Durchführung“ der jeweiligen Sequenz) als Pauschalannahme zu verstehen, die bedarfsgemäß auf die einzelnen Einheiten aufzuschlüsseln ist. Administrative Aufgaben der Lehrkraft sind ebenfalls diesem Punkt zuzuordnen.

6.2.1 T1 - Einführung und thematische Einordnung

Die Inhalte des Kompetenzbereichs stellen größtenteils Neuland dar. Folglich ist die thematische Einführung das erklärte Ziel dieser Unterrichtseinheit. Anhand einer Internet-Recherche sollen die SchülerInnen eine allgemeine Idee vom Kompetenzbereich erhalten, die in weiterer Folge im Zuge einer Gruppendiskussion verdichtet und konkretisiert wird.

Übersicht und Eckdaten

Bezeichnung	T1 - Einführung und thematische Einordnung
Themenbereiche	Merkmale der Industriellen Informationstechnik/Automation, relevante Begriffe u. Zusammenhänge
Adressierte Anforderungen	B1, B4, C1 u. C10
Unterrichtsmethoden	LehrerInnenvortrag (V), Einzelarbeit (E) u. Gruppendiskussion (G)
Durchführung	Aufgabenstellung vorstellen und ins Thema einführen (V, ca. 5 Min.), Durchführung der Internetrecherche (E, ca. 15 Min.) und der Gruppendiskussion (G, ca. 25 Min.)
Erforderliche Ressourcen	Internet, Rechner u. didaktisch aufbereitete Arbeitsblätter
Dauer	1 Unterrichtseinheit (50 Min.)
Quellen	-

Aufgabenstellungen

Arbeitsblatt, mit folgendem Inhalt:

Lernziele:

Was sollen Sie nach diesem Unterrichtsblock können?

- Sie wissen Bescheid über die wesentlichen Merkmale der Industriellen Informationstechnik und können diese auch wiedergeben.
- Sie können relevante Begriffe der Industriellen Informationstechnik einordnen und erkennen deren Zusammenhänge.

Herzlich willkommen in der Welt der Industriellen Informationstechnik (INIT). Nachdem Sie nun wissen, was das Ziel dieser Unterrichtseinheit ist, sind Sie an der Reihe.

Wie Sie im Zuge der nachfolgenden Recherche feststellen werden, handelt es sich bei der INIT um einen überaus vielseitigen Fachbereich, der unterschiedliche Betrachtungsweisen ermöglicht. Uns interessiert vor allem jene des Informatikers/der Informatikerin. Damit Sie eine grobe Vorstellung haben, wohin die Reise geht, dient nachfolgende Abbildung 6.2, die eine sogenannten Schlagwortwolke (engl. *tag cloud*) zeigt mit Begriffen, die in Zusammenhang mit der INIT stehen.

 **Aufgabenstellung T1.1:** Führen Sie unter Zuhilfenahme des Internets eine Recherche zum Thema INIT durch. Finden Sie hierbei (i) die Bedeutung der angeführten Begriffe heraus und überlegen Sie, wie diese (ii) untereinander in Beziehung stehen könnten. Bereiten Sie sich mit Notizen für die nachfolgende Gruppendiskussion vor.



Abbildung 6.2: Schlagwortwolke mit Begriffen zur Internet-Recherche

Lösungsvorschläge

Nach Abschluss der Einzelarbeit ist es Aufgabe der Lehrkraft, Ordnung in das Gedanken- gebilde der SchülerInnen zu bringen. Dies erfolgt in einer von der Lehrkraft moderierten Gruppendiskussion. Ergebnis dieser Diskussion könnte beispielsweise eine ergänzte Schlag- wortwolke sein, wie in Abbildung 6.3 dargestellt.

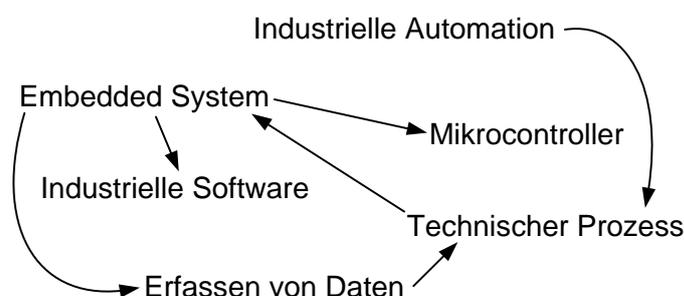


Abbildung 6.3: Schlagwortwolke mit in Beziehung gesetzten Begriffen zur Industriellen Informationstechnik

Grundsätzlich kann die Industrielle Automation als ein Anwendungsgebiet der INIT ge- sehen werden. Hierbei ist es das Ziel, einen technischen Prozess zu automatisieren. Dazu können sogenannte „Embedded Systems“ eingesetzt werden, welche wiederum einen „Mi- krocontroller“ als zentrale Hardwarekomponente aufweisen. Um die spezielle Aufgabe des technischen Prozesses zu realisieren, ist weiters eine Software-Komponenten - „Industrielle Software“ - erforderlich. Das Erfassen von beispielsweise physikalischen Größen oder Daten jeglicher Art zählt zu den typischen Aufgaben eines Embedded Systems in der Industriellen Automation. Diese Daten stehen in direktem Zusammenhang mit dem technsichen Prozess.

Je nach Qualität der Diskussionsbeiträge kann die Schlagwortwolke um weitere Begriffe ergänzt werden, sodass schlussendlich alle identifizierten Begriffe von den SchülerInnen zu- geordnet werden können. Auf jeden Fall sollte am Ende der Diskussion eindeutig feststehen, dass

- Embedded Systems für eine bestimmte Aufgabe entwickelt werden und nicht zwangs- läufig über eine herkömmliche Benutzerschnittstelle verfügen müssen wie beispielsweise herkömmliche PC-Systeme.

- besondere Anforderungen an diese Systeme im Allgemeinen (z.B. Robustheit des Gehäuses) und
- an die Software im Speziellen (z.B. Echtzeitverhalten, oder allgemeiner Vergleich zu Bürosoftware) gestellt werden.

Ergänzend sollte ausgewähltes Bildmaterial zu einem einfachen technischen Prozess oder beispielhaften Embedded System vorhanden sein (z.B. in Form eines Handouts).

Anmerkungen

Aufgrund der knapp bemessenen Unterrichtszeit erscheint als sinnvoll, die Recherche anhand einiger Begriffe (siehe Schalgwortwolke, Abbildung 6.2) in die gewünschte Richtung zu lenken. Alternativ könnten diese auch weggelassen und ausschließlich ein zentraler Suchbegriff vorgegeben werden. Das Arbeitsblatt selbst sollte einen Bereich für die eigenen Notizen und zur gemeinsamen Erarbeitung der Lösung aufweisen.

6.2.2 T2 - Der Mikrocontroller

Die SchülerInnen wissen über den Aufbau und die Funktionsweise herkömmlicher PC-Systeme Bescheid (siehe Vorbedingungen, Abbildung 5.4). Was ein Embedded PC-System ist, sollte aufgrund der vorhergehenden Einheit T1 (siehe Kapitel 6.2.1) bekannt sein. Diese Unterrichtseinheit hat zum Ziel, den prinzipiellen Aufbau eines Mikrocontrollers anhand klassischer Anwendungsbeispiele zu vermitteln.

Übersicht und Eckdaten

Bezeichnung	T2 - Der Mikrocontroller
Themenbereiche	Mikrocontrolleraufbau u. -merkmale, Anwendungsbeispiel, konkrete Mikrocontrollertechnologie
Adressierte Anforderungen	A1, B1, B4, C1 u. C10
Unterrichtsmethoden	LehrerInnenvortrag (V), Einzelarbeit (E) u. Partnerarbeit (P)
Durchführung	Aufgabenstellungen vorstellen und ins Thema einführen (V, ca. 5 Min.), Aufgabenstellung 2.1 (E, ca. 10 Min.), Aufgabenstellung 2.2 (E, ca. 10 Min.) und Aufgabenstellung 2.3 (P, ca. 15 Min.)
Erforderliche Ressourcen	Didaktisch aufbereitete Arbeitsblätter
Dauer	1 Unterrichtseinheit (50 Min.)
Quellen	[83, 84, 85]

Aufgabenstellungen

Arbeitsblatt 1, mit folgendem Inhalt:

Lernziele:

Was sollen Sie nach diesem Unterrichtsblock können?

- Sie wissen, wozu man einen Mikrocontroller grundsätzlich benötigt.
- Sie verstehen dessen prinzipiellen Aufbau im Vergleich zu einem PC-System und können die einzelnen Komponenten wiedergeben.
- Sie kennen erste Mikrocontroller-Komponenten (I/O-Modul mit A/D-Wandler) und verstehen das Zusammenwirken anhand eines konkreten Produktes.

Es stellt sich die Frage „Was ist ein Mikrocontroller und worin unterscheidet sich dieser von einem Mikroprozessor²“?

Wie Sie aus vorangegangenen Schuljahren sicherlich noch wissen, verfügt jeder herkömmliche PC (vgl. Home-PC) über einen Mikroprozessor. Wenn Sie nun auf solch einem PC ein Programm starten - in der Fachsprache auch als „ausführen“ bezeichnet - wird das auszuführende Programm - oder zumindest relevante Programmteile davon - in den Hauptspeicher geladen und dessen Anweisungen vom Mikroprozessor Schritt für Schritt abgearbeitet. Daten, die durch das Abarbeiten der Programmanweisungen zur Laufzeit erzeugt werden, werden entweder in einem Register des Mikroprozessors, im Cache-Speicher, um die Zugriffszeit zu verkürzen, oder eben im Hauptspeicher abgelegt. Werden bestimmte Daten über die Laufzeit des Programms hinweg benötigt, gilt es diese dauerhaft abzuspeichern. Hierfür werden in der Regel Festplatten verwendet. Damit die einzelnen Komponenten untereinander zusammenwirken können, sind sogenannte Bussysteme vorhanden.

Wie Sie außerdem seit der vorangegangenen Unterrichtseinheit wissen, kommt den Abmessungen eines Embedded Systems eine gewisse Bedeutung zu. Beispielsweise legen die Rahmenbedingungen fest, dass ein zu entwickelndes System nicht größer als eine handelsübliche Zündholzschachtel sein darf, funktional aber den Ansprüchen eines Computers gerecht werden muss (z.B. Daten über Schnittstellen einlesen, verarbeiten, speichern u. ggf. an ein anderes Computer-System übermitteln). Nun ja, dass man keine Hauptplatine samt Bestückung, Festplatte und Netzteil in eine Zündholzschachtel bekommt, bedarf keines weiteren Experiments, oder? Genau deswegen gibt es Mikrocontroller. Ein Mikrocontroller ist grundsätzlich ein vollwertiger Computer, der die wesentlichen Funktionseinheiten auf engstem Raum integriert; deswegen auch die Bezeichnung „Ein-Chip-Mikrocomputer“.

 **Aufgabenstellung T2.1:** Sie kennen die Komponenten eines herkömmlichen PC-Systems und wissen über deren Zusammenwirken Bescheid. Basierend darauf vervollständigen Sie das Blockschaltbild eines Mikrocontrollers (siehe Abbildung 6.4). Es sollte hervorgehen, wo im Gesamtsystem eine Komponente positioniert ist und wie diese miteinander verbunden sind. Zeichnen Sie hierfür die Bussysteme ein und legen Sie die Kommunikationsrichtungen fest.

²Hinsichtlich des Vergleiches Mikrocontroller und Mikroprozessor ist klarzustellen, dass dieser genau genommen nicht zulässig ist - ein Mikrocontroller inkludiert einen Prozessor und ist somit ein Computersystem.

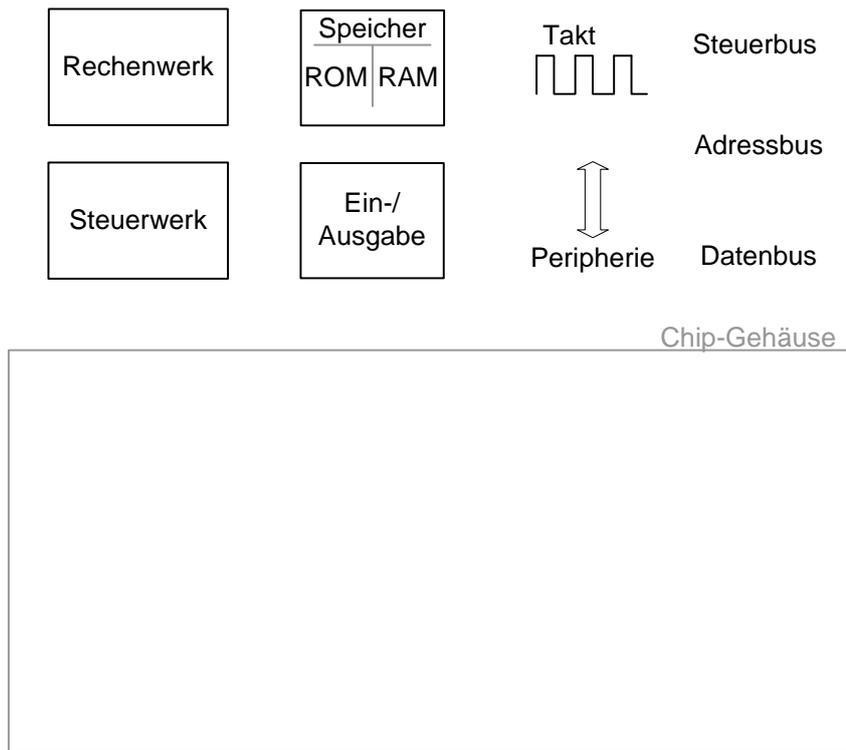


Abbildung 6.4: Blockschaltbild eines Mikrocontrollers

Arbeitsblatt 2, mit folgendem Inhalt:

Betrachten Sie die Abbildung 6.5. Diese zeigt eine typische Architektur am Beispiel des AVR-Mikrocontrollers *ATmega328P*. Hierbei handelt es sich um einen 8-Bit-Mikrocontroller, der über die wesentlichen peripheren Komponenten verfügt. Nachdem Sie bei der vorhergehenden Aufgabenstellung T2.1 das grundlegende Konzept eines Mikrocontrollers kennengelernt haben, gilt es dieses Wissen auf eine konkrete Realisierung umzulegen. Betrachten Sie hierfür die Beispiel-Architektur des AVR-Mikrocontrollers.

Aufgabenstellung T2.2: Ordnen Sie die Komponenten des AVR-Mikrocontroller-Blockschaltbildes (siehe Abbildung 6.5) den Funktionsbereichen (Mikroprozessor (RW u. SW), Speicher, Peripherie bzw. Ein-/Ausgabe) der Abbildung 6.4 zu. Jetzt stellt sich noch die Frage, worin sich eigentlich ein Mikrocontroller von einem Mikroprozessor unterscheidet, ist dieser oft gebrachte Vergleich eigentlich zulässig? Zusatzaufgaben: Welchem Architekturmodell folgen Mikrocontroller? Denken Sie in diesem Zusammenhang an eine überaus renommierte Universität.

Arbeitsblatt 3, mit folgendem Inhalt:

Ein typisches Anwendungsbeispiel für einen Mikrocontroller ist das Messen einer physikalischen Größe. Abbildung 6.6 zeigt ein mobile Wetterstation mit akustischem Signalgeber (Aktor). Diese misst mittels Sensoren die Lufttemperatur und -feuchtigkeit. Über- oder unterschreitet ein Wert den festgelegten Max- oder Minimalwert, wird eine akustische Warnmeldung ausgelöst. Die Sensoren und der Signalgeber sind physisch mit mehreren Pins des Mikrocontrollers - so werden dessen Anschlüsse bezeichnet - verbunden.

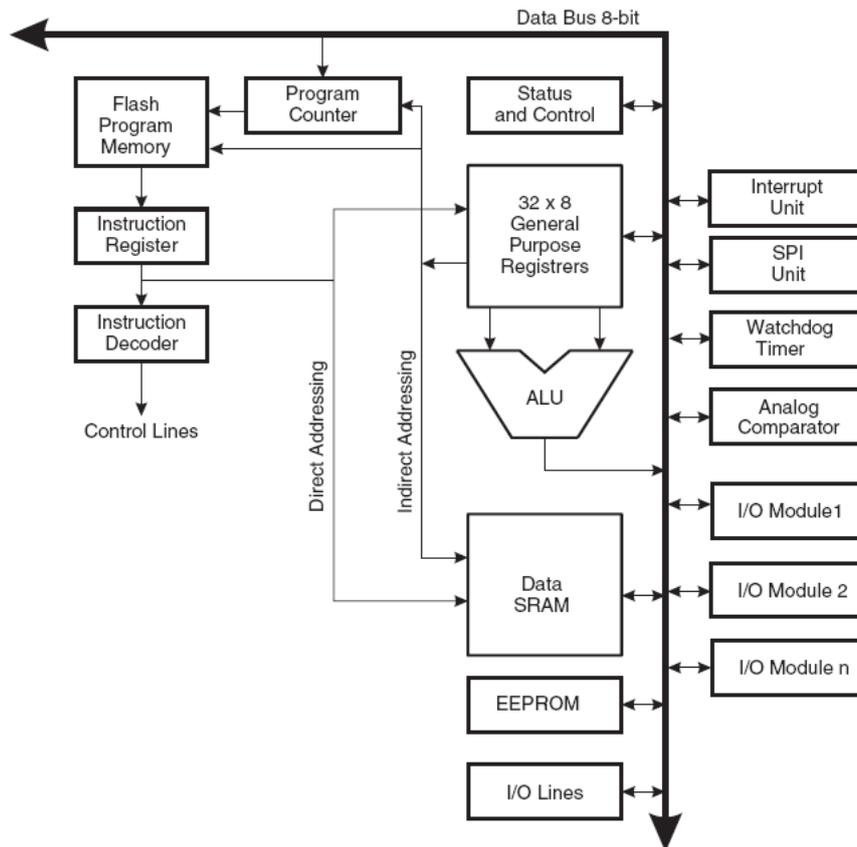


Abbildung 6.5: Blockschaltbild des AVR-Mikrocontrollers ATmega328P [85]

Aufgabenstellung T2.3: Diskutieren Sie mit einem Kollegen/einer Kollegin folgende Punkte: (i) Wie gestaltet sich Ihrer Meinung der logische Informationsfluss bei dieser Prinzipschaltung in Abbildung 6.6. Zeichnen Sie diesen mit Pfeilen ein. Daraus lässt sich doch schließen, dass ein Mikrocontroller zwischen Ein- und Ausgang bei den Pins unterscheidet. (ii) Vermerken Sie zusätzlich, bei welchen/welchem Pin(s) es sich daher um einen Ein- beziehungsweise Ausgang handeln muss. Wie Sie vielleicht noch aus „Grundlagen der Elektrotechnik“ wissen, unterscheidet man zwischen analogen und digitalen Signalen. (iii) Wo würden Sie ein analoges und wo ein digitales Signal vermuten - vor allem unter dem Aspekt, dass heutzutage jeder gebräuchliche Computer - also auch ein Mikrocontroller - ein Digitalrechner ist. Begründen Sie Ihre Annahmen.

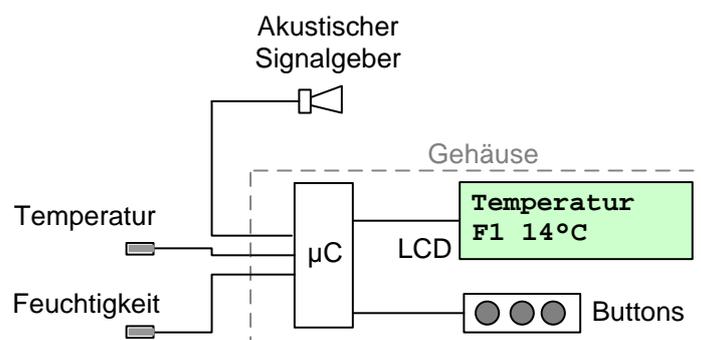


Abbildung 6.6: Anwendungsbeispiel eines Mikrocontrollers (Prinzipschaltbild)

Lösungsvorschläge

Bei der Aufgabenstellung T2.1. sind relativ konkrete Ergebnisse zu erwarten. Besonderes Augenmerk ist auf die Anbindung der Bussysteme und die Festlegung der Informationsflussrichtung zu legen. In diesem Zusammenhang ist von den SchülerInnen zu erkennen, dass der Informationsfluss auf dem Datenbus bei einem „Read-only“-Speicher unidirektional erfolgt, wie Abbildung 6.7 zeigt. Gleiches gilt für die Adressierung der Baugruppen. Hinsichtlich der zuvor angeführten Speicher gilt es zu ergänzen, dass bei aktuellen Mikrocontroller-Technologien der Schreib-/Lesespeicher als SRAM und der Festwertspeicher als EEPROM realisiert werden. In der Regel verfügen diese auch noch über einen sogenannten „Flash“-Speicher für den Programmcode.

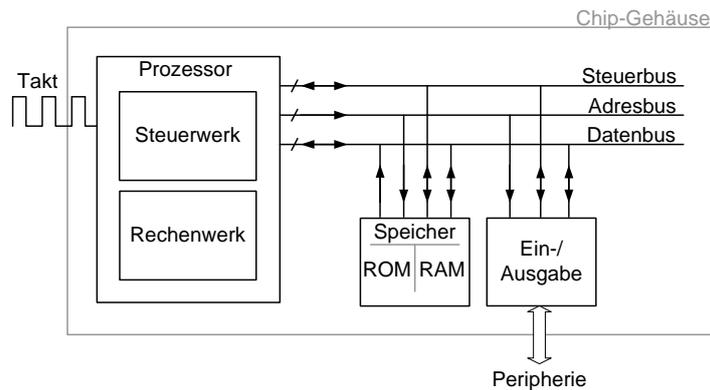


Abbildung 6.7: Blockschaltbild eines Ein-Chip-Mikrorechners (vgl. Mikrocontroller, Eigendarstellung nach [83])

Bei der Aufgabenstellung T2.2 sind unterschiedliche Ergebnisse zu erwarten. Die meisten werden CPU-spezifische Komponenten wie die ALU, deren Arbeitsregister oder das Befehlszeiger-Register erkennen und entsprechend zuordnen. Die zentrale Erkenntnis sollte aber sein, dass das von den SchülerInnen verfasste Programm im „Flash“-Speicher abgelegt wird (siehe Abbildung 6.6, logische Folge: Befehlszeiger, Flash-Speicher, Befehlsregister u. Befehlsdecodierer). Die Lehrkraft ist dann in weiterer Folge aufgefordert, die Zusammenhänge im Detail zu erläutern und eventuelle Unklarheiten zu beseitigen. Hinsichtlich des Vergleiches Mikrocontroller und Mikroprozessor ist klarzustellen, dass dieser genau genommen nicht zulässig ist - ein Mikrocontroller inkludiert einen „Prozessor“ und ist somit ein Computersystem. Alternativ könnte die Abbildung 6.7 dahingehend ergänzt werden, dass dieser ein PC-System im Detail gegenübergestellt wird, um letzte Zweifel zu beseitigen.

Bei der Aufgabenstellung T2.3 wird das Ziel verfolgt, dass die SchülerInnen weitere Mikrocontroller-Konzepte kennenlernen. Hierzu zählt die Unterscheidung zwischen Ein- und Ausgängen und dem entsprechenden Informationsfluss sowie der Unterscheidung zwischen analogen und digitalen Signalen (mit konkretem Verweis auf die Komponenten A/D-Wandler).

In diesem Zusammenhang sollte von den SchülerInnen am Ende der Unterrichtseinheit erkannt und verstanden werden,

- warum man ein μC als Ein-Chip-Mikrocomputer bezeichnet und aus welchen Komponenten dieser besteht.

- dass ein Vergleich Mikroprozessor vs. Mikrocontroller genau genommen nicht zulässig ist (mit Bezug auf Aufgabenstellung T2.2).
- dass μC überall dort eingesetzt werden können, wo „Intelligenz“ auf kleinstem Raum erforderlich ist.
- dass μC in der Regel eine einzige Aufgabe erfüllen im Gegensatz zu einem herkömmlichen PC.

Anmerkungen

Es ist zu erwarten, dass einige SchülerInnen vor allem bei den Aufgabenstellungen T2.2 und T2.3 Unterstützung benötigen. Die grundsätzliche Arbeitsweise eines Prozessors sollte hingegen bekannt sein. Insofern es die zeitlichen Rahmenbedingungen erlauben, kann eine diesbezügliche Wissensauffrischung erfolgen (z.B. mit einem Simulations-Tool oder einem Gruppenspiel (Anforderung A1), bei welchem SchülerInnen das Verhalten eines Prozessors nachspielen).

6.2.3 T3 - Leistungsklassen und Auswahlkriterien

Diese Unterrichtssequenz soll Bewusstsein bei den SchülerInnen dahingehend bilden, dass es eine Vielzahl unterschiedlicher Mikrocontroller-Familien/Typen gibt. Es werden hierbei relevante Auswahlkriterien in Erfahrung gebracht, anhand welcher die Typauswahl erfolgen kann. Da sich diese Sequenz über zwei Unterrichtseinheiten erstreckt, erfolgt eine geteilte Durchführung.

Übersicht und Eckdaten

Bezeichnung	T3 - Leistungsklassen und Auswahlkriterien
Themenbereich	Mikrocontroller-Technologien, Bauteilauswahl
Adressierte Anforderungen	B1, B2, B3, C1, C4 u. C10
Unterrichtsmethoden	LehrerInnenvortrag (V), Einzelarbeit (E) evt. Partnerarbeit (P)
Durchführung	1. Teil: Aufgabenstellungen vorstellen u. ins Thema einführen (V, ca. 5 Min.), Aufgabenstellung 3.1 (E, ca. 20 Min.) u. Aufgabenstellung 3.2 (E, ca. 20 Min.) 2. Teil: Aufgabenstellung 3.3 (E, ca. 45 Min.)
Erforderliche Ressourcen	Internet, PC
Dauer	2 Unterrichtseinheiten (100 Min.)
Quelle	[86]

Aufgabenstellungen

Arbeitsblatt, mit folgendem Inhalt:

Lernziele:

Was sollen Sie nach diesem Unterrichtsblock können?

- Sie verfügen über konkretes Produktwissen im Bereich der Mikrocontroller-Technik.
- Sie kennen zentrale Anforderungen, die bei der Auswahl eines Mikrocontrollers von Bedeutung sind.
- Sie wissen über die Bedeutung der Bauteilerauswahl Bescheid und können diesbezügliche Begriffe einordnen.

Die Qual der Wahl! So könnte man die riesige Anzahl der am Markt verfügbaren Mikrocontroller zum Ausdruck zu bringen.

Angenommen, Ihnen wird die Aufgabe übertragen, einen Prototypen zu entwickeln. Da Sie der Meinung sind, dass das Projekt den Prototypen-Status nie und nimmer überschreiten wird, machen Sie sich keine all zu großen Gedanken bei der Bauteilerauswahl. Eine gewisse Zeit später, nachdem es der Prototyp wider Erwarten doch zum fertigen Produkt geschafft hat und mittlerweile rund 1000 Stk./Monat produziert und verkauft werden, wird Ihnen klar, dass keiner Ihrer KollegInnen die seinerzeitige Auswahl des Mikrocontrollers infrage gestellt hat. Ihnen jedoch wird bewusst, dass es eigentlich Controller gibt, die dasselbe können, jedoch um die Hälfte kleiner und vor allem um 5 Euro je Stück günstiger sind. Am besten, Ihr Chef erfährt das nie!

 **Aufgabenstellung T3.1:** Angenommen, Sie planen den Kauf eines Notebooks. Welche Ausstattung (z.B. Schnittstellen) ist Ihnen wichtig und anhand welcher Kenngrößen treffen Sie Ihre Auswahl? Fertigen Sie eine Tabelle und tragen Sie aktuelle Werte (z.B. Arbeitsspeicher: 4GB) ein. Nachdem Sie die Tabelle erstellt haben, überlegen Sie noch kurz, wieviel Speicherplatz eine Office-Anwendung oder am besten das gesamte Office-Paket in Anspruch nimmt.

Mit diesem Wissen gilt es nun herauszufinden, inwieweit sich die Ausstattung oder die Kenngrößen eines Laptops mit jenen eines Mikrocontrollers decken.

 **Aufgabenstellung T3.2:** Verschaffen Sie sich unter Zuhilfenahme des Internets einen Überblick bezüglich der am Markt verfügbaren Mikrocontroller (max. drei Anbieter: z.B. AVR von Atmel o. PIC von Microchip). Identifizieren Sie hierbei die wesentlichen Ausstattungsmerkmale, über die Mikrocontroller allgemein verfügen, und vor allem wichtige Kenngrößen (z.B. RAM/ROM-Speicher). Welche Merkmale erscheinen Ihnen als überaus wichtig bei der Auswahl und wovon machen Sie das primär abhängig? Definieren Sie die Begriffe „Leistungsklassen“ und „Familie“ im Kontext der Mikrocontroller. Wie sieht es mit der Installation von SW-Paketen aus, die hinsichtlich der Größe einer Office-Anwendung gleichkommen?

Nachdem Sie über einen groben Überblick bezüglich Mikrocontroller-Leistungsklassen und -Familien verfügen, gilt es das Wissen anzuwenden. Bearbeiten Sie nachfolgende Aufgabenstellung T3.3 als Hausübung.

 **Aufgabenstellung T3.3:** Entwerfen Sie ein (fiktives) Anwendungsbeispiel und zeichnen Sie hierfür ein Prinzipschaltbild (vgl. Unterricht). Wählen Sie einen Mikrocontroller aus und begründen Sie Ihre Entscheidung (max. eine A4-Seite).

Lösungsvorschläge

Alle Aufgabenstellungen ansich bieten viel Spielraum für individuelle Lösungsansätze. Aufgabenstellung T3.3 erfordert außerdem noch ein erhebliches Maß an Kreativität. Es zeigt sich, inwieweit der jeweilige Schüler/die jeweilige Schülerin die Inhalte der ersten Unterrichtseinheiten verinnerlicht hat. Das Augenmerk bei der Aufgabenstellung T3.3 liegt vor allem darauf, ob die SchülerInnen imstande sind, einerseits eine plausible Aufgabenstellung zu definieren und andererseits, die wesentlichen Anforderungen zu identifizieren. Als Mikrocontroller-Leistungsmerkmale können beispielsweise die Wortbereite oder andere Architekturmerkmale sowie technische Merkmale (z.B. Taktfrequenz, Energiebedarf o. Abwärme) aber auch Merkmale des Speichers und der Peripherie (z.B. serielle und parallele Ein- u. Ausgabe, A/D-Wandler o. Interrupt) herangezogen werden. Ebenfalls von Bedeutung sind ökonomische Merkmale (z.B. Preis, Verfügbarkeit o. Support).

Allgemein sollte von den SchülerInnen erkannt werden, dass Mikrocontroller hinsichtlich möglicher Hardware- oder Software-Erweiterungen im Gegensatz zu PC-Systeme nicht annähernd so flexibel sein müssen. Das gilt für Mikrocontroller generell, denn sie müssen ausschließlich für eine einzige Aufgabe gewappnet sein. Das ist mit ein Grund für den um ein Vielfaches kleineren (flüchtigen als auch nicht-flüchtigen) Speicher bei Mikrocontrollern. Außerdem kommt der Benutzerinteraktion in der Regel keine so große Bedeutung zu wie bei PC-Systemen. Genauso ist die Rechenleistung an sich nicht unbedingt das Kriterium bei Mikrocontrollern. Weitere typische Anforderungen an Mikrocontroller sind ein gutes Echtzeitverhalten, wodurch diese über keine sogenannten „Cache“-Speicher verfügen, sowie ein geringer Energieverbrauch.

In diesem Zusammenhang sollte von den SchülerInnen am Ende der Unterrichtseinheit erkannt und verstanden werden, dass

- sich Mikrocontroller erheblich von herkömmlichen PC-Systemen unterscheiden. Dies ist auf die unterschiedlichen Anforderungen zurückzuführen, die aus dem jeweiligen Einsatzzweck resultieren.
- bei der Auswahl eines Mikrocontrollers andere Faktoren in Betracht zu ziehen sind.
- es eine Vielzahl von Mikrocontroller-Herstellern mit unterschiedlichen Familien gibt.
- die Bauteilwahl gut überlegt sein muss (siehe einleitendes Praxisbeispiel).

Anmerkungen

Die Aufgabenstellung T3.1 ist ohne Internet zu bewältigen, sodass die SchülerInnen zum Reflektieren von bereits gelernten Inhalten angehalten sind. Treten bei der Durchführung erhebliche Wissenslücken (bei einzelnen SchülerInnen) auf, kann die Sozialform Partnerarbeit eingesetzt werden. Aufgabenstellung T3.2 zielt auf die Förderung der Recherchekompetenz bei den SchülerInnen ab.

Die Aufgabenstellung T3.1 und T3.2 gilt es im Unterricht (der ersten Unterrichtseinheit) zu bewältigen. Aufgabenstellung T3.3 ist als Hausübung geplant. Die Ergebnisse (eine A4-Seite) werden dann in der zweiten Unterrichtseinheit von ausgewählten SchülerInnen vorgestellt. Die Auswahl könnte beispielsweise mit einem Zufallsgenerator erfolgen. Somit wird nicht zuletzt der Spielwitz angeregt.

6.2.4 T4 - Relevante Baugruppen und deren Bedeutung

Diese Unterrichtssequenz verfolgt das Ziel, dass die SchülerInnen selbstständig das Wissen bezüglich der Bedeutung und der Funktionsweise relevanter Funktionseinheiten/Baugruppen eines Mikrocontrollers erarbeiten, bewerten und dieses wiederum transferieren. Zu diesen Baugruppen zählen der Speicher, der A/D-Wandler, das Konzept der Interrupts sowie der zeitgeberbasierten Einheiten. Die Durchführung dieser Unterrichtssequenz erstreckt sich über vier Unterrichtseinheiten, bei welcher unterschiedliche Methoden zum Einsatz kommen.

Übersicht und Eckdaten

Bezeichnung	T4 - Relevante Baugruppen und deren Bedeutung
Adressierte Anforderungen	B1, B2, B3, B5, C1, C4 u. C10
Unterrichtsmethoden	LehrerInnenvortrag (V), Einzelarbeit (E) u. Gruppenarbeit (G)
Durchführung	Aufgabenstellungen vorstellen und ins Thema einführen (V, ca. 20 Min.), Aufgabenstellung 4.1 (E, ca. 45 Min.), Aufgabenstellung 4.2 (E, ca. 45 Min.), Aufgabenstellung 4.3 (E, ca. 45 Min.) u. Aufgabenstellung 4.4 (G, ca. 45 Min.)
Erforderliche Ressourcen	Internet, PC u. didaktisch aufbereitete Arbeitsblätter
Dauer	4 Unterrichtseinheiten (200 Min.)
Quellen	[86]

Aufgrund des komplexen Aufbaus dieser Unterrichtssequenz erfolgt eine ergänzende Darstellung des Punktes „Durchführung“. Wie Abbildung 6.8 zeigt, gliedert sich der Ablauf in vier Phasen; wobei jede Phase einer Unterrichtseinheit (50 Min.) entspricht. In Phase 1 werden die SchülerInnen gemäß der Sozialform Gruppenpuzzle (siehe Kapitel 4.4.2) eingeteilt und erhalten den Arbeitsauftrag. Dieser umfasst die Literaturrecherche zu einer der einleitend angeführten Themenbereiche. Die SchülerInnen sind angehalten, sich umfangreiche Notizen

zu machen mit dem Ziel, das erworbene Wissen zu transferieren. In der Phase 2 („Wissenscheck anhand von Kontrollpunkten“) erhalten alle Expertengruppen einen Fragenkatalog, der sozusagen als Wissenscheck dient, ob die Erkenntnisse der Recherche ausreichend sind oder nicht. Der Lehrkraft obliegt es weiters, sich ein persönliches Bild vom Wissensstand der jeweiligen Gruppen beziehungsweise SchülerInnen zu machen. Je nachdem, wie das Ergebnis dieser Selbstevaluation ausfällt, können ergänzende Arbeitsunterlagen bereitgestellt werden. Hierbei erscheint es als zielführend, keine Kopien aus Fachbüchern zu verteilen, sondern Unterlagen, die bereits verdichtete Informationen enthalten, um den/die SchülerIn an das Wesentliche heranzuführen.

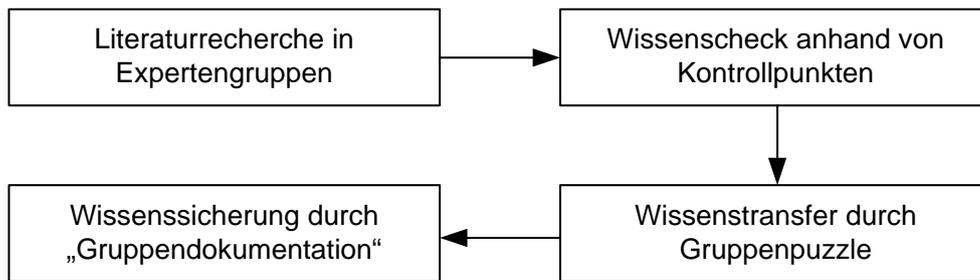


Abbildung 6.8: Vier Phasen der Unterrichtssequenz „T4 - Relevante Baugruppen und deren Bedeutung“

Bei der Phase 3 erfolgt der Wissenstransfer in den sogenannten Unterrichtsrunden. Hierbei lehren die jeweiligen ExpertInnen die NovizInnen. Diese Phase soll auch zur Selbstreflektion bei den Experten dienen. In der letzten Einheit dieser Unterrichtssequenz steht den ExpertInnen nochmals ein wenig Zeit zum Überarbeiten der Dokumentation zur Verfügung, die schlussendlich in der jeweiligen Expertengruppe zu einem Kompendium zusammenzufassen ist. Dieses soll dann im Laufe der Übungseinheiten als Nachschlagewerk dienen.

Aufgabenstellungen

Die Aufgabenstellungen dieser Unterrichtssequenz werden am Themenbereich „Zeitgeberbasierte Einheiten“ exemplarisch vorgestellt.

Arbeitsblatt 2, mit folgendem Inhalt:

Lernziele:
 Was sollen Sie nach diesem Unterrichtsblock können?

- Sie kennen typische Baugruppen/Funktionseinheiten eines Mikrocontrollers und können diese auch wiedergeben.
- Sie wissen über deren Bedeutung Bescheid und können diese einer konkreten Anwendung zuordnen.

Wie Sie mittlerweile wissen, verfügt ein Mikrocontroller über mehrere Funktionseinheiten, die beispielsweise zur Anbindung der Peripherie dienen, und je nach Aufgabenstellung zum Einsatz kommen. Im Detail zählen hierzu

- der Analog/Digital-Wandler,
- die Interrupts,
- die zeitgeberbasierten Einheiten aber auch
- der Speicher (bzw. das Konzept der Speichererweiterung)

Sie beschäftigen sich mit dem Thema „Zeitgeberbasierte Einheiten“ (ZbE) und lernen, worum es sich hierbei im Detail handelt und wozu sie bei der Entwicklung typischer Anwendungen dienlich sein können. Um eine grobe Vorstellung zu erhalten, werden zu Beginn einige praktische Problemstellungen angeführt, bei deren Lösung ZbE überaus hilfreich sind.

Hierzu zählt zum Beispiel das Zählen von Ereignissen. Angenommen, Sie möchten die Anzahl von Paketen, die über ein Förderband von A nach B transportiert werden, laufend erfassen, um am Ende des Tages eine konkrete Aussage bezüglich der Stückzahl treffen zu können. In Kombination mit dem entsprechenden Sensor (z.B. Lichtschranke) könnte diese Aufgabe mit einer ZbE realisiert werden. Ebenso können Sie das Messen von Zeiten bewerkstelligen. Damit Sie nicht verschlafen, können Sie damit auch die Funktion eines Weckers realisieren. Oder man erzeugt Impulsfolgen, um beispielsweise einen Piezosummer zum Ertönen zu bringen. Sie sehen, ZbE sind überaus wichtig.

Endziel ist, dass Sie das Prinzip einer ZbE soweit verstehen, um es Ihren KollegInnen erklären zu können. Nachdem es in der Klasse nur wenige ExpertInnen zu diesem Thema gibt, ist Ihr Beitrag überaus wichtig.

 **Aufgabenstellung T4.1:** Recherchieren Sie unter Zuhilfenahme des Internets und bereitgestellter Literaturquellen zum Thema ZbE. Versuchen Sie, die zugrundeliegenden Konzepte und weniger eine konkrete Mikrocontroller-Implementierung zu verstehen. Verwenden Sie keinen Programmcode, fertigen Sie vielmehr Skizzen mit ergänzenden Bemerkungen an.

Arbeitsblatt 2, mit folgendem Inhalt:

Nachdem Sie die erste Phase abgeschlossen haben und bereits ein angehender Experte auf dem Gebiet der ZbE sind, versuchen Sie, nachfolgende Fragen und Aufgabenstellung zuerst in Form einer Einzelarbeit zu bewältigen. Anschließend diskutieren Sie Ihre Ergebnisse mit den anderen ExpertInnen Ihres Themenbereiches.

 **Aufgabenstellung T4.2:** Aufgabenstellung/Kontrollfragen zum Themenbereich ZbE. Dokumentieren Sie Ihre Erkenntnisse.

- Zähler können grundsätzlich aufwärts wie auch abwärts zählen, jedoch wie weit und von wo beginnend?
- Irgendwer muss den Zähler steuern, welche Möglichkeiten sehen Sie hier?
- Siehe Abbildung 6.9: Über welchen Zählerstand verfügt der Zähler am Ende?
- Versuchen Sie, das Grundprinzip der Zeitmessung am Beispiel einer Impulsdauer (Takt=1MHz) darzustellen. Der Beginn/das Ende der Messung soll mit dem Freigabe-Flag (siehe Abbildung 6.9) erfolgen. Zeichnen Sie ein Zeitablaufdiagramm.

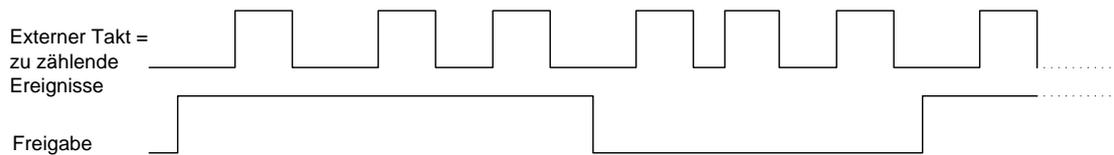


Abbildung 6.9: Zählen externer Ereignisse

- Zeigen Sie anhand eines Beispiels die Funktionsweise einer sogenannten „Capture- und-Compare-Einheit“.
- Im Zusammenhang mit der Pulsweitenmodulation wird oft von einem Frequenzteiler gesprochen. Stellen Sie dessen Funktion grafisch dar.

Arbeitsblatt 3, mit folgendem Inhalt:

Es ist an der Zeit, dass Sie als Experte die anderen von Ihrem Know-how profitieren lassen.

-  **Aufgabenstellung T4.3:** Überlegen Sie, wie Sie Ihr Wissen an andere weitergeben. Fertigen Sie deshalb eine Stichwortliste, nach der Sie vorgehen möchten, erzählen Sie frei und in eigenen Worten. Können Sie Skizzen einsetzen? Überlegen Sie auch, wie Sie feststellen, ob das Thema in der Stammgruppe verstanden wurde (z.B. indem Sie die oben angeführten oder eigene Fragen im Rahmen eines kleinen „Tests“ stellen). Nach dieser 15-minütigen Vorbereitungsphase gehen Sie in die Stammgruppen und zeigen, was Sie können.

Arbeitsblatt 4, mit folgendem Inhalt:

Nachdem Sie Ihr Know-how erfolgreich transferiert haben, gilt es dieses auch längerfristig zu sichern - wäre ja schade, wenn etwas verloren ginge, oder?

-  **Aufgabenstellung T4.4:** Verfassen Sie in der Expertengruppe ein Kompendium, das die wesentlichsten Inhalte, Ideen und Anmerkungen des Themenbereiches ZbE enthält. Ziel ist, dass dieses Werk eine entsprechende Hilfestellung bei den Übungseinheiten darstellt.

Lösungsvorschläge

Bei dieser Unterrichtssequenz sind aufgrund der eingeräumten Freiheiten unterschiedlichste Ergebnisse zu erwarten. Schlussendlich obliegt es der Lehrkraft, diese in die gewünschte Richtung zu lenken. Hierfür dienen beispielsweise die Kontrollfragen der Aufgabenstellung T4.2. Anhand dieser kann die gewünschte Korrektur erfolgen. Daher erscheint es als sinnvoll, auch Lösungen anzubieten. Eine Alternative wäre, wie einleitend erwähnt, ausgearbeitet Inhalte zum jeweiligen Themenbereich bereitzustellen anstatt der Fragen. Somit können die SchülerInnen ebenfalls überprüfen, ob Sie das Thema verstanden haben.

Anmerkungen

Hinsichtlich der Durchführung gilt es anzumerken, dass die Qualität dieser Unterrichtssequenz in einem hohen Maße mit der SchülerInnenanzahl korreliert. Vor allem bei der Durchführung des Gruppenpuzzles, da es doch einiges an Zeit in Anspruch nimmt, bis beispielsweise das Raumsetting eingerichtet ist und alle ihren Platz gefunden haben. Weiters erscheint die stundenweise Durchführung generell als Herausforderung bei dieser Unterrichtssequenz.

6.3 Übungseinheiten

Dieser Abschnitt fasst die Übungseinheiten der Mikrocontrollertechnik zusammen. Diese finden parallel zu den Theorieeinheiten statt und umfassen im Gegensatz dazu zwei Wochenstunden, die in einem Block von 100 Minuten abgehalten werden. Der Aufbau der Übungseinheiten folgt dem gleichen Schema wie die Theorieeinheiten. Um die definierten Ziele zu erreichen, werden insgesamt fünf Übungseinheiten mit unterschiedlicher Dauer festgelegt, wobei die Vorstellung der Übungseinheit „UE5 - Finalisierung Gruppenprojekt 2“ (UE5) zusammengefasst erfolgt.

Nachdem die Praxisnähe ein erklärtes Ziel ist (Anforderung D12), sind entsprechende Werkzeuge und Komponenten einzusetzen. Daher werden diesbezüglich einige Überlegungen zu Beginn angestellt.

Bei der Auswahl der Entwicklungswerkzeuge ist die Anforderung D6 zu berücksichtigen, die den Einsatz frei verfügbarer Entwicklungswerkzeuge festlegt. Demnach bietet sich der *Atmega*-Controller [85] der AVR-Familie an. Für diese Hardware-Plattform gibt es mit dem *AVR-Studio* [73] eine Entwicklungsumgebung (IDE), mit dem *avr-gcc* [74] einen Compiler und mit dem *HAPSIM* [75] einen Simulator, die frei verfügbar sind. Darüber hinaus ist eine Vielzahl unterschiedlicher „Evaluation Kits“ (EK) verfügbar, die sich als motivierenden Ergänzung oder Alternative zum HAPSIM-Simulator anbieten. Durch den Einsatz von EKs wird Anforderung D7 Rechnung getragen.

Stichwort EK: Das Angebot ist vielfältig. Eine mögliche Lösungsvariante stellt der *Arduino Ethernet*-EK dar [87]. Dieser bietet ein akzeptables Preis-Leistungs-Verhältnis und eignet sich nicht zuletzt deshalb für den Einsatz an Schulen. Die Ausstattung des EK im Allgemeinen sowie des AVR-Mikrocontrollers (ATmega328P) im Besonderen erfüllen die Anforderungen. Wie der Abbildung A.2 zu entnehmen ist, kann die Basisfunktion des EK mit weiteren HW-Komponenten (z.B. LEDs, Tastern oder einem LCD) über die ausgeführten Stecker erweitert werden. Somit können Schaltungen von den SchülerInnen dimensioniert und aufgebaut werden, indem nicht zuletzt das Wissen aus vorhergehenden Jahrgängen Anwendung findet. Dieser Lösungsansatz bietet einen akzeptablen Kompromis hinsichtlich der in Kapitel 5.2.3 geführten Diskussion, inwieweit die IT-Ausbildung Elektrotechnik- beziehungsweise Elektronik-Aspekte behandeln soll.

Der Vollständigkeit halber gilt es zu erwähnen, dass die Firma Arduino eine eigene IDE [88] bereitstellt. Diese ist jedoch so konzipiert, dass wesentliche Konzepte als auch Herausforderungen der Embedded Software-Entwicklung verborgen bleiben. Daher ist deren Einsatz

nicht zielführend. Nachdem, wie in Kapitel A.2 dargestellt, die Arduino-Hardware auch ohne deren IDE betrieben werden kann, stellt das kein Problem dar - im Gegenteil, denn das Board verfügt über ein Ethernet-Interface, mit welchem ansprechende Projekte im 5. Jahrgang des Kompetenzbereichs realisiert werden können.

6.3.1 UE1 - Embedded Software Entwicklung

Ziel der ersten Übungssequenz ist die Einführung in die Embedded Software-Entwicklung. Hierzu zählt die Definition wichtiger Begriffe, das Einrichten einer Arbeitsumgebung, das Kennenlernen der Entwicklungswerkzeuge und Hardwarekomponenten sowie ein sogenanntes „Hello World“-Programm.

Übersicht und Eckdaten

Bezeichnung	UE1 - Embedded Software Entwicklung „Hello World“
Themenbereich	Entwicklungswerkzeuge und -komponenten
Adressierte Anforderungen	B1, B3, B4, C1, C4, C12 u. D3
Unterrichtsmethoden	LehrerInnenvortrag (V), Einzelarbeit (E)
Durchführung	Aufgabenstellungen vorstellen und ins Thema einführen (V, ca. 20 Min.), Aufgabenstellung 1.1 (E, ca. 30 Min.), Aufgabenstellung 1.2 (E, ca. 40 Min.), Aufgabenstellung 1.3 (E, ca. 30 Min.) u. Aufgabenstellung 1.4 (E, ca. 80 Min.)
Erforderliche Ressourcen	Internet, PC, EK
Dauer	2 Unterrichtseinheiten (200 Min.)
Quellen	[73, 74, 75, 85, 86, 87, 89]

Aufgabenstellungen

Arbeitsblatt, mit folgendem Inhalt:

<p>Lernziele: Was sollen Sie nach diesem Unterrichtsblock können?</p> <ul style="list-style-type: none"> • Sie kennen zentrale Aspekte sowie Komponenten der hardwarenahen Softwareentwicklung und können diese auch wiedergeben. • Sie können typische Entwicklungswerkzeuge nennen, anwenden und verstehen deren Zusammenwirken. • Sie verfügen über produktspezifische Kenntnisse hinsichtlich des einfachen Schaltungsaufbaus.
--

Inwieweit denken Sie eigentlich bei der objektorientierten Programmierung mit C# über die Beschaffenheit des Rechners, auf welchem Ihr Programm ausgeführt wird, nach?

Wenn überhaupt, dann nur oberflächlich, oder? Was ja grundsätzlich nichts Negatives ist; denn aufgrund der zwei Abstrahierungsschichten, die mit dem Betriebssystem und der .NET-Plattform zwischen Ihrem Programm und der Hardware liegen, ist es grundsätzlich auch „nicht zwingend notwendig“. Das gilt nicht unbedingt für die Embedded Software-Entwicklung.

 **Aufgabenstellung UE1.1:** Recherchieren Sie unter Zuhilfenahme des Internets und der angeführten Leitfragen folgende Begriffe im Kontext der Mikrocontrollertechnik:

- Hardwarenahe Programmierung und Plattformabhängigkeit
- Embedded C und Assembler
- Compiler und Linker
- Evaluation Kit

Leitfragen zur Recherche:

- Ist die Kenntnis des μC (z.B. Aufbau, Funktionsweise, Register, Ein-/Ausgänge (Ports) etc.) bei der hardwarenahen Programmierung von Bedeutung? Siehe auch einleitende Frage.
- Können C-basierte Programme, die für die Plattform A (vgl. μC vom Hersteller A) entwickelt wurden, ohne weiteres auf der Plattform B (vgl. μC vom Hersteller B) ausgeführt werden? Welche Bedeutung kommt in diesem Zusammenhang dem Compiler zu?
- Angenommen, ein Programm ist in Assembler verfasst worden, wäre es dann möglich, dieses auf unterschiedlichen Plattformen problemlos auszuführen?
- Ist es möglich, ein μC -Programm zu schreiben, ohne dass eine Entwicklungsumgebung eingesetzt wird? Macht hierbei die Wahl der Programmiersprache (C oder Assembler) oder des μC einen Unterschied?
- Ist ein EK unbedingt erforderlich, wenn es gilt, eine Wetterstation (mit LCD-Display und Tastenfeld) prototypisch für eine bestimmte μC -Plattform zu entwickeln? Wenn ja, worin könnte der Vorteil liegen?

Nachdem nun einige Begriffe der Embedded Software-Entwicklung klarer erscheinen, ist es an der Zeit, eine Arbeitsumgebung einzurichten. Die Aufgabenstellung UE1.2 beschäftigt sich mit der Installation und Konfiguration jener SW-Komponenten, die für die Übung elementar sind.

 **Aufgabenstellung UE1.2:** Downloaden und installieren Sie folgende SW-Pakete gemäß den Anleitungen der jeweiligen Download-Quelle:

1. AVR-GCC C-Compiler von WinAVR

2. Entwicklungsumgebung AVR Studio (in der Version 4, Versionsabhängigkeit mit HAPSIM)
3. Simulator HAPSIM

Nach erfolgreicher Installation und Konfiguration der Software-Pakete erfolgt die Umsetzung eines „Hello World“-Mikrocontroller-Beispiels. Ziel ist das Blinken einer LED. Legen Sie im ersten Schritt unter Zuhilfenahme der AVR Studio-Hilfe ein Projekt an. Abbildung 6.10 zeigt die erforderlichen Einstellungen. Im zweiten Schritt übernehmen Sie den Quellcode aus dem Listing 6.1 in die Quellcode-Datei „ue1.2_LED.c“ und führen das Programm aus. Zuvor ist jedoch noch der HAPSIM zu starten und zu konfigurieren (siehe Abbildung 6.11).



Abbildung 6.10: AVR Studio 4 - Einstellungen beim Anlegen eines Projektes [73]

Listing 6.1: Quellcode Aufgabenstellung UE1.2

```

1  #define F_CPU 1000000
2
3  #include <avr/io.h>
4  #include <util/delay.h>
5
6  int main(void){
7      DDRB=0x01; // Setze das Datenrichtungsregister des Port B
8      while(1){ // Arbeitsschleife
9          PORTB=0x01; // Setze das 1.Bit/Pin des Port B auf "high"
10         _delay_ms(100); // Verzögerungsfunktion
11
12         PORTB=0x00; // Setze das 1.Bit/Pin des Port B auf "low"
13         _delay_ms(100);
14     }
15     return 0;
16 }

```

Gratulation! Sie haben soeben Ihr erstes Embedded Software-Projekt erfolgreich umgesetzt. Wie Sie erahnen können, hat Ihnen die Entwicklungsumgebung (IDE) einiges an Arbeit abgenommen. Damit Sie eine konkrete Vorstellung der Arbeitsschritte, die vom Quellcode bis zur hex-Datei erforderlich sind, bekommen, sind diese im Detail zu erarbeiten. Sämtliche Werkzeuge (vgl. Programme), die hierfür von der IDE verwendet werden, sind im SW-Pakte WinAVR enthalten. Im Wesentlichen sind das drei Arbeitsschritte.

 **Aufgabenstellung UE1.3:** Finden Sie mit den Quellen Ihrer Wahl diese zentralen Arbeitsschritte heraus und zeigen Sie die Anwendung anhand des Quellcodes im Listing 6.1. Tipp: Sehen sich den Output des AVR Studios beim Kompilieren im „Build-Fenster“ an.

ratsam, eine Installation in einer virtuellen Umgebung vorzubereiten. Nach erfolgter Installation gilt es ein erstes Projekt anzulegen. Hierbei ist es das Ziel, dass die SchülerInnen erste Funktionen des AVR Studios kennenlernen und das Zusammenwirken mit dem Simulator HAPSIM verstehen. Die SchülerInnen machen sich hierbei durch exploratives Arbeiten mit den Werkzeugen vertraut. Sie sollen beispielsweise erfahren, dass der Debugger des AVR Studios gestartet werden muss, damit im HAPSIM die Mikrocontroller-Auswahl getätigt werden kann (siehe Abbildung 6.11, 1). Es wird mit den Abbildungen 6.10 und 6.11 bewusst wenig Information bereitgestellt - diese sollen ausschließlich zur Orientierung dienen. Je nach Erfolg bei dieser Aufgabenstellung ist gegebenenfalls eine Reflexion ratsam.

Durch den Einsatz von Entwicklungsumgebungen wird die Komplexität der „darunter liegenden Arbeitsschritte“ verborgen. Um diese an das Tageslicht zu befördern, dient Aufgabenstellung UE1.3 (Anforderung D1). Es gilt von den SchülerInnen zu erkennen, dass das AVR Studio auf die Ressourcen von WinAVR gemäß Abbildung 6.12 zurückgreift.

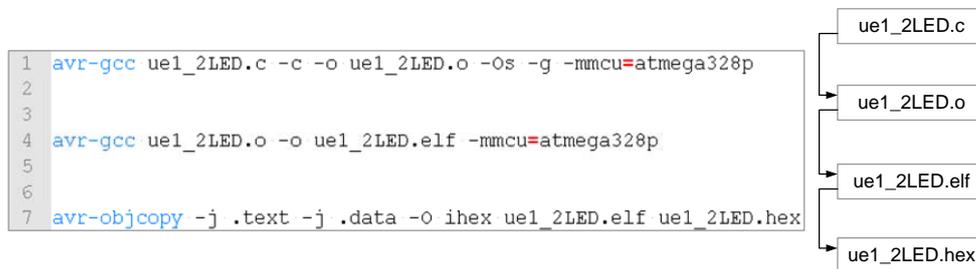


Abbildung 6.12: Build-Prozess mit WinAVR

Bei der Aufgabenstellung UE1.4 gilt es im ersten Schritt bestehendes Wissen hinsichtlich der Dimensionierung einer Leuchtdiode (LED) anzuwenden. Die LED kann hierbei direkt von einem Port des Mikrocontrollers betrieben werden. Wie dem Datenblatt [85] des ATmega328P-Controllers zu entnehmen ist, ist der maximale Strom mit 40mA je I/O-Pin festgelegt. Das reicht für eine herkömmliche LED (z.B. mit $I_F=20\text{mA}$ und $U_F=2\text{V}$). Somit kann mit dem Wissen, dass die Ausgangsspannung des I/O-Pins bei 5V liegt, die Berechnung des Vorwiderstandes R_V mit der Formel 6.1 erfolgen, wobei U_R der Spannungsabfall am Widerstand und I_F der Maximalstrom ist, der durch die LED vorgegeben ist.

$$R_V = \frac{U_R}{I_F} = \frac{U_{Ges} - U_F}{I_F} \quad (6.1)$$

Die Verlustleistung P_V am Vorwiderstand R_V kann mit Formel 6.2 ermittelt werden.

$$P_V = \frac{U_R^2}{R_V} \quad (6.2)$$

Es ist davon auszugehen, dass einige SchülerInnen bei der Dimensionierung als auch Inbetriebnahme des Boards Unterstützung benötigen werden. Daher sind beispielsweise SchülerInnen, die die jeweilige Aufgabenstellung bereits absolviert haben, dazu angehalten, Unterstützung zu leisten.

Anmerkungen

Bei Aufgabenstellung UE1.1 und UE1.3 ist einerseits eine Einzelarbeit zur Recherche sowie andererseits eine Gruppendiskussion zur Ergebniskonkretisierung und -verdichtung vorgesehen. Das bedeutet: beim veranschlagten Zeitbudget gilt es entsprechend Zeit für die Diskussion zu berücksichtigen.

Bei der Aufgabenstellung UE3.2 ist es im ersten Schritt nicht zwingend erforderlich, dass der Code im Detail verstanden wird. Es geht vielmehr darum, dass die SchülerInnen trotz des thematischen Neulandes relativ zeitnahe zu einem Erfolgserlebnis - dem Blinken der LED - gelangen. Dieses Erlebnis kann wesentlich zur Schaffung einer positiven Grundeinstellung gegenüber den Lehrinhalten des Kompetenzbereichs beitragen. Ähnliches gilt für Aufgabenstellung UE3.4; auch hier muss nicht alles gleich zu Beginn verstanden werden.

Da es sich um einen überaus ambitionierten Übungsblock handelt, ist es gegebenenfalls erforderlich, Teile der Aufgabenstellung als Hausübung zwischen den beiden Übungseinheiten in Auftrag gegeben werden.

Es gilt noch anzumerken, dass das AVR Studio bereits in der Version 5 verfügbar ist. Gegenwärtig ist jedoch noch keine Unterstützung vom HAPSIM gegeben; was bei einem möglichen Einsatz des Arduino-Boards jedoch kein allzu großes Problem darstellt.

6.3.2 UE2 - How to use a port

Nachdem bei der ersten Übungssequenz (siehe Kapitel 6.3.1) ein überaus ambitionierter Einstieg erfolgt, ist die Darlegung der Hintergründe primäres Ziel dieses Abschnitts. Die SchülerInnen sollen die Funktionsweise eines Mikrocontroller-Ports verstehen und diesen in einem Programm verwenden können.

Übersicht und Eckdaten

Bezeichnung	UE2 - How to use a port
Themenbereiche	Digital I/O: Portzugriff durch Bitmanipulation
Adressierte Anforderungen	B1, B5, C1, C4, C12, D3 D6, D7
Unterrichtsmethoden	LehrerInnenvortrag (V), Einzelarbeit (E)
Durchführung	Aufgabenstellungen vorstellen und ins Thema einführen (V, ca. 15 Min.), Aufgabenstellung 2.1 (E, ca. 20 Min.), Aufgabenstellung 2.2 (E, ca. 20 Min.), Aufgabenstellung 2.3 (E, ca. 60 Min.) u. Aufgabenstellung 2.4 (E, ca. 15 Min., Aufgabenstellung 2.5 (E, ca. 30 Min.) u. Aufgabenstellung 2.6 (E, ca. 60 Min.)
Erforderliche Ressourcen	PC, EK
Dauer	3 Unterrichtseinheiten (300 Min.)
Quellen	[85, 86, 90]

Aufgabenstellungen

Arbeitsblatt, mit folgendem Inhalt:

Lernziele:

Was sollen Sie nach diesem Unterrichtsblock können?

- Sie kennen die Funktionsweise eines Ports und können diesen in einem Programm verwenden.
- Sie verstehen die Bitmanipulation durch den Einsatz von Bitmasken und Schiebeoperatoren.
- Sie können Datentypen auswählen und grundlegende Kontrollstrukturen verfassen.

Datenblatt, die Rettung - ein Bild sagt bekanntlich mehr als 1000 Worte!

Wie Sie bereits festgestellt haben, ist das Datenblatt [85] des ATmega328P-Controllers äußerst umfangreich. Um den Aufbau und die Funktionsweise des Controllers zu verstehen, könnte man das Datenblatt von der ersten bis zur letzten Seite lesen. Viel Spass! Zielführender ist, wenn man sich einen Überblick verschafft. Hierfür dienen vor allem zusammenfassende Darstellungsformen wie beispielsweise Grafiken oder Tabellen. Somit ist man zumindest für erste Aufgaben gerüstet. Der Rest kann *on demand* gelesen werden. Abbildung 6.13 zeigt eine interessante Grafik des Datenblatts. Hierbei handelt es sich um die Pin-Belegung des ATmega328P-Controllers. Uns interessieren vor allem die Ein- und Ausgänge - auch Ports genannt - des Controllers. Wie der Abbildung zu entnehmen ist, verfügt jeder Pin über eine „Primärfunktion“ (z.B. VCC, PB0, PD3 etc.) und nahezu jeder über eine „Sekundärfunktion“ (z.B. ADC0, MISO etc.).

Abbildung 6.13 zeigt die Pin-Belegung des ATmega328P (Bauform 28PDIP). Die Pins sind in zwei Spalten angeordnet, mit den Pinnummern 1 bis 14 auf der linken Seite und 28 bis 15 auf der rechten Seite. Jeder Pin ist mit einer Primärfunktion und einer Sekundärfunktion beschriftet.

(PCINT14/RESET) PC6	1	28	PC5 (ADC5/SCL/PCINT13)
(PCINT16/RXD) PD0	2	27	PC4 (ADC4/SDA/PCINT12)
(PCINT17/TXD) PD1	3	26	PC3 (ADC3/PCINT11)
(PCINT18/INT0) PD2	4	25	PC2 (ADC2/PCINT10)
(PCINT19/OC2B/INT1) PD3	5	24	PC1 (ADC1/PCINT9)
(PCINT20/XCK/T0) PD4	6	23	PC0 (ADC0/PCINT8)
VCC	7	22	GND
GND	8	21	AREF
(PCINT6/XTAL1/TOSC1) PB6	9	20	AVCC
(PCINT7/XTAL2/TOSC2) PB7	10	19	PB5 (SCK/PCINT5)
(PCINT21/OC0B/T1) PD5	11	18	PB4 (MISO/PCINT4)
(PCINT22/OC0A/AIN0) PD6	12	17	PB3 (MOSI/OC2A/PCINT3)
(PCINT23/AIN1) PD7	13	16	PB2 (SS/OC1B/PCINT2)
(PCINT0/CLKO/ICP1) PB0	14	15	PB1 (OC1A/PCINT1)

Abbildung 6.13: Pin-Belegung des ATmega328P (Bauform 28PDIP) [85]

Abbildung 6.14 zeigt ein Blockdiagramm des ATmega328P-Controllers, das in engem Bezug zu den Gehäuse-Pins des Controllers (siehe Abbildung 6.13) steht. Mehrere Pins werden zu einem sogenannten Port zusammengefasst. Daher bezeichnet der Begriff Port allgemein eine Gruppe physischer Ein- und/oder Ausgänge eines Mikrocontrollers.

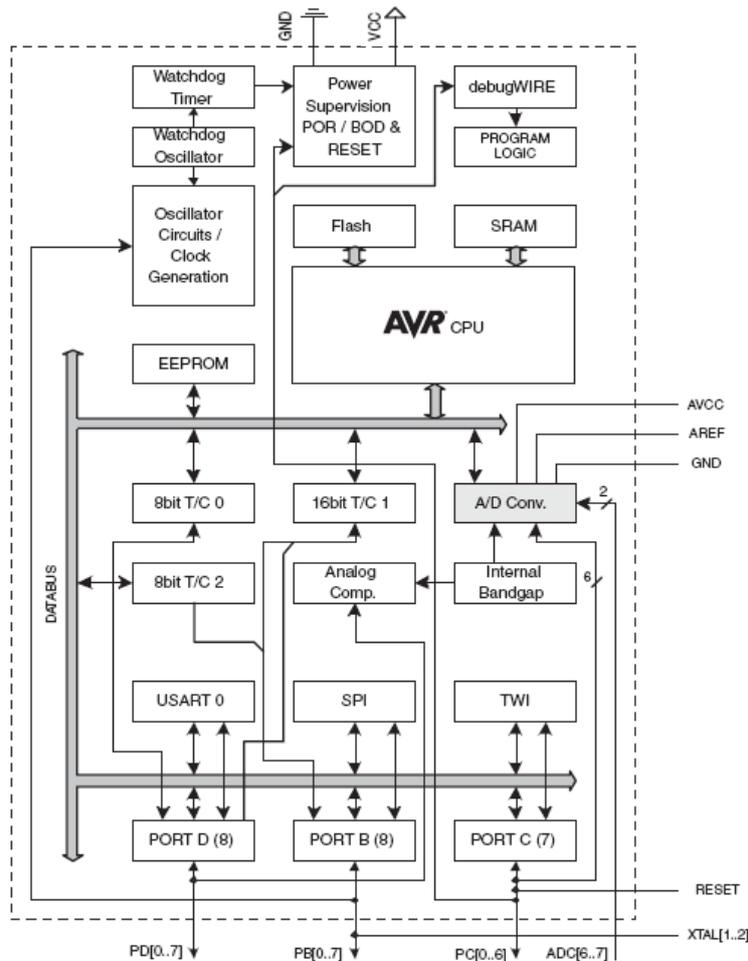


Abbildung 6.14: Blockdiagramm des ATmega328P [85]

Aufgabenstellung UE2.1: Finden Sie heraus, über wieviele Ports der Controller verfügt und ordnen Sie die Pins der Abbildung 6.13 dem jeweiligen Port zu. Verwenden Sie hierfür unterschiedliche Farben. Versuchen Sie weiters, einen Zusammenhang zwischen den Abbildungen 6.13 und 6.14 herzustellen.

Da jetzt Klarheit bezüglich der Anzahl und der allgemeinen Funktionsweise der Ports herrscht, stellt sich die Frage, wie man einen Port in einem Programm verwenden kann? Die Antwort: Es kommt darauf an.

Generell verfügt jeder physische Port des ATmega-Controllers über drei Register (DDRx, PORTx u. PINx (x=[B,C,D])). Hierbei handelt es sich um Speicherbereiche (8 Bit=1Byte) im SRAM (siehe Abbildung 6.15). Um eines dieser Register anzusprechen, erfolgt im Programmcode die entsprechende Zuweisung (siehe Listing 6.1); doch dazu später mehr.

Wozu gibt es drei Register für einen Port? Mit dem Register DDRx (*Data Direction Register*) wird festgelegt, ob der jeweilige Pin des Ports ein Ein- oder Ausgang ist. Es gilt: 1=Ausgang, 0=Eingang. Das bedeutet, jedes Bit des DDRx-Bytes im SRAM hat direkten Bezug auf einen Pin, wie der Abbildung 6.15 auch zu entnehmen ist (die ersten 4 Bits (0 bis 3) sind auf logisch „1“ gesetzt, daher handelt es sich bei den Pins 0 bis 3 um Ausgänge, die restlichen sind als Eingänge konfiguriert). Wenn mit dem DDRx die „Richtung“ eines Pins festgelegt ist, kommen die zwei verbleibenden Register ins Spiel. Generell gilt:

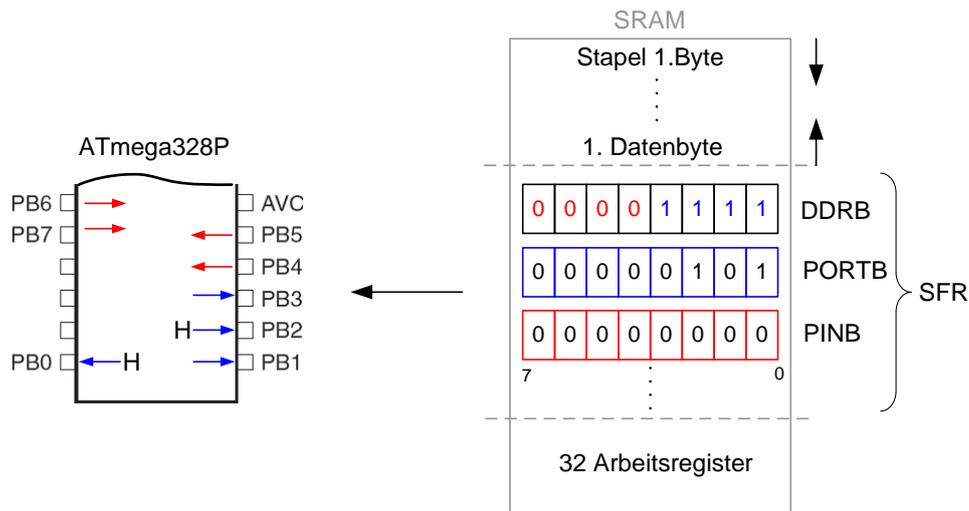


Abbildung 6.15: Die Port-Register und ihre Zusammenhänge

- PORTx dient zum Setzen von Ausgangssignalen,
- PINx dient zum Abfragen von Eingangssignalen.

Gilt es jetzt zum Beispiel mehrer LEDs an einem Port zu schalten, sind die entsprechenden Pins des Ports als Ausgang im DDRx zu setzen. Ein- und Ausgeschaltet werden die LEDs mit dem jeweiligen Pin des PORTx-Registers. Dieser Logik folgend, würden somit die LEDs der Pins 0 und 2 des Controllers (siehe Abbildung 6.15) leuchten, da im Register PORTB die entsprechenden Bits auf „1“ gesetzt sind.

Aufgabenstellung UE2.2: Angenommen, Sie möchten am Port B des Controllers ein Laufflicht mit 4 LEDs betreiben. Überlegen Sie die Register-Zustände für sieben Zeitintervalle (Laufflichtabfolge: LED 1 >LED 2 >LED 3 >LED 4 >LED 3 >LED 2 >LED 1 >LED 2 usw.).

Jetzt stellt sich die Frage, wie man einem Register den gewünschten Zustand zuweisen kann? Hierfür gibt es mehrere Möglichkeiten: entweder die Dezimal-, Binär- oder Hexadezimalschreibweise. Angenommen, dem Register PORTB ist der Wert $45_{(10)}$ zuzuweisen, dann kann das gemäß den in Tabelle 6.1 vorgestellten Möglichkeiten erfolgen. Üblicherweise wird die hexadezimale Darstellungsform verwendet.

Tabelle 6.1: Darstellung und Registerzuweisung eines Zahlenwertes

	Darstellung	Zuweisung
Dezimal	45	PORTB=45;
Binär	0b00101101	PORTB=0b00101101
Hexadezimal	0x2D	PORTB=0x2D

Aufgabenstellung UE2.3: Da Sie mittlerweile über das erforderliche Know-how verfügen, um das 4-LED-Laufflicht zu realisieren, ergänzen Sie das Programm aus Listing 6.1 dahingehend, dass sich die gewünschte Laufflicht-Funktionalität aus Aufgabenstellung UE2.2 ergibt. Testen Sie das Programm mit dem Simulator.

Sie haben mit dem 4-LED-Laufflicht ihr erstes Programm bewerkstelligt, das die Pins 0 bis 3 des Port B beansprucht. Leider haben sich die Rahmenbedingungen dahingehend geändert, indem die Pins 4 bis 7 des Ports anderweitig benötigt werden.

Aufgabenstellung UE2.4: Überlegen Sie, ob diese Information irgendeine Bedeutung für Ihr Programm haben könnte. Wenn ja, auf welcher Ebene? Was wären mögliche Gegenmaßnahmen?

Wie die Aufgabenstellung UE2.4 zeigt, ist beim Port-Zugriff Vorsicht geboten. Um diesen entsprechend zu gestalten, ist die sogenannte Bitmanipulation in Kombination mit Bitmasken überaus hilfreich. Wie die Bezeichnungen bereits vermuten lassen, können mit Hilfe dieser Techniken einzelne Bits manipuliert werden. Abbildung 6.16 zeigt die an den Port B angeschlossenen LEDs und den gegenwärtigen Zustand des „Ausgangsregisters“ PORTB zum Zeitpunkt t_0 . Ziel ist eine Zustandsänderung im Register PORTB, sodass nicht mehr die LEDs 1 und 3 sondern 2 und 4 leuchten, ohne die Pins 4 bis 7 des Registers zu verändern. Überlegen Sie, mit welcher logischen Verknüpfung eine bestimmte Bitmaske mit dem Zustand des Registers PORTB zum Zeitpunkt t_0 verknüpft werden muss, damit das Register den Zustand zum Zeitpunkt t_1 einnimmt. Wenden Sie hierbei die angeführten Wahrheitstabellen an.

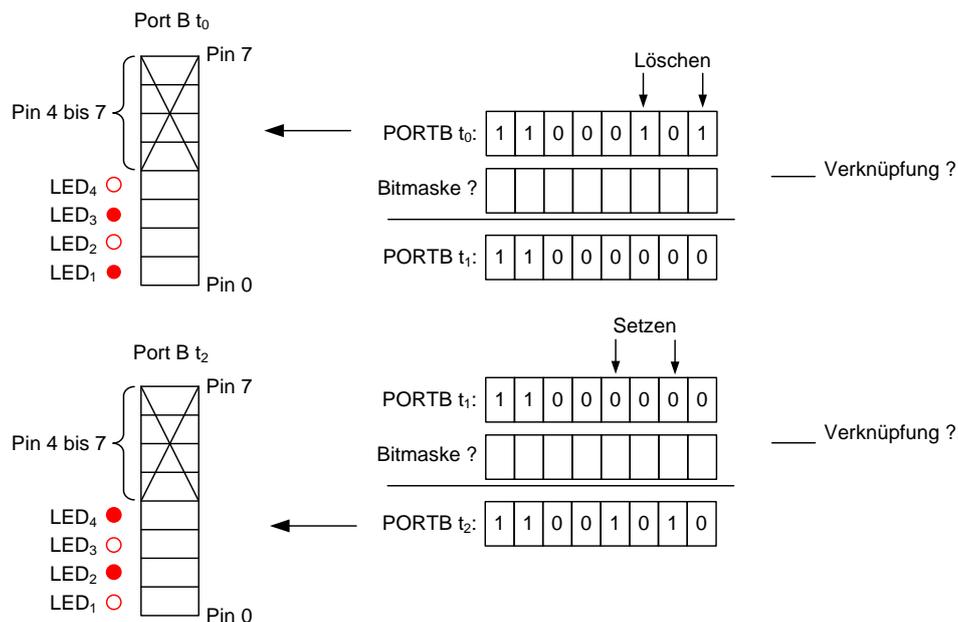


Abbildung 6.16: Bitmanipulation durch den Einsatz von Bitmasken

Tabelle 6.2: Konjunktion - Bool'scher Operator \wedge

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

Tabelle 6.3: Disjunktion - Bool'scher Operator \vee

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

Der Compiler kann mit den Bool'schen Operatoren \wedge und \vee nicht arbeiten, daher sind auf Programmebene die sogenannten Bitoperatoren „&“ und „|“ erforderlich. Das Listing 6.2 zeigt die Anwendung auf Programmebene.

Listing 6.2: Bitmaskierung und Portzuweisung mit C

```

1 //Programmauszug
2
3 PORTB=0x03; // Pin 0 und 1 des Port B sind "high"
4 PORTB=PORTB & 0xfe; // Pin 0 wird mit dieser Maske gelöscht
5 // PORTB hat jetzt folgenden Inhalt: 0000 0010 = 0x02 = 2
6
7 PORTB=PORTB | 0x01; // Pin 0 wird mit dieser Maske gesetzt
8 // PORTB hat jetzt folgenden Inhalt: 0000 0011 = 0x03 = 3

```

 **Aufgabenstellung UE2.5:** Setzen Sie die soeben gelernte Bitmanipulation anhand der Zuweisung von Bitmasken beim Programm aus Aufgabenstellung UE2.4 ein. Ziel: Ausschließlich jene Pins dürfen geändert werden, an welchen die LEDs angeschlossen sind.

Wie Sie bei der Aufgabenstellung UE2.5 sicherlich festgestellt haben, ist die Erzeugung der Masken eine häufige Fehlerquelle, oder? Es gibt in diesem Zusammenhang weitere Operatoren, die das Erstellen von Masken wesentlich vereinfachen. Das sind die sogenannten Schiebeoperatoren (nach rechts schieben mit dem Operator „>>“ und nach links mit „<<<“). Allgemein gilt:

$$1 \ll \text{Bitpositionen}$$

Die praktische Anwendung dieser Schiebeoperatoren ist eigentlich ganz einfach, wie Abbildung 6.17 zeigt. Man erzeugt die Masken mit den Pins, die man auf „high“ setzen möchte (z.B. für Pin 1 „(1 <<1)“ und Pin 3 „(1 <<3)“) und verknüpft diese in weiterer Folge mit dem Port-Ausgangsregister (z.B. PORTB). Hierbei kommt der Operator „|“ zum Einsatz, da das Setzen eines Pins gefordert ist. Wie sich die Anwendung auf Programmebene gestaltet, ist nebenan zu sehen; wobei die Variante mit den Pin-Konstanten des jeweiligen Ports (siehe ARR Studio, „avr/io.h“) zu bevorzugen ist.

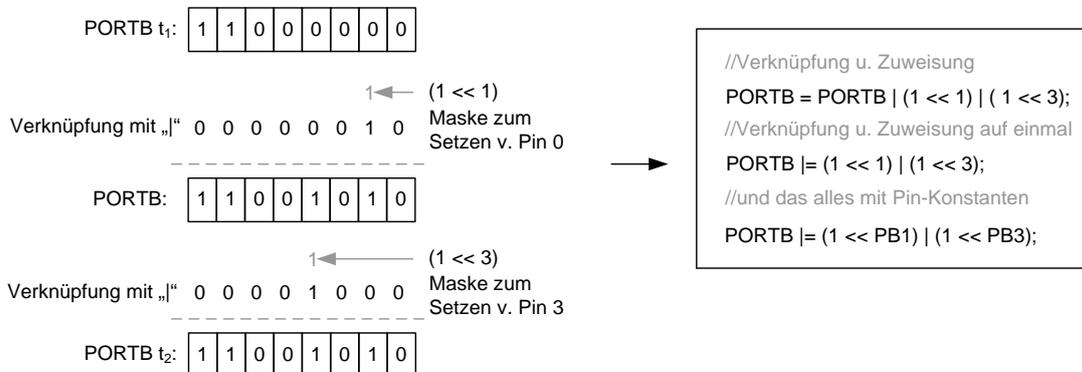


Abbildung 6.17: Bitmanipulation und Bitmaskenerstellung mit Schiebeoperatoren

Diese Methode kann auch beim Löschen angewendet werden. Hierfür erzeugt man zuerst ebenfalls die Maske(n), für die Pins, die man löschen möchte. Da das Löschen mit logisch „0“

erfolgen muss, ist die Maske noch zu invertieren. Hierfür gibt es den Operator „~“ (Tilde). Und anstatt des „|“-Operators ist natürlich dieser „&“ zu verwenden.

 **Aufgabenstellung UE2.6:** Bearbeiten Sie im ersten Schritt das Übungsblatt A.7 und adaptieren Sie im zweiten Schritt das Programm aus Aufgabenstellung UE2.5 dahingehend, dass die Maskenerstellung mittels Schiebeoperatoren zum Einsatz kommt.

Lösungsvorschläge

Die Aufgabenstellung UE2.1 zielt auf das Erkennen der Ports und deren jeweilige(n) Funktion(en) ab. Jeder Port verfügt grundsätzlich über E/A-Funktion und kann über die entsprechenden Port-Register angesprochen werden. Neben dieser Primärfunktion weisen Ports eine Sekundärfunktion auf. Beispielsweise kann der A/D-Konverter nur über Port C, USART nur über Port B oder die SPI-Schnittstelle ausschließlich über den Port B angesprochen werden.

Anhand der Aufgabenstellung UE2.2 soll die Funktionsweise eines Ports allgemein verdeutlicht werden, indem der Zusammenhang der drei Register DDRx, PORTx und PINx veranschaulicht wird. Den SchülerInnen gilt es Abbildung 6.18 bereitzustellen, die die relevanten Portvariablen (SFR, *Special Function Register*) zusammengefasst darstellt.

SRAM	SFR	Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0x38	0x18	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
0x37	0x17	DDRB	DDB7	DDB6	DDB5	DDB4	DDB3	DDB2	DDB1	DDB0
0x36	0x16	PINB	PINB7	PINB6	PINB5	PINB4	PINB3	PINB2	PINB1	PINB0
0x32	0x12	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
0x31	0x11	DDRD	DDD7	DDD6	DDD5	DDD4	DDD3	DDD2	DDD1	DDD0
0x30	0x10	PIND	PIND7	PIND6	PIND5	PIND4	PIND3	PIND2	PIND1	PIND0
0x2A	0x0A	UCSRB	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCSZ2	RXB8	TXB8
0x4C	0x2C	TCNT1	16bit Timerregister ohne Bitbezeichner							
0x4C	0x2C	TCNT1L	Low-Byte des 16bit Timerregisters							
0x4D	0x2D	TCNT1H	High-Byte des 16bit Timerregisters							

Abbildung 6.18: Beispiele für vordefinierte Portvariablen und Bitbezeichner [90]

Bevor die SchülerInnen mit dem Programmieren beginnen (Aufgabenstellung UE2.3), gilt es das Grundgerüst eines Mikrocontrollerprogramms kurz zu erläutern (z.B. relevante Präprozessoranweisungen, Arbeitsschleife und vor allem plattformspezifische Datentypen). Je nachdem, wie ausgeprägt das Vorwissen bezüglich der C-Programmierung ist, gilt es zum Beispiel grundlegende Kontrollstrukturen vorzustellen. In diesem Fall würden sich übungsbegleitend ergänzende Folien anbieten, wie etwa in Abbildung A.8 (Anhang) beispielhaft dargestellt. Im Zusammenhang mit Datentypen sollte auf jeden Fall der Unterschied zwischen „signed“ und „unsigned“ sowie mögliche Problematiken bei der Verwendung (z.B. Überlauf einer Variable aufgrund eines zu kleinen Wertebereichs) bekannt sein. Weiters sind die SchülerInnen bezüglich der Konsequenzen bei der Datentypauswahl zu sensibilisieren, da der Programmspeicher um ein Vielfaches geringer ist als bei einem herkömmlichen PC-Systemen.

Aufgabenstellung UE2.4 dient als möglicher Indikator, inwieweit bereits erste Konzepte verstanden wurden. Durch die direkte Wertzuweisung bei einer Portvariabel werden alle

Bits/Pins verändert, was gegebenenfalls nicht gewünscht ist. Das gilt es im günstigsten Fall von den SchülerInnen zu erkennen. Aufbauend auf diesen Erkenntnissen sollen mit der Aufgabenstellung UE2.5 Lösungsmöglichkeiten erarbeitet werden. Hierbei werden mit der Bitmanipulation durch den Einsatz von Masken und Schiebeoperatoren grundlegende Konzepte vermittelt. Ergänzend zur UND-, ODER- u. NICHT-Verknüpfung ist die XOR-Verknüpfung zu erwähnen, mit welcher das sogenannte „Toggeln“ eines Bits bewerkstelligt werden kann.

Schlussendlich sollten alle SchülerInnen nach Absolvierung der sechs Aufgabenstellungen und den damit verbundenen Zwischenlösungen in etwa bei dem in Listing 6.3 angeführten Lösungsvorschlag anlangen. Grundsätzlich könnten die Portvariablen DDRB und PORTB auch mit einer direkten Zuweisung (z.B. DDRB=0x0f o. PORTB=1) initialisiert werden. Die dargestellte Variante dient somit ausschließlich für Übungszwecke.

Listing 6.3: C-basiertes Laufflicht mit 4 LEDs am Port B (Pin 0 bis 3)

```

1 #include <avr/io.h>
2 #include <util/delay.h>
3
4 #define IS_LED4 32
5 #define IS_LED1 1
6 #define STEPS_TO_MOVE 1
7
8 unsigned char toggle=1;
9
10 int main(void){
11
12     DDRB |= (1<<PD0)|(1<<PD1)|(1<<PD2)|(1<<PD4); // Pin 0–3 Ausgänge
13     PORTB |= (1<<PD0); //Pin 0 aktiv setzten...
14
15     while(1){
16         if(toggle){
17             PORTB <<=STEPS_TO_MOVE; //...und mit Schiebeoperator rauf...
18             if(PORTB==IS_LED4) toggle=0;
19         }
20         else{
21             PORTB >>=STEPS_TO_MOVE; //...und wieder runter schieben.
22             if(PORTB==IS_LED1) toggle=1;
23         }
24     }
25     return 0;
26 }

```

Anmerkungen

Diese Übungsequenz ist für drei Unterrichtseinheiten (zu a 100 Minuten) veranschlagt. Insofern noch Zeit zur Verfügung steht, können weitere Programmierthemen (z.B. globale und lokale Variablen in Kombination mit Funktionen sowie Modifikatoren *static*, *const* und *volatile* behandelt werden. Beispielsweise könnte die Laufflichtfunktion des Listings 6.3 in eine Funktion mit einer globalen oder lokalen Variable „toggle“ ausgelagert werden. Bei der lokalen könnte wiederum *static* zum Einsatz kommen.

Für die vorgestellten Aufgabenstellung werden 220 Minuten budgetiert. Die verbleibenden 80 Minuten dienen einerseits als Puffer oder andererseits für ergänzende Inhalte (siehe vorhergehenden Absatz). Für SchülerInnen, die erheblich schneller sind als die anderen, könnte der Aufbau der Schaltung ein mögliches Additum darstellen (Anforderung B2).

6.3.3 UE3 - Anbindung peripherer Komponenten

Diese Übungssequenz beschäftigt sich mit der Anbindung externer Komponenten. Im Fokus stehen hierbei Tasten als Ein- und ein LCD als Ausgabekomponente.

Übersicht und Eckdaten

Bezeichnung	UE3 - Anbindung peripherer Komponenten
Themenbereiche	Digitale I/O: Tasten und LCD
Adressierte Anforderungen	B1, C1, C4, C12, D3, D6, D7
Unterrichtsmethoden	LehrerInnenvortrag (V), Einzelarbeit (E)
Durchführung	Aufgabenstellungen vorstellen und ins Thema einführen (V, ca. 20 Min.), Aufgabenstellung 3.1 (E, ca. 20 Min.), Aufgabenstellung 3.2 (E, ca. 60 Min.) u. Aufgabenstellung 3.3 (E, ca. 100 Min.)
Erforderliche Ressourcen	PC, EK
Dauer	2 Unterrichtseinheiten (200 Min.)
Quellen	[75, 90, 91]

Aufgabenstellungen

Arbeitsblatt, mit folgendem Inhalt:

<p>Lernziele: Was sollen Sie nach diesem Unterrichtsblock können?</p> <ul style="list-style-type: none"> • Sie können externe Ereignisse im Programm erfassen und als Konsequenz darauf eine bestimmte Aktionen ausführen. • Sie wissen über die Bedeutung von Treibern allgemein bescheid. • Sie können ein ausgewähltes LCD in ein Programm einbinden und verwenden.
--

Bei Menschen kann aufgrund externer Ereignisse eine Zustandsänderung bewirkt werden - also zumindest manchmal! Wie sieht das aber bei einem Mikrocontroller aus?

Eigentlich ganz einfach, man benötigt beispielsweise einen Taster, der an den Controller angeschlossen und dessen Zustand vom Programm laufend abgefragt wird. Wie bereits bekannt ist, können die Pins eines Ports entweder als Ein- oder als Ausgang verwendet werden.

Hinsichtlich eines Tasters ist es wohl naheliegend, dass es sich um einen Eingang handeln muss. Demzufolge benötigen wir eine „0“ an gewünschter Stelle bei der Portvariable DDRx. Aber welcher Port kommt hierfür infrage?

Aufgabenstellung UE3.1: Finden Sie anhand der Abbildung 6.14 mögliche Ports heraus. Begründen Sie Ihre Entscheidung.

Wie die Abbildung 6.19 zeigt, ist am Port B, Pin 7 (PB7), ein Taster angeschlossen. Wird der Taster betätigt (siehe strichlierte Linie), erkennt das der Controller am PB7 und setzt den entsprechenden Pin des Registers PINB auf „1“. Ist der Taster nicht betätigt, dann steht eine „0“ im PINB, Bit 7.

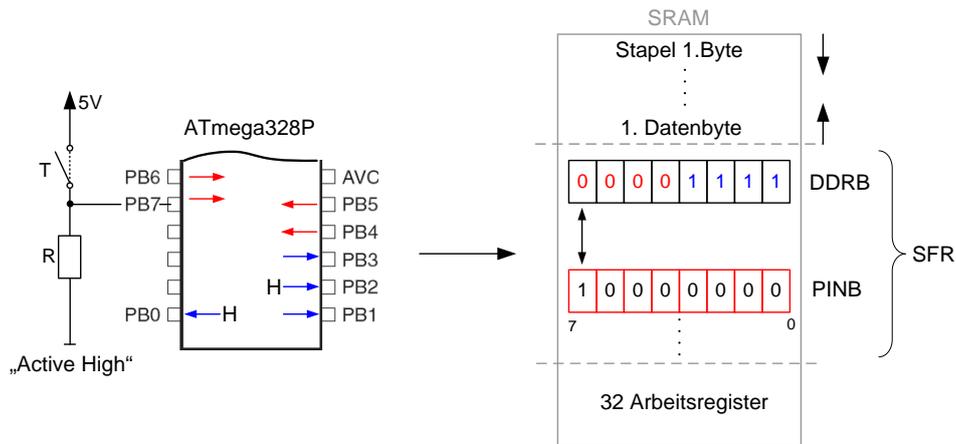


Abbildung 6.19: Portregister und ihre Zusammenhänge bei der Verwendung des Ports als Eingang

Jetzt bedarf es nur noch weniger Codezeilen, um den Taster abzufragen, der an PB7 angeschlossen ist. Wie dem Listing 6.4 zu entnehmen ist, kann die Abfrage mit einer *if*-Bedingung bewerkstelligt werden. Nun stellt sich die Frage, wie es zur Erfüllung dieser Bedingung kommt? Es gilt: Eine *if*-Bedingung ist dann wahr, wenn der Ausdruck in der Klammer größer „0“ ist.

Listing 6.4: Abfragen eines Taster am Port B

```

1 //Programmauszug
2 while(1){
3     if(PINB & (1<<PB7)){
4         cnt++;
5     }
6     _delay_ms(100);
7 }

```

Aufgabenstellung UE3.2: (i) Zeigen Sie, warum die Bedingung erfüllt ist, wenn die Taste an PB7 betätigt wird. Tipp: Eine Betrachtung auf Bit-Ebene ist zielführend. (ii) Nachdem Sie die Verarbeitung eines Tasten-Ereignisses verstanden haben, schreiben Sie ein Programm, welches das „4-LED-Lauflicht“ aus UE2.3 um die Funktion des Ein-/Ausschaltens mit einem Taster (Togglefunktion) erweitert. Verwenden Sie die LEDs und einen Taster des HAPSIM.

Um beispielsweise ein Bedienelement für eine Produktionsmaschine zu realisieren, reichen LEDs bei Weitem nicht aus. Es braucht eine Ausgabeeinheit, die eine zweckmäßige Informationsdarstellung ermöglicht - ein LCD (*Liquid Crystal Display*). Wie Sie vielleicht schon wissen, verfügt der HAPSIM über eines. Wie Sie in Kürze feststellen werden, ist die Verwendung gar nicht so einfach. Das wird spätestens bei der Betrachtung des LCD-Datenblatts [91] deutlich. Um ein LCD verwenden zu können, ist also jede Menge Programmieraufwand erforderlich - oder es gibt bereits einen Treiber für die jeweilige Hardware-Plattform. Im Falle des LCD vom HAPSIM, dem HD44780, trifft das zu. Der Treiber ist auf der Projektseite [75] zu finden.

 **Aufgabenstellung UE3.3:** Downloaden Sie den LCD-Treiber von der HAPSIM-Projekt-Website und binden Sie diesen in Ihr Programm ein. Die Datei „lcd.c“ ist dem AVR Studio-Ordner „Source Files“ hinzuzufügen. Die Dateien „lcd.h“ und „bool.h“ (sogenannte Header-Dateien) sind wiederum im Ordner „Header Files“ abzulegen. Der Treiber - also der Quellcode - wird mit `#include „lcd.h“` in das Programm eingebunden. Um den Treiber zu verstehen, sind die Funktionsbeschreibungen der Datei „lcd.h“ überaus hilfreich. Anhand dieser geht die Verwendung des Display-Treibers hervor.

Nachdem Sie den Treiber erfolgreich eingebunden und dessen Verwendung verstanden haben, gilt es eine erste Display-Ausgabe zu realisieren. Schreiben Sie ein Programm, das den Schriftzug „Hello World“ von links nach rechts laufen lässt, bis dieser am rechten Ende des Displays verschwindet (siehe Abbildung 6.20).



Abbildung 6.20: „Hello World“-Ausgabe auf dem LCD [75]

Lösungsvorschläge

Aufgabenstellung UE3.1 soll wiederum zum Allgemeinverständnis des Mikrocontrollers beitragen, indem dessen Architektur zu analysieren ist. Die SchülerInnen sollten aufgrund der bidirektionalen Pfeile im Blockdiagramm 6.14 erkennen, dass Port B, C und D grundsätzlich als Eingänge infrage kommen. Die Wahl sollte jedoch immer im Gesamtkontext der Aufgabenstellung erfolgen. Beispielsweise kann der A/C-Konverter ausschließlich über den Port C verwendet werden, wodurch dieser nicht die erste Wahl sein kann, wenn die A/D-Konverterfunktionalität benötigt wird. Diese Überlegungen gilt es zu diskutieren.

Bei der Aufgabenstellung UE3.2 sind folgende Zusammenhänge von Bedeutung: Wird der Taster gedrückt, so ist das Bit 7 im Portregister PINB gesetzt (siehe Abbildung 6.21). Durch die Und-Verknüpfung mit der entsprechenden Maske wird ausschließlich das 7. Bit selektiert. Somit ergibt das den Wert 128, der größer „0“ ist; wodurch die Bedingung erfüllt ist.

Bei der Aufgabenstellung UE3.3 gilt es einen Treiber zu verwenden. Die SchülerInnen sind gefordert, unbekannte Programmteile im ersten Schritt zu verstehen, um diese im zweiten

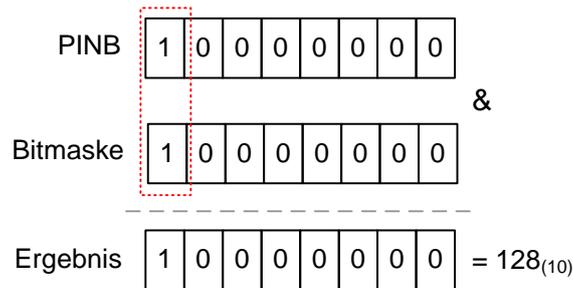


Abbildung 6.21: Tasten-Ereignis mittels Bitmanipulation verarbeiten

Schritt anwenden zu können. Augenmerk ist auch auf die Verwendung der Treiberkonstanten (siehe Listing 6.5, z.B. Zeile 7 oder 11) zu legen. Anhand diese Beispiels soll deutlich werden, dass eine Display-Ausgabe mehrer Schritte erfordert (1. Löschen, 2. Positionieren u. 3. Ausgeben).

Listing 6.5: Ausgabe von Hello World auf dem LCD

```

1 #include <string.h>
2 #include <avr/delay.h>
3 #include "lcd.h"
4
5 int main(void){
6
7     lcd_init(LCD_DISP_ON); //Display initialisieren
8     lcd_clrscr(); //Display löschen – sicher ist sicher!
9
10    while(1){
11        for(char x=0; x<=LCD_DISP_LENGTH; x++){ //Zählschleife, die die x-Position inkrementiert
12            lcd_clrscr();
13            lcd_gotoxy(x, 0); //Ausgabekoordinaten setzen
14            lcd_puts("Hello World"); //Ausgabe
15            _delay_ms(100);
16        }
17    }
18    return 0;
19 }

```

Anmerkungen

Als Additum würde sich eine Erweiterung der Aufgabenstellung UE3.3 anbieten; beispielsweise könnte man den Schriftzug von links Zeichen für Zeichen wieder anzeigen, oder mit der Y-Koordinate des Displays arbeiten. Als weitere Zusatzaufgabe würde der Aufbau der Schaltung gelten - sowohl bei Aufgabenstellung UE3.2 als auch bei UE3.3.

6.3.4 UE4 - E/A-Baugruppen

Diese Übung behandelt mit dem Analog/Digital-Wandler (ADW) sowie der Unterbrechungssteuerung (engl. *Interrupt*) zwei grundlegende Baugruppen eines Mikrocontrollers.

Übersicht und Eckdaten

Bezeichnung	UE4 - E/A-Baugruppen
Themenbereiche	Messen analoger Größen, Konzept von Interrupts
Adressierte Anforderungen	B1, B6, C1, C4, C12, D3, D7
Unterrichtsmethoden	LehrerInnenvortrag (V), Einzelarbeit (E)
Durchführung	Aufgabenstellungen vorstellen und ins Thema einführen (V, ca. 10 Min.), Aufgabenstellung 4.1 (E, ca. 10 Min.), Aufgabenstellung 4.2 (E, ca. 100 Min.), Aufgabenstellung 4.3 (E, ca. 20 Min.) u. Aufgabenstellung 4.3 (E, ca. 60 Min.)
Erforderliche Ressourcen	Internet, PC, EK
Dauer	2 Unterrichtseinheiten (200 Min.)
Quellen	[85, 90, 92]

Aufgabenstellungen

Arbeitsblatt, mit folgendem Inhalt:

Lernziele:

Was sollen Sie nach diesem Unterrichtsblock können?

- Sie können einen A/D-Wandler konfigurieren und im Programm verwenden.
- Sie wissen über die Bedeutung von Interrupts Bescheid und können diese anwenden.

Alles zu seiner Zeit! Mikrocontroller nehmen es überaus genau, wenn man ihnen eine Aufgabe überträgt.

Zu Beginn der Theorieeinheiten haben wir ein konkretes Anwendungsbeispiel der Mikrocontrollertechnik behandelt (siehe Abbildung 6.6). Bei diesem gilt es neben der Temperatur auch die Luftfeuchtigkeit zu messen. Wie kann das bewerkstelligt werden? Richtig, mit einem Analog/Digital-Wandler (ADW).

 **Aufgabenstellung UE4.1:** Finden Sie anhand der Abbildung 6.14 die hierfür infrage kommenden Ports heraus. Begründen Sie Ihre Entscheidung.

Eine mögliche Schaltung für das zuvor erwähnte Beispiel ist Abbildung 6.22 dargestellt. Je nach angelegter Spannung, leuchtet eine der LEDs auf. Es gilt: Analoge Eingangsspannung größer 3V, dann LED3 ein; kleiner 2V, dann LED2 ein; sonst LED3 ein.

Seit der Theorieeinheit ist Ihnen die prinzipielle Arbeitsweise eines ADW bekannt. Damit man diesen im Programm verwenden kann, gilt es einige Register des ADW zu konfigurieren.

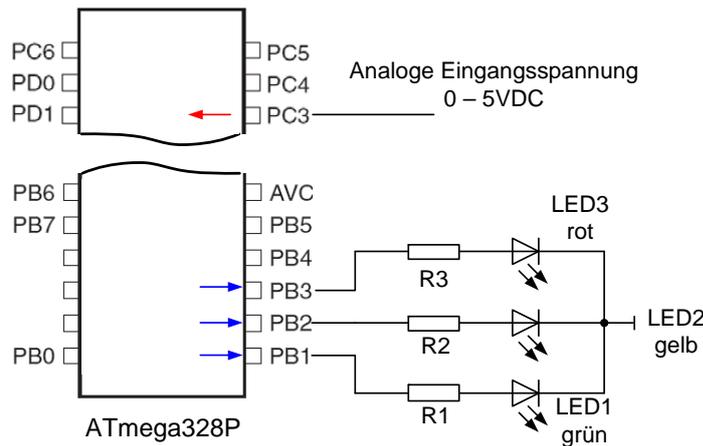


Abbildung 6.22: AD-Wandler-Beispielsschaltung (Beschaltung von AREF u. AVCC nicht dargestellt, interne Referenzspannung gewählt)

ADMUX	0	1	0	0	0	0	1	1
	REFS1	REFS0	ADLAR	MUX4	MUX3	MUX2	MUX1	MUX0

Abbildung 6.23: Beispiel-Konfiguration des ADW-Konfigurationsregisters ADMUX [85]

Hierzu zählen das „ADMUX“ und das „ADCSRA“. Mit dem ADMUX wird die Referenzspannung als auch der Kanal, an welchem das analoge Signal anliegt, festgelegt. Abbildung 6.23 zeigt die für das oben angeführte Beispiel erforderliche Konfiguration des Registers.

Das Bit „REFS0“ wird gesetzt, da mit der internen Referenzspannung gearbeitet wird, indem am AREF-Pin ein Kondensator angeschlossen ist. Ebenso sind die Bits „MUX0“ und „MUX1“ zu setzen, um den Kanal 3, Pin ADC3/PC3 zu selektieren. Als nächstes gilt es das Register ADCSRA im Detail zu betrachten.

Damit man den ADW verwenden kann, ist dieser zu aktivieren. Das kann mit dem Setzen des Bits „ADEN“ erledigt werden. Da Interrupt-Routinen noch nicht behandelt wurden in der Übung, wird auf dieses Feature des ADW (noch) nicht zurückgegriffen und das entsprechenden Konfigurations-Bit (ADIE) auf „0“ gesetzt. Wie bekannt ist, kann der ADW mit dem Tempo der Mikrocontroller-CPU nicht schritthalten, daher muss dieses reduziert werden. Hierfür dienen die Bits „ADPS2“ bis „ADPS0“, der so genannte „Prescaler“ (dt. Vorteiler). Da der ADW der AVR-Familie einen Arbeitsbereich zw. 50kHz und 200kHz hat und wir von einem CPU-Takt von 8MHz ausgehen, ergibt sich ein Bereich zw. 40 und 160. Da es den nächst kleineren Prescaler-Wert in diesem Bereich zu wählen gilt, ist das 64. Die hierfür im Datenblatt festgelegte Bitfolge lautet 110, die es bei ADPS2, ADPS1 und ADPS0 einzutragen gilt. Somit arbeitet der ADW mit einer Frequenz von 125kHz bei einem CPU-Takt von 8MHz. Diese Bits können beispielsweise gleich bei der Initialisierung des ADW gesetzt werden (siehe Abbildung 6.24, graue Markierungen). Gleiches gilt für die Interrupt-spezifischen Bits ADIE und ADIF sowie für ADATE und dem „Ein-/Ausschalter“ des ADW, ADEN.

ADCSRA	1	0	0	0	0	1	1	0
	ADEN	ADSC	ADATE	ADIF	ADIE	ADPS2	ADPS1	ADPS0

Abbildung 6.24: Beispiel-Konfiguration des ADW-Konfigurationsregisters ADCSRA [85]

Um den ADW zu verwenden, geht man in der Regel folgendermaßen in einem Programm vor:

- ADW initialisieren (1-mal)
- Messung starten
- Überprüfen, ob die Messung abgeschlossen ist
- Messwert auslesen

Um eine Messung zu starten, ist das Bit ADSC auf „1“ zu setzen. Vorsicht beim Zugriff, denn die anderen Bits dürfen keinesfalls überschrieben werden. Denken Sie an die Bitmaskierungstechniken! Der ADW beginnt dann mit der Messung. Ist die Messung abgeschlossen, setzt dieser das Bit ADSC auf „0“ und befüllt die Variable „ADCW“ mit dem Messwert, die es weiterzuverarbeiten gilt. Hierbei muss man, wie bei der Theorieeinheit festgestellt, den Wert entsprechend umrechnen. Die nächste Messung wird wiederum mit dem Setzen des ADSC auf „1“ gestartet usw.

 **Aufgabenstellung UE4.2:** Schreiben Sie ein Programm, das das einleitend beschriebene Verhalten (mit dem Erfassen des Analogsignals und dem Schalten dreier LEDs) implementiert. Als Schaltungsgrundlage dient Abbildung 6.22. Verwenden Sie das Datenblatt des Controllers, um weitere Informationen einzuholen. Überlegen Sie, wie Sie das analoge Eingangssignal mit relativ wenig Aufwand realisieren können.

Bei der vorhergehenden Aufgabenstellung werden bereits Interrupts erwähnt. Es stellt sich die Frage, wozu diese nun im Detail verwendet werden können?

Wie Sie dem Listing 6.7 (aus der Aufgabenstellung UE4.2) entnehmen können, wird in der Zeile 18 laufend der Status des Bits ADSC abgefragt, um festzustellen, ob die Messung abgeschlossen wird. Ähnliches haben Sie bereits beim Abfragen von Tasten erfahren. Diese Vorgehensweise nennt man „Polling“ (dt. Abfragen). Das ist grundsätzlich eine Möglichkeit, die aber vor allem bei größeren Programmen zu Hindernissen führen kann. Eine Alternative stellen die sogenannten Interrupts dar.

Nachdem der theoretische Hintergrund bereits in der Theorieeinheit erarbeitet wurde, erfolgt die Betrachtung auf Programmebene. Vereinfacht ausgedrückt definiert man im Programm eine Funktion, die immer dann aufgerufen wird, wenn ein bestimmtes Ereignis (z.B. ein Eingang wechselt den Zustand, weil ein Taster betätigt wurde, oder der ADW ist mit einer Messung fertig etc.) eintritt. Das bedeutet, die CPU bricht den aktuellen Ausführungspfad im Programm ab, widmet sich dieser Funktion - auch Interrupt Service Routine (ISR) genannt - und kehrt nach deren Bearbeitung zum eigentlichen Programmpfad zurück. Hinsichtlich der zu definierenden Funktion gilt es noch zu erwähnen, dass der Funktionsname als auch der Parametername nicht frei wählbar sind. Diese sind von der Plattform beziehungsweise vom Compiler vorgegeben. Das Listing 6.6 zeigt ein Beispiel für die ISR des ADW. Die Funktion „`SIGNAL(SIG_SPI)`“ wird immer dann aufgerufen, wenn eine Messung fertig ist, aber das wissen Sie ja bereits.

Listing 6.6: Anwendung der Interrupt Service Routine für den ADW

```
1 #include <avr/io.h>
2 #include <avr/interrupt.h>
3
4 SIGNAL(SIG_SPI){ //Definition der/einer Interrupt Service Routine
5     // do something
6 }
7
8
9 int main(void){
10
11
12
13     while(1){
14
15         //Some code...
16
17     }
18     return 0;
19 }
```

 **Aufgabenstellung UE4.3:** Adaptieren Sie das Programm aus der Aufgabenstellung UE4.2 dahingehend, dass das Abfragen des Analogsignals (Kanal 3) in der ISR durchgeführt wird. Ändern Sie anhand des Datenblatts die Konfiguration relevanter Register.

Interrupts können, wie bereits erwähnt, auch in Kombination mit einer Taste eingesetzt werden. Hierfür sind sogenannte „externe Interrupts“ erforderlich. Die sind begrenzt und nur an bestimmten Eingängen des Controllers verfügbar. Außerdem ist dann auch eine andere ISR zu implementieren.

 **Aufgabenstellung UE4.4:** Eruiieren Sie anhand des Datenblatts, (i) welche Controller-Eingänge hierfür infrage kommen und (ii) wie die entsprechenden Interrupt-Register „MCUCR“ und „GICR“ der externen Interrupts zu konfigurieren sind. Ziel ist, dass bei einer fallenden Flanke eines Tasters am Pin PD2 die Portvariable PORTB um eins inkrementiert wird. Der aktuelle Wert ist am LCD zu visualisieren.

Lösungsvorschläge

Aufgabenstellung UE4.1 zielt auf das Erkennen des Ports C ab. Weiters sollte erkannt werden, dass der ADW des ATmega328P über sechs Kanäle verfügt, die von ADC0 bis ADC6 reichen.

Bei der Aufgabenstellung 4.2 gilt es grundsätzlich die Funktionsweise des ADW zu verstehen, also wie dessen Register zu konfigurieren sind. Diese Aufgabenstellung erfordert hinsichtlich der Konfiguration des ADMUX-Registers ausgeprägte Elektronikkenntnisse, die nicht im Zentrum der Übung stehen. Daher erfolgt ein konkreter Lösungsvorschlag. Grundsätzlich sind bei dieser Übung zwei Aspekte von Bedeutung: die allgemeine Implementierung der ADW-Programmlogik (z.B. Messung starten, auf Ergebnis warten etc.), die auf unterschiedlichste Art und Weise bewerkstelligt werden kann sowie das Verständnis der Weiterverarbeitung des

Messwertes. Das setzt beispielsweise voraus, dass die SchülerInnen wissen, dass es sich beim ADW des ATmega328P um einen Wandler mit 10 Bit Auflösung handelt. Das hat die Konsequenz, dass der analoge Wertebereich in 1023 (1024-1) gleich große Abschnitte zu unterteilen ist, wodurch zwangsläufig die Messgenauigkeit darunter leidet. Daher ist beispielsweise die Angabe der x-ten Kommastelle nicht immer sinnvoll. Listing 6.7 zeigt eine Minimalversion der Aufgabenstellung UE2.4.

Listing 6.7: ADW Beispielsanwendung

```

1 #include <avr/io.h>
2 #include <util/delay.h>
3
4
5 int main(void){
6
7     DDRB=(1<<PB1)|(1<<PB2)|(1<<PB3); // LED-Pins am Port B auf Ausgang setzen
8
9     ADMUX=(1<<REFS0) | (1<<MUX1) | (1<<MUX0); //Interne Ref.-spg. u. Kanal 3 setzen
10    ADCSRA=(1<<ADEN) | (1<<ADPS2) | (1<<ADPS1); //ADW starten u. Prescaler setzen
11
12    unsigned int result=0;
13
14    while(1){
15
16        ADCSRA |= (1<<ADSC); //Messung starten
17
18        while(ADCSRA & (1<<ADSC)); //Warten, bis die Messung abgeschlossen ist
19
20        result=ADCW; //Ergebnis auslesen...
21
22        if(result > (3*1024)/5) PORTB=(1<<PB3); //...u. auswerten
23
24        else if(result < (2*1024)/5) PORTB=(1<<PB2);
25
26        else PORTB=(1<<PB1);
27    }
28    return 0;
29 }
```

Bei der Aufgabenstellung UE4.3 ist das Register ADCSRA anzupassen und die ISR zu implementieren. Hinsichtlich Aufgabenstellung UE4.4 gilt es festzuhalten, dass den SchülerInnen eine Tabelle mit sämtlichen Bezeichnern für die ISR auszuhändigen ist. Zu erwähnen ist weiters, dass alternativ zum Funktionsbezeichner „SIGNAL“ auch „INTERRUPT“ eingesetzt werden kann. In diesem Fall kann die Servicefunktion unterbrochen werden, bei SIGNAL nicht.

Anmerkungen

Als Additum würde sich bei der letzten Aufgabe wiederum der Schaltungsaufbau anbieten. Bei den ADW-Beispielen ist dieser unumgänglich, da der HAPSIM die ADW-Funktion nicht unterstützt.

6.3.5 UE5 - Finalisierung Gruppenprojekt 2

Nachdem die SchülerInnen parallel zu den Übungseinheiten zwei Projekte zu absolvieren haben, soll mit der Einheit UE6 die Möglichkeit geboten werden, das zweite Projekt (siehe Kapitel 6.4) im Rahmen des Unterrichts zu finalisieren, bevor die Vorstellung der Ergebnisse in der Theorieeinheit „T6 - Projektpräsentation“ erfolgt. Diese Annahme setzt jedoch eine entsprechende Berücksichtigung im Stundenplan, beziehungsweise beim allgemeinen Ablauf voraus.

6.4 Projekte

Ergänzend zu den Theorie -und Übungseinheiten gilt es zwei Projekte zu absolvieren (siehe Abbildung 6.1, Projekt 1 und Projekt 2). Nachdem ein Mindestmaß an Grundlagen zur eigenständigen Absolvierung erforderlich ist, ist der Beginn für „Projekt 1“ Mitte der zweiten Übungssequenz vorgesehen. Zu diesem Zeitpunkt sollten die Grundlagen des Portzugriffs soweit verinnerlicht sein, dass ein Projekt mit nachfolgender Aufgabenstellung durchführbar ist. Hierbei handelt es sich nicht unbedingt um einen klassischen Projektauftrag, der konkrete Vorgaben an das Endprodukt richtet, sondern vielmehr um Vorgaben auf Codeebene, die es zu berücksichtigen gilt (Anforderung D4). Der Projektauftrag könnte wie folgt lauten:

Projekt 1 - Disco, Disco

Nachdem Sie bereits erste Bekanntschaft mit den Herausforderungen des Portzugriffs bei Mikrocontrollern gemacht haben, gilt es dieses Know-how bei einer Kreativaufgabe anzuwenden. Für diese gibt es ausschließlich Vorgaben auf Code- und Hardwareebene, die sich wie folgt gestalten: Das Programm muss

- aus mindestens 30 LOCs (Lines of Code) bestehen (Kommentarzeilen zählen nicht).
- Hierbei sind zwei unterschiedliche Ports zu verwenden; wobei für die Portzuteilung folgender „Algorithmus“ gilt: $\text{Ergebnis} = \langle \text{Katalognummer} \rangle \bmod 3 + 1$
 - Ergebnis = 1: Port B und C
 - Ergebnis = 2: Port C und D
 - Ergebnis = 3: Port D und B
- Insgesamt sind acht LEDs zu verwenden, die bei einem Port an die Pins 0 bis 3 und beim anderen an die Pins 4 bis 7 zu schalten sind.
- Setzen Sie mindestens eine weitere Schleife (ergänzend zur Arbeitsschleife) sowie eine *if-else if-else*- oder *switch*-Kontrollstruktur ein.

Das Wichtigste: Was Sie mit den LEDs machen, obliegt Ihrer Kreativität!

Diese Aufgabenstellung bezweckt einerseits, dass die SchülerInnen ihrer Kreativität (Anforderung B3) freien Lauf lassen können. Andererseits sind jene Themenschwerpunkte enthalten, die bis zum gegenwärtigen Zeitpunkt Bestandteil der Übung waren. Somit sind die SchülerInnen gefordert, die Übungsinhalte durch außerschulische Arbeit - sozusagen als Hausübung - zu vertiefen. Weiters wird mit der dynamischen Port-Zuweisung bei der Aufgabenstellung der Plagiarismus zumindest in Ansätzen erschwert (Anforderung D5). Es ist zu erwarten, dass die SchülerInnen einen gewissen Spieltrieb, wenn nicht sogar Wettbewerbsgedanken entwickeln, was wiederum als intrinsische Motivationsbildung zu werten ist (Anforderung B3).

Das „Projekt 2“ zielt auf die Entwicklung einer typischen Anwendung ab. Als Referenzbeispiel dient das Prinzipschaltbild aus Abbildung 6.6. Als Sozialform kommt im Gegensatz zu Projekt 1 die Gruppenarbeit zum Einsatz. Die maximale Gruppengröße liegt bei drei SchülerInnen. Die Gruppenzusammenstellung obliegt den SchülerInnen.

Projekt 2 - Messanwendung

Ziel dieses Projektes ist die Entwicklung einer Messanwendung mit einfachem Bedienelement und LCD-Ausgabe. Im Detail gilt es

- ein Programm zu schreiben, das die erforderlichen Funktionalitäten implementiert,
- die dazugehörige Schaltung aufzubauen,
- eine entsprechende Projektdokumentation zu fertigen und
- die Ergebnisse im Rahmen eines Kurzvortrages vorzustellen.

Folgende Sensoren stehen zur Auswahl bereit: Temperatur- und Feuchtigkeitssensoren, Fotowiderstände, Ultraschall-Entfernungssensoren, Drucksensoren und Alkohol-Gas-Sensoren. Die Sensorwahl obliegt ebenfalls der Gruppe. Für alle Sensoren stehen Beispielschaltungen zur Verfügung.

An die Anwendung werden folgende Anforderungen gestellt:

- Messungen müssen manuell über das Bedienelement gestartet werden (keinen Automatikmodus!).
- Die letzten 10 Messwerte sollen temporär gespeichert und mittels Bedienelements abrufbar sein.
- Die Bedienung muss mit maximal fünf Tasten möglich sein.

Hinsichtlich der zu erstellenden Projektdokumentation gelten die Vorgaben der Kompetenzbereiche „Projektmanagement“ und „Softwareentwicklungsprozess“. Überlegen Sie hierbei den Einsatz einer „Statemachine“. Fertigen Sie die entsprechenden Diagramme (z.B. Zustandsdiagramm, Kontrollflussdiagramm) und dokumentieren Sie zentrale Bestandteile des Quellcodes.

Wie beim Projekt 1 wird den SchülerInnen ein gewisser Entscheidungsfreiraum eingeräumt mit dem Ziel der Motivationsförderung, da sich unter diesen Umständen die SchülerInnen eher mit dem Projekt identifizieren. Da es sich um ein Gruppenprojekt handelt, das parallel zum Regelunterricht des Kompetenzbereichs abzuwickeln ist, könnte die Projektplanung und/oder die Projektumsetzung interdisziplinär erfolgen (Anforderung B6). Wie in Kapitel 5.2.2, Szenario 1 oder 2, beschrieben, bieten sich hierfür mehrere Kompetenzbereiche an.

Anzumerken gilt es den Aufwand für die Lehrkraft, den offene Projektaufgabenstellungen mit sich bringen.

6.5 Übungskatalog

Ergänzend zu den Übungseinheiten ist der Einsatz eines Übungskatalogs vorgesehen. Dieser stellt für jeden Themenschwerpunkt mehrere Übungsbeispiele zur Verfügung, die grundsätzlich freiwillig zu absolvieren sind. Sinnvoll erscheint, wenn es drei bis fünf Übungsbeispiele für jeden Themenbereich gibt, deren Absolvierung in entsprechendem Ausmaß zur Gesamtbeurteilung beiträgt. Die Abgaben könnten digital erfolgen. Um ausschweifenden Plagiarismus zu verhindern, sind stichprobenartige Kontrollen im Rahmen der Übungseinheiten zielführend, bei welchen die SchülerInnen die Lösung eines Beispiels präsentieren. Das setzt ein Mindestmaß an Verständnis für die jeweilige Aufgabenstellung voraus - auch wenn es eins zu eins übernommen wurde.

Wie bei den Projekten ist diese Maßnahme mit einem erheblichen Mehraufwand für die Lehrkraft verbunden - vor allem dann, wenn man den Übungskatalog von Jahr zu Jahr adaptiert.

6.6 Leistungsbeurteilung

Wie der Abbildung 6.1 zu entnehmen ist, erfolgt mit „P1“ eine Überprüfung des Vorwissens zu Beginn; wobei hier keine Prüfung im herkömmlichen Sinn gemeint ist. Es sollte vielmehr ein Diagnosecheck sein, um die Ist-Situation in Erfahrung zu bringen. Die Durchführung ist grundsätzlich anonym, mit der Option, seine Identität preiszugeben, um möglicherweise Bonuspunkte zu sammeln. Negative Ergebnisse sollten bei der Beurteilung nicht berücksichtigt werden. Relevante Wissensbereiche sind beispielsweise laut Abbildung 5.4 Zahlensysteme. Hierzu zählen primär das Binär- und Hexadezimalsystem sowie deren Umwandlung. Ebenfalls interessant ist, inwieweit die Elektronik- oder Programmierkenntnisse ausgeprägt sind. Bei der hardwarenahen Programmierung kommt beispielsweise der Datentypenauswahl eine größere Bedeutung zu als bei der klassischen OOP, mit welcher die SchülerInnen bisher primär konfrontiert waren.

Einen zentralen Bestandteil der Beurteilung stellt die Wissensüberprüfung „P2“ am Ende des Themenblocks dar; wobei das nicht zwingend eine schriftliche Überprüfung sein muss. Unter Umständen kann eine Überprüfung, ob die Inhalte eines Lernvertrags (vgl. Kapitel 4.5.1) eingehalten wurden, erfolgen. Der Zeitpunkt erscheint aber aufgrund des Themenwechsels als sinnvoll (siehe Abbildung 6.1). Beim Einsatz einer schriftlichen Überprüfung ist

es von Bedeutung, konkrete Kompetenzen zu ermitteln. Abbildung 6.25 zeigt ein Beispiel, das sich thematisch mit den Inhalten der Übung 6.3.3 deckt und die Anwendung der in dieser Einheit erworbenen Kompetenzen erfordert.

Angenommen, an den Pins 4 bis 7 des Port B sind Taster angeschlossen, deren Zustand Sie im Programm abfragen müssen. Wie greifen Sie auf die entsprechende Portvariable zu (siehe „tmp=“) und wie lauten demnach die „case“-Bedingungen?

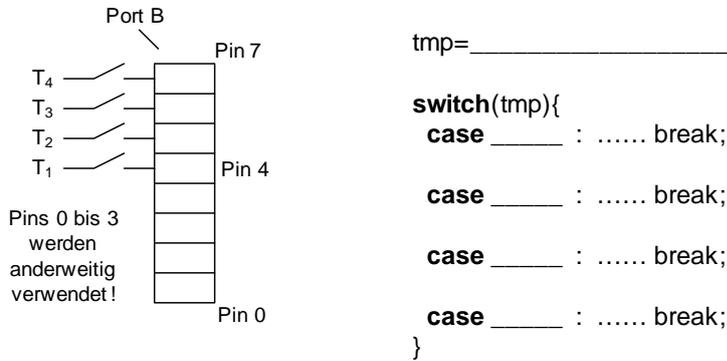


Abbildung 6.25: Testbeispiel, um konkrete Handlungskompetenzen zu erfassen

Der/die SchülerIn muss erkennen, dass mithilfe von Bitmasken die entsprechenden Bits der Portvariable „PORTB“ zu selektieren sind und deren Zahlenwert (16 bis 128) in der „switch“-Kontrollstruktur abzufragen ist.

Nachdem in den Übungseinheiten konkrete Ergebnisse in Form von Programmen erzielt werden, ist der Einsatz der Portfoliotechnik (vgl. Kapitel 4.5.3) vorstellbar - vor allem in Ergänzung mit den freiwilligen Übungsaufgaben des Übungskataloges, die ebenfalls gemäß eines festgelegten Schlüssels in die Gesamtbeurteilung einfließen. Zusammenfassend lässt sich festhalten, dass es eine Vielzahl von Möglichkeiten gibt, um zu einer aussagekräftigen Gesamtnote zu gelangen (Anforderung D8).

7.1 Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit der Entwicklung didaktischer Entwürfe für den Kompetenzbereich Industrielle Informationstechnik (INIT) einer Höheren Technischen Lehranstalt (HTL) für Informationstechnologie (IT). Den Ausgangspunkt bilden hierbei die einleitend formulierten Fragestellungen in Kapitel 1. In Hinblick darauf liegen die inhaltlichen Schwerpunkte auf der Darstellung von Bildungsstandards und der damit verbundenen Kompetenzorientierung, der Untersuchung der Ausgangssituation und dem Entwurf eines didaktischen Gesamtkonzeptes sowie ausgewählter Unterrichtseinheiten.

Nachdem der Fachbereich IT einer HTL im Fokus der Ausarbeitung steht, wird dieser Schultyp einleitend vorgestellt. Zusammenfassend lässt sich konstatieren, dass dieser Schultyp durchaus etwas Besonderes darstellt: einerseits aufgrund der großen Tradition und dem Zuspruch, den dieser Schultyp erfährt, andererseits, weil ein internationaler Vergleich nur bedingt möglich ist. Ebenso behandelt Kapitel 2 die Lerntheorie „Konstruktivismus“, die als Grundlage für die Kompetenzorientierung angesehen werden kann. Nur durch aktive Beteiligung und eigene Erfahrungen vonseiten der SchülerInnen kann ein nachhaltiger Aufbau von Wissensstrukturen erfolgen. Darüber hinaus wird veranschaulicht, dass Lernen auf unterschiedlichste Art und Weise erfolgen kann und hierbei Faktoren wie beispielsweise die Vergessenskurve von Bedeutung sind.

Das Projekt „Bildungsstandards in der Berufsbildung“ des Bundesministeriums für Unterricht, Kunst und Kultur (BMUKK) wird in Kapitel 3 am Beispiel der Fachrichtung IT vorgestellt. Bei der Betrachtung des Projekt-Ist-Standes ist deutlich geworden, dass wesentliche Schritte bei der Entwicklung der IT-Bildungsstandards vollzogen wurden, Details jedoch noch ausstehen. Das manifestiert sich zum Beispiel an der überschaubaren Anzahl prototypischer Unterrichtsbeispiele, die die Dimension eines Deskriptors aufzeigen sollen. Wie sich das Thema Unterricht im Kontext der Kompetenzorientierung gestaltet, ist Inhalt von Kapitel 4. Zentrale Inhalte sind hierbei die Definition von Merkmalen eines kompetenzorientierten Unterrichts sowie die Vorstellung handlungsorientierter Unterrichtsmethoden. Diese zeichnen sich vor allem durch eine maximale Schülerzentrierung aus. Die Lehrkraft übernimmt hierbei

mehr und mehr die Rolle des Coaches. Wichtige Vertreter dieser Methoden sind allgemein die Partner- oder Gruppenarbeit sowie spezifische Formen der Gruppenarbeit (z.B. Gruppenpuzzle o. Projektarbeit). Neben der Unterrichtsdurchführung werden auch entsprechende Methoden zur Leistungsbeurteilung vorgestellt, die den Ansprüchen der Kompetenzorientierung gerecht werden.

Anforderungen an die Unterrichtskonzeption werden in Kapitel 5 erarbeitet. Im ersten Schritt werden hierbei die Vorgaben vom Gesetzgeber als auch jene Rahmenbedingungen, welche auf die schulspezifische Implementierung des Kompetenzbereichs INIT an der HTL-Krems zurückzuführen sind, analysiert. Im zweiten Schritt erfolgt eine Erhebung des State of the Art hinsichtlich „Embedded Systems als Unterrichtsgegenstand“. Die Analyseergebnisse werden abschließend in konkrete Anforderungen überführt. Der Übersicht halber werden vier Themencluster - von „Cluster A: Pädagogik und Lernpsychologie“ bis hin zu „Cluster D: Embedded Systems als Unterrichtsgegenstand“ - gebildet.

Im letzten Abschnitt dieser Arbeit erfolgt die Entwicklung eines didaktischen Gesamtkonzepts für den 4. Jahrgang des Kompetenzbereichs INIT mit der Konkretisierung einzelner Unterrichtseinheiten. Gemäß den definierten Anforderungen in Kapitel 5.4 erfolgt eine Unterteilung in Theorie- und Übungseinheiten, die sich inhaltlich an zentralen Themenblöcken der Mikrocontrollertechnik orientieren. Somit ist eine optimale Verzahnung von Theorie und Praxis sichergestellt. Das Gesamtkonzept umfasst weiters Projektvorschläge, einen Übungskatalog sowie einen Vorschlag zur Umsetzung der Leistungsbeurteilung.

7.2 Diskussion

Die einleitend formulierten Fragen können nun wie folgt zusammengefasst beantwortet werden:

F1: Wie muss Unterricht gestaltet sein, um die Bildungsstandards des Kompetenzbereichs Industrielle Informationstechnik zu erfüllen?

In Kapitel 6 wird ein didaktisches Gesamtkonzept vorgestellt (siehe Abbildung 6.1), mit welchem die Vorgaben der Bildungsstandards grundsätzlich erzielt werden können. Jedoch gilt auch in diesem Fall, wie so oft im Bereich der Informatik: „Viele Wege führen nach Rom!“. Nachdem die schulspezifische Implementierung des Lehrplans eine Unterteilung in Theorie- und Praxisunterricht vorgibt, ist diese Grundlage der Planung. Damit sich der Theorie- und Praxisunterricht optimal ergänzen, werden drei Themenblöcke (B1 bis B3) definiert, welche die Inhalte vorgeben. Eine Konkretisierung aus dem jeweiligen Blickwinkel erfolgt in den einzelnen Einheiten. Generell ist anzumerken, dass die Theorieeinheiten überaus offen gestaltet sind und somit erheblichen Raum für konstruktivistische Prozesse bei den SchülerInnen bieten. Grundlage bilden handlungsorientierte Unterrichtsmethoden, die abwechselnd zum Einsatz kommen. Bei den Theorieeinheiten steht das Konzeptwissen im Vordergrund, konkretes Produktwissen erfahren die SchülerInnen bei der Programmierung eines AVR-Mikrocontrollers in den Übungseinheiten. Jedoch wird auch hier auf allgemein gültige Konzepte, wie beispielsweise der Bitmanipulation, Wert gelegt. Im Gegensatz zu den Theorieeinheiten

sind die Übungseinheiten hinsichtlich des Ablaufs und der zu erzielenden Ergebnisse konkreter. Das ist der Tatsache geschuldet, dass es sich um eine Einführung in die Mikrocontrollerprogrammierung handelt. Es gilt dafür Sorge zu tragen, dass nach den Mikrocontrollereinheiten ein Mindestmaß an (Fach-)Kompetenz vorhanden ist. Um hierfür eine Ausgangsbasis zu schaffen, dient der Wissenscheck zu Beginn. Dieser soll eventuelle Wissenslücken aufzeigen, auf die es entsprechend zu reagieren gilt.

Weiters gilt es anzumerken, dass für den Kompetenzbereich INIT ausschließlich ein prototypisches Unterrichtsbeispiel (der Kategorie „Entwickeln“) verfügbar ist. Das ist insofern problematisch, da die Formulierungen der Bildungsstandards überaus allgemein gehalten sind und somit viel Interpretationsspielraum hinsichtlich der konkreten Realisierung bieten. Außerdem wird generell deutlich, dass die Vorgaben des Lehrplans als auch der Bildungsstandards durchaus ambitioniert sind in Anbetracht des zur Verfügung stehenden Zeitbudgets. In diesem Zusammenhang stellt sich die Frage, inwieweit der Themenschwerpunkt SPS mit dem Ausbildungsprofil einer IT-HTL korreliert und ob es nicht zielführender wäre, den Fokus eher auf die Mikrocontrollertechnik zu richten.

F2: Wie kann die individuelle Lernzielerreichung auf fachlicher Ebene sichergestellt werden, ohne die Entwicklung personaler und sozialer Kompetenzen zu vernachlässigen?

Dem didaktischen Gesamtkonzept liegt generell die Überlegung zugrunde, dass der Theorieunterricht aufgrund der eingesetzten Unterrichtsmethoden den wesentlichen Beitrag zur Bildung der personalen als auch sozialen Kompetenz beiträgt. Demgegenüber sind die Übungseinheiten so konzipiert, dass die individuelle Leistungserbringung im Vordergrund steht. Als Grundlage dienen Aufgabenstellungen, die konkrete Ergebnisse einfordern. Somit ist jeder Schüler/jede Schülerin angehalten, diese als Einzelaufgabe zu bewältigen. Da die Aufgabenstellungen Leitprogrammcharakter aufweisen, ist eine selbstständige Bearbeitung möglich - zumindest für begabte SchülerInnen. Das erlaubt der Lehrkraft eine Fokussierung der Aufmerksamkeit auf jene SchülerInnen, die weniger begabt sind und dadurch mehr Unterstützung benötigen. Eine weitere Maßnahme ist der Übungskatalog. Es ist davon auszugehen, dass auch bei den Gruppenarbeiten ein Know-how-Transfer zwischen den SchülerInnen erfolgt. Außerdem kann das „Projekt 2“ (siehe Kapitel 6.4) durchaus auch als Individualisierungsmaßnahme verstanden werden. Bei den Aufgabenstellungen der Übung sind, wie bereits erwähnt, konkrete Ergebnisse zu erbringen, dies ist beim „Projekt 2“ nur bedingt der Fall - die SchülerInnen können die Umsetzung je nach persönlichen Fähigkeiten individuell gestalten. Es gilt jedoch eine Mindestanforderung zu erfüllen.

F3: Welche Bedeutung wird der schulspezifischen Implementierung des Lehrplans (beim Einsatz handlungsorientierter Methoden) zuteil?

Die Unterteilung in Theorie- und Praxisunterricht bringt grundsätzlich Vor- und Nachteile mit sich. Beispielsweise kann die Zuordnung von „Theorie behandelt allgemeine Konzepte und die Praxis eine konkrete Technologie“ von den SchülerInnen als durchaus positiv aufgenommen werden. Problematisch ist hingegen die 50-Minutentaktung. Die

Erprobung der Konzepte wird zeigen, wie sich die Aufspaltung einer Aufgabenstellung auf mehrere Tage, die beispielsweise auf der Methode Gruppenpuzzle basiert (siehe Kapitel 6.2.4), in der Unterrichtspraxis umsetzen lässt. Ein weiteres Erfolgskriterium beim Einsatz handlungsorientierter Methoden stellt die Klassengröße dar. Ein Gruppenpuzzle mit beispielsweise 30 SchülerInnen erscheint als überaus herausfordernd - für beide Seiten.

Anforderung B5 verlangt ausreichend Zeit zum Reflektieren. Demnach wird das zur Verfügung stehende Zeitbudget nicht immer zur Gänze ausgeschöpft, sodass Zeit für entsprechende Maßnahmen bleibt.

Um die geforderte Interdisziplinarität zu realisieren, sind bereits bei der Planung des Stundtplans entsprechende Maßnahmen zu tätigen. Das setzt intensive Koordinierungsmaßnahmen aufseiten der betroffenen Lehrkräfte voraus.

F4: Wie muss der Unterricht gestaltet sein, damit Kreativität und intrinsische Motivation entstehen können?

Der überaus offene Ansatz bei der Durchführung der Theorieeinheiten bietet, wie bereits angeführt, entsprechend Spielraum zur persönlichen Entfaltung. Ebenso der Motivation zuträglich ist der Umstand, dass die eingesetzten Methoden aktives Handeln in unterschiedlichen Rollen erfordern. Beispielsweise nehmen die SchülerInnen beim Gruppenpuzzle die Rolle des Lehrenden ein. In diesem Zusammenhang kann davon ausgegangen werden, dass jeder Schüler seine Sache möglichst gut machen möchte und somit auch entsprechend motiviert an die Aufgabenstellung herangeht.

Die Übungen sind so konzipiert, dass die Mikrocontrollertechnik mit praktischer Anwendung im Vordergrund steht und nicht das „Lernen von Embedded-C“. Eine weitere Maßnahme ist der geplante Einsatz eines „Evaluation Kit“. Dadurch können die SchülerInnen ihre Programme auf realer Hardware testen, was für zusätzliche Motivation sorgen soll. Nachdem Projektaufgabenstellungen überaus offen formuliert sind (siehe Projekt 1), beziehungsweise Entscheidungsfreiheit hinsichtlich der einzusetzenden Hardware anbieten (siehe Projekt 2), ist ebenfalls mit einem ausgeprägten Identifikationspotenzial zu rechnen. Im günstigsten Fall entsteht ein Wettbewerbsgedanke, der wiederum motivationsfördernd wirken kann.

Hinsichtlich der Aufgabenstellungen ist noch allgemein zu erwähnen, dass diesen in der Regel eine Frage oder Aussage vorausgeht, die das Interesse der SchülerInnen wecken soll - gemäß dem Motto „Das möchte ich jetzt aber wirklich wissen!“ (Anforderung A2).

A.1 Kompetenzmodell Informationstechnologie



Abbildung A.1: Kompetenzmodell Informationstechnologie [29]

A.2 Programmierung des Arduino Ethernet-Boards

Zur Durchführung der in Kapitel 6.3 vorgestellten praktischen Unterrichtssequenzen ist der Einsatz des Arduino-Ethernet Boards [87] vorgesehen. Um das Board zu programmieren, ist die Anbindung über das *Serial Peripheral Interface* (SPI) des Mikrocontrollers erforderlich. Die hierfür notwendige Steckverbindung ist bereits auf dem Board ausgeführt (siehe Abbildung A.2, Programmierschnittstelle), sodass es nur mehr eines handelsüblichen „USB-Serial-Adapters“ bedarf, der auf das Board aufgesteckt wird. Zur Verwendung des USB-Serial-Adapters sowie zur Programmierung an sich werden einerseits ein Treiber und andererseits der AVR-Downloader *AVRDUDE* [93] benötigt.

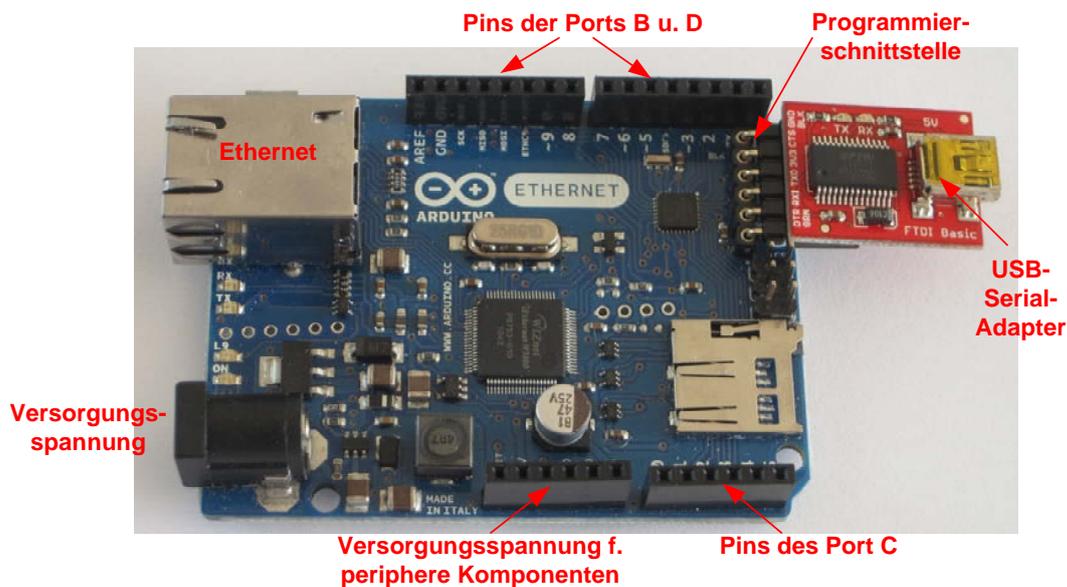


Abbildung A.2: Arduino Ethernet-Board

Der Einfachheit halber empfiehlt sich der Download der Arduino-IDE - auch wenn diese nicht unbedingt zum Einsatz kommt bei den Übungen. Der Vorteil ist aber, dass die zuvor erwähnten Entwicklungswerkzeuge in diesem SW-Paket bereits enthalten sind, da die Arduino-IDE ebenfalls auf diese Komponenten zurückgreift. Im Detail gilt es folgende Arbeitsschritte zu absolvieren:

1. Download und Installation der Arduino-IDE (Quelle: [88]).
2. Download des USB-Driver-Plugins (Datei „Arduino USBSerial.inf“, Quelle: [94]). Diese Datei ist in das Verzeichnis „Drivers“ des Installationsverzeichnisses der Arduino-IDE zu kopieren (siehe Verzeichnis „arduino-1.0“).
3. Das Board mittels USB-Kabel mit dem Rechner verbinden und den Treiber des USB-Serial-Adapters manuell installieren. Falls erforderlich, ist in [95] eine Installationsanleitung enthalten. Nach Abschluss der Installation kann das Board bereits mittels der Arduino-IDE verwendet werden: Arduino-IDE starten, entsprechenden COM-Port unter „Tools >Serial Port“ auswählen, eine der Beispielsanwendungen laden („File >Examples >1.Basics >Blink“) und auf das Board spielen (Schaltfläche „Upload“). Der verwendete COM-Port kann unter Windows über „Systemsteuerung >Gerätemanager“

ausfindig gemacht und geändert werden. Wie der Beispielsfall in Abbildung A.3 zeigt, wird der USB-Serial-Adapter unter „USB Serial Port (COM26)“ gelistet. Durch einen Doppelklick auf diesen Eintrag gelangt man zu den Einstellungen, die entsprechend zu adaptieren sind.

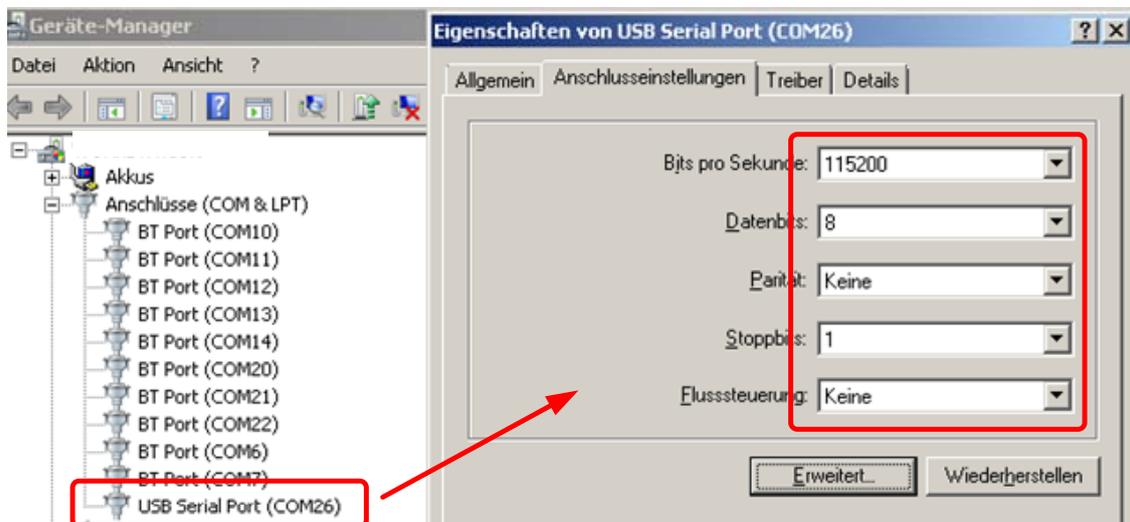


Abbildung A.3: Konfigurationsbeispiel des COM-Ports für den USB-Serial-Adapter

4. Nachdem das AVR Studio zum Einsatz kommt, bedarf es eines weiteren Werkzeuges zur Programmierung des Arduino-Bords: *AVRDUDE*. Dieses ist ebenfalls bereits im Installationsverzeichnis der Arduino-IDE enthalten (siehe „hardware >tools >avr >bin“). Vor der Verwendung gilt es noch die Konfigurationsdatei „avrdude.conf“ zu ergänzen. Diese befindet sich im Verzeichnis „hardware >tools >avr >etc“. Im Bereich der „Programmer“ gilt es einen entsprechenden Eintrag zur Verwendung des USB-Konverters zu erstellen (siehe Abbildung A.4).

```

946 programmer
947     id      = "ftdifriend";
948     desc   = "design ftdi adapter, reset=dtr sck=tx mosi=rts miso=cts";
949     type   = serbb;
950     reset  = ~4;
951     sck    = ~3;
952     mosi   = ~7;
953     miso   = ~8;
954 ;
955

```

Abbildung A.4: Ergänzung der AVRDUDE-Konfigurationsdatei „avrdude.conf“

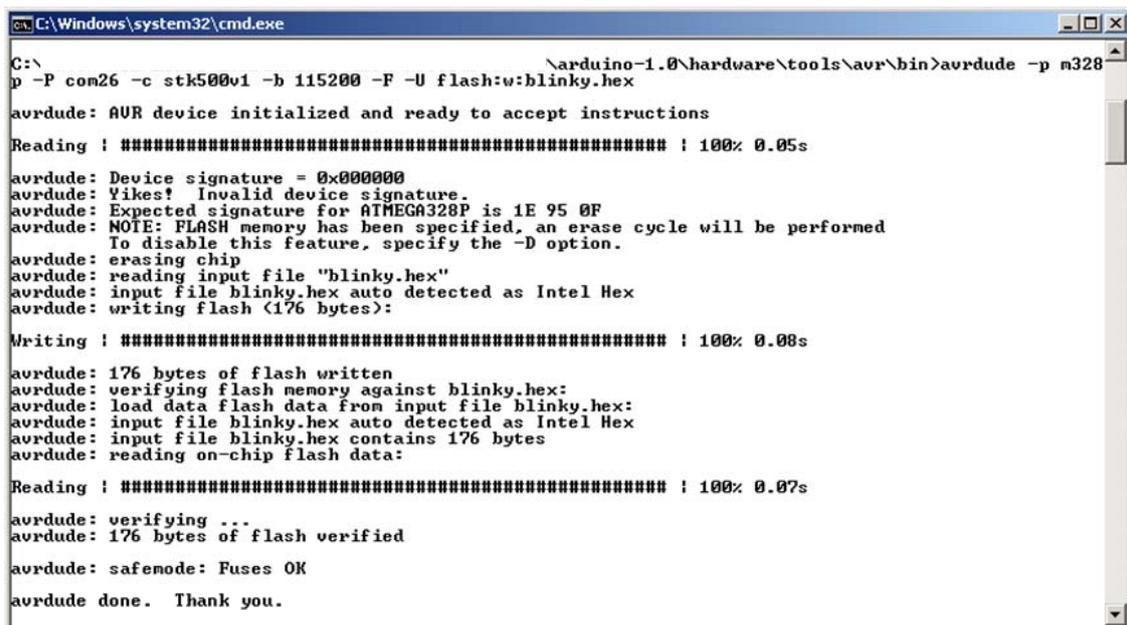
Der Aufruf des Programms *avrdude.exe* erfordert einige Parameter. Um die Arbeit effizienter zu gestalten, empfiehlt sich die Erstellung einer sogenannten Batch-Datei - zumindest unter Windows. Abbildung A.5 zeigt eine mögliche Realisierungsform. Die erste Zeile bewirkt einen Reset auf dem Arduino-Board. Der COM-Port ist entsprechend anzupassen. Die Parameter von AVRDUDE gestalten sich wie folgt: -p legt den Prozessor fest, -P den COM-Port, -c den „Programmer“ (da es sich um eine modifizierte Version von AVRDUDE handelt, kann der „Programmer“ *arduino* verwendet

werden; alternativ *stk500v1*), *-b* gibt die Baudrate an (überschreibt den Default-Wert, siehe Abbildung A.3) und *-U* ist der Parameter fürs Programmieren („flash“ ist der Zielspeicher des Programms, *w* bedeutet schreiben und „*blinky.hex*“ ist das Programm, das auf den Controller zu laden ist). Für weiterführende Information siehe AVRDUDE-Dokumentation [93]. Die hex-Datei wird vom AVR Studio beim Build-Vorgang erzeugt und ist im jeweiligen Projektunterverzeichnis „default“ zu finden. In diesem Zusammenhang ist nochmals festzuhalten, dass im AVR-Studio der richtige Prozessor (ATmega328P) sowie eine gültige Taktfrequenz einzustellen sind. Alternativ kann das auf Programmebene durch Präprozessoranweisungen erfolgen.

```
1 mode com26 dtr=on
2 avrdude -p m328p -P com26 -c arduino -b 115200 -F -U flash:w:blinky.hex
```

Abbildung A.5: Batchdatei-Beispiel mit AVRDUDE zum Programmieren des Arduino-Bords

Abbildung A.6 zeigt das Ergebnis einer erfolgreichen Ausführung der Batch-Datei in Abbildung A.5.



```
C:\Windows\system32\cmd.exe
C:\> \arduino-1.0\hardware\tools\avr\bin>avrdude -p m328p -P com26 -c stk500v1 -b 115200 -F -U flash:w:blinky.hex
avrdude: AVR device initialized and ready to accept instructions
Reading : ##### : 100% 0.05s
avrdude: Device signature = 0x000000
avrdude: Yikes! Invalid device signature.
avrdude: Expected signature for ATMEGA328P is 1E 95 0F
avrdude: NOTE: FLASH memory has been specified, an erase cycle will be performed
To disable this feature, specify the -D option.
avrdude: erasing chip
avrdude: reading input file "blinky.hex"
avrdude: input file blinky.hex auto detected as Intel Hex
avrdude: writing flash (176 bytes):
Writing : ##### : 100% 0.08s
avrdude: 176 bytes of flash written
avrdude: verifying flash memory against blinky.hex:
avrdude: load data flash data from input file blinky.hex:
avrdude: input file blinky.hex auto detected as Intel Hex
avrdude: input file blinky.hex contains 176 bytes
avrdude: reading on-chip flash data:
Reading : ##### : 100% 0.07s
avrdude: verifying ...
avrdude: 176 bytes of flash verified
avrdude: safemode: Fuses OK
avrdude done. Thank you.
```

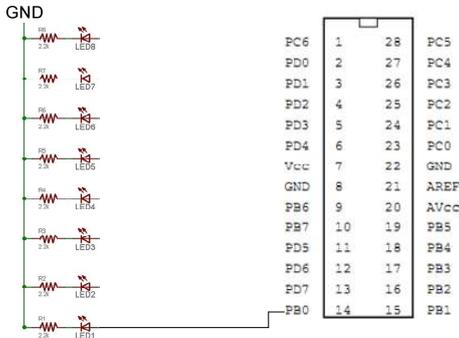
Abbildung A.6: Kommandozeilen-Output von AVRDUDE bei erfolgreicher Ausführung des Programms

A.3 Übungsblatt Bitmanipulation

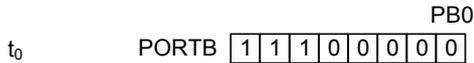
Schuljahr 201x/1y

Übungsblatt „Bitmanipulation“

1 LED am Port B (PORTB):



Pin 1 bis 7 werden anderweitig verwendet und dürfen keinesfalls überschrieben werden!



LED 1 ein:
Maske: _____

Schiebeoperator: _____

Port-Zuweisung: _____



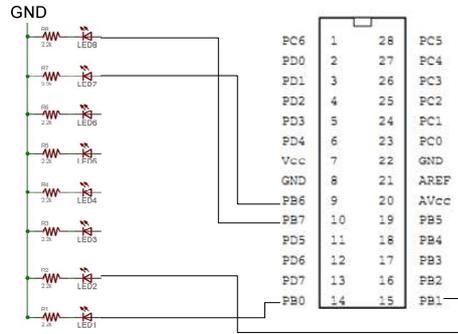
LED 1 aus:
Maske: _____

Schiebeoperator: _____

Port-Zuweisung: _____



4 LEDs am Port B:



Pins 2-5 werden anderweitig verwendet und dürfen keinesfalls überschrieben werden!



LEDs 1 u.2 ein
Maske: _____

Schiebeoperator: _____

Port-Zuweisung: _____



LEDs 1 u. 2 aus
Maske: _____

Schiebeoperator: _____

Port-Zuweisung: _____



LEDs 7 u.8 ein
Maske: _____

Schiebeoperator: _____

Port-Zuweisung: _____



A.4 Ergänzende Unterrichtsmaterialien

<h3>Kontrollstrukturen – Allgemein</h3> <ul style="list-style-type: none"> Es gilt: <table border="1" style="margin-left: 20px;"> <tr> <td>Wert ungleich Null bedeutet <i>wahr</i>: ausführen</td> </tr> <tr> <td>Wert gleich Null bedeutet <i>falsch</i>: nicht ausführen</td> </tr> </table> Bsp.: <pre> 1 while (1) { ... } // Konstante: Schleife immer ausführen 2 3 if(schalter) ... // nur wenn Variable ungleich Null 4 if (a >= 0) ... // nur wenn a größer oder gleich Null 5 if (a == 0 && b == 0) ... // nur wenn a gleich Null UND b gleich Null </pre> 	Wert ungleich Null bedeutet <i>wahr</i> : ausführen	Wert gleich Null bedeutet <i>falsch</i> : nicht ausführen	<h3>Kontrollstrukturen – if und if-else</h3> <ul style="list-style-type: none"> if <ul style="list-style-type: none"> Def.: <code>if (Bedingung) Ja-Anweisung; oder Blockanweisung { }</code> if-else <ul style="list-style-type: none"> Def.: <code>if (Bedingung) Ja-Anweisung oder Block; else Nein-Anweisung oder Block;</code> Bsp.: <pre> 1 if (PORTD & (1 << PD7)) // Wenn Bit 7 v. PORTD 1 ist, dann ja, sonst else 2 { 3 PORTD++; // 20-Bit für High 4 write(); 5 } 6 else // Ende Ja-Block 7 { 8 PORTD--; // Main-Block für Low 9 write(); 10 } // Ende Nein-Block </pre>
Wert ungleich Null bedeutet <i>wahr</i> : ausführen			
Wert gleich Null bedeutet <i>falsch</i> : nicht ausführen			
<h3>Kontrollstrukturen – if-else if-else</h3> <ul style="list-style-type: none"> if – else if – else <ul style="list-style-type: none"> Bsp.: <pre> 1 if(PORTD & (1 << PD7)) // Wenn Bit 7 v. PORTD 1 ist, dann ja 2 { 3 ... 4 } 5 else if(PORTD & (1 << PD6)) // Wenn Bit 6 v. PORTD 1 ist, dann ja 6 { 7 ... 8 } 9 else { // sonst else 10 } </pre> Alternative: switch 	<h3>Kontrollstruktur - switch</h3> <ul style="list-style-type: none"> switch <ul style="list-style-type: none"> Def.: <pre> switch (Auswahl Ausdruck) { case Konstante_1 : Anweisungsfolge_1; break; case Konstante_2 : Anweisungsfolge_2; break; default: Anweisungsfolge_n; break; } </pre> Bsp.: <pre> 1: int plus oder w, y; // Auswahlbedingung und Ergebnis 2: x = P1SD; // Wertzuweisung 3: switch(x) // Fallunterscheidung 4: { 5: case 0 : y = 0x00; break; // Fall x == 0 6: case 1 : y = 0x31; break; // Fall x == 1 7: case 2 : y = 0x32; break; // Fall x == 2 8: default: y = 0xFF; // *Fehlerfall* x > 2 9: } </pre> 		

Abbildung A.8: Ergänzende Unterrichtsmaterialien - Einführung in Embedded C

Abbildungsverzeichnis

2.1	Lernformen (Eigendarstellung nach [10])	5
3.1	Kompetenzmodell der HTL-Fachrichtung Informationstechnologie (Eigendarstellung nach [29])	16
4.1	Bestandteile der Unterrichtsplanung und deren Zusammenwirken [12]	22
4.2	Unterrichtsplanung nach Hartmann [16]	24
4.3	Raum-Setting beim Frontalunterricht [34]	25
4.4	Ablauf eines PBL-Zyklus (Eigendarstellung nach [43] u. [44])	30
4.5	Aufteilung in Experten- und Unterrichtsrunde [16]	33
4.6	Aufbau einer Leitprogrammeinheit [16]	34
5.1	Stundetafel Lehrplan IT [61]	39
5.2	HTBL-Krems im Überblick	46
5.3	Identifikationskriterien für die Interdisziplinarität	49
5.4	Vorwissen - relevante Themenbereiche für die Industrielle Informationstechnik	52
5.5	Verschiedene Methoden, Maßnahmen etc. und deren Einfluss auf die Bereiche „Finance“, „Education“ und „Motivation“ [76]	61
5.6	Komponenten der Lernumgebung [78]	63
5.7	Zusammenfassende Darstellung der Kurs-Meetings [78]	64
6.1	Übersicht der Unterrichtsplanung für den Kompetenzbereich Industrielle Informationstechnik, 4. Jahrgang einer HTL für Informationstechnologie	73
6.2	Schlagwortwolke mit Begriffen zur Internet-Recherche	78
6.3	Schlagwortwolke mit in Beziehung gesetzten Begriffen zur Industriellen Informationstechnik	78

6.4	Blockschaltbild eines Mikrocontrollers	81
6.5	Blockschaltbild des AVR-Mikrocontrollers ATmega328P [85]	82
6.6	Anwendungsbeispiel eines Mikrocontrollers (Prinzipschaltbild)	82
6.7	Blockschaltbild eines Ein-Chip-Mikrorechners (vgl. Mikrocontroller, Eigendarstellung nach [83])	83
6.8	Vier Phasen der Unterrichtssequenz „T4 - Relevante Baugruppen und deren Bedeutung“	88
6.9	Zählen externer Ereignisse	90
6.10	AVR Studio 4 - Einstellungen beim Anlegen eines Projektes [73]	94
6.11	HAPSIM Simulator - Konfiguration der LED-Peripherie für den Port B, Pin 0 [75]	95
6.12	Build-Prozess mit WinAVR	96
6.13	Pin-Belegung des ATmega328P (Bauform 28PDIP) [85]	98
6.14	Blockdiagramm des ATmega328P [85]	99
6.15	Die Port-Register und ihre Zusammenhänge	100
6.16	Bitmanipulation durch den Einsatz von Bitmasken	101
6.17	Bitmanipulation und Bitmaskenerstellung mit Schiebeoperatoren	102
6.18	Beispiele für vordefinierte Portvariablen und Bitbezeichner [90]	103
6.19	Portregister und ihre Zusammenhänge bei der Verwendung des Ports als Eingang	106
6.20	„Hello World“-Ausgabe auf dem LCD [75]	107
6.21	Tasten-Ereignis mittels Bitmanipulation verarbeiten	108
6.22	AD-Wandler-Beispielsschaltung (Beschaltung von AREF u. AVCC nicht dargestellt, interne Referenzspannung gewählt)	110
6.23	Beispiel-Konfiguration des ADW-Konfigurationsregisters ADMUX [85]	110
6.24	Beispiel-Konfiguration des ADW-Konfigurationsregisters ADCSRA [85]	110
6.25	Testbeispiel, um konkrete Handlungskompetenzen zu erfassen	117
A.1	Kompetenzmodell Informationstechnologie [29]	123
A.2	Arduino Ethernet-Board	124
A.3	Konfigurationsbeispiel des COM-Ports für den USB-Serial-Adapter	125
A.4	Ergänzung der AVRDUDE-Konfigurationsdatei „avrdude.conf“	125
A.5	Batchdatei-Beispiel mit AVRDUDE zum Programmieren des Arduino-Bords	126
A.6	Kommandozeilen-Output von AVRDUDE bei erfolgreicher Ausführung des Programms	126
A.7	Vorschlag eines Übungsblatts für die Bitmanipulation durch den Einsatz von Bitmasken mit Schiebeoperator	128
A.8	Ergänzende Unterrichtsmaterialien - Einführung in Embedded C	129

Tabellenverzeichnis

2.1	Einordnung österreichischer Bildungsgänge gemäß den ISCED-Bildungslevels (Eigendarstellung nach [25])	12
5.1	Deskriptoren für den Kompetenzbereich Industrielle Informationstechnik [29]	43
5.2	Deskriptoren für den Kompetenzbereich Industrielle Informationstechnik (Fortsetzung) [29]	44
5.3	Implementierung des Kompetenzbereichs Industrielle Informationstechnik an der HTL Krems	47
5.4	Kompetenzfelder und -bereiche der Fachtheorie und -praxis im Überblick . .	48
5.5	Laborübung des Embedded System-Kurses an der RWTA - Ablauf und Inhalte [71]	58
5.6	Inhalte des Embedded System-Kurs an der TUT [72]	60
5.7	Anforderungen des Bereichs „Pädagogik und Lernpsychologie“	66
5.8	Anforderungen des Bereichs „Kompetenzorientiertes Unterrichten“	67
5.9	Anforderungen des Bereichs „Gesetzliche und institutionelle Vorgaben“ . . .	68
5.10	Anforderungen des Bereichs „Embedded System als Unterrichtsgegenstand“ .	69
6.1	Darstellung und Registerzuweisung eines Zahlenwertes	100
6.2	Konjunktion - Bool'scher Operator \wedge	101
6.3	Disjunktion - Bool'scher Operator \vee	101

Literaturverzeichnis

- [1] BIFIE. „Internetplattform für Bildungsforschung, Innovation u. Entwicklung des österreichischen Schulwesens“, Bildungsstandards - Rechtliche Grundlagen, 2011. [Online]. Available: <https://www.bifie.at/node/48>, visited: Jänner 2012.
- [2] BIFIE. „Internetplattform für Bildungsforschung, Innovation u. Entwicklung des österreichischen Schulwesens“, Standardisierte Reife- und Diplomprüfung, 2011. [Online]. Available: <https://www.bifie.at/srdp>, visited: Jänner 2012.
- [3] Brandner, A. „derStandard.at - Welche Reformen braucht Österreich wirklich?“, Juni 2011. [Online]. Available: <http://derstandard.at/1308186303131/Welche-Reformen-braucht-Oesterreich-wirklich>, visited: Jänner 2012.
- [4] Müller-Funk, W. „derStandard.at - Warum ich das Wort Reform nicht mehr hören kann?“, März 2011. [Online]. Available: <http://derstandard.at/1297819499756/Verweigerung-von-Wandel-Warum-ich-das-Wort-Reform-nicht-mehr-hoeren-kann>, visited: Jänner 2012.
- [5] Salcher, A. „derStandard.at - Einzelne können sehr wohl etwas ändern“, Juli 2011. [Online]. Available: <http://derstandard.at/1310511895520/Autor-und-Aufreger-Einzelne-koennen-sehr-wohl-etwas-aendern>, visited: Jänner 2012.
- [6] BMUKK. „Website Bildungsstandards in der Berufsbildung“. [Online]. Available: <http://www.bildungsstandards.berufsbildendeschulen.at/de/leitideen.html>, visited: Jänner 2012.
- [7] BGBl. Nr. 300/2011. Lehrpläne der Höheren technischen und gewerblichen Lehranstalten (Lehrplan 2011) - Bekanntmachung der Lehrpläne für den Religionsunterricht, September 2011. [Online]. Available: http://www.ris.bka.gv.at/Dokumente/BgblAuth/BGBLA_2011_II_300/BGBLA_2011_II_300.pdf, visited: Jänner 2012.
- [8] HTBL Krems. Website der Höheren Technischen Bundeslehranstalt Krems, 2011. [Online]. Available: <http://www.htlkrems.ac.at>, visited: Jänner 2012.

- [9] Schulorganisationsgesetz. BGBl. Nr. 242/1962, April 2011. [Online]. Available: <http://www.ris.bka.gv.at/GeltendeFassung/Bundesnormen/10009265/Sch0G%2c%20Fassung%20vom%2004.12.2011.pdf>, visited: Jänner 2012.
- [10] H. Schröder. *Lernen - Lehren - Unterricht - Lernpsychologische und didaktische Grundlagen*. Oldenbourg Verlag, 2 edition, 2002, ISBN 3-486-25973-3.
- [11] R. Duit. Piko Briefe - Der fachdidaktische Forschungsstand kurzgefasst, Februar 2010. [Online]. Available: <http://www.ipn.uni-kiel.de/projekte/piko/pikobriefe032010.pdf>, visited: Septmeber 2011.
- [12] L. Humbert. *Didaktik der Informatik*. Teubner Verlag, 2 edition, 2006, ISBN 13 978-3-8351-0112-8.
- [13] E. von Glasersfeld. *Wege des Wissens. Konstruktivistische Erkundungen durch unser Denken*. Carl-Auer-Systeme Verlag, 2 edition, 1997, ISBN 978-3896700049.
- [14] P. Hubwieser. *Didaktik der Informatik*. Springer Verlag, 3 edition, 2007, ISBN 139783540724773.
- [15] C. E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423 and 623–656, July and October 1948.
- [16] W. Hartmann, M. Näf, and R. Reichert. *Informatikunterricht planen und durchführen*. Springer Verlag, 1 edition, 2007, ISBN 139783540344841.
- [17] HTL Innovativ. „internetplattform für berufsbildene Schulen“. [Online]. Available: http://www.htl.at/de/htlat/organisation_standorte.html, visited: Septmeber 2011.
- [18] BMUKK. „Berufsbildende Schulen in Österreich - Eine Informationsbroschüre der Sektion Berufsbildung“, Jänner 2011. [Online]. Available: http://www.abc.berufsbildendeschulen.at/upload/1825_D_Gesamtbroesch%FCre%20%282011%29.pdf, visited: Septmeber 2011.
- [19] BMUKK. „HTL Q-SYS - Das Qualitätsmanagementsystem der technischen Schulen Österreichs“, November 2005. [Online]. Available: http://www.qibb.at/fileadmin/content/qibb/Dokumente/Q-handbooks/Q-HB_II2.pdf, visited: Septmeber 2011.
- [20] A. Schneeberger. Mittelfristige Perspektiven der HTL - Erhebung und Analysen zur Sicherung und Weiterentwicklung der Ausbildungsqualität, Juni 2008. [Online]. Available: http://www.ibw.at/html/rb/pdf/rb_43_schneeberger.pdf, visited: Septmeber 2011.
- [21] Kleine Zeitung. „HTL-Absolventen sind sehr gefragt“, Jänner 2011. [Online]. Available: <http://www.kleinezeitung.at/steiermark/liezen/trieben/2662499/htl-absolventen-sind-sehr-gefragt.story>, visited: Septmeber 2011.
- [22] OÖ Nachrichten. „HTL-Absolventen trotz Krise gefragt“, April 2011. [Online]. Available: <http://www.nachrichten.at/oberoesterreich/muehlviertel/art69,358581,B>, visited: Septmeber 2011.

- [23] A. Schneeberger and A. Petanovitsch. HTL und Qualifikationsbedarf der Wirtschaft - Analysen zur Arbeitsmarktlage und europäischer Vergleiche, März 2009. [Online]. Available: http://www.ibw.at/de/pruefungsunterlagen?page=shop.getfile&file_id=306&product_id=288, visited: Septmeber 2011.
- [24] UNESCO. International Standard Classification of Education, 1997. [Online]. Available: http://www.unesco.org/education/information/nfsunesco/doc/isced_1997.htm, visited: Septmeber 2011.
- [25] A. Schneeberger. Internationale Einstufung der österreichischen Berufsbildung - Adäquate ISCED-Positionierung als bildungspolitische Herausforderung, Mai 2010, ISBN 978-3-902742-21-6. [Online]. Available: http://www.ibw.at/de/pruefungsunterlagen?page=shop.getfile&file_id=404&product_id=365, visited: Septmeber 2011.
- [26] Europäische Kommission. Der Europäische Qualifikationsrahmen für lebenslanges Lernen, 2008, ISBN 9789279084720. [Online]. Available: http://ec.europa.eu/education/policies/educ/eqf/eqf08_de.pdf, visited: Septmeber 2011.
- [27] „Website der Koordinierungsstelle für den NQR in Österreich“. [Online]. Available: http://www.lebenslanges-lernen.at/home/nationalagentur_lebenslanges_lernen/nqr_koordinierungsstelle/, visited: Septmeber 2011.
- [28] U. Fritz. „Bildungsstandards in der Berufsbildung - Projekthandbuch“, Herausgeber: BMUKK, Mai 2011. [Online]. Available: http://www.bildungsstandards.berufsbildendesschulen.at/fileadmin/content/bbs/Handbuch_BIST_Mai2011.pdf, visited: Septmeber 2011.
- [29] S. Niemeyer. „Höhere technische Lehranstalt für Informationstechnologie - Fachrichtungsstandard“ Herausgeber: BMUKK, September 2011. [Online]. Available: http://www.bildungsstandards.berufsbildendesschulen.at/fileadmin/content/bbs/AGBroschueren/Fachrichtungsstandard_IT_08-09-2011.pdf, visited: Septmeber 2011.
- [30] J. Jaklin. IMST-Newsletter: „Bildungsstandards und Kompetenzorientierung in der Berufsbildung“, Herausgeber: Institut für Unterrichts- und Schulentwicklung, Universität Klagenfurt, 2011. [Online]. Available: https://www.imst.ac.at/app/webroot/files/ueber_imst/oeffentlichkeitsarbeit/imst_newsletter35.pdf, visited: Septmeber 2011.
- [31] F. E. Weinert. Concept of competence: A conceptual clarification. In *Defining and selecting key competencies* (pp. 45-65). Göttingen: Hofgreffe, 2001.
- [32] L. W. Anderson, D. R. Krathwohl, and P. W. Airasian. *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. Allyn & Bacon, 2 edition, 2000, ISBN 978-0801319037.

- [33] G. Hager. „Höhere technische Lehranstalt für Informationstechnologie - Bildungsstandards in der Berufsbildung“ Herausgeber: BMUKK, Jänner 2009. [Online]. Available: http://www.berufsbildendeschulen.at/fileadmin/content/bbs/AGBroschueren/IT_V8.pdf, visited: Septmeber 2011.
- [34] T. Reinbacher. Lehre von Software-Verifikation in der Berufsbildung unter dem Aspekt von Bildungsstandards. Master's thesis, Technische Universität Wien, 2010.
- [35] U. Fritz. „Kompetenzorientiertes Unterrichten - Ggrundlagenpapier“ Herausgeber: BMUKK, März 2011. [Online]. Available: http://www.berufsbildendeschulen.at/fileadmin/content/bbs/KU/Grundlagenpapier_KU_Maerz2011.pdf, visited: Septmeber 2011.
- [36] Kersten Reich. *Konstruktivistische Didaktik: Lehr- und Studienbuch mit Methodenpool*. Beltz Verlag, 3 edition, 2006, ISBN 978-3407254108.
- [37] M. Schwabe, R. Radinger, and G. Sommer-Binder. Bildung in Zahlen - Schlüsselindikatoren und Analysen, 2011, ISBN 978-3-902703-77-4. [Online]. Available: http://www.statistik.at/web_de/Redirect/index.htm?dDocName=055469, visited: Septmeber 2011.
- [38] H. Gudjons. *Frontalunterricht - neu entdeckt: Integration in offene Unterrichtsformen*. Utb Verlag, 2 edition, 2007, ISBN 978-3825229481.
- [39] K. Reich. Methodenpool, 2007. [Online]. Available: <http://www.uni-koeln.de/hf/konstrukt/didaktik/index.html>, visited: Septmeber 2011.
- [40] H. Barrows. *Problem-Based Learning Applied to Medical Education*. Southern Illinois University, 2007, ISBN 978-0931369346. Rev. Edition von 1994.
- [41] B. Dutch. Writing Problems for Deeper Understanding. pages 47–53, 2001.
- [42] K. Reich. Methodenpool - Problem Based Learning, 2007. [Online]. Available: http://methodenpool.uni-koeln.de/problembased/frameset_vorlage.html, visited: Septmeber 2011.
- [43] J. H. C. Moust, P. A. J. Bouhuijs, and H. G. Schmidt. *Problemorientiertes Lernen*. Ullstein Medical Verlag, 1999.
- [44] A. Weber. *Problem-Based Learning - Ein Handbuch für die Ausbildung auf der Sekundarstufe II und der Tertiärstufe*. hep verlag, 2 edition, 2007, ISBN 978-3-03905-381-0.
- [45] C. Hmelo-Silver. Problem-based learning: What and how do students learn? *Educational Psychology Review*, 16:235–266, 2004.
- [46] M. Hamilton, J. Harland, and L. Padgham. Experiences in teaching computing theory via aspects of problem-based learning. In *Proceedings of the fifth Australasian conference on Computing education - Volume 20*, ACE '03, pages 207–211, Darlinghurst, Australia, Australia, 2003. Australian Computer Society, Inc.

- [47] J. O’Kelly and J. P. Gibson. Robocode & problem-based learning: a non-prescriptive approach to teaching programming. *SIGCSE Bull*, 38:217–221, June 2006.
- [48] G. R. Norman and H. G. Schmidt. The psychological basis of problem-based learning: A review of the evidence. *Academic Medicine*, 67:557–565, 1992.
- [49] K. Reich. Methodenpool - projektarbeit, 2007. [Online]. Available: http://methodenpool.uni-koeln.de/projekt/frameset_projekt.html, visited: Septmeber 2011.
- [50] K. Frey. *Die Projektmethode: Der Weg zum bildenden Tun*. Beltz, 10 edition, 2005, ISBN 978-3407253941.
- [51] U. Schröder. ETH-Leitprogramme - Anfangsunterricht, 2006. [Online]. Available: http://lufgi9.informatik.rwth-aachen.de/lehre/ss06/fdi/folien/06_Umethoden3.pdf, visited: Septmeber 2011.
- [52] K. Reich. Methodenpool - leittext, 2007. [Online]. Available: http://methodenpool.uni-koeln.de/leittext/frameset_leittext.html, visited: Septmeber 2011.
- [53] M. Sauter. Design eines programmierbaren Rechners, Mai 2006. [Online]. Available: http://www.educ.ethz.ch/unt/um/inf/rec/design_rechner/Sauter_Rechnerdesign_mit_Deckblatt.pdf, visited: Septmeber 2011.
- [54] M. Sauter. Design eines programmierbaren Rechners, Mai 2006. [Online]. Available: http://www.educ.ethz.ch/unt/um/inf/rec/design_rechner/Sauter_Rechnerdesign_mit_Deckblatt.pdf, visited: Septmeber 2011.
- [55] C. Diehl and M. Leupp. Komplexe Zahlen - Ein Leitprogramm in Mathematik, Jänner 2010. [Online]. Available: <http://www.educ.ethz.ch/unt/um/mathe/aa/kz/Leitprogramm.pdf>, visited: Septmeber 2011.
- [56] Leistungsbeurteilungsverordnung (LBVO). In der Fassung der Bekanntmachung vom 17.08.2011, August 2011, veröffentlicht in: BGBl. Nr. 371/1974. [Online]. Available: <http://www.ris.bka.gv.at/GeltendeFassung/Bundesnormen/10009375/Leistungsbeurteilungsverordnung%2c%20Fassung%20vom%2017.08.2011.pdf>, visited: Septmeber 2011.
- [57] Amrhein-Kreml, R. and Bartosch, I. and Breyer, G. and Dobler, K. and Koenne, C. and Mayr, J. and Schuster, A. *Prüfungskultur. Leistung und Bewertung (in) der Schule*. Alpen-Adria-Universität Klagenfurt, 1 edition, 2008, ISBN 978-3-200-01285-1.
- [58] E.-M. Schumacher. Beispiel für einen Lernkontrakt/-vertrag, Juni 2006. [Online]. Available: http://www.lehridee.de/data/doc/id_315/Lernkontrakt.pdf, visited: Septmeber 2011.
- [59] K. Reich. Methodenpool - Tagebuch, 2007. [Online]. Available: http://methodenpool.uni-koeln.de/tagebuch/frameset_tagebuch.html, visited: Septmeber 2011.

- [60] BMUKK. Verordnung der Bundesministerin für Unterricht, Kunst und Kultur über die Lehrpläne für Höhere technische und gewerbliche Lehranstalten; Bekanntmachung der Lehrpläne für den Religionsunterricht, September 2011. [Online]. Available: http://www.bmukk.gv.at/schulen/recht/erk/lp_htl_rel.xml, visited: Jänner 2012.
- [61] Lehrplan der höheren Lehranstalt für Informationstechnologie - Anlage 1.5, September 2011. [Online]. Available: http://www.htl.at/fileadmin/content/Lehrplan/HTL_VO_2011/BGB1_II_Nr_300_2011_Anlage_1_5.pdf, visited: Jänner 2012.
- [62] Lehrplan der höheren Lehranstalt für Informationstechnologie - Anlage 1.1.9, September 2011. [Online]. Available: http://www.htl.at/fileadmin/content/Lehrplan/HTL_Informationstechnologie.pdf, visited: Jänner 2012.
- [63] Lehrplan der höheren Lehranstalt für Informationstechnologie - Anlage 1, September 2011. [Online]. Available: http://www.htl.at/fileadmin/content/Lehrplan/HTL_VO_2011/BGB1_II_Nr_300_2011_Anlage_1.pdf, visited: Jänner 2012.
- [64] Messner, R. and Blum, W. *Selbstständiges Lernen im Fachunterricht*. kassel university press GmbH, 1 edition, 2006, ISBN 978-3-89958-193-5.
- [65] Grechenig, T. and Bernhart, M. and Breiteneder, R. and Kappl, K. *Softwaretechnik - Mit Fallbeispielen aus realen Entwicklungsprojekten*. Pearson Studium Verlag, 1 edition, Oktober 2009, ISBN 978-3-86894-007-7.
- [66] ACM. „Website Advancing Computing as a Science and Profession - Curricula Recommendations“, 2012. [Online]. Available: <http://www.acm.org/education/curricula-recommendations>, visited: Septmeber 2011.
- [67] Soldan, D. and Hughes, J. and Impagliazzo, J. and McGettrick, A. and Nelson, V.P. and Srimani, P.K. and Theys, M.D. Computer Engineering 2004 - Curriculum Guidelines for Undergraduate Degree Programs in Computer Engineering, December 2004. [Online]. Available: http://www.acm.org/education/education/curric_vols/CE-Final-Report.pdf, visited: Septmeber 2011.
- [68] Cardoso, J.M.P. New challenges in computer science education. *SIGCSE Bull.*, 37:203–207, June 2005.
- [69] M. Grimheden and M. Törngren. How should embedded systems be taught? Experiences and snapshots from Swedish higher engineering education. *SIGBED Rev.*, 2:34–39, October 2005.
- [70] J.K. Muppala. Bringing embedded software closer to computer science students. *SIGBED Rev.*, 4:11–16, January 2007.
- [71] A. Stollenwerk, C. Jongdee, and S. Kowalewski. An undergraduate embedded software laboratory for the masses. In *Proceedings of the 2009 Workshop on Embedded Systems Education*, WESS '09, pages 34–41, New York, NY, USA, 2009. ACM.

- [72] J.-M. Vanhatupa, A. Salminen, and H.-M. Järvinen. Organizing and evaluating course on embedded programming. In *Proceedings of the 10th Koli Calling International Conference on Computing Education Research*, Koli Calling '10, pages 112–117, New York, NY, USA, 2010. ACM.
- [73] Atmel. „Website Atmel - AVR Studio 4“. [Online]. Available: <http://www.atmel.com/tools/AVRSTUDIO4.aspx>, visited: Jänner 2012.
- [74] WinAVR. „Website WinAVR - GNU GCC compiler for C and C++“. [Online]. Available: <http://winavr.sourceforge.net/>, visited: Jänner 2012.
- [75] HAPSIM. „Website HAPSIM - Helmi's AVR Periphery Simulator V2.19“. [Online]. Available: <http://www.helmix.at/hapsim/>, visited: Jänner 2012.
- [76] B. Weiss, G. Gridling, and M. Proske. A case study in efficient microcontroller education. *SIGBED Rev.*, 2:40–47, October 2005.
- [77] A.N. Kumar. The effect of closed labs in computer science I: an assessment. *J. Comput. Small Coll.*, 18:40–48, May 2003.
- [78] M. Apiola, M. Lattu, and T.A. Pasanen. Creativity and intrinsic motivation in computer science education: experimenting with robots. In *Proceedings of the fifteenth annual conference on Innovation and technology in computer science education*, ITiCSE '10, pages 199–203, New York, NY, USA, 2010. ACM.
- [79] Harrison, O. *Open Space Technology: Ein Leitfaden für die Praxis*. Schäffer-Poeschel Verlag, 2 edition, 2011, ISBN 978-3791031347.
- [80] Lego Mindstorms. „Website Lego Mindstorms“, 2012. [Online]. Available: <http://mindstorms.lego.com/en-us/Default.aspx>, visited: Septmeber 2011.
- [81] S. Pollard and R.C. Duvall. Everything I needed to know about teaching I learned in kindergarten: bringing elementary education techniques to undergraduate computer science classes. In *Proceedings of the 37th SIGCSE technical symposium on Computer science education*, SIGCSE '06, pages 224–228, New York, NY, USA, 2006. ACM.
- [82] R. Romeike. *Kreativität im Informatikunterricht*. PhD thesis, Universität Potsdam - Mathematisch-Naturwissenschaftlichen Fakultät, 2008. [Online]. Available: <http://www.informatikdidaktik.de/Forschung/Schriften/RomeikeDiss2008.pdf>, visited: Septmeber 2011.
- [83] Mueller, H. and Walz, L. *Mikroprozessortechnik*. Vogel Verlag, 6 edition, 2002, ISBN 3-8023-1895-1.
- [84] Herold, H. and Lurz, B. and Wohlrab, J. *Grundlagen der Informatik*. Pearson Verlag, 1 edition, 2007, ISBN 978-3-82737305-2.
- [85] Atmel. „Website Atmel - Datenblatt ATmega328P Mikrocontroller“. [Online]. Available: http://atmel.com/dyn/resources/prod_documents/8271S.pdf, visited: Jänner 2012.

- [86] Brinkschulte, U. and Ungerer, T. *Mikrocontroller und Mikroprozessoren*. Springer Verlag, 3 edition, 2010, ISBN 978-3-642-05397-9.
- [87] Arduino. „Website Arduino - Arduino Ethernet“. [Online]. Available: <http://arduino.cc/en/Main/ArduinoBoardEthernet>, visited: Jänner 2012.
- [88] Arduino. „Website Arduino - Arduino Software“. [Online]. Available: <http://arduino.cc/en/Main/Software>, visited: Jänner 2012.
- [89] Arduino. „Website Arduino - Schaltplan des Arduino Ethernet Kits“. [Online]. Available: <http://arduino.cc/en/uploads/Main/arduino-ethernet-schematic.pdf>, visited: Jänner 2012.
- [90] Schmitt, G. *Mikrocomputertechnik mit Controllern der Atmel AVR-RISC-Familie*. Oldenbourg Verlag, 4 edition, 2007, ISBN 978-3-486-58790-6.
- [91] Hitachi. „Website LCD-Linux - Datenblatt LCD Hitachi HD44780U“. [Online]. Available: <http://lcd-linux.sourceforge.net/pdfdocs/hd44780.pdf>, visited: Jänner 2012.
- [92] Barnett, R. and O’Cull, L. and Cox, Sarah. *Embedded C Programming and the Atmel AVR*. Delmar Cengage Learning, 2 edition, 2007, ISBN 978-1-4180-3959-2.
- [93] AVRDUDE. „Website AVRDUDE - AVR Downloader“. [Online]. Available: <http://www.nongnu.org/avrdude>, visited: Jänner 2012.
- [94] Arduino. „Website Arduino - USB Driver Plugin“. [Online]. Available: http://arduino.cc/en/uploads/Main/Arduino_USBSerial.zip, visited: Jänner 2012.
- [95] Arduino. „Website Arduino - USB Driver Installationsanleitung“. [Online]. Available: <http://arduino.cc/en/Guide/UnoDriversWindowsXP>, visited: Jänner 2012.