

Technische Universität Berlin
Fakultät V – Verkehrs- und Maschinensysteme
Institut für Konstruktion, Mikro- und Medizintechnik
Fachgebiet Methoden der Produktentwicklung und Mechatronik

Bachelorarbeit

Vergleichende Analyse und Bewertung von eingebetteten Systemen zur Untersuchung eines Brennstoffzellenhybridfahrzeugmodells in der universitären Lehre

zur Erlangung des akademischen Grades
Bachelor of Science
(B. Sc.)

eingereicht von: Leonard Freisem 369 244

eingereicht am: 15. Oktober 2020

Betreuer: Prof. Dr.-Ing. Dietmar Göhlich
Dr.-Ing. Tu-Anh Fay

Aufgabenstellung



Fachgebiet Methoden der Produktentwicklung und Mechatronik

Prof. Dr.-Ing. D. Göhlich



Institut für Maschinenkonstruktion und Systemtechnik
Fakultät V - Verkehrs- und Maschinensysteme
Technische Universität Berlin

Bachelorarbeit für Leonard Freisem

„Vergleichende Analyse und Bewertung von eingebetteten Systemen zur Untersuchung eines Brennstoffzellenhybridfahrzeugmodells in der universitären Lehre“

Aufgrund der Mobilitätswende und den angestrebten Klimazielen nimmt Elektromobilität einen größeren Stellenwert ein. Die Brennstoffzelle spielt neben der Batterie eine große Rolle bei der Umstellung auf alternative Antriebe. In der Lehre, im Modul Electric vehicle technologies and applications, steht den Studierenden der „Heliocentris - HyDrive – Electric Vehicle Trainer“ (i. F. HyDrive) zur Verfügung. Der HyDrive soll mittels hardwarenahen Experimenten Schüler und Studierende an die Eigenschaften der Wasserstoffbrennstoffzelle (PEMFC) führen. Damit können in einem realen Umfeld erste Versuche mit Brennstoffzellen gemacht werden. Allerdings weist der Versuchsaufbau einige Schwächen auf und es lässt sich nicht eigenständig anpassen oder erweitern.

Eingebettete Systeme spielen bei der Entwicklung mechatronischer Systeme eine wichtige Rolle und praktische Erfahrungen sind eine wichtige Grundlage im Ingenieurswesen. Das V-Modell bildet häufig die Grundlage der industriellen Entwicklung, die Lehre vermittelt unterdessen viele theoretische Aspekte.

Im Rahmen der Abschlussarbeit soll untersucht werden wie sich der HyDrive sinnvoll ergänzen lässt, damit Studierende die Entwicklung nach dem V-Modell erlernen und mit dem HyDrive praktisch umsetzen können.

Es ergibt sich folgende Vorgehensweise:

- Literaturrecherche
 - Methodenorientierte Produktentwicklung
 - Lehre im Ingenieurswesen
- Anforderungsanalyse im Hinblick auf das System Lehre
- Vergleich verschiedener eingebetteter Systeme zur Untersuchung eines vorhandenen Prüfstandes mit Brennstoffzellenhybridfahrzeugmodell
 - Ableiten von Bewertungskriterien aus der Anforderungsanalyse
 - Recherche zu geeigneten Lösungskonzepten für eingebettete Systeme
 - Bewertung der recherchierten Lösungskonzepte
- Umsetzung des besten Konzeptes und Umbau des vorhandenen Prüfstandes
- Bewertung des neuen Prüfstandes mit gleichen Kriterien in Hinblick auf das System Lehre

Betreuerin: Dr.-Ing. Tu-Anh Fay

Berlin, den 14.07.2020

Prof. Dr.-Ing. Dietmar Göhlich
Leiter des Fachgebiets

Auslage

„Hiermit erkläre ich mich damit einverstanden, dass meine Studien-, Bachelor- bzw. Masterarbeit in der Instituts- und Universitätsbibliothek ausgelegt und zur Einsichtnahme aufbewahrt werden darf.“

„Hiermit erkläre ich, dass ich nicht mit der Auslage und Einsichtnahme meiner Studien-, Bachelor- bzw. Masterarbeit in der Instituts- und/oder der Universitätsbibliothek einverstanden bin.“

Ich habe die Auswahl zur Auslage meiner Studien-, Bachelor- bzw. Masterarbeit in der Instituts- und/o-
der der Universitätsbibliothek zur Kenntnis genommen:

Berlin, den 15. Oktober 2020



.....
Leonard Freisem 369 244

Eidesstattliche Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig und eigenhändig sowie ohne unerlaubte Hilfe und ausschließlich unter Verwendung der aufgeführten Quellen und Hilfsmittel angefertigt habe.
Die selbstständige und eigenständige Anfertigung versichert an Eides statt:

Berlin, den 15. Oktober 2020



.....

Leonard Freisem 369 244

Kurzfassung

Mit der Energiewende und der Elektrifizierung des Antriebsstrangs steigt auch die Bedeutung der Brennstoffzelle im Automobilbereich. Mit zunehmender Komplexität einzelner Komponenten im Automobil, werden immer mehr eingebettete Systeme zur Steuerung der Komponenten benötigt. Die Überschneidung der Fachrichtungen Maschinenbau, Elektrotechnik und Informatik während des Entwicklungsprozesses müssen durch die Lehre übermittelt werden. Zu diesem Zweck wird der HyDrive – Electric Vehicle Trainer, kurz HyDrive, von der Firma Heliocentris angeboten. Darin enthalten ist ein Brennstoffzellenhybridfahrzeug, ein Rollenprüfstand und eine Elektrolysestation. Die Komponenten sind jeweils in Modellgröße ausgeführt. Der HyDrive wird mittels eingebettetem System gesteuert und vermittelt Grundlagen hinsichtlich der Brennstoffzellentechnik, elektrischer Maschinen und der Versuchsmethodik. Nachteilig sind die fehlenden Möglichkeiten der Programmierung und einer Systementwicklung. Zur Optimierung des HyDrive hinsichtlich der Lehre analysiert und bewertet diese Arbeit unterschiedliche eingebettete Systeme mit Rapid-Prototyping Eigenschaften. Für die Bewertung der vier Systeme HyDrive, Arduino Mega 2560, dSpace MicroAutoBox II und ETAS EDUkit (Prototyp) wird die VDI 2225 angewendet. Für die Bewertung werden zunächst Kriterien wie Kompetenzvermittlung und Handhabung erörtert. Grundlage dafür sind unter Anderem die heutigen Anforderungen an die Lehre des Ingenieurwesens. Bewertet wurde durch eine Gruppe von drei Lehrenden des Fachgebietes MPM der TU-Berlin, sowie durch eine Gruppe von vier Studierenden der Fachrichtungen Elektrotechnik, Maschinenbau, Informatik im Maschinenwesen und Automotive Systems. Die Bewertung hat gezeigt, dass das ETAS EDUkit entsprechend der verwendeten Kriterien am besten für die Optimierung des HyDrive geeignet ist. Daraufhin folgte der Umbau des HyDrive auf ein eingebettetes System mit Rapid-Prototyping-Eigenschaften (RP-HyDrive). Ungeachtet dessen, dass der RP-HyDrive nicht fertig gestellt werden konnte, können im aktuellen Zustand wesentliche Aspekte der Brennstoffzellentechnologie, Systementwicklung und der Umgang mit Rapid-Prototyping-Systemen vermittelt werden.

Abstract

With the energy revolution and the electrification of the drive train the importance of the fuel cell in the automotive sector is also increasing. With the increasing complexity of individual components in the automobile more and more embedded systems are needed to control the components. The overlapping of the different fields of mechanical engineering, electrical engineering and computer science during the development process must be conveyed through teaching. For this purpose the HyDrive - Electric Vehicle Trainer (HyDrive) is offered by the company Heliocentris. It includes a fuel cell hybrid vehicle, a roller test bench and an electrolysis station. The components are each in model size. The HyDrive is controlled by an embedded system and teaches the basics of fuel cell technology, electrical machines and experimental methodology. The HyDrive is not programmable and offers no possibilities for system development which are major tasks in today's engineering. To optimise the HyDrive in terms of teaching, this thesis analyses and evaluates different embedded systems with rapid prototyping properties. VDI 2225 is applied for the evaluation of the four systems HyDrive, Arduino Mega 2560, dSpace MicroAutoBox II and ETAS EDUkit (prototype). For the evaluation, criteria such as competence transfer and handling are first discussed. The basis for this is, among other things, the current demands on the teaching of engineering. The evaluation was carried out by a group of three teachers from the MPM department of the TU Berlin, as well as a group of four students from the fields of electrical engineering, mechanical engineering, computer science in mechanical engineering and automotive systems. The evaluation showed that the ETAS EDUkit is best suited for optimizing the HyDrive according to the criteria used. This was followed by the conversion of the HyDrive to an embedded system with rapid prototyping features (RP-HyDrive). Regardless of the fact that the RP-HyDrive could not be completed, in its current state essential aspects of fuel cell technology, system development and the handling of rapid prototyping systems can be taught.

Inhaltsverzeichnis

Abkürzungsverzeichnis	iv
Abbildungsverzeichnis	v
Tabellenverzeichnis	vi
1 Einleitung	1
1.1 Motivation der Arbeit	1
1.2 Vorgehensweise in der Arbeit	1
2 Grundlagen	2
2.1 Anforderungsliste	2
2.2 Nutzwertanalyse nach VDI 2225	3
2.3 Entwicklung mechatronischer Systeme mit dem V-Modell	4
2.4 Embedded Systems / Rapid Prototyping	6
2.5 Brennstoffzellen im Automobil	6
2.6 Konzept Heliocentris HyDrive	7
3 Methodik	11
3.1 Anforderungsanalyse	11
3.1.1 Anforderungen an die Hochschullehre	11
3.1.2 Anforderungen gegeben durch den HyDrive	12
3.2 Auswahl der Konzepte	14
3.2.1 Konzept Arduino	15
3.2.2 Konzept dSpace MicroAutoBox II	17
3.2.3 Konzept ETAS EDUkit	19
3.3 Konzeptvergleich anhand der Steuerung einer LED	22
3.4 Wahl der Bewertungskriterien für die Nutzwertanalyse	28
3.5 Bewertung nach VDI 2225	30
3.5.1 Bewertung durch die Lehre	34
3.5.2 Bewertung durch Studierende	37
3.5.3 Auswertung der Bewertung	39
4 Durchführung und Umsetzung	41
4.1 Bestandsaufnahme	41
4.2 Schaltpläne	42
4.2.1 Fahrzeug	42
4.2.2 Tankstelle	44
4.2.3 Prüfstand	45
4.3 Zusätzlich benötigte Hardware zum Messen und Verstellen	45
4.3.1 Vorgefertigte Module	45
4.3.2 ETAS-Hardware	47

4.4 Programmierung	47
4.4.1 Fahrzeug	47
4.4.2 Tankstelle	51
4.4.3 Prüfstand	53
5 Auswertung	56
5.1 Bewertung des RP-HyDrive	56
5.2 Herausforderungen während der Arbeit	56
5.3 Kritische Betrachtung der Arbeit	57
5.4 Ausblick	59
Literatur	60
A Anhang	63
A.1 Anforderungsliste	63

Abkürzungsverzeichnis

BEV	batteriebetriebenes Fahrzeug
DC-DC	Spannungswandler für Gleichstrom
DoE	Design of Experiment
EE	Experiment Environment
ESDL	Programmiersprache - Embedded Systems Description Language
EVTech	Lehrveranstaltung Electric Vehicles Technologies and Applications
FCEV	mit Brennstoffzellen betriebenes Fahrzeug
HyDrive	HyDrive – Electric Vehicle Trainer der Firma Heliocentris Academia
IDE	integrated development environment
MAB II	MicroAutoBox II der Firma dSpace
NC	Zustand eines Relais: Normally Closed (=Öffner)
NO	Zustand eines Relais: Normally Open (=Schließer)
PWM	Pulsweitenmodulation
RP	Rapid Prototyping
RP-HyDrive	HyDrive auf Basis von eingebetteten Systemen mit RP-Eigenschaften
RTI	Real-Time Interface
V-ECU	virtuelles Steuergerät
ZIF	Zero Insertion Force

Abbildungsverzeichnis

2.1	V-Modell als Makrozyklus nach VDI 2206 [5, S.29]	5
2.2	Schematische Darstellung eines Brennstoffzellenhybridfahrzeugs [11, S.178]	7
2.3	HyDrive - Electric Vehicle Trainer [12]	8
2.4	Brennstoffzellen-stack [12]	8
2.5	Wasserstofftank [12]	8
2.6	Wasserabscheider [12]	8
2.7	Rollenprüfstand [12]	9
2.8	Tankstelle [12]	9
2.9	Grafische Oberfläche des HyDrive	10
3.1	Schaltplan des HyDrive Fahrzeugs [14, S.173]	13
3.2	Arduino Mega 2560 [15]	15
3.3	dSpace Produkte gegliedert am V-Modell [19, S.18, 19]	17
3.4	dSpace MicroAutoBox II Vorderseite [19, S.20]	17
3.5	dSpace MicroAutoBox II Rückseite [20]	17
3.6	MicroAutoBox Break-Out Box [22]	18
3.7	Das ETAS EDUkit: Hardware und Software in einem Paket	19
3.8	Draufsicht der Platine des EDUkit	20
3.9	Die Softwarekomponenten im V-Modell	21
3.10	Eingangssignal ohne Pull-Up/Pull-Down-Widerstand [28]	23
3.11	LED mittels Taster und Arduino steuern [27]	23
3.12	Programmcode der MAB II für die Steuerung einer LED	25
3.13	Verwendete Bauteile des EDUkits zur Steuerung einer LED	26
3.14	Programmcode des EDUkits für die Steuerung einer LED	27
3.15	Properties Fenster eines Literals	27
3.16	Grafische Darstellung der Bewertungsdaten von Lehrpersonal und Studierenden	40
4.1	Pinbelegung des EDUkit	42
4.2	Schaltplan des Fahrzeugs	43
4.3	Schaltplan der Elektrolyse	44
4.4	Schaltplan des Prüfstands	45
4.5	Programmierung des RP-HyDrive	47
4.6	Programmierung des RP-HyDrive - Fahrzeug	49
4.7	Programmierung des RP-HyDrive Zustand der Brennstoffzelle („FuelCellMode“)	50
4.8	Programmierung des RP-HyDrive Schalten eines Relais („FuelCell-Control“)	51
4.9	Programmierung des RP-HyDrive - Tankstelle	52
4.10	Programmierung des RP-HyDrive - Umrechnung zum Einlesen eines Analogeinganges .	53
4.11	Programmierung des RP-HyDrive - Prüfstand	54
4.12	Programmierung des RP-HyDrive - Umrechnung Prozentwert in PWM-Arbeitszyklus .	55
5.1	Grafische Darstellung der Bewertungsdaten	58

Tabellenverzeichnis

2.1	Beispiel einer Anforderungsliste	3
2.2	Beispiel einer Punktbewertungsskala nach VDI 2225 Blatt 3	3
2.3	Beispiel eines Formblattes zur Bewertung nach VDI 2225 Blatt 3	4
3.1	Geforderte Schnittstellen an die RP-Hardware	14
3.2	Zusammenfassung der Konzepte	15
3.3	Stromstärke I und Leistung P der Verschiedenen USB Typen	16
3.4	Arduino Mega 2560 - Ein- und Ausgänge	16
3.5	dSpace MicroAutoBox II - Ein- und Ausgänge	18
3.6	ETAS EDUkit - Ein- und Ausgänge	21
3.7	Konzeptvergleich LED - MAB II Pinbelegung	26
3.8	Konzeptvergleich LED - verwendete Bauteile	28
3.9	Vorlage Gewichtungsmatrix	30
3.10	Urteilsschema	33
3.11	Gewichtungsmatrix der Lehre	35
3.12	Bewertungsblatt der Lehre	36
3.13	Gewichtungsmatrix der Studierenden	37
3.14	Bewertungsblatt der Studierenden	38
4.1	Technischen Daten der Komponenten des HyDrive	41

1 Einleitung

1.1 Motivation der Arbeit

Ein Aufgabengebiet der Lehre ist es Studierende an neue Fachinhalte und damit zusammenhängende Methoden heranzuführen. Um die Inhalte und Methoden zu vermitteln bieten sich verschiedene Möglichkeiten, die Lehrveranstaltung mit praktischen Elementen zu ergänzen. [1, S.2] Der HyDrive – Electric Vehicle Trainer (HyDrive) bietet Schüler:innen und Studierenden erste Einblicke in die Brennstoffzellen-hybridtechnologie, damit zusammenhängende Komponenten und grundlegende methodische Elemente der Bearbeitung von Experimenten. Der HyDrive wird als Ergänzung der Lehrinhalte in der Lehrveranstaltung „Electric Vehicles Technologies and Applications“ (EVTech) angeboten.

Während der Arbeit mit dem HyDrive wurde Optimierungspotential hinsichtlich der verwendeten Hardware festgestellt, da z.B. die Verbindung zum Aufbau über eine Bluetooth-Verbindung hergestellt wird. Der Verbindungsauflbau führt bereits vor Beginn der eigentlichen Arbeit zu Problemen und soll im Rahmen eines Umbaus auf ein eingebettetes System mit Rapid Prototyping (RP) -Eigenschaften optimiert werden. Ein HyDrive auf Basis eines eingebetteten RP-Systems (RP-HyDrive) würde weitere Entwicklungsmöglichkeiten im Rahmen der Lehrveranstaltung EVTec oder einer eigenen auf den RP-HyDrive ausgerichteten Lehrveranstaltung schaffen. Denkbar wäre den Studierenden die Aufgabe zu vermitteln, den RP-HyDrive kritisch zu betrachten und eine Funktion weiter zu entwickeln.

1.2 Vorgehensweise in der Arbeit

Im Folgenden wird die Vorgehensweise zur Optimierung des HyDrive erläutert. In Kapitel 2 werden zunächst methodische Grundlagen, die technische Bedeutung von Brennstoffzellen und der Aufbau des vorhandenen Prüfstands erläutert.

Kapitel 3 behandelt die Bewertung verschiedener Konzepte auf Basis eingebetteter Systeme mit RP-Eigenschaften zur Optimierung des Aufbaus. Dabei wird mit der Anforderungsanalyse begonnen, unterschiedliche Konzepte zur Optimierung aufgestellt und nach VDI 2225 bewertet und diskutiert.

Das daraus resultierende Konzept wird im Kapitel 4 praktisch umgesetzt. Dazu werden die Schritte von der Bestandsaufnahme des vorhandenen Prüfstandes und des verwendeten Konzeptes, über Schaltpläne, Programmierung und praktische Umsetzung erläutert.

Kapitel 5 setzt sich abschließend kritisch mit dem Umbau und dessen Eignung für die universitäre Lehre auseinander. Die in Kapitel 3 verwendeten Bewertungskriterien werden für die Evaluierung und zur Erörterung von weiterem Optimierungspotenzial verwendet.

2 Grundlagen

Für die vorliegende Arbeit werden im theoretischen Teil grundlegende Methoden der Produktentwicklung angewandt. Ziel ist die Bewertung unterschiedlicher Konzepte eingebetteter Systeme (s. Kapitel 2.4) nach VDI 2225 (s. Kapitel 2.2) hinsichtlich der Anwendbarkeit in der universitären Lehre. Für die Bewertung der Konzepte wird zunächst eine Anforderungsliste (s. Kapitel 2.1) benötigt. Im Anschluss an die Bewertung wird die theoretisch beste Lösung umgesetzt. Für ein besseres Verständnis werden im Folgenden wesentliche Vorgehensweisen erläutert. Als Grundlage dieser Arbeit dient der HyDrive - ein Brennstoffzellenhybridfahrzeug in Modellgröße. Dafür wird die Bedeutung der Brennstoffzelle im Automobil in Kapitel 2.5 angeschnitten, woraufhin der Aufbau des HyDrive in Kapitel 2.6 erläutert wird. Für den Umbau werden Konzepte eingebetteter Systeme mit RP-Eigenschaften verwendet. In Kapitel 2.4 werden RP-Systeme näher erklärt. Das V-Modell wird häufig für die industrielle Entwicklung mechatronischer Systeme wie den HyDrive verwendet. Für ein besseres Verständnis wird in Kapitel 2.3 der Entwicklungsprozess entlang des V-Modells erläutert.

2.1 Anforderungsliste

Das deutsche Institut für Normung e.V. (DIN) beschreibt eine Anforderung als:

"Beschaffenheit, Fähigkeit oder Leistung, die ein Produkt, Prozess oder die am Prozess beteiligte Person erfüllen oder besitzen muss, um einen Vertrag, eine Norm, eine Spezifikation oder andere, formell vorgegebene Dokumente zu erfüllen". **DIN 69901-5 - 3.7 [2, S.6]**

Das ist die Grundlage für die Anforderungsliste. Die Anforderungsliste ist ein Hilfsmittel bei der Entwicklung und Konstruktion. Das (Teil-)Projekt wird tabellarisch in Anforderungen für Baugruppen und Bauteile aufgeteilt. Die Anforderungen sind präzise und frei von Auslegungsspielräumen, sodass jeder verantwortliche Entwickler die Anforderungen genauestens in die Entwicklung mit einbringen kann. Jeder Anforderung wird das Attribut Forderung (F) oder Wunsch (W) hinzugefügt. Die Attribute Forderung und Wunsch sind ausschlaggebend für die Konzeptfindung und -bewertung (i.F. nach VDI-2225-3). Eine Forderung muss in jedem Fall durch das Konzept und den Entwickler erfüllt werden, ein Wunsch kann, muss aber nicht. Neben planerischen Aspekten bilden Anforderungslisten die Verhandlungsgrundlage zwischen Kunden und Auftragsnehmer. Ausgeführt wird die Anforderungsliste in Tabellenform und könnte z.B. wie in Tabelle 2.1 [3, S.43] dargestellt, aussehen.

Tabelle 2.1: Beispiel einer Anforderungsliste [3, S.43]

Anforderungsliste für Sicherheits - Ablassventil				Blatt: 1 Seite:3
Änderung	F W	Pos.	Anforderung	Verantwortliche
1.9.73		22	Ventilteller mit ebener Dichtfläche (Kein Ventikelgel)	
"		23	Keine Starre Verbindung Ventilteller - Spindel	
"	W	24	Einfache Möglichkeit Dichtflächen auszubauen oder auszutauschen	
...	
Ersetzt [Datum] Ausgabe vom [Datum]				

Neben den Attributen „Änderung“, „Anforderung“ und „Verantwortliche“ können die „Quelle“ einer Anforderung oder Begründungen, bzw. „Kommentare“ bezogen auf die Anforderung eingetragen werden.

2.2 Nutzwertanalyse nach VDI 2225

Im Produktentwicklungsprozess bieten sich gerade am Anfang eine Vielzahl an Konzepten und Lösungsmöglichkeiten. Ziel der Entwicklung ist eine möglichst lange Konkurrenzfähigkeit am Markt und ein Produkt mit großer technischer und wirtschaftlicher Reife. [4, S.2] Auf Grundlage verschiedener Kriterien und Gewichtungen können mit der Nutzwertanalyse nach VDI 2225 unterschiedliche Konzepte bewertet und ausgewählt werden.

Im ersten Schritt legt der verantwortliche Entwickler Bewertungsmerkmale fest. Diese orientieren sich an der zuvor erstellten Anforderungsliste (s. Kapitel 2.1). Die Forderungen müssen von jedem Konzept erfüllt werden und sind bei der Bewertung nicht von Bedeutung. Die festgelegten Merkmale werden auf Grundlage der Wünsche gebildet. Wenn möglich werden die Bewertungsmerkmale von Mitarbeitern aus verschiedenen Fachrichtungen festgelegt, um ein möglichst breit gefächertes Spektrum an Merkmalen in die Bewertung einfließen zu lassen. [4, S.3] Für eine erste Orientierung sollten ca. 10 bis 15 technische Eigenschaften in die Bewertung mit einfließen. Die technischen Eigenschaften sollten als Vorzüge, z.B. „Temperaturbeständigkeit“ anstelle von „Temperaturempfindlichkeit“ formuliert werden. [4, S.4]

Im zweiten Schritt wird jedem Merkmal n eine Skala von 0 bis 4 zugeteilt. Dabei können konkrete Werte, wie z.B. 70 kW als Leistung, einem Skalenwert zugeordnet werden oder die Skala wird ordinal aufgebaut. Im Konzept kann das jeweilige Merkmal als „sehr gut“, „gut“, „ausreichend“, „gerade noch tragbar“ oder als „unbefriedigend“ erfüllt werden. Tabelle 2.2 zeigt als Beispiel eine solche Punkteskala. [4, S.4]

Tabelle 2.2: Beispiel einer Punktbewertungsskala nach VDI 2225 Blatt 3 [4, S.4]

Annäherung an die ideale Verwirklichung:	
sehr gut (ideal)	p = 4 Punkte
gut	p = 3 Punkte
ausreichend	p = 2 Punkte
gerade noch tragbar	p = 1 Punkt
unbefriedigend	p = 0 Punkte

Die Konzepte werden mit der technisch idealen Lösung verglichen. Je näher die durchschnittliche Punktzahl eines Konzeptes an der 4 liegt, desto näher liegt das Konzept am technischen Ideal. [4, S.4]

Für den Fall, dass die (Einzel-)Merkmale bzw. Hauptmerkmale eine Gewichtung zugewiesen bekommen, muss dies in der Berechnung für den gewichteten Mittelwert mitberücksichtigt werden. Die Summe der Gewichtungen einer Gruppe von Einzelmerkmalen muss dabei immer 1 ergeben.

Aus Gesprächen mit der Betreuerin Tu-Anh Fay ergab sich, dass die einzelnen Gewichtungen erfahrungs-gemäß in einem Bereich von 10% bis 35 % liegen sollten. Dadurch wird der Einfluss der am geringsten gewichteten Merkmale gewährleistet. Neben unterschiedlichen Gewichtungen von einzelnen Fachbereichen kann die Gewichtung im Nachhinein für eine Sensitivitätsanalyse variiert werden. Dabei können die Auswirkungen von Randbedingungen z.B. Fertigungsbedingungen oder Marktveränderungen betrachtet und alternative Gewichtung vergeben werden.

In Anlehnung an Bild 15 in [4, S.13] zeigt Tabelle 2.3 ein Beispiel für ein Formblatt zur Bewertung eines fiktiven Produktes. In diesem Formblatt werden die Einzelmerkmale, wie auch die Hauptmerkmale unterschiedlich gewichtet. Der Quotient aus dem Bewertungsergebnis und dem Ideal, entspricht einer technischen Wertigkeit. Nach Abschnitt 1.2.4 der VDI 2225 Blatt 3 werden die technischen Wertigkeiten unterteilt in:

$> 0,8$	sehr gut
$\geq 0,7$	gut
$\geq 0,6$	befriedigend
$< 0,6$	unbefriedigend

Die in Tabelle 2.3 beispielhaft dargestellten Konzepte haben jeweils eine technische Wertigkeit von ca. 0,57 und 0,59 wodurch beide in den Bereich „unbefriedigend“ fallen. [4, S.4] Für dieses Beispiel müssen die Konzepte überarbeitet und erneut bewertet werden.

Tabelle 2.3: Beispiel eines Formblattes zur Bewertung nach VDI 2225 Blatt 3

Technische Merkmale	Gewichtung	Gewichtung Einzelmerkmal	Konzept 1	Konzept 2	Ideal
Hauptmerkmal 1	0,4				
Merkmal 1	0,6	0,24	3	2	4
Merkmal 2	0,4	0,16	2	3	4
Hauptmerkmal 2	0,6				
Merkmal 3	0,35	0,21	4	3	4
Merkmal 4	0,65	0,39	1	2	4
Summe			2,27	2,37	4

2.3 Entwicklung mechatronischer Systeme mit dem V-Modell

Mechatronische Systeme sind heutzutage aus einem Fahrzeug nicht mehr wegzudenken. Vor dem Einzug von Mikroprozessoren in die Automobilwelt wurde beispielsweise die Schaltung des Getriebes rein mechanisch betätigt. Mit der Verwendung von Mikroprozessoren in Steuergeräten lässt sich die Schaltung heutzutage über einen Knopfdruck am Lenkrad betätigen. Dabei wird ein Signal vom Taster am Lenkrad an ein Steuergerät gesendet und vom Steuergerät verarbeitet. Den eigentlichen Schaltvorgang führt das Steuergerät aus. Die Kombination aus mechanischen und elektronischen Komponenten ergeben ein mechatronisches System.

Das mechatronische System kann auf Makroebene betrachtet werden, wie auch in seinen mechanischen und elektronischen Bestandteilen auf Mikroebene. Für die Entwicklung derartiger Systeme gibt es nicht den einen Weg zur Lösungsfindung und das eine Schema zur Bearbeitung des Problems. Das V-Modell (Abb. 2.1) bietet eine Richtschnur und beschreibt das grundsätzliche Vorgehen zur Bearbeitung mechatronischer Projekte. Es findet seinen Ursprung in der Softwareentwicklung und wurde an die Anforderungen der Mechatronik angepasst. [5, S.26]

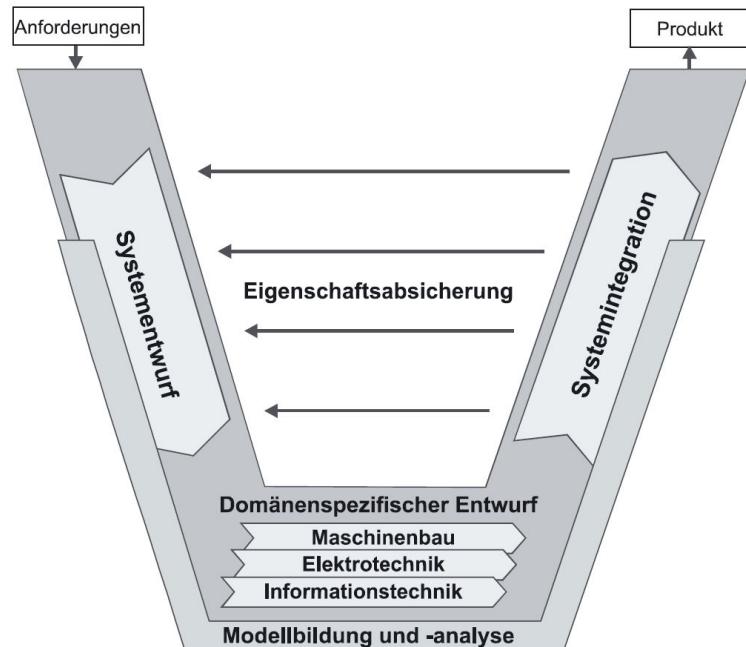


Abbildung 2.1: V-Modell als Makrozyklus nach VDI 2206 [5, S.29]

Anforderungen: Zunächst werden durch den Entwicklungsauftrag Anforderungen beschrieben, die das Produkt zu erfüllen hat. Mehr dazu s. Kapitel 2.1. [5, S.29]

Systementwurf: Als nächstes wird Fachbereichsübergreifend ein Lösungskonzept erarbeitet. Das Produkt wird in Einzelkomponenten zerlegt und jede Komponente erhält eine zu erfüllende Funktionsweise zugeordnet. Die Komponenten werden als System getestet, um die physikalischen und logischen Wirkungsweisen des Gesamtsystems abzusichern. [5, S.29]

Domänenpezifischer Entwurf: Auf Grundlage des zuvor erarbeiteten Lösungskonzeptes, arbeiten die Fachbereiche ihre Lösungen aus. In diesem Zyklusabschnitt ist der Detailgrad am höchsten. Die einzelnen Komponenten auf Mikroebene (Bauteil, Baugruppe, Funktion, Klasse,...) werden durch die Fachbereiche ausgelegt und berechnet, sodass die angestrebten Funktionsweisen erfüllt werden. [5, S.30]

Systemintegration: Die von den Fachbereichen erstellten Komponenten für das mechatronische System werden in das System integriert, um das Zusammenwirken innerhalb der Systemumgebung zu testen. [5, S.30]

Eigenschaftsabsicherung: Während der Entwicklung einzelner Komponenten muss regelmäßig überprüft werden, ob der Entwurfsfortschritt dem Lösungskonzept und den Anforderungen entspricht. Dies gilt ebenso für einzelne Subsysteme und das Gesamtsystem, sodass die geforderten Systemeigenschaften erfüllt werden. [5, S.30]

Modellbildung und -analyse: Parallel zu den bis hier beschriebenen Phasen des V-Modells werden die Systemeigenschaften rechnergestützt abgebildet und simuliert. Dabei werden Modelle einzelner Komponenten erzeugt, die bei der Systemintegration unterstützen und die Fehlersuche erleichtern. [5, S.30]

Produkt: Mit jedem Durchlauf des Makrozyklus V-Modell entsteht ein Produkt. Bis zu einem Serienprodukt sind meistens mehrere Iterationen notwendig. Zwischenprodukte sind dabei z.B. das Labormuster, das Funktionsmuster, das Vorserienprodukt etc. Das Produkt konkretisiert sich mit jedem Durchlauf. [5, S.30]

Bei systemkritischen Komponenten empfiehlt es sich ggf. zunächst einen beinahe serientauglichen Stand anzustreben, um großen Schäden am Gesamtsystem vorzubeugen. Somit ist die Anwendung des Modells in seiner zeitlichen Abfolge nicht immer gleich der logischen Reihenfolge. [5, S.26]

2.4 Embedded Systems / Rapid Prototyping

Die Begriffe „Embedded Systems“ - zu deutsch: eingebettete Systeme - und RP sind im Entwicklungsprozess direkt miteinander verknüpft. Ein eingebettetes System bezeichnet eine Rechenmaschine, die zur Verwendung einer Maschine bzw. einem Gerät beiträgt, ohne direkt für den Nutzer sichtbar zu sein. [6, S.8]

Der Nutzer gibt Anweisungen über ein Eingabemodul wie Knöpfe oder indirekt mittels Gaspedal über Sensoren an ein Steuergerät. Dieses verarbeitet die Eingangssignale und gibt Signale an die Aktorik aus. Das Steuergerät als Rechenmaschine ist das eingebettete System. Die Hard- und Software entscheidet darüber, welche Aktoren wie angesteuert werden. Rapid Prototyping beschreibt den Prozess, neue Funktionen für eingebettete Systeme schnell evaluieren und optimieren zu können. [7]

Um die Kosten bei der Entwicklung zu reduzieren wird anstelle der echten Hardware eine RP-Hardware verwendet. Diese besitzt eine Vielzahl an Ein- und Ausgängen für digitale, wie auch analoge Signale, um ein breites Spektrum an Variationen zu ermöglichen. Die Software eingebetteter Systeme wird ebenfalls zunächst auf dem PC simuliert und das Verhalten durch 'virtual Prototyping' abgebildet. [7] Die Entwicklungsabteilungen beziehen erfahrungsgemäß Rapid Prototyping Produkte von externen Unternehmen wie der ETAS GmbH.

Eine Alternative zu vorgefertigten Produkten ist die Lochraster- oder Steckplatine. Auf den Platinen können sinngemäß Schaltungen aufgebaut werden. Es werden zu diesem Zweck steckbare Bauelemente verwendet und die Schaltung hat eine vom späteren Design abweichende Form. In dieser Form lassen sich Schaltungen testen und validieren, ohne zuvor eine Platine fertigen zu lassen. [8, S.288]

2.5 Brennstoffzellen im Automobil

Mit dem Ziel der Reduzierung von Treibhausgasen werden Antriebe auf Basis von regenerativen Energien immer wichtiger. [9, S.129] Dabei spielt die Elektromobilität eine wesentliche Rolle. Allerdings muss an dieser Stelle zwischen rein batteriebetriebenen Fahrzeugen (BEV) und Fahrzeugen mit Brennstoffzellen (FCEV) unterschieden werden. Beide Fahrzeugtypen werden mit elektrischen Maschinen betrieben. Der Unterschied liegt in der Energiequelle. Der Name „Batteriebetriebenes Fahrzeug“ ist selbsterklärend. Eine Batterie dient als Energiespeicher und versorgt das Fahrzeug und damit die elektrische Maschinen des Antriebsstranges mit Strom. Die elektrischen Maschinen können das Fahrzeug antreiben und Energie mittels Rekuperation in die Batterie speisen. Bei der Rekuperation wird der Motor im Generatorbetrieb verwendet, wodurch die Batterie geladen und das Fahrzeug elektrisch gebremst wird. Dadurch wird die

reduzierte kinetischen Energie während des Bremsprozesses teilweise in der Batterie gespeichert und geht nicht nur in Wärmeenergie über die Bremsscheiben verloren. Die Reichweite mit einer Energieladung ist bei einem BEV durch die Kapazität der Batterie beschränkt und kann mit Hilfe von Rekuperation geringfügig erweitert werden. [9, S.130] Eine Herausforderung der BEV ist das Gewicht der Batterie. Aktuell kann von ca. 100 – 200 Wh/kg ausgegangen werden, was mit einer Batteriekapazität von 40 kWh zu einer Reichweite von ca. 200 km und einem Energiespeichergewicht von 300kg führt. [9, S.133] Die Brennstoffzelle dagegen erzeugt bei der elektrochemischen Reaktion von Wasserstoff und Sauerstoff kontinuierlich elektrische Energie und als Nebenprodukt Wasser. Als Energiespeicher wird Wasserstoff in Druckbehältern mitgeführt und der Sauerstoff wird aus der Luft entnommen. [10, S.101] Durch den Gasförmigen bzw. flüssigen Brennstoff Wasserstoff kann ein Brennstoffzellenfahrzeug innerhalb von wenigen Minuten betankt und die Fahrt fortgesetzt werden. Die Leistung der Brennstoffzelle ist ähnlich einem Verbrennungsmotor abhängig vom Betriebspunkt. Um die Brennstoffzelle im optimalen Betriebspunkt zu betreiben und Lastsprünge beim Beschleunigen nicht auf die Brennstoffzelle zu übertragen, wird eine Pufferbatterie verwendet. Neben einer Effizienzsteigerung der Brennstoffzelle wird dadurch auch die Speicherung der Bremsenergie - also Rekuperation - ermöglicht. Es gibt unterschiedliche Hybridisierungsgrade, die durch das Verhältnis der Gesamtleistung des Brennstoffzellenstacks und der Kapazität der Batterie bestimmt werden. Abb. 2.2 zeigt ein Brennstoffzellenhybridfahrzeug mit verhältnismäßig kleiner Batterie. [11, S.178]

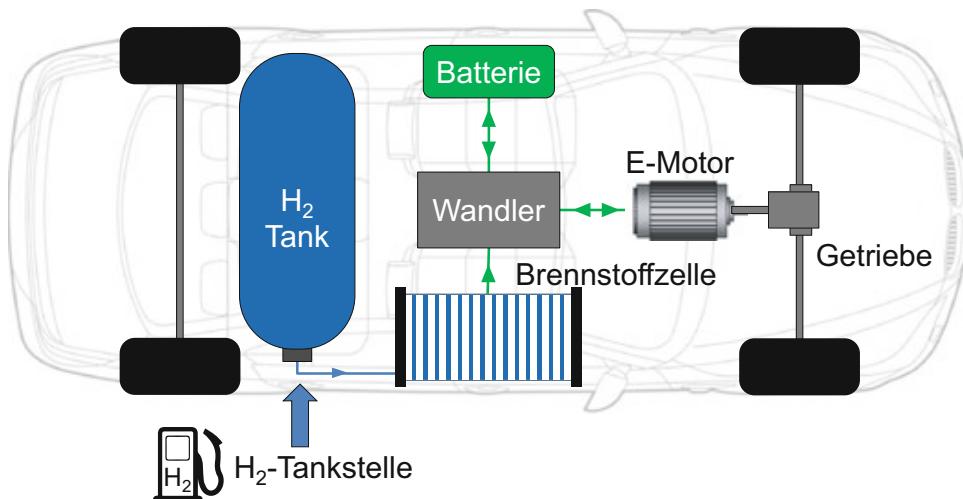


Abbildung 2.2: Schematische Darstellung eines Brennstoffzellenhybridfahrzeugs [11, S.178]

2.6 Konzept Heliocentris HyDrive

Der HyDrive (s. Abb. 2.3) von der Firma Heliocentrisacademia [12] bietet Schüler:innen und Studierenden mit praxisnahen Experimenten Einblicke in die Funktionsweise von Brennstoffzellenhybridfahrzeugen. Der Aufbau in Modellgröße besteht aus dem Fahrzeug, einem Rollenprüfstand und einer Brennstoffzelle für die Elektrolyse. Zur Steuerung und Datenerfassung wird die HyDrive-Software benötigt, sowie ein Bluetooth Dongle um sich mit der Hardware verbinden zu können. Der Prüfstand mit Fahrzeug ist ca. 40 cm lang und 12cm breit. Die Elektrolyse steht auf einer quadratischen Fläche mit 12,5 cm Breite. Das „Energy Control Board“ ist ca. 12 cm lang, 8,5 cm breit und stellt die Bluetooth-Schnittstelle zum Computer bereit. Mit dem Board werden alle Fahrzeugkomponenten, wie auch das Prüfstandboard verbunden. Es bildet das zentrale Steuergerät und befindet sich im hinteren Teil des Fahrzeugs.

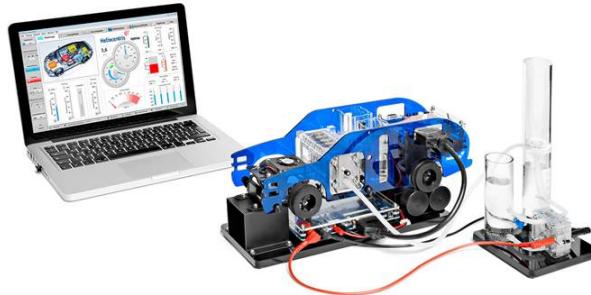


Abbildung 2.3: HyDrive - Electric Vehicle Trainer [12]

Das Prüfstandboard liegt auf dem Prüfstand auf, ist 16 cm breit, 10 cm lang und versorgt die Elektrolyse mit Spannung, wie auch die E-Maschine des Prüfstandes. Die genannten Boards sind eingebettete Systeme im HyDrive ausgestattet mit festen Funktionen. Die Funktionen können verwendet, jedoch weder verändert noch ergänzt werden. Das Produkt kostet telefonischen Angaben der Firma Heliocentrisacademia zufolge ca. 2500 €. In den folgenden Passagen wird auf die einzelnen Baugruppen Prüfstand, Elektrolyse bzw. Tankstelle und Fahrzeug genauer eingegangen.

Das Fahrzeug enthält ein Brennstoffzellenstack (s. Abb. 2.4) bestehend aus fünf Brennstoffzellen. Die Brennstoffzellen erzeugen je nach Betriebspunkt eine Leistung von bis zu 1 W und eine Spannung von max. 4,5 V. Parallel zum Brennstoffzellenstack geschaltet ist ein Superkondensator mit einer Kapazität von 3,3 F und einer maximalen Ausgangsspannung von 5 V. Der Superkondensator kann einerseits durch das Brennstoffzellenstack oder mittels Rekuperation über die elektrische Maschine (Motor) des Fahrzeugs geladen werden. Zur Erhöhung des Luftstroms entlang der Brennstoffzellenmembranen befindet sich vor den Zellen ein Lüfter. Zur Speicherung von Wasserstoff im Fahrzeug sind zwei 30 cm³ Tanks (s. Abb. 2.5) in der Mitte des Fahrzeugs eingelassen und mit einem Wasserabscheider (s. Abb. 2.6) verbunden. Zur Simulation eines elektrischen Verbrauchers wie einer Klimaanlage, dem Radio o.ä. kann mittels Schieberegler eine elektrische Last gesteuert werden. Die Last bezieht die Energie aus der vom Anwender ausgewählten Energiequelle Kondensator oder Brennstoffzellenstack.



Abbildung 2.4: Brennstoffzellen- stack [12]



Abbildung 2.5: Wasserstofftank [12]



Abbildung 2.6: Wasserabscheider [12]

Der Rollenprüfstand (s. Abb. 2.7) besteht aus zwei Achsen die auf einem Bock montiert sind. Die vordere Achse wird durch einen E-Motor betrieben. Im vorderen Drittel des Prüfstandes befindet sich ein zweiter Bock. Darauf sind zwei Stifte montiert, die zur Fixierung des Fahrzeugs dienen. Für Prüfläufe wird die Hinterachse des Fahrzeugs zwischen den zwei Achsen des Prüfstandes platziert und auf die zwei Stifte im vorderen Drittel aufgeschoben.

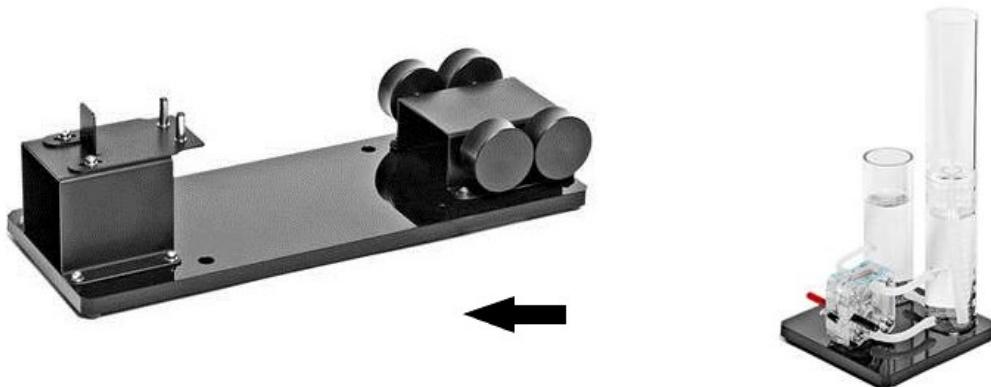


Abbildung 2.7: Rollenprüfstand [12]

Abbildung 2.8: Tankstelle [12]

Die Tankstelle dient der lokalen Zwischenspeicherung und zur Erzeugung von Wasserstoff mittels Elektrolyse. Der Wasserstoff wird mit einer Brennstoffzelle (s. Abb. 2.8) hergestellt, die für eine maximale Spannung von 4 V und einen Strom von 2,5 A ausgelegt ist. Der linke Turm in Abb. 2.8 wird mit destilliertem Wasser gefüllt und dient als Wasserquelle für die Elektrolyse. Der rechte Turm in Abb. 2.8 wird ebenfalls gefüllt und dient als Wasserstoffspeicher. Während der Produktion von Wasserstoff wird das Wasser im rechten Turm verdrängt. Das Wasser im Turm fungiert als Dichtmittel und verhindert das Entweichen von Wasserstoff. In Abb. 2.8 am rechten Turm eine Skala zu erkennen. Anhand des verdrängten Wassers, kann das produzierte Wasserstoffvolumen anhand der Skala abgelesen werden. Am oberen Rand des Tanks ist ein Anschluss zur Betankung des Fahrzeuges vorgesehen.

Die Software dient zur Steuerung und Datenerfassung. Beim Start der Benutzeroberfläche wird zunächst die Bluetooth-Verbindung zum HyDrive hergestellt. Nach erfolgreicher Verbindung können die Fahrzeugdaten empfangen und das Fahrzeug gesteuert werden. Auf der linken Seite der Benutzeroberfläche (s. Abb. 2.9) befindet sich eine Leiste mit Schaltern für das De-/Aktivieren wesentlicher Komponenten. Darunter fallen die Elektrolyse, das Brennstoffzellenstack sowie der Superkondensator. Zum Steuern des Ventilators ist ein stufenloser Schieberegler vorgesehen und für die Simulation eines elektrischen Verbrauchers kann mit einem weiteren Schieberegler eine elektrische Last eingestellt werden. Die zuvor und im Folgenden erwähnten Schieberegler werden durch eine numerische Eingabemöglichkeit jeweils unter dem entsprechendem Schieberegler ergänzt. Über den Reiter „Control“ gelangt man zur Fahrzeug- und Prüfstandumgebung. In diesem Bereich lässt sich die Fahrzeugeleistung in Form von Vortrieb oder Rekuperationsleistung sowie die Prüfstandeleistung über Schieber einstellen. In der Mitte der Ansicht werden die Raddrehzahl, wie auch die aktuelle Fahrzeugeleistung in jeweils einer Messuhr dargestellt. Des Weiteren werden alle Spannungen von Brennstoffzelle, Superkondensator und Elektrolyse wie auch der zugehörige Stromfluss angezeigt. Der Reiter „Energy Flow“ visualisiert den Energiefluss des gesamten Systems. Darüber hinaus lassen sich unter den Reitern „Live Plots“ beliebige Fahrzeugdaten sowie unter „Electrolyzer“, „Fuel Cell“ und „Supercap“ komponentenspezifische Daten in einem Oszilloskop plotten. Zum HyDrive wird ein knapp 180 Seiten langer Experiment Guide geliefert in dem umfangreich die wesentlichen Bestandteile und Funktionen von Brennstoffzellentechnologie wie auch Elektromobilität erarbeitet werden.

2.6 Konzept Heliocentris HyDrive

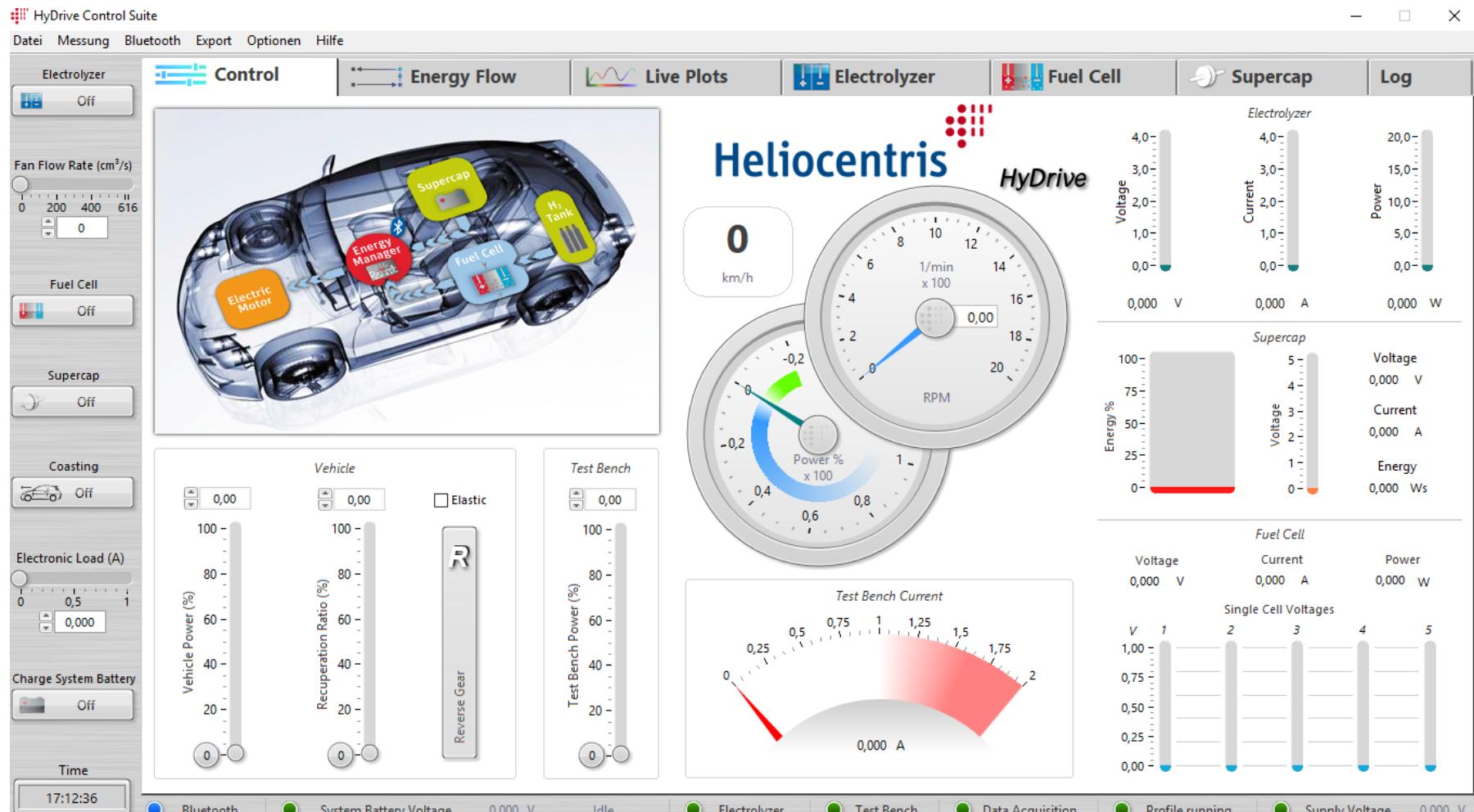


Abbildung 2.9: Grafische Oberfläche des HyDrive

3 Methodik

3.1 Anforderungsanalyse

Zur Erstellung einer initialen Anforderungsliste (s. Anhang) werden zunächst in Kapitel 3.1.1 die Anforderungen gegeben durch die Hochschullehre erörtert. Diese beziehen sich hauptsächlich auf die zu vermittelnden Inhalte und Kompetenzen heutiger Ingenieure. In Kapitel 3.1.2 werden die Anforderungen gegeben durch den HyDrive besprochen. Die Anforderungen beziehen sich hauptsächlich auf die Eigenschaften der RP-Hardware.

3.1.1 Anforderungen an die Hochschullehre

Mit der zunehmenden Digitalisierung und steigenden Komplexität technischer Errungenschaften steigt auch der Anspruch an den Ingenieur hinsichtlich seiner Kompetenzfelder. Computergestützte Entwicklung ist heutzutage Stand der Technik, wodurch sich der Ingenieur mit einer Vielzahl an Programmen auseinandersetzen muss. Funktionen werden für Steuergeräte programmiert und fachspezifische Programme von Ingenieuren und Programmierern gemeinsam entwickelt. Die zuvor genannten Aufgaben bilden nur einen kleinen Teil der Arbeit eines Ingenieurs ab und lassen sich durchaus fortsetzen. Ein wesentlicher Bestandteil sind nicht zuletzt mechatronische Systeme. Bei deren Entwicklung werden Grundkenntnisse über elektronische Komponenten, Informationsverarbeitung und Systemtheorie vorausgesetzt. Gleichzeitig bedarf es Grundkenntnisse über Thermodynamik, Strömungslehre und technische Mechanik, wie auch Wissen über Regelungstechnik und Software-Technik. [13, S.426]

Es wird klar, dass der Ingenieur von heute nicht nur Fachexperte eines spezifischen Bereichs sein kann. Nur mit einem ganzheitlichen und vernetztem Denken kommt der Ingenieur mit den richtigen Methoden zu innovativen Lösungen für die täglichen Arbeitsaufgaben. Dies erfordert, dass während des Studiums ein Maß an Überschneidung zwischen den einzelnen Fachbereichen vermittelt wird, was durch die fachbereichsübergreifende Lehre in Elektrotechnik, Maschinenbau und Informatik gewährleistet werden sollte. Dadurch wird die interdisziplinäre Kommunikation erleichtert, woraus ebenso Voraussetzung an die Sozialkompetenz folgen um Sachverhalte einem interdisziplinären Team sinngemäß erklären zu können. [1, S.1] Auf dieser Grundlage lassen sich die vier folgenden Kompetenzfelder definieren [1, S.9] :

Fachkompetenz:

Umgangssprachlich kann die Fachkompetenz auch als Fachwissen bezeichnet werden. Bei der Vermittlung von Fachkompetenz ist das Ziel die Vermittlung von neuen Fachinhalten, wie auch die Inhalte mit bereits Gelerntem zu verknüpfen. Für den Bereich der Ingenieurswissenschaften bedeutet Fachkompetenz relevante Technologien zu verstehen, einsetzen und weiterentwickeln zu können. Ein Augenmerk sei auch auf die Aktualität gerichtet, da veraltetes Fachwissen in der Praxis nicht von Bedeutung ist. [1, S.10]

Mit Blick auf den HyDrive bezieht sich die Fachkompetenz auf die Brennstoffzellenhybridtechnologie und den Prüfstandaufbau. Charakteristik einer Brennstoffzelle, Möglichkeiten der Energierückgewinnung mittels Rekuperation und der Energiefluss sind ein Teil der Technologie.

Nachdem ausreichend Wissen über das vorhandene System gesammelt und Optimierungspotential gefunden wurde, kann mit Hilfe eingebetteter Systeme und der Funktionsentwicklung das Systemverhalten angepasst werden. Also wird das Fachwissen hinsichtlich Funktionsentwicklung und der Verwendung

eingebetteter Systeme vermittelt. Hinzu kommt das Fachwissen abhängig von der entwickelten Erweiterung.

Methodenkompetenz: Die Methodenkompetenz bezeichnet den Weg zum Ziel und die verwendeten Werkzeuge. Also wie man zur Lösung einer Problemstellung kommt. Dabei beginnt die Problemlösung mit dem Finden verschiedener Konzepte und der Entwicklung einer Lösungsstrategie. Wie zuvor angeprochen ist das Arbeitsfeld eines Ingenieurs sehr vielfältig. So ist die Bewertung verschiedener Konzepte (z.B. nach VDI 2225 in 2.2), die Entwicklung gemäß dem V-Modell (s. 2.3) und die Dokumentationsweise ebenso von Relevanz wie Planung, Organisation und Moderation. [1, S.10] Hinzu kommt mit steigendem Digitalisierungsgrad ein steigendes Datenaufkommen. Dabei ist die systematische Verarbeitung von Bedeutung um die Ergebnisse gezielt in den Entwicklungsprozess mit einbringen zu können. Das kann neben der Datenanalyse ebenfalls die Modellbildung beinhalten. [1, S.13]

Sozialkompetenz: „Eine Kette ist nur so stark wie ihr schwächstes Glied“. Dieses Sprichwort ist auch außerhalb der Konstruktion, also für die Entwicklung innerhalb eines Teams von Bedeutung. Jedes Glied also jeder Entwickler, Gruppenleiter und Teamleiter haben ihre Stärken. Die Verbindung dazwischen hält das Team zusammen. Dabei kommt es auf die Sozialkompetenzen jedes Einzelnen an. Sozialkompetenz beinhaltet unter anderem Teamfähigkeit, aktive Kommunikation und das Herunterbrechen komplexer technischer Sachverhalte. Insbesondere der Austausch innerhalb eines interdisziplinären Teams erfordert die Fähigkeit Sachverhalte für fachfremde Teammitglieder herunterzubrechen. [1, S.11] Aufgrund der steigenden Komplexität technischer Entwicklungen und Entwicklungsmethoden steigt die Bedeutung zu wissen, wer entsprechende Fertigkeiten zur Lösung von Problemen besitzt und entsprechend mit der Person zu kommunizieren. [1, S.13]

Digitalkompetenz: Durch die zunehmende Digitalisierung in vielen Bereichen des Ingenieurwesens, werden zunehmend IT-Kenntnisse und deren Ausbau vorausgesetzt, um neue Technologien anwenden zu können. Um auf Systemebene die richtige Technologie auswählen zu können, wird digitale Fachkompetenz benötigt. Mit steigendem Digitalisierungsgrad steigen ebenso die aufkommenden Datenmengen. Für eine sinnvolle Nutzung werden die Daten computergestützt mit Algorithmen und anderen Werkzeugen ausgewertet. Zur Auswertung ist die Interaktion des Nutzers erforderlich und damit nicht zuletzt dessen Digitalkompetenz auf Ebene der Methodenkompetenz. [1, S.13]

3.1.2 Anforderungen gegeben durch den HyDrive

Im folgenden Abschnitt werden die Anforderungen an eine RP-Hardware besprochen, die durch die Hardware des HyDrive's gefordert werden. Diese Anforderungen müssen von den Konzepten erfüllt werden, damit der HyDrive vollständig auf Basis eines eingebetteten Systems bedient und weiterentwickelt werden kann.

Die unter Kapitel 2.6 genannten Hardwarekomponenten müssen vollständig durch das RP-Konzept abgedeckt werden. Im Fahrzeug muss der Motor mittels Pulswidtemodulation (PWM) vorwärts, wie auch rückwärts drehen können. Außerdem muss eine Rekuperationsleistung einstellbar sein. Im Endeffekt wird zur Steuerung des Fahrzeugmotors nur ein PWM-Signal benötigt. Das PWM kann mit einem Schalter in Form von Relais oder Transistoren geschaltet werden.

Dabei gilt es drei Zustände zu berücksichtigen:

- Vorwärts fahren,
- Rückwärts fahren,
- Rekuperieren - also bremsen.

Das Fahrzeug bietet seitens HyDrive fünf unterschiedliche Zustände bezüglich der Energiequellen:

- Brennstoffzelle zum Antreiben des Motors,
- Superkondensator zum Antreiben des Motors,
- Laden des Superkondensators über die Brennstoffzelle,
- Laden des Superkondensators mittels Rekuperation,
- und das so genannte Segeln, bei dem das Fahrzeug rollt, ohne Energie für den Motor aufzubringen.

Der Anwender kann die verschiedenen Zustände in der Software schalten und über die Programmierung entsprechende Transistoren oder Relais betätigen.

Für die Simulation eines Radios, einer Klimaanlage oder eines vergleichbaren Verbrauchers im Fahrzeug kann über eine elektrische Last der Energieverbrauch erhöht werden. Dafür wird ebenfalls ein PWM-Signal benötigt, um den Verbrauch zu regulieren.

Abb.3.1 zeigt den Schaltplan des Fahrzeugs, während die Brennstoffzellen als Energieressource genutzt werden und eine elektrische Last (E-Load) sowie der Motor (M) des Fahrzeugs in Betrieb sind. Die roten Pfeile symbolisieren dabei den Stromfluss. Der Kondensator (SC) wird in diesem Fall nicht als Energiequelle genutzt.

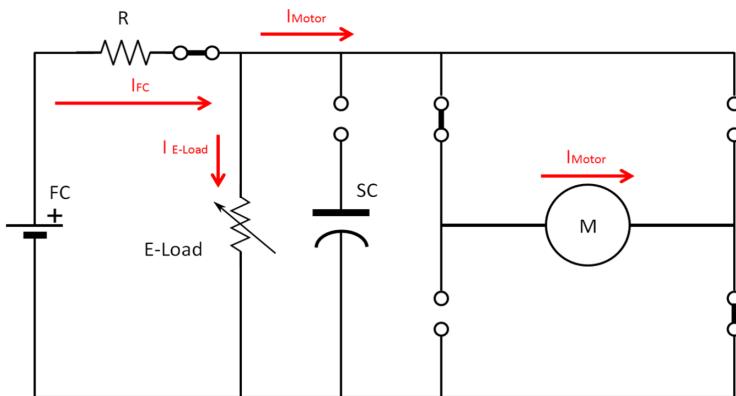


Abbildung 3.1: Schaltplan des HyDrive Fahrzeugs [14, S.173]

Der Lüfter im Fahrzeug ist ebenfalls stufenlos einstellbar und muss aus diesem Grund mit einem PWM-gesteuert werden. Die Versorgung des Lüfters dient jedoch Versuchszwecken und wird aus diesem Grund nicht durch die Energieressourcen des Fahrzeugs betrieben.

Zur Messung des Stromverbrauchs durch den Motor wird ein Stromsensor benötigt. Ebenso um den Verbrauch der elektrischen Last aufnehmen zu können. Außerdem muss die Spannung an den fünf Brennstoffzellen, dem Kondensator und der elektrischen Last gemessen werden, um die Leistungen zu berechnen. Um die Fahrzeuggeschwindigkeit messen zu können, muss ein Raddrehzahlsensor eingelesen werden.

Die elektrische Maschine des Prüfstands wird wie der Lüfter extern versorgt. Diese muss mittels PWM-Signal stufenlos einstellbar sein. Um eine Leistungsbilanz aufstellen zu können, kann die Spannung und der Strom an der E-Maschine gemessen werden. Die Leistungsbilanz bezieht sich z.B. darauf, wie viel Energie dem Fahrzeug über den Prüfstand zugefügt wird und wie viel Energie bei der Rekuperation gespeichert wird.

Die Wasserstofferzeugung wird mit einer Brennstoffzelle und dem Prinzip der Elektrolyse durchgeführt. Um die verbrauchte Leistung zu berechnen, muss die Spannung und der Strom gemessen werden. Mit der gemessenen Leistung kann darüber hinaus der Systemwirkungsgrad berechnet werden. Damit die

Elektrolyse vom Programm gesteuert werden kann, wird ein Schalter in Form eines Relais oder Transistors benötigt. Das Betanken des Fahrzeugs führt der Anwender über ein händisch betätigtes Ventil durch. Ebenso sind die Füllstandsanzeige in der Tankstelle und im Fahrzeug vom Anwender über eine Skala abzulesen. Dies kann mittels Sensoren und einem elektronischen Ventil mechatronisch umgesetzt werden.

Zusammengefasst lassen sich die benötigten Schnittstellen zur Inbetriebnahme des HyDrives in Tabelle 3.1 ablesen.

Tabelle 3.1: Geforderte Schnittstellen an die RP-Hardware

Schnittstelle	Verwendung	Anzahl
Digital	schalten des Brennstoffzellenstacks schalten des Kondensators schalten des Motors im Fahzeug schalten der Elektrolyse schalten der Drehrichtung des Prüfstands Drehzahlsensor	1 1 2 1 1 1
	gesamt digital	7
PWM	Motor des Fahzeugs elektrische Last im Fahrzeug Ventilator im Fahrzeug Motor des Prüfstands	1 1 1 1
	gesamt PWM	4
Analog	Spannung der Brennstoffzellen Spannung des Kondensators Strom der elektrischen Last Strom des Motors im Fahzeug Spannung und Strom der Elektrolyse Strom und Spannung des Prüfstands	5 1 1 1 2 2
	gesamt analog	12

3.2 Auswahl der Konzepte

Basierend auf den Anforderungen der Hochschullehre und den bereitgestellten Möglichkeiten des HyDrives wurde eine Anforderungsliste erstellt (s. Anhang). Die in Tabelle 3.1 gelisteten Schnittstellen müssen durch die RP-Hardware bereitgestellt werden. Daraus ergeben sich zur Ansteuerung des HyDrive Aufbaus für Studierende und Hochschulen unterschiedliche Möglichkeiten. Der Arduino ist bereits im Umfeld der Lehre erprobt [1, S.22] und bietet eine Möglichkeit für Studierende und Hochschulen den Aufbau umzusetzen und zu erweitern. Für diese Arbeit wird auf den Arduino Mega 2560 zurückgegriffen, da dieser im Gegensatz zum Arduino Uno ausreichend Ein-/Ausgänge besitzt. Um auf industrieller Ebene einen Mitbewerber zu vergleichen, wird an dieser Stelle die dSpace MicroAutoBox II hinzugezogen. Das Produkt wird zur Vorentwicklung mechatronischer Systeme verwendet. Als vierter Konzept wird ein Prototyp der ETAS GmbH hinzugezogen. Es handelt sich dabei um das EDUkit, welches eine einfache Bedienbarkeit im Sinne der Arduino Idee und gleichzeitig die automotive Aspekte hinsichtlich Softwareentwicklung nach dem V-Modell verspricht.

Die Tabelle 3.2 fasst wesentliche Aspekte der im Folgenden verglichenen RP-Hardware zusammen.

Tabelle 3.2: Zusammenfassung der Konzepte

	Konzept 1 HyDrive	Konzept 2 Mega 2560	Konzept 3 ACE-Kit	Konzept 4 EDUkit
Gehäusekonzept	in Fahrzeug integriert	frei wählbar	Fräsgehäuse (Aluminium)	modularer 3D-Druck
Programmierung	-	C/C++/Java	MATLAB/Simulink mit dSpace Blockset	ETAS ASCET 7
Steuerung	Umgebung basierend auf LabVIEW	Arduino IDE	dSpace ControlDesk	INCA 7.x
Datenauswertung	Umgebung basierend auf LabVIEW	Arduino IDE	MATLAB Skripte	MDA 8/ ASCMO 5
Hardware-erweiterung	-	modular z.B. ArduinoShield	modular	modular z.B. ArduinoShield
Hardware-schnittstelle	-	Arduino Pinout	Zif I/O - Connector MAB Break-Out Box	Molex-Connector/ Arduino Pinout
Software-schnittstelle	Bluetooth	USB	Ethernet	Ethernet

3.2.1 Konzept Arduino

Arduino ist für Technikinteressierte eine Möglichkeit mechatronische Projekte mit einfachen Mitteln umzusetzen. Es handelt sich hierbei um ein quelloffenes RP-System. Im Internet stehen eine Vielzahl an Foren, Dokumentationen und Baukästen zur Verfügung, um einen leichten Einstieg in die Arduino-Welt zu erhalten und eigene Ideen zu verwirklichen. Den Arduino Mega 2560 gibt es in diversen Kits preiswert zwischen 35€ und 100€ [15] (Stand 24.09.2020) zu kaufen. Das Board ist ca. 10 cm lang und ca. 5,3 cm breit. [15] Zur Programmierung wird die Arduino IDE verwendet. IDE steht für integrated development environment oder zu deutsch integrierte Entwicklungsumgebung. Die Software ist frei erhältlich. Nach der Installation der Arduino IDE kann mittels ein paar Klicks via USB die Verbindung mit dem Arduino hergestellt werden.



Abbildung 3.2: Arduino Mega 2560 [15]

Die Hardware:

Zur Herstellung der ersten Verbindung zwischen PC und Arduino wird lediglich ein USB-Kabel benötigt. Die Hardware wird in dem Fall über den USB-Port des Computers versorgt. Es sind jedoch die Werte aus Tabelle 3.3 zu beachten, wenn mit dem Arduino größere Lasten wie Motoren oder Leuchten angesteuert werden sollen.

Tabelle 3.3: Stromstärke I und Leistung P der Verschiedenen USB Typen [16]

USB Typ	max. Stromstärke [mA]	max. Leistung [W]
USB 1.0/1.1	100	0,5
USB 2.0	500	2,5
USB 3.0/3.1	900	4,5

Für die Versorgung des Arduino Mega 2560 (s. Abb. 3.2) werden laut [17] ca. 70 mA benötigt. Das entspricht mit der USB-Versorgungsspannung von 5 V einer Leerlaufleistung von 0,35 W. Die Leistung jedes USB-Typen ist damit ausreichend. Je nach Anwendungsfall sollte ein Netzteil, eine Batterie oder eine alternative Spannungsversorgung in Erwagung gezogen werden.

Die Pins sind durch die schwarzen Pin-Out-Leisten (s. Abb. 3.2) mit Jumperkabeln zu erreichen. Tabelle 3.4 bietet eine Übersicht der Ein- und Ausgänge. Auf dem Arduinoboard und den Pin-Out-Leisten ist für eine bessere Übersicht die Pin-Bezeichnung aufgedruckt.

Tabelle 3.4: Arduino Mega 2560
Ein- und Ausgänge [15]

Schnittstellentyp	Anzahl Pins
Digital	
I/O davon PWM	54 15
Analog	
Input	16
Seriell	4

Wie bereits angesprochen existieren diverse Baukästen und Starter-Kits im Handel. Darüber hinaus werden so genannte „Shields“ angeboten. Dabei handelt es sich um vorgefertigte Baugruppen zur Erfüllung einer bestimmten Funktion. Shields können entweder direkt auf den Arduino gesteckt oder über Jumperkabel mit dem Arduino verbunden werden. Sollte die Hardware Schaden genommen haben, kann der einzelne Arduino für ca. 35€ [15] (Stand 24.09.2020) neu gekauft werden oder das betroffene Bauteil durch Löten ersetzt werden. Für das Ersetzen von Bauteilen muss durch Testen herausgefunden werden, welches Bauteil defekt ist. Dem Schaltplan kann das verwendete Bauteil entnommen werden. [15]

Die Software:

Um die Module verwenden zu können, muss der Arduino entsprechend programmiert werden. Zu diesem Zweck wird die kostenlos erhältliche Entwicklungsumgebung „Arduino IDE“ verwendet (Hier verfügbar: [18]). Die Programmierung erfolgt zunächst in C, kann jedoch auch in den Sprachen C++ und Java erfolgen. Für Fragen und diverse Anwendungsfälle kann der Entwickler auf die zahlreichen Foren und Bibliotheken im Internet zurückgreifen. Zur Visualisierung von Messwerten können über den Reiter „Werkzeuge“, „Serieller Plotter“ Signale in einem Oszilloskop angezeigt werden. Zur Erstellung und Erweiterung bestehender Projekte werden Programmierkenntnisse vorausgesetzt, wie auch eine gute Dokumentation im Code.

3.2.2 Konzept dSpace MicroAutoBox II

Die MicroAutoBox II (i.F. MAB II) von der dSpace GmbH wurde speziell für RP-Anwendungen im Automotivebereich entwickelt. [19, S.20] Aufgabe der Autobox ist das Testen von Algorithmen und Vorseiensoftware. Sie kann in Kombination mit dem späteren Steuergerät verwendet werden, um Funktionen zu überwachen oder zu triggern. Alternativ dazu kann die MAB II als Steuergerät für ein System dienen und bietet mit der zugehörigen Software eine Lösung, die das V-Modell abdeckt. Diese Lösung wird für Hochschulanwendungen im so genannten ACE-Kit angeboten. [19, S.4] Es beinhaltet eine Hardware und das zugehörige Softwarepaket. [19, S.6] Informationen eines dSpace-Mitarbeiters zufolge liegt der Preis für das ACE-Kit mit einer MAB II als Hardware bei ca. 6000€.

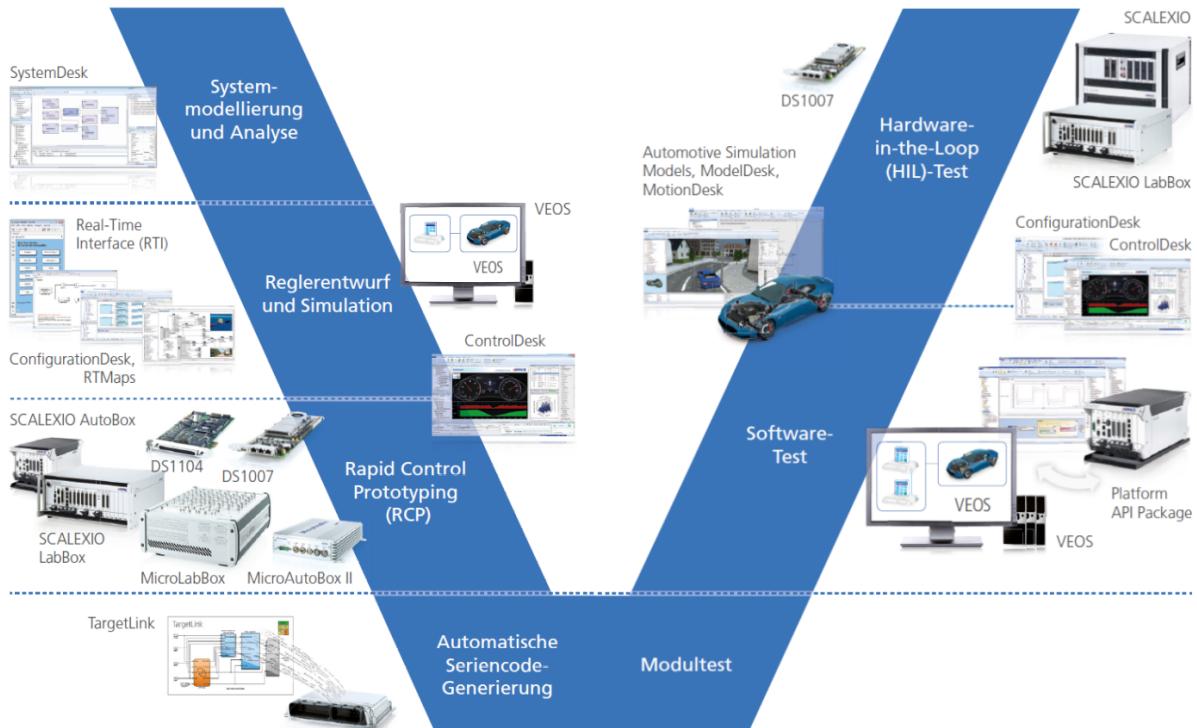


Abbildung 3.3: dSpace Produkte gegliedert am V-Modell [19, S.18, 19]

Das Paket enthält die Softwareprodukte aus Abb. 3.3, womit der Entwicklungsprozess nach dem V-Modell abgedeckt wird. Für die folgende Arbeit ist eine MAB II (Spezifikation 1401/1513 - s. Abb. 3.4) in Kombination mit der Software Real-Time Interface (RTI) sowie ControlDesk gegeben.



Abbildung 3.4: dSpace MicroAutoBox II
Vorderseite [19, S.20]



Abbildung 3.5: dSpace MicroAutoBox II
Rückseite [20]

Die Hardware:

Die MAB II ist 20 cm breit, 20,5 cm lang und 5 cm hoch. Zur Inbetriebnahme wird die MAB II über den linken Anschluss in Abb.3.4 mit einer Spannung zwischen 6 V und 40 V sowie einer maximalen Leistung von 50 W versorgt. [21, S.5] Um auf die MAB II zugreifen zu können wird der Ausgang „Host-PC“ mit dem Ethernetport des Computers verbunden. Über den „Host-PC“-Port kann ebenso wie beim Arduino die AutoBox programmiert, wie auch live Daten erfasst werden. Hinzu kommt die Möglichkeit im Betrieb Parameter der Software verstellen zu können (s. Software). Auf der Rückseite (s. Abb.3.5) befindet sich der „Zero Insertion Force Connector“ (ZIF-Connector) der Zugang zu allen analogen und digitalen Ein-/Ausgängen bietet. Alternativ zum ZIF-Connector lassen sich mit der MicroAutoBox Break-Out Box (s. Abb.3.6) [22] die Pins der MAB II herausführen.

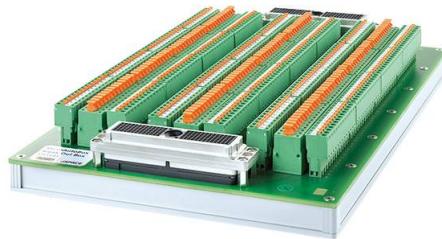


Abbildung 3.6: MicroAutoBox Break-Out Box [22]

Vorab werden alle Signale mit einem Kabel auf die MicroAutoBox Break-Out Box geführt. Dadurch erübrigts sich die Handhabung des ZIF-Connectors und das Crimpen der Kontakte für den Connector um Signale hinzuzufügen. Außerdem kann ein bestehender Kabelbaum angeschlossen und parallel die Signalzustände gemessen werden. Es sind die Hinweise in der Dokumentation der Break-Out Box zu beachten. Die Dokumentation ist im „mydSpace“ - Bereich herunterladbar. Im Schadensfall muss die MAB II an die dSpace GmbH geschickt und repariert werden.

Tabelle 3.5: dSpace MicroAutoBox II
Ein- und Ausgänge [21, S.4]

Schnittstellentyp	Anzahl Pins
Digital	
Input	24
Output	24
davon PWM	24
Analog	
Input	32
Output	8
Busschnittstellen	
CAN	6
(automotive) Ethernet	1
RS232	3
K/L-Line bzw. LIN	3
ECU	2

Tabelle 3.5 beinhaltet die für diese Arbeit relevanten Schnittstellen. Darüber hinaus lassen sich alle Schnittstellen dem Datenblatt [21] entnehmen. Die Busschnittstellen ECU 1, ECU 2 sowie Ethernet werden über die Lemo-Buchsen an der Vorderseite der MAB II bereitgestellt. Des Weiteren findet sich auf der

Vorderseite eine USB-Schnittstelle, ebenfalls in Form einer Lemo-Buchse. [21, S.9] Hier können externe Massenspeicher wie USB-Sticks oder USB-Festplatten zum Speichern von Messdaten angeschlossen werden. [21, S.7]

Die Software:

Zur Programmierung der MAB II wird eine MATLAB/Simulink Lizenz, sowie das passende Blockset von der dSpace GmbH benötigt. Das Blockset wird bei der Real-Time Interface (RTI)-Installation in die MATLAB/Simulink-Bibliotheken eingebunden. Daraus ergibt sich eine grafische Programmierumgebung in MATLAB/Simulink um Ein-/Ausgänge der MAB II anzusteuern und auszulesen, sowie Regler zu entwerfen. Da RTI auf MATLAB/Simulink basiert, lässt sich bei grundlegenden Fragen auf die Foren von MATLAB zurückgreifen. RTI spezifische Fragen müssen über den Support bearbeitet werden. [23] Der Vollständigkeit halber seien die Tools SystemDesk und TargetLink erwähnt. Die Tools sind in der Anwendung mit RTI vergleichbar, jedoch zur Konzeptionierung und Programmierung von Steuergeräten gedacht. SystemDesk bietet die grafische Programmierumgebung und TargetLink ist für die Migration des Codes auf dem Steuergerät nach AUTOSAR-Standard gemacht. Bevor die Programmierung auf einem Steuergerät getestet, wird bietet die Software VEOS die Möglichkeit an, die sogenannte V-ECU (virtuelles Steuergerät) am PC zu testen. [24] ControlDesk kann für das Messen und Verstellen während Funktionstests verwendet werden. Zum einen ermöglicht es, die V-ECUs auf dem PC zu testen, zum anderen kommt das Tool bei der Systemintegration zum Einsatz. Die Oberfläche von ControlDesk bietet verschiedene Darstellungsmöglichkeiten für Messwerte in Form von Oszilloskopen, numerischen Anzeigeelementen oder Tachogeneratoren. Neben der Visualisierung von Messdaten dient ControlDesk der Applikation von Steuergeräten und RP-Hardware. [25]

3.2.3 Konzept ETAS EDUkit

Mit einer Grundausstattung an Hardware sowie einer Softwarepalette, die den Entwicklungsprozess nach dem V-Modell abdeckt, werden Studierende mit dem ETAS EDUkit (s. Abb. 3.7) an die Entwicklung nach automotive Standards herangeführt. Dabei bietet die Hardware eine flexible Umgebung mit einfacher Bedienung gleich der eines Arduinos und die Software deckt die Entwicklungsaspekte nach automotive Standards ab. Der Preis des Prototypen beläuft sich auf ca. 1500€ und beinhaltet die EDUkit-Hardware, die im Folgenden erläuterte Software und einige Videotutorials. Die Videotutorials sollen den Einstieg in die Software erleichtern. Das EDUkit ist auf Anfrage bei der ETAS GmbH erhältlich, wie auch spezifischer Support zur Software. Um die Platine anzufertigen werden Studierenden und Hochschulen von der ETAS GmbH die Baupläne zur Verfügung gestellt. Das Produkt befindet sich derzeit im Prototypenstatus.

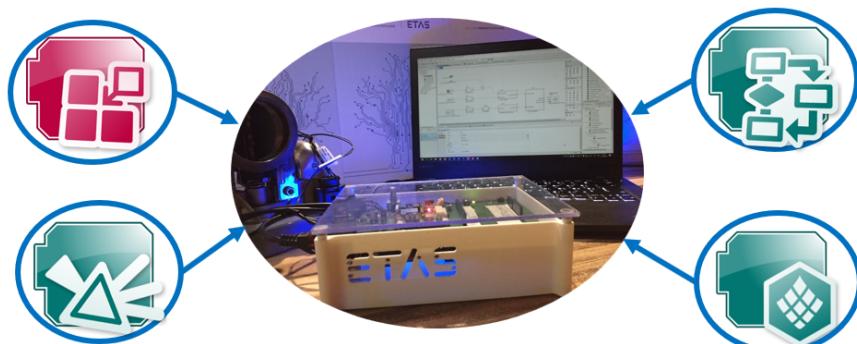


Abbildung 3.7: Das ETAS EDUkit: Hardware und Software in einem Paket

Die Hardware: Das EDUkit ist in seiner aktuellen Ausführung 12,5 cm breit, 17,5 cm lang und 6,5 cm hoch. Zur Inbetriebnahme wird ein Kabel mit einem Stecker der Firma MOLEX (MOLEX 39012020) sowie ein Netzteil mit einer Leistung von max. 100 W benötigt. Das EDUkit wird mit einer Spannung zwischen 9 V und 20 V betrieben und bezieht ohne zusätzliche Hardware eine Leistung von ca. 6 W. Zur Verbindung mit einem Rechner wird lediglich ein LAN-Kabel benötigt. Welches Programm verwendet wird, hängt ganz vom Anwendungsfall ab, aber dazu unter „Software“ mehr.

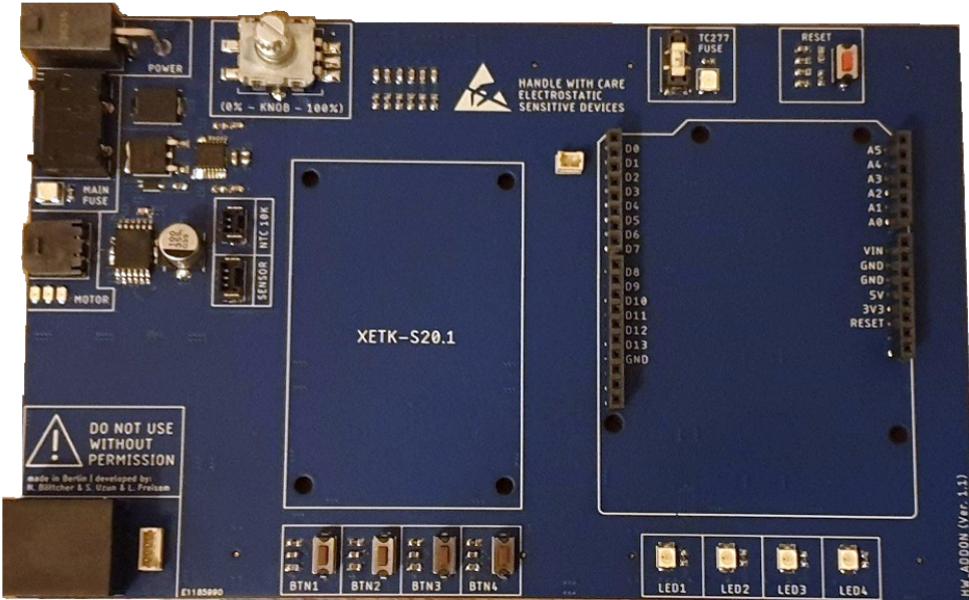


Abbildung 3.8: Draufsicht der Platine des EDUkit

Auf der Platine des EDUkit befindet sich eine H-Brücke, um beispielsweise einen Motor anzusteuern. Dafür steht ein Stecker der Firma MOLEX (43645-0200) auf dem Board zur Verfügung. Die H-Brücke ist mit einer KFZ-Sicherung von 5 A geschützt, die im Falle einer Überlastung gewechselt werden kann. Für die Überwachung der H-Brücke ist ein Stromsensor verbaut. Damit können zum Schutz der Sicherung z.B. Notabschaltungen realisiert werden. Es stehen außerdem ein Anschluss für einen 10 kΩ (MOLEX 45111-0206) sowie ein aktiv Sensor (MOLEX 45111-0306) und ein Potenziometer (PEC11R-4220K-S0024) für erste Versuche zur Verfügung. Des Weiteren bietet das Board vier Taster, vier blaue LED's, eine grüne, gelbe und rote LED, die ebenso wie die zuvor genannten Komponenten programmiert werden können. Um weitere Hardware ansteuern zu können, befindet sich auf dem rechten Drittel der Platine (s. Abb.3.8) eine ArduinoShield Schnittstelle. Das ermöglicht einerseits den schnellen Zugriff auf die Pins über Jumperkabel, auf der anderen Seite können alle ArduinosShields verwendet werden. So stehen diverse fertige Baugruppen für Projekte zur Verfügung. In Tabelle 3.6 sind die Schnittstellen des EDUkits tabellarisch gelistet. Im Falle einer kompletten Auslastung des Boards bietet die ETAS-Produktpalette eine Auswahl kompatibler Module, um z.B. weitere Analogeingänge messen zu können (s. [26]). Die digitalen Pins können ebenfalls als PWM Pins programmiert werden, wenn die drei PWM-Schnittstellen nicht ausreichen. Das kann mit der, im Folgenden näher erklärten, Software ASCET gemacht werden.

Tabelle 3.6: ETAS EDUkit
Ein- und Ausgänge

Schnittstellentyp	Anzahl Pins
PWM-Motorausgang	1
Sensor Eingang	
NTC 10kΩ aktiv	1
Analog	1
I/O davon PWM	14
Input	2
Netzwerk	6
Ethernet	1

Die Software: Neben der Hardwareansicht in Abb.3.7 sind ebenso die Softwarekomponenten des EDUkits dargestellt. Wie sich diese in das V-Modell einordnen, zeigt Abb.3.9. Mit der Software ASCET (links in Abb. 3.9) kann die Steuergerätesoftware vollständig entlang des V-Modells entwickelt werden. Zunächst werden einzelne Komponenten erstellt und anschließend gefahrlos in der Experiment Environment(EE) getestet. Im Anschluss werden die Komponenten zu einem System integriert und als solches getestet. Die einzelnen Funktionen können in Form von reinem C-Code oder auch als Blockdiagramme erstellt werden. Parallel zum Blockdiagramm wird im Hintergrund durch ASCET eine ESDL-Datei generiert. ESDL steht für Embedded Systems Description Language und ist eine Java-ähnliche Sprache. Die Funktionsentwicklung hat zu jedem Zeitpunkt Zugriff auf den ESDL-Code und kann den Code bei Bedarf anpassen.

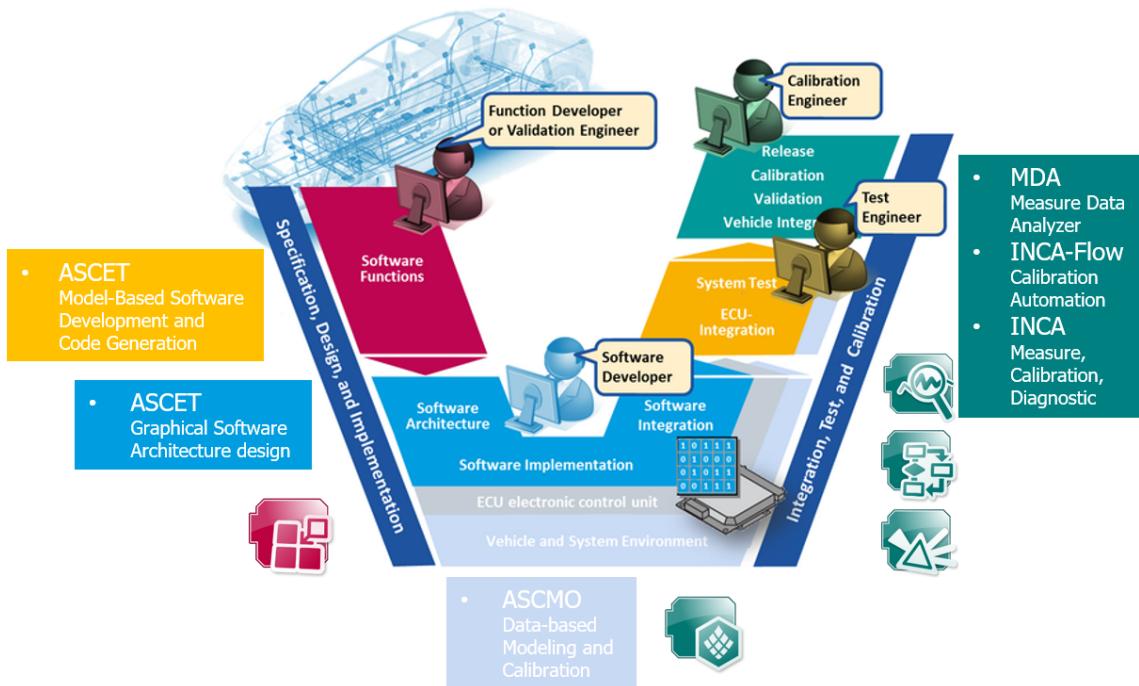


Abbildung 3.9: Die Softwarekomponenten im V-Modell

Mit Hilfe einer State-Machine lassen sich die zuvor erstellten Funktionen in ASCET miteinander vernetzen und Zustände beschreiben bzw. die Übergänge einzelner Zustände definieren. Zum Abschluss wird ein Modell gebildet, dass den Zuständen entspricht, die Funktionen miteinander verknüpft sowie Variablen und Parameter enthält. Variablen werden initialisiert und von der Programmierung gesetzt. Parameter sind vom Nutzer veränderbare Verstellgrößen und im laufenden Betrieb kalibrierbar.

Zum Testen der einzelnen Funktionen kann ein virtuelles Steuergerät erzeugt und in der EE getestet werden. Zu diesem Zweck wird das Steuergerät simuliert. Für die Fehleranalyse können parallel zur Simulation in der EE die Werte der Steuergerätegrößen im ASCET dargestellt werden. Somit kann der Datenfluss einzelner Variablen gezielt nachvollzogen werden, um die Fehlerursache zu finden.

Nach erfolgreich abgeschlossenen Softwaretests wird mit dem Codegenerator automatisiert die Software für das EDUkit erzeugt.

Im nächsten Schritt wird mit INCA (rechts unten in Abb.3.9) die Software auf das EDUkit geschrieben. INCA wird für den Zugriff auf das reale Steuergerät verwendet. Regler, Offsets, Funktionsparameter, wie auch Kennlinien und -felder können mit INCA im laufenden Betrieb angepasst bzw. bedatet werden. Außerdem können mit INCA alle in ASCET als Variablen definierten Größen grafisch in Oszilloskopen, numerischen Anzeigen oder Ähnlichem dargestellt werden. So wird zu jedem Zeitpunkt während des Applikationsprozesses der Zugriff auf die Daten im Steuergerät ermöglicht. Wenn der Applikationsprozess in INCA aufgezeichnet wurde, kann mit MDA (rechts oben in Abb.3.9) die Aufzeichnung erneut betrachtet und gezielt untersucht werden.

Parallel zum Applikationsprozess bietet ASCMO (unten in Abb. 3.9) die Möglichkeit in acht Schritten einen statistischen Versuchsplan (engl.: Design of Experiment - DoE) zu erstellen. Nach Abschluss der Versuchsreihe kann in ASCMO ein Datenmodell erzeugt werden. Ziel des Versuchsplanes könnte ein Modell des Brennstoffzellenstacks sein, z.B. zur Untersuchung der Abhängigkeiten zwischen den Eingängen Sauerstoffgehalt, Wasserstoffgehalt und Temperatur, zu den Ausgängen Strom und Spannung. Die DoE-Methodik ist kein Bestandteil dieser Arbeit und wird aus diesem Grund nicht näher behandelt. Mit INCA-Flow hat der Applikator die Möglichkeit den Versuchsplan aus ASCMO automatisiert auszuführen zu lassen. Dafür wird auf der grafischen Oberfläche von INCA-Flow ein Versuchsablaufplan erstellt und das ASCMO-DoE verknüpft. Neben den Aspekten der Automatisierung können Notabschaltungen oder Trigger umgesetzt werden, die nur in der Testphase benötigt werden und somit nicht in der Software des Steuergerätes existieren.

3.3 Konzeptvergleich anhand der Steuerung einer LED

Das Ansteuern einer LED ist eine Möglichkeit sich mit einem neuen System vertraut zu machen. Für die Steuerung der LED wird ein Taster verwendet, der ein Signal an den Mikrocontroller sendet und die LED daraufhin leuchtet. Die LED leuchtet solange der Taster gedrückt bleibt. Der Versuch enthält keine beweglichen Teile und benötigt keine hohen Ströme, wodurch der Mikrocontroller schaden nehmen könnte. Außerdem ist der Versuch mit ca. 10 Bauteilen samt Kabeln aufgebaut. Die Programmierung kann somit gefahrlos getestet werden. Im folgenden werden die Konzepte miteinander verglichen. Der Vergleich soll die Art der Programmierung mit dem jeweiligen RP-System nahelegen und ebenso vergleichen wie viele Bauteile verwendet werden müssen um eine LED mittels Taster anzusteuern.

Konzept 1 - Der HyDrive

Der HyDrive kann für diese Aufgabe nicht betrachtet werden, da das Produkt für keinerlei Änderungen an Soft- und Hardware vorgesehen ist.

Konzept 2 - Der Arduino

Im Internet stehen zur Orientierung diverse Beispiele zur Verfügung. An dieser Stelle wird auf ein Bei-

spiel von daniel-ziegler.com zurückgegriffen [27]. Der Taster schaltet die Spannung von 5 V (rotes Kabel) an den Pin D2 (gelbes Kabel) durch. Ist der Taster gedrückt, liegt eine Spannung von 5 V am Pin an [27]. Wird der Taster losgelassen, so fließen die Elektronen über den Pull-Down Widerstand von $10\text{ k}\Omega$ ab und die Spannung am Pin geht auf 0 V. Außerdem wird der Pull-Down Widerstand verwendet, um Signalschwankungen des Tasters zu unterbinden. Abb. 3.10 zeigt auf einem Zeitraum von 10 Sekunden die Signalschwankungen ohne einen Pull-Down Widerstand [28].

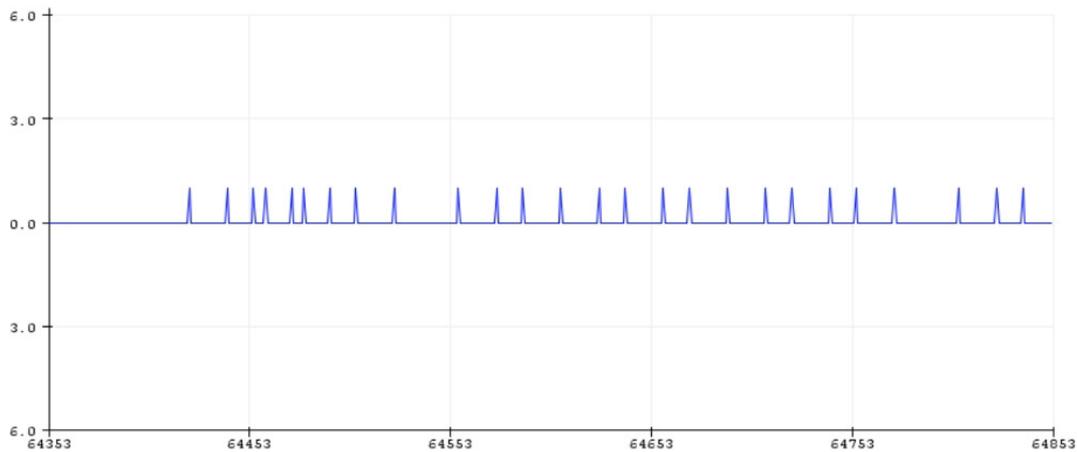


Abbildung 3.10: Eingangssignal ohne Pull-Up/Pull-Down-Widerstand [28]

Die LED wird über den Pin D3 (grünes Kabel) mit 5 V versorgt solange der Taster gedrückt ist. Je nach LED wird für die Spannungsversorgung der LED ein Vorwiderstand benötigt. In diesem Fall werden $220\text{ }\Omega$ verwendet. Abb. 3.11 zeigt den Aufbau basierend auf dem Arduino Uno.

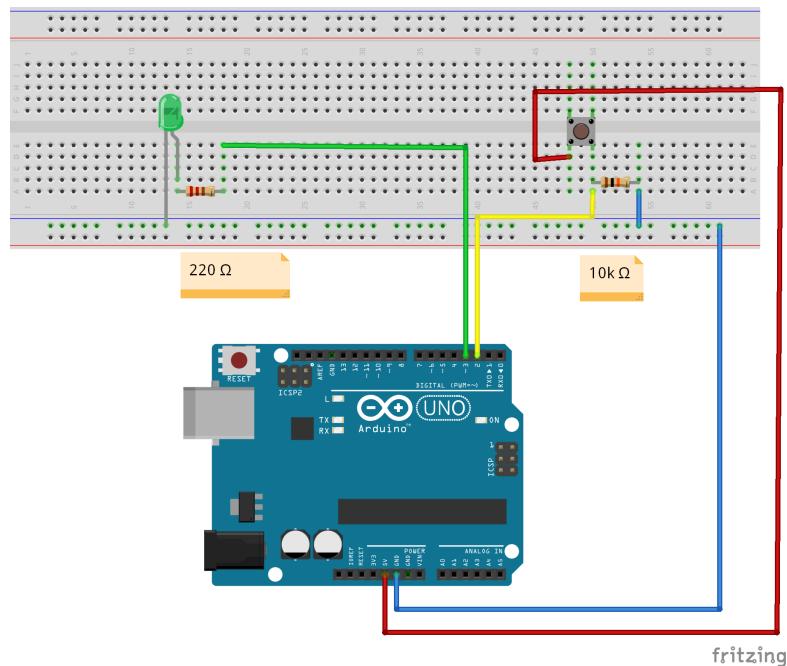


Abbildung 3.11: LED mittels Taster und Arduino steuern [27]

Der Aufbau kann identisch auf den in dieser Arbeit verwendeten Arduino Mega 2560 angewandt werden. Die Bauteile werden über die blaue Schiene des Steckbretts jeweils mit Masse verbunden und über das blaue Kabel zum GND Pin des Arduinos geführt.

Der nachfolgende Code [27] dient der LED-Ansteuerung mittels Arduino Mega 2560.

```
#define TASTER      2 //Pin 2
#define LED_GRUEN    3 //Pin 3

bool led_an = false;

void setup()
{
    pinMode(LED_GRUEN, OUTPUT);
    pinMode(TASTER, INPUT);
}

void loop()
{
    if(digitalRead(TASTER) == HIGH)
    {
        if(led_an == true)
        {
            led_an = false;
            digitalWrite(LED_GRUEN, LOW);
        }

        else
        {
            led_an = true;
            digitalWrite(LED_GRUEN, HIGH);
        }
    }
    delay(150);
}
```

Zunächst werden mit `#define` die Pinnummer für Taster und LED definiert. Im nächsten Schritt müssen in der Funktion `void setup()` den digital Pins mit `pinMode('Pinnummer', INPUT/OUTPUT)` die Funktion Ein- bzw. Ausgang zugewiesen werden. Die Variable `led_an` vom Typ `boolean` wird mit `false` initialisiert, damit die LED bei Programmstart nicht zufällig leuchtet falls noch ein anderer Wert im RAM stehen sollte. Die Funktion `void loop()` ist die eigentliche Funktion, welche zyklisch ausgeführt wird. Mit `if(digitalRead(TASTER) == HIGH)` wird abgefragt, ob der Pin des Tasters aktiviert ist. Wenn dem so ist, leuchtet die LED aufgrund des `else`-Zweiges der `if`-Bedingung `if(led_an == true)` auf.

Konzept 3 - MAB II

An dieser Stelle wird die Programmierung vorweggenommen, da für den Aufbau und die Umsetzung einer kleinen und wichtigen Änderung, die Programmierung fertiggestellt sein muss. In Abb. 3.12 wird die Programmierung in der MATLAB/Simulink Umgebung dargestellt.

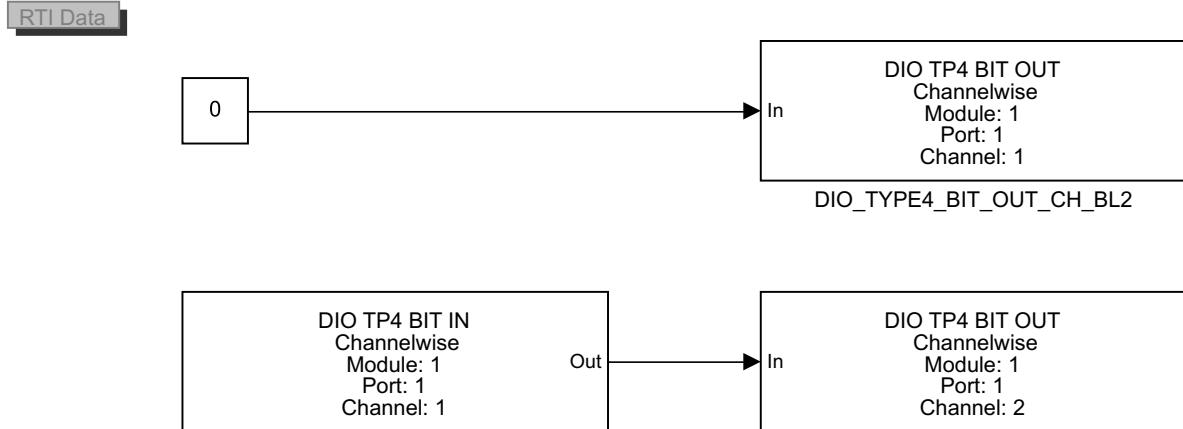


Abbildung 3.12: Programmcode der MAB II für die Steuerung einer LED

Die im folgenden verwendeten Bezeichnungen beziehen sich auf die in den Blöcken stehenden Namen. Auf die Bezeichnung **Modul** wird nicht weiter eingegangen. Die Bezeichnungen **Port** und **Channel** definieren den jeweiligen Pin, werden im folgenden jedoch nicht explizit genannt und können der Abb. 3.12 entnommen werden. Der belegte Pin im ZIF-Stecker kann Tabelle 3.7 entnommen werden. Das Programm wird von oben links nach unten rechts erklärt.

Zunächst wird eine 0 vom Typ boolean in den Block **DIO TP4 BIT OUT** geführt. Der Block **DIO TP4 BIT OUT** ist für die Spannungsversorgung des Tasters notwendig. Die 0 definiert den Zustand „Low“ des Pins zum Zeitpunkt der Initialisierung. Aus Sicherheitsgründen wurde an dieser Stelle eine 0 gewählt, da andernfalls am Pin eine Spannung anliegt und auf dem Steckbrett bei Unachtsamkeit ein Kurzschluss entstehen könnte. Der Wert kann in ControlDesk auf 1, also „High“ gesetzt und somit der Taster aktiviert werden.

Das Signal des Tasters wird über den Block **DIO TP4 BIT IN** unten links in Abb. 3.12 eingelesen und das Ergebnis direkt weitergegeben an den Block **DIO TP4 BIT OUT** unten rechts. Wenn demnach der Taster gedrückt ist, ist der Ausgang des Blocks **DIO TP4 BIT IN** „High“ und wird übergeben an den Eingang des Blocks **DIO TP4 BIT OUT**. Dadurch wird, wie beim Block **DIO TP4 BIT OUT** in der ersten Zeile, der Ausgang auf „High“ gesetzt und die LED leuchtet solange der Taster gedrückt ist.

Für die Umsetzung der Aufgabe kann an dieser Stelle auf den Aufbau des Arduinos zurückgegriffen werden. Dafür muss eine Verbindung zwischen Steckbrett und MAB II hergestellt werden. Anstelle der Jumperkabel zwischen Arduino und Steckbrett, müssen vier Kabel (s. Tabelle 3.7) vom Steckbrett in den ZIF-Connector geführt werden. Außerdem ist der Pull-Down Widerstand von $10\text{ k}\Omega$ gegen einen $4,7\text{ k}\Omega$ Widerstand auszutauschen. Das liegt daran, dass die MAB II, gemäß Datenblatt, den Zustand eines digitalen Pins unterhalb von 1,2 V als „Low“ definiert. Während der Umsetzung wurde zunächst auf der Schaltung des Arduino mit $10\text{ k}\Omega$ aufgebaut. Dabei wurde festgestellt, dass die LED nach kurzem Betätigen des Taster, dauerhaft geleuchtet hat. Das liegt an dem kurzzeitigen Strom, der von dem Versorgungspin des Tasters, durch den Taster und in den digitalen Eingang der MAB II fließt. Einfach ausgedrückt, sammeln sich die Elektronen durch besagten Strom am digitalen Eingang und können nach Lösen des Tasters wieder über den Pull-Down Widerstand abfließen. Ist dieser Widerstand jedoch zu groß gewählt, fällt die Spannung nicht unter den Wert von 1,2 V. Die MAB II wertet den Zustand des Pins als „High“ aus und die LED leuchtet dauerhaft. Mit einem Multimeter kann die Restspannung U am Pin gemessen und mittels Ohm'schem Gesetz der geflossene Strom I berechnet werden.

$$I = \frac{U}{R} \quad (3.1)$$

Der Strom aus Gleichung 3.1 wird für die Dimensionierung des neuen Widerstandes R benötigt. Durch Umstellen von Gleichung 3.1 nach dem Widerstand R und Einsetzen der gegebenen Daten, ergibt sich ein Wert von ca. $5\text{ k}\Omega$. Zur Verfügung steht lediglich ein $4,7\text{ k}\Omega$ Widerstand. Daraus ergibt sich eine Restspannung von ca. 1 V, was von der MAB II als „Low“ erkannt wird.

Tabelle 3.7: Konzeptvergleich LED - MAB II Pinbelegung

Verwendung	Bezeichnung	Pin
Taster Versorgung	DigP 1 ch 1 out	2G
Taster Signal	DigP 1 ch 1 in	2M
LED Signal	DigP 1 ch 2 out	2F
Erdung	GND	1M

Hinweis: Damit 5 V an den digitalen Ausgängen anliegen, verbindet das mit der MAB II gelieferte Jumperkabel im ZIF-Stecker, entsprechend den Vorgaben des Datenblattes, die Pins N1 und M1.

Konzept 4 - EDUkit

Für diese Aufgabe bieten sich zwei Lösungsmöglichkeiten an. Der zuvor verwendete Aufbau des Arduinos oder die auf dem EDUkit verbauten Komponenten. Für diese Aufgabe werden also keine weiteren Bauteile benötigt da auf die integrierten Komponenten zurückgegriffen wird. Betätigt wird „Button 1“ und die „LED 1“ soll leuchten(s. Abb. 3.13).

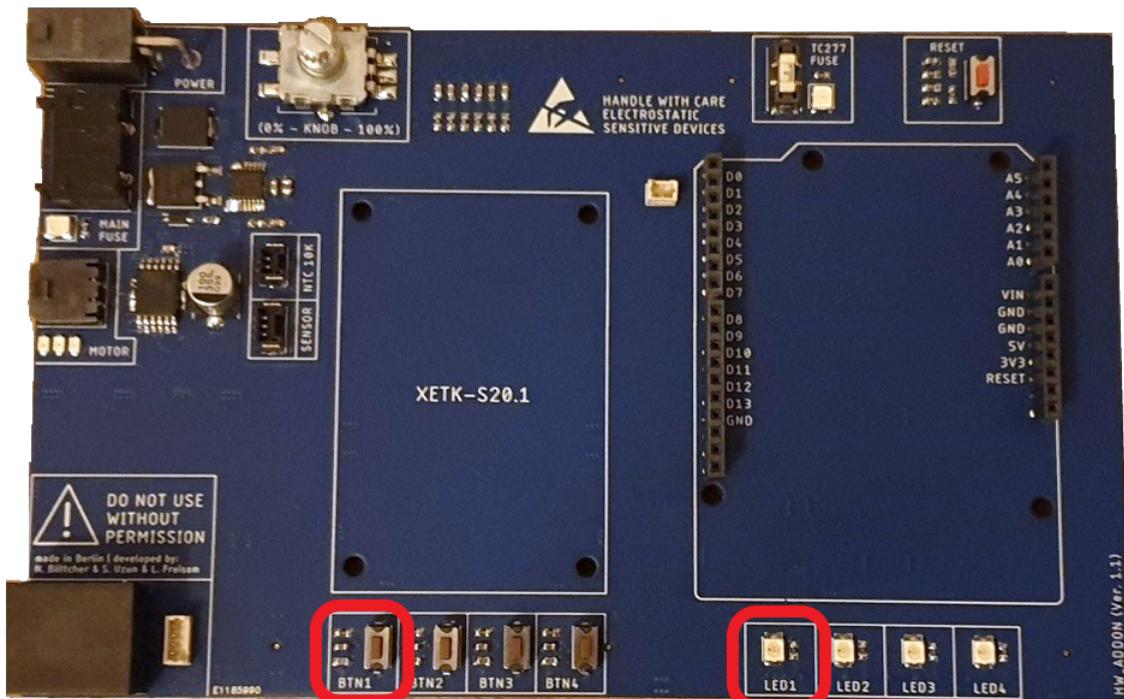


Abbildung 3.13: Verwendete Bauteile des EDUkits zur Steuerung einer LED

Seitens der ETAS GmbH wird für diese Arbeit zu diesem Zweck ein ca. 5 minütiges Video bereit gestellt. Das Ergebnis ist in Abb. 3.14 dargestellt.

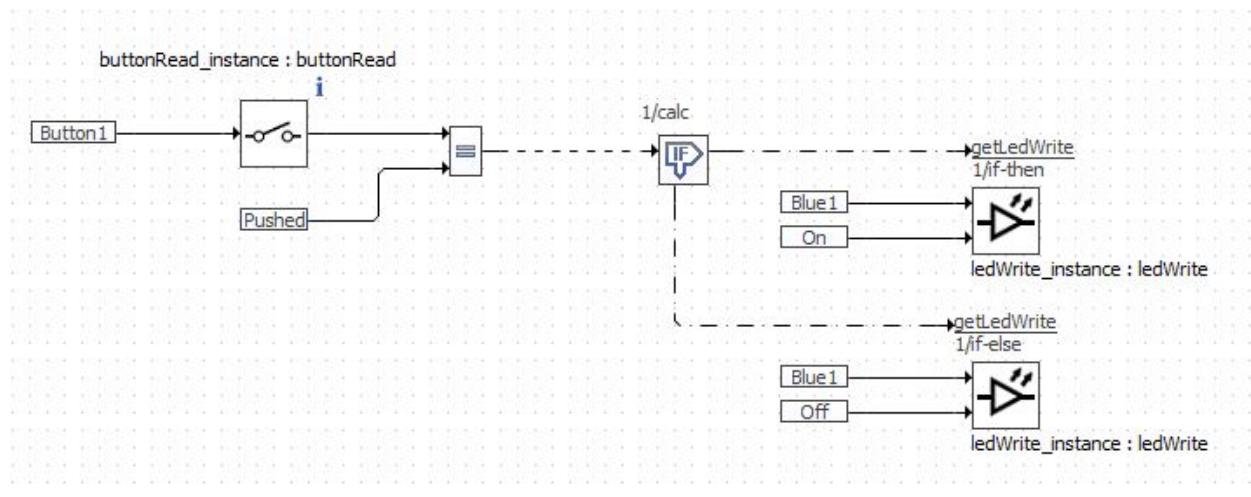


Abbildung 3.14: Programmcode des EDUKits für die Steuerung einer LED

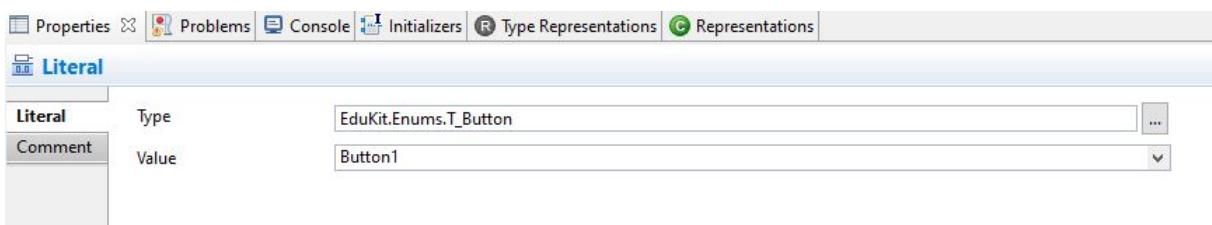


Abbildung 3.15: Properties Fenster eines Literals

Zunächst wird der Taster „Button 1“ mit einem Literal (rechts in Abb. 3.14) definiert. Im „Properties“-Fenster (s. Abb. 3.15) wird dem Literal der Typ EduKit.Enums.T_Button zugewiesen. Dieser Typ kann über die Schaltfläche mit den drei Punkten in der Zeile „Type“ ausgewählt werden. Im Anschluss kann der Wert Button1 in der Zeile „Value“ über die Dropdown-Liste ausgewählt werden.

Dieses vorgehen kann auf die Literals „Pushed“ vom Typ EduKit.Enums.T_ButtonState, „Blue1“ vom Typ EduKit.Enums.T_Led und „On“ vom Typ EduKit.Enums.T_LedState übertragen werden. Über den „Project Explorer“ (rechts in der ASCET-Umgebung) kann auf die bereitgestellten Klassen des EDUKits zugegriffen werden. Mit drag-and-drop kann aus dem Ordner „Classes“ die Klasse „buttonRead“ in die Programmieroberfläche gezogen werden. Gleches Prinzip gilt für die Klasse „ledWrite“. Wenn die Pins der „ledWrite“ Instanz mit den Eingängen „Blue1“ und „On“ verbunden sind, lassen sich die drei Objekte markieren, kopieren und erneut einfügen. Im Anschluss muss lediglich der Wert des Literal On in Off geändert werden.

Zur Programmierung der Logik wird aus der Palette (in ASCET rechts von der Programmieroberfläche) der „=“-Block gezogen, wie in Abb. 3.14 verbunden und das Ergebnis mit dem „if-else“-Block verbunden. Der rechte Ausgang des „if-else“-Blocks wird mit der „ledWrite“ Instanz verbunden, die den Literal „On“ erhält. Der untere Ausgang des „if-else“-Blocks wird mit der zweiten „ledWrite“ Instanz verbunden. Die verwendeten Bauteile können Tabelle 3.8 entnommen werden.

Tabelle 3.8: Konzeptvergleich LED - verwendete Bauteile

	Konzept 2 Mega 2560	Konzept 3 ACE-Kit	Konzept 4 EDUkit
Steckbrett	1	1	0
LED	1	1	0
Vorwiderstand	1	1	0
Pullup-/Pulldown-Widerstand	1	1	0
Taster	1	1	0
Hardwareconnector	0	1	0
Jumperkabel	5	5	0
Summe	10	11	0

3.4 Wahl der Bewertungskriterien für die Nutzwertanalyse

Die Bewertungskriterien orientieren sich an der Anforderungsliste (s. Anhang) und den Anforderungen an die Hochschullehre in Kapitel 3.1.1. Für die Bewertung der vier Konzepte sind die Wünsche in der Anforderungsliste von Bedeutung. Die Erfüllung der Pflichten wird von jedem Konzept vorausgesetzt und ist deshalb für die nachfolgende Bewertung nicht von Relevanz. Die Wünsche lassen sich in vier Hauptmerkmale zusammenfassen:

- Lehre und Weiterbildung
- Voraussetzungen und Randbedingungen
- Inbetriebnahme und Gebrauch
- Problembehandlung

Unter den Punkt „Lehre und Weiterbildung“ fallen die bereits in Kapitel 3.1.1 angesprochenen Kompetenzfelder. Die Fachkompetenz sollte hinsichtlich dem Umgang eingebetteter Systeme und Funktionsentwicklung ausgebaut werden. Neben dem Fachwissen sollte die Methodenkompetenz hinsichtlich der Aufgabenbearbeitung mit dem V-Modell gelehrt werden. Für den Ausbau der Sozialkompetenz sollte die Möglichkeit, innerhalb eines Interdisziplinären Teams Aufgaben zu bearbeiten, gegeben sein. Bei der Arbeit mit und Erweiterung von mechatronischen Systemen ist dies zwangsläufig erforderlich. Die Digitalkompetenz bezieht sich in diesem Fall auf das Arbeiten mit neuen Technologien, entwickeln von Systemverständnis und der Verarbeitung von Messdaten. Der Ausbau der Kompetenzfelder kann mit gezielten oder frei wählbaren Aufgaben bzgl. einer Prüfstanderweiterung für die Studierenden geschehen. Dabei wird praxisnah die Arbeit entlang des V-Modells erlebt. Dahingehend ist der HyDrive in seinem aktuellen Zustand beschränkt, da der HyDrive keine Systemerweiterungen ermöglicht.

Die „Voraussetzungen und Randbedingungen“ betrachten vor allem die Zugänglichkeit der Hardware. Der Preis könnte für Studierende und Hochschulen von Relevanz sein. Der Student hat erfahrungsgemäß ein geringes Einkommen und muss ggf. auf kostengünstige RP-Systeme zurückgreifen. Dahingegen wird an Hochschulen für die Lehre ein Kontingent von z.B. 10 Baukästen benötigt, um Praktika in Kleingruppen durchzuführen. Außerdem sollte der Aufbau auf einen ca. 50 x 100 cm großen Tisch passen um einerseits gut verstaubar zu sein und andererseits während der Verwendung ebenfalls ausreichend Platz auf dem Tisch für Laptop und Schreibmaterial zu belassen. Damit sowohl Hochschule als auch Studierende die Möglichkeit haben sich mit den Systemen auseinander zu setzen, sollte die Software und Lizenzierung über das Internet erreichbar sein. Ebenso sollten Hochschule und Studierende die Möglich-

keit haben die Hardware zu erwerben.

Unter „Inbetriebnahme und Gebrauch“ wird die eigentliche Arbeit mit dem System verstanden, beginnend mit der Installation. Benötigt die Installation 5 Minuten oder mehrere Stunden, müssen zusätzliche Peripheriegeräte erworben werden und vor allem ist der Einstieg in die Entwicklungsumgebung intuitiv? Einen ersten Einblick in die Programmierung und den Umgang mit den RP-Systemen bietet Kapitel 3.3. Der HyDrive als mechatronisches System kann wie zuvor angesprochen mit der Integration einer RP-Hardware weiterentwickelt werden. Dazu sollte die Software übersichtlich programmiert werden können und die Hardware durch Module erweiterbar sein.

Der Punkt „Problembehandlung“ lässt sich untergliedern in Softwareprobleme und Hardwareprobleme. Sollte Hardwareseitig etwas beschädigt werden, wäre schnelle Abhilfe durch Baupläne und eine Online-Community erreicht, wenn die Bauteile im Elektronikfachgeschäft erwerbbar sind und durch Löten bzw. Stecken leicht auszutauschen sind. Bei Softwareproblemen kann eine Community ebenfalls schnell weiterhelfen und einen Telefonsupport ersetzen. Somit lassen sich die Hauptmerkmale jeweils unterteilen in weitere Einzelmerkmale:

- Lehre und Weiterbildung:

- Fachkompetenz
- Methodenkompetenz
- Sozialkompetenz
- Digitalkompetenz

- Voraussetzungen und Randbedingungen:

- Zugänglichkeit/Erwerbbarkeit
- Preis Hard-und Software
- Größe

- Inbetriebnahme und Gebrauch:

- erste Schritte
- benötigtes Zubehör zum Ansteuern einer LED
- Erweiterbarkeit der Software
- Erweiterbarkeit der Hardware

- Problembehandlung:

- Supportkonzept
- Reparatur

3.5 Bewertung nach VDI 2225

Die vier verschiedenen Konzepte werden mit Hilfe der VDI 2225 gemeinsam bewertet. Dafür wurden in Kapitel 3.4 Hauptmerkmale und zu jedem Hauptmerkmal weitere Einzelmerkmale ausgearbeitet. Die Hauptmerkmale werden in der Gewichtungsmatrix gegenüber gestellt. Tabelle 3.9 zeigt die daraus resultierende Gewichtungsmatrix.

Tabelle 3.9: Gewichtungsmatrix leer

Kriterien/Kriterien	Lehre und Weiterbildung	Voraussetzungen/Randbedingungen	Inbetriebnahme/Gebrauch	Problembehandlung	Summe
Lehre und Weiterbildung	x				
Voraussetzungen/Randbedingungen		x			
Inbetriebnahme/Gebrauch			x		
Problembehandlung				x	

Die Zeilen geben jeweils die Gewichtung des Hauptmerkmals wieder. Überwiegt die Priorität des Hauptmerkmals in der Zeile gegenüber dem der Spalte wird eine „1“ eingetragen. Andernfalls eine „0“. Die Einträge der Zeile werden addiert und in der Spalte „Summe“ eingetragen. Daraus ergibt sich die Platzierung der Hauptmerkmale, wonach eine prozentuale Gewichtung zwischen 10% und 35%, entsprechend den Vorgaben in Kapitel 2.2, vergeben werden kann.

Die Einzelmerkmale werden entsprechend dem Urteilsschema in Tabelle 3.10 bewertet. Für eine eindeutige Definition der Kompetenzen werden die Kompetenzfelder im Folgenden zusammengefasst. Für eine Detaillierte Erläuterung sind die Kapitel 3.1.1 und Kapitel 3.1.2 zu betrachten.

1. Fachkompetenz:

- Fachwissen
 - bzgl. Brennstoffzellenhybridfahrzeugen,
 - Prüfstandaufbauten,
 - Programmierung und Entwicklung eingebetteter Systeme

2. Methodenkompetenz:

- Verwendung von methodischen Werkzeugen zum Lösen von Problemstellungen
 - Datenanalyse
 - Modellbildung
- Arbeiten nach dem V-Modell

3. Sozialkompetenz:

- Interdisziplinäres Arbeiten
- Herunterbrechen komplexer technischer Sachverhalte für Fachfremde

4. Digitalkompetenz:

- Verständnis für neue Technologien wie Softwareprodukte
- Arbeiten auf Systemebene
 - z.B. System Fahrzeug mit (unter-)Komponenten

Die Kompetenzen können entsprechend der Skalierung in Tabelle 3.10 auf einer Skala von 0 - 4 zwischen „vermittelt nicht“ und „vermittelt sehr gut“ bewertet werden. Die vergebenen Punkte zeigen klar den Vorteil der Lösungen von der dSpace GmbH und der ETAS GmbH. Jedes Kompetenzfeld wird mit vier Punkten bewertet. Das liegt insbesondere an den umfangreichen Softwarepaketen, die mit der Hardware geliefert werden (s. dSpace - Kapitel 3.2.2 und ETAS - Kapitel 3.2.3). Neben den Aspekten, die der HyDrive hinsichtlich Fachkompetenz mitbringt, können zusätzlich Aspekte der Arbeit mit RP-Hardware und eingebetteten Systemen vermittelt werden. Der HyDrive vermittelt mit dem umfangreichen „Experimentbuch“ neben Fachkompetenzen auch ein Maß an Fehleranalyse, jedoch keinerlei Entwicklungsmöglichkeiten gemäß dem V-Modell oder der Modellbildungsaspekte. Der Arduino dahingegen bildet ein Mittel zwischen den zuvor genannten Lösungen. Bedingt durch die Erweiterbarkeit kann in einem interdisziplinären Team, wie auch Konzept 3 und Konzept 4, der Aufbau erweitert werden und dadurch die Sozialkompetenz gefördert werden. Auf der anderen Seite vermittelt der Arduino aufgrund des kleinen Softwarepaketes nur ausreichend die Methodenkompetenzen.

Die Zugänglichkeit der Hard- und Software wird in drei Stufen unterteilt. „Problemlos“ also über das Internet oder im Handel ohne eine menschliche Zwischeninstanz wird mit vier Punkten gewertet. „Auf Anfrage“ bedeutet eine menschliche Zwischeninstanz im Erwerbsprozess. Die Software ist über das Internet in einem Downloadcenter oder auf Anfrage vom Hersteller ebenso wie die Hardware erhältlich. Dafür werden zwei Punkte vergeben, da es für Hochschulen und Studierende ausreichend ist. Vorteilhaft ist an dieser Stelle der erste Kontakt zwischen Hersteller und Studierenden bzw. Hochschule. Dabei können weitere Punkte wie Preise, Erweiterbarkeit etc. erfragt werden.

Der Preis für das Hard- und Softwarepaket wird linear gewertet. Die Skala lässt sich der Tabelle 3.10 entnehmen.

Die Größe des Aufbaus wird rein objektiv bewertet. Der HyDrive selber gilt als das Ideal. Wenn sich der Umbau auf das RP-System in den HyDrive integrieren ließe, wäre das ebenfalls sehr gut. Jedoch lässt sich die letztendliche Größe des späteren Aufbaus zu diesem Zeitpunkt noch nicht abschätzen, sodass die Größe der jeweiligen Hardware gewertet wird.

Hinsichtlich des Punktes „Inbetriebnahme/Gebrauch“ schneidet der HyDrive am besten ab. Nach der Softwareinstallation kann mit den Experimenten direkt begonnen werden. Dagegen muss zur Inbetriebnahme von Konzept 2, 3 und 4 mehr Zeit investiert werden. An dieser Stelle wird davon ausgegangen, dass lediglich eine Verbindung zwischen PC und Hardware hergestellt werden soll und der RP-HyDrive im Anschluss gesteuert werden kann. Der Arduino überzeugt mit einer einfachen Bedienbarkeit. Der COM-Port muss eingestellt werden, woraufhin die Verbindung zum Arduino hergestellt werden kann. Jedoch müssen zur Inbetriebnahme des RP-HyDrive ggf. weitere Bibliotheken eingebunden werden und das kann aufgrund der Programmierumgebung sehr unübersichtlich werden.

Die Installation und Inbetriebnahme der MAB II ist übersichtlich gestaltet. Jedoch bedarf es einer Vielzahl an Schritten um die Lizenzierung abzuschließen und die nötige Software wie Compiler auf dem PC freizuschalten. Bei Problemen muss die richtige Dokumentation gefunden werden und im Zweifelsfall ist man an den Support angewiesen. Das hängt sicherlich auch mit der Komplexität und den Möglichkeiten des Systems MAB II zusammen und ist für diese Bewertung dadurch nur ausreichend. Auf die genauen Schritte zur Inbetriebnahme soll jedoch nicht weiter eingegangen werden.

Das EDUkit dagegen ist hinsichtlich „erste Schritte“ noch in den Kinderschuhen. Eine Dokumentation zur Inbetriebnahme ist aktuell in der Anfertigung. Für diese Arbeit wurde von der ETAS GmbH eine stichpunktartige Installationsanleitung zur Verfügung gestellt, die den Anwender ausreichend an das EDUkit heranführt. Neben der Installationsanleitung stehen auf Anfrage ca. fünf Videos zur Verfügung, die den Einstieg in die Programmierumgebung erleichtern. Somit waren die ersten Schritte verständlich und der Punkt „benötigtes Zubehör zum Ansteuern einer LED“ konnte bearbeitet werden. Ebenso werden der Punkt „benötigtes Zubehör zum Ansteuern einer LED“ nach den Ergebnissen in Kapitel 3.3 bewertet

und mit Platzierungen abgestuft. Wenn das Experiment jedoch nicht durchgeführt werden kann, wird dies als unbefriedigend gewertet.

Die ersten Schritte, wie auch die Erweiterbarkeit von Hard- und Software können, 0 Punkte, 2 Punkte oder 4 Punkte erhalten. Dabei wird bei den ersten Schritten darauf geachtet, ob diese „selbsterklärend“ sind und eine Anleitung nahezu überflüssig ist, Tutorials verständlich gestaltet und ein leichter Einstieg in das RP-System gegeben ist. Zwei Punkte werden vergeben, wenn die Inbetriebnahme nahezu nur mit dem lesen einer Anleitung funktioniert und somit die ersten Schritte „verständlich“ sind. Der Einstieg in das RP-System kann mit einigen Beispielen innerhalb der verwendeten Software gestützt sein. Sind jedoch die ersten Schritte ohne jegliche Hilfestellung zu bewältigen, wird es bei Studierenden erfahrungsgemäß dazu führen, dass das RP-System nicht weiter genutzt wird.

Tabelle 3.10: Urteilsschema

Urteilsschema		E=4 ser gut (ideal)	E=3 gut	E=2 ausreichend	E=1 gerade noch Tragbar	E=0 unbefriedigend
Einzelkriterien						
Lehre und Weiterbildung	Fachkompetenz	ordinal	vermittelt sehr gut	vermittelt gut	vermittelt	vermittelt bedingt
	Methodenkompetenz	ordinal	vermittelt sehr gut	vermittelt gut	vermittelt	vermittelt bedingt
	Sozialkompetenz	ordinal	vermittelt sehr gut	vermittelt gut	vermittelt	vermittelt bedingt
	Digitalkompetenz	ordinal	vermittelt sehr gut	vermittelt gut	vermittelt	vermittelt nicht
Voraussetzungen/Randbedingungen	Zugänglichkeit/Erwerbarkeit	problemlos	-	auf Anfrage	-	gar nicht
	Preis Hard- und Software	0	<1000	<2500	<3500	>5000
	Größe	HyDrive	Arduino	ETAS	dSpace	-
Inbetriebnahme/Gebrauch	erste Schritte	selbsterklärend	-	verständlich	-	unverständlich
	benötigtes Zubehör	EDUkit	Arduino	dSpace	-	nicht möglich
	zum Ansteuern einer LED					
	Erweiterbarkeit Software	problemlos	-	anspruchsvoll	-	unmöglich
Problembehandlung	Erweiterbarkeit Hardware	problemlos	-	anspruchsvoll	-	unmöglich
	Supportkonzept	Community	-	auf Anfrage	-	gar nicht
	Reparatur	problemlos	-	auf Anfrage	-	gar nicht

Da für die Hochschullehre ein leichter Einstieg ebenso von Bedeutung ist, kann gleichermaßen in diesem Fall null Punkte vergeben werden. Die Erweiterung von vorhandenem Softwarecode sollte im besten Fall „problemlos“ durch einfaches Einbinden von Funktionen geschehen können. Außerdem ist eine gute Lesbarkeit von Code von Vorteilen und dient einem besseren Verständnis. „Anspruchsvoll“ wird die Programmierung und Implementierung neuer Funktionen, sobald der Code Gefahr läuft nicht richtig kommentiert zu werden. Insbesondere bei textueller Programmierung muss auf eine übersichtliche Struktur geachtet werden, damit jeder im Team den Code lesen und erweitern kann. Sollte die Software nicht erweiterbar sein, werden null Punkte vergeben.

Die Hardwareerweiterung bezieht sich auf die Möglichkeiten neue Bauelemente und Baugruppen in das vorhandene System einzubinden. Wenn mit ein paar Jumperkabeln der Versuch an das vorhandene System angeschlossen werden kann, gestaltet sich die Umsetzung neuer Ideen als „Problemlos“. Wenn jedoch zunächst ein Stecker neu bepinnt werden muss, ist das schon als „Anspruchsvoll“ einzustufen. Hinzu kommt, dass ggf. Spezialwerkzeug zum Crimpen oder Endpinnen notwendig ist, wie auch die Crimpkontakte erworben werden müssen. Ist keine Hardwareerweiterung möglich, werden an dieser Stelle null Punkte vergeben.

Zu guter Letzt wird die Problembehandlung betrachtet. Dabei ist das Supportkonzept bei Fragen zur Software oder Fragen bzgl. ganzer Projekte zu bewerten. Online-Communities sind in diesem Fall für Studierende und Lehrende ideal, da in Foren zum einen Fragen gestellt werden können, zum anderen aus den Fragen und Diskussionen anderer Communitymitglieder Lösungen auf das eigene Problem projiziert werden können. „Auf Anfrage“ bezieht sich erneut auf Anfragen direkt beim Hersteller. Da die Anfrage ggf. mit Kosten verbunden sein kann und meist nur die konkrete Frage beantwortet, wird für diese Art von Supportkonzept zwei Punkte vergeben. Wenn keine Möglichkeit des Supports gegeben ist, werden null Punkte vergeben.

Im Falle einer Beschädigung der Hardware ist schnelle Abhilfe von Vorteilen, damit umgehend weitergearbeitet werden kann. Wenn die Baupläne und -anleitungen im Internet zur Verfügung stehen, Komponenten im Einzelhandel erhältlich und leicht austauschbar sind stellt sich die Reparatur als Problemlos heraus. Wenn die Bauteile und Baupläne nur auf Anfrage verfügbar sind hindert die zusätzliche Kommunikation und damit verbundene Wartezeit mit dem Hersteller ein direktes Reparieren. Jedoch ist diese Art der Reparatur noch ausreichend, um ggf. innerhalb von 2-3 Tagen weiter zu arbeiten. Wenn jedoch die Hardware dem Hersteller zu gesendet werden muss und neben den Bauteilkosten weitere Kosten für die Reparatur anfallen ist das als unbefriedigend einzustufen.

Die Aspekte der Lehre werden von Studierenden, wie Lehrenden im universitären Alltag genutzt, jedoch aus unterschiedlichen Sichtweisen betrachtet. Im Folgenden werden die vier unterschiedlichen Konzepte zum einen durch eine Gruppe von drei Mitarbeitern des Fachgebietes MPM der TU-Berlin bewertet, zum anderen durch eine Gruppe von vier Studierenden aus den Studiengängen Automotive Systems, Elektrotechnik, Informationstechnik im Maschinenwesen und Maschinenbau. Dadurch wird ein breites Spektrum an späteren Anwendern des RP-HyDrives und von RP-Systemen abgedeckt.

3.5.1 Bewertung durch die Lehre

In diesem Abschnitt wird die Bewertung durch das Lehrpersonal beschrieben. Zunächst wurden die Hauptmerkmale in Tabelle 3.11 gegenübergestellt. Daraufhin wurde auf Grundlage von Tabelle 3.11 den Hauptmerkmalen in Tabelle 3.12 Gewichtungen zwischen 10% und 35% zugeteilt und den Einzelmerkmalen jeweils eine Gewichtung vergeben.

Tabelle 3.11: Gewichtungsmatrix der Lehre

Kriterien/Kriterien	Lehre und Weiterbildung	Voraussetzungen/Randbedingungen	Inbetriebnahme/Gebrauch	Problembehandlung	Summe
Lehre und Weiterbildung	x	1	1	1	3
Voraussetzungen/Randbedingungen	0	x	0	1	1
Inbetriebnahme/Gebrauch	1	1	x	1	3
Problembehandlung	0	1	1	x	2

Es wird deutlich, dass die Hauptmerkmale „Lehre und Weiterbildung“ die gleiche Gewichtung von drei Punkten erhalten, wie „Inbetriebnahme und Gebrauch“. Diese zwei Hauptmerkmale hängen für die Lehre durch die Punkte „Erweiterbarkeit Software“ und „Erweiterbarkeit Hardware“ miteinander zusammen. Die Möglichkeit der Erweiterbarkeit bildet eine Grundlage für weiterführende Fachkompetenz. Der Weg zu und die Entwicklung von Erweiterungen durch Studierende fördert neben der Fachkompetenz auch die Methoden- und Sozialkompetenz. Die Digitalkompetenz wird durch die Verwendung von Softwaretools und die Arbeit auf Systemebene des Fahrzeugs gefördert. Das Hauptmerkmal „Problembehandlung“ wird mit zwei Punkten bewertet. Die Einzelmerkmale „Supportkonzept“ und „Reparatur“ werden gleichermaßen gewichtet. Die Lehre benötigt einen Ansprechpartner, um konkrete Probleme schnell und einfach lösen zu können. Ebenso sollten die Reparaturmaßnahmen eigenständig durch Lehrpersonal durchführbar sein, sodass im Schadensfall die Hardware innerhalb von ein bis zwei Tage wieder einsatzbereit ist. Die Bauteile sollten zu diesem Zweck im Einzelhandel bzw. im Internet für die Lehre erwerbbar sein. Die geringste Gewichtung erhält das Hauptmerkmal „Voraussetzungen/Randbedingungen“. Z.B. spielt das Einzelmerkmal „Größe“ für die Lehre nur eine sehr untergeordnete Rolle. Der RP-HyDrive soll für Studierende zu jeder Zeit zugänglich sein und aus dem Grund ist ein Verstauen oder Transport nicht vorgesehen. Der Preis für den RP-HyDrive mit allen Komponenten ist zwar durch die Lehre in dem Hauptmerkmal „Voraussetzungen/Randbedingungen“ am höchsten gewichtet worden, jedoch ist die Bedeutung des Preises gegenüber anderen Merkmalen gering. Eine Anschaffung wie den HyDrive oder den RP-HyDrive muss im Umfeld der Lehre geplant und beantragt werden, um für eine bestimmte Stückzahl an (RP-)HyDrive die Gelder zu erhalten und einen Kurs bzw. Praktikum füllen zu können. Dadurch rückt das Einzelmerkmal (global betrachtet) in den Hintergrund.

Die Methodenkompetenz erhält durch das Lehrpersonal eine Gewichtung von 35%. Wichtig ist in diesem Zusammenhang vor allem der Aspekt verschiedenste Systeme unterschiedlicher Fachrichtungen mit grundlegenden Entwicklungsmethoden entwickeln zu können. Methoden des V-Modells, RP-Systeme etc. gewinnen immer mehr an Bedeutung. Da dadurch die Digitalkompetenz mit der Methodenkompetenz eng verknüpft ist, wird die Digitalkompetenz mit 30% gewichtet. Die Fachkompetenz im Zusammenhang mit Brennstoffzellenhybridfahrzeugen, Prüfstandaufbauten und die Entwicklung eingebetteter Systeme erhalten 20%. Das liegt im wesentlichen daran, dass lediglich grundlegende Informationen der zuvor erwähnten Systeme vermittelt werden können. Viele weitere Informationen sind sehr systemspezifisch und können teilweise nur im Zusammenhang mit dem zu entwickelnden System erlernt werden. Die Sozialkompetenz erhält mit 15% die geringste Gewichtung der Kompetenzfelder. Dabei ist diese Kompetenz auch aus Sicht der Lehre keinesfalls zu vernachlässigen. Das ist auch aus den Lehrveranstaltungen des an dieser Stelle bewertenden Fachgebietes MPM zu entnehmen. Viele der angebotenen Lehrveranstaltungen beinhalten eine Hausaufgabe mit Bearbeitung in Kleingruppen (Beispiel: Konstruktionslehre 3 [29]).

Tabelle 3.12: Bewertungsblatt der Lehre

Hauptmerkmale	G HM	Einzelmerkmale	G EM	G	Konzept 1 HyDrive		Konzept 2 Mega 2560		Konzept 3 ACE-kit		Konzept 4 EDUkit	
					Einzeln (E)	E · G	Einzeln	E · G	Einzeln	E · G	Einzeln	E · G
Lehre und Weiterbildung	0,35	Fachkompetenz	0,2	0,07	3	0,21	3	0,21	4	0,28	4	0,28
		Methodenkompetenz	0,35	0,12	1	0,12	2	0,25	4	0,49	4	0,49
		Sozialkompetenz	0,15	0,05	3	0,16	4	0,21	4	0,21	4	0,21
		Digitalkompetenz	0,3	0,11	2	0,21	4	0,42	4	0,42	4	0,42
Voraussetzungen/ Randbedingungen	0,1	Zugänglichkeit/ Erwerbbarkeit	0,2	0,02	2	0,04	4	0,08	2	0,04	2	0,04
		Preis Hard- und Software	0,5	0,05	1	0,05	3	0,15	0	0,00	2	0,10
		Größe	0,3	0,03	4	0,12	3	0,09	1	0,03	2	0,06
Inbetriebnahme/ Gebrauch	0,35	erste Schritte	0,3	0,11	4	0,42	2	0,21	2	0,21	2	0,21
		benötigtes Zubehör zum Ansteuern einer LED	0,1	0,04	0	0,00	3	0,11	2	0,07	4	0,14
		Erweiterbarkeit Software	0,3	0,11	0	0,00	2	0,21	4	0,42	4	0,42
		Erweiterbarkeit Hardware	0,3	0,11	0	0,00	4	0,42	4	0,42	4	0,42
Problembehandlung	0,2	Supportkonzept	0,5	0,10	2	0,20	4	0,40	2	0,20	2	0,20
		Reparatur	0,5	0,10	0	0,00	4	0,40	0	0,00	4	0,40
Summe	1		4	1		1,53		3,15		2,79		3,39

G-HM = Gewichtung Hauptmerkmal

G-EM = Gewichtung Einzelmerkmal

G = Gesamtgewichtung

Wie zuvor erwähnt ist das Hauptmerkmal „Inbetriebnahme/Gebrauch“ mit „Lehre und Weiterbildung“ gleichermaßen zu 35% gewichtet. Bis dahin wurden lediglich die Einzelmerkmale „Erweiterbarkeit Software“ und „Erweiterbarkeit Hardware“ aus dem Hauptmerkmal „Inbetriebnahme/Gebrauch“ betrachtet. Mit 30% Gewichtung ist das Einzelmerkmal „erste Schritte“ im Umgang mit der Hardware gleichermaßen von Bedeutung, wie die Erweiterungsmöglichkeiten.

3.5.2 Bewertung durch Studierende

Ziel der Untersuchung in diesem Abschnitt ist die Bewertung der unterschiedlichen Konzepte durch die Studierenden. Studierende sind nicht zuletzt Anwender der zu untersuchenden Hardware in Praktika im Umfeld der universitären Lehre. Wie im vorangehenden Kapitel 3.5.1 wird zunächst die Gewichtungsmatrix (s. Tabelle 3.13) ausgefüllt.

Tabelle 3.13: Gewichtungsmatrix der Studierenden

Kriterien/Kriterien	Lehre und Weiterbildung	Voraussetzungen/ Randbedingungen	Inbetriebnahme/ Gebrauch	Problembe- handlung	Summe
Lehre und Weiterbildung	x	1	1	1	3
Voraussetzungen/ Randbedingungen	0	x	1	1	2
Inbetriebnahme/ Gebrauch	1	0	x	1	2
Problembehandlung	1	0	0	x	1

Das Kriterium „Lehre und Weiterbildung“ erhält von der Studierendengruppe mit 35% die größte Gewichtung. Das liegt vor allem daran, dass ein möglichst großer Lernertrag aus einer Unterrichtsveranstaltung erzeugt werden soll. Die Vermittlung von Methodenkompetenz spielt dabei mit 35% die größte Rolle, da dieses Kompetenzfeld in der Regel auf eine Vielzahl von Problemen angewandt werden kann. Der Umgang mit neuer Software, wie auch Möglichkeiten der Datenauswertung am Computer bildet für die Studierenden einen ersten Einblick in die zukünftigen Aufgaben eines Ingenieurs. Aus diesem Grund wir die Digitalkompetenz mit 30% gewichtet.

Für Studierende ist es von Bedeutung die verwendete Hard- und Software ggf. privat erwerben zu können. Aus diesem Grund erhält das Hauptmerkmal „Voraussetzungen/ Randbedingungen“ eine Gewichtung von 25%. An dieser Stelle spielt vor allem der Preis eine große Rolle. Dieser sollte so gering wie möglich sein, damit Studierende das System erwerben können. Darum wird der Preis für Hard- und Software mit 50% gewichtet. Die Größe des Aufbaus erhält eine Gewichtung von 30%, da gegenüber dem Einzelmerkmal „Zugänglichkeit/Erwerbbarkeit“ mit 20% eine höhere Priorität vorliegt. Im Gegensatz zur Lehre mit überwiegend großen Schreibtischen in den Praxisräumen sind Studierende platztechnisch eingeschränkt und wünschen sich ebenso die Möglichkeit mit Kommilitonen und Freunden an den Projekten arbeiten zu können. Zu diesem Zweck wäre sollte das RP-System kompakt und ggf. in einem kleinen Koffer transportierbar sein. Hinsichtlich der „Zugänglichkeit/Erwerbbarkeit“ ist der Erwerb im Einzelhandel oder über eine Webseite wünschenswert.

Tabelle 3.14: Bewertungsblatt der Studierenden

Hauptmerkmale	G HM	Einzelmerkmale	G EM	G	Konzept 1 HyDrive		Konzept 2 Mega 2560		Konzept 3 ACE-kit		Konzept 4 EDUkit	
					Einzeln (E)	E · G	Einzeln	E · G	Einzeln	E · G	Einzeln	E · G
Lehre und Weiterbildung	0,35	Fachkompetenz	0,2	0,07	3	0,21	3	0,21	4	0,28	4	0,28
		Methodenkompetenz	0,35	0,12	1	0,12	2	0,25	4	0,49	4	0,49
		Sozialkompetenz	0,15	0,05	3	0,16	4	0,21	4	0,21	4	0,21
		Digitalkompetenz	0,3	0,11	2	0,21	4	0,42	4	0,42	4	0,42
Voraussetzungen/ Randbedingungen	0,25	Zugänglichkeit/ Erwerbbarkeit	0,2	0,05	2	0,10	4	0,20	2	0,10	2	0,10
		Preis Hard- und Software	0,5	0,13	1	0,13	3	0,38	0	0,00	2	0,25
		Größe	0,3	0,08	4	0,30	3	0,23	1	0,08	2	0,15
Inbetriebnahme/ Gebrauch	0,25	erste Schritte	0,5	0,13	4	0,50	2	0,25	2	0,25	2	0,25
		benötigtes Zubehör zum Ansteuern einer LED	0,1	0,03	0	0,00	3	0,08	2	0,05	4	0,10
		Erweiterbarkeit Software	0,2	0,05	0	0,00	2	0,10	4	0,20	4	0,20
		Erweiterbarkeit Hardware	0,2	0,05	0	0,00	4	0,20	4	0,20	4	0,20
Problembehandlung	0,15	Supportkonzept	0,4	0,06	2	0,12	4	0,24	2	0,12	2	0,12
		Reparatur	0,6	0,09	0	0,00	4	0,36	0	0,00	4	0,36
Summe	1		4	1		1,85		3,11		2,40		3,13

G-HM = Gewichtung Hauptmerkmal

G-EM = Gewichtung Einzelmerkmal

G = Gesamtgewichtung

Mit 25% erhält das Hauptmerkmal „Inbetriebnahme/Gebrauch“ die gleiche Gewichtung, wie das zuvor genannte Hauptmerkmal „Voraussetzungen/ Randbedingungen“. Grund dafür sind insbesondere die „ersten Schritte“. Der Einstieg in ein neues System sollte nur 2-3 Stunden in Anspruch nehmen. Nach dieser Zeit sollte z.B. die erste LED angesteuert worden sein und die grundlegenden Prinzipien der Hard- und Softwareumgebung klar sein. Dabei sollte eine LED zwar mit so wenig wie möglich Bauteilen angesteuert werden können, jedoch ist die Umsetzung der Hardware erfahrungsgemäß eine geringere Herausforderung, als die Inbetriebnahme des Systems. Der Studierendengruppe zufolge führt eine komplizierte Inbetriebnahme zu einer Abneigung gegenüber dem System. Die Erweiterbarkeit von Hard- und Software wird mit jeweils 20% gewichtet.

Bei Anwendungen im privaten Umfeld ist die Erweiterbarkeit von Hard- und Software nur bedingt von Relevanz. Jedoch sollte das RP-System auch für andere Projekte nutzbar sein und wird aus diesem Grund mit jeweils 20% gewichtet. Gern gesehen ist die Einfachheit hinsichtlich Hardwareerweiterungen, die das EDUkit und der Arduino jeweils mitbringen. Es kann auf fertige Shields zurückgegriffen werden und einfache mechatronische Probleme wie die Ansteuerung eines Motors innerhalb kurzer Zeit durchgeführt werden. Die „Problembehandlung“ erhält mit 15% die geringste Gewichtung. Wenn RP-Systeme in der Lehre und Kleingruppen angewandt werden, kann die Gruppe als eine Art Forum genutzt werden. Wird das RP-System außerhalb der Lehre durch Studierende erworben, wäre ein Forum unabdingbar, da auf diesem Weg in erster Linie Hilfestellungen erlangt werden können. Des weiteren kann bei der Recherche zur Problemlösung auf Inspirationen für das laufende oder auch weitere Projekte gestoßen werden. Auf der anderen Seite ist im Schadensfall eine Reparatur- bzw. Bauanleitung eine Essentielle Hilfestellung, um Fehler selbstständig zu lokalisieren und beschädigte Bauteile eigenhändig auszutauschen. Die Reparatur, ebenso, wie die Anschaffung des RP-Systems, sollte kostengünstig und die Bauteile im Einzelhandel oder online durch Studierende erwerbbar sein. Da mit einer beschädigten Hardware keine oder nur bedingt Projekte umgesetzt werden können, mit einem unbefriedigendem Supportkonzept dagegen mehr Zeit in Anspruch genommen wird, spielen die Reparaturmöglichkeiten mit 60% gegenüber dem Supportkonzept mit 40% Gewichtung eine übergeordnete Rolle.

3.5.3 Auswertung der Bewertung

Die Bewertungsdaten aus den Kapiteln 3.5.1 und Kapitel 3.5.2 wurden in Abb. 3.16 dargestellt. Die unterschiedlichen Ergebnisse der vier Konzepte lassen sich klar voneinander unterscheiden. Der Übersicht halber, wurden beide Bewertungen eines Konzeptes nebeneinander dargestellt. Für die Wahl, des im folgenden verwendeten Konzeptes, wird der Durchschnitt aus den Bewertungen der Bewertungsgruppen ermittelt. Dabei werden die Gruppen gleichmäßig gewichtet, da beide Bewertungsgruppen gleichermaßen von einem Umbau des HyDrive betroffen sind.

Die beiden Konzepte EDUkit und Arduino liegen nach Abb. 3.16 sehr nah beieinander. Dagegen ist das Konzept 1, also der HyDrive in seiner aktuellen Form, nach VDI 2225 in Kapitel 3.5 mit einer technischen Wertigkeit von durchschnittlich 0,42 Punkten als unbefriedigend einzustufen. Konzept 3, also die MAB II, ist mit einer technischen Wertigkeit von 0,65 befriedigend und somit unter diesen Kriterien ebenso ungeeignet für den Umbau des HyDrive. Konzept 2, der Arduino, mit durchschnittlich 0,78 Punkten liegt nah an einem sehr gutem Konzept und wird durch das Konzept 4, dem EDUkit, mit 0,82 Punkten übertroffen.

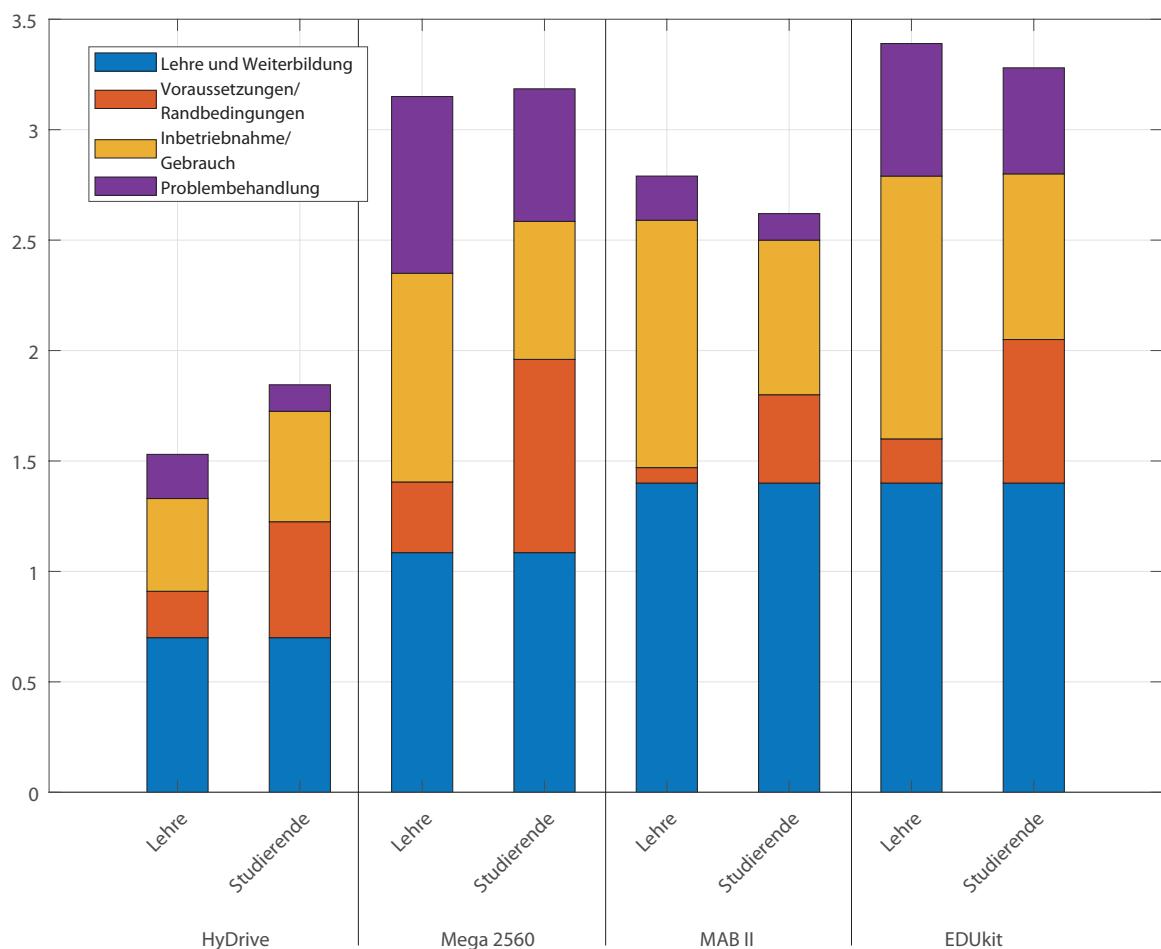


Abbildung 3.16: Grafische Darstellung der Bewertungsdaten von Lehrpersonal und Studierenden

Unter den in dieser Arbeit verwendeten Kriterien zeichnet sich das EDUkit als bestes Konzept aus. Aus diesem Grund wird für den Umbau des HyDrive zum RP-HyDrive das EDUkit verwendet. Im Kapitel „Kritische Betrachtung der Arbeit“ (s. Kapitel 5.3) wird die Bewertung mit einer Gleichgewichtung der Haupt- und Einzelmerkmale diskutiert.

4 Durchführung und Umsetzung

In Kapitel 3.5 stellt sich das ETAS EDUkit als bestes Konzept zur Optimierung des HyDrives heraus. Ziel ist ein einziges Steuergerät zur Bedienung des HyDrives, was zur Folge hat, dass die Controlboards aus dem bisherigen Aufbau ersetzt werden. Zunächst werden in Kapitel 4.1 einzelne Komponenten des HyDrives und deren technische Eigenschaften näher untersucht. Daraufhin folgt eine Darstellung der Schaltungen für die Baugruppen:

Fahrzeug (Fzg)
Tankstelle
Prüfstand (PST)

in Kapitel 4.2. In den Schaltungen wird auf im Handel erhältliche Module zurückgegriffen. Diese sind z.B. als „Motortreiber-Modul“ gekennzeichnet. Aus den Schaltplänen ergeben sich die benötigten Module (s. Kapitel 4.3), um den HyDrive vollständig mit dem EDUkit bedienen zu können. Abschließend wird in Kapitel 4.4 auf die Programmierung der einzelnen Baugruppen eingegangen. Bei der Programmierung werden englische Bezeichnungen verwendet, um Umlaute, wie im Wort „Prüfstand“ zu vermeiden. Die Wörter werden hauptsächlich ausgeschrieben, sodass die Programmierung leserlich bleibt. Ein weiterer Vorteil ergibt sich bei der späteren Handhabung in INCA oder ASCMO aus den ausgeschriebenen Bezeichnungen. Bei der Suche von Variablen kann das englische Wort entsprechend eingegeben werden. Die Entwicklung der Programmierung und Implementierung von Hardware, wie auch Software erfolgt jeweils Komponentenweise. Nach erfolgreichen Komponententests werden die Komponenten nach und nach in das System HyDrive integriert und ebenfalls erneut getestet. Die ASCET-Programmierung kann innerhalb der „Experiment Environment“ vorab getestet werden, bevor die Software auf das EDUkit geflasht wird. Darauf soll in dieser Arbeit nicht eingegangen werden.

4.1 Bestandsaufnahme

Tabelle 4.1: Technischen Daten
der Komponenten des HyDrive [30, S.4]

Baugruppe	Komponente	Spannung [V]	Strom [A]
Fahrzeug	Motor	2	1,8
	Ventilator	5	0,07
	elektrische Last	5	2
	Brennstoffzelle	4,5	
	Kondensator	5	
Tankstelle	Brennstoffzelle	4	2,5
Prüfstand	Motor	2	1,8

In Kapitel 3.1.2 Tabelle 3.1 wurden bereits die benötigten Schnittstellen gelistet. Die Daten zu den einzelnen Komponenten im HyDrive werden in Tabelle 4.1 dargestellt. Seitens EDUkit liegt die maximale Versorgungsspannung von Modulen und Sensoren bei 5 V. Das gilt ebenfalls für die logischen Ausgänge der digitalen Pins. Der Messbereich der Analogeingänge liegt zwischen 0 V und 5 V.

Für die Versorgung der Logik der Module werden die Ausgänge des EDUkit genutzt, ebenso wie die Erdung. Das wird in den folgenden Kapiteln vorausgesetzt und nicht erneut erwähnt.

4.2 Schaltpläne

In den Folgenden Kapiteln werden die Schaltpläne und verwendeten Module erklärt um eine Umsetzung des RP-HyDrive mit dem EDUkit zu realisieren. Der Übersicht halber wird in Abb. 4.1 die Pinbelegung des EDUkits vorweg dargestellt. Im Folgenden wird auf die wichtigsten Funktionen der verwendeten Pins eingegangen.

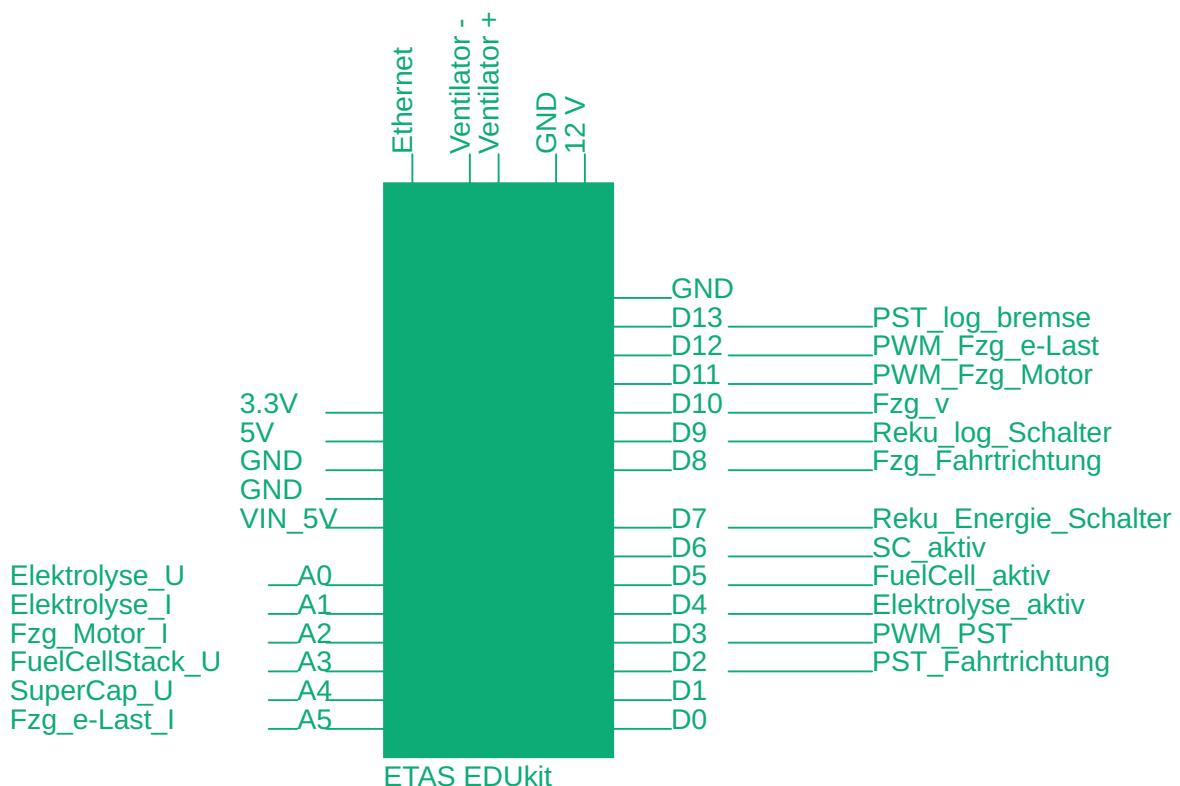


Abbildung 4.1: Pinbelegung des EDUkit

4.2.1 Fahrzeug

Die Komponenten des Fahrzeugs können jeweils Tabelle 4.1 entnommen werden. Der Übersicht halber wird der Ventilator nicht dargestellt. Er wird mit dem PWM des EDUkit angesteuert. Da die H-Brücke vom EDUkit die Versorgungsspannung von 12 V ausgibt, wird der so genannte Arbeitszyklus vom PWM durch die Programmierung beschränkt (s. 4.4.1). Dadurch wird der Mittelwert der Spannung für den Ventilator auf max. 5 V begrenzt. [31, S.308]

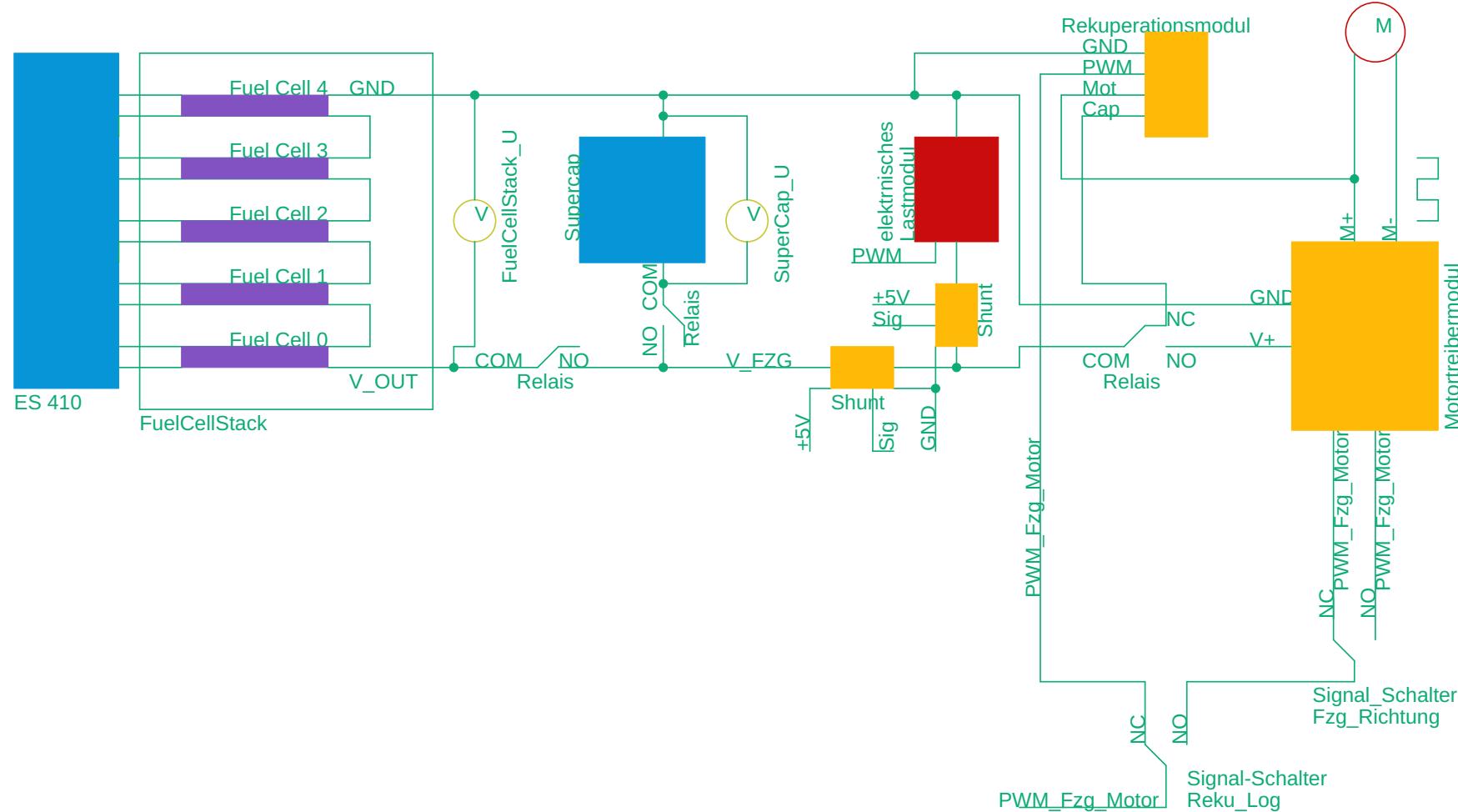


Abbildung 4.2: Schaltplan des Fahrzeugs

Die Spannung des Brennstoffzellenstacks wird vom EDUkit an Pin A3 eingelesen. Die Spannungen der einzelnen Brennstoffzellen werden mit einer ES410 gemessen. Dies entspricht erfahrungsgemäß automotive Standards, wonach in der Testphase Modelle von einzelnen Komponenten erzeugt werden und das Verhalten als Datenmodell im Steuergerät hinterlegt ist. Um das Stack als Energiequelle zu trennen wird ein Relais verwendet und über den Pin D5 angesteuert. Der Pin COM wird als Eingang genutzt. Zur Sicherheit wird der Ausgang NO (Normally Open) verwendet. Im Normalfall ist also das Fahrzeug von der Brennstoffzelle getrennt. Die Spannung kann dennoch jeder Zeit gemessen werden, da die Messstelle vor dem Relais eingebaut werden soll.

Der identische Aufbau wird für den Superkondensator vorgenommen. Die Spannung des Kondensators wird über das EDUkit an Pin A4 eingelesen und das Relais über Pin D6 angesteuert. Um den Stromfluss im Fahrzeug zu messen, wird der im HyDrive bereits verbaute Stromsensor verwendet. Für ein besseres Verständnis wird der Sensor an dieser Stelle dennoch im Schaltplan eingezeichnet. Ein weiteres Relais wird verwendet, um zwischen Rekuperation, also bremsen, und fahren auszuwählen und den Energiefloss entsprechend zu leiten. Die Energie wird dadurch entweder über das Rekuperationsmodul in den Superkondensator gespeist oder aus den Brennstoffzellen bzw. dem Superkondensator in das Motortreibmodul geleitet.

Für die Leistungssteuerung wird ein PWM-Signal ausgehend vom EDUkit Pin D11 verwendet. Damit zwischen den Zuständen „Rekuperieren“ und „Fahren“ geschaltet werden kann, wird ein weiteres Relais verwendet (s. Abb. 4.2 „Signal-Schalter-Reku_Log“). Das PWM-Signal wird über den Eingang COM eingeführt und über den Ausgang NC (Normally connected) in das Rekuperationsmodul geleitet. Damit wird das Fahrzeug im Ruhezustand „Standby“ automatisch elektrisch gebremst. Das Signal wird vom Ausgang NO des zuvor genannten Relais weiter in das Relais „Signal_Schalter_Fzg_Richtung“ geleitet. NC ist dabei vorwärts und NO rückwärts. Von da aus geht das PWM-Signal in das Motortreibmodul. Daran angeschlossen ist schlussendlich der Motor des Fahrzeugs. Der Motor ist zusätzlich mit dem Rekuperationsmodul verbunden und wird darüber im Rekuperationszustand gebremst. Zur Simulation der elektronischen Last wird ein Lastmodul benötigt. Dieses wird über PWM angesteuert und der Stromverbrauch der Last über einen weiteren Shunt am Pin A5 gemessen.

4.2.2 Tankstelle

In Abb. 4.3 ist der Schaltplan der Tankstelle oder auch Wasserstofferzeugung dargestellt. Da nur eine Versorgungsnetzteil mit 12 V zur Verfügung steht und die Brennstoffzelle laut Datenblatt [30, S.4] für die Elektrolyse mit max. 4 V betrieben werden darf, wird ein Spannungswandler für Gleichstrom (DC-DC) verwendet. Der DC-DC wird auf 4 V Ausgangsspannung eingestellt. Damit die Elektrolyse über das EDUkit geschaltet werden kann, wird der Ausgang NO eines Relais verwendet. Dadurch wird verhindert, dass die Elektrolyse dauerhaft Wasserstoff erzeugt, wenn das EDUkit nicht eingeschaltet ist. Das Relais wird über den Pin D5 des EDUkits betätigt. Die Spannung an der Brennstoffzelle wird über den Pin A0 eingelesen und der Stromverbrauch über den Pin A1 mittels Shunt gemessen.

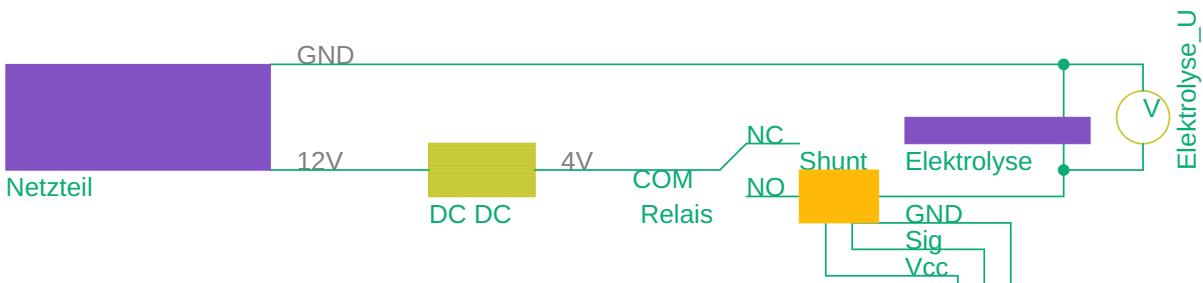


Abbildung 4.3: Schaltplan der Elektrolyse

4.2.3 Prüfstand

Ähnlich wie bei der Versorgung der Elektrolyse wird auch beim Prüfstand ein DC-DC benötigt (s. Tabelle 3.1). Dieser reduziert die Spannung von 12 V auf die Betriebsspannung von 2 V des Prüfstandmotors. Die gesamte Schaltung ist Abb. 4.4 zu entnehmen. Zur Spannungsüberwachung wird die Motorspannung nach dem DC-DC über die ES 410 eingelesen, die der Übersicht halber an der Stelle nicht eingezzeichnet ist. Mit einem Shunt wird der Stromverbrauch ebenfalls über die ES 410 eingelesen. Über ein Motortreibermodul wird der Prüfstandsmotor mit dem Strom versorgt. Um den Prüfstand zu bremsen ist ein elektronisches Lastmodul vorgesehen.

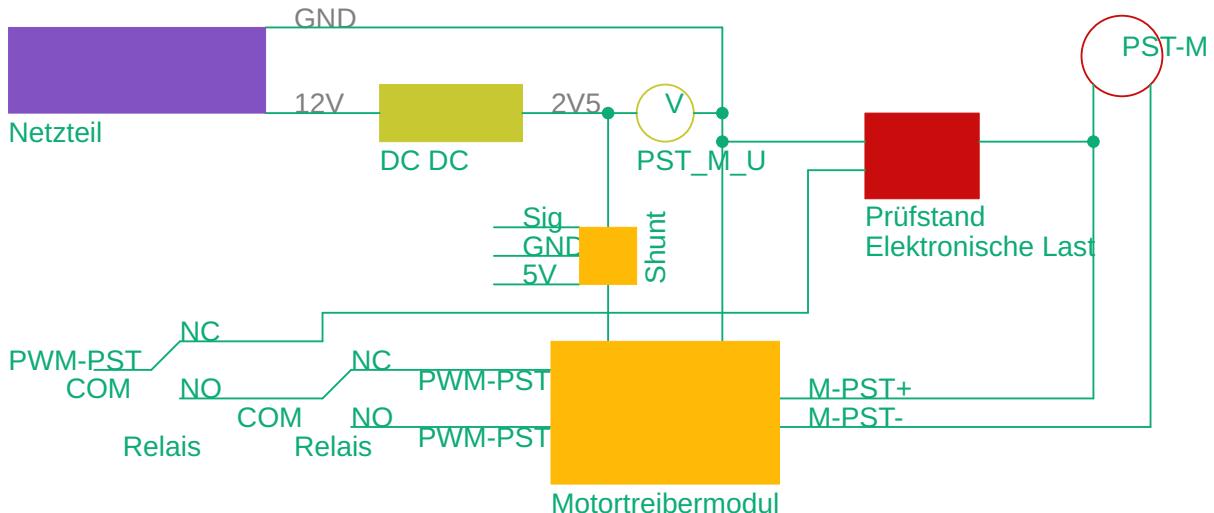


Abbildung 4.4: Schaltplan des Prüfstands

Zur Steuerung des Motortreibermoduls und der elektronischen Last kann wie beim Fahrzeug auf jeweils das gleiche PWM-Signal zurückgegriffen werden. Das PWM-Signal wird von Pin D3 bezogen und zunächst zwischen den Zuständen „Bremse“ und „Schub“ mit einem Relais geschaltet. Das Relais wird über Pin D12 geschaltet. Unter dem Zustand „Schub“ kann mit einem Relais, gesteuert über Pin D2, die Richtung des Prüfstands gewählt werden.

4.3 Zusätzlich benötigte Hardware zum Messen und Verstellen

Die in Kapitel 4.2 aufgeführten Schaltpläne enthalten einzelne Module für bestimmte Aufgaben. Darunter fallen Motortreibermodule, DC-DC und weitere. In den folgenden Kapiteln werden die dafür verwendeten Module kurz erklärt.

4.3.1 Vorgefertigte Module

Um den Prüfstandsmotor und die Elektrolyse mit Spannung aus dem Netzteil zu versorgen wird ein DC-DC verwendet. Der Spannungsregler XL4016 der Firma XLSEMI eignet sich für die Anforderungen bzgl. geforderter Leistung. Dieser ist bei z.B. AZ-Delivery in einem fertigen Modul erhältlich. [32] Es wird pro Baugruppe (Elektrolyse, Prüfstand) ein Spannungswandlermodul verwendet.

Für Fahrzeug und Prüfstand wird jeweils ein Motortreibermodul benötigt, welches den Anforderungen entsprechend Tabelle 3.1 genügt. Damit das Fahrzeug vorwärts, wie auch rückwärts fahren kann, wäre

an dieser Stelle eine H-Brücke geeignet. Problematisch ist jedoch die geringe Spannung der anzutreibenden Motoren. Davon ausgehend, dass min. eine Spannung von 2 V vorliegt findet sich leider kein passendes Modul. Als fertiges Modul steht lediglich ein Modul mit einer H-Brücke des Typen L9110 zur Verfügung (Hier erhältlich: [33]). Dieses Modul benötigt mindestens eine Eingangsspannung von 2,5 V und ist somit zunächst nicht für den Prüfstandmotor geeignet. Für das Fahrzeug kommt dieses Modul jedoch in Frage, da die Spannung des Stacks und des Kondensators oberhalb von 2,5 V liegen. Dabei wird davon ausgegangen, dass eine Zelle im Bereich von ca. 0,7 V bis 0,8 V ihren Betriebsbereich hat und fünf Zellen im Stack verwendet werden. [34, S.16] Für die Motorsteuerung des Prüfstands kann nach Versuchen dennoch auf ein Modul mit der H-Brücke L9110 zurückgegriffen werden. Die Versuche haben gezeigt, dass der Motor sich trotz höherer Spannung von 2,5 V nicht erhitzt. Außerdem lässt sich das PWM auf 80% begrenzen, wodurch die Spannung im Mittel maximal 2 V erreicht. Mit der Verwendung der H-Brücke lassen sich die Motoren jeweils vorwärts, als auch rückwärts drehen.

Zum Schalten der Zustände von Fahrtrichtung, Bremsen, Rekuperation, Elektrolyse und Signalen stehen zum einen das Relais Shield „Relay Shield v3.0“ zum Aufsetzen auf das EDUkit und ein Relais Modul MF-6402393 von MAKERFACTORY zur Verfügung. Die Module enthalten insgesamt acht Relais und somit ausreichend, um die acht Schalter des späteren Aufbaus zu betätigen.

Um das Fahrzeug mittels Rekuperation bremsen zu können, wird ein so genannter „High side switch“ benötigt. Dieser kommt zum Einsatz, wenn die positiven Pole von Energiequelle und Verbraucher miteinander verbunden werden und einen Stromfluss führen. [35] Zum Schalten dieses Stromflusses wurde eine P-Kanal MOSFET-Schaltung aus einem Arduinobeispiel (s. [35]) nachgebaut.

Um die elektrische Last im Fahrzeug umzusetzen wird auf ein MOSFET-Modul (hier erhältlich: [36]) zurückgegriffen. Der Strom soll wie beim HyDrive auf ca. 1 A beschränkt werden. Das ist mit einem PID-Regler und dem PWM-Signal, welches zur Steuerung verwendet wird möglich. Die Ausgänge des Moduls werden über einen Leistungswiderstand verbunden, damit der Strom begrenzt wird. Um über einen möglichst breiten Spannungsbereich der Energiequellen 1 A Strom zu beziehen wird ein Leistungswiderstand von 1Ω zwischen die Pole geschaltet. Dadurch ist die Spannung nach dem ohm'schen Gesetz gleich dem Strom. Mit einer maximalen Spannung ausgehend vom Kondensator 5 V erhält man den maximalen Strom von 5 A. Der Strom soll mit der ASCET-Programmierung und einem PID-Regler auf 1 A beschränkt werden.

Für eine bessere Regelung des MOSFET's wird anstelle des Verbauten IRF 520 MOSFET's das MOSFET IRLZ 44 N verwendet. Dieses MOSFET erfüllt alle Anforderungen an Durchlassspannung und -strom. Der Vorteil des IRLZ 44 N ist laut Datenblatt ein geringerer Widerstand bei kleinerer Schaltspannung und daraus resultierend ein früheres Durchschalten des MOSFETS. Das MOSFET IRLZ 44 N wurde durch ein Arduino-Forum gefunden und anstelle des Verbauten MOSFET's IRF 520 empfohlen. [37]

Für das Bremsen des Prüfstandes kann die umgebaute elektrische Last des Fahrzeugs verwendet werden. Hierbei bremst der Motor in Abhängigkeit des PWM-Signals und damit einhergehend dem Kurzschlussstrom.

Zur Strommessung wird ein Modul mit dem ACS712 Chip verwendet. Das Modul gibt es für verschiedene Strommessbereiche und ist jeweils für eine Betriebsspannung von 5 V ausgelegt. Somit kompatibel mit dem EDUkit. Für die Messstellen in den Schaltplänen (s. Abb. 4.2, Abb. 4.4 und Abb. 4.3) sind gemäß Tabelle 3.1 ± 5 A Messbereich ausreichend (hier erhältlich [38]).

Weitere Informationen zu der hier beschriebenen Hardware sind den jeweiligen Datenblättern zu entnehmen.

4.3.2 ETAS-Hardware

In der Entwicklungsphase von Komponenten werden häufig Systeme mit Zusatzhardware vermessen und aus den ermittelten Daten Modelle gebildet (s. Kapitel 3.5). Deshalb wird auch in diesem Fall auf das Zusatzmodul ES 410 der ETAS GmbH zurückgegriffen, um die einzelnen Spannungen der Brennstoffzellen messen zu können. [26] Außerdem wird die Spannung und der Stromverbrauch des Prüfstandmotors über die ES 410 ermittelt. Um in INCA die Daten Zeitsynchron erfassen zu können wird eine ES592 als Schnittstelle zwischen EDUkit, ES410 und Computer verwendet. [39] Darüber hinaus werden folgende Kabel benötigt:

Stromversorgungskabel, K100

Ethernet-PC-Verbindungskabel, CBE100

Adapterkabel für Ethernetverbindung 1 GBit/s bis 100 MBit/s, CBAE330

Ethernet-Verbindungs- und Stromversorgungskabel, CBEP420.1

4.4 Programmierung

In den folgenden Unterkapiteln wird die Programmierung der einzelnen Baugruppen:

- Fahrzeug (Vehicle)
- Tankstelle (Electrolyser)
- Prüfstand (TestBench)

beschrieben. Abb. 4.5 stellt die oberste Ebene der Programmierung dar.

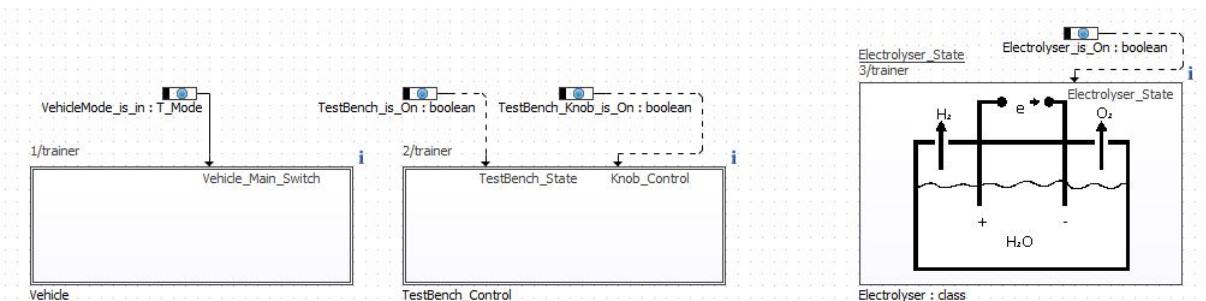


Abbildung 4.5: Programmierung des RP-HyDrive

Der linke Statement Block mit der Bezeichnung „Vehicle“ wird als erstes ausgeführt. Durch den Parameter „VechicleMode_is_in“ wird zwischen den verschiedenen Energiezuständen für das Fahrzeug gewählt. Dazu mehr in Kapitel 4.4.1. An den mittleren Block mit der Bezeichnung „TestBench_Control“ (s. Kapitel 4.4.3) werden die Parameter „TestBench_is_On“ und „TestBench_Knob_is_On“ übergeben. Die Klasse „Electrolyser“ (s. Kapitel 4.4.3) erhält den Parameter „Electrolyser_is_On“. Ein Statement Block fasst ähnlich wie eine Klasse verschiedene Funktionen zusammen. Eine Klasse bietet hinsichtlich Ausführbarkeit und Vorkonfiguration mehr Vorteile gegenüber einem Statement Block.

4.4.1 Fahrzeug

Im Folgenden wird die Programmierung des Fahrzeugs betrachtet. Dabei ist zu beachten, dass die Abarbeitung des Programmcodes durch den Mikrocontroller entsprechend der Zahlen oberhalb der Instanzen erfolgt. Demnach wird zunächst der Statement Block „Vehicle_ Information“, dann „FuelCellMode“,

„SuperCapMode“ und so weiter abgearbeitet. Dem Verständnis halber wird das Programm davon abweichend erklärt. Die Programmierung des Fahrzeugs ist unterteilt in die fünf Energiezustände des Fahrzeugs (s. Abb. 4.6):

Standby	- Keine der Energiequellen ist eingeschaltet
FuelCellMode	- Das Brennstoffzellenstack wird als Energiequelle genutzt
SuperCapMode	- Der Kondensator wird als Energiequelle genutzt
ChargeMode	- Laden des Kondensators mit der Brennstoffzelle
RecuMode	- Laden des Kondensators über Rekuperation

Diese Energiezustände erhalten jeweils den Parameter „VehicleMode“ und geben den Schaltzustand für Kondensator „SC_switch“ und Brennstoffzelle „FC_switc“ aus. Die Schaltzustände der Energiezustände werden in ein logisches „oder“ geführt und das Ergebnis an die jeweilige Kontrollklasse „FuelCell_Control“ und „SuperCap_Control“ übergeben. Zur Ergebniskontrolle wurden die Variablen „FC_is_active“ und „SC_is_active“ zwischen das logische „oder“ und die „*_Control“-Klasse eingefügt. In der Testphase der Programmierung kann damit der Ausgang des logischen „oder“ überprüft werden. Während das Programm in der EE simuliert wird, erscheint je nach Zustand über der Variable ein `true` oder `false`.

Aus dem Block „RecuMode“ wird zusätzlich der Zustand für die Steuerung von Relais für das Umlieiten des PWM-Signals („Recu_Lo_swi“) auf das Rekuperationsmodul und der Stromleitung zwischen Motor und Kondensator („Recu_Po_swi“) ausgegeben. Die Zustände werden an die Klasse „Relais“ übergeben und darin verarbeitet.

Der Block „Power_Control“ erhält ebenfalls den Parameter „VehicleMode“. Hinzu kommen die Parameter „Vehicle_Direction“ und „Vehicle_Engine_Sp“. Mit dem Parameter „Vehicle_Direction“ wird die Drehrichtung des Motors vorgegeben. Es kann zwischen Forward und Reverse gewählt werden. Der Parameter „Vehicle_Engine_Sp“ kann durch den Anwender zwischen 0 und 100% variiert werden. „Sp“ steht dabei für Setpoint. Ausgegeben wird von dem „Power_Control“-Block die Werte „Vehicle_Dir_Swi“ zur Steuerung des PWM-Signals und der Wert „Motor_Control“. Der Wert „Motor_Control“ wird an die Klasse „Motor_Control“ übergeben, welcher darüber das PWM an das System ausgibt. Das PWM-Signal kann über die Zustände der Relais und in Abb. 4.2 nachverfolgt werden. Abschließend sei der Block „Vehicle_Information“ erwähnt. Er berechnet den Ladezustand des Kondensators, die vorliegenden Spannungen und weitere Informationen des Fahrzeugs.

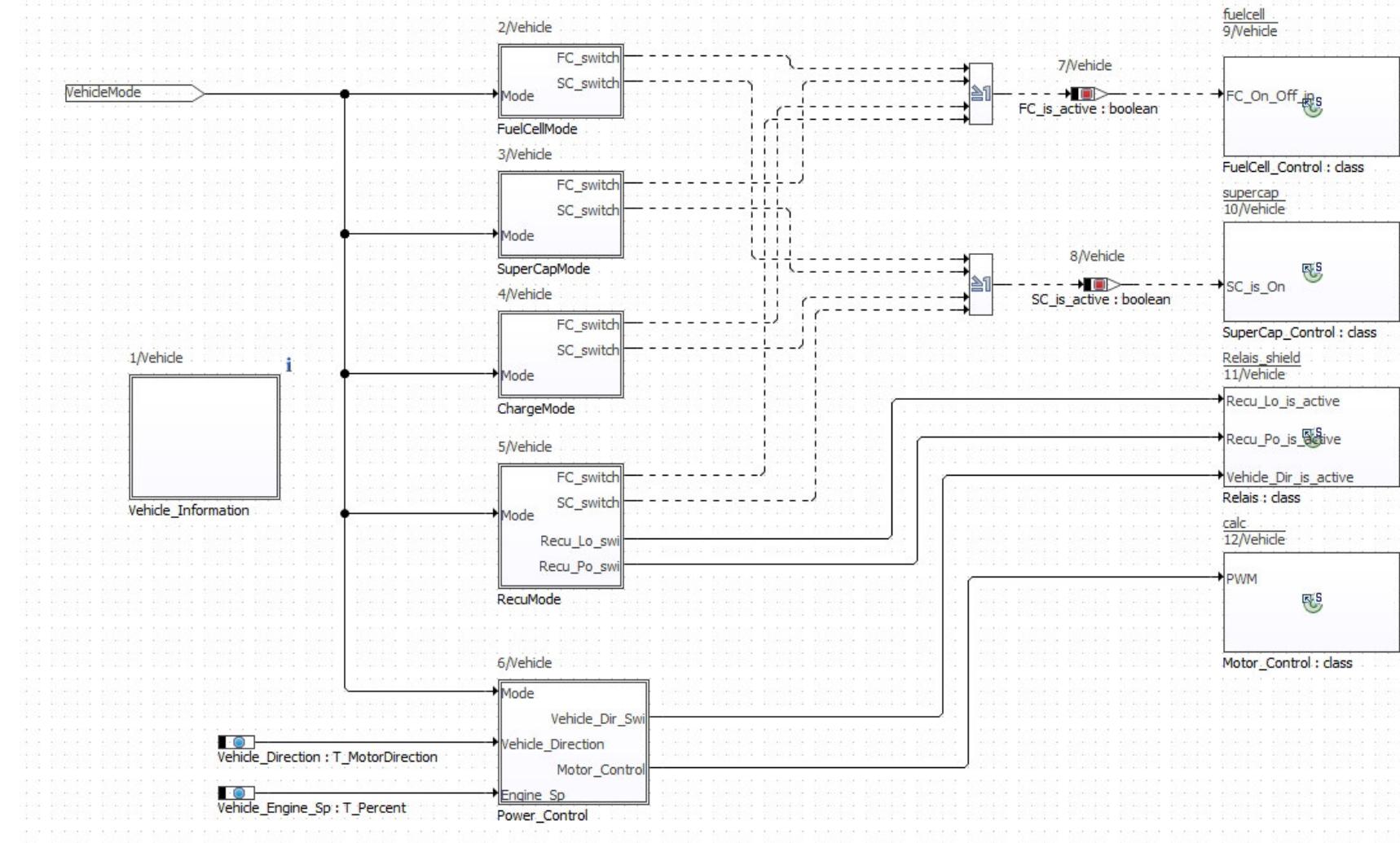


Abbildung 4.6: Programmierung des RP-HyDrive - Fahrzeug

Exemplarisch wird in Abb. 4.7 der Programmablauf einer Zustandsbestimmung für den Brennstoffzellenmodus (FuelCellMode) erläutert.

Im folgenden werden zwei Variablen vom Typ Boolean benötigt. „SC_switch_FC“ ist der Zustand des Kondensatorschalters (SC_switch) des Blockes Brennstoffzelle(_ FC). Und die Variable „FC_switch_FC“ als Zustand des Brennstoffzellenschalters („FC_switch“) im Block Brennstoffzelle(,, FC). Wie zuvor erwähnt, erhält der Statement Block den Fahrzeugmodus (VehicleMode), kurz „Mode“ (links oben in Abb. 4.6) in diesem Fall. Der „Mode“ wird mit dem Literal „FuelCell“ abgeglichen. Aus dem „=“ - Block wird ein Wert vom Typ Boolean ausgegeben und an den „if-else“ - Block übergeben. Die Variable „FCell_Mode“ zwischen den zuvor genannten Operationsblöcken dient für den Nutzer zur Überprüfung des Ergebnisses. Wenn das Ergebnis `true` ist, werden die mit „1/if-then“ gekennzeichneten Aktionen ausgeführt.

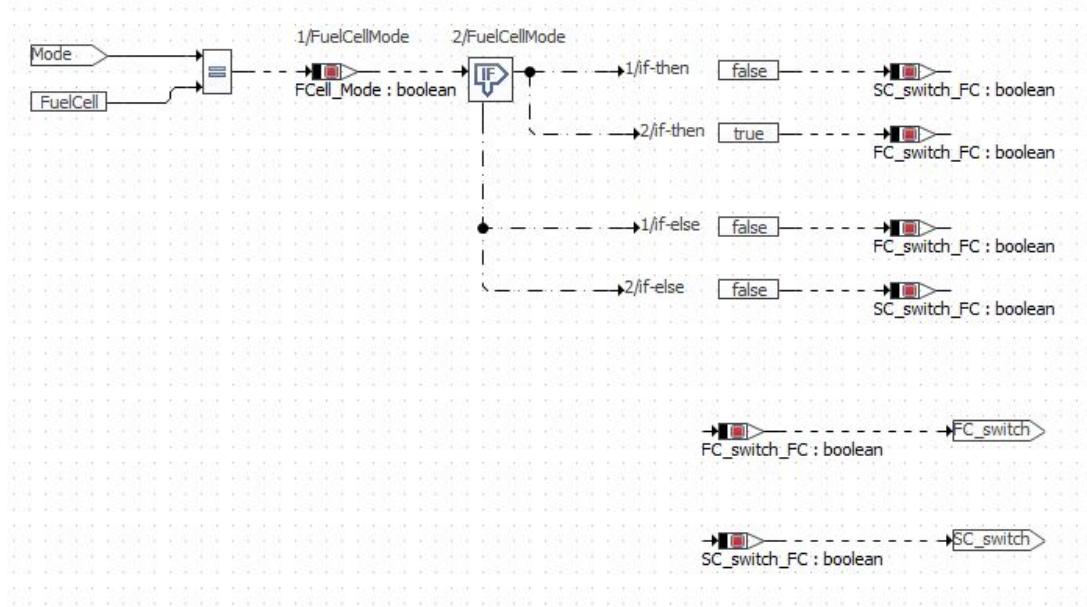


Abbildung 4.7: Programmierung des RP-HyDrive
Zustand der Brennstoffzelle („FuelCellMode“)

Das bedeutet demnach, die Variable „SC_switch_FC“ wird auf `false` und die Variable „FC_switch_FC“ auf `true` gesetzt. Andernfalls werden beide Variablen auf `false` gesetzt. Die Werte der beiden Variablen werden über die Ausgänge „FC_switch“ und „SC_switch“ aus dem Statement Block herausgeführt.

Der Signalfluss ist Abb. 4.6 zu entnehmen. Demnach gibt das logische „Oder“ ein `true` aus, welches an die Klasse „FuelCell_Control“ übergeben wird.

Der Inhalt der Klasse ist in Abb. 4.8 abgebildet. In der Klasse wird die Variable „is_on“ als Kontrollvariable verwendet. Das Relais der Brennstoffzelle wird über die Variable „FuelCell_switch“ betätigt. Als weitere Information über den Zustand der Energieversorgung soll die LED 2 leuchten. Damit erhält der Nutzer ein optisches Signal. Der Zustand der LED wird in der Variable „FuelCell_LED_State“ definiert. Zunächst wird der Pin des Relais und dessen Funktion als `Output`, also `Ausgang` definiert. Wie in Abb. 4.7 wird ein „if-else“ - Block verwendet. Je nach Zustand wird entsprechend das Relais über den Block „Set_FC“ und die LED über den Block „FuelCell_LED_active“ geschaltet.

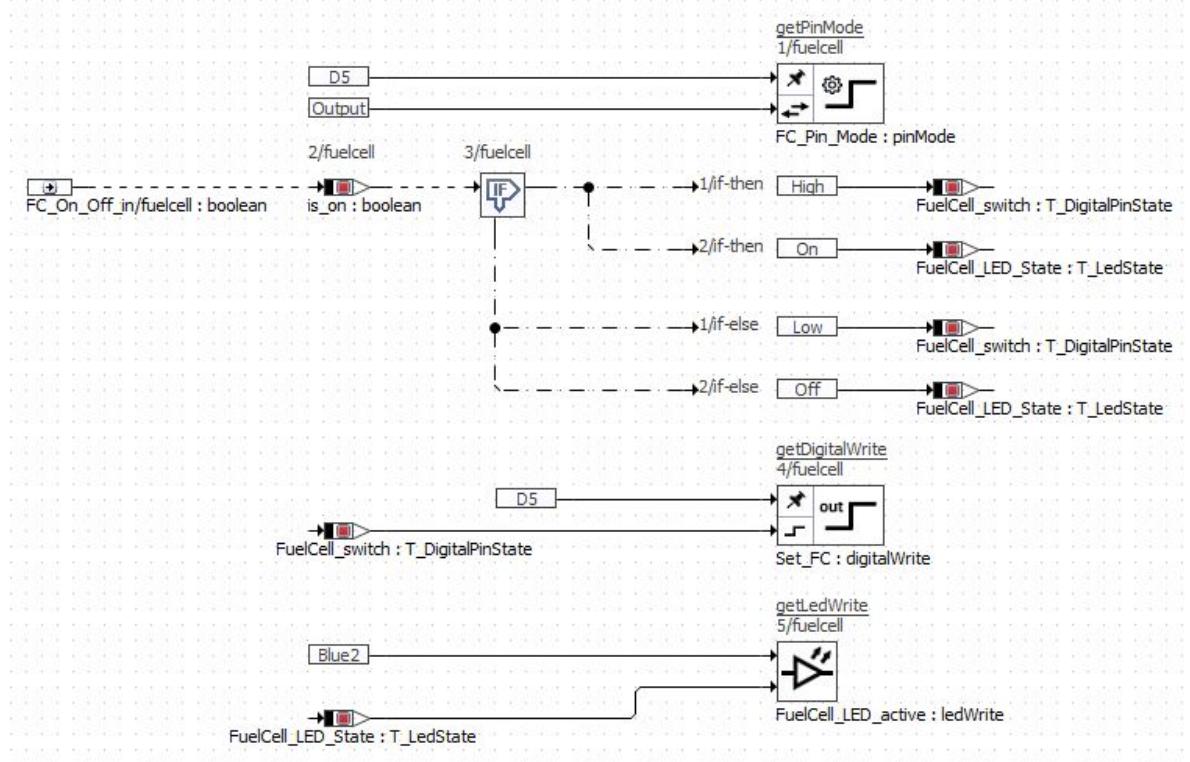


Abbildung 4.8: Programmierung des RP-HyDrive
Schalten eines Relais („FuelCell-Control“)

4.4.2 Tankstelle

Viele Programmteile der Tankstelle in Abb.4.9 sind aus den vorangegangenen Ausschnitten in Abb.4.6, Abb.4.7 und Abb.4.8 bereits bekannt. Darum wird im Folgenden auf die bis dato unerwähnten Programmlemente und der Programmablauf eingegangen.

Der Klasse wird der Wert von „Electrolyser_is_on“ übergeben und je nach Wert werden die Variablen „Electrolyser_Relais_bit“ und „Electrolyser_LED_State“ gesetzt. Dadurch kann das Relais für die Wasserstofferzeugung geschaltet werden und die grüne LED fängt an zu leuchten. Zur Berechnung der benötigten Leistung für die Wasserstofferzeugung werden Strom über den Analogeingang A1 und die Spannung über den Analogeingang A0 gemessen. Die Werte werden mit den Blöcken „analogRead_instance_I“ für den Strom und „analogRead_instance_U“ für die Spannung eingelesen. „*_I“ steht für Strom und „*_U“ für die Spannung. Der Einfachheit halber wird der Pfad der Spannung weiterverfolgt.

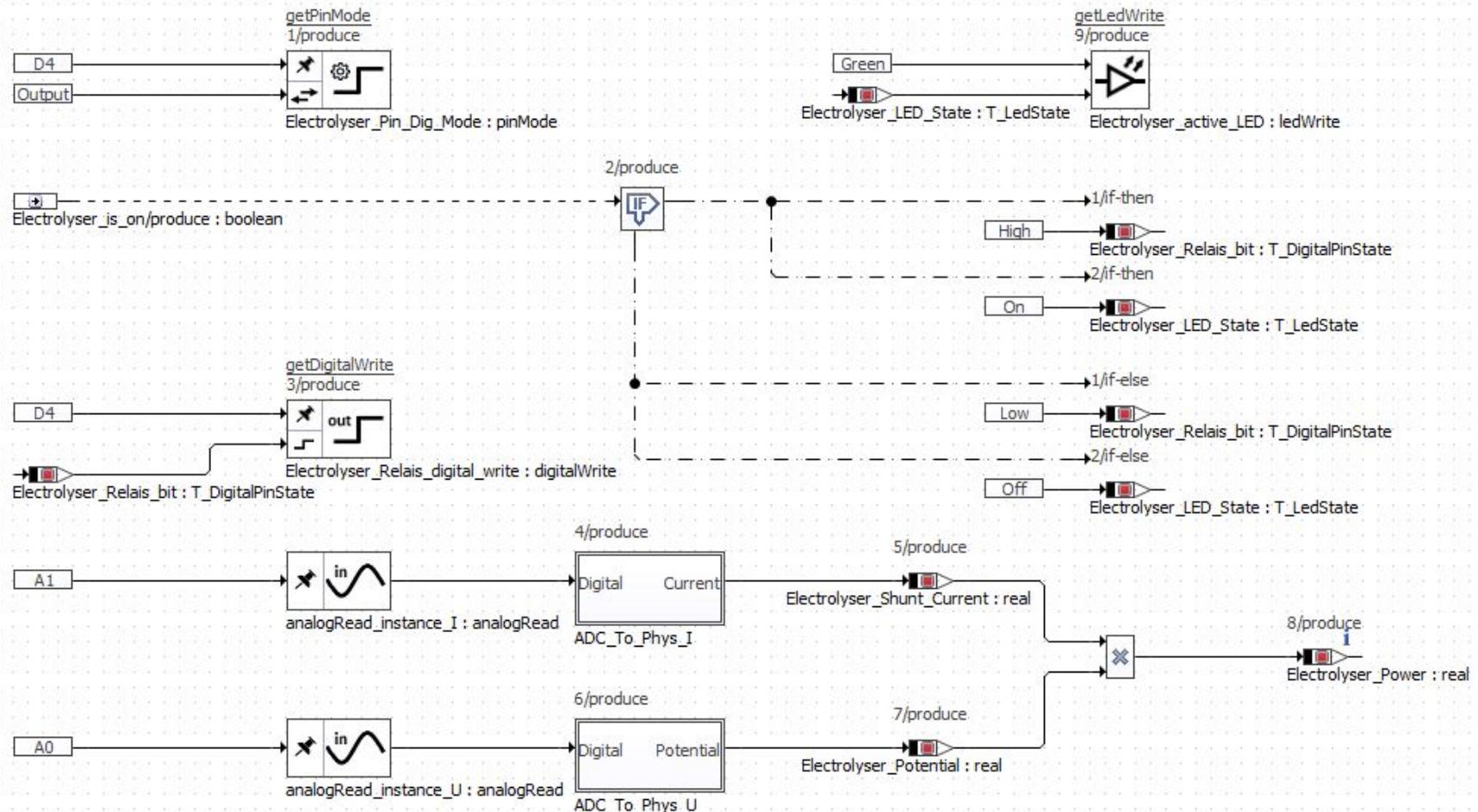


Abbildung 4.9: Programmierung des RP-HyDrive - Tankstelle

Das Ergebnis aus dem Block „analogRead_ instance_ U“ ist ein Integer mit dem Wert der Spannung zwischen 0 und 4095. Der Wert wird an den Statement Block „ADC_ To_ Phys_ U“ übergeben und umgerechnet. Das Ergebnis wird anschließend in die Variable „Electrolyser_ Potential“ geschrieben.

In Abb. 4.10 wird der Integerwert der „analogRead_ instance_ U“ übergeben und in die Variable „Electrolyser_ Potenital_ Dig“ geschrieben. Der darauf folgende Block ist ein „Type Conversion Operator, - Block und wird verwendet, um aus dem Wert einer Variable vom Typ Integer einen Wert vom Typ Real zu erzeugen. Dieser Schritt wird für die Multiplikation mit einem realen Faktor benötigt. Der Faktor berechnet sich aus dem Quotienten von Messbereich (0 V - 5 V) und dem Auflösungsbereich (0 - 4095). Der Wert für „Electrolyser_ faktor_ U“ ist somit $\approx 0,001221$. Das Vorgehen kann ähnlich auf die Berechnung des fließenden Stroms übertragen werden. Jedoch sind dabei die sensorspezifischen Angaben im Datenblatt zu beachten.

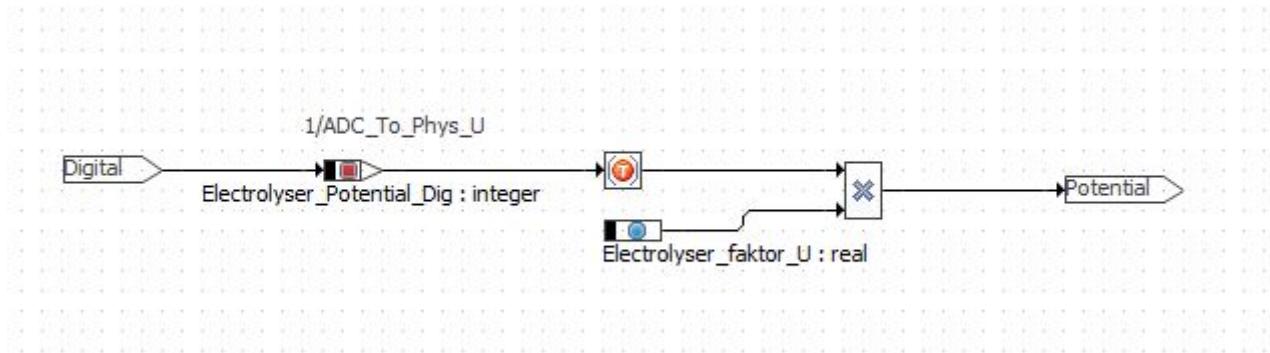


Abbildung 4.10: Programmierung des RP-HyDrive - Umrechnung zum Einlesen eines Analogeinganges

Die Werte für Strom und Spannung können im INCA ausgegeben werden. Abschließend wird mittels Multiplikation der Wert der Leistung in die Variable „Electrolyser_ Power“ geschrieben und kann ebenso in INCA ausgegeben werden.

4.4.3 Prüfstand

Die Programmierung des Prüfstandes wird wie bei der Tankstelle auf die neuen Elemente beschränkt. Zunächst sei jedoch der Programmablauf und die Funktionsweise erklärt.

Der „TestBench_ State“ wird an den Statement Block übergeben. Wenn der Eingang `true` ist, wird der Prüfstand aktiv geschaltet. Das bedeutet, dass über die Variable „TestBench_ Control_ Load“, bzw. über das Drehpotentiometer („knobRead_ instance“) die Leistung in Prozent des Prüfstandes eingestellt werden kann. Außerdem leuchtet die gelbe LED als optisches Signal auf.

Die Auswahl zwischen Potentiometer und manueller Eingabe wird über den Parameter „Knob_ Control“ getroffen. Der Wert des Potentiometers muss wie ein Analogeingang umgerechnet werden. Das Potentiometer deckt einen Winkel von 180° ab und die Ausgabe soll in Prozent erfolgen. Somit ist der Umrechnungsfaktor ≈ 0.0244 .

Zur Berechnung des PWM-Arbeitszyklus wird der Wert des Potentiometers, bzw. des Parameters „TestBench_ Control_ Load“ an den Statement Block „Load_ to_ DutyCycle“ übergeben. Darin findet die Umrechnung des Prozentwertes in den PWM-Arbeitszyklus statt und wird in die Variable „TestBench_ Machine_ PWM_ DutyCycle“ geschrieben. Der Wert der Variable wird unten rechts in Abb. 4.11 an den Block „TestBench_Machine_PWM“ übergeben.

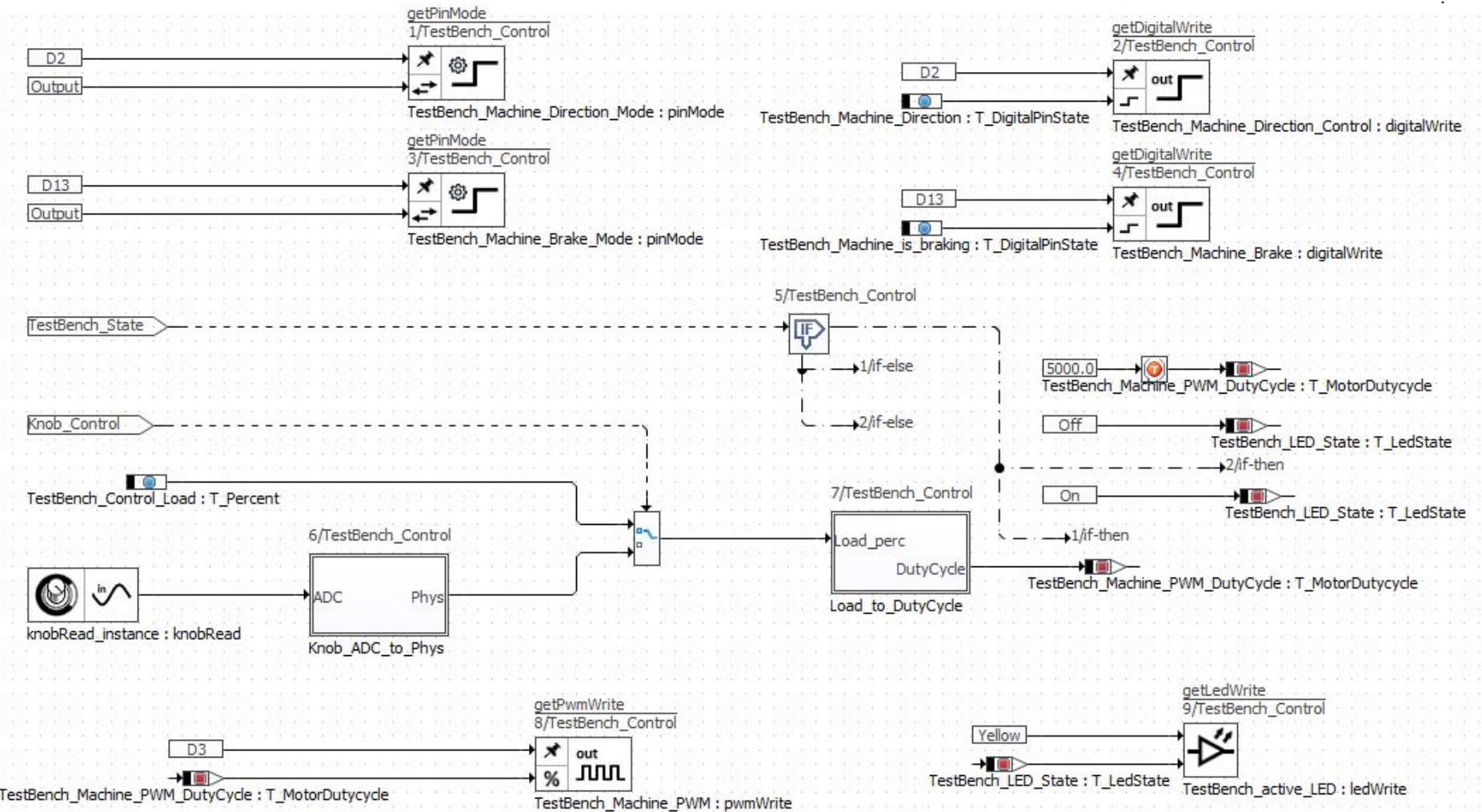


Abbildung 4.11: Programmierung des RP-HyDrive - Prüfstand

Die Programmierung der Berechnung des PWM-Arbeitszyklusses wird in Abb. 4.12 dargestellt. Zunächst erhält der Statement Block den geforderten Prozentwert über den Eingang „Load_ perc“. PWM-Arbeitszyklus darf einen Wert zwischen 0 und 5000 annehmen. 5000 entspricht 0% und 0 einem Wert von 100%. Die Konstante „Load_ Fak“ hat den Wert 50, wird mit dem Eingang „Load_ perc“ multipliziert und in die Variable „Load_ DutyCycle“ geschrieben. Die Konstante „DutyCycle_ Fak“ hat den Wert von 0,8. Dadurch wird die mittlere Spannung des PWM-Signals auf 2 V entsprechend den 80% aus Kapitel 4.3.1 begrenzt. Der maximale Arbeitszyklus (5000) wird vom aktuellen Wert subtrahiert und daraus der Betrag gebildet. Abschließend wird mit dem „Type Conversion Operator“ der Wert in einen Integer umgerechnet und über den Ausgang „DutyCycle“ ausgegeben.

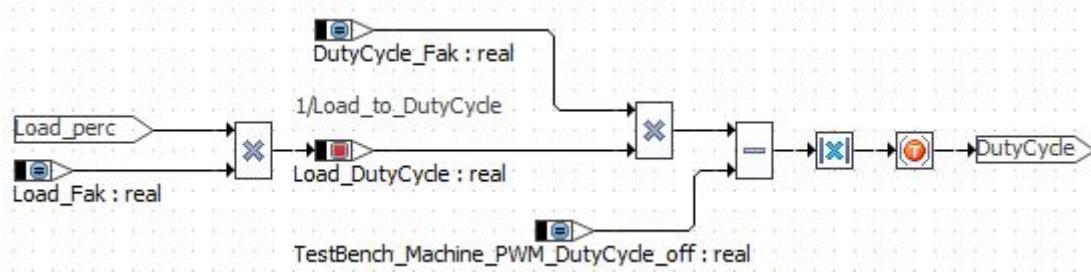


Abbildung 4.12: Programmierung des RP-HyDrive -
Umrechnung Prozentwert in PWM-Arbeitszyklus

5 Auswertung

5.1 Bewertung des RP-HyDrive

Mit der Bewertung des neuen Aufbaus soll die Erfüllung der Haupt- und Einzelmerkmale der Bewertung evaluiert werden. Der RP-HyDrive auf Basis des EDUkits bietet in seinem momentanen Zustand die Möglichkeit den aktuellen Aufbau mit Funktionen zu ergänzen und weiter zu entwickeln. Aufgrund von Zeitmangel, sowie einem fehlenden PWM-Signal lässt sich die elektrische Last im Fahrzeug zum Zeitpunkt der Abgabe nicht bedienen. Die elektrische Last des Prüfstandes, wie auch die ETAS Hardware Peripherie konnten aus Zeitgründen nicht getestet werden. Außerdem konnten der Raddrehzahlsensor, welcher im HyDrive verbaut ist, nicht vermessen und Programmiert werden. Es fehlen die Informationen über die Art und Auflösung des Sensors um fortzufahren. Bis auf die zuvor genannten Mankos erfüllt der RP-HyDrive die Funktionen des HyDrive. In den weiteren Schritten kann die Implementierung einzelner Komponenten, wie den elektrischen Lasten, entlang des V-Modells gelehrt werden, sowie der Umgang mit RP-Hardware bei der Inbetriebnahme der ETAS Hardware vermittelt werden. Nach der Fertigstellung können durch Studierende und Lehrpersonal weitere Funktionen hinsichtlich Hybridtechnik, Betankung und vielen weiteren Aspekten entwickelt werden. Die Funktionsentwicklung inkl. Hardware kann in interdisziplinären Kleingruppen innerhalb einer Lehrveranstaltung vorgenommen werden. Damit vermittelt der RP-HyDrive sehr gut die Kompetenzen des Hauptmerkmals „Lehre und Weiterbildung“. Weiterführend lassen sich mit den Inhalten des EDUkit - Softwarepaketes die Methoden der Testautomatisierung (INCA-Flow) und Modellbildung (ASCMO) lehren und somit die Vermittlung der Kompetenzen noch weiter ausbauen.

5.2 Herausforderungen während der Arbeit

Dieses Kapitel soll Anreize zur Optimierung des EDUkit schaffen, aber auch klarstellen, aus welchem Grund gewisse Baugruppen nicht vollständig getestet werden konnten und damit nicht in das System integriert wurden.

Zunächst wird auf den Umgang mit dem EDUkit eingegangen. Die ersten Schritte in der Programmierumgebung ASCET waren zunächst überwältigend. Diese Erfahrung ist bei industriell genutzten Programmen jedoch aufgrund der zahlreichen Möglichkeiten normal. Im späteren Verlauf der Arbeit wurden von der ETAS GmbH für diese Arbeit Videos bereitgestellt. Die Videos erleichtern den Einstieg hinsichtlich dem Umgang mit der Entwicklungsumgebung in ASCET, sowie dem EDUkit und der Programmierweise. Das zeigt, dass das Produkt ETAS EDUkit derzeit weiterentwickelt. Dennoch wäre eine Übersicht zum Programm ASCET und gängige Beispiele sehr hilfreich gewesen. Mit dem Ziel das verwendete RP-System kritisch zu betrachten und dessen Möglichkeiten hervorzuheben war der Aufbau eines weiteren PWM-Signals in ASCET für einen Maschinenbaustudenten in kurzer Zeit nur schwer umsetzbar. Aus diesem Grund existiert im aktuellen RP-HyDrive keine elektronische Last. Denkbar wäre eine Ergänzung dieser Funktion durch Studierende, z.B. innerhalb einer Hausarbeit. Dabei stellt sich, mit der grafischen Programmierumgebung, ein wesentlicher Vorteil gegenüber dem Arduino heraus. Die Einarbeitung in das System, wie auch die Erweiterung kann ohne Programmierkenntnisse erfolgen. Komplexe Systeme können dadurch besser vermittelt werden.

Positiv aufgefallen ist die einfache Inbetriebnahme ähnlich dem Arduino. Selbst mit ein paar Stichpunk-

ten gelang es das EDUkit einzurichten. Dagegen war die Inbetriebnahme der MAB II deutlich umständlicher. Ein fehlender Compiler, Systemumgebungsvariablen und die manuelle Initialisierung im Control-Desk erforderten Zeit und Geduld. Möglicherweise ist der fehlende Compiler und die Systemumgebungsvariablen ein Einzelfall, jedoch bietet die Lizenzierung des EDUkit weiter Vorteile gegenüber der MAB II. Die EDUkit-Lizenzen sind an die MAC-Adressen des Computers gebunden und erfordern lediglich einen Lizenzschlüssel in Form einer .lic-Datei. Für die Verwendung der MAB II wird ein USB-Dongle benötigt, der auf dem Postweg versendet wird. Zusätzlich erhält der Anwender eine Excel-Datei per E-Mail, die die Lizenzschlüssel enthält. Die weitere Arbeit (s. Kapitel 3.3) mit der MAB II war sichtbar simpel. Da das System auf MATLAB/Simulink basiert, kann bei MATLAB-Fragen auf zahlreiche Foren zurückgegriffen werden. Bei MAB II-spezifischen Fragen, musste jedoch auf die Handbücher zurückgegriffen werden. Ein Forum für das EDUkit wäre definitiv von Vorteilen und würde bei Problemen gut aushelfen.

Weitere Herausforderungen beziehen sich explizit auf den Umbau des HyDrive. Ziel war die Umsetzung mittels erwerbbaren Modulen durchzuführen, sodass der Aufbau durch Studierende und Lehrpersonal nachzubauen ist. Dadurch wurde der Aufbau und die Kabelführung unübersichtlich. Ein weiteres Problem stellten die niedrigen Spannungen und zugleich hohen Ströme der Motoren dar, wie in Kapitel 4.3.1 nachzulesen ist. Ohne eine Eigenanfertigung konnte die Rekuperationsfunktion jedoch nicht auskommen. So konnte auf einen Beitrag bezogen auf einen Arduino zurückgegriffen und die Rekuperation umgesetzt werden.

5.3 Kritische Betrachtung der Arbeit

Die in dieser Arbeit vorgenommene Bewertung erfolgte durch Lehrpersonal und Studierende, die im späteren Berufsleben, während Forschungsprojekten oder bereits während des Studiums mit eingebetteten Systemen konfrontiert werden. Aufgrund von persönlicher Befangenheiten in der Bewertung durch das Lehrpersonal und den Studierenden soll in diesem Abschnitt eine Gleichgewichtung der Haupt- und Einzelmerkmale vorgenommen werden. Ziel ist der Vergleich zwischen den unterschiedlichen Gewichtungen und die Betrachtung von weiteren Bewertungsmöglichkeiten. Für einen besseren Vergleich werden die Daten grafisch in Abb. 5.1 dargestellt.

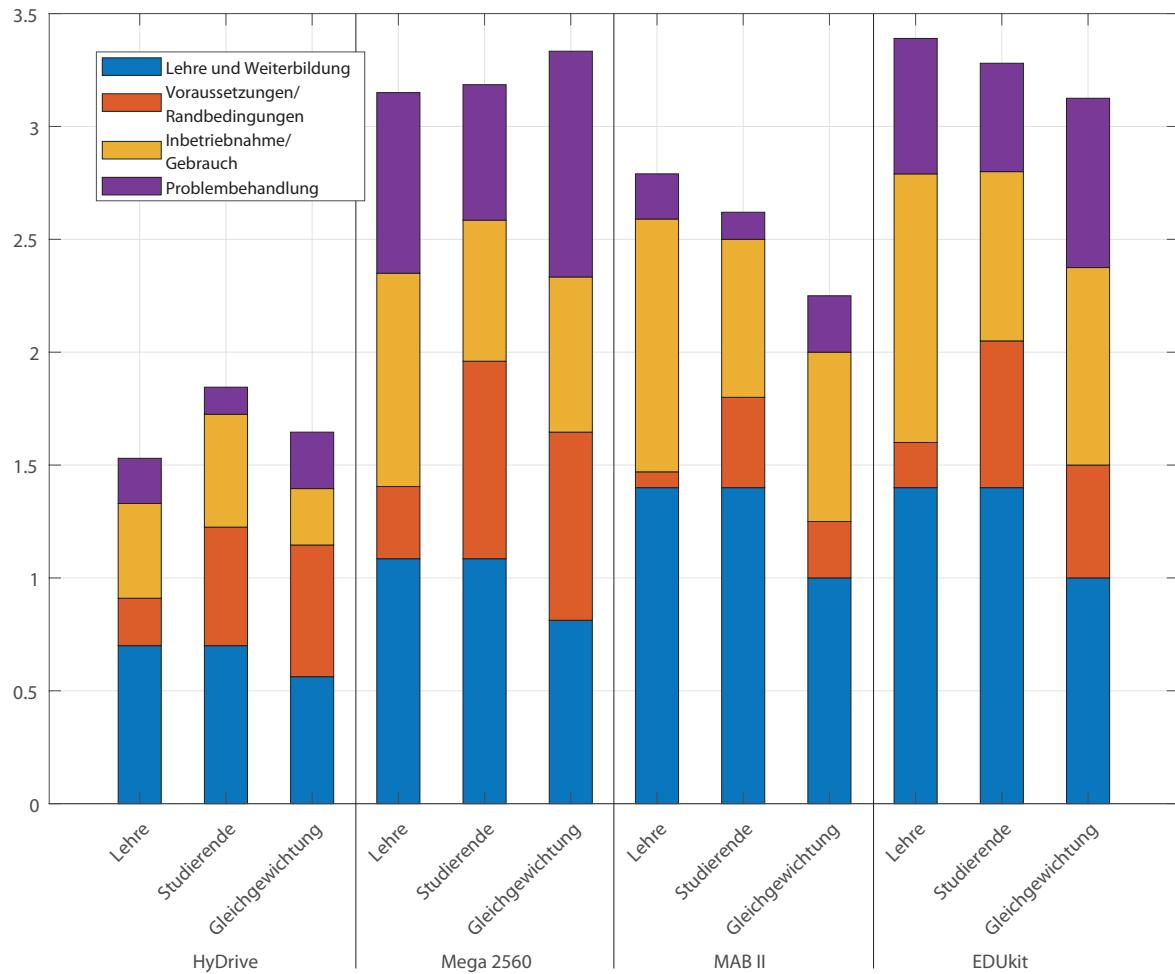


Abbildung 5.1: Grafische Darstellung der Bewertungsdaten

Dabei ist jeweils der Graph mit der Bezeichnung „* - Gleichgewichtung“ zu betrachten. In diesem Fall wird ersichtlich, dass der Arduino Mega 2560 mit 5%-Punkten Vorsprung gegenüber dem EDUkit am besten Abschneidet. Das gilt für die in dieser Arbeit betrachteten Haupt- und Einzelmerkmale. Unter Betrachtung der einzelnen Graphen jedes Konzeptes, so liegt zwischen den verschiedenen Bewertungsgruppen und der Gleichverteilung eine Differenz von max. 0,54 Punkte. Das entspricht einem Unterschied von 13,5% bei der MAB II und stellt das Konzept mit 56% bereits als unbefriedigend dar. Diese Analyse soll die Abhängigkeit des Ergebnisses von den bewertenden Personen und den damit entstehenden systematischen Fehler zeigen. Jedoch ist auch eine gewisse Subjektivität im Entwicklungsprozess von Relevanz. Dadurch werden Merkmale hervorgehoben und es lassen sich Prioritäten der bewertenden Gruppen ableiten. Daran kann die Weiterentwicklung der Produkte bei Bedarf anknüpfen und neue Innovationen hervorbringen. Die bewerteten Produkte werden in Zukunft vermutlich weiterentwickelt und aus diesem Grund ist es, aus Sicht des Autors, durchaus Sinnvoll weitere Bewertungen durchzuführen. Daraus könnte sich ergeben, dass eine alternative zum EDUkit gefunden wird oder die Eignung des EDUkits hinsichtlich der Kriterien gestiegen ist. Unabhängig von dem Entwicklungsstand der bewerteten Produkte können mit den selben Produkten weitere Bewertungen durchgeführt werden, die sich mit den einzelnen Hauptmerkmalen im Detail auseinandersetzen.

5.4 Ausblick

Aus den in dieser Arbeit bewerteten Systemen hebt sich das ETAS EDUkit hervor. Mit der Umsetzung hat sich bereits hard- und softwarespezifisches Optimierungspotential in Kapitel 5.2 herausgestellt. In Zukunft kann neben den offenen Punkten in Kapitel 5.1 die Programmierung hinsichtlich der Struktur optimiert werden. Anstelle der Statement Blöcke könnten verschiedene Klassen verwendet werden mit fallabhängiger Ausführung. Das wäre eine Programmoptimierung, die gleichzeitig den Prozessor entlasten würde. Der RP-HyDrive bietet auch auf Hardwareebene weitere Ausbaumöglichkeiten. So könnte anstelle der verschiedenen Hardwaremodule ein einzelnes Modul mit den aktuellen Funktionen entwickelt werden, welches in den RP-HyDrive integriert oder auf das EDUkit aufgesetzt wird. Die Programmoptimierung kann aus Sicht des Autors bereits durch Studierende in Kleingruppen durchgeführt werden. Zu diesem Zweck sollte ein Lehrkonzept gestaltet werden, mit dem die Studierenden zunächst an den RP-HyDrive herangeführt werden und Schwachstellen, sowie Optimierungspotential ausfindig machen. Für die Optimierung und Bearbeitung des Systems ist eine Einführung in die Software ASCET für die Studierenden empfehlenswert. Im weiteren Verlauf sind Einführungen in die Software INCA, INCA-Flow und ASCMO je nach Umfang der Lehrveranstaltung und Aufgabe der Studierenden empfehlenswert. Zusammengefasst bietet der RP-HyDrive das Potential die Grundlagen der Brennstoffzellenhybridtechnologie zu vermitteln, sowie weiterführen die Aspekte der Entwicklung nach Automotivestandards zu zeigen.

Literatur

- [1] S. R. Sorko und W. Irsa, Hrsg., *Interaktive Lehre des Ingenieursstudiums: Technische Inhalte handlungsorientiert unterrichten*. Berlin: Springer Vieweg, 2019, ISBN: 978-3-662-56223-9.
- [2] Deutsches Institut für Normung e.V., *DIN 69901-5: Projektmanagement: Projektmanagementsysteme Teil 5: Begriffe*, DIN 69901-5, Berlin: Beuth Verlag GmbH, 2009-01-00. (besucht am 04.09.2020).
- [3] J. Feldhusen und K.-H. Grote, *Pahl/Beitz Konstruktionslehre*, 8. Aufl. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, ISBN: 978-3-642-29568-3. DOI: 10.1007/978-3-642-29569-0.
- [4] VDI Verein Deutscher Ingenieure, *VDI 2225 Blatt 3: Konstruktionsmethodik - Technisch-wirtschaftliches Konstruieren - Technisch-wirtschaftliche Bewertung*, VDI 2225 Blatt 3, Berlin: Beuth Verlag GmbH, 1998-11-01. (besucht am 15.09.2020).
- [5] ———, *VDI 2206: Entwicklungsmethodik für mechatronische Systeme*, VDI 2206, Berlin: Beuth Verlag GmbH, 2004-06-01. (besucht am 05.09.2020).
- [6] R. Gessler, *Entwicklung Eingegebetteter Systeme*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014, ISBN: 978-3-8348-1317-6. DOI: 10.1007/978-3-8348-2080-8.
- [7] ETAS GmbH. (o. J.). “Rapid Prototyping.” ETAS GmbH, Hrsg., Adresse: https://www.etas.com/de/anwendungen/applications_software_engineering-rapid_prototyping.php (besucht am 08.09.2020).
- [8] B. Reusch, *Rechnergestützter Entwurf und Architektur mikroelektronischer Systeme: GME/GI/ITG-Fachtagung Dortmund, 1./2. Oktober 1990*, Ser. Informatik-Fachberichte. Berlin und Heidelberg: Springer, 1990, Bd. 255, ISBN: 978-3-642-84304-4. DOI: 10.1007/978-3-642-84304-4.
- [9] R. Marenbach, J. Jäger und D. Nelles, *Elektrische Energietechnik: Grundlagen, Energieversorgung, Antriebe und Leistungselektronik*, 3., aktualisierte Auflage. Wiesbaden: Springer Vieweg, 2020, ISBN: 978-3-658-29491-5. DOI: 10.1007/978-3-658-29492-2.
- [10] H. Tschöke, P. Gutzmer und T. Pfund, *Elektrifizierung des Antriebsstrangs: Grundlagen - vom Mikro-Hybrid zum vollelektrischen Antrieb*, Ser. ATZ/MTZ-Fachbuch. 2019, ISBN: 978-3-662-60355-0. DOI: 10.1007/978-3-662-60356-7.
- [11] M. Klell, H. Eichlseder und A. Trattner, *Wasserstoff in der Fahrzeugtechnik*. Wiesbaden: Springer Fachmedien Wiesbaden, 2018, ISBN: 978-3-658-20446-4. DOI: 10.1007/978-3-658-20447-1.
- [12] Heliocentris Academia. (o. J.). “HyDrive – Electric Vehicle Trainer.” Heliocentris Academia, Hrsg., Adresse: <http://heliocentrisacademia.com/portfolio-item/hydrive-electric-vehicle-trainer/> (besucht am 12.09.2020).
- [13] B. Breuer und K. H. Bill, *Bremsenhandbuch*. Wiesbaden: Vieweg+Teubner Verlag, 2012, ISBN: 978-3-8348-1796-9. DOI: 10.1007/978-3-8348-2225-3.
- [14] Heliocentris Academia, *HyDrive - Electric Vehicle Trainer: Experiment Guide*, Heliocentris Academia, Hrsg., 2019.
- [15] Arduino. (o. J.). “Arduino Mega 2560 Rev3.” Arduino, Hrsg., Adresse: <https://store.arduino.cc/arduino-mega-2560-rev3> (besucht am 10.09.2020).

-
- [16] Stefan Draeger. (o. J.). "Arduino Lektion 83: Stromversorgung." Draeger-IT, Hrsg., Adresse: <https://draeger-it.blog/arduino-lektion-83-stromversorgung/?cn-reloaded=1> (besucht am 27.06.2020).
 - [17] Arduino-projekte.info, Hrsg. (28.09.2017). "Stromverbrauch Arduino & Wemos Boards," Adresse: <https://arduino-projekte.info/stromverbrauch-arduino-wemos-boards/> (besucht am 12.09.2020).
 - [18] Arduino. (o. J.). "Software." Arduino, Hrsg., Adresse: <https://www.arduino.cc/en/Main/Software> (besucht am 15.06.2020).
 - [19] dSpace, *Hochschulkampagnen: Exklusive Angebote für Hochschulen, 2019/2020*. Adresse: https://www.dspace.com/de/gmb/home/medien/brochures/ace_kit.cfm.
 - [20] Tobias Birell. (2012). "Rapid Prototyping Systems In Wärtsilä Laboratory," Adresse: <https://www.semanticscholar.org/paper/Rapid-Prototyping-Systems-In-W%C3%A4rtsil%C3%A4-Laboratory-Birell/d34899af651d48e2285f3c7e6a194a1d26fed938> (besucht am 13.09.2020).
 - [21] dSpace, *MicroAutoBox II*, 8/2020.
 - [22] ———, (o. J.). "MicroAutoBox Break-Out Box." dSpace, Hrsg., Adresse: https://www.dspace.com/de/gmb/home/products/hw/micautob/microautobox2/mabx_breakoutbox.cfm (besucht am 24.09.2020).
 - [23] ———, (o. J.). "Real-Time Interface (RTI): Implementierungssoftware zur Ausführung von Modellen auf dSPACE Hardware." dSpace, Hrsg., Adresse: <https://www.dspace.com/de/gmb/home/products/sw/impsw/real-time-interface.cfm> (besucht am 13.09.2020).
 - [24] ———, (o. J.). "SystemDesk: Modellieren von Systemarchitekturen und Generieren virtueller Steuergeräte." dSpace, Hrsg., Adresse: https://www.dspace.com/de/gmb/home/products/sw/system_architecture_software/systemdesk.cfm#143_25611 (besucht am 13.09.2020).
 - [25] ———, (o. J.). "ControlDesk: Universelle modulare Experiment- und Instrumentiersoftware für die Entwicklung elektronischer Steuergeräte." dSpace, Hrsg., Adresse: <https://www.dspace.com/de/gmb/home/products/sw/experimentandvisualization/controldesk.cfm> (besucht am 13.09.2020).
 - [26] ETAS GmbH. (o. J.). "ES400 – Messmodule." ETAS GmbH, Hrsg., Adresse: https://www.etas.com/de/portfolio/compact_es400_measurement_modules.php (besucht am 16.09.2020).
 - [27] Daniel Ziegler. (20.07.2017). "Taster am Arduino - LED Schalten." Daniel Ziegler, Hrsg., Adresse: <https://daniel-ziegler.com/arduino/mikrocontroller/2017/07/02/arduino-taster/> (besucht am 24.09.2020).
 - [28] Alex. (13.05.2019). "Pull-Up vs. Pull-Down Widerstand." AEQ-WEB, Hrsg., Adresse: <https://www.aeq-web.com/pull-up-vs-pull-down-resistor-micro-controller/> (besucht am 24.09.2020).
 - [29] Dr.-Ing. Tu-Anh Fay. (10.04.2020). "Lehrinhalte Konstruktionslehre 3." TU-Berlin, Hrsg., Adresse: https://www.konstruktionslehre.tu-berlin.de/menue/konstruktionslehre_3/ (besucht am 07.10.2020).
 - [30] Heliocentris Academia, *HyDrive – Electric Vehicle Trainer: Experiment Set for Teaching Fuel Cell Electric Vehicle Technology*, Heliocentris Academia, Hrsg.

- [31] J. Specovius, *Grundkurs Leistungselektronik: Bauelemente, Schaltungen und Systeme*, 9., überarbeitete und aktualisierte Auflage, Ser. Lehrbuch. Wiesbaden: Springer Vieweg, 2018, ISBN: 978-3-658-21169-1. DOI: 10.1007/978-3-658-21169-1.
- [32] AZ-delivery. (o. J.). “XL4016 Step-Down Buck Converter DC-DC.” AZ-delivery, Hrsg., Adresse: <https://www.az-delivery.de/products/xl4016-step-down-buck-converter-dc-dc> (besucht am 04. 10. 2020).
- [33] SEGOR-electronics GmbH. (o. J.). “L9110-Modul: 2-Kanal MotorTreibermodul: 2,5..12V je 0,8A.” SEGOR-electronics GmbH, Hrsg., Adresse: <https://www.segor.de/#Q=L9110-Modul&M=1> (besucht am 09. 10. 2020).
- [34] Klaus Tüber, *Analyse des Betriebsverhaltens von Polymer-Elektrolyt-Membran-Brennstoffzellen für portable Systeme*, 2004.
- [35] Alex. (o. J.). “P-Kanal MOSFETs am Mikrocontroller.” AEQ-WEB, Hrsg., Adresse: <https://www.aeq-web.com/p-channel-mosfet-driver-arduino-microcontroller/> (besucht am 27. 08. 2020).
- [36] AZ-delivery. (o. J.). “IRF520 MOS Driver Modul 0-24V 5A.” AZ-delivery, Hrsg., Adresse: <https://www.az-delivery.de/products/irf520-mos-driver-modul-0-24v-5a?variant=38524090962> (besucht am 10. 10. 2020).
- [37] Doc_Arduino. (27.01.2017). “IRF520 Mosfet Modul.” forum.arduino.cc, Hrsg., Adresse: <https://forum.arduino.cc/index.php?topic=449954.0> (besucht am 09. 10. 2020).
- [38] AZ-delivery. (o. J.). “ACS712 Stromsensor 5A Messbereich.” AZ-delivery, Hrsg., Adresse: <https://www.az-delivery.de/products/acs712-5a> (besucht am 11. 10. 2020).
- [39] ETAS GmbH. (o. J.). “ES59x – Universelle Schnittstellenmodule,” Adresse: <https://www.etas.com/de/portfolio/es59x.php> (besucht am 10. 10. 2020).

A Anhang

A.1 Anforderungsliste

Anforderungsliste für die Erweiterung des HyDrives

MPM		RP-HyDrive HyDrive auf basis eines eingebetteten Systems				Issued on 10/08/2020, Version 1 Responsible: Leonard Freisem
ID	D/W	Requirement	Source	Responsible	Changes	Comments
1		Funktion		Leonard Freisem		
1-1	D	Für die Spannungsüberwachung muss an jeder der fünf Brennstoffzellen separat die Spannung gemessen werden	HyDrive			entsprechend dem HyDrive
1-2	D	Um den Ladungszustand des Kondensators zu bestimmen muss die Spannung und der Strom an ihm gemessen werden	HyDrive			entsprechend dem HyDrive
1-3	D	Zur Analyse des Stromflusses im Fahrzeug muss ein Stromsensor in die Schaltung des Fahrzeugs integriert werden	HyDrive			entsprechend dem HyDrive
1-4	D	Um den Sauerstofffluss an den Brennstoffzellen zu erhöhen muss eine Schaltung die Möglichkeit bieten den Lüfter mittels PWM zu steuern.	HyDrive			wie beim HyDrive
1-5	D	Um die Last vom Fahrzeug zu regeln muss der Motor über PWM gesteuert werden	HyDrive			entsprechend dem HyDrive
1-6	D	Um die Last des Prüfstandes zu regeln muss der Motor über PWM gesteuert werden	HyDrive			entsprechend dem HyDrive
1-7	D	Um die Elektrolyse ansteuern zu können muss ein Schalter (Relais, MOSFET, o.ä.) vorhanden sein	HyDrive			wie beim HyDrive Der Schalter wird in Form eines Relais ausgeführt und über die Programmierung angesteuert
2		Lehre und Weiterbildung		Leonard Freisem		
2-1	W	Nach der Verwendung des Prüfstandes sollten Studierende ihre Fachkompetenzen in den Bereichen Embedded Systems und Programmierung ausgebaut haben	Anforderungen an die Hochschullehr e			basierend auf den Aufgaben des heutigen Ingenieurwesens
2-2	W	Während der Nutzung des Prüfstandes sollten Methoden des V-Modells durch das Lehrpersonal an die Studierenden praktisch vermittelt werden können	Anforderungen an die Hochschullehr e			Vermittlung der Theorie mit praktischer Untermauerung durch Hardware
2-3	W	Während der Arbeit mit dem Prüfstand sollte die Arbeit in einem interdisziplinären Team stattfinden	Anforderungen an die Hochschullehr e			Vermittlung der Theorie mit praktischer Untermauerung durch Hardware
2-4	W	Während der Arbeit mit dem Prüfstand sollte das Verständnis für unterschiedliche Möglichkeiten der Datenauswertung erweitert werden	Anforderungen an die Hochschullehr e			Vermittlung der Theorie mit praktischer Untermauerung durch Hardware
2-5	W	Während der Verwendung des Prüfstandes sollte das Verständnis für neue Technologien gestärkt werden	Anforderungen an die Hochschullehr e			Vermittlung der Theorie mit praktischer Untermauerung durch Hardware
2-6	W	Zur Beurteilung des Systemwirkungsgrades kann ein Stromsensor in der Elektrolyseanlage die Möglichkeit bieten den Stromfluss zu erfassen	Studierende			Mögliche Ergänzung des HyDrive, um weitere Informationen zu vermitteln

3		Voraussetzungen/Randbedingungen				
3-1	W	Für die Arbeit mit dem Prüfstand sollte dieser mit Laptop und Peripherie auf einen 50x100cm Schreibtisch passen	Studierende			basierend auf den gegebenen Daten der RP-Systeme und des HyDrive. Betrachtet die Nutzung außerhalb der Universität durch Studierende
3-2	W	Damit Hochschulen und Studierende die Hardware verwenden, sollte die Hardware und Software ein gutes Preisleistungsverhältnis bieten	Studierende und Lehrende			basierend auf den gegebenen Daten der Hersteller der RP-Systeme. Wenn das Preisleistungsverhältnis nicht stimmt, wird das Produkt abgelehnt
3-3	W	Für die Inbetriebnahme sollte die Hardware und Software für Studierende und Lehrende gut erreichbar sein	Studierende und Lehrende			erreichbar - also download über das Internet, Bauteile selber bauen können oder online bestellen
4		Inbetriebnahme/Gebrauch		Leonard Freisem		
4-1	W	Während der Vorbereitung sollte der Aufbau für Lehrende selbsterklärend sein	Lehrende			Geringe Einarbeitungszeit durch intuitive Bedienung
4-2	W	Die erste Begegnung mit dem Vehicle Trainer sollte intuitiv sein und den „Entdeckergeist“ wecken	Studierende und Lehrende			Erweckt Interesse sich mit dem RP-HyDrive auseinander zu setzen
4-3	W	Für ein erstes Experiment sollte mit 1x LED, Jumperkabeln, 2x Widerstand und einem Taster die LED angesteuert werden können	Studierende			ungefährlicher Einstieg in die Programmierumgebung. Erster Versuchsaufbau und nach Erfolgreicher Umsetzung Erfolgsgefühl bei Studierenden
4-4	W	Für die Inbetriebnahme sollte die Software schnell installiert sein	Studierende			innerhalb von 2-3 Stunden. Da das System zunächst unbekannt ist, wird ansonsten Kosten/Nutzen in Frage gestellt
4-5	W	Für die Inbetriebnahme sollte der Nutzer fähig sein die Hardware mit herkömmlichen Mitteln (LAN-Kabel, USB-Kabel,...) anzuschließen	Studierende und Lehrende			Spezialwerkzeug und Spezielle Bauteile, wie auch Kabel sind ggf. kostenintensiv und schwer erhältlich
4-6	W	Für die Programmierung der Hardware sollte die Programmierumgebung dem Anwender die Möglichkeit bieten einzelne Pins direkt anzusteuern/auszulesen	Studierende			grafische Programmierumgebung oder Bibliotheken, wobei eine grafische Programmierumgebung bevorzugt wird
4-7	W	Zur Erweiterung bestehender Systeme und Funktionen sollte die Programmierumgebung übersichtlich sein	Studierende und Lehrende			grafische Programmierumgebung oder Bibliotheken, wobei eine grafische Programmierumgebung bevorzugt wird
4-8	W	Zur Erweiterung bestehender Systeme und Funktionen sollten Hardware modular erweiterbar sein	Studierende und Lehrende			optimalerweise mit Jumperkabeln testen und im Anschluss mit richtigen Kabeln implementieren
5		Problembehandlung		Leonard Freisem		
5-1	W	Für die Reparatur beschädigter Bauteile und Baugruppen sollten Baupläne und -Anleitungen verfügbar sein	Studierende und Lehrende			Baupläne Optimalerweise direkt im Internet erhältlich. Die Bauteile sollten online als einzelperson in kleinen Stückzahlen erwerbbar sein
5-2	W	Für die Beantwortung von Fragen bzgl. Soft- und Hardware sollte eine Community oder ein Herstellersupport zur Verfügung stehen	Studierende und Lehrende			Ein Forum erleichtert die schnelle Hilfe. Ein Supportteam ist nur zu bestimmten Zeiten erreichbar und kann somit nicht immer aushelfen