

Masterarbeit

ENTWICKLUNG EINES GENERISCHEN ABGAS- PRÜFAUTOMATISIERUNGSMODULS

ausgeführt am



FACHHOCHSCHULE DER WIRTSCHAFT

Fachhochschul-Masterstudiengang
Automatisierungstechnik-Wirtschaft

von

Markus Jauk, BSc

1810322012

betreut und begutachtet von

FH-Prof. Dipl.-Ing. Dieter Lutzmayr


Graz, im Jänner 2020

A handwritten signature in blue ink, appearing to read 'Jauk M.', is written above a horizontal dotted line.

Unterschrift

EHRENWÖRTLICHE ERKLÄRUNG

Ich erkläre ehrenwörtlich, dass ich die vorliegende Arbeit selbstständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen nicht benützt und die benutzten Quellen wörtlich zitiert sowie inhaltlich entnommene Stellen als solche kenntlich gemacht habe.

A handwritten signature in blue ink, appearing to read 'Zauer H.', is positioned above a horizontal dotted line.

Unterschrift

DANKSAGUNG

Hiermit möchte ich mich bei allen bedanken, die mich bei dieser wissenschaftlichen Arbeit unterstützt haben. Zuerst möchte ich meiner Freundin, meiner Familie und meinen Freunden danken, die mich während meines Studiums moralisch unterstützt und motiviert haben. Anschließend möchte ich Herrn FH-Prof. Dipl.-Ing. Dieter Lutzmayr für die fachliche Betreuung, Unterstützung und guten Ratschläge für diese Masterarbeit danken. Außerdem möchte ich mich bei der Firma Kristl, Seibt & Co GmbH, sowie Herrn Dipl.-Ing. (FH) Dr. techn. Oliver Lampl für die Betreuung zur Entstehung dieser Masterarbeit bedanken.

KURZFASSUNG

Die Firma Kristl, Seibt & Co GmbH entwickelt und produziert Prüfstände im Bereich der Fahrzeugtechnik. In den vergangenen Jahren wurden verschiedene standardisierte Softwarelösungen für den Betrieb von Motoren-, Antriebsstrang- und Rollenprüfständen entwickelt. Bis dato wurde noch kein standardisiertes Softwaresystem entwickelt, das die weltweiten Prozesse der Messung von Abgasemissionen von Fahrzeugen beinhaltet.

Das Hauptziel dieser Arbeit ist, ein strukturiertes und generalisiertes Softwaresystem zu entwickeln, welches die weltweiten Testprozeduren im Bereich der Abgasmessung von Fahrzeugen abdeckt. Zunächst werden dazu die Grundlagen eines Abgasrollenprüfstands analysiert. Darüber hinaus, werden Methoden der Softwareentwicklung zur Strukturierung und Darstellung komplexer Softwaresysteme erarbeitet. Danach werden bestehende Softwaresysteme im Bereich der Abgasmessung von Fahrzeugen evaluiert.

Nach diesen Untersuchungen wird eine Softwarearchitektur konzipiert, die eine generalisierte Struktur für die Messgeräte, Fahrzyklen und das Verhalten des gesamten Prüfzyklus beinhaltet. Ausgehend davon wird die modellierte Softwarearchitektur in das Prüfstandsautomatisierungssystem Tornado der Firma Kristl, Seibt & Co GmbH implementiert und getestet. Im Vergleich zu den bestehenden Projekten können durch die erarbeitete Vorgehensmethodik nicht nur objektbasierte Strukturen modelliert, sondern auch generische Programmabläufe entwickelt werden. Somit entsteht ein Softwaremodul, das hinsichtlich Erweiterbarkeit, Flexibilität und Testbarkeit neue Maßstäbe setzt.

ABSTRACT

The company Kristl, Seibt & Co GmbH develops and produces test benches in the field of the automotive engineering. In recent years, various standardized software solutions for the operation of engine, powertrain and chassis dynamometers have been developed. However, no standardized software system that includes the global processes of measuring vehicle exhaust emissions has been developed.

The main objective of this thesis is to design a structured and generalized software system that executes test procedures in the field of vehicle emission measurement. Initially, the basic requirements of an exhaust-gas dynamometer are analyzed. Furthermore, software development methods to structure and represent complex software systems are gathered. Thereafter, existing software systems in the field of emission measurement of vehicles are evaluated.

After these investigations, a software architecture is designed that includes a generalized structure for the measuring devices, the driving cycles and the entire testing procedures. Based on this, the modelled software architecture is implemented and verified in the test bench automation system Tornado by Kristl, Seibt & Co GmbH. Compared to the existing projects, with this method, not only object-based structures are modelled, but also generic program sequences are developed. Thus a software module is created, which sets new standards in terms of expandability, flexibility and testability.

INHALTSVERZEICHNIS

1	Einleitung.....	1
1.1	Ausgangssituation.....	1
1.2	Aufgabenstellung und Untersuchungsinteresse.....	1
1.3	Zielsetzung.....	1
1.4	Gliederung der Arbeit.....	2
1.5	Unternehmen.....	2
2	Grundlagen eines Abgasrollenprüfstands.....	3
2.1	Überblick über die Abgasnormen.....	3
2.1.1	Verordnungen und Regulierungen.....	4
2.1.2	Grenzwerte.....	6
2.1.3	Fahrzeugklassen.....	10
2.1.4	Die Typgenehmigung in Europa.....	11
2.1.5	Fahrzyklen der Typ-1-Prüfung.....	13
2.1.6	Real Driving Emissions.....	15
2.2	Testprozedur und Messgeräte der Typ-1-Prüfung.....	16
2.2.1	Abgasaufbereitung.....	17
2.2.2	Analysatoren und Detektoren.....	18
2.2.3	Messung der Partikelmasse.....	19
2.2.4	Messung der Partikelanzahl.....	19
2.2.5	Impinger-/Kartuschenverfahren und FTIR-Spektroskopie.....	19
2.2.6	Strom- und Spannungsmessung.....	19
2.3	Berechnung der Emissionen nach der Typ-1-Prüfung.....	20
3	Softwareentwicklung.....	21
3.1	Phasen der Softwareentwicklung.....	21
3.2	Requirements Engineering.....	23
3.2.1	Aufgaben des Requirements-Engineerings.....	23
3.2.2	Arten von Anforderungen.....	25
3.3	Softwarearchitektur.....	25
3.3.1	Die UML Diagrammfamilie.....	27
3.3.2	Das Entity-Relationship-Modell.....	29
3.3.3	Die Fundamental Modeling Concepts.....	30
4	Das Automatisierungssystem Tornado.....	31
4.1	Die Entwicklungsumgebung Tornado Designer.....	31
4.1.1	Variablen.....	33
4.1.2	Benutzerdefinierte Datentypen.....	33
4.1.3	Katalogelemente.....	33
4.1.4	Befehlsprogramme.....	34
4.1.5	Parametertabellen.....	34
4.1.6	Prüfprogramme.....	34

4.1.7	Treiber	34
4.2	Analyse der bestehenden Projekte.....	35
4.2.1	Prüfstand der TU Graz	35
4.2.2	Prüfstand der FH Joanneum.....	38
4.2.3	Prüfstände bei einem OEM	39
4.3	Auswertung und Vergleich der Projekte.....	41
5	Konzeptionierung des Softwaremoduls.....	43
5.1	Wesentliche Anforderungen an das Softwaremodul.....	43
5.1.1	Funktionale Anforderungen	43
5.1.2	Qualitätsanforderungen.....	44
5.2	Entwurfsphase des Abgasprüfautomatisierungsmoduls.....	45
5.2.1	Strukturentwurf.....	45
5.2.1.1	Positionierung im TORMASTER-Projekt.....	45
5.2.1.2	Struktur der Messgeräte	46
5.2.1.3	Struktur des Abgasprüfautomatisierungsmoduls	55
5.2.2	Steuerungsentwurf	67
5.2.2.1	Fahrzyklensteuerung.....	67
5.2.2.2	Beutelsteuerung.....	69
5.2.2.3	Ablauf der Prüfzyklen	71
6	Implementierung in Tornado.....	75
6.1	Implementierung der Messgeräte.....	75
6.2	Implementierung des Abgasprüfautomatisierungsmoduls	78
6.2.1	Logik der Fahrzyklensteuerung	80
6.2.2	Logik der Beutelsteuerung.....	84
6.2.3	Messgeräteansteuerung im Abgasprüfautomatisierungsmodul	84
6.3	Funktionstest	86
6.4	Erkenntnisse.....	90
7	Fazit und Ausblick.....	91
	Literaturverzeichnis.....	94
	Abbildungsverzeichnis	97
	Tabellenverzeichnis	99
	Abkürzungsverzeichnis	101

1 EINLEITUNG

Dieses Kapitel verschafft einen ersten Einblick in den Inhalt und die Gliederung der vorliegenden Masterarbeit. Zunächst werden die Ausgangssituation und die Aufgabenstellung bzw. das Untersuchungsinteresse erläutert, sowie eine Zielsetzung definiert. Anschließend wird das Unternehmen Kristl, Seibt & Co GmbH näher vorgestellt.

1.1 Ausgangssituation

Die Firma Kristl, Seibt & Co GmbH ist seit mehreren Jahren bestrebt, sogenannte standardisierte Softwarepakete für die verschiedenen Bereiche der Prüfstandsautomatisierung zu entwickeln. Beispielsweise wurden im hausinternen Prüfstandsautomatisierungssystem Tornado standardisierte Pakete für Rollen-, Antriebsstrang- und Motorenprüfstände erarbeitet. An diesen Prüfständen werden unterschiedliche Prüfungen im Bereich der Fahrzeugtechnik durchgeführt. Eine dieser Prüfungen umfasst den Prozess der Abgasmessung von Fahrzeugen. Dieser Prozess konnte bislang jedoch nicht als standardisiertes Softwarepaket der Firma Kristl, Seibt & Co GmbH bereitgestellt werden.

1.2 Aufgabenstellung und Untersuchungsinteresse

Die Aufgabe besteht darin, ein Softwaremodul zu entwickeln, welches die weltweiten Testprozeduren im Bereich der Abgasmessung von Fahrzeugen abdeckt. Dazu soll untersucht werden, welche verschiedenen Varianten der Abgaszyklen existieren und ob es Möglichkeiten gibt, eine allgemeine Struktur der Zyklen zu schaffen. Des Weiteren wird der Frage nachgegangen, welche Messgeräte bei solchen Prüfzyklen benötigt werden, um ein gesetzeskonformes Ergebnis zu erlangen. Bezüglich der Messgeräte soll eine Struktur geschaffen werden, die die Kommunikation und Abläufe (Initialisierung, Kalibrierung etc.) umfasst. Da für die Analyse der Emissionen verschiedene Messgeräte von verschiedenen Herstellern mit unterschiedlichsten Ansteuerungen benötigt werden, soll untersucht werden, wie eine Klassifizierung auf der Ebene der Messgeräte stattfinden kann. Um hierzu allgemeine abstrakte Strukturen in der Software erstellen zu können, soll eine geeignete Vorgehensmethodik der Softwareentwicklung gefunden werden, welche die Konzeptentwicklung, Programmierung und Wartung des Abgasautomatisierungsmoduls unterstützt.

1.3 Zielsetzung

Ziel dieser Arbeit ist es, aus den Untersuchungen der Anforderungen und dem erlangten Wissen der Softwareentwicklung, ein standardisiertes Softwaremodul für das Prüfstandsautomatisierungssystem Tornado zu konzeptionieren und implementieren.

1.4 Gliederung der Arbeit

Diese Masterarbeit ist in einen theoretischen und einen praktischen Teil gegliedert.

Der theoretische Teil befasst sich mit der Untersuchung der Grundlagen eines Abgasrollenprüfstandes und der Findung geeigneter Hilfsmittel in der Softwareentwicklung. Dazu werden zunächst die weltweiten Verordnungen, Grenzwerte, Prüfzyklen sowie benötigten Messgeräte im Bereich der Abgasprüfung von Personenkraftwagen und leichten Nutzfahrzeugen erarbeitet. Anschließend werden die Phasen der Softwareentwicklung, mit Fokus auf die Anfangsphasen Requirements-Engineering und Softwarearchitektur, untersucht.

Im Praxisteil wird das erlangte Wissen verwendet, um bestehende Projekte im Bereich der Abgasprüfung von Gesamtfahrzeugen zu untersuchen. Zudem wird die geeignetste Methode zur Entwicklung der Softwarearchitektur des Abgasprüfautomatisierungsmoduls ausgewählt und anhand dessen, der Aufbau und der Ablauf konzipiert. Danach werden die ersten Implementierungsergebnisse im Prüfstandsautomatisierungssystem Tornado erläutert.

1.5 Unternehmen

Die Firma Kristl, Seibt & Co GmbH (KS Engineers) bietet seit über 40 Jahren Lösungen in den Bereichen Automotive Testing, Industrial Automation und Gebäudetechnik (Building Facilities) an. Das Unternehmen wurde im Jahr 1972 gegründet und umfasste zunächst nur den Bereich der Gebäudetechnik. Im Jahr 1990 erweiterte sich das Geschäftsfeld um Automotive Testing und Industrial Automation. Der Hauptsitz des Unternehmens befindet sich in Graz (Österreich), weitere Standorte liegen in Stuttgart (Deutschland), Zürich (Schweiz) und Pune (Indien). Das gebündelte Wissen aus den Bereichen Maschinenbau, Elektrotechnik, Regelungstechnik, Elektronik und Informatik macht die Firma Kristl, Seibt & Co GmbH zum idealen Partner bei der Entwicklung von effizienten Lösungen und der Abwicklung von Projekten.¹

Automotive Engineering

Diese Abteilung beinhaltet den gesamten Rahmen der Entwicklung von Automobilprüfständen. Sie befasst sich mit Motorprüfständen, Rollenprüfständen, Heißgasprüfständen, Komponentenprüfständen, Getriebeprüfständen, Konditionieranlagen, Batteriesimulatoren, und der Entwicklung des hausinternen Prüfstandsautomatisierungssystems Tornado.¹

Industrial Automation

Die Abteilung ‚Industrial Automation‘ umfasst die Entwicklung von Leitsystemen, die Programmierung von Zutrittssystemen und das Bauen und Optimieren von Batteriesimulatoren. Des Weiteren finden hier auch alle Planungen und Entwicklungen im Bereich des Sondermaschinenbaus statt.¹

Building Facilities

Die Abteilung der Gebäudetechnik (Building Facilities) bietet Lösungen in den Bereichen der Heizungs-, Lüftungs- und Klimatechnik, sowie die Planung und Errichtung von Sanitärsystemen.¹

¹ Vgl. Kristl, Seibt & Co GmbH (2019), Online-Quelle [01.12.2019].

2 GRUNDLAGEN EINES ABGASROLLENPRÜFSTANDS

Im folgenden Kapitel werden die allgemeinen Kriterien und Anforderungen eines Abgasrollenprüfstandes für die Messung und Auswertung der Abgasemissionen eines Personenkraftwagens untersucht. Dazu wird zunächst ein Überblick der weltweit existierenden Abgasnormen, Verordnungen, Grenzwerte und Prüfzyklen geschaffen, sowie die erforderlichen Messgeräte näher erläutert. Da im Bereich der Fahrzeugtechnik Strecken bzw. Geschwindigkeiten oft mit der Einheit Kilometer (km), anstatt mit der SI-Einheit Meter (m), angegeben werden, wird in den folgenden Kapiteln bevorzugt die Einheit Kilometer verwendet.

2.1 Überblick über die Abgasnormen

Weltweit gibt es mehrere geltende Abgasgesetzgebungen für PKWs und leichte Nutzfahrzeuge. Diese Gesetzgebungen wurden innerhalb bestimmter Länder bzw. Vereinigungen von Ländern verabschiedet und haben auch nur in diesen Regionen Wirksamkeit. Abbildung 1 zeigt einen Überblick über die aktuell wirksamen Abgasnormen auf der Welt. Die USA, Europa sowie Japan, China, Südkorea, Indien und auch Brasilien haben ihre eigenen Gesetzgebungen, die die Prozeduren und Abgasgrenzwerte im Bereich der PKW regulieren. Oft werden die in den verschiedenen Ländern geltenden Grenzwerte (z. B. Euro 5) als Abgasnorm bezeichnet. Die Haupttreiber bei der Entwicklung neuer Vorgaben sind die USA und Europa, an deren Verordnungen alle weiteren Länder ihre Gesetzgebungen anlehnen.²

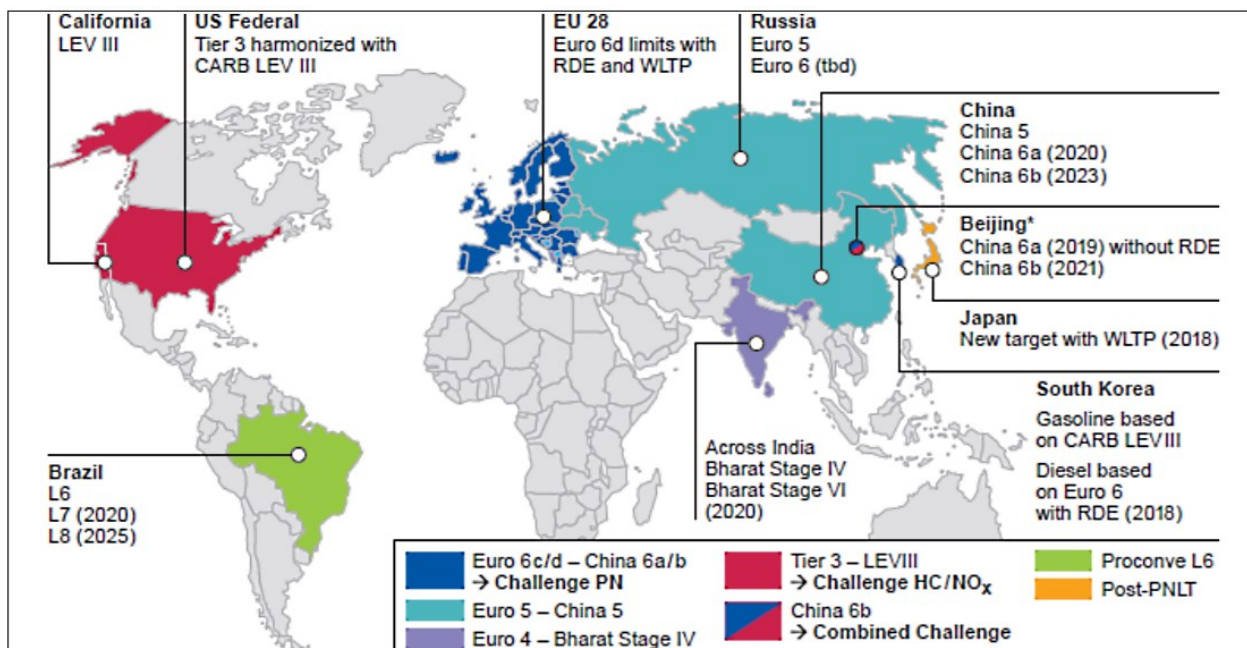


Abbildung 1: Aktuelle weltweite Emissionsgesetzgebungen, Quelle: Continental (2019), Online-Quelle [01.12.2019], S. 18.

In den nachfolgenden Kapiteln werden die Unterschiede zwischen den geltenden Grenzwerten und den damit verbundenen Verordnungen und Prozeduren der verschiedenen Regionen näher erläutert.

² Vgl. Continental (2019), Online-Quelle [01.12.2019], S. 17 ff.

2.1.1 Verordnungen und Regulierungen

Europa

Die United Nations Economic Commission for Europe (UNECE), eine Wirtschaftskommission der Vereinigten Nationen für Europa, ist nicht nur für die Integration und Kooperation europäischer und nicht europäischer Länder zuständig, sondern sie legt auch Normen und Standards in verschiedenen Bereichen fest.³ Die UNECE hat in den letzten Jahren auch Verordnungen und Richtlinien im Bereich der Abgasmessung von Fahrzeugen definiert. Die zwei bekanntesten und umfassendsten Regelungen in diesem Bereich sind:⁴

- UNECE Regulation Nr.83 (UNECE R-83)
- UNECE Global Technical Regulation Nr.15 (UNECE GTR-15)

Die UNECE R-83 beinhaltet die einheitlichen Bedingungen für die Genehmigung der Fahrzeuge hinsichtlich der Emission von Schadstoffen aus dem Motor, entsprechend den Kraftstoffanfordernissen des Motors. Jedoch hatte die UNECE das Ziel, eine grenzübergreifende Harmonisierung von technischen Regelungen zu schaffen. Das Resultat ist die UNECE GTR-15, die die weltweit harmonisierten Prozeduren von Leichtkraftfahrzeugen beschreibt und die UNECE R-83 ablöst.⁵

Das Hauptproblem bestand darin, dass mit der UNECE-R83 bzw. mit dem Fahrzyklus dahinter, dem neuen europäischen Fahrzyklus (kurz NEFZ), keine realistischen Abgasemissionen gemessen werden konnten. Mit der UNECE GTR-15 bzw. dem weltweit harmonisierten Fahrzyklus (WLTC, Worldwide harmonized Light vehicles Test Cycle) und den dort beschriebenen Prozeduren (WLTP, Worldwide harmonized Light vehicles Test Procedure) konnte dies nun erreicht werden.⁶

Die festgelegten technischen Rahmenbedingungen der UNECE werden von der Europäischen Union (EU) unter folgenden Verordnungen übernommen:⁷

- EU 692/2008
- EU 2017/1151

USA

In den USA gibt es zwei verschiedene Organisationen, die die Regelungen im Bereich der Schadstoffemissionen von Fahrzeugen definieren. Dazu gehören die US-amerikanische Umweltschutzbehörde Environmental Protection Agency (EPA) und eine sich im Staat Kalifornien

³ Vgl. UNECE (o.J.), Online-Quelle [01.12.2019].

⁴ Vgl. Engeljehring (2017), Online-Quelle [01.12.2019], S. 12.

⁵ Vgl. Engeljehring (2017), Online-Quelle [01.12.2019], S. 15.

⁶ Vgl. Verband der Automobilindustrie (o.J.), Online-Quelle [01.12.2019].

⁷ Vgl. Continental (2019), Online-Quelle [01.12.2019], S. 22.

befindliche, seit 1966 abgespaltene und eigenständige Behörde namens California Air Resources Board (CARB).⁸

Die technischen Rahmenbedingungen, Prozesse und Analyseverfahren zur Kontrolle der Emissionen von neuen und in Betrieb genommenen Fahrzeugen und Motoren werden in folgenden Verordnungen gesammelt:⁹

- Code of Federal Regulation 40 Part 86 (kurz: 40 CFR Part 86)
- Code of Federal Regulation 40 Part 1065 (kurz: 40 CFR Part 1065)
- Code of Federal Regulation 40 Part 1066 (kurz: 40 CFR Part 1066)

Die CFR ist eine Sammlung der Bundesverordnungen der Vereinigten Staaten, wobei die EPA unter Nr. 40 Kapitel 1 alle Anordnungen zum Schutze der Umwelt definiert hat. In Teil 86 (40 CFR Part 86) waren bislang alle Verordnungen im Bereich der Abgasemissionen für Fahrzeuge hinterlegt. 40 CFR Part 1066 löst seit 2015 nach und nach die Verordnungen der 40 CFR Part 86 ab und ist somit der neue Standard im Bereich der Emissionsmessung für Fahrzeuge. Teilweise gibt es in der 40 CFR Part 1066 Verweise zur 40 CFR Part 1065, die zwar Prozesse für die Messung der Abgasemissionen für Verbrennungsmotoren beschreibt, jedoch auch allgemeine Anforderungen an die Abgasmesstechnik definiert. Mit dem Jahr 2019 ersetzt die 40 CFR Part 1066 die 40 CFR Part 86 komplett. Auch in Kalifornien gelten diese Verordnungen.¹⁰

China

In China lehnen sich die Vorgaben hauptsächlich an jene von Europa, aber auch an Teile der 40 CFR an. Die Bezeichnungen der Regulierungen betreffend der Abgasemissionen in China sind:¹¹

- GB 18352.5
- GB 18352.6

Japan

In Japan werden alle Anforderungen und Regulierungen betreffend der Fahrzeuge und deren Abgase in den Test Requirements and Instructions for Automobile Standards (TRIAS) beschrieben. Dabei sind folgende TRIAS relevant:¹²

- TRIAS 31-J042
- TRIAS 99-006-02
- TRIAS 08-002-02

⁸ Vgl. Continental (2019), Online-Quelle [01.12.2019], S. 104.

⁹ Vgl. Engeljehring (2017), Online-Quelle [01.12.2019], S. 19.

¹⁰ Vgl. Engeljehring (2017), Online-Quelle [01.12.2019], S. 19.

¹¹ Vgl. Delphi (2019), Online-Quelle [01.12.2019], S. 37.

¹² Vgl. Japan Automobile Standard Internationalization Center (o.J.), Online-Quelle [01.12.2019].

TRIAS 08-002-02 soll als neuer Maßstab der Abgasemissionen in Japan gelten und beinhaltet im Wesentlichen die Prozesse des WLTP bzw. der UNECE GTR-15.¹³

Südkorea

Südkorea wendet Verordnungen sowohl aus Europa als auch aus den USA an. Für Dieselfahrzeuge werden die UNECE R-83 und ab 2018 die UNECE GTR-15 angewendet und im Bereich der benzinbetriebenen Fahrzeuge gelten die Verordnungen 40 CFR Part 86 sowie ab 2018 die Vorgaben der 40 CFR Part 1066. Die in Südkorea zuständige Behörde ist das Ministry of Environment (MOE).¹⁴

Indien

In Indien gelten immer noch die Vorgaben aus der europäischen Richtlinie UNECE R-83. Eine derzeitige Umstellung auf die neuen technischen Richtlinien der UNECE GTR-15 ist noch nicht fixiert.¹⁵

Brasilien

In Brasilien lehnt sich die Gesetzgebung an jene der 40 CFR aus den USA an. Die sogenannte Associação Brasileira de Normas Técnicas (ABNT) ist die Zuständige Behörde, die die technischen Rahmenbedingungen für Brasilien festlegt.¹⁶

2.1.2 Grenzwerte

Europa

In Europa werden die Grenzwerte der Abgasemissionen für verschiedene Fahrzeugklassen (siehe Kapitel 2.1.3) definiert. Es wurden bislang sechs unterschiedliche Grenzwertsätze definiert, welche mit Euro 1 bis Euro 6 bezeichnet werden. In vielen Regionen der Welt werden diese Grenzwerte übernommen. Des Weiteren gibt es eine Unterscheidung zwischen den Grenzwerten für dieselbetriebene Fahrzeuge und benzinbetriebene Fahrzeuge. Wird von einem Fahrzeughersteller ein neues Fahrzeug entwickelt, so sind für dieses die aktuell geltenden Grenzwerte der betreffenden Region maßgebend. In Tabelle 1 ist die Entwicklung der Euro-Grenzwerte ersichtlich. Sie gelten für Fahrzeuge der Klasse M und müssen bei einer Zertifizierung nach Typ-1-Messung eingehalten werden.¹⁷

¹³ Vgl. Japan Automobile Standard Internationalization Center (o.J.), Online-Quelle [01.12.2019]

¹⁴ Vgl. Delphi (2019), Online-Quelle [01.12.2019], S. 44 f.

¹⁵ Vgl. Delphi (2019), Online-Quelle [01.12.2019], S. 41 ff.

¹⁶ Vgl. Delphi (2019), Online-Quelle [01.12.2019], S. 36.

¹⁷ Vgl. Continental (2019), Online-Quelle [01.12.2019], S. 37 ff.

Stufe	Eintrittsdatum	CO	THC	THC+NO _x	NMHC	NO _x	PM	PN
		g km ⁻¹						
Benzin (Fremdzündung)								
Euro 1	1992.07	2,72	-	0,97	-	-	-	-
Euro 2	1996.01	2,20	-	0,50	-	-	-	-
Euro 3	2000.01	2,30	0,20	-	-	0,15	-	-
Euro 4	2005.01	1,00	0,10	-	-	0,08	-	-
Euro 5a	2009.09	1,00	0,10	-	0,068	0,06	0,005	-
Euro 5b	2011.09	1,00	0,10	-	0,068	0,06	0,0045	-
Euro 6	2014.09	1,00	0,10	-	0,068	0,06	0,0045	6,0•10 ¹¹
Diesel (Kompressionszündung)								
Euro 1	1992.07	2,72	-	0,97	-	-	0,14	-
Euro 2	1996.01	1,00	-	0,7	-	-	0,08	-
Euro 3	2000.01	0,64	-	0,56	-	0,50	0,05	-
Euro 4	2005.01	0,50	-	0,30	-	0,25	0,025	-
Euro 5a	2009.09	0,50	-	0,23	-	0,18	0,005	-
Euro 5b	2011.09	0,50	-	0,23	-	0,18	0,0045	6,0•10 ¹¹
Euro 6	2014.09	0,50	-	0,17	-	0,08	0,0045	6,0•10 ¹¹

Tabelle 1: Euro-Grenzwerte der Fahrzeugklasse M, Quelle: Delphi (2019), Online-Quelle [01.12.2019], S. 8 ff.

Der Grenzwertsatz Euro 6 unterteilt sich in Euro 6b, Euro 6c, Euro 6d-Temp und Euro 6d. Der Euro-6b-Grenzwertsatz wurde im September 2014 eingeführt und soll die Emissionen entsprechend der Typ-1-Prüfung (siehe Kapitel 2.1.5) nach dem Fahrzyklus NEFZ begrenzen. Seit September 2017 gilt der Grenzwertsatz Euro 6c. Die Grenzwerte bei Euro 6c blieben gleich, jedoch änderte sich der Prüfzyklus vom NEDC auf den WLTC. Die neuen Grenzwertsätze Euro 6d-Temp und Euro 6d beinhalten für die Prüfungen am Prüfstand immer noch die gleichen Grenzwerte, jedoch sind Limitierungen hinsichtlich der Real-Driving-Emissions (RDE) hinzugekommen.¹⁸

Alle europäischen Abgasnormen beinhalten Limitierungen der Abgas- und Partikelemissionen. Zu den Abgasemissionen zählen das Kohlenstoffmonoxid (CO), die Stickoxide (NO_x), die Gesamtkohlenwasserstoffe (THC) und die Nicht-Methan-Kohlenwasserstoffe (NMHC). Die Partikelemissionen werden in die Partikelmasse (PM) und die Partikelanzahl (PN) unterteilt. In weiten Teilen Europas gilt derzeit der europäische Grenzwertsatz Euro 6.¹⁹

¹⁸ Vgl. ADAC (2019), Online-Quelle [01.12.2019].

¹⁹ Vgl. DieselNet (o.J.), Online-Quelle [01.12.2019]

USA

In den USA werden Fahrzeuge und deren Abgasemissionen in sogenannte Bins gruppiert. Es gibt sieben Bins, die im Prinzip die Grenzwerte der NMOG- und NO_x-Emissionen in Milligramm pro Meile (mg/mi) beschreiben. Sie werden als Bin 160, Bin 125, Bin 70, Bin 50, Bin 30, Bin 20 und Bin 0 bezeichnet. Nach den europäischen Bestimmungen werden die Grenzwerte auf einzelne Fahrzeuge angewendet, was in den USA anders gehandhabt wird. So werden die Fahrzeuge zwar jedes Jahr in einen der ‚Bins‘ zertifiziert, jedoch sind für den Hersteller nur die Emissionswerte des Flottendurchschnitts relevant, da diese reguliert werden. Die Grenzwerte in den USA werden von der EPA definiert und werden mit ‚Tier‘ (übersetzt Stufe) bezeichnet. Es wurden bislang drei Grenzwertstufen veröffentlicht.²⁰

Die in Kalifornien eigenständige Behörde CARB definiert eigene Grenzwertstandards mit der Bezeichnung Low Emission Vehicles (LEV). Auch bei den kalifornischen Abgasgrenzwerten gibt es drei Stufen, wobei die aktuellste als ‚LEV III‘ bezeichnet wird. Die CARB wendet nicht die Unterscheidung der Bins an, sondern definiert Kategorien wie Ultra Low Emission Vehicles (ULEV), Super Ultra Low Emission Vehicles (SULEV) und Zero Emission Vehicles (ZEV). In der nachfolgenden Tabelle werden die Emissionsgrenzwerte für US Tier 3 und die kalifornische Einteilung nach LEV III veranschaulicht. Diese Grenzwerte gelten für die Fahrzeugkategorien Light Duty Vehicles (LDV), Light Duty Trucks (LDT) und Medium Duty Passenger Vehicles (MDPV).²¹

US Tier 3	LEV III	NMOG+NO _x	PM	CO	HCHO
Bin	Kategorie	g km ⁻¹			
Bin 160	LEV 160	0,0994	0,0019	0,0026	0,00248
Bin 125	ULEV 125	0,0777	0,0019	0,0013	0,00248
Bin 70	ULEV 70	0,0435	0,0019	0,0011	0,00248
Bin 50	ULEV 50	0,0311	0,0019	0,0011	0,00248
Bin 30	SULEV 30	0,0186	0,0019	0,0006	0,00248
Bin 20	SULEV 20	0,0124	0,0019	0,0006	0,00248
Bin 0	ZEV	0	0	0	0

Tabelle 2: Emissionsgrenzwerte nach Tier 3 und LEV III, Quelle: Continental (2019), Online-Quelle [01.12.2019], S. 104.

Ähnlich wie in Europa gibt es Grenzwerte für Abgasemissionen und Partikelemissionen. Jedoch werden aber auch andere Abgaskomponenten, wie z. B. unverbrannte Kohlenwasserstoffe (Engl. Non Methane Organic Gases NMOG) und Formaldehyd (HCHO), begrenzt. Im Unterschied zu den europäischen Grenzwertsätzen, wird hierbei nicht zwischen Grenzwerten für diesel- und benzinbetriebenen Fahrzeugen unterschieden. Was die Abgas- und Partikelemissionen von Fahrzeugen anbelangt, gelten die amerikanischen bzw. kalifornischen Vorschriften als die strengsten weltweit.²²

²⁰ Vgl. Continental (2019), Online-Quelle [01.12.2019], S. 104 f.

²¹ Vgl. Continental (2019), Online-Quelle [01.12.2019], S. 104 f.

²² Vgl. Delphi (2019), Online-Quelle [01.12.2019], S. 23 f.

China

In China werden zwar die Fahrzeuge wie in Europa in Klassen eingeteilt, jedoch sind dort eigene Grenzwerte definiert. Es gibt die Grenzwerte China 4, China 5 und China 6, wobei auch, analog zum europäischen Modell, in Grenzwerte für Dieselfahrzeuge und Benzinfahrzeuge unterschieden wird. Bislang werden in Peking und Shanghai separate Grenzwerte angewendet. Ab 2020 sollen aber alle Regionen in China die Abgasemissionsgrenzwerte von China 6 anwenden.²³

Indien

In Indien gelten die sogenannten Bharat-Standards, die in Anlehnung an die europäischen Grenzwerte definiert wurden. Die zuletzt geltenden Grenzwerte (2010 eingeführt) werden Bharat Stage 4 genannt, die sich an den Euro-4-Grenzwerten orientieren. Ab 2020 sollen die Grenzwerte Bharat Stage 6 gelten, die wiederum an die Euro-6-Grenzwerte angelehnt sind. Somit wird in Indien eine Regelung, die dem Euro 5 entspricht, übersprungen.²⁴

Südkorea

In Südkorea werden entweder die EU- oder die US-Emissionsgrenzwerte angewendet. Welche Grenzwerte angewendet werden, ist kraftstoffabhängig. So werden bei Dieselfahrzeugen die europäischen Grenzwerte verwendet und bei benzinbetriebenen Fahrzeugen jene aus den USA bzw. Kalifornien. Die Grenzwerte für Benzinfahrzeuge werden als Korean Low Emission Vehicles (KLEV) bezeichnet.²⁵

Japan

In Japan sind in den letzten Jahren mehrere Grenzwerte in Abhängigkeit von bestimmten Fahrzyklen eingeführt worden. Im Jahr 2000 wurde der New-Short-Term-Grenzwertsatz mit dem Fahrzyklus 10-15 Mode und 11 Mode eingeführt. Mit dem Jahr 2005 änderten sich die Grenzwerte und auch die Bezeichnung zu New Long Term, während der Fahrzyklus gleich blieb. Im Jahr 2009 wurden die Post-Long-Term-Grenzwerte mit einem neuen Fahrzyklus namens JC08 eingeführt. Seit 2018 gelten die Grenzwerte der Future Regulations, wobei die Fahrzyklen des WLTP angewendet werden.²⁶

Brasilien

In Brasilien werden die Grenzwertsätze der Abgasemissionen in sogenannte ‚Proconve‘ aufgeteilt. Es gibt drei dieser Grenzwertsätze, die mit L5, L6 und L7 bezeichnet werden. Proconve L7 soll als neuer Standard eingeführt werden und wird voraussichtlich ab 2020 gelten.²⁷

Abbildung 2 veranschaulicht detailliert die weltweit wichtigsten Abgasemissionsgrenzwerte und deren Veränderungen im zeitlichen Verlauf.

²³ Vgl. Delphi (2019), Online-Quelle [01.12.2019], S. 37.

²⁴ Vgl. Delphi (2019), Online-Quelle [01.12.2019], S. 42.

²⁵ Vgl. Delphi (2019), Online-Quelle [01.12.2019], S. 44.

²⁶ Vgl. Delphi (2019), Online-Quelle [01.12.2019], S. 32 f.

²⁷ Vgl. Delphi (2019), Online-Quelle [01.12.2019], S. 36.

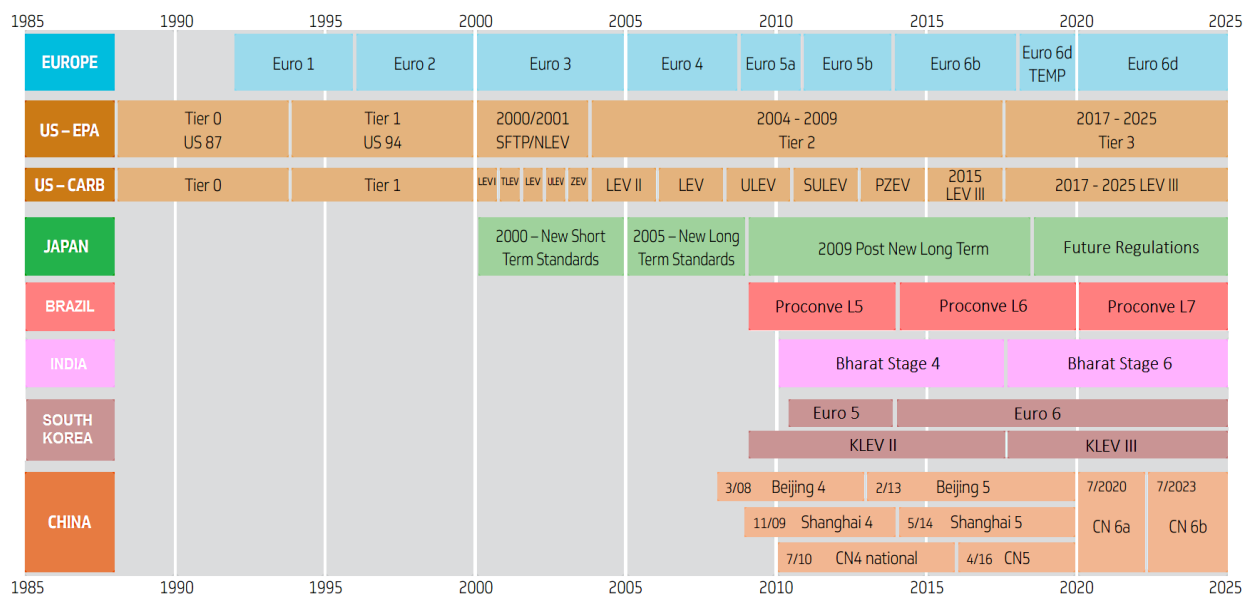


Abbildung 2: Entwicklung der Emissionsgrenzwerte, Quelle: Delphi (2019), Online-Quelle [01.12.2019], S. 1 (leicht modifiziert).

2.1.3 Fahrzeugklassen

Fahrzeuge werden nach Definition der Europäischen Gemeinschaft (EG) in verschiedene Fahrzeugklassen unterteilt. Folgende Klassifizierungen finden sich in den Richtlinien und Verordnungen der EG wieder:²⁸

- **Klasse L**

Zu den Fahrzeugen der Klasse L zählen sowohl zwei- oder dreirädrige Kraftfahrzeuge als auch leichte vierrädrige Kraftfahrzeuge. Zusätzlich ist mit den Bezeichnungen L1e bis L7e eine detailliertere Unterscheidung dieser Klasse definiert. Fahrzeuge der Klasse L werden in der Verordnung 2013/168/EU definiert.

- **Klasse M**

Fahrzeuge der Klasse M sind zur Personenbeförderung ausgelegte und gebaute Kraftfahrzeuge mit mindestens vier Rädern. Diese Fahrzeugklasse wird wiederum in den Klassen M1, M2 und M3 kategorisiert. Eine detaillierte Beschreibung dieser Fahrzeugklasse ist in der Richtlinie 2007/46/EG zu finden.

- **Klasse N**

Auch in der Richtlinie 2007/46/EG enthalten, sind Fahrzeuge der Klasse N. Sie beinhaltet alle für die Güterbeförderung gebauten und ausgelegten Fahrzeuge mit mindestens vier Rädern. Auch hier wird in die Klassen N1, N2, N3 unterteilt.

²⁸ Vgl. Österreich GV (o.J.), Online-Quelle [01.12.2019].

- **Klasse T**

Alle land- oder forstwirtschaftlichen Zugmaschinen auf Rädern gehören zur Klasse T. Gemäß der Verordnung Nr. 167/2013/EU beinhaltet diese Fahrzeugklasse auch weitere Unterklassen von T1 bis T4.

In den USA gilt eine eigene Kategorisierung der Fahrzeuge. Die folgende Auflistung veranschaulicht alle wichtigen Fahrzeugklassen:²⁹

- **Light Duty Vehicles (LDV)**

LDV sind Fahrzeuge, die in der Lage sind, bis zu zwölf Personen zu transportieren.

- **Light Duty Truck (LDT)**

LDT sind Transporter mit einem minimalen Beladungsgewicht von 1701 kg

- **Heavy Duty Trucks (HDT)**

HDT sind motorbetriebene Fahrzeuge mit einem Beladungsgewicht von über 3855,5 kg

Wesentlich hierbei ist, dass die Kategorisierung LDV der europäischen Fahrzeugklasse M entspricht.³⁰

2.1.4 Die Typgenehmigung in Europa

Typgenehmigungen stellen sicher, dass die gesetzlichen Sicherheitsbestimmungen und Umweltstandards bei neu entwickelten Fahrzeugen eingehalten werden. Für Fahrzeuge der Klasse M und N gilt in Europa aktuell die EG-Typgenehmigung 2007/46/EG, ab dem 1. September 2020 tritt jedoch schon eine neue Rahmenverordnung für Typgenehmigungen in Kraft, die sogenannte EU 2018/858.³¹

Bei einer Typgenehmigung werden folgende Bereiche unterschieden:³²

- **Vor Erhalt der Typgenehmigung**

Eine Typgenehmigungsbehörde muss sicherstellen, dass alle Anforderungen hinsichtlich Sicherheit und Umweltbeeinflussung erfüllt werden. Danach erhält das Auto ein Zertifikat, das auch Certificate of Conformity (CoC) genannt wird.

- **Nach Erhalt der Typgenehmigung**

Nach Erhalt der Typgenehmigung muss der Hersteller gewährleisten, dass alle Richtlinien nach der Produktion immer noch eingehalten werden. Dies geschieht im Rahmen der Conformity of Production (CoP). Bei diesem Verfahren müssen Fahrzeuge stichprobenartig aus der Produktion herausgenommen und die Prozeduren der Typgenehmigung wieder durchlaufen werden.

²⁹ Vgl. Transportpolicy (o.J.), Online-Quelle [01.12.2019].

³⁰ Vgl. Transportpolicy (o.J.), Online-Quelle [01.12.2019].

³¹ Vgl. Continental (2019), Online-Quelle [01.12.2019], S. 34.

³² Vgl. Continental (2019), Online-Quelle [01.12.2019], S. 34 f.

Je nach Fahrzeug- bzw. Kraftstofftyp sind verschiedene Prüfverfahren für die Typgenehmigung vorzunehmen. Tabelle 3 veranschaulicht die vorgeschriebenen Prüfverfahren.

Fahrzeugklasse	Fahrzeuge mit Fremdzündungsmotor einschließlich Hybridfahrzeuge				Fahrzeuge mit Selbstzündungsmotoren einschließlich Hybridfahrzeuge
	Benzin	Flüssiggas	Erdgas	Wasserstoff	Diesel
Gasförmige Schadstoffe (Prüfung nach Typ 1)	✓	✓	✓	✓	✓
Partikelmasse (Prüfung nach Typ 1)	✓	-	-	-	✓
Partikelanzahl (Prüfung nach Typ 1)	✓	-	-	-	✓
Gasförmige Schadstoffe (Prüfung nach Typ 1a RDE)	✓	✓	✓	✓	✓
Partikelanzahl (Prüfung nach Typ 1a RDE)	✓	-	-	-	✓
Leerlaufemissionen (Prüfung nach Typ 2)	✓	✓	✓	-	-
Kurbelgehäuseemissionen (Prüfung nach Typ 3)	✓	✓	✓	-	-
Verdunstungsemissionen (Prüfung nach Typ 4)	✓	-	-	-	-
Dauerhaltbarkeit (Prüfung nach Typ 5)	✓	✓	✓	✓	✓
Niedertemperaturemissionen (Prüfung nach Typ 6)	✓	-	-	-	-

Tabelle 3: Prüfvorschriften für die Typgenehmigung, Quelle: Continental (2019), Online-Quelle [01.12.2019], S. 55 (leicht modifiziert).

In der nachfolgenden Auflistung werden die unterschiedlichen Typgenehmigungen näher erläutert:³³

- **Typ 1: Prüfung der Abgasemissionen bei Kaltstart**

Bei diesem Test werden die Abgasemissionen, aber auch die Partikelemissionen während eines standardisierten Fahrzyklus auf einem Rollenprüfstand gemessen und ausgewertet.

- **Typ 1a: Prüfung der Emissionen im praktischen Fahrbetrieb (Real Driving Emissions)**

Bei dieser Prüfung werden die Abgasemissionen nicht auf einem Prüfstand, sondern im Realbetrieb auf der Straße gemessen. Dies erfolgt über ein sogenanntes Portable-Emission-Measurement-System (PEMS).

- **Typ 2: Prüfung der Emissionen von Kohlenmonoxid im Leerlauf**

Bei dieser Typprüfung sind die im Leerlauf auftretenden Abgasemissionen eines Fahrzeugs zu messen.

³³ Vgl. EU 2017/1151 (2017), S. 212 ff.

- **Typ 3: Prüfung der Gasemissionen aus dem Kurbelgehäuse**

Mit der Typ-3-Prüfung wird getestet, ob aus dem Entlüftungssystem des Kurbelgehäuses Abgasemissionen in die Atmosphäre entweichen. Bei dieser Prüfung gilt das Fahrzeug als vorschriftsmäßig, wenn der gemessene Druck des Kurbelgehäuses höher als der Luftdruck während der Messdauer ist.

- **Typ 4: Bestimmung der Verdunstungsemissionen aus Fahrzeugen**

Bei dieser Prüfung wird der Verlust an Kohlenwasserstoffen durch Verdunstung aus dem Kraftstoffsystem von Fahrzeugen bestimmt. Die Messung der verdunsteten Kohlenwasserstoffe wird in einer gasdichten, rechteckigen Messkammer durchgeführt, in der sich das Fahrzeug befindet.

- **Typ 5: Überprüfung der Dauerhaltbarkeit von emissionsmindernden Einrichtungen**

Die Typ 5 Prüfung entspricht einer Alterungsprüfung, bei der nach bestimmten Kilometerständen des Fahrzeugs Typ-1-Prüfungen auf einem Rollenprüfstand oder auch auf Teststrecken durchgeführt werden sollen.

- **Typ 6: Niedertemperaturtest**

Diese Prüfung entspricht der Typ-1-Prüfung, jedoch bei einer Temperatur von -7 °C.

Zusätzlich ist vorgeschrieben, dass alle Fahrzeuge mit einem On-Board-Diagnose-System (OBD) ausgerüstet sein müssen. Auch die Abgastrübung und die Messung der CO₂-Emissionen zur Ermittlung des Kraftstoffverbrauchs sind obligatorisch.³⁴

2.1.5 Fahrzyklen der Typ-1-Prüfung

Fahrzyklen geben Rahmenbedingungen vor, um unterschiedliche Fahrzeuge bezüglich ihres Kraftstoffverbrauchs und ihrer Abgasemissionen vergleichen zu können. Auf einem Rollenprüfstand wird ein Fahrzyklus mit einem bestimmten Geschwindigkeitsprofil vorgegeben, das ein Fahrer / eine Fahrerin mit dem Fahrzeug nachfahren muss.³⁵

Im Laufe der Jahre wurden von verschiedenen Behörden unterschiedliche Fahrzyklen entwickelt. Die wichtigsten Fahrzyklen für PKWs werden in Tabelle 4 aufgelistet.

Bezeichnung	Langform
NEDC / NEFZ	New European Driving Cycle / Neuer europäischer Fahrzyklus
NIDC	New Indian Driving Cycle
CADC	Common Artemis Driving Cycle
FTP-75	Federal Test Procedure
UDDS	Urban Dynamometer Driving Schedule

³⁴ Vgl. Braess/Seiffert (2013), S. 32 f.

³⁵ Vgl. Borgeest (2017), S. 3 f.

HWFET	Highway Fuel Economy Test Cycle
SC03	-
US06	-
NYCC	New York City Cycle
11 Mode	-
10-15 Mode	-
JC08	-
WLTC Class 1	Worldwide harmonized Light vehicles Test Cycle (Class 1)
WLTC Class 2	Worldwide harmonized Light vehicles Test Cycle (Class 2)
WLTC Class 3a	Worldwide harmonized Light vehicles Test Cycle (Class 3a)
WLTC Class 3b	Worldwide harmonized Light vehicles Test Cycle (Class 3b)

Tabelle 4: Fahrzyklen der Typ-1-Prüfung, Quelle: DieselNet (o.J.), Online-Quelle [01.12.2019].

Alle bekannten Fahrzyklen werden in sogenannte Phasen eingeteilt. Diese beschreiben beispielsweise eine Stadt-, Land- oder auch Autobahnfahrt. Der in Abbildung 3 ersichtliche WLTC Class 3b wird z. B. in die vier Phasen Low, Medium, High und Extra High eingeteilt. Die beiden Phasen eines NEFZ werden hingegen als ECE bzw. Urban Driving Cycle (UDC) und Extra Urban Driving Cycle (EUDC) bezeichnet.³⁶

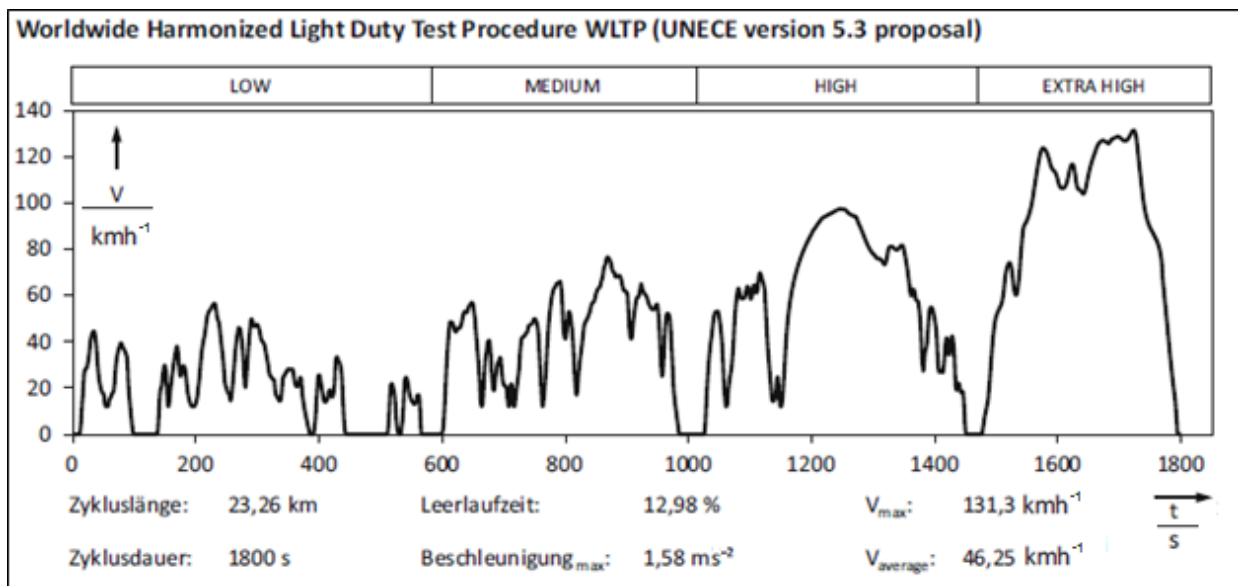


Abbildung 3: Phaseneinteilung des WLTC Class 3b, Quelle: Pischinger/Seiffert (2016), S. 335.

Die WLTC wurden von der UNECE als neue Standardzyklen eingeführt und sollen in Zukunft eine weltweite Harmonisierung der Fahrzyklen bewirken. Derzeit gibt es vier Unterscheidungen dieser harmonisierten Zyklen, welche mit folgenden Kriterien auszuwählen sind:³⁷

³⁶ Vgl. Pischinger/Seiffert (2016), S. 335.

³⁷ Vgl. DieselNet (o.J.), Online-Quelle [01.12.2019].

Fahrzyklus	Leistungsgewicht (LG)	Max. Geschwindigkeit
WLTC Class 1	$LG \leq 22 \text{ W/kg}$	
WLTC Class 2	$22 < LG \leq 34$	
WLTC Class 3a	$LG > 34$	$< 120 \text{ km/h}$
WLTC Class 3b		$\geq 120 \text{ km/h}$

Tabelle 5: Auswahlkriterium der WLTC Fahrzyklen, Quelle: DieselNet (o.J.), Online-Quelle [01.12.2019].

Eine weitere Besonderheit der WLTC ist, dass das Schaltverhalten zum Geschwindigkeitsverlauf nicht mehr fix vom Gesetzgeber vorgegeben wird, sondern über Eingabe verschiedener Fahrzeugparameter berechnet werden muss. Alle anderen bekannten Fahrzyklen haben bei Fahrzeugen mit einem manuellen Getriebe vorgegebene Schaltpunkte.³⁸

Es gibt auch Fahrzyklen, bei denen definierte Pausen zwischen den Geschwindigkeitsphasen vorgegeben sind. In diesen Pausen muss der Motor abgestellt werden. Ein Beispiel dafür ist der US-Zyklus FTP-75. Dieser Fahrzyklus ist in drei Phasen unterteilt, wobei die erste Phase als Kaltstartphase und die zweite Phase als Stabilisierungsphase bezeichnet wird. Nach der zweiten Phase wird der Motor für mindestens 540 Sekunden bis maximal 660 Sekunden abgestellt. In dieser Abstellphase (im Engl. oft auch als ‚Soakphase‘ bezeichnet) ist kein Geschwindigkeitsprofil aktiv. Erst nach Ablauf der Wartezeit kann die dritte Phase begonnen werden, die als Warmstartphase bezeichnet wird. In dieser gelten die gleichen Geschwindigkeitsvorgaben wie in der ersten Phase. Der Fahrzyklus wird in Abbildung 4 dargestellt.³⁹

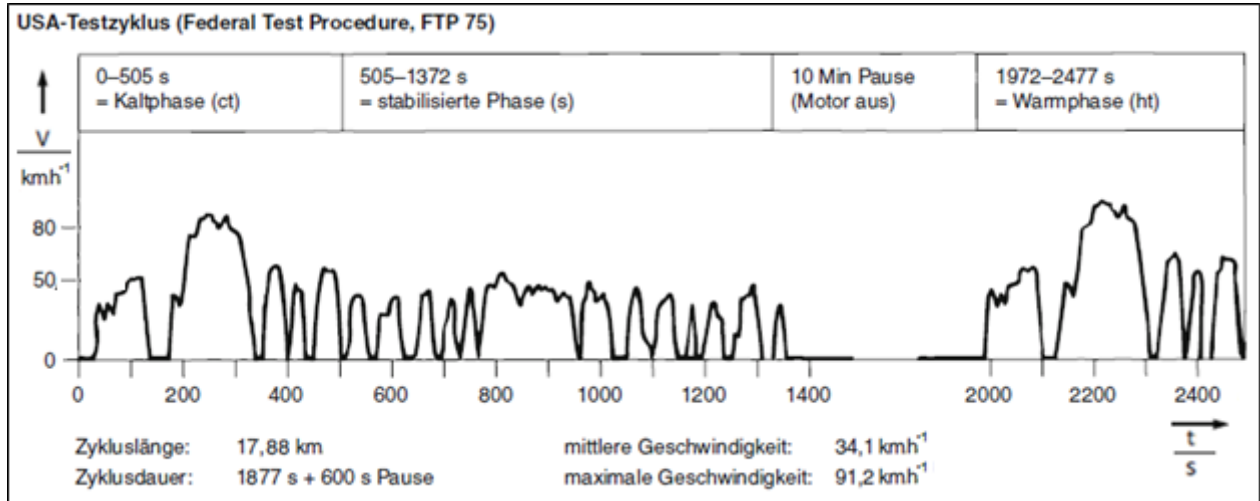


Abbildung 4: Phaseneinteilung des FTP-75, Quelle: Pischinger/Seiffert (2016), S. 335.

2.1.6 Real Driving Emissions

Unter dem Begriff RDE ist ein Prüfverfahren zum Messen und Auswerten von Abgasemissionen von Fahrzeugen im realen Straßenbetrieb zu verstehen. Die derzeitige Emissionsgesetzgebung soll durch eine

³⁸ Vgl. DieselNet (o.J.), Online-Quelle [01.12.2019].

³⁹ Vgl. DieselNet (o.J.), Online-Quelle [01.12.2019].

Überprüfung der RDE ergänzt werden. Die Messung der Emissionen erfolgt über ein sogenanntes Portable PEMS, bei dem die Abgaskonzentrationen des Fahrzeugs sekundlich integriert werden. Des Weiteren gibt es kein definiertes Geschwindigkeitsprofil, vielmehr werden sogenannte ‚Boundary Conditions‘ (Gültigkeitsbereiche) festgelegt, die die Fahrt definieren. Mit dieser Prozedur ist das Fahrzeug mehr Einflussfaktoren (Steigung, Temperatur etc.) ausgesetzt als im standardisierten Ablauf der Typ-1-Prüfung am Prüfstand. Jedoch ist es auch möglich, diese Faktoren während der Realfahrt aufzuzeichnen und diese wieder am Prüfstand nachzustellen.⁴⁰

2.2 Testprozedur und Messgeräte der Typ-1-Prüfung

Das Fahrzeug wird bei einem Abgastest auf einem Rollenprüfstand platziert und an eine Abgasabsaugung angeschlossen. Die Rolle simuliert durch eingestellte Parameter die Straßenlast, während der Fahrer / die Fahrerin dem vorgegebenen Fahrprofil folgt. Das vom Fahrzeug erzeugte Abgas wird über eine Constant-Volume-Sampling (CVS)-Anlage mit Umgebungsluft verdünnt. Das verdünnte Abgas, aber auch die reine Umgebungsluft werden anschließend in Beuteln gesammelt. Die Befüllungsdauer der Beutel entspricht der Zeitspanne der einzelnen Phasen eines Fahrzyklus. Umfasst ein Fahrzyklus zwei Phasen, so werden auch zwei Beutel befüllt. Der Inhalt der Beutel wird den Analysatoren zugeführt, womit die verschiedenen Konzentrationen der Abgaskomponenten gemessen werden. Es besteht auch die Möglichkeit, dass das verdünnte Abgas kontinuierlich über die Analysatoren geführt und gemessen wird. Ein Teil des verdünnten Abgases wird für die Partikelanalyse (Partikelanzahl und Partikelmasse) verwendet. In Abbildung 5 ist ein schemenhafter Aufbau eines Abgasrollenprüfstandes dargestellt.⁴¹

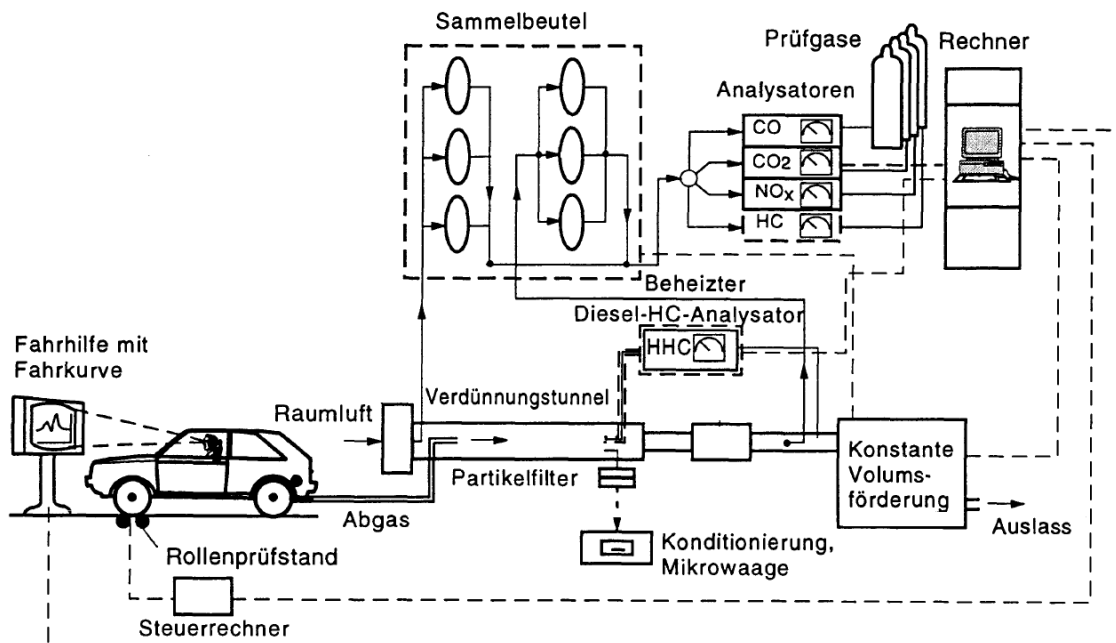


Abbildung 5: Schema eines Abgasrollenprüfstandes, Quelle: Klingenberg (1995), S. 262.

⁴⁰ Vgl. Pischinger/Seiffert (2016), S. 386 f.

⁴¹ Vgl. Hausenberger (2018), Online-Quelle [01.12.2019], S. 30 f.

Je nach Gesetzgebung und Region werden verschiedene Messmethoden und Messgeräte gefordert (siehe Abbildung 6). In den folgenden Kapiteln werden die Standardmessgeräte näher beschrieben, wobei hauptsächlich auf die Messtechnik der sogenannten verdünnten Abgasanalyse (grüner Bereich in Abbildung 6) eingegangen wird. Die unverdünnte Abgasanalyse (roter Bereich in Abbildung 6) wird oft verwendet, um weitere nicht gesetzrelevante Ergebnisse zu ermitteln, die z. B. für die Weiterentwicklung von Abgassträngen an Fahrzeugen benötigt werden (z. B. die Ermittlung der Katalysatoreffizienz).⁴²

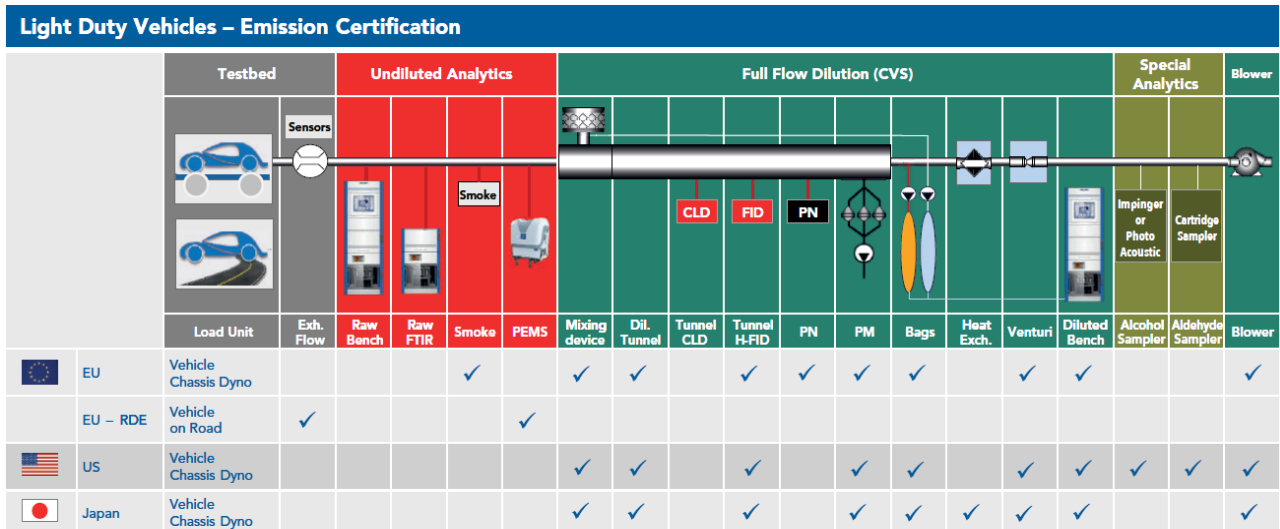


Abbildung 6: Messtechnik an Abgasrollen, Quelle: Engeljehring (2016), Online-Quelle [01.12.2019], S. 55 (leicht modifiziert).

2.2.1 Abgasaufbereitung

Die CVS-Anlage wird zur Verdünnung des emittierten Abgases mit Umgebungsluft verwendet. Des Weiteren wird mit Hilfe dieser Anlage der Abgasstrom durch verschiedene Venturidüsen und spezielle Pumpenanordnungen konstant gehalten. Das Verfahren dahinter wird Critical Flow Venturi (CFV)-Verfahren genannt. Grund der Verdünnung ist, dass Kondensationen von Wasserdampf und die hieraus resultierenden Stickoxid-Verluste sowie die Nachreaktionen vermieden werden können. Die CVS-Anlage kann das verdünnte Abgas und die Verdünnungsluft auf zwei verschiedene Weisen für den Analysator bereitstellen: Entweder wird das verdünnte Abgas kontinuierlich an den Analysator geführt oder es wird in Beutel gefüllt. Bei Zertifizierungen wird die Beutelanalyse verlangt, jedoch sollten beide Methoden zum gleichen Ergebnis führen, da die Schadstoffkonzentration in einem Beutel nach Abschluss der Befüllung genauso groß wie der Mittelwert der Konzentration im verdünnten Abgas über den Zeitraum der Beutelfüllung ist.⁴³

⁴² Vgl. Engeljehring (2016), Online-Quelle [01.12.2019], S. 64 f.

⁴³ Vgl. Reif (2015), S. 141.

2.2.2 Analysatoren und Detektoren

Die Abgasgesetzgebungen geben strikt vor, welche Messverfahren verwendet werden müssen, um die verschiedenen Abgaskomponenten zu erfassen. In Tabelle 6 sind die wichtigsten Abgaskomponenten und die Verfahren zu deren Messung aufgelistet. Die Konzentration der Abgaskomponenten wird in Parts per million (ppm) gemessen.⁴⁴

Abgaskomponente	Verfahren
CO, CO ₂	Nichtdispersiver Infrarot Analysator (NDIR)
NO _x	Chemilumineszenz-Detektor (CLD)
THC	Flammenionisation-Detektor (FID)
CH ₄	Kombination von gaschromatographischem Verfahren oder einem Cutter, und Flammenionisations-Detektor (GC-FID)

Tabelle 6: Messverfahren für Schadstoffe, Quelle: Reif (2015), S. 143.

NDIR

Ein NDIR nutzt die Eigenschaft bestimmter Gase aus, Infrarot-Strahlung in einem schmalen Wellenlängenbereich zu absorbieren. Dabei erzeugt die absorbierte Strahlung Wärme, die wiederum gemessen werden kann. Jedoch funktioniert dieses Prinzip nur bei Molekülen, deren Atome aus mindestens zwei unterschiedlichen Elementen bestehen (z. B. CO₂, CO, SO₂).⁴⁵

CLD

Beim Verfahren des CLD wird das Messgas in einer Reaktionskammer mit Ozon gemischt. Das Ozon wird durch eine Hochspannungsentladung aus Sauerstoff erzeugt. Das im Messgas enthaltene Stickstoffmonoxid oxidiert in dieser Umgebung und es entsteht Stickstoffdioxid. Danach befinden sich die Moleküle in einem angeregten Zustand. Sobald sie wieder in den Grundzustand gelangen, wird Energie in Form von Licht freigesetzt. Dieses kann mittels eines Photomultipliers gemessen werden, um danach auf die Stickstoffmonoxid-Konzentration des Messgases rückzuschließen. Da aber die Gesetzgebungen die Messung der Konzentration aller Stickstoffoxide (NO_x) vorschreiben, muss auch das Stickstoffdioxid des Abgases gemessen werden. Dies geschieht durch einen Konverter, der das NO₂ in NO umwandelt, das danach wiederum durch den CLD geführt wird.⁴⁶

FID

Das Messgas wird in einer Wasserstoffflamme verbrannt, was zu einer Ionisierung führt. Die Ionen werden mittels Elektroden getrennt und es entsteht messbarer Strom. Anhand des Stromes kann proportional auf die gesamte Menge an Kohlenwasserstoffen (Engl. Total Hydro Carbon, THC) geschlossen werden.⁴⁷

⁴⁴ Vgl. Reif (2015), S. 142.

⁴⁵ Vgl. Reif (2015), S. 143 f.

⁴⁶ Vgl. Reif (2015), S. 144 f.

⁴⁷ Vgl. Reif (2015), S. 145.

GC-FID und Cutter-FID

Für die Bestimmung der Methan-Konzentration im Abgas wird auch der FID verwendet, jedoch muss zuvor das Methangas (CH₄) mittels eines speziellen Elementes separiert werden. Dies geschieht entweder über eine Gaschromatographensäule (GC-FID) oder einen beheizten Katalysator, der dafür sorgt, dass die Nicht-CH₄-Kohlenwasserstoffe oxidieren (Cutter-FID). Eine GC-FID kann nicht für kontinuierliche Messungen verwendet werden, sodass dieses Messverfahren nur für Beutelmessungen anwendbar ist.⁴⁸

2.2.3 Messung der Partikelmasse

Für die Bestimmung der Partikelmasse werden gravimetrische Verfahren angewendet. Dabei wird ein Teil des verdünnten Abgases phasenweise über mehrere Filter geleitet. Diese Filter werden vor und nach dem Test gewogen. Die Differenz ergibt die Partikelmasse die sich während des Fahrzyklus am Filter gesammelt hat. Dieses Verfahren ist aufwändig in der Vorbereitung, da eine Konditionierung der Partikelfilter notwendig ist, um Umwelteinflüsse zu minimieren.⁴⁹

2.2.4 Messung der Partikelanzahl

Beim Messverfahren zur Ermittlung der Partikelanzahl wird ein kleiner Teil des verdünnten Abgasstromes mit gesättigtem Butanoldampf vermischt. Das Butanol kondensiert an den Festkörperpartikeln des Abgases, die dadurch an Größe gewinnen. Danach kann mit Hilfe einer Streulichtmessung die Partikelanzahl ermittelt werden.⁵⁰

2.2.5 Impinger-/Kartuschenverfahren und FTIR-Spektroskopie

Wie schon in Abbildung 6 ersichtlich, wird dieses Verfahren in den USA oder auch für bestimmte Kraftstoffe (wenn der Ethanol-Gehalt größer als 22 % ist) verlangt. Dabei handelt es sich um ein Verfahren, womit die Alkohole, Aldehyde und weitere Abgaskomponenten gemessen werden können. Mit Einführung der UNECE GTR-15 (nicht in den USA geltend) können diese Komponenten auch mit einem sogenannten Fourier-Transformation-Infrarot (FTIR)-Messgerät ermittelt werden.⁵¹

2.2.6 Strom- und Spannungsmessung

Da sich die Antriebssysteme der Fahrzeugtechnik in den vergangenen Jahren stark weiterentwickelt haben, sind in der neuen harmonisierten Testprozedur WLTP auch Vorgaben hinsichtlich der Messung verbauten Batteriesystemen festgelegt. Der Lade- und Entladestrom sowie die Spannung der verbauten Batterien (Hybridbatterie, Bordnetzatterie), auch Rechargeable-Electric-Energy-Storage (REES)-Systeme genannt, müssen mit aufgezeichnet werden.⁵²

⁴⁸ Vgl. Reif (2015), S. 145.

⁴⁹ Vgl. Reif (2015), S. 145 f.

⁵⁰ Vgl. Reif (2015), S. 145 f.

⁵¹ Vgl. Engeljehring (2016), Online-Quelle [01.12.2019], S. 59.

⁵² Vgl. ECE/TRANS/180/Add.15/Amend.4 (2018), S. 234 ff.

2.3 Berechnung der Emissionen nach der Typ-1-Prüfung

Nachdem die Testprozedur der Typ-1-Prüfung durchgeführt ist und alle Komponenten des Abgases gemessen sind, muss die Abgasemissionsmasse berechnet werden. Die ermittelten Abgasemissionsmassen der einzelnen Komponenten können im Anschluss mit den Grenzwerten der Abgasnormen verglichen werden. Folgende Formel beschreibt die Berechnung der Emissionen:⁵³

$$M_i = \left(\frac{V_{mix} \cdot Q \cdot k_h \cdot C_i \cdot 10^{-6}}{d_i} \right) \quad (1)$$

$M_i/\text{g km}^{-1}$	Emissionsmasse je Phase i
V_{mix}/l	Volumen des verdünnten Abgases je Phase i im Normzustand (z. B.: 273,15 K und 101325 Pa)
Q/gl^{-1}	Dichte der Abgaskomponente bei Normtemperatur und -druck
$k_h/ -$	Feuchtigkeitskorrekturfaktor
C_i/ppm	Korrigierte Konzentration je Phase i
d_i/km	Gefahrenre Strecke je Phase i

Der Faktor 10^{-6} in der Formel 1 wird benötigt, da die gemessene Konzentration mit der Einheit ppm in die Formel eingetragen wird und auf den Prozentsatz umgerechnet werden muss. Die verdünnte Abgaskonzentration wird um die Verdünnungsluft korrigiert, um die tatsächlich vom Fahrzeug emittierten Abgase zu erhalten. Der Feuchtigkeitskorrekturfaktor wird aus den Umgebungsbedingungen (Temperatur, Druck und relative Feuchte) berechnet und ist nur für die Abgasemissionen der Stickoxide anzuwenden.⁵⁴

⁵³ Vgl. ECE/TRANS/180/Add.15/Amend.4 (2018), S. 245.

⁵⁴ Vgl. ECE/TRANS/180/Add.15/Amend.4 (2018), S. 245.

3 SOFTWAREENTWICKLUNG

In diesem Kapitel werden die Phasen der Softwareentwicklung untersucht. Es werden Lösungen gesucht, die es ermöglichen, komplexe Softwaresysteme besser darzustellen und somit einen sauberen Entwicklungsprozess zu gewährleisten. Zunächst wird der gesamte Softwarelebenszyklus betrachtet und anschließend detaillierter auf die Anfangsphasen eingegangen. Diese Erkenntnisse sollen im weiteren Verlauf der Masterarbeit in der Konzeptionierung des Abgasprüfautomatisierungsmoduls angewendet werden.

3.1 Phasen der Softwareentwicklung

Ein Softwaresystem durchläuft während seiner Lebenszeit verschiedene Stadien. Eines dieser Stadien ist die Softwareentwicklung, die wiederum in einzelne Phasen unterteilt werden kann. Um den Prozess dahinter besser beschreiben zu können, wurde der Begriff des Software-Entwicklungslebenszyklus (Engl. Software-Development-Life-Cycle) eingeführt. Es gibt mehrere Vorgehensmodelle, mit denen der Software-Entwicklungslebenszyklus beschrieben werden kann, wie z. B. traditionelle Modelle wie das Wasserfall- oder das V-Modell, aber auch agile bzw. iterative Vorgehensmodelle wie Extreme Programming (kurz XP) oder Scrum. Im Allgemeinen werden bei der Entwicklung eines Softwaresystems sechs Phasen durchlaufen (siehe Abbildung 7).⁵⁵

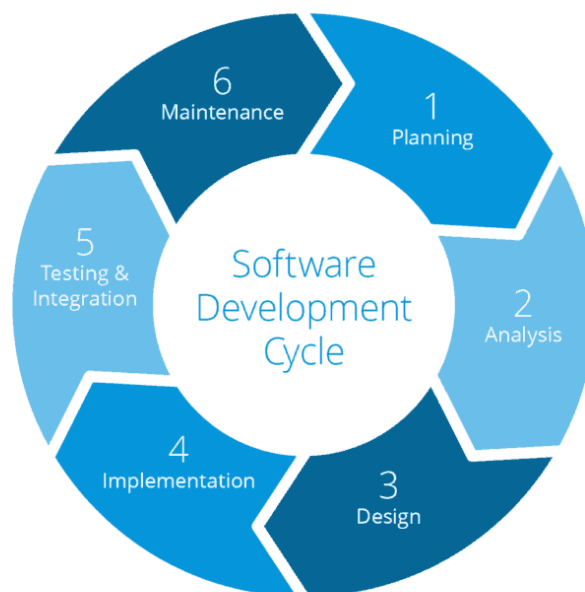


Abbildung 7: Software-Entwicklungslebenszyklus, Quelle: Van der Hoek (2018), Online-Quelle [01.12.2019].

⁵⁵ Vgl. Dumke (2003), S. 18 ff.

Die sechs Phasen des Software-Entwicklungslebenszyklus werden wie folgt definiert:

Planung

In der Planungsphase müssen die Anforderungen an das zu erstellende Softwareprodukt gesammelt werden. Ein intensiver Kontakt mit den Kunden und Stakeholdern ist in dieser Phase notwendig, um Informationen, Dokumente oder auch Normen zusammenzutragen. Zu diesem Zeitpunkt werden das Lasten- und Pflichtenheft für das Softwaresystem erstellt.⁵⁶

Analyse

Die gesammelten Anforderungen aus der Planungsphase müssen im Anschluss in der Analysephase bewertet werden. Die Anforderungen sind hinsichtlich Korrektheit, Vollständigkeit, Sachgerechtigkeit, Konsistenz und Machbarkeit zu prüfen. Die Anforderungsanalyse, die Erhebung der Spezifikation (Planungsphase) und die Umsetzungskontrolle bilden gemeinsam das sogenannte Requirements-Engineering.⁵⁷

Entwurf

In der Entwurfsphase wird die Architektur des Softwaresystems bestimmt. Ziel ist es, eine strukturelle Grundlage zu schaffen, um einerseits einen Überblick über das komplexe Softwaresystem zu erlangen und andererseits die nachfolgende Implementierung bzw. Programmierung zu erleichtern. Durch die Planung der Struktur und des Programmaufbaus können schon im Vorhinein durch Modellierungen verschiedene Lösungsansätze erstellt und danach evaluiert werden. In dieser Phase sollen auch schon Überlegungen hinsichtlich der Testbarkeit des Systems bzw. der Systemteile angestellt werden.⁵⁸

Programmierung

Die Programmierung, oder auch Implementationsphase genannt, ist die Phase in der Softwareentwicklung, in der die Software wirklich umgesetzt wird. Das heißt, die Modellierungen und Entwurfsergebnisse aus der vorherigen Phase werden in Programmcodes umgewandelt. Werden in der Programmierphase neue Erkenntnisse gewonnen, so sind diese wieder in die Modellentwürfe einzubauen.⁵⁸

Validierung/Verifikation

In der Testphase soll erprobt werden, ob das System die Anforderungen in der Praxis erfüllt und ob das Programm den vorgegebenen Spezifikationen entspricht. Hierzu sollen verschiedene Testmethoden, wie z. B. das statische oder dynamische Testen angewendet werden. Auch Testverfahren auf den unterschiedlichen Testebenen (Komponententests, Integrationstests, Systemtests oder Abnahmetests) helfen, das Softwaresystem Schritt für Schritt zu überprüfen.⁵⁹

⁵⁶ Vgl. Dumke (2003), S. 24 f.

⁵⁷ Vgl. Krypczyk/Bochkur (2018), S. 169 f.

⁵⁸ Vgl. Krypczyk/Bochkur (2018), S. 171.

⁵⁹ Vgl. Krypczyk/Bochkur (2018), S. 172 f.

Auslieferung und Wartung

Der Software-Entwicklungszyklus ist mit der Übergabe des programmierten und getesteten Softwaresystems an den Kunden noch nicht beendet. Nicht nur fehlerhafte Programmteile, die in der Testphase übersehen wurden, müssen ausgebessert werden, sondern auch Neuerungen und Verbesserungen des Softwaresystems können vom Kunden nach der Übergabe gefordert werden. Dies führt dazu, dass zuvor abgeschlossene Phasen wieder durchlaufen werden müssen.⁶⁰

3.2 Requirements Engineering

Requirements-Engineering tritt in der Anfangsphase des Softwareentwicklungszyklus (Planung und Analyse) auf und beschreibt den Prozess rund um die Anforderungen (Engl. Requirements) eines Softwaresystems. Es kann in verschiedenen Vorgehensmodellen der Softwaretechnik angewendet werden und ist sowohl als abgeschlossener als auch als begleitender Prozess im Softwareentwicklungszyklus zu sehen.⁶¹

In den nachfolgenden Kapiteln sollen die wichtigsten Merkmale des Requirements-Engineerings erarbeitet werden, um diese für die Entwicklung des Abgasprüfautomatisierungsmoduls anzuwenden.

3.2.1 Aufgaben des Requirements-Engineerings

Anforderungen ermitteln

Bei der Ermittlung der Anforderungen an Softwaresysteme kommen drei Anforderungsquellen in Frage. Die erste Anlaufstelle zur Informationssammlung sind Stakeholder, also projektbetroffene Personen. Als zweite Informationsquelle werden alle Arten von Dokumentationen verwendet. Dies können Normen, Gesetzestexte, aber auch normale Beschreibungen eines Systems sein. Die dritte Anforderungsquelle können im Betrieb laufende Systeme sein. Vorgängersysteme, aber auch Konkurrenzsysteme stellen ebenfalls potentielle Informationsquellen dar.⁶²

Anforderungen dokumentieren

Die Anforderungen sind die Basis für eine Systementwicklung und haben einen entscheidenden Einfluss auf den späteren Projektverlauf. Die dokumentierten Anforderungen können unter Umständen rechtliche Relevanzen haben und sollen daher die Struktur-, Funktions- und Verhaltensperspektive auf das System beinhalten. Die Dokumentation kann sowohl in natürlicher Sprache, als auch mit Hilfe von konzeptionellen Modellen und Diagrammen (z. B. dem Anforderungsdiagramm) erstellt werden. Vorteile bei der Dokumentation von Modellen ist, dass diese gleichzeitig im nächsten Schritt der Softwareentwicklung, also der Softwarearchitektur bzw. dem Design der Software, herangezogen werden können. Jedoch kann bei grafischen Modellen der erforderliche Detaillierungsgrad der Anforderungen oft nicht erfüllt werden. Für die Dokumentation von Anforderungen in natürlicher Sprache können auch sogenannte Sprachschablonen

⁶⁰ Vgl. Krypczyk/Bochkur (2018), S. 173.

⁶¹ Vgl. Pohl/Rupp (2015), S. 3 ff.

⁶² Vgl. Pohl/Rupp (2015), S. 21 ff.

(auch Patterns genannt) eingesetzt werden, die als Baupläne für die syntaktische Struktur einer Anforderung gelten (siehe Abbildung 8). Wichtige Qualitätskriterien bei der Dokumentation von Anforderungen sind die Eindeutigkeit, die Konsistenz, eine klare Struktur, die Erweiterbarkeit des Dokuments, die weitestgehende Vollständigkeit und die Nachvollziehbarkeit.⁶³

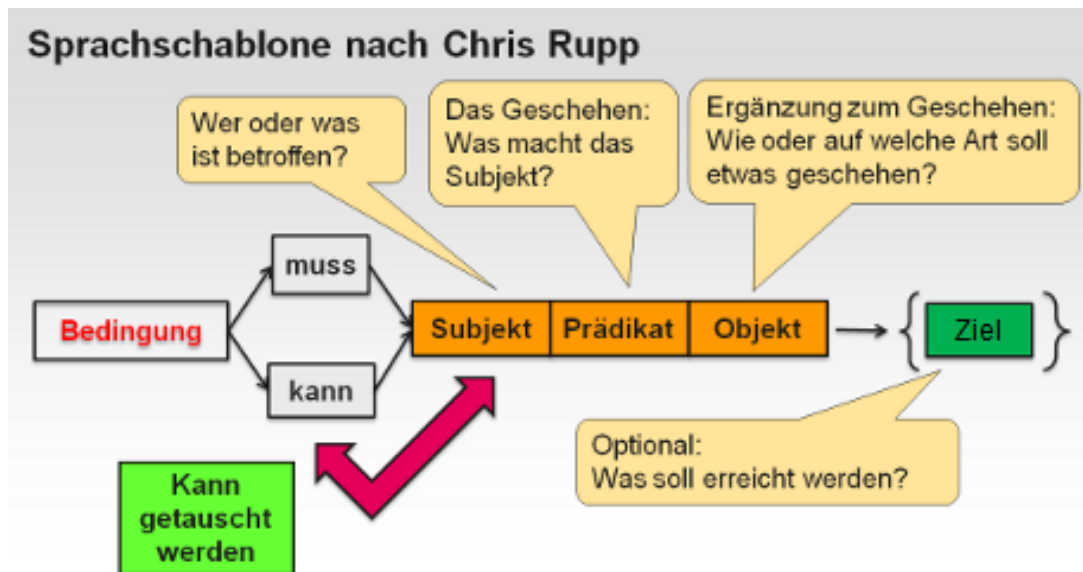


Abbildung 8: Sprachschablone, Quelle: DemoSys (o.J.), Online-Quelle [01.12.2019].

Anforderungen prüfen

Um eine bestimmte Qualität der Anforderungen zu gewährleisten, müssen Qualitätskriterien festgelegt werden. Laut Pohl und Rupp sollen beim Prüfen von Anforderungen drei Qualitätsaspekte beachtet werden. Der Qualitätsaspekt ‚Inhalt‘ soll Aufschluss geben, ob alle relevanten Anforderungen im erforderlichen Detaillierungsgrad ermittelt wurden. Der zweite Qualitätsaspekt ‚Dokumentation‘ legt fest, ob alle Anforderungen gemäß den Dokumentations- und Spezifikationsvorschriften dokumentiert wurden. Der letzte Qualitätsaspekt ‚Abgestimmtheit‘ legt fest, dass alle Stakeholder mit den dokumentierten Anforderungen übereinstimmen und es somit keine Konflikte mehr gibt. Es gibt verschiedene Techniken, um die Prüfung von Anforderungen durchzuführen. Die am häufigsten angewendete Prüfung ist das sogenannte Review, das in den Ausprägungsformen Stellungnahme, Inspektion und Walkthrough ausgeführt werden kann. Aber auch das perspektive Lesen oder der Einsatz von Checklisten können bei der Überprüfung von Anforderungen hilfreich sein.⁶⁴

Anforderungen verwalten

Ein weiteres wichtiges Element des Requirements-Engineerings ist die Verwaltung der Anforderungen. Dabei ist es von Bedeutung, Anforderungen zu kategorisieren und eine Struktur in das Dokumentieren der Anforderungen einfließen zu lassen, um hinsichtlich etwaiger Anforderungsänderungen den Überblick zu behalten. Es ist auch sinnvoll, Anforderungen zu priorisieren, um im darauffolgenden Entwicklungsprozess

⁶³ Vgl. Pohl/Rupp (2015), S. 35 ff.

⁶⁴ Vgl. Pohl/Rupp (2015), S. 95 ff.

mit den wichtigsten Elementen des zu entwickelnden Systems zu beginnen. Auch Versionierungen von Anforderungen sollen durchgeführt werden, um spätere Konflikte mit Stakeholdern zu vermeiden.⁶⁵

3.2.2 Arten von Anforderungen

Laut Pohl und Rupp werden im Bereich des Requirements-Engineering drei Arten von Anforderungen unterschieden:⁶⁶

- **Funktionale Anforderungen**

Diese Anforderungen betreffen die Funktion eines Systems bzw. das, was das System leisten soll. Die funktionalen Anforderungen können noch weiter in Funktions-, Verhaltens- und Strukturanforderungen unterteilt werden. Funktionale Anforderungen sind projektabhängig.

- **Qualitätsanforderungen**

Qualitätsanforderungen oder auch nicht funktionale Anforderungen legen die Qualität eines Softwaresystems fest und beziehen sich somit auf Qualitätsmerkmale, die durch funktionale Anforderungen nicht abgedeckt werden. Die Performance, Zuverlässigkeit, Skalierbarkeit oder auch die Portabilität des Systems sind dabei entscheidende Qualitätsträger.

- **Randbedingungen**

Randbedingungen sind nicht als Anforderungen zu sehen, sondern sie schränken funktionale und nicht funktionale Anforderungen ein. Sie müssen bei der Betrachtung der Anforderungen berücksichtigt werden, da sie den Lösungsraum des zu entwickelnden Systems eingrenzen. Randbedingungen können z. B. Kosten oder verfügbare Ressourcen sein.

3.3 Softwarearchitektur

Die Softwarearchitektur ist ein Spezialgebiet des Software-Engineering. Damit können die Struktur, Grenzen und Schnittstellen der Komponenten innerhalb eines Softwaresystems verständlicher gemacht werden. Der Softwarearchitekt befasst sich aber nicht nur mit der statischen Systemstruktur, sondern legt auch besonderes Augenmerk auf die nicht funktionalen Anforderungen, wie Skalierbarkeit, Performanz oder Verfügbarkeit. Sein Aufgabenbereich ist breit gefächert. Er ist die zentrale Drehscheibe zwischen Kunden, Projektmanagern, Designern und Entwicklern. So muss er funktionale und nicht funktionale Anforderungen in einer Softwarearchitektur abbilden können, Architektur- und Technologie-Entscheidungen argumentieren können und dafür sorgen, dass bei der Implementation nicht von der vorgegebenen Architektur abgewichen wird.⁶⁷

Der Bereich der Softwarearchitektur gewinnt stark an Bedeutung, sobald die Größe und Komplexität eines Softwaresystems zunimmt. In der Softwarearchitektur kann durch Grob- und Feinplanung die Komplexität

⁶⁵ Vgl. Pohl/Rupp (2015), S. 119 ff.

⁶⁶ Vgl. Pohl/Rupp (2015), S. 8 f.

⁶⁷ Vgl. Schatten/Demolsky (2010), S. 200.

eines Softwaresystems reduziert werden. Dies wird grundsätzlich durch die Zerlegung des Gesamtsystems in einzelne Teilsysteme oder Systemkomponenten erreicht.⁶⁸

Bei der Softwarearchitektur geht es nicht nur um die Strukturierung eines Systems, sondern es müssen mehrere Einflussfaktoren berücksichtigt werden. Im Verlauf der Entwicklung eines Softwareprojekts sind verschiedene Rollen und Typen involviert, die unterschiedliche Sichtweisen auf die Umsetzung haben. Das 4+1-Sichten-Modell der Softwarearchitektur (siehe Abbildung 9) stellt die zu beachtenden Sichtweisen (Engl. views) in der Softwarearchitektur grafisch dar. Für jede Sichtweise können bestimmte Diagramme und Modelle als Hilfestellung genutzt werden.⁶⁹

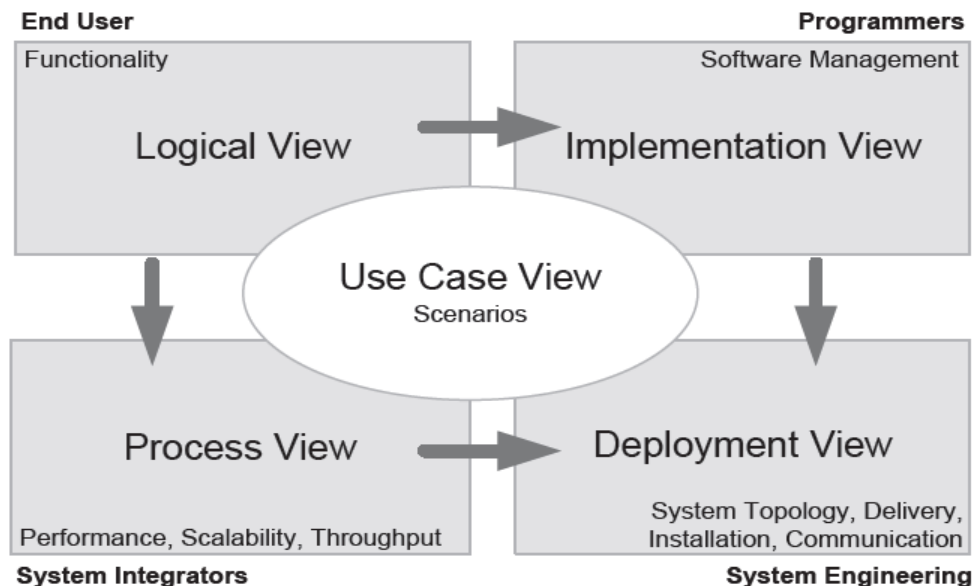


Abbildung 9: 4+1 Sichtenmodell, Quelle: Schatten/Demolsky (2010), S. 207.

Logische Sicht (Logical View)

Diese Sicht beschäftigt sich mit der Funktionalität des Systems für den Endbenutzer. In der logischen Sicht werden Klassen und Subsysteme definiert, die mittels Unified-Modeling-Language (UML) in Klassen- oder auch Paketdiagrammen dargestellt werden können. Diese Sicht wird oft auch als Kontextsicht bezeichnet.⁷⁰

Prozesssicht (Process View)

Die Prozesssicht, auch Ablauf- bzw. Laufzeitsicht genannt, beschreibt alle dynamischen Aspekte der Abläufe des zu entwickelnden Systems, also sämtliche Interaktionen oder die Kommunikation zwischen den Komponenten. Des Weiteren wird in dieser Sicht der Fokus auf Fehlertoleranz, Performance und Erweiterbarkeit gelegt. In UML können bei dieser Sicht beispielsweise Sequenz-, Aktivitäts- oder Kommunikationsdiagramme verwendet werden.⁷¹

⁶⁸ Vgl. Dunkel/Holitschke (2003), S. 15.

⁶⁹ Vgl. Schatten/Demolsky (2010), S. 206 f.

⁷⁰ Vgl. Kleuker (2018), S. 147 f.

⁷¹ Vgl. Schatten/Demolsky (2010), S. 207 f.

Implementierungssicht (Implementation View)

Bei der Implementierungssicht oder Entwicklungssicht geht es um die Verwaltung der erstellten Klassen und Pakete in der Implementierungsphase. Das System wird hier aus der Sicht des Entwicklers beschrieben. Das statische Zusammenspiel von Komponenten wird mit Hilfe von Komponentendiagrammen in UML dargestellt.⁷²

Physikalische Sicht (Deployment View)

In der physikalischen Sicht wird die Hardware, auf der die zu entwickelnde Software später läuft, betrachtet, aber auch, welche Protokolle für die Kommunikation der einzelnen Netzwerke verwendet werden sollen. Sie wird auch Verteilungssicht genannt und kann in UML mit sogenannten Verteilungsdiagrammen dargestellt werden.⁷³

Szenarien (Use Case View)

Die Szenarien oder auch Anwendungsfallsicht genannt dienen als Ergänzung zu den bisherigen Sichten. Es werden die Anwendungsfälle und Aktivitäten als Szenarien abgebildet und somit Schnittstellenfunktionen zwischen den anderen Sichten geschaffen. UML bietet verschiedene Darstellungsmöglichkeiten an, wie z.B. das Anwendungsfalldiagramm oder Aktivitätsdiagramm.⁷⁴

3.3.1 Die UML Diagrammfamilie

Die UML ist eine vereinheitlichte visuelle Modellierungssprache, entwickelt von der Object Management Group (OMG), deren Werkzeuge Softwarearchitekten, aber auch Entwickler beim Analysieren, Designen und Entwickeln von Softwaresystemen unterstützen sollen. Die UML gilt als grafische Standard-Notation und wird vor allem für die moderne Softwareentwicklung, wie z. B. die objektorientierte Softwareentwicklung, eingesetzt. Derzeitiger Standard im Bereich der Softwareentwicklung ist UML2.⁷⁵

In UML2 werden umfassendere Modellierungsmöglichkeiten im Rahmen der Softwareentwicklung bereitgestellt. Dabei wird zwischen zwei grundlegenden Arten von Diagrammen unterschieden: den Strukturdiagrammen und den Verhaltensdiagrammen. Die Strukturdiagramme werden verwendet, um statische Zusammenhänge beschreiben zu können, aber sie eignen sich auch für weitere Aspekte, wie physikalische Elemente, Laufzeitkonfigurationen und domänenspezifische Entitäten. Verhaltensdiagramme werden zur Darstellung der zeitlichen Abhängigkeiten herangezogen, aber auch, um konkrete Abläufe zu visualisieren und dynamische Aspekte eines Systems aufzuzeigen.⁷⁶

Abbildung 10 gibt einen Überblick über die genannten Diagrammart und deren Zusammenhänge.

⁷² Vgl. Schatten/Demolsky (2010), S. 207.

⁷³ Vgl. Kleuker (2018), S. 148 f.

⁷⁴ Vgl. Schatten/Demolsky (2010), S. 208.

⁷⁵ Vgl. Rouse (2016), Online-Quelle [01.12.2019].

⁷⁶ Vgl. Schatten/Demolsky (2010), S. 165 f.

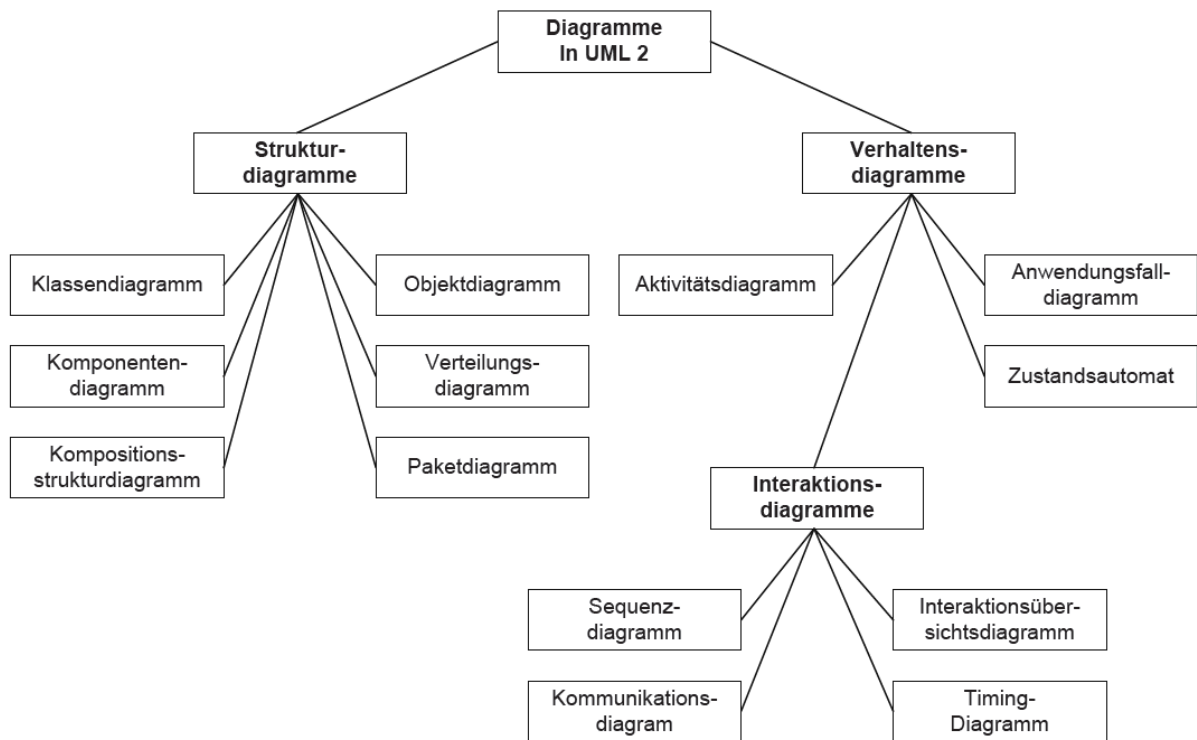


Abbildung 10: Überblick der UML 2 Diagrammfamilie, Quelle: Schatten/Demolsky (2010), S. 165.

In der Softwareentwicklung wird überwiegend die UML eingesetzt. Im Jahre 2007 wurde jedoch die grafische Modellierung in UML2 im Bereich der Systementwicklung um die Systems-Modeling-Language (SysML) erweitert. Die meisten Diagramme der UML2-Diagrammfamilie konnten in SysML übernommen werden oder mussten nur leicht modifiziert werden. Die SysML bietet aber auch neue Diagramme wie das Anforderungsdiagramm oder das Zusicherungsdiagramm.⁷⁷

In der folgenden Auflistung werden die für den späteren Anwendungsfall wichtigsten Diagramme der UML2-Diagrammfamilie in Anlehnung an Schatten und Demolsky kurz erläutert:⁷⁸

- **Paketdiagramm**

Mit Paketdiagrammen kann eine übersichtliche Gliederung der UML-Modelle erreicht werden.

- **Klassendiagramm**

Das Klassendiagramm ermöglicht das Abbilden der Struktur eines bestimmten Softwaresystems. Es zeigt vor allem die statischen Eigenschaften des Systems sowie deren Beziehungen zueinander.

⁷⁷ Vgl. SYSML (o.J.), Online-Quelle [01.12.2019].

⁷⁸ Vgl. Schatten/Demolsky (2010), S. 165 ff.

- **Komponentendiagramm**

Mit dem Komponentendiagramm werden die Schnittstellen und Subsysteme, aus denen ein System besteht, dargestellt.

- **Zustandsautomat**

Mit Zustandsautomaten können Zustände eines Objekts, einer Schnittstelle oder sonstiger Elemente dargestellt werden. Je nach Ereignis wechselt der Zustandsautomat von einem Zustand in den anderen.

- **Aktivitätsdiagramm**

Mit Aktivitätsdiagrammen kann ein bestimmter flussorientierter Prozess oder Algorithmus grafisch dargestellt werden.

3.3.2 Das Entity-Relationship-Modell

Mit einem Entity-Relationship-Modell (kurz ER-Modell) können, wie der Name schon sagt, Beziehungen zwischen Entitäten bzw. Objekten grafisch dargestellt werden. Dieses Modell wurde im Jahr 1976 von Peter Chen entwickelt, um Datenmodelle grafisch darzustellen und deren komplexen Aufbau vereinfacht wiederzugeben. Dieses Diagramm wird vorwiegend in der Entwurfs-Phase von Datenbanken eingesetzt, jedoch kann dieses Modell auch in anderen Anwendungsfällen der Softwareentwicklung hilfreich sein.⁷⁹

Das ER-Modell besteht im Wesentlichen aus drei Elementen, die im Folgenden beschrieben werden:⁸⁰

- **Entitätstyp**

Der Entitätstyp bezeichnet ein reales oder auch abstraktes Informationsobjekt, das im Plural beschrieben werden soll, da dieser mehrere einzelne Entitäten umfasst. Der Entitätstyp wird im Modell mit einem Rechteck dargestellt.

- **Beziehungstyp**

Mit dem Beziehungstyp werden die Zusammenhänge zwischen den Entitätstypen beschrieben. Dieser Typ wird im Modell mit Rauten dargestellt.

- **Attribute**

Mit Attributen können sowohl Entitäts- als auch Beziehungstypen näher beschrieben werden. Die Darstellung im Modell erfolgt durch ovale Symbole. Schlüsselattribute werden im Modell unterstrichen angezeigt und sind die Menge an Attributen, die eine Entität eindeutig identifizieren.

Im ER-Modell werden alle zusammenhängenden Typen mit Linien miteinander verbunden (siehe Abbildung 11).

⁷⁹ Vgl. Gadatsch (2019), S. 9.

⁸⁰ Vgl. Gadatsch (2019), S. 9 ff.

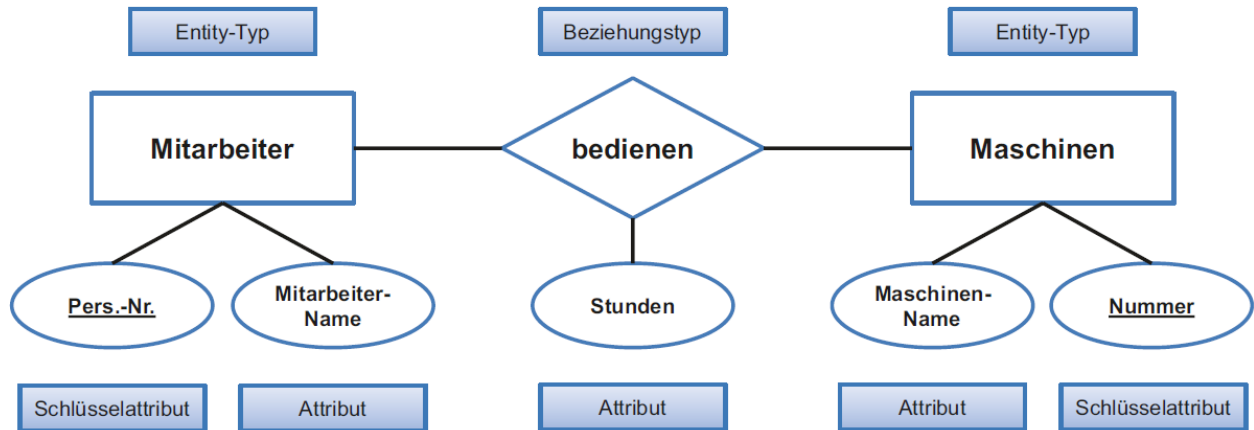


Abbildung 11: Beispiel eines ER-Modells, Quelle: Gadatsch (2019), S. 11.

In weiterer Folge wurden aus dem ER-Modell verschiedene Notationen und Diagramme entwickelt. Beispielsweise wurde von Charles Bachmann eine Art von Datenstrukturdiagramm erarbeitet, das die Struktur von Daten und deren Beziehungen untereinander darstellt. In dieser Modellierungsform werden nur Entitäten (Rechtecke) und Pfeile, die die Zusammenhänge zwischen den Entitäten darstellen sollen, verwendet.⁸¹

3.3.3 Die Fundamental Modeling Concepts

Die Fundamental-Modeling-Concepts (FMC) dienen zur Modellierung komplexer Systeme. Da die FMC die Verwendung von Software nicht zwingend voraussetzen, können prinzipiell auch nicht programmierte Systeme beschrieben werden, also z. B. reine Hardwaresysteme. Jedoch werden FMC auch verwendet, um Softwaresysteme zu konzeptionieren. Jedes informelle System wird nach den Prinzipien der FMC in drei grundsätzliche Strukturkategorien unterteilt: die Aufbau-, Werte- und Ablaufstruktur. Aus diesen Kategorien können drei verschiedene Diagrammtypen abgeleitet werden. Abbildung 12 stellt die Kategorien mit den zugehörigen Aspekten dar.⁸²

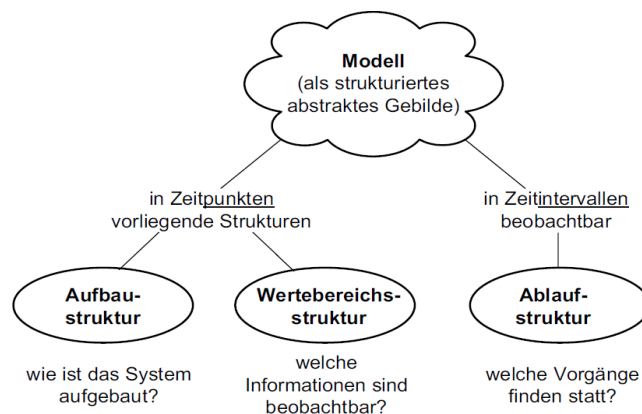


Abbildung 12: Strukturtypen eines informationellen Systemmodells, Quelle: Tabelaing (2006), S. 255.

⁸¹ Vgl. Bachmann (o.J.), Online-Quelle [01.12.2019].

⁸² Vgl. Tabelaing (2006), S. 253 ff.

4 DAS AUTOMATISIERUNGSSYSTEM TORNADO

Im folgenden Kapitel wird das Automatisierungssystem Tornado vorgestellt, in das das Softwaremodul implementiert werden soll. Es wird hauptsächlich auf die Entwicklungsumgebung und die dort bereitgestellten Konfigurationselemente eingegangen, um danach entscheiden zu können, wie das Softwaremodul optimal in diese Umgebung implementiert werden kann. Danach werden Konfigurationen, die bereits im Bereich der Abgasanalyse für Gesamtfahrzeuge entwickelt wurden, analysiert und bewertet.

4.1 Die Entwicklungsumgebung Tornado Designer

Die von Kristl, Seibt & Co GmbH entwickelte Tornado Software Suite besteht aus verschiedenen Softwaresystemen. Kernelement dieses Softwarepakets ist die Entwicklungsumgebung Tornado Designer. Alle Konfigurationselemente werden zunächst in Tornado Designer erstellt, konfiguriert und danach in der Laufzeitumgebung Tornado Runtime oder in der Beauftragungsumgebung Tornado Order Management gestartet und ausgeführt. In Tornado Designer wird festgelegt, wie ein Prüfablauf an einem Prüfstand abzarbeiten ist und welche Messdaten während einer Testprozedur erfasst werden. Im Entwicklungs- und Inbetriebnahmeteam von Tornado wurden in den letzten Jahren sogenannte Masterprojekte entwickelt, in denen Grundkonfigurationen für verschiedene Prüfstandstypen implementiert sind. In Abbildung 13 sind die bestehenden Masterprojekte ersichtlich.

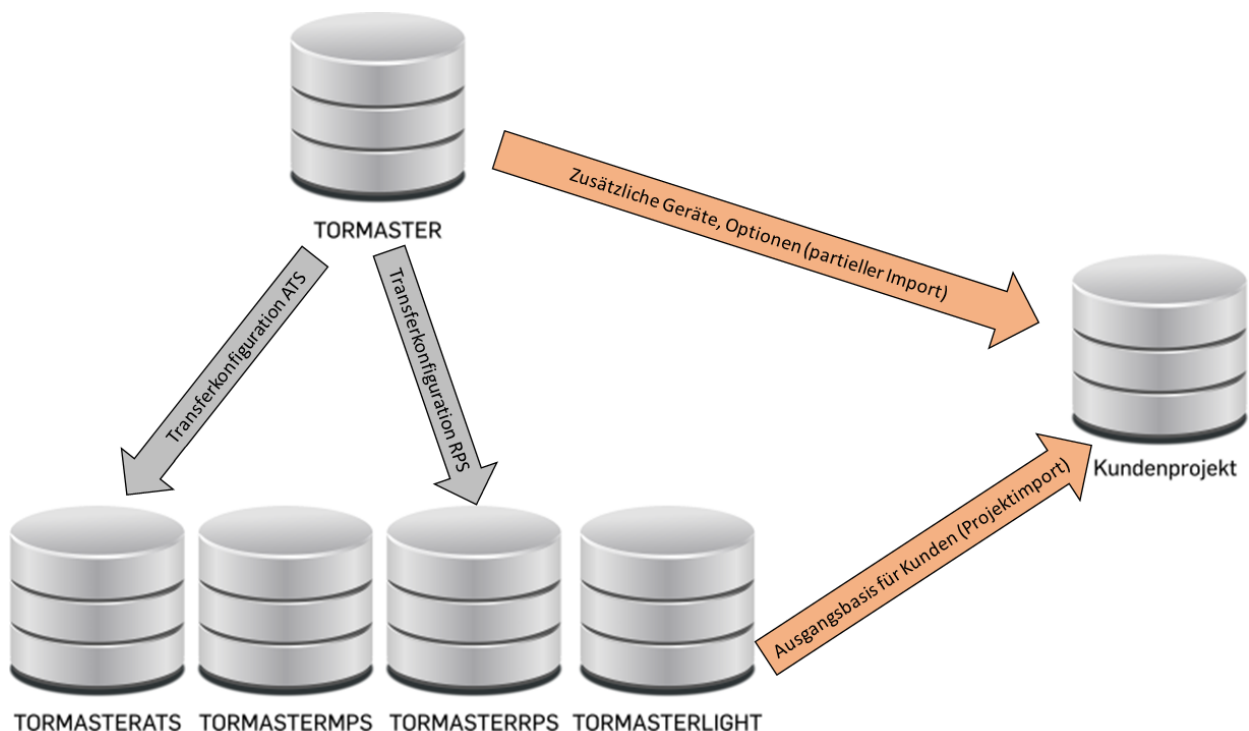


Abbildung 13: Projekthierarchie der Tornado-Masterprojekte, Quelle: Eigene Darstellung.

Das Tornado-Projekt TORMASTER dient als übergeordnetes Projekt für verschiedene Module. Von dort aus können diese Module an die prüfstandsspezifischen Masterprojekte, aber auch an Kundenprojekte transferiert werden. Die prüfstandsspezifischen Masterprojekte sind wie folgt unterteilt:

- **TORMASTERATS** → Masterprojekt für Antriebsstrangprüfstände
- **TORMASTERMPS** → Masterprojekt für Motorenprüfstände
- **TORMASTERRPS** → Masterprojekt für Rollenprüfstände
- **TORMASTERLIGHT** → Masterprojekt für Kleinanwendungen

Da das Abgasprüfautomatisierungsmodul so allgemein und gekapselt wie möglich aufgebaut werden soll, wird es in das TORMASTER-Projekt implementiert. So ist es möglich, den Umfang der Abgasmesstechnik und den dahinterliegenden Prüfablauf sowohl an Rollenprüfständen als auch an Antriebsstrangprüfständen zu übertragen.

In Abbildung 14 ist das Fenster bzw. die Oberfläche von Tornado Designer in der Version 5 ersichtlich. Im linken Bereich des Fensters sind die zur Verfügung stehenden Konfigurationselemente in einer Baumansicht für den Benutzer bereitgestellt. Der rechte Teil der Oberfläche, auch Listenfenster genannt, dient zur Bearbeitung der verschiedenen Konfigurationselemente. Alle Konfigurationselemente können durch partielles Exportieren in andere Projekte importiert werden. In den folgenden Kapiteln werden die wichtigsten Konfigurationselemente und deren bereitgestellten Funktionen von Tornado Designer in dessen Softwareversion 5 näher erläutert.

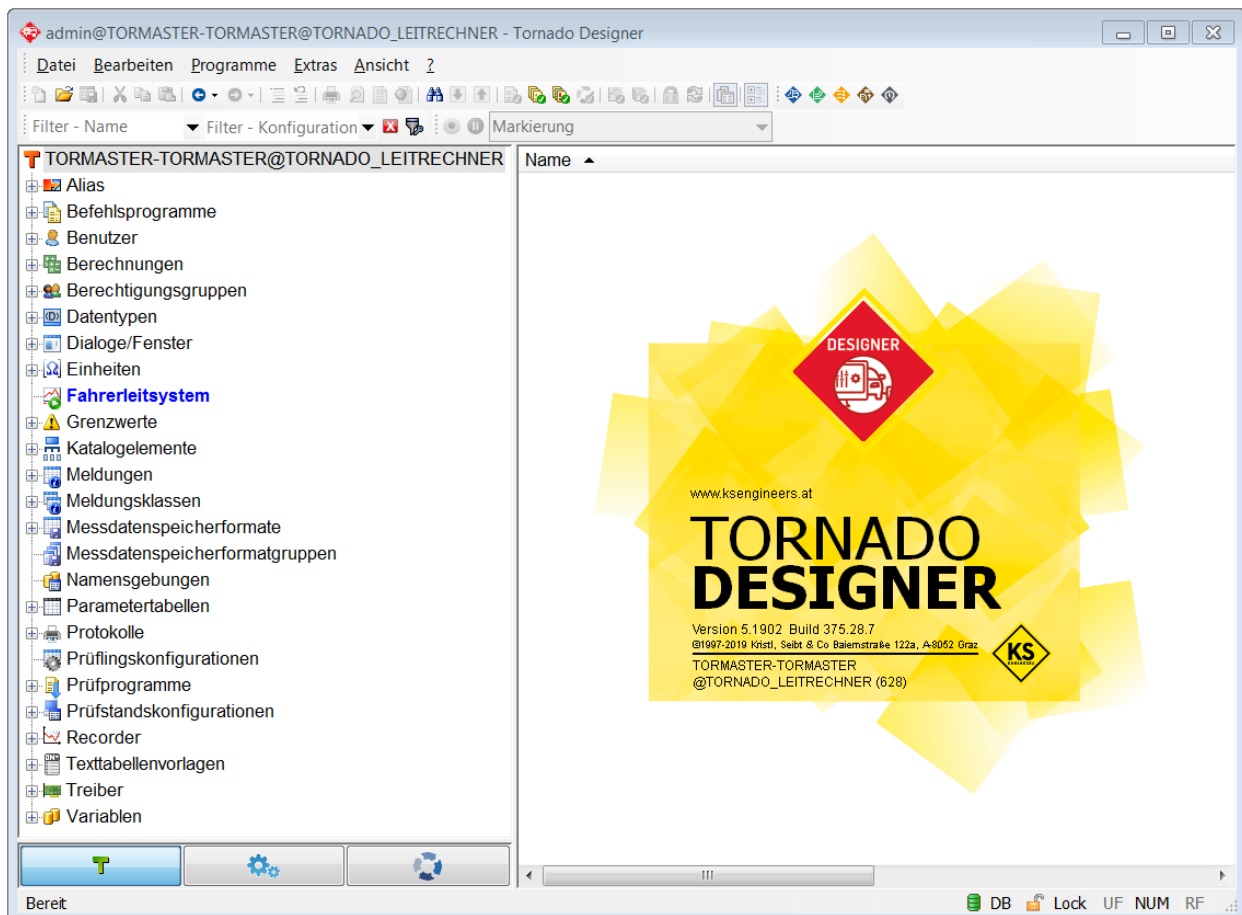


Abbildung 14: Oberfläche des Tornado Designer, Quelle: Eigene Darstellung.

4.1.1 Variablen

Variablen sind die zentralen Speicherelemente in Tornado. Sie halten Werte von Mess- oder Berechnungsgrößen und können mit einer bestimmten Einheit und Datentyp definiert werden. Tornado-Variablen können mit folgenden Grunddatentypen deklariert werden:

- **Ganzzahl**
Die Variable wird in die Menge der ganzen Zahlen dargestellt und entspricht einem Wertebereich eines 32 Bit signed Integers.
- **Fließkommazahl**
Die Variable wird mit einer Genauigkeit von 32 Bit (Single) dargestellt.
- **Text**
Die Variable kann eine bestimmte Anzahl an Zeichen und Zeichenketten beinhalten.
- **Texttabelle**
Die Variable kann mehreren Texten mit dazugehörigen Zahlenwerten definiert werden (z. B. ‚0‘ entspricht dem Text ‚Nein‘ und ‚1‘ entspricht dem Text ‚Ja‘).
- **Digital**
Die Variable wird als boolesche Variable definiert und besitzt somit nur die Wertigkeit 0 oder 1.

Variablen können sowohl als Skalar (entspricht einem einzelnen Wert), als auch als Kennlinie oder Kennfeld (entspricht einer bestimmten Anzahl an Werten; in der Softwareentwicklung als Array bezeichnet) definiert werden. Die Wertigkeit einer Variable, die mit einem Zahlen-Datentyp definiert ist, kann auch einen sogenannten ‚Ungültigen Wert‘ besitzen.

4.1.2 Benutzerdefinierte Datentypen

Im Tornado Designer ist auch das Anlegen von benutzerdefinierten Datentypen möglich. Diese Datentypen können dann ebenfalls bei Variablen definiert werden. In benutzerdefinierten Datentypen lassen sich sowohl Grunddatentypen als auch wieder benutzerdefinierte Datentypen implementieren. Eine Variable mit einem solchen benutzerdefinierten Datentyp erzeugt automatisch für jedes Attribut des Datentyps, inklusive der vom Basisdatentyp geerbten Attribute, eine neue Variable. Analog zur objektorientierten Programmierung agiert in Tornado eine Variable als Objekt und der benutzerdefinierte Datentyp als Klasse des Objekts. Ähnlich wie in den meisten Programmiersprachen wird der Übergang in eine Subdatenstruktur einer Variablen mit einem Punktoperator angezeigt. Die Vorteile dieser benutzerdefinierten Datentypen liegen in der Wiederverwendbarkeit von Variablenstrukturen und der sauberen Strukturierung der Variablen.

4.1.3 Katalogelemente

Katalogelemente sind benannte Instanzen von benutzerdefinierten Datentypen mit konkreten Werten für die einzelnen Attribute. Sie beschreiben somit den Zustand, bzw. den Wert der Variablen. Zum besseren Verständnis wird folgendes Beispiel definiert: Ein benutzerdefinierter Datentyp namens Motor mit den Attributen ‚Anzahl_Zylinder‘ und ‚Hubraum‘ wird erstellt. Zwei Variablen (Motor_1 und Motor_2) mit dem

benutzerdefinierten Datentyp ‚Motor‘ werden definiert. Es können mehrere Katalogelemente erstellt werden (z. B. ‚Kleiner Motor‘ mit Anzahl_Zylinder = 3 und Hubraum = 1 dm³; ‚Großer Motor‘ mit Anzahl_Zylinder = 16 und Hubraum = 8 dm³), die auf die Variablenstrukturen ‚Motor_1‘ oder ‚Motor_2‘ geladen werden können.

4.1.4 Befehlsprogramme

Befehlsprogramme bestehen in Tornado aus einer Liste von (sequenziellen) Anweisungen. Es können verschiedene Befehlsprogrammtypen verwendet werden. Im Rahmen der Entwicklung des Automatisierungssystems Tornado wurde auch ein bestimmter Typ des Befehlsprogramms, das sogenannte Tornado-Script-Befehlsprogramm, entwickelt. Es handelt sich um eine domänenspezifische Programmiersprache, mit der unkompliziert Anweisungen, Bedingungen und Schleifen programmiert werden können. Für komplexere Programmlösungen ist der zweite Befehlsprogrammtyp, die sogenannten Visual-Basic-Script (VBS)-Befehlsprogramme, hilfreich. Diese beinhalten alle Funktionen und Funktionalitäten der Programmiersprache VBS. Des Weiteren wurden in die VBS-Befehlsprogrammen Funktionen eingebaut, mit denen auf Konfigurationselemente des Tornadosystems zugegriffen werden kann.

4.1.5 Parametertabellen

Parametertabellen werden in Tornado verwendet, um zusammenhängende Zahlenreihen zu schaffen. Beispielsweise kann hier ein Geschwindigkeitsverlauf implementiert werden, der als erste Spalte die Zeit und als zweite Spalte die Geschwindigkeit beinhaltet. Diese Zahlenreihen können manuell erstellt werden, aber auch über eine Textdatei (.txt, .csv) importiert werden. Durch sogenannte Parametertabellengruppen werden mehrere Parametertabellen zusammengefasst.

4.1.6 Prüfprogramme

Prüfprogramme dienen in Tornado der Definition einer bestimmten Prüfsequenz. Diese Prüfprogramme werden mit Tornado Script erstellt und beinhalten im Gegensatz zu Befehlsprogrammen zusätzliche Systemeigenschaften. Des Weiteren sind die Funktionen des Fahrerleitsystems nur in Zusammenhang mit einem Prüfprogramm ausführbar. Das Fahrerleitsystem dient als Anzeige der Fahrkurve für den Prüfstandsfahrer bzw. die Prüfstandsfahrerin.

4.1.7 Treiber

Treiber sind in Tornado die Schnittstelle zu anderen Geräten, die über bestimmte Busverbindungen (TCP-IP, Profibus etc.) mit dem Tornado-Rechner verbunden sind. Manche Geräte besitzen bestimmte Kommunikationsprotokolle, mit denen Anweisungen und Einstellungen gesetzt und Messwerte abgefragt werden können. Jeder Treiber in Tornado beinhaltet Befehle und Register, die mit Tornado-Variablen verknüpft werden können. Treiberbefehle wie z. B. ‚Messgas aufschalten‘ können dann mittels einer digitalen Variablen verknüpft werden. Wird die Variable z. B. von null auf eins gesetzt, so sendet der Treiber das entsprechende Kommando aus dem Kommunikationsprotokoll an das Gerät. Andersherum werden bei Registern mit zyklischen Kommandos Messdaten aus dem Gerät ausgelesen und der entsprechenden Variablen zugewiesen.

4.2 Analyse der bestehenden Projekte

Nachdem der Aufbau der Tornado Software Suite, die derzeitige Projektstruktur mit den Masterkonfigurationen sowie die Struktur und Konfigurationselemente von Tornado Designer analysiert wurden, werden in den anschließenden Kapiteln die bereits bestehenden, in Tornado entwickelten Projekte im Bereich der Abgasrollenprüfstände für PKW betrachtet.

4.2.1 Prüfstand der TU Graz

An der TU Graz wurde im Jahr 2001 der erste Abgasrollenprüfstand mit dem Automatisierungssystem Tornado in Betrieb genommen. Das damalige Automatisierungssystem Tornado mit der Version 3 verfügte noch nicht über alle Konfigurationselemente und Konfigurationsmöglichkeiten. Benutzerdefinierte Datentypen, Katalogelemente oder Parametertabellen gab in dieser Version noch nicht.

Systemaufbau

Im Systemaufbau dieses Projekts ist auffällig, dass das Geschwindigkeits- und Schaltungspunktprofil im Prüfprogramm mit Hilfe von Betriebspunkt-Befehlen umgesetzt sind. Betriebspunkte sind Sollwertsprünge von bestimmten Variablen. Mit mehreren aufeinander folgenden Betriebspunkten kann somit ein Fahrzyklus vorgegeben werden. Für jeden Fahrzyklus wird ein eigenes Prüfprogramm angelegt, bei dem die Betriebspunkte aus Textdateien importiert und erstellt werden. Auch die Ansteuerung der Messgeräte wird in jedem Prüfprogramm noch einmal implementiert. Der Ablauf und die Steuerung des Prozesses sind somit sehr unübersichtlich. Abbildung 15 zeigt den Start des Prüfprogramms eines WLTC.

Zeile	Befehl	Parameter	Komme...
1	Befehlsprogramm aufrufen	Init FLS WLTC	
2	Befehlsprogramm aufrufen	2 Minuten Evakuieren	
3	Fahrerleitsystem	Initialisierung	
4	Dialog/Fenster öffnen	Dialog - Warten auf Start	
5	Warten auf Ereignis	Taste_Prüfprogramm_Start > 0,5 -	
6	Dialog/Fenster schließen	Dialog - Warten auf Start	
7	Zuweisung	zyklisch_messen = 1 -	
8	Befehlsprogramm aufrufen	Stoppuhr Start	
9			
10	Kommentar	Beginn des Fahrzyklus	
11	Fahrerleitsystem	Start	
12	Befehlsprogramm aufrufen	Prüflauf - Phase 1 Start	
13	Betriebspunkt	1; keine Messung; v_soll=0,0; steigung_soll=0,0; gang_soll=0,0	@t = 0
14	Betriebspunkt	10; keine Messung; v_soll=0,0; steigung_soll=0,0; gang_soll=0,0	@t = 1
15	Betriebspunkt	1; keine Messung; v_soll=0,0,2; steigung_soll=0,0; gang_soll=0,1	@t = 11
16	Betriebspunkt	1; keine Messung; v_soll=0,2;1,7; steigung_soll=0,0; gang_soll=1,1	@t = 12
17	Betriebspunkt	1; keine Messung; v_soll=1,7;5,4; steigung_soll=0,0; gang_soll=1,1	@t = 13
18	Betriebspunkt	1; keine Messung; v_soll=5,4;9,9; steigung_soll=0,0; gang_soll=1,1	@t = 14
19	Betriebspunkt	1; keine Messung; v_soll=9,9;13,1; steigung_soll=0,0; gang_soll=1,1	@t = 15
20	Betriebspunkt	1; keine Messung; v_soll=13,1;16,9; steigung_soll=0,0; gang_soll=1,2	@t = 16
21	Betriebspunkt	1; keine Messung; v_soll=16,9;21,7; steigung_soll=0,0; gang_soll=2,2	@t = 17
22	Betriebspunkt	1; keine Messung; v_soll=21,7;26; steigung_soll=0,0; gang_soll=2,2	@t = 18
23	Betriebspunkt	1; keine Messung; v_soll=26;27,5; steigung_soll=0,0; gang_soll=2,2	@t = 19
24	Betriebspunkt	1; keine Messung; v_soll=27,5;28,1; steigung_soll=0,0; gang_soll=2,3	@t = 20
25	Betriebspunkt	1; keine Messung; v_soll=28,1;28,3; steigung_soll=0,0; gang_soll=3,3	@t = 21
26	Betriebspunkt	1; keine Messung; v_soll=28,3;28,8; steigung_soll=0,0; gang_soll=3,3	@t = 22
27	Betriebspunkt	1; keine Messung; v_soll=28,8;29,1; steigung_soll=0,0; gang_soll=3,3	@t = 23
28	Betriebspunkt	1; keine Messung; v_soll=29,1;30,8; steigung_soll=0,0; gang_soll=3,3	@t = 24
29	Betriebspunkt	1; keine Messung; v_soll=30,8;31,9; steigung_soll=0,0; gang_soll=3,3	@t = 25
30	Betriebspunkt	1; keine Messung; v_soll=31,9;34,1; steigung_soll=0,0; gang_soll=3,3	@t = 26
31	Betriebspunkt	1; keine Messung; v_soll=34,1;36,6; steigung_soll=0,0; gang_soll=3,3	@t = 27
32	Betriebspunkt	1; keine Messung; v_soll=36,6;39,1; steigung_soll=0,0; gang_soll=3,4	@t = 28
33	Betriebspunkt	1; keine Messung; v_soll=39,1;41,3; steigung_soll=0,0; gang_soll=4,4	@t = 29
34	Betriebspunkt	1; keine Messung; v_soll=41,3;42,5; steigung_soll=0,0; gang_soll=4,4	@t = 30
35	Betriebspunkt	1; keine Messung; v_soll=42,5;43,3; steigung_soll=0,0; gang_soll=4,4	@t = 31
36	Betriebspunkt	1; keine Messung; v_soll=43,3;43,9; steigung_soll=0,0; gang_soll=4,4	@t = 32
37	Betriebspunkt	1; keine Messung; v_soll=43,9;44,4; steigung_soll=0,0; gang_soll=4,4	@t = 33
38	Betriebspunkt	1; keine Messung; v_soll=44,4;44,5; steigung_soll=0,0; gang_soll=4,4	@t = 34
39	Betriebspunkt	1; keine Messung; v_soll=44,5;44,2; steigung_soll=0,0; gang_soll=4,4	@t = 35
40	Betriebspunkt	1; keine Messung; v_soll=44,2;42,7; steigung_soll=0,0; gang_soll=4,4	@t = 36

Abbildung 15: Start des Prüfprogramms eines WLTC im TU-Graz-Projekt, Quelle: Eigene Darstellung.

Eine Implementierung neuer Fahrzyklen ist sehr aufwendig und fehleranfällig, da zusätzlich zum Einspielen des Fahrprofils im Prüfprogramm immer auch die Steuerungen der Messgeräte und die Phasenumschaltung der Fahrkurve mitberücksichtigt werden müssen.

Ergänzend zu den bereits ermittelten Fahrzyklen aus Kapitel 2.1.5 sind im Projekt der TU Graz zwei weitere Fahrzyklen zu finden. Diese beiden Fahrzyklen sind entsprechend des Handbook Emission Factors for Road Transport (HBEFA) implementiert und sind in der nachfolgenden Tabelle aufgelistet:⁸³

Bezeichnung	Langform
ERMES V8	European Research for Mobile Emission Sources Version 8
IUFUC	Inrets Urbain Fluide Courte

Tabelle 7: Neue Fahrzyklen aus dem HBEFA, Quelle: Keller/Hausberger/Wüthrich (2017), Online-Quelle [01.12.2019].

Bei der Untersuchung des ERMES V8 ist aufgefallen, dass die Auswertung der Emissionen nur in drei Phasen (Urban, Extra Urban und Motorway) relevant ist. In den sogenannten Preconditioning-Phasen und der Post-Motorway-Phase werden keine Abgasanalysen durchgeführt (keine Beutelbefüllung und keine Modalanalyse). Dadurch muss zwischen auszuwertenden und nicht auszuwertenden Phasen unterschieden werden. Abbildung 16 stellt den Fahrzyklus ERMES V8 mit den erwähnten Phasen dar. Die auszuwertenden und nicht auszuwertenden Phasen ergeben somit einen Fahrzyklus mit sieben Phasen.

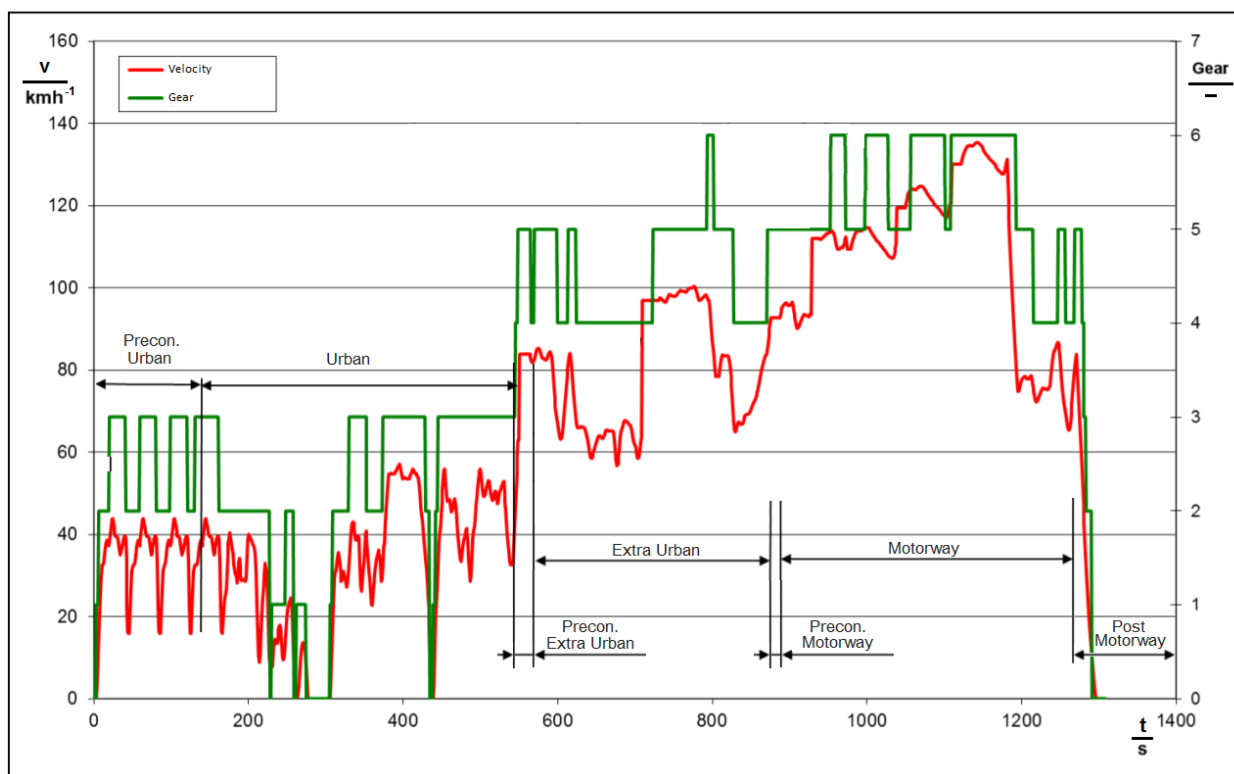


Abbildung 16: ERMES V8 mit Phasenaufteilung, Quelle: Matzer/Rexeis (2016), Online-Quelle [01.12.2019], S. 8 (leicht modifiziert).

⁸³ Vgl. Keller/Hausberger/Wüthrich (2017), Online-Quelle [01.12.2019].

Die Messdaten der Messgeräte werden in einer zyklischen Messdatei (2 Hz) erfasst. Zusätzlich werden während des Prüfzyklus mittels Tornado-Berechnungen Mittelwerte ermittelt und Integrationen einzelner Messkanäle durchgeführt und in eine separate Messdatei geschrieben. Die finale Auswertung der Messwerte bzw. die Berechnung der Emissionen, die während des Fahrzyklus entstanden sind, wird nicht vom Automatisierungssystem Tornado übernommen. Die TU Graz verwendet ein eigenes Berechnungstool, mit dem die Emissionsergebnisse aus der erzeugten Messdatei ermittelt werden.

Messgeräte

Die nachfolgende Tabelle veranschaulicht, welche Messgeräte am Prüfstand verbaut sind, und beinhaltet weitere Informationen zu den Messgeräten.

Messung	Gerätebezeichnung	Hersteller	Kommunikationsprotokoll
Abgaskonzentration	AVL AMA i60	AVL GmbH	AK-Protokoll
Abgasvolumen	Horiba CVS-7000	K.K. Horiba	AK-Protokoll
Partikelmasse über Filter	(Eigenbau)	K&S	Ansteuerung über eine Speicherprogrammierbare Steuerung (SPS)
Partikelanzahl	TSI CPC 3790	TSI Inc.	TSI-Protokoll
Alkohol/Aldehyd- Konzentration	FTIR IAG	IAG GmbH	AK-Protokoll

Tabelle 8: Messgeräte der TU Graz, Quelle: Eigene Darstellung.

Die Messgeräte werden hauptsächlich mit dem sogenannten AK-Protokoll gesteuert. Bei diesem Protokoll handelt es sich um zeichenbasierte Kommandos, mit denen Messgeräte gesteuert werden können. Die AK-Kommandos bestehen aus vier Zeichen und können in folgende drei Kommandotypen unterteilt werden:⁸⁴

- EXXX Einstellkommandos
- AXXX Abfragekommandos
- SXXX Steuerkommandos

Zum Beispiel können mit dem Befehl EKFG und weiteren Parametern die Aktivierung und Deaktivierung der Analysatoren eingestellt werden. Mit dem Steuerbefehl STBY wird das Messgerät in den Standbybetrieb gesetzt, und ASTZ ist der Abfragebefehl, mit dem der Zustand des Messgerätes abgefragt wird. Die meisten Befehle finden sich in allen Messgeräten wieder, jedoch gibt es auch AK-Kommandos, die gerätespezifisch sind. Jedes Messgerät, das über AK-Kommandos gesteuert wird, hat eine eigene Dokumentation der anzuwendenden Befehle.⁸⁵

⁸⁴ AVL (Hrsg.) (2012), S. 101.

⁸⁵ AVL (Hrsg.) (2012), S. 102 ff.

Das Partikelanzahlmessgerät verwendet das von der Firma TSI eigens entwickelte TSI-Protokoll. Ähnlich wie bei den AK-Befehlen geschieht die Kommunikation über verschiedene Zeichenkommandos. Bei diesem Kommunikationsprotokoll wird in sogenannte Read-Kommandos und Set-Kommandos unterteilt, wobei Read-Kommandos mit R und Set-Kommandos mit S beginnen.⁸⁶

4.2.2 Prüfstand der FH Joanneum

Der Abgasrollenprüfstand im Prüffeld der FH Joanneum wurde anfangs mit Tornado 3 in Betrieb genommen. Im Jahr 2014 wurde der Prüfstand mit neuen Messgeräten ausgestattet und das Automatisierungssystem wurde auf Tornado 4 upgedatet. Im Zuge dieses Updates wurde die Ansteuerung der Messgeräte neu implementiert und für die verschiedenen Fahrzyklen mittels Parametertabellen in das Torando-Projekt integriert.

Systemaufbau

In diesem Projekt werden separate Prüfprogramme für die verschiedenen Fahrzyklen implementiert. Im Unterschied zum Tornado-Projekt der TU Graz werden die Geschwindigkeitsprofile und Schaltpunktprofile nicht über Betriebspunkte des Prüfprogramms gesteuert, sondern über Parametertabellen, in denen sowohl das Geschwindigkeitsprofil als auch das Schaltpunktprofil integriert sind. In Abbildung 17 ist ein NEFZ mit einem Schaltpunktprofil für Sechs-Gang-Getriebe ersichtlich. Nachteil dieser Variante ist, dass separate Parametertabellen für Fünf- oder Sechs-Gang-Getriebe erstellt werden müssen, obwohl das Geschwindigkeitsprofil dahinter immer gleich bleibt.

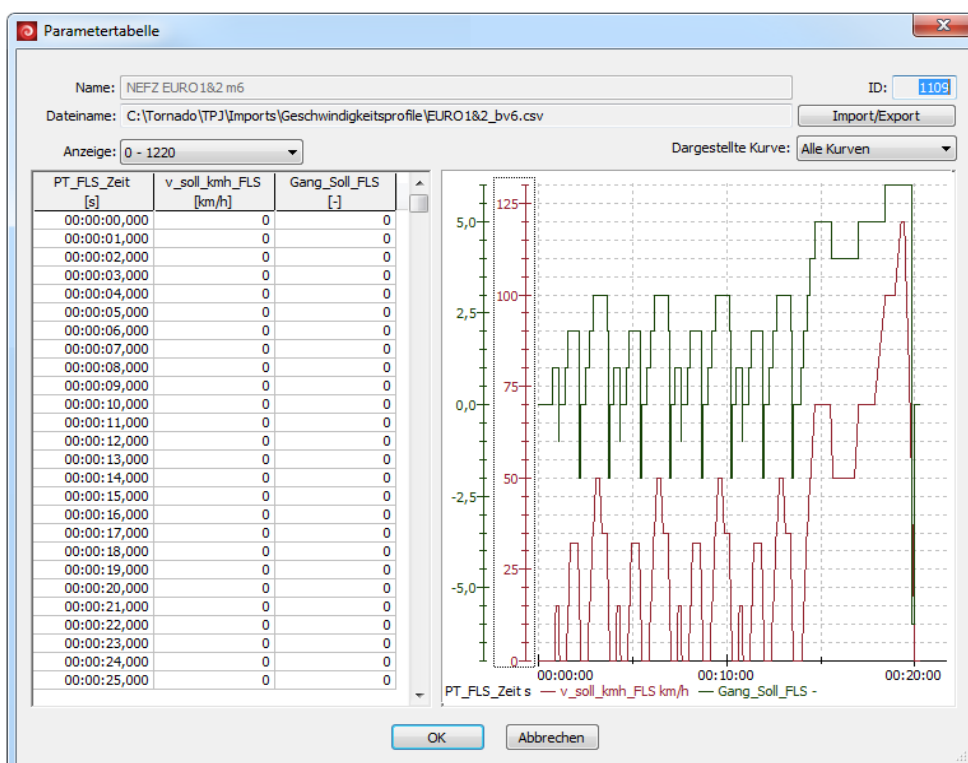


Abbildung 17: Aufbau der Parametertabelle. Quelle: Eigene Darstellung.

⁸⁶ TSI (2007), Online-Quelle [01.12.2019], S. 79.

Alle von den Messgeräten erzeugten Messwerte werden zyklisch mit 10 Hz erfasst. Zusätzlich werden während der Laufzeit der Prüfzyklen Auswertungen in Tornado-Berechnungen durchgeführt. Im Unterschied zum Projekt der TU Graz werden die finale Auswertung und Berechnung der Emissionswerte vom Automatisierungssystem Tornado übernommen.

Messgeräte

Seit der Modernisierung des Prüfstands im Jahr 2014 werden folgende Messgeräte verwendet:

Messung	Gerätebezeichnung	Hersteller	Kommunikationsprotokoll
Abgaskonzentration	AVL AMA i60	AVL GmbH	AK-Protokoll
Abgasvolumen	AVL CVS-i60	AVL GmbH	AK-Protokoll
Partikelmasse über Filter	AVL PSS i60	AVL GmbH	AK-Protokoll
Partikelanzahl	AVL APC 489	AVL GmbH	AK-Protokoll
Strom-/Spannungs- messung	HIOKI 3390	HIOKI GmbH	SCP-Protokoll

Tabelle 9: Messgeräte der FH Joanneum, Quelle: Eigene Darstellung.

An diesem Prüfstand werden Messgeräte der Firma AVL GmbH verwendet, die über das Kommunikationsprotokoll AK angesteuert werden. Das Messgerät zur Erfassung des Lade- und Entladestroms der Batterie des Fahrzeugs wird mit einem sogenannten SCP-Protokoll gesteuert. Dieses Protokoll basiert, ähnlich wie das AK-Protokoll, auf ASCII-Zeichen. Jedes Kommando beginnt mit einem Doppelpunkt. Abfragebefehle werden mit einem Fragezeichen abgeschlossen.

4.2.3 Prüfstände bei einem OEM

Die Inbetriebnahme und Entwicklung des Automatisierungssystems Tornado für die Abgasrollenprüfstände eines Erstausrüsters (Engl. Original Equipment Manufacturer, OEM) war im Jahr 2013 das erste größere Projekt im Bereich der Abgasmessung von PKW. Es wurde ein bestehendes System abgelöst und durch die Tornado Software Suite ersetzt. Im Zuge der Entwicklung des Projekts wurde auch das Beauftragungssystem Tornado Ordermanagement implementiert. Die Konfigurationselemente benutzerdefinierte Datentypen und Katalogelemente wurden im Zuge dieser Entwicklung auch eingeführt.

Systemaufbau

Die Fahrzyklen in diesem Projekt werden in Parametertabellen gespeichert, wobei es eine Trennung zwischen Parametertabellen für Geschwindigkeits- und Schaltpunktprofile gibt. Zusätzlich ist es möglich, auch zeitliche Abläufe für Steigungen zu hinterlegen. Somit bleibt das vom Gesetzgeber vorgeschriebene Geschwindigkeitsprofil immer gleich, lediglich ein Schaltpunkt- und/oder ein Steigungsprofil wird parallel dazu geladen.

In diesem Projekt wird der Ablauf der Fahrzyklen über ein Prüfprogramm in Tornado abgearbeitet. Ein im Hintergrund betriebenes Befehlsprogramm erkennt Phasenwechsel im ablaufenden Fahrzyklus und regelt

auch die Ansteuerung aller Messgeräte. Durch ständige Erweiterungen und Anpassungen an Kundenwünsche ist dieses Befehlsprogramm unübersichtlich und hinsichtlich der Testbarkeit schwierig zu handhaben.

In diesem Projekt werden erstmals benutzerdefinierte Datentypen verwendet. Jedoch sind diese benutzerdefinierten Datentypen mit kundenspezifischen Parametern versehen, oder sie beinhalten Parameter, die nicht zur allgemeinen Struktur beitragen. Beispielsweise gibt es eine Datentypstruktur für einen Fahrzyklus der zwar die Geschwindigkeits-, Schaltpunkt- und Steigungsprofile beschreibt, jedoch werden aber Strukturen der Gesetzgebung und Messgeräteeinstellungen ergänzt.

Abläufe wie z. B. die Kalibrierung von Messgeräten werden immer wieder neu implementiert, obwohl es sich um einen allgemeinen Ablauf des Messgerätes handelt, der gekapselt und wiederverwendet werden kann. Die Variablenstruktur, aber auch die allgemeine Ansteuerung der Messgeräte ist mit der Struktur der Testprozedur vermischt. Somit gibt es in diesem Tornado-Projekt keine klare Grenze, welche Konfigurationselemente dem Messgerät und welche der Testprozedur des Abgastests angehören.

In diesem Projekt werden spezielle kundenspezifische Fahrzyklen abgearbeitet, bei denen in einer nachfolgenden Phase eine Wiederbefüllung der Beutel ausgeführt wird. Beispielsweise wird ein dreiphasiger Prüfzyklus gefahren, bei dem in der ersten und dritten Phase das Beutelpaar 1 befüllt wird und in Phase 2 das zweite Beutelpaar. Somit ist das Füllvolumen des ersten Beutelpaares auf die erste und dritte Phase aufgeteilt.

In diesem Projekt ist eine weitere wichtige Erweiterung der normalen gesetzeskonformen Abgasprozedur zu finden. Dabei wird ein Fahrzyklus mehrere Male hintereinander abgefahren und ausgewertet. Diese Testprozedur wird für Hybridfahrzeuge verwendet, um die elektrische Reichweite der Fahrzeuge zu ermitteln.⁸⁷

Im Projekt des OEM werden die Messdaten sowohl während des Prüfzyklus als auch am Ende eines Tests ausgewertet. Die Auswertung der Messdaten geschieht zum größten Teil durch ein implementiertes Analyseprogramm, das die zyklische Messdatei am Ende des Prüfzyklus durchläuft und Berechnungsergebnisse (z. B. Mittelwert oder Integration) der Phasen ermittelt. Zusätzlich werden aber auch Messdaten während des Prüfzyklus mittels Tornado-Berechnungen ermittelt, wodurch schwer nachzuvollziehen ist, mit welcher Variante die Messergebnisse erfasst wurden.

Messgeräte

Bei den Abgasrollenprüfständen des OEM wird hauptsächlich Messtechnik der Firma K.K. Horiba verwendet. Die nachfolgende Tabelle gibt einen Überblick der verschiedenen Messgeräte:

⁸⁷ Vgl. ECE/TRANS/180/Add.15/Amend.4 (2018), S. 280 ff.

Messung	Gerätebezeichnung	Hersteller	Kommunikationsprotokoll
Abgaskonzentration	Horiba Mexa One	K.K. Horiba	AK-Protokoll
Abgasvolumen	Horiba CVS One	K.K. Horiba	AK-Protokoll
Partikelmasse über Filter	Horiba PMU One	K.K. Horiba	AK-Protokoll
Partikelanzahl	Horiba Mexa 2000 SPCS	K.K. Horiba	AK-Protokoll
Rohabgaskonzentration	Horiba Mexa One	K.K. Horiba	AK-Protokoll
Strom- /Spannungsmessung	HIOKI 3193	Hioki GmbH	SCP-Protokoll

Tabelle 10: Messgeräte des OEM, Quelle: Eigene Darstellung.

4.3 Auswertung und Vergleich der Projekte

Aus der Analyse der drei Tornado-Projekte im Bereich der Abgasprüfautomatisierung für Fahrzeuge sind folgende Schlüsse zu ziehen:

Einbindung von Fahrzyklen

In allen drei Projekten ist eine eigene Lösung zur Einbindung der Fahrzyklen bzw. der Geschwindigkeits- und Schaltpunktprofile zu finden. Die Funktionalität, Fahrzyklen über Betriebspunkte in ein Prüfprogramm zu importieren ist unübersichtlich und hinsichtlich Wartbarkeit schwer handzuhaben. Abgesehen davon, ist die Verwendung von Parametertabellen zur Einbindung der Fahrzyklen sinnvoller. Die Parametertabellen für das Geschwindigkeitsprofil, das Schaltpunktprofil und auch für das Steigungsprofil zu trennen, wie es bei dem Tornado-Projekt des OEM der Fall ist, ermöglicht es die Profile variabel zusammenzufügen. Die Profile nochmals phasenweise aufzuteilen würde die Einbindung der Fahrzyklen noch flexibler gestalten.

Flexibilität bei Änderung der Messgeräte

Bei allen drei Varianten sind die Konfigurationselemente der Messgeräte mit denen des Ablaufs der Testprozedur verwoben. Es gibt keine klare Abgrenzung bzw. Schnittstellen zwischen den Messgeräten und der Testprozedur der Abgasprüfzyklen. Des Weiteren ist die neue Implementierung und der Austausch von Messgeräten in den Projekten schwer handzuhaben. Eine klare Aufteilung der Messgeräte und der Testprozedur ist in der Entwicklung des neuen standardisierten Abgasprüfautomatisierungsmoduls anzustreben.

Auswertung der Messdaten

In allen drei Projekten werden verschiedene Varianten zur Aufzeichnung und Auswertung der relevanten Messdaten herangezogen. Aus der Analyse geht hervor, dass es möglich ist, die Auswertung der Messdaten in Tornado zu implementieren. Die im Projekt des OEM implementierte Variante, die Auswertung aus der zyklischen Messdatei zu generieren, ist für eine Nachvollziehbarkeit der Messdaten am sinnvollsten. Somit soll diese Variante in das standardisierte Abgasprüfautomatisierungsmodul übernommen werden.

Testbarkeit der Prüfprozedur

In den Tornado-Projekten der TU Graz und der FH Joanneum werden für die verschiedenen Fahrzyklen separate Prüfprogramme angelegt. Jedes davon beinhaltet die Ansteuerung der Messgeräte und den Ablauf der Phasen bzw. des Fahrzyklus. Somit muss jedes Prüfprogramm für sich getestet werden, um sicherzustellen, dass der Ablauf in jeder Variante funktioniert. Beim Tornado-Projekt des OEM wird der Ablauf zwar zentral auf einem Prüfprogramm und einem Steuerungsprogramm abgearbeitet, jedoch ist der Programmaufbau unübersichtlich und für Neuerungs- und Wartungsarbeiten schwierig zu handhaben. Im neuen Abgasprüfautomatisierungsmodul muss somit eine strukturierte Variante der Ablaufsteuerung gestaltet werden. Durch eine Kapselung der Abläufe sollen einzelne Teile des Programmablaufs separat und unabhängig überprüfbar sein.

Datenstruktur

Erst mit der Einbindung von benutzerdefinierten Datentypen in Tornado konnte eine strukturierte Ordnung der Daten bzw. der Variablen in Tornado geschaffen werden. Im Tornado-Projekt des OEM lassen sich gute Ansätze von benutzerdefinierten Datentypen finden, mit denen die Grundstruktur eines Abgasrollenprüfstands aufgebaut werden kann. Jedoch müssen die benutzerdefinierten Datentypen des neu zu entwickelnden Abgasprüfautomatisierungsmoduls eine strikte Trennung zwischen Ablauf, Messaufgabe, Messgeräten etc. beinhalten.

Abhängigkeit der Rolle während der Prüfprozedur

In allen drei Projekten dient die Rolle am Prüfstand, aus Sicht der Abgasautomatisierung, nur zur Aufzeichnung der Geschwindigkeit und der gefahrenen Strecke des Fahrzeugs während des Fahrzyklus. Deshalb kann das System der Rolle vom eigentlichen Prüfprozess der Abgassysteme getrennt werden. Über eine Schnittstelle muss lediglich die Geschwindigkeit bzw. die gefahrene Strecke übergeben werden.

5 KONZEPTIONIERUNG DES SOFTWAREMODULS

Im folgenden Kapitel wird die Konzeptionierung des Abgasprüfautomatisierungsmoduls gemäß den ersten Schritten des Softwareentwicklungszyklus erarbeitet. Zunächst werden die Planung und Analyse der Anforderungen, also das Requirements-Engineering, durchgeführt. Im Anschluss werden mittels der grafischen Hilfsmittel der Softwarearchitektur Entwürfe der Datenstruktur und des Programmaufbaus entwickelt.

5.1 Wesentliche Anforderungen an das Softwaremodul

Gemäß den Aufgaben des Requirements-Engineerings (siehe Kapitel 3.2.1) wurden aus dem Kapitel 1 die allgemeinen Anforderungen aus den Richtlinien und Vorgaben der Abgasprüfautomatisierung für Fahrzeuge ermittelt. Auch eine Analyse der Bestandsysteme (siehe Kapitel 4.2) hat wichtige Informationen geliefert und neue Funktionalitäten aufgedeckt, die in der Entwicklung und Anforderungsspezifikation des Softwaremoduls mitberücksichtigt werden. In Kapitel 4.3 werden Vor- und Nachteile der bestehenden System zusammengefasst und mögliche Verbesserungen definiert. Die folgenden Kapitel beinhalten die Dokumentation und Zusammenfassung der wesentlichen Anforderungen bzw. Spezifikationen für das Abgasprüfautomatisierungsmodul. Wie schon in Kapitel 3.2.1 erwähnt, kann die Dokumentation in grafischer oder natürlicher Sprache erfolgen. Für die nachfolgenden Spezifikationen wird die Variante der natürlichen Sprache verwendet, um die erforderlichen Details der Anforderungen besser festlegen zu können. Die Idee der Sprachschablonen wird verwendet, um eine kurze und prägnante Definition der Anforderungen zu ermöglichen. Des Weiteren wird zur besseren Übersichtlichkeit und Verwaltung festgelegt, dass die zu dokumentierenden Anforderungen in tabellarischer Form aufgelistet und fortlaufend nummeriert werden.

Nach der Abstimmung der Anforderungen mit den Stakeholdern der Firma Kristl, Seibt & Co GmbH und einem Review des Erstentwurfs der Anforderungen werden folgende funktionale und nicht funktionale Anforderungen für das Abgasprüfautomatisierungsmodul (APAM) fixiert.

5.1.1 Funktionale Anforderungen

Nr.	Anforderung
1	Das APAM soll in Tornado entwickelt werden.
2	Das APAM soll als allgemeines Modul in das Tornado-Masterprojekt TORMASTER implementiert werden.
3	Das APAM soll durch einfaches Importieren des Moduls in Tornado-Projekte für Rollenprüfstände und Antriebstrangprüfstände eingebunden werden können.
4	Das APAM soll alle in Kapitel 2.1.1 erwähnten Richtlinien und Verordnungen berücksichtigen.
5	Das APAM soll alle in Kapitel 2.1.2 erwähnten Grenzwertsätze beinhalten.
6	Das APAM soll alle in Tabelle 4 und Tabelle 7 beinhalteten Fahrzyklen abarbeiten können.

7	Die Konfigurationselemente der Messgeräte in Tornado sollen separat und unabhängig vom Abgasprüfablauf aus dem Tornado-Masterprojekt zu exportieren sein. Anmerkung: Hintergrund ist, dass diese Messgeräte auch unabhängig vom Abgasprüfablauf in anderen Projekten verwendet werden sollen.
8	Das APAM soll alle erforderlichen Messmethoden und Messgeräte aus Kapitel 2.2 beinhalten.
9	Das APAM soll alle in Kapitel 4.2 erwähnten Messgeräte ansteuern können (siehe Tabelle 8, Tabelle 9, Tabelle 10).
10	Die Parametertabellen der Fahrprofile sollen in phasenbezogenen Geschwindigkeits-, Schaltpunkt- und Steigungstabellen implementiert werden.
11	Das APAM soll eine gekapselte Struktur zwischen den Messgeräten und des Prüfprozesses beinhalten.
12	Im APAM soll die Auswertung der Messdaten aus der zyklischen Messdatei heraus erfolgen.

Tabelle 11: Funktionale Anforderungen, Quelle: Eigene Darstellung.

5.1.2 Qualitätsanforderungen

Nr.	Anforderung
1	Neue Fahrzyklen müssen für den Benutzer unkompliziert und mit wenig Aufwand implementierbar sein.
2	Das APAM soll eine flexible Umstellung der Messgeräte im Tornado-Projekt gewährleisten. Anmerkung: Wenn z. B. ein Messgerät durch ein Messgerät mit gleicher Funktionalität eines anderen Herstellers ausgetauscht wird.
3	Das APAM soll durch einen strukturierten Aufbau hinsichtlich Testbarkeit und Erweiterbarkeit besser als die bestehenden Projekte handzuhaben sein.

Tabelle 12: Qualitätsanforderungen, Quelle: Eigene Darstellung.

5.2 Entwurfsphase des Abgasprüfautomatisierungsmoduls

In der nächsten Phase der Konzeptionierung sollen mit den beschriebenen Hilfsmitteln der Softwarearchitektur aus Kapitel 3.3, Lösungen für die Struktur und den Ablauf des Softwaremoduls entwickelt werden. Im Unternehmen Kristl, Seibt & Co GmbH werden für die Entwicklung und Erstellung von Softwarekonzepten zwei Modellierungswerkzeuge verwendet. Es handelt sich um die Programme Enterprise Architect und Microsoft Visio. Da Enterprise Architect eine Vielzahl an Hilfsmitteln zur Erstellung von Architekturmodellen für die Softwareentwicklung bereitstellt, wird die Konzeptionierung mit diesem Programm vorgenommen.

Für den Entwurf der Softwarearchitektur kommen folgende, aus dem Kapitel 3.3, erarbeitete Modellierungssprachen und Diagrammtypen in Frage:

- Unified Modelling Language
- System Modelling Language
- Entity-Relationship-Model
- Fundamental Modelling Concepts

Sowohl die UML- und SysML-Diagramme als auch das ER-Modell werden von Enterprise Architect unterstützt. Die Diagrammtypen des FMC werden von Enterprise Architect nicht bereitgestellt. Die Modellierungssprachen SysML und FMC dienen hauptsächlich zur Beschreibung der Systemebene einer Software. Die Modellierungssprache UML und das ER-Modell fokussieren sich auf die Konzeptionierung der Struktur und des Ablaufs einer Software bzw. eines Programms. Im Rahmen dieser Arbeit soll der Aufbau und Abfolge des Prüfzyklus des Abgasprüfautomatisierungsmoduls erstellt werden, womit die UML-Diagrammfamilie und das ER-Modell besser geeignet sind als die Diagrammtypen der FMC und SysML. Die UML bietet im Vergleich zum ER-Modell eine größere Auswahl an Diagrammtypen, mit denen sowohl die Struktur als auch das Verhalten modelliert werden können. Ein weiterer wesentlicher Vorteil der UML-Diagrammfamilie ist, dass diese Modellierungssprache sich in der Softwareentwicklung etabliert hat und durch seine große Community etwaige Probleme schnell gelöst werden können. Aus diesen Gründen wird die Architektur des Abgasprüfautomatisierungsmoduls mittels UML modelliert.

5.2.1 Strukturentwurf

Ausgehend vom 4+1-Sichtenmodell der Softwarearchitektur wird zunächst die logische Sicht der Softwarearchitektur betrachtet. Ausgehend von dieser Betrachtung wird die Struktur bzw. der Aufbau des zu entwickelnden Softwaremoduls erstellt. In den folgenden Kapiteln werden Konzepte für die Struktur der Messgeräte und des Abgasprüfautomatisierungsmoduls erarbeitet. Die Modelle und die darin verwendeten Elemente werden mit den englischen Begriffen bezeichnet, da die Implementierung der Variablen im TORMASTER-Projekt auch in dieser Sprache stattfinden soll.

5.2.1.1 Positionierung im TORMASTER-Projekt

Das TORMASTER-Projekt ist in drei Hauptbereiche unterteilt und jedes Konfigurationselement wird einem von ihnen zugeordnet. Diese Bereiche, oder in Tornado auch benutzerdefinierte Gruppen genannt, werden als Geräte, Module und System bezeichnet. Unter den Geräten sind alle Konfigurationselemente enthalten,

die zur Kommunikation und Steuerung von Messgeräten oder externen Softwaresystemen benötigt werden. Unter Modulen werden verschiedene Funktionen, Prozesse oder Hilfsmittel, die nicht direkt einem Gerät zuzuordnen sind, gesammelt. Unter System sind die zu Beginn eines Tornado-Projekts bestehenden Systemkonfigurationselemente gruppiert. Abbildung 18 zeigt die benutzerdefinierten Gruppen im TORMASTER-Projekt.

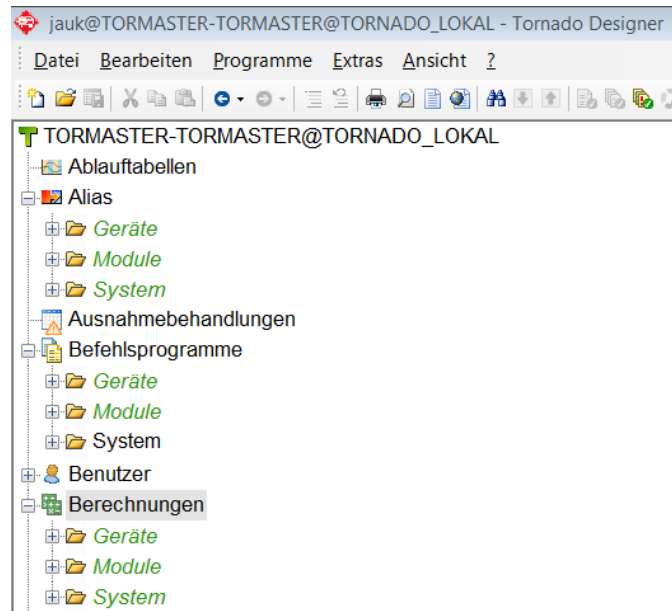


Abbildung 18: Allgemeiner Aufbau des TORMASTER-Projekts. Quelle: Eigene Darstellung.

Um eine Kapselung der verwendeten Messgeräte und dem Prüfprozess des Abgasprüfautomatisierungsmoduls zu schaffen, werden alle Funktionalitäten der Messgeräte unter der Gruppierung Geräte gesammelt und der Prozess des Abgasprüfautomatisierungsmoduls wird unter der Gruppierung Module aufgebaut.

5.2.1.2 Struktur der Messgeräte

Im ersten Schritt der Konzeptionierung werden die Struktur und der Aufbau der Variablen sowie die notwendigen Funktionen der Messgeräte erarbeitet. In Kapitel 2.2 wurden die erforderlichen Messgeräte für einen gesetzeskonformen Abgasprüfzyklus ermittelt. Des Weiteren wurden die verschiedenen Hersteller und Gerätetypen an bereits vorhandenen Prüfständen erarbeitet. Die Messgeräte der unterschiedlichen Hersteller haben hinsichtlich Kommunikation, Messwerte und Ansteuerung viele Gemeinsamkeiten, sodass eine Generalisierung der Messgeräte definiert werden kann. Grundsätzlich besitzt jeder Prüfstand folgende Messgerätetypen:

- Analysator zur Analyse der verdünnten Abgase
- Gerät zur Aufbereitung bzw. Verdünnung des Abgases
- System zur Beladung der Abgaspartikel auf Filtern
- System zur Messung der Partikelanzahl
- Messgerät zur Erfassung der Spannung und des Stroms von Batteriesystemen

Zusätzlich werden noch folgende Messgerätetypen vereinzelt eingesetzt:

- Analysator zur Analyse der unverdünnten Abgase
- Messung weiterer Abgaskomponenten mittels des FTIR-Prinzips

In der nachfolgenden Tabelle werden die generalisierten Bezeichnungen der Messgerätetypen aufgelistet.

Abkürzung	Langform	Beschreibung
EMSDIL	Emission Measurement System Diluted	Analysatoreinheit zur Analyse der verdünnten Abgase
CVS	Constant Volume System	Gerät zur Aufbereitung bzw. Verdünnung des Abgases
PFS	Particle Filtering System	System zur Beladung der Abgaspartikel auf Filtern
PCS	Particle Counting System	System zur Messung der Partikelanzahl
EPA	Electrical Power Analyser	Messgerät zur Erfassung der Spannung und des Stroms von Batteriesystemen
EMSRW	Emission Measurement System Raw	Analysator zur Analyse der unverdünnten Abgase (Rohabgas)
FTIR	Fourier transform infrared spectroscopy	Einheit zur Messung weiterer Abgaskomponenten mittels FTIR

Tabelle 13: Nomenklatur der Messgerätetypen, Quelle: Eigene Darstellung.

Diese Abkürzungen sollen sowohl in der Entwurfsphase als auch in der späteren Implementierungsphase als Präfix für Variablen, Befehlsprogramme und weitere benötigte Konfigurationselemente der Messgeräte dienen.

Im Enterprise-Architect-Projekt wird mittels sogenannter Pakete die gleiche strukturelle Trennung von Geräten und Modulen erstellt. Das Paket mit dem Titel Geräte enthält wiederum die in Tabelle 13 erwähnten Gerätebezeichnungen als Unterpakete. Mit Hilfe eines UML-Paketdiagramms kann die Gliederung der Gerätestruktur übersichtlich dargestellt werden (siehe Abbildung 19).

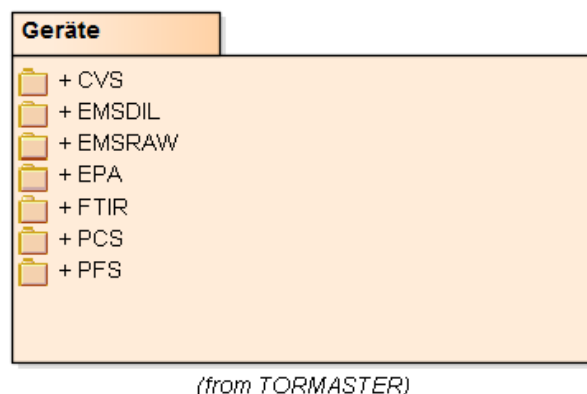


Abbildung 19: Paketdiagramm der Geräte, Quelle: Eigene Darstellung.

Nachdem die allgemeine Gruppierung und Nomenklatur der Messgerätetypen definiert sind, wird die Struktur der Messgerätetypen konzeptioniert. Als abstrakte Basisklasse wird die Klasse Device festgelegt. Da jedes Gerät in Tornado mit einem Treiber verknüpft ist, beinhaltet die Basisklasse Device zusätzlich die innere Klasse Driver. Abbildung 20 zeigt das in Enterprise Architect modellierte UML-Klassendiagramm der Struktur eines allgemeinen Geräts.

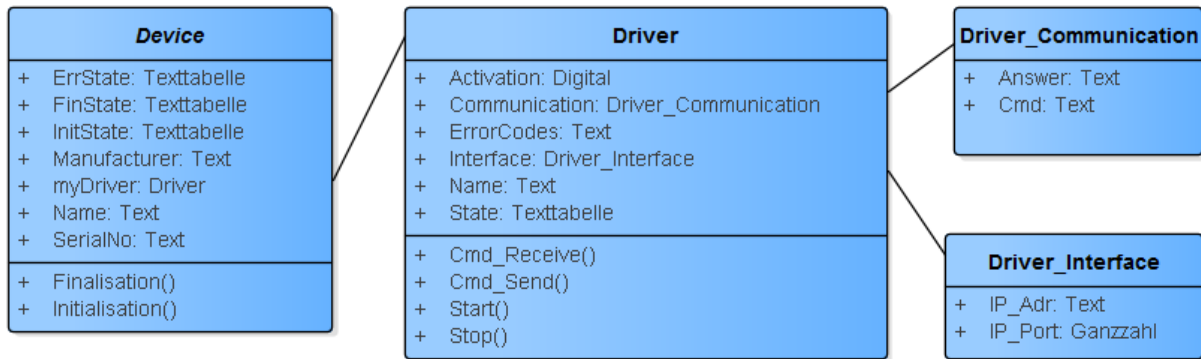


Abbildung 20: Klassendiagramm Device, Quelle: Eigene Darstellung.

Jeder Treiber beinhaltet zusätzlich Klassen, die die Attribute der Schnittstelle (Interface) und der Kommunikation (Communication) beschreiben. Zur besseren Nachvollziehbarkeit in der späteren Implementierungsphase werden in diesem EA-Projekt die Grunddatentypen von Tornado hinterlegt. In Tabelle 14 werden die verwendeten Attribute der Abbildung 20 mit einer kurzen Beschreibung aufgelistet.

Attribut	Beschreibung
Device.Name	Beinhaltet den Namen des Geräts
Device.SerialNo	Beinhaltet die Seriennummer des Geräts
Device.Manufacturer	Beinhaltet den Herstellernamen des Geräts
Device.InitState	Gibt den Status der Initialisierung des Geräts an
Device.FinState	Gibt den Status der Finalisierung des Geräts an
Device.ErrState	Gibt an, wenn in der Kommunikation des Geräts ein Fehler aufgetreten ist
Driver.Activation	zur Aktivierung und Deaktivierung des Treibers in Tornado
Driver.Name	Der Name des Treiber-Konfigurationselements wird hinterlegt
Driver.State	Gibt den Status des Treibers wieder. Die Texttabelle ist ein Systemelement von Tornado und enthält alle Zustände eines Treibers
Interface.IP_Adr	Beinhaltet die IP-Adresse des zu kommunizierenden Gerätes bzw. Treibers
Interface.IP_Port	Beinhaltet den IP-Port des zu kommunizierenden Gerätes bzw. Treibers
Communication.Cmd	Beinhaltet den zu sendenden Treiberbefehl
Communication.Answer	Beinhaltet die Antwort eines Treiberbefehls

Tabelle 14: Attribute des Klassendiagramms Device, Quelle: Eigene Darstellung.

Die Methoden Cmd_Receive() und Cmd_Send() sollen darstellen, dass jeder Treiber die Möglichkeit besitzt, Befehle an des Gerät zu senden und zu empfangen. Die Methoden Start() und Stop() sollen zur Aktivierung und Deaktivierung des Treibers dienen. Jedes Gerät besitzt die Möglichkeit einer Initialisierung

und Finalisierung. Diese Funktion wird mit den Methoden Initialisation() und Finalisation() der Klasse Device definiert.

Im nächsten Schritt der Entwurfsphase der Messgeräte werden die definierten Messgerätetypen aus der Basisklasse Device abgeleitet. Jeder Messgerätetyp beinhaltet eine Übermenge an Messwerten und beschreibenden Parametern. Des Weiteren kann jeder Messgerätetyp mit einem bestimmten Kommunikationsprotokoll gesteuert und ausgelesen werden. Zudem besitzt jeder Hersteller eines Gerätetyps eine bestimmte Befehlssyntax zur Ausführung von Aktionen und zum Auslesen von Werten des Messgeräts. Im Erstentwurf (siehe Abbildung 21) werden diese strukturellen Überlegungen anhand des Messgerätetyps CVS in einem Klassendiagramm dargestellt.

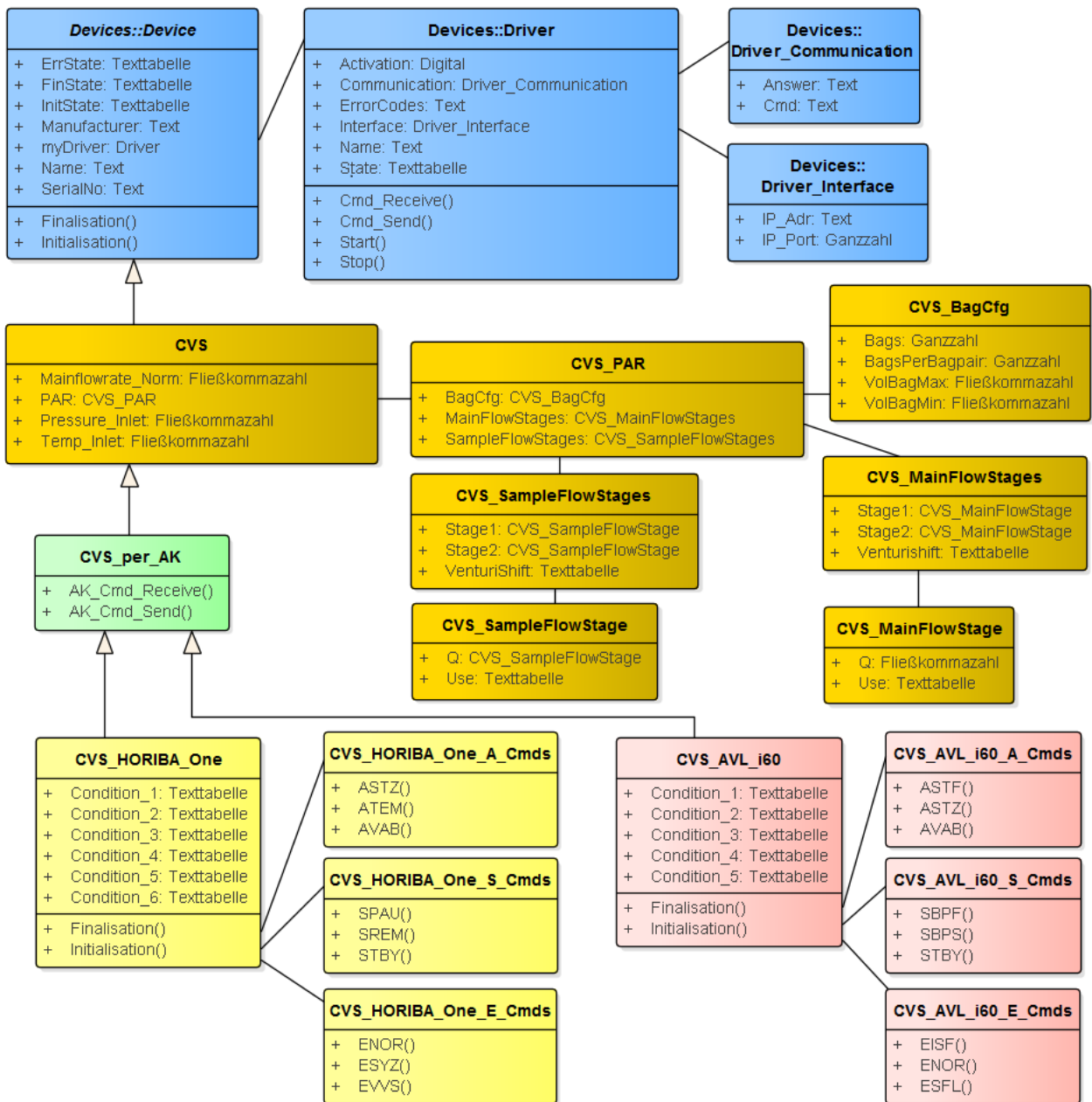


Abbildung 21: Erstentwurf der Struktur für den Gerätetyp CVS, Quelle: Eigene Darstellung.

In Abbildung 21 ist ersichtlich, dass die Klasse des Messgerätetyps wiederum von der Basisklasse Device abgeleitet wird. Grundsätzlich können mit dieser Struktur die Kommunikation und die Beschreibung eines

bestimmten Messgerätetyps abgebildet werden, wobei in vielen Anwendungsfällen in Tornado lediglich die Struktur der Kommunikation benötigt wird. Die Beschreibung der Messgeräte ist vor allem bei der Verwendung von speziellen Prüfläufen interessant, da aus diesen Informationen ein automatisierter Ablauf erstellt werden kann. Aus diesem Grund wird entschieden, die Struktur der Kommunikation und die der Beschreibung des Messgeräts zu trennen. Die Beschreibung der Messgeräte wird in einer separaten Struktur aufbereitet (siehe Kapitel 5.2.1.3).

Im Verlauf dieses Kapitels werden die Klassen zum Aufbau der Kommunikation der Messgerätetypen beschrieben. Es werden lediglich die Strukturen der Messgerätetypen CVS, EMSDIL, PFS und PCS modelliert, da die Hauptstrukturen der anderen Geräte ähnlich aufgebaut sind.

CVS

Im Klassendiagramm der CVS (siehe Abbildung 22) werden die Attribute und Methoden dargestellt, mit denen der Aufbau der allgemeinen Gerätekommunikation möglich ist. Alle beschreibenden Attribute der Geräte (z. B. SerialNo aus der Basisklasse Device) und die des Messgerätetyps CVS (Klasse CVS_PAR) werden nicht benötigt. Da der Treiber ein wesentliches Element der Kommunikation in Tornado ist, wird dieser in der Basisklasse Device beibehalten. Die Klasse CVS beschreibt die abgeleitete Klasse des Messgerätetyps CVS. In ihr sind alle relevanten Messdaten des Messgerätetyps CVS enthalten. In der abgeleiteten Klasse CVS_per_AK werden die Methoden AK_Cmd_Receive() und AK_Cmd_Send() implementiert, in der die Funktionen zum Senden, Empfangen und Verarbeiten von AK-Befehlen integriert sind. Darunterliegend sind die herstellerspezifischen Klassen CVS_HORIBA_One und CVS_AVL_i60 angeordnet. Messgeräte, die über das AK-Kommunikationsprotokoll gesteuert werden, besitzen sogenannte Zustände, die den momentanen Zustand eines Messgeräts beschreiben. Da die Zustände je nach Hersteller variieren können, werden sie (Condition_x) erst in der Subklasse der herstellerspezifischen Gerätebeschreibung definiert. Jedes herstellerspezifische Messgerät, das mit dem AK-Kommunikationsprotokoll gesteuert wird, hat eine Methodensammlung aus Antwort-, Steuer- und Einstellungsbefehlen. Diese werden gesondert in eigenen Klassen (*_A_Cmds, *_S_Cmds, *_E_Cmds) gruppiert. Die Methoden Initialisation() und Finalisation() der Basisklasse Device werden in den herstellerspezifischen Klassen CVS_HORIBA_One und CVS_AVL_i60 überschrieben, da sich die Befehle in diesen Abläufen herstellerspezifisch ändern können.

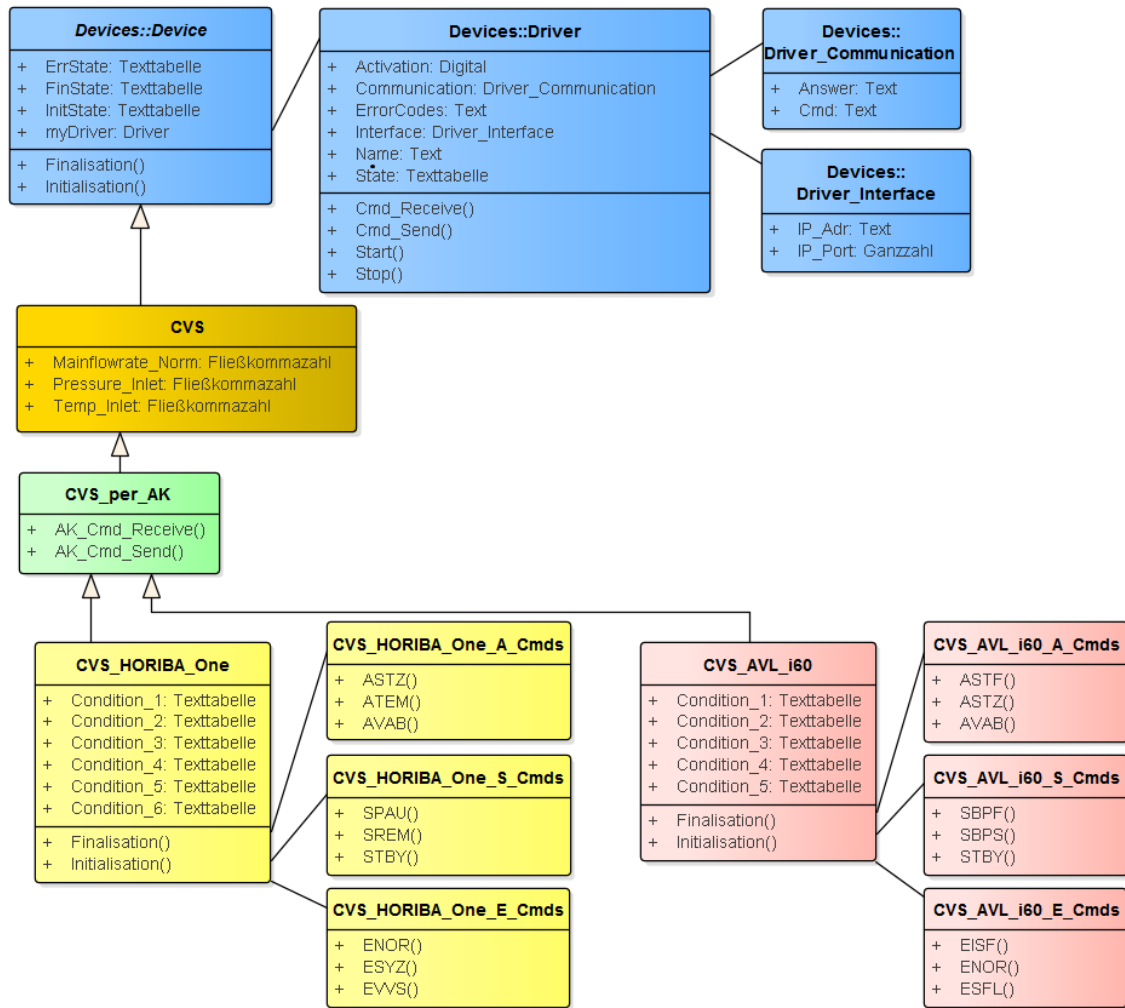


Abbildung 22: Klassendiagramm CVS, Quelle: Eigene Darstellung.

In der nachfolgenden Tabelle werden die Attribute der Klasse CVS aus dem Klassendiagramm der CVS (Abbildung 22) näher erläutert.

Attribut	Beschreibung
Mainflowrate_Norm	Normierter Durchfluss
Pressure_Inlet	Aktueller Druck im Einlass
Temp_Inlet	Aktuelle Temperatur im Einlass

Tabelle 15: Attribute des Klassendiagramms CVS, Quelle: Eigene Darstellung.

EMSDIL

Die Klasse der Analyatoreinheit zur Messung des verdünnten Abgases, kurz EMSDIL, hat den gleichen grundsätzlichen Aufbau der Kommunikationsstruktur. In Abbildung 23 ist das Klassendiagramm bezogen auf den Messgerätetyp EMSDIL dargestellt. In der Klasse EMSDIL werden wieder die relevanten Messdaten dieses Messgerätetyps gesammelt. Bei diesem Gerätetyp werden die Konzentrationen und weitere Messwerte immer einer bestimmten Komponentenummer zugewiesen. Jede Komponentenummer wird wiederum einer bestimmten Abgaskomponente zugeordnet. Dieser

Zusammenhang kann mit dem AK-Kommando AKFG (z. B. AKFG 0 CO2 K1 CO K2 CH4 K3 THC ...) aus dem Gerät ausgelesen werden. Des Weiteren wird jede Komponentenummer einer bestimmten Linie zugewiesen. Auch dieser Zusammenhang kann ausgelesen werden (z. B. AKFG 0 KV L1 K1 K2 K3). Somit sind alle komponentenbezogenen Messwerte immer einer Linie und einer bestimmten Abgaskomponente zugeordnet. Beispielsweise ergeben die AKFG-Antworten ‚AKFG 0 THC K1 CH4 K2 CO2 K3 CO K4‘ und ‚AKFG 0 KV L1 K1 K2 L2 K3 K4‘ die Belegung der Variable ‚EMSDIL.L2.CO2.Concentration‘.

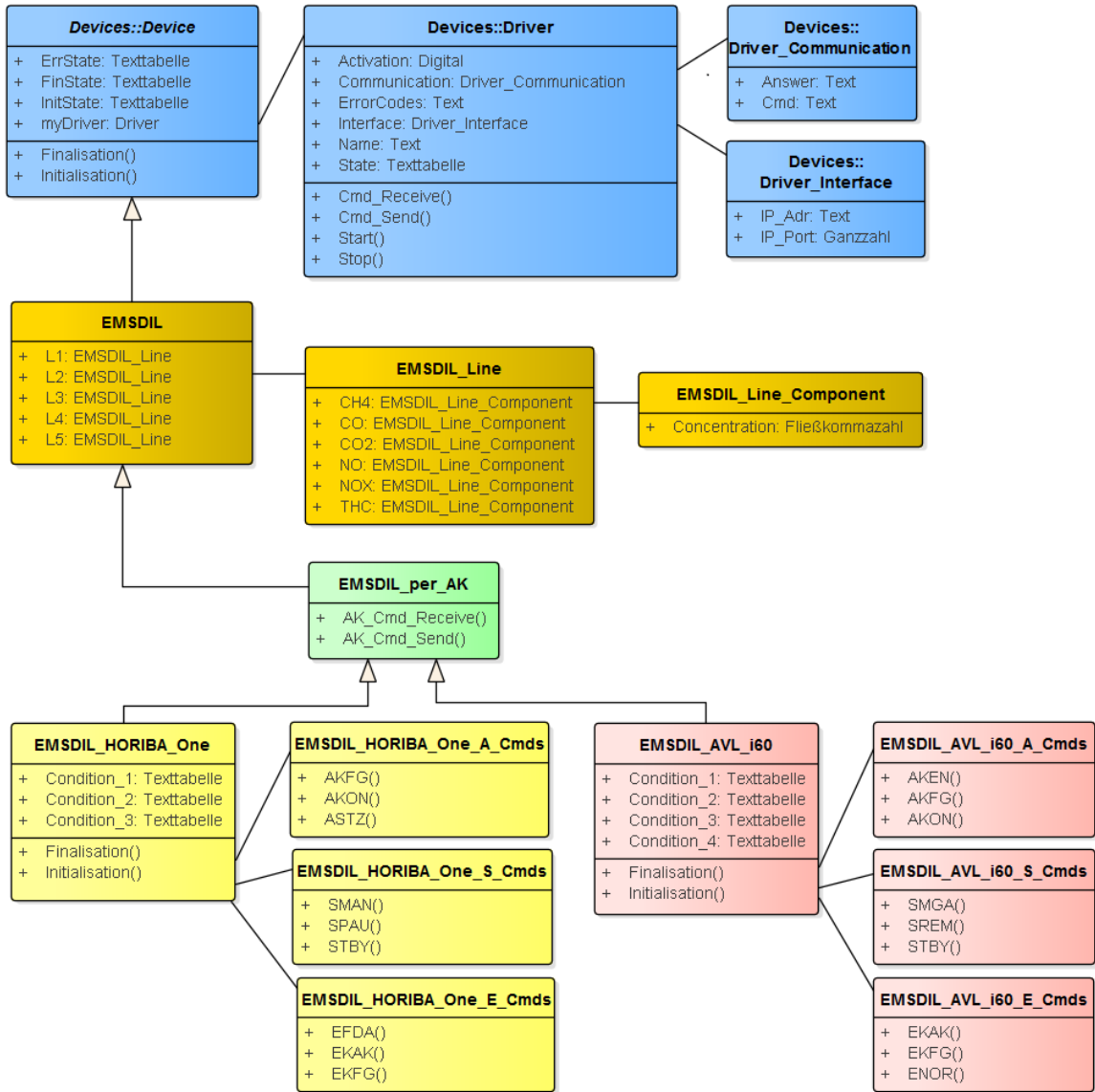


Abbildung 23: Klassendiagramm EMSDIL, Quelle: Eigene Darstellung.

In der Tabelle 16 werden die Attribute der Klasse EMSDIL aus dem Klassendiagramm der EMSDIL (Abbildung 23) näher erläutert.

Attribut	Beschreibung
Lx.y.Concentration	Konzentration der Komponente y (z. B. CO, CO2 etc.), auf der Linie x (z. B. 1 – 5)

Tabelle 16: Attribute des Klassendiagramms EMSDIL, Quelle: Eigene Darstellung.

PCS

Im Klassendiagramm des Partikelzählersystems (siehe Abbildung 24) ist die Unterscheidung zwischen einem Gerät mit AK-Protokoll und TSI-Protokoll ersichtlich. Auch bei dem TSI Gerät werden die Befehle in Methoden für das Steuern (Set) und das Lesen (Read) gruppiert. Desweiteren werden in der Unterklasse PCS_per_TSI die Methoden zum Senden und Empfangen von TSI-Befehlen angepasst.

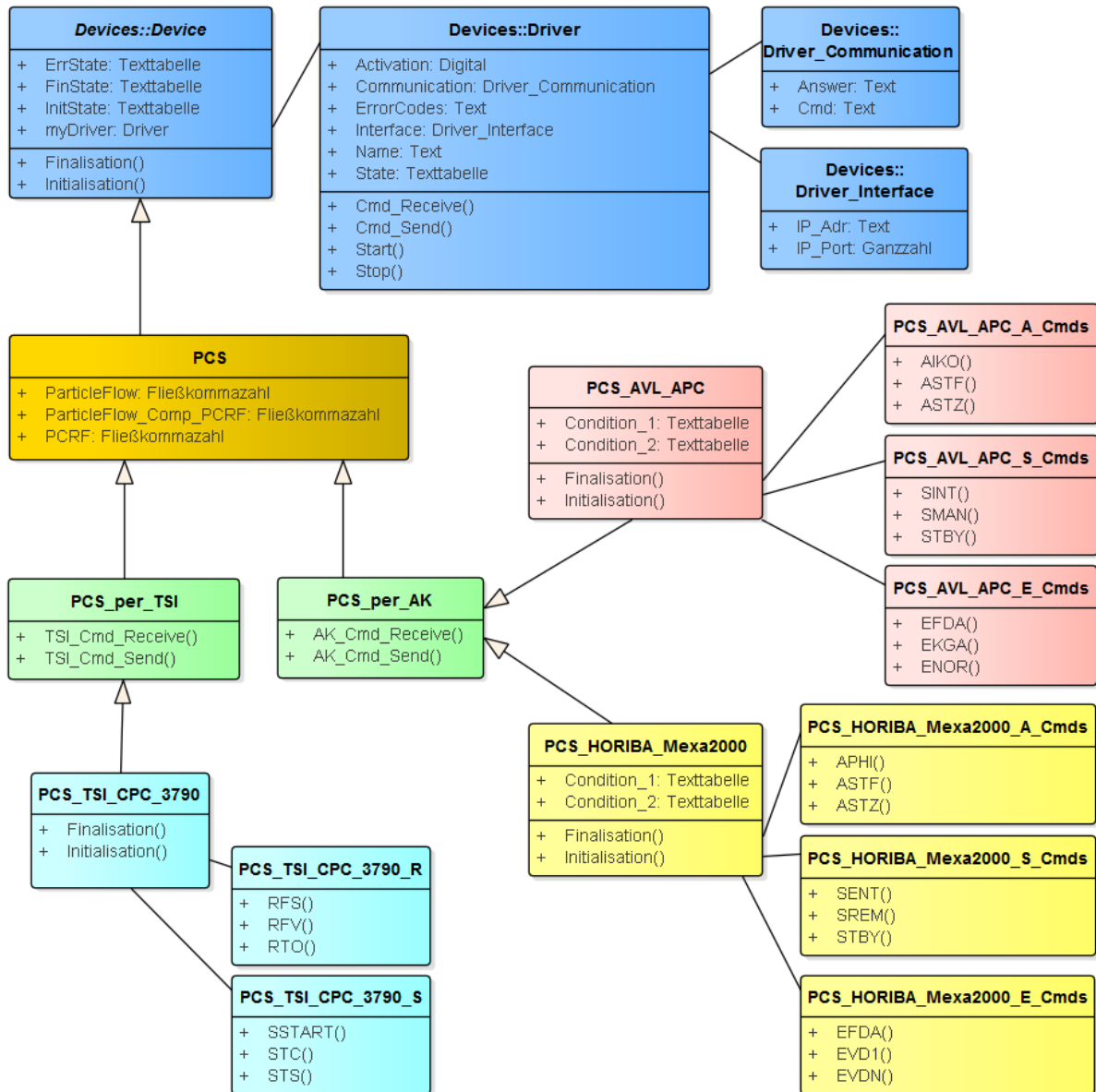


Abbildung 24: Klassendiagramm PCS, Quelle: Eigene Darstellung.

In der nachfolgenden Tabelle werden die Attribute der Klasse PCS aus dem Klassendiagramm der PCS (Abbildung 24) näher erläutert.

Attribut	Beschreibung
ParticleFlow	Aktueller Partikelvolumenstrom
ParticleFlow_Comp_PCRF	Aktueller Partikelvolumenstrom, korrigiert um den Reduktionsfaktor
PCRF	Partikelreduktionsfaktor (Engl. particle concentration reduction factor)

Tabelle 17: Attribute der Klasse PCS, Quelle: Eigene Darstellung.

PFS

Im Aufbau der Klassenstruktur für das Partikelfiltersystem findet sich der gleiche Strukturaufbau wie bei den zuvor erstellten Diagrammen wieder (siehe Abbildung 25). Da im Projekt der TU Graz dieser Messgerätetyp über eine SPS angesteuert wird und dafür kein befehlsbasiertes Kommunikationsprotokoll notwendig ist, besitzt die Klasse PFS_TU_Grz keine weiteren Gruppierungen von Methoden.

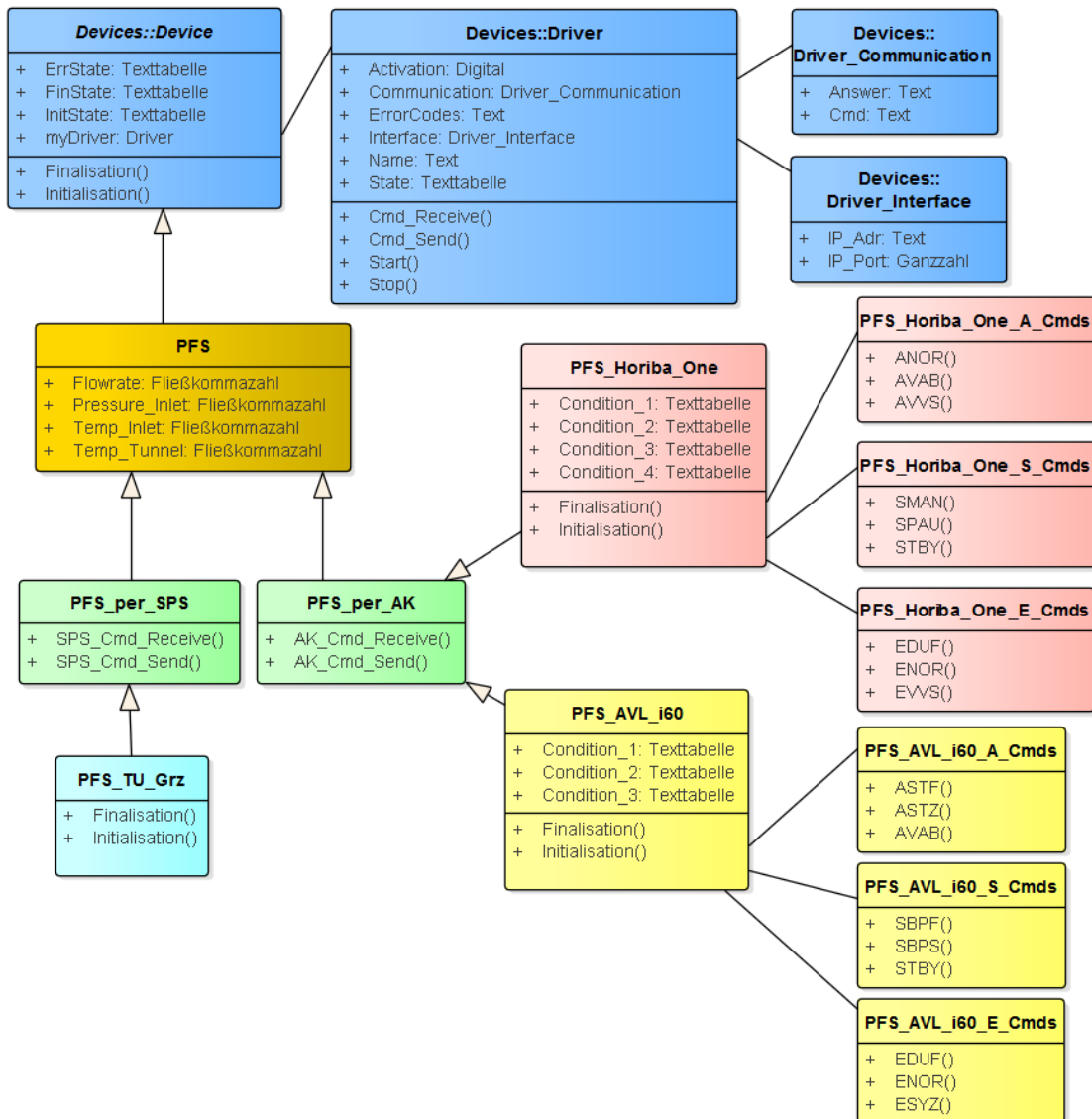


Abbildung 25: Klassendiagramm PFS, Quelle: Eigene Darstellung.

Die nachfolgende Tabelle beinhaltet die Attribute der Klasse PFS mit einer entsprechenden Beschreibung.

Attribut	Beschreibung
Flowrate	Durchfluss am Partikelfilter
Pressure_Inlet	Druck am Einlass
Temp_Inlet	Temperatur am Einlass
Temp_Tunnel	Temperatur im Tunnel

Tabelle 18: Attribute der Klasse PFS, Quelle: Eigene Darstellung.

5.2.1.3 Struktur des Abgasprüfautomatisierungsmoduls

Im nächsten Schritt der Konzeptionierung wird die Struktur des Abgasprüfautomatisierungsmoduls entwickelt, die in zwei Bereiche aufgeteilt wird. Der erste Bereich umfasst die Messgeräteeinbindung in das Abgasprüfautomatisierungsmodul und beinhaltet wiederum zwei Strukturen. Die erste Struktur definiert den grundsätzlichen Aufbau des Prüfstands und somit die verbauten Messgeräte. Die zweite Struktur soll die Entscheidungsgrundlage bei den Fragen sein, mit welchen Messgeräten und mit welcher Einstellung der Prüfzyklus durchlaufen wird. Die beiden Strukturen werden folgendermaßen definiert:

- Prüfstandsbeschreibung
- Geräteeinstellung

Der zweite Bereich deckt die Definition des Abgasprüfzyklus ab. Dabei beinhaltet eine Struktur den Fahrzyklus und definiert somit den grundsätzlichen Ablauf des Prüfzyklus, während eine weitere Struktur die Informationen der Gesetzgebung, des verwendeten Kraftstoffes usw. umfasst. Die Strukturen werden folgendermaßen benannt:

- Fahrzyklus
- Testaufgabe

Um eine Unterscheidung des Abgasprüfautomatisierungsmoduls zu den anderen Modulen im TORMASTER-Projekt zu gewährleisten, wird ein geeignetes Präfix definiert. Das Modul wird mit als Tornado Emission Measurement for Vehicle (TEMV) bezeichnet. Alle verwendeten Konfigurationselemente (Variablen, Befehlsprogramme etc.) werden mit diesem Präfix versehen. Die erwähnten Strukturen werden im weiteren Verlauf dieses Kapitels mittels Klassendiagrammen näher erläutert.

Prüfstandsbeschreibung

Wie schon in Kapitel 5.2.1.2 erwähnt, ist eine Trennung der Kommunikations- und Beschreibungsstruktur der Messgerätetypen vorgesehen. In der Datenstruktur der Prüfstandsbeschreibung finden sich sowohl allgemeine Informationen des Prüfstands wieder, aber auch die Beschreibungsstruktur zu den einzelnen Messgerätetypen. Auch hier ist das Ziel, eine allgemeine, herstellerunabhängige Struktur zu schaffen. In dieser Struktur sollen Informationen gesammelt werden, die nicht z. B. per Befehl aus dem Messgerät ausgelesen werden können, jedoch wichtig für die Ansteuerung sind.

In Abbildung 26 ist die Struktur eines Abgasprüfstands dargestellt. Superklasse ist die Klasse Testbed, die wiederum die Klasse Devices enthält. Unter der Klasse Devices werden die Beschreibungen aller Messgeräte gesammelt. Jeder Messgerätetyp hat als Basisklasse die Klasse Device, die alle allgemeinen Attribute eines Geräts umfasst. In Abbildung 26 ist nun auch die Beschreibungsstruktur des Messgerätetyps CVS enthalten, die zunächst in der Struktur des allgemeinen Messgerätetyps konzeptioniert wurde (siehe Abbildung 21).

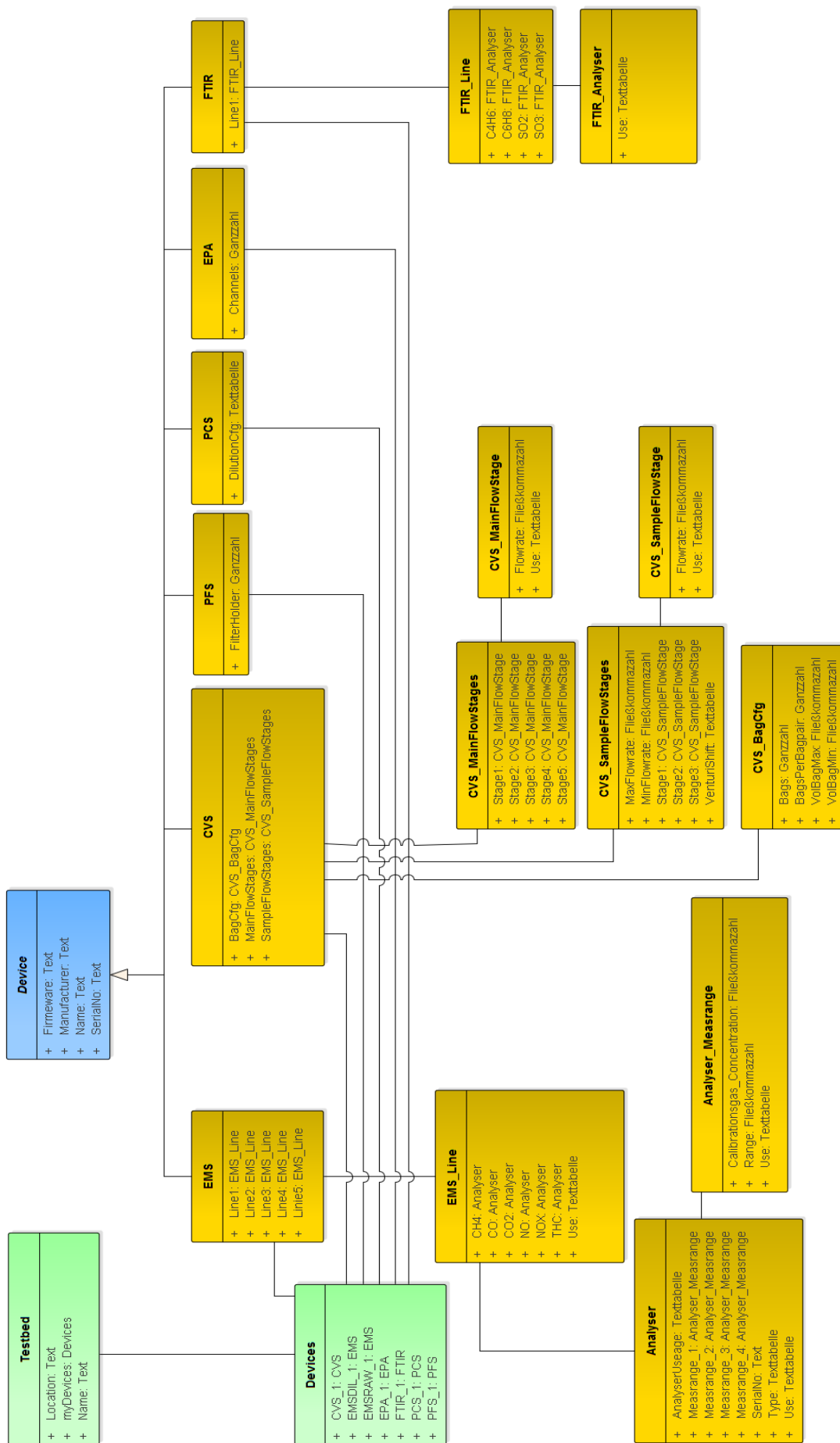


Abbildung 26: Klassendiagramm Testbed, Quelle: Eigene Darstellung.

EMSDIL und EMSRAW

Jede Analyatoreinheit kann bis zu fünf Linien besitzen (Klasse EMS_Line), die verschiedene Abgasanalytoren beinhalten können. Jeder Analytator (Klasse Analyser) kann wiederum maximal vier Messbereiche aufweisen (Klasse Analyser_Measrange). Die Klasse Analytator beinhaltet noch folgende Attribute:

Attribut	Beschreibung
Use	Definiert, ob der Analytator verwendet wird
AnalytatorUsage	Definiert, ob der Analytator für Beutel und/oder für eine kontinuierliche Analyse verwendet wird
SerialNo	Gibt die Seriennummer des Analytators wieder
Type	Gib den Analytortyp an (z. B. FID, CLD etc.)

Tabelle 19: Attribute der Klasse Analyser, Quelle: Eigene Darstellung.

Jeder Messbereich (Analyser_Measrange) besitzt folgende Attribute:

Attribut	Beschreibung
Use	Definiert ob dieser Messbereich verwendet wird
Range	Gibt den Endwert des Messbereichs an
Calibrationgas_Concentration	Entspricht der Konzentration der angeschlossenen Kalibriergasflasche

Tabelle 20: Attribute der Klasse Analyser_Mearange, Quelle: Eigene Darstellung.

CVS

Die Klasse CVS besteht aus einer Beutelkonfiguration (BagCfg), den Hauptdurchsatzstufen (MainFlowStages) und den Probenahmedurchsatzstufen (SampleFlowStages). Die Attribute der Beutelkonfiguration sind in folgender Tabelle erläutert:

Attribut	Beschreibung
Bags	Anzahl der verbauten Beutel (z. B. 4)
BagsPerBagpair	Anzahl der gepaarten Beutel (z. B. 2, der Abgasbeutel und Luftbeutel)
VolBagMax	Maximales Füllvolumen eines Beutels
VolBagMin	Minimales Füllvolumen eines Beutels

Tabelle 21: Attribute der Klasse CVS_BagCfg, Quelle: Eigene Darstellung.

Ein CVS-Gerätetyp kann mittels einer Stufenumschaltung mehrere Hauptdurchsätze einstellen. Der Hauptdurchsatz beschreibt den konstanten Volumenstrom des verdünnten Abgases und ist ein Parameter der Emissionsermittlung. Jede Hauptdurchsatzstufe besitzt somit folgende Attribute:

Attribut	Beschreibung
Use	Gibt an, ob diese Durchsatzstufe verwendet wird
Flowrate	Gibt die hinter der Stufe liegende Durchflussrate an

Tabelle 22: Attribute der Klasse CVS_MainFlowStage, Quelle: Eigene Darstellung.

Der Probenahmedurchsatz gibt an, mit welcher Flussrate die Beutelbefüllung durchgeführt wird. Nach der Recherche der Messgeräte konnte festgestellt werden, dass sowohl eine stufenbehaftete als auch eine stufenlose Umschaltung des Probenahmedurchsatzes erfolgen kann. Die Beschreibung der stufenbehafteten Umschaltung ist in der Klasse CVS_SampleFlowStage zu finden. Der Aufbau dieser Klasse ist gleich der Klasse CVS_MainFlowStage (siehe Tabelle 22). In Tabelle 23 werde die restlichen Attribute der Klasse SampleFlowStages definiert.

Attribut	Beschreibung
VenturiShift	Gibt an ob eine stufenlose oder stufenbehaftete Umschaltung der Probenahmeflussrate vorliegt
MaxFlowrate	Gibt den maximalen Durchfluss bei stufenloser Umschaltung vor
MinFlowrate	Gibt den minimalen Durchfluss bei stufenloser Umschaltung vor

Tabelle 23: weitere Attribute der Klasse CVS_SampleFlowStages, Quelle: Eigene Darstellung.

PFS

In der nachfolgenden Tabelle ist das Attribut der Klasse PFS definiert.

Attribut	Beschreibung
FilterHolder	Gibt die Anzahl der Filterhalter an

Tabelle 24: Attribute der Klasse PFS, Quelle: Eigene Darstellung.

PCS

Die Tabelle 25 enthält das Attribut der Klasse PCS.

Attribut	Beschreibung
DilutionCfg	Gibt die verschiedenen Verdünnungsraten an

Tabelle 25: Attribute der Klasse PCS, Quelle: Eigene Darstellung.

EPA

In der Klasse der elektrischen Leistungsanalytoren wird das folgende Attribut definiert.

Attribut	Beschreibung
Channels	Gibt die Anzahl der verschiedenen Kanäle zur Messung von Strom und Spannung an

Tabelle 26: Attribute der Klasse EPA, Quelle Eigene Darstellung.

FTIR

Die Struktur des FTIR kann ähnlich zur Struktur des EMS aufgebaut werden. Es wird nur eine Linie benötigt, die die verschiedenen Abgaskomponenten beinhaltet. Es sind auch keine Messbereiche oder Funktionszeiten erforderlich, da diese Messgeräte diese Eigenschaften nicht besitzen. Lediglich folgendes Attribut wird benötigt, um einen Analysator der Linie zu definieren:

Attribut	Beschreibung
Use	Definiert, ob der Analysator verwendet wird

Tabelle 27: Attribute der Klasse FTIR_Analyser, Quelle: Eigene Darstellung.

Geräteeinstellung

Die Geräteeinstellung definiert, welche Messgerätetypen bei den verschiedenen Prüfzyklen verwendet werden. Des Weiteren wird in dieser Struktur festgelegt, welche Einstellungen diese Messgeräte für den Ablauf des Prüfzyklus benötigen. Die Abbildung 27 zeigt das Klassendiagramm der Geräteeinstellung. Mit der Basisklasse DeviceSettings werden alle Geräteeinstellungen strukturiert.

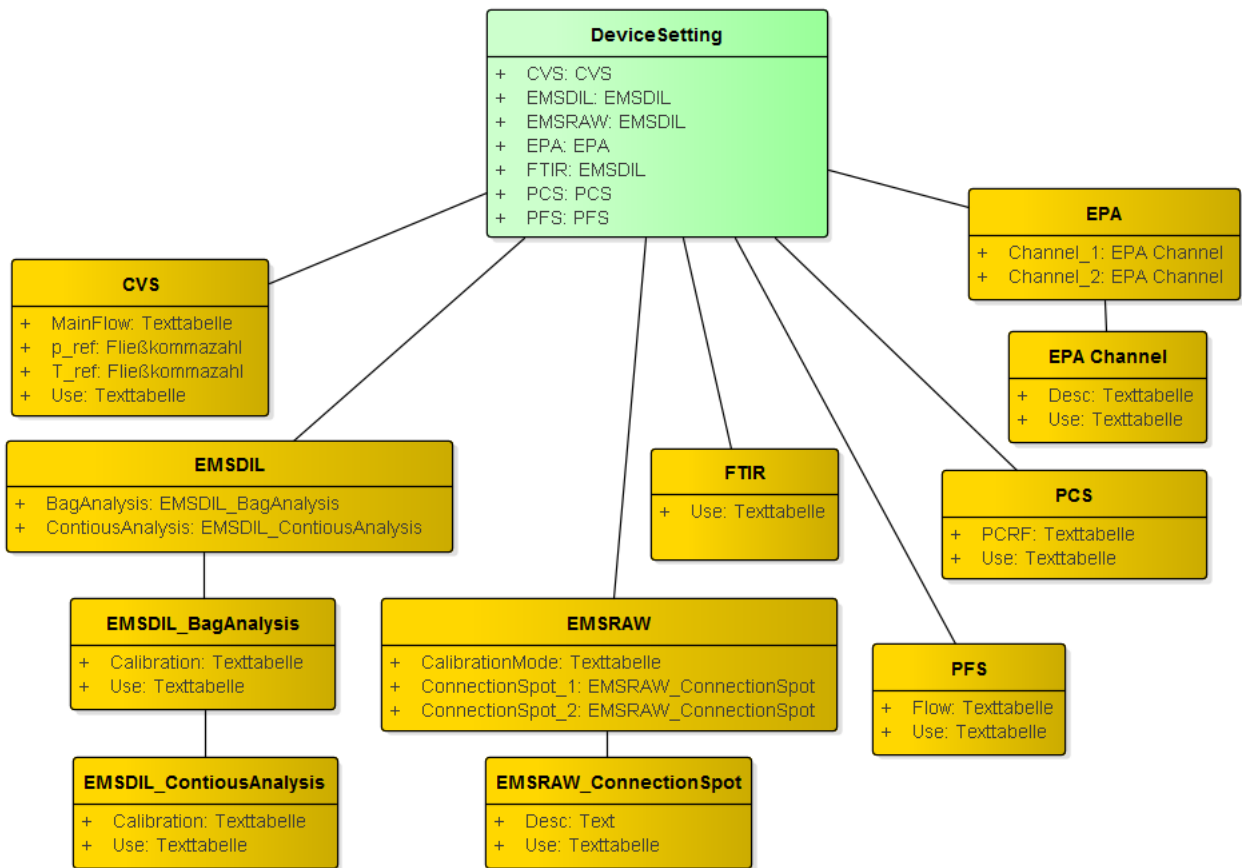


Abbildung 27: Klassendiagramm DeviceSetting, Quelle: Eigene Darstellung.

Mit dem Attribut Use kann entschieden werden, welche Messgerätetypen für den Ablauf verwendet werden sollen. Zusätzlich besitzt jeder Messgerätetyp spezifische Einstellparameter wie z. B. den Solldurchfluss (z. B. Attribut: Flow und MainFlow). Beim Messgerätetyp CVS müssen darüber hinaus noch Standardbedingungen, Temperatur (T_ref) und Druck (p_ref) angegeben werden. Dadurch wird das ermittelte Volumen immer auf diese Bedingungen umgerechnet. Bei der Analysatoreinheit zur Analyse der

verdünnten Abgase (EMSDIL) kann entschieden werden, ob eine Beutel- und/oder kontinuierliche Analyse durchgeführt werden soll. Bei der Analysatoreinheit zur Analyse der unverdünnten Abgase (EMSRW) wird definiert, ob und wo eine Entnahmestelle an das Fahrzeug angeschlossen wird (z. B. vor oder nach dem Katalysator). Zusätzlich kann ausgewählt werden, ob eine Kalibrierung der Messgeräte erfolgen soll (Attribut: Calibration). Bei der elektrischen Leistungsmessung können mehrere Messstellen am Fahrzeug relevant sein, somit wird in der Klasse EPA definiert, welche Kanäle verwendet werden und wo diese angeschlossen werden. Für das Partikelanzahlssystem kann der einzustellende Reduktionsfaktor ausgewählt werden.

Fahrzyklus

Mit Analyse der Fahrzyklen in Kapitel 2.1.5 und Kapitel 4.2.1 kann festgelegt werden, dass der grundsätzliche Aufbau eines Fahrzyklus aus Phasen und sogenannten Motorabstellphase (Soakphasen) besteht. Des Weiteren muss zwischen auszuwertenden und nicht auszuwertenden Phasen im Fahrzyklus unterschieden werden. Aus den Untersuchungen kann definiert werden, dass ein Fahrzyklus aus maximal vier auszuwertenden Phasen besteht. Unter Berücksichtigung eines Fahrzyklus mit nicht auszuwertenden Phasen (z. B. ERMES V8), kann jedoch ein Fahrzyklus bis sieben Phasen besitzen. Für das Konzept der Struktur des Fahrzyklus werden acht Phasen und sieben Motorabstellphasen definiert. Somit können alle bestehenden Fahrzyklen abgebildet werden. Falls in Zukunft neue Fahrzyklen mit mehr Phasen bestimmt werden, kann die Struktur problemlos erweitert werden.

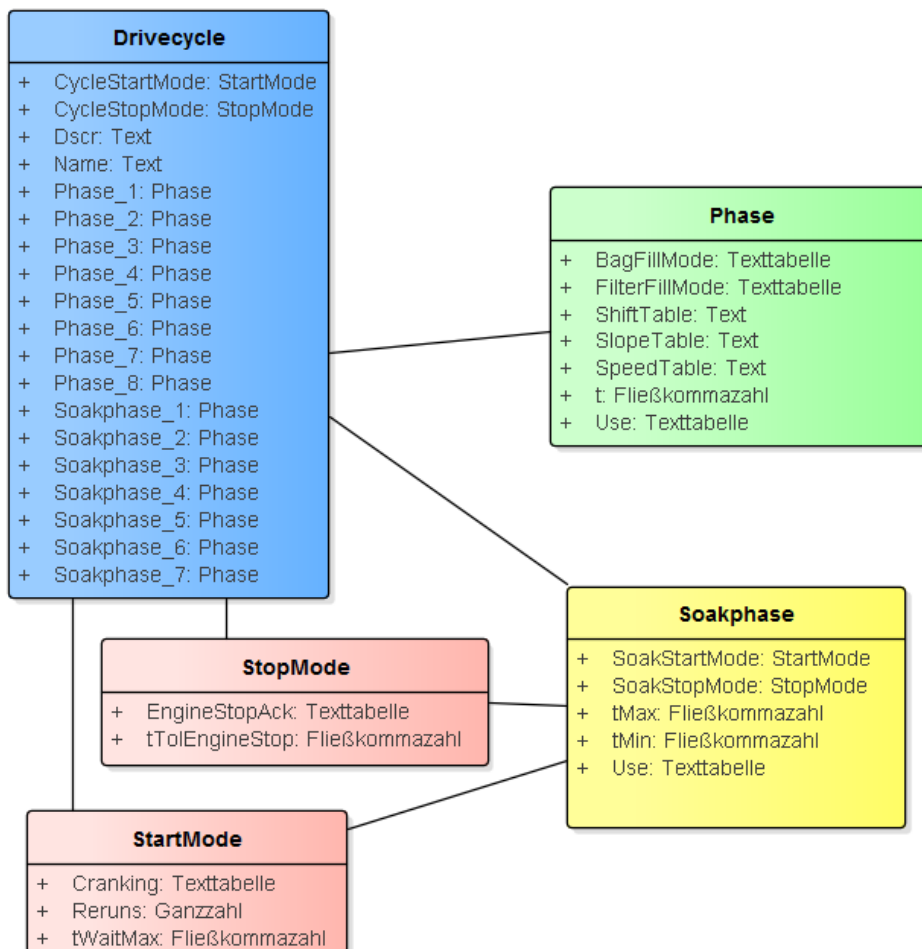


Abbildung 28: Klassendiagramm Drivecycle, Quelle: Eigene Darstellung.

Die Klassen StartMode und StopMode beschreiben den Ablauf des Motorstarts und des Motorstopps. Im Ablauf des Fahrzyklus muss der Fahrer / die Fahrerin mit einer Bedieneinheit bestätigen, dass der Motor gestartet bzw. abgestellt wurde. Bei älteren Fahrzeugen kann es vorkommen, dass der Motor bei einem Kaltstart nicht sofort anspringt und somit mehrere Versuche notwendig sind, um den Motor zu starten. Bei Motorstopp soll der Fahrer / die Fahrerin bestätigen, dass der Motor wirklich gestoppt wurde. Diese erweiterten Bedingungen des Motorstarts und Motorstopps sind in den Verordnungen der 40 CFR Part 86 vermerkt.

Die Attribute der Klasse StartMode sind in folgender Tabelle aufgeführt:

Attribut	Beschreibung
Cranking	Definiert, ob eine mehrmalige Motorstartüberprüfung stattfinden soll
Reruns	Definiert die maximale Anzahl an Versuchen des Motostarts
tWaitMax	Definiert die maximale Dauer zwischen Motorstart und Motorstartbestätigung

Tabelle 28: Attribute der Klasse StartMode, Quelle: Eigene Darstellung

Die Attribute der Klasse StopMode sind in der Tabelle unten aufgelistet:

Attribut	Beschreibung
EngineStopAck	Definiert, ob der Motorstopp bestätigt werden soll
tTolEngineStop	Definiert die maximale Dauer zwischen Fahrkurvenende und Motorstoppbestätigung

Tabelle 29: Attribute der Klasse StopMode, Quelle: Eigene Darstellung

Um so flexibel wie möglich zu bleiben, werden die Geschwindigkeits-, Steigungs- und Schaltpunktparametertabelle separat zu jeder Phase definiert. In Tabelle 30 werden die Attribute der Klasse Phase dargestellt.

Attribut	Beschreibung
Use	Definiert, ob eine Phase benutzt werden soll
SpeedTable	Gibt den Namen der Parametertabelle des Geschwindigkeitsverlaufs an
ShiftTable	Gibt den Namen der Parametertabelle des Schaltpunktverlaufs an
SlopeTable	Gibt den Namen der Parametertabelle des Steigungsverlaufs an
t	Gibt die Dauer der Phase an (optional, um den Verlauf zu verkürzen)
BagFillMode	Gibt das zu befüllende Beutelpaar in der Phase an
FilterFillMode	Gibt den zu beladenen Filter in der Phase an

Tabelle 30: Attribute der Klasse Phase, Quelle: Eigene Darstellung.

Die Attribute BagFillMode und FilterFillMode werden definiert, um die Ansteuerung und Auswertung der Messgeräte so flexibel wie möglich zu gestalten. Diese Attribute können folgende Definitionen aufweisen (hier am Beispiel des BagFillMode):

- Kein Beutelpaar
- Nächstes Beutelpaar
- Beutelpaar 1
- ...
- Beutelpaar 4
- Beutelpaar der Phase 1
- ...
- Beutelpaar der Phase 7

Mit ‚kein Beutelpaar‘ wird definiert, dass in dieser Phase keine Ansteuerung und Auswertung passieren. Mit der Definition ‚Nächstes Beutelpaar‘ erfolgt die Ansteuerung der Beutelpaare der Reihe nach. Mit der Definition ‚Beutelpaar x‘ (x ist 1 bis 4) kann der Phase ein bestimmtes Beutelpaar zugewiesen werden und mit der Definition ‚Beutelpaar der Phase y‘ (y ist 1 bis 7) kann ein bestimmtes Beutelpaar in einer weiteren Phase weiter befüllt werden. Der gleiche Aufbau findet sich im Attribut FilterFillMode, bezogen auf Filter, wieder. Somit ist jede Möglichkeit der Ansteuerung und Auswertung der Phasen abgedeckt.

Eine Soakphase wird mit folgenden Attribute beschrieben:

Attribut	Beschreibung
Use	Definiert, ob eine Soakphase benutzt werden soll
tMax	Gibt die maximale Dauer der Soakphase an
tMin	Gibt die minimale Dauer der Soakphase an

Tabelle 31: Attribute der Klasse Soakphase, Quelle: Eigene Darstellung.

Da die Soakphase per Definition einen Motorstopp und einen Motorstart verlangt, sind auch in dieser Klasse die Unterklassen StartMode und StopMode definiert.

Testaufgabe

In der Testaufgabe werden alle weiteren relevanten Einstellungen gesammelt, die die Ansteuerung und Auswertung des Prüfzyklus betreffen. Das Klassendiagramm zur Testaufgabe wird in Abbildung 29 dargestellt. Die Klasse Testtask umfasst folgende Elemente:

- EmissionLimits
- Law
- Fuel
- Testcondition
- Multicycle

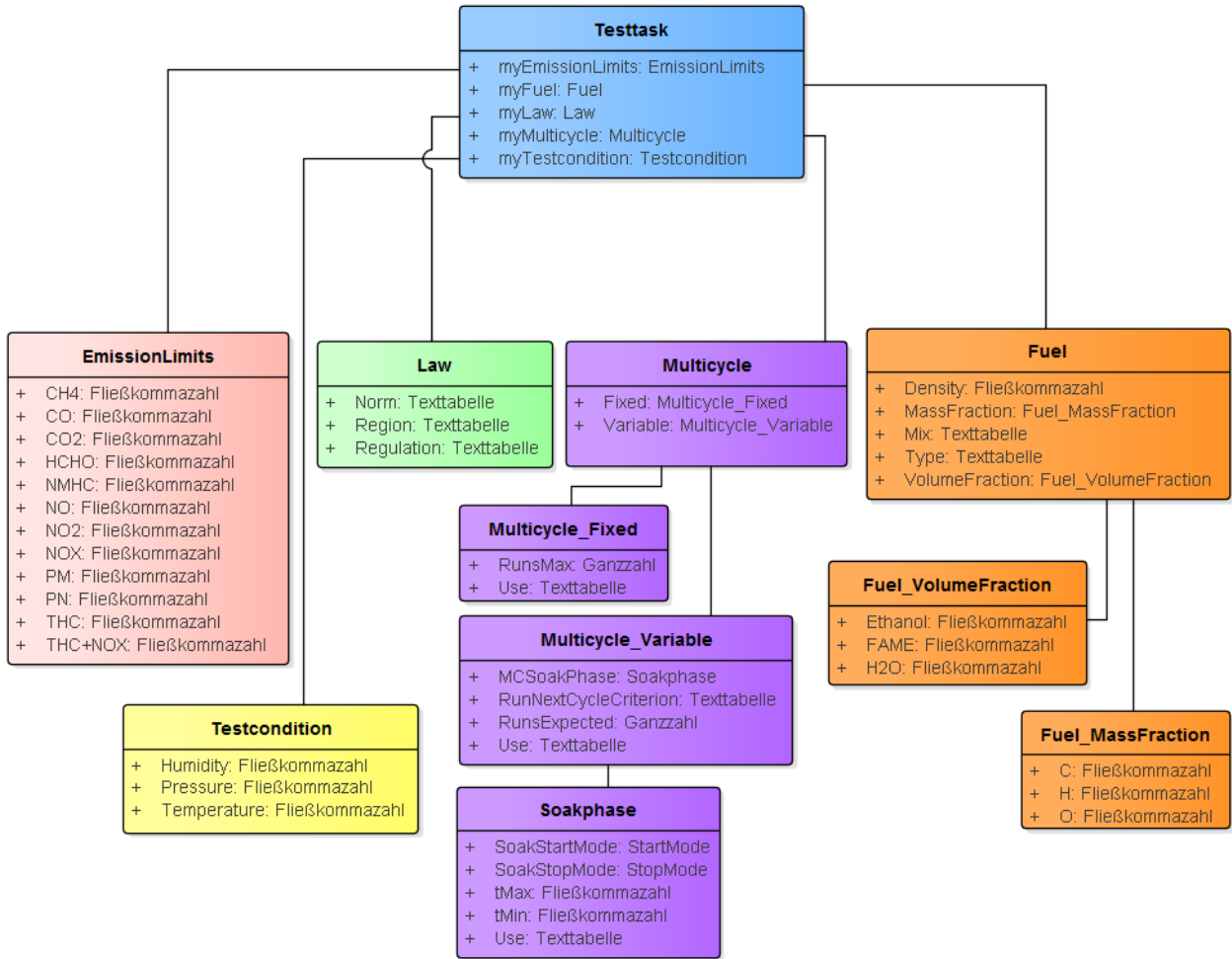


Abbildung 29: Klassendiagramm Testtask, Quelle: Eigene Darstellung.

Die Klasse EmissionLimits beinhaltet alle Emissionsgrenzwerte, die in Kapitel 2.1.2 Grenzwerte ermittelt wurden. Die Klasse Law umfasst Informationen über die Abgasnorm und die damit verbundene Verordnung (abgeleitet aus Kapitel 2.1.1). Die Attribute der Klasse Law sind in der nachfolgenden Tabelle beschrieben.

Attribut	Beschreibung
Region	Definiert, welcher Region die Gesetzgebung angehört (z. B. Europa)
Norm	Gibt an, welche Abgasnorm angewendet werden soll (z. B. Euro 6)
Regulation	Gibt die technische Richtlinie bzw. Verordnung an (z. B. EU 2017/1151)

Tabelle 32: Attribute der Klasse Law, Quelle: Eigene Darstellung.

Die Klasse Testcondition definiert die Umgebungsbedingungen des Prüfraums, die während des Prüfzyklus eingestellt werden. In Tabelle 33 sind die Attribute der Klasse Testconditions aufgeführt.

Attribut	Beschreibung
Temperature	Definiert, welche Umgebungstemperatur im Prüfstand eingestellt wird
Pressure	Definiert, welcher Umgebungsdruck im Prüfstand eingestellt wird
Humidity	Definiert, welche Umgebungsfeuchtigkeit im Prüfstand eingestellt wird

Tabelle 33: Attribute der Klasse Testcondition, Quelle: Eigene Darstellung.

Mit der Klasse Fuel werden alle Informationen zum Kraftstoff dargestellt. Diese Informationen sind notwendig, um den Kraftstoffverbrauch bzw. die Berechnung des Kraftstoffverbrauchs zu definieren. In der nachfolgenden Tabelle werden die Attribute der Klasse Fuel beschrieben.

Attribut	Beschreibung
Density	Gibt die Dichte des Kraftstoffes an
Type	Definiert den Typ des Kraftstoffes (z. B. Benzin)
Mix	Definiert das Kraftstoffgemisch (z. B. Benzin E5)
MassFraction	Beinhaltet die Attribute zur Definition der Massenanteile des Kraftstoffs
VolumeFraction	Beinhaltet die Attribute zur Definition der Volumenanteile des Kraftstoffs

Tabelle 34: Attribute der Klasse Fuel, Quelle: Eigene Darstellung.

Die Einstellungen des Mehrfachzyklus werden auch in die Testaufgabe implementiert. Es werden zwei Varianten bereitgestellt. In der ersten Variante kann eine fix definierte Anzahl an sich wiederholenden Fahrzyklen angegeben werden. Die zweite Variante ermöglicht es, dass der Fahrer / die Fahrerin nach jedem Fahrzyklus entscheiden kann, ob ein weiterer Fahrzyklus gestartet wird. Diese Variante stellt somit eine Variable Anzahl der zu wiederholenden Fahrzyklen dar.

Der fix definierte Mehrfachzyklus wird mit folgenden Attributen der Klasse Multicycle_Fixed bestimmt:

Attribut	Beschreibung
Use	Gibt an, ob ein fix definierter Mehrfachzyklus abgefahren wird
RunsMax	Definiert die Anzahl der zu wiederholenden Fahrzyklen

Tabelle 35: Attribute der Klasse Multicycle_Fixed, Quelle: Eigene Darstellung.

Die Klasse Multicycle_Variable enthält hingegen folgende Attribute:

Attribut	Beschreibung
Use	Gibt an, ob ein variabler Mehrfachzyklus abgefahren wird
RunsExpected	Definiert die erwartete Anzahl der zu wiederholenden Fahrzyklen
RunNextCycleCriterion	Definiert, welches Kriterium herangezogen werden soll, dass der nächste Fahrzyklus gefahren wird
MCSoakPhase	Mit dieser Struktur kann entschieden werden, ob vor dem Start des nächsten Fahrzyklus eine Soakphase eingeleitet wird

Tabelle 36: Attribute der Klasse Multicycle_Variable, Quelle: Eigene Darstellung.

Ergebnisstruktur

In allen drei bestehenden Projekten wurde kaum Wert auf die Strukturierung der Ergebnisvariablen gelegt. Daher wird in dieser Entwurfsphase auch eine allgemeine Struktur für alle ermittelten Messwerte und berechneten Ergebnisse erstellt. Im Wesentlichen wird zwischen phasenbezogenen und fahrzyklenbezogenen Messdaten unterschieden. In Abbildung 30 ist der Erstentwurf für die phasenbezogenen Ergebnisstruktur zu sehen. In dieser Struktur wird zwischen den Ergebnissen der Messgeräte (Devices), des Prüflaufs (Testrun) und den Ergebnissen der anschließenden Emissionsberechnung (Calculation) unterschieden.

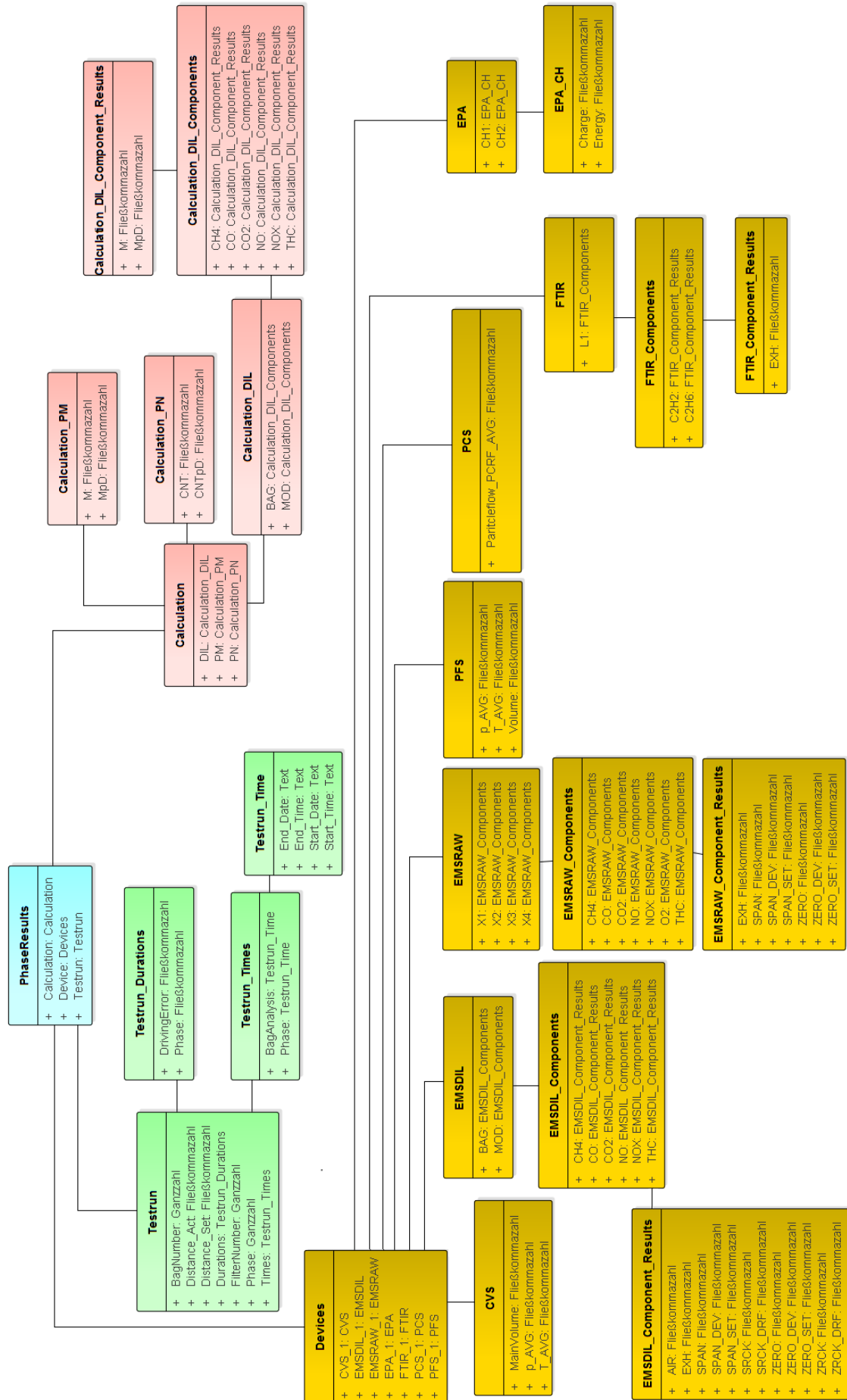


Abbildung 30: Klassendiagramm PhaseResults, Quelle: Eigene Darstellung.

Abbildung 31 zeigt die Struktur der Fahrzyklenergebnisse. Im Unterschied zu den Phasenergebnissen wird hier keine Struktur der Ergebnisse der Messgeräte benötigt, da das Auslesen der relevanten Messdaten aus den Messgeräte immer phasenbezogen ist.

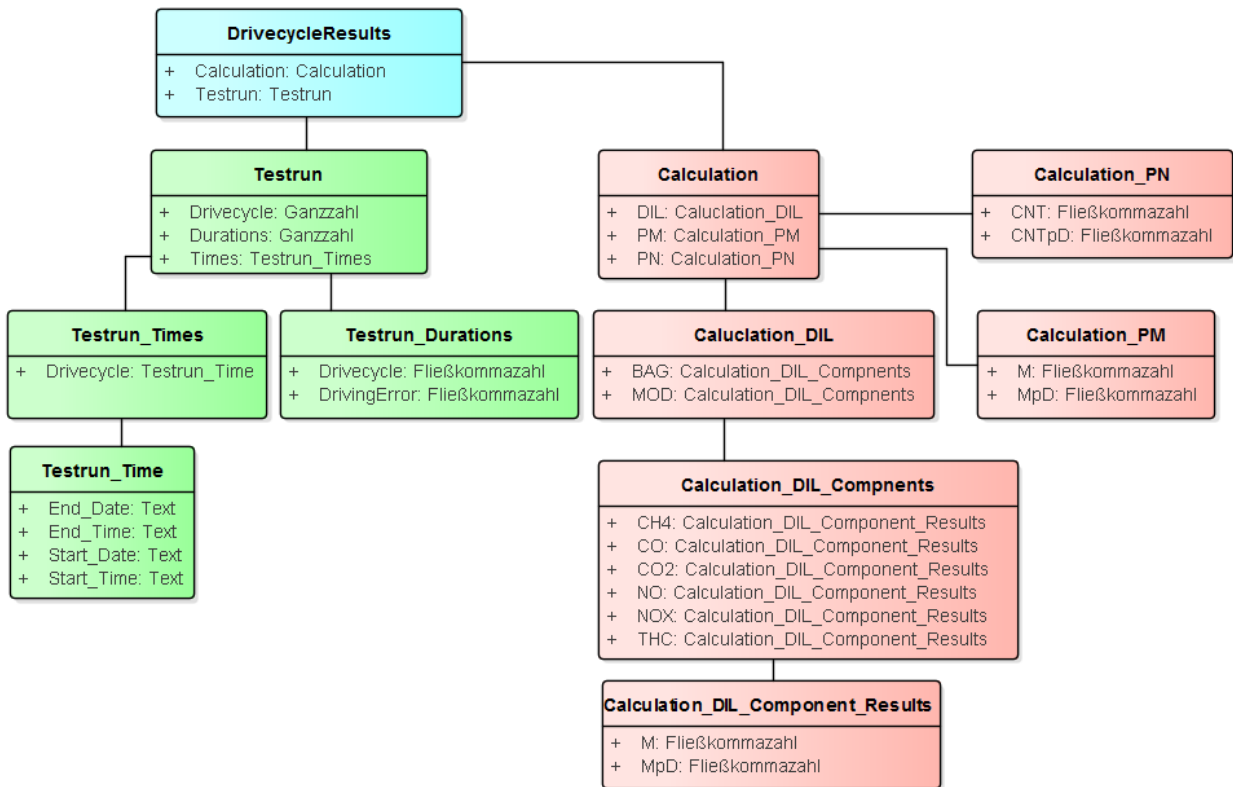


Abbildung 31: Klassendiagramm DrivecycleResults, Quelle: Eigene Darstellung.

Auf eine detaillierte Beschreibung der Klassendiagramme aus Abbildung 30 und Abbildung 31 wird an dieser Stelle verzichtet. Im Wesentlichen werden in dieser Struktur Mittelwert- und Integrationsergebnisse sowie allgemeine Informationen (Zeitstempel, Dauer etc.) des Prüfzyklus gesammelt.

5.2.2 Steuerungsentwurf

Im Kapitel des Steuerungsentwurfs wird das zu entwickelnde Softwaremodul, aus Sicht des Prozesses nach dem 4+1-Sichtenmodell betrachtet. Dabei werden Konzepte für den Ablauf und das Verhalten des Softwaremoduls entwickelt. Aus der Analyse der bestehenden Projekte (siehe Kapitel 4.2) ging hervor, dass die Steuerungen der Prüfzyklen hinsichtlich der Testbarkeit und Neuimplementierung schwer zu handhaben sind. In dieser Entwurfsphase sollen die erwähnten Probleme durch ein neues Steuerungskonzept beseitigt werden. Die Verhaltensdiagramme der UML-Diagrammfamilie bieten mehrere Diagrammtypen an, um einen Programmablauf strukturiert zu modellieren. In den folgenden Kapiteln werden mittels dieser Verhaltensdiagramme Lösungen für einen allgemeinen Ablauf eines Prüfzyklus modelliert.

5.2.2.1 Fahrzyklensteuerung

Wesentliche Elemente des gesamten Prüfzyklus der Typ-1-Prüfung ist die Steuerung des Fahrzyklus. Aus den vorangehenden Untersuchungen kann definiert werden, dass ein Fahrzyklus mit einem Motorstart beginnt, danach zwischen verschiedenen Phasen wechselt und mit einem Motorstopp in eine Motorabstellphase übergeht oder das Ende des Fahrzyklus erreicht ist. Zusätzlich muss das Verhalten des Fahrzyklus in einem variablen und fix definierten Mehrfachzyklus berücksichtigt werden. Der gesamte Ablauf eines Fahrzyklus kann somit mittels eines Zustandsautomaten beschrieben werden. In Abbildung 32 ist der Zusammenhang der Zustände grafisch dargestellt. Die folgende Auflistung, beschreibt die einzelnen Zustände des Zustandsautomaten:

- StartEngine

In diesem Zustand leitet der Fahrer / die Fahrerin den Motorstart und somit den Fahrzyklusstart ein. Über eine Bedieneinheit wird der Motorstart bestätigt, sodass die Zustandsmaschine den Fahrzyklus startet, bzw. in den nächsten Zustand, der Phase, wechselt. Für den Motorstart wird im Zustand ‚StartEngine‘ auf die Variablenstruktur der verschiedenen StartMode-Klassen zugegriffen (CycleStartMode, SoakStartMode) und somit entschieden wie der Motorstartablauf stattfinden soll. Zusätzlich wird in diesem Zustand die Fahrkurve für das Fahrerleitsystem vorbereitet.

- Phase

In diesem Zustand wird die Dauer der Phase abgewartet. Zu Beginn dieses Zustands werden Steuerungsbefehle an die Messgeräte gesendet (z. B. Starten der Filterbeladung des PFS). Am Ende des Zustands wird erkannt, ob ein neue Phase beginnt oder ein Motorstopp eingeleitet werden muss. Der Phasenwechsel in einem Fahrzyklus und der Wechsel eines Fahrzyklus in einem fix definierten Mehrfachzyklus sind im Zustandsübergang gleich. Es wird lediglich neu in den Zustand der Phase gewechselt.

- StopEngine

In diesem Zustand stellt der Fahrer / die Fahrerin den Motor ab und bestätigt dies mit einer Bedieneinheit. Danach wird entschieden, ob der nächste Zustand eine Motorabstellphase, eine Motorabstellphase eines variablen Mehrfachzyklus oder das Ende des Fahrzyklus ist. In diesem

Zustand wird auf die Struktur der verschiedenen StopMode-Klassen zugegriffen (CycleStopMode, SoakStopMode) und somit entschieden wie der Ablauf des Motorstopps stattfinden soll.

- Soakphase

In diesem Zustand wird eine Minimaldauer abgewartet. Ist diese erreicht, kann der Wechsel zum Zustand ‚StartEngine‘ und somit der Motorstart für die nächste Phase eingeleitet werden. Während der Soakphase ist keine Parametertabelle aktiv bzw. wird dem Fahrer / der Fahrerin keine Fahrkurve am Fahrerleitbildschirm angezeigt. Lediglich die verstrichene Zeit der Soakphase wird dem Fahrer / der Fahrerin angezeigt.

- MCvarSoakphase

Wird ein variabler Mehrfachzyklus definiert, kann ähnlich zum Zustand ‚Soakphase‘ eine minimale Dauer abgewartet werden (Abhängig von der Parametrierung der Testaufgabe). Ist die minimale Dauer erreicht, entscheidet der Fahrer / die Fahrerin, ob ein neuer Fahrzyklus gestartet wird, und somit ein Motorstart eingeleitet wird, oder das Ende des Tests eingeleitet werden soll.

- EndOfTest

Dieser Zustand beschreibt den Abschluss des Fahrzyklus. Hier werden abschließende Steuerungen an die Messgeräte gesendet und auf die restliche Auswertung der Messdaten abgewartet (z. B. abwarten bis alle Beutelpaare analysiert wurden).

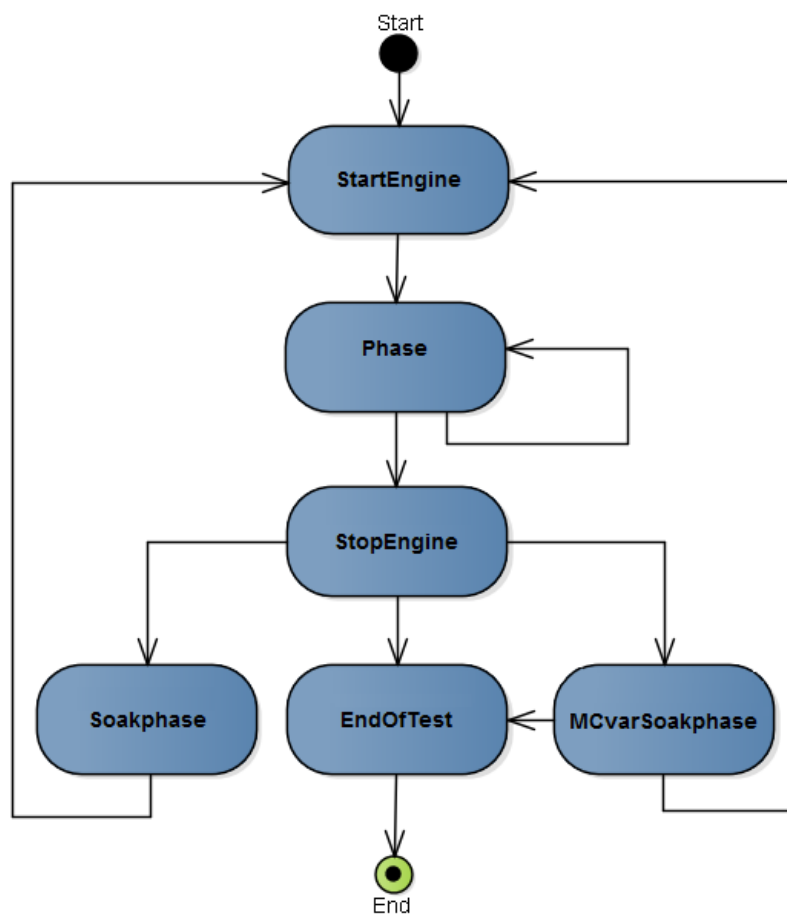


Abbildung 32: Zustandsautomat der Fahrzyklensteuerung, Quelle: Eigene Darstellung.

5.2.2.2 Beutelsteuerung

Die Ansteuerung der Beutelpaare ist ein wesentlicher Bestandteil des Prüfzyklus. Jedoch wird die Steuerung der Messgerätetypen (EMSDIL, CVS) nicht im Zustandsautomaten der Fahrzyklensteuerung abgearbeitet, sondern ein separater Zustandsautomat definiert. Dadurch ist der Ablauf der Beutelansteuerung übersichtlicher. Abbildung 33 stellt den Zustandsautomat der Beutelsteuerung mit den verschiedenen Übergängen dar. In der nachfolgenden Auflistung werden alle Zustände näher erläutert:

- **Empty**

Dieser Zustand beschreibt ein leeres, bereits evakuiertes Beutelpaar. In diesem Zustand wird gewartet, bis aus dem Zustandsautomaten der Fahrzyklensteuerung der Trigger gesetzt wird (Einstieg in Zustand ‚Phase‘ der Fahrzyklensteuerung). Im Anschluss kann in den Zustand ‚Filling‘ gesprungen werden. Wird erkannt, dass das Testende, bzw. Fahrzyklusende erreicht wurde (Zustand ‚EndOfTest‘ der Fahrzyklensteuerung), kann das Ende der Zustandsmaschine für dieses Beutelpaar eingeleitet werden.

- **Filling**

Mit diesem Zustand wird die Befüllung des Beutelpaares beschrieben. Die Dauer der Befüllung ist gleich der Phasendauer. Ist die Befüllung abgeschlossen, muss entschieden werden, ob das Beutelpaar fertig befüllt ist oder eine weitere Befüllung des Beutepaares in einer anschließenden Phase vorgesehen ist und somit das Beutelpaar nur teilweise befüllt wurde.

- **Filled**

Dieser Zustand beschreibt ein fertig befülltes Beutelpaar. In diesem Zustand muss entschieden werden, ob eine Analyse der Beutelpaare bereits möglich ist. Werden beispielsweise die Analysatoren für die modale Analyse benötigt, so kann die Beutelanalyse erst gestartet werden, wenn die Analysatoren für die modale Analyse nicht mehr gebraucht werden (z. B. in einer Soakphase oder am Ende des Tests). Diese Entscheidung ist abhängig vom Aufbau der Analysatoreinheit. Besitzt eine Analysatoreinheit zwei Linien mit den gleichen Analysatoren, ist auch eine Analyse während der Phase möglich, da eine Linie die modale Analyse abarbeitet und die zweite Linie die Beutelanalyse durchführen kann.

- **Analysing**

In diesem Zustand wird ein Beutelpaar analysiert. Dieser Ablauf beinhaltet das Analysieren des Luftbeutels und des Abgasbeutels. Sobald das Analysieren des Beutelpaares beendet ist, wechselt der Zustandsautomat in den Zustand ‚Analysed‘.

- **Analysed**

Dieser Zustand beschreibt ein fertig analysiertes Beutelpaar. Die Entscheidung, wann ein analysiertes Beutelpaar in die Evakuierung wechseln kann, ist wiederum vom Aufbau der Analysatoreinheit abhängig. Dieser Sachverhalt wird im weiteren Verlauf dieses Kapitels erklärt (siehe Darstellung und Erläuterung der Tabelle 37).

- **Evacuate**

In diesem Zustand wird das Evakuieren der Beutelpaare eingeleitet. Ist das Beutelpaar entleert so kann in den Zustand ‚Empty‘ gewechselt werden.

- **Partial Filled**

Dieser Zustand beschreibt ein teilweise befülltes Beutelpaar. Wird die Phase der Weiterbefüllung im Zustandsautomaten der Fahrzyklensteuerung eingeleitet, so kann dieses Beutelpaar weiter befüllt werden und somit in den Zustand ‚Filling‘ gewechselt werden.

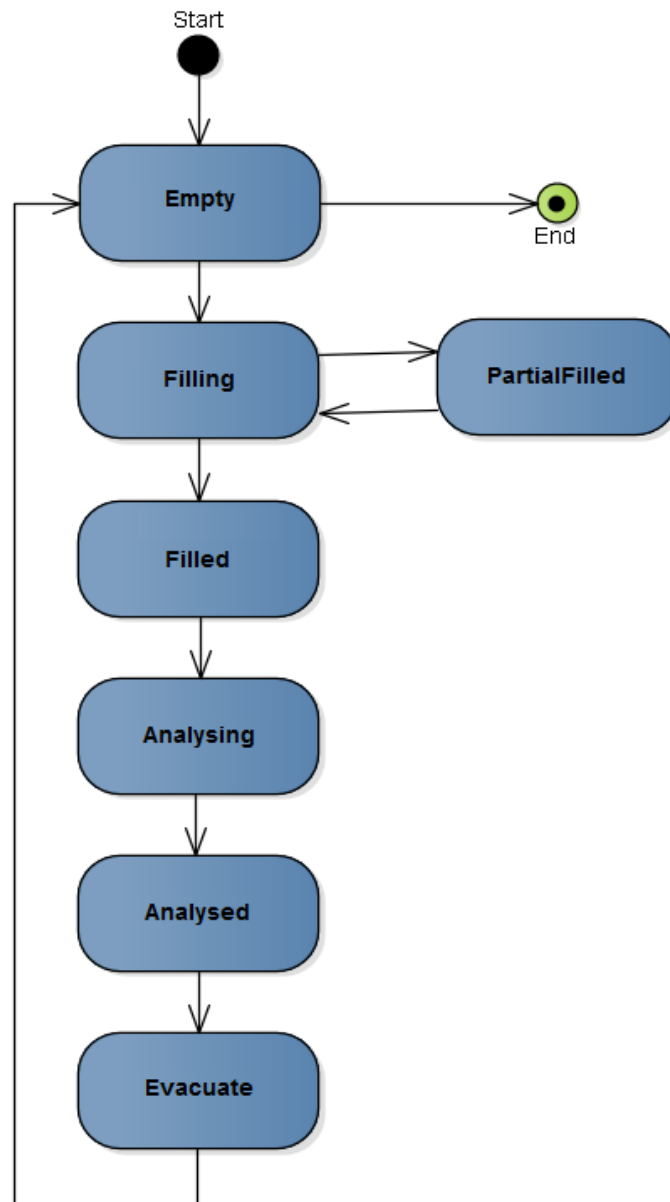


Abbildung 33: Zustandsmaschine der Beutelsteuerung, Quelle: Eigene Darstellung.

Der in Abbildung 33 dargestellte Zustandsautomat ist lediglich für ein Beutelpaar konzipiert. Da es unter Umständen Anlagen mit bis zu vier Beutelpaaren geben kann, wird um diesen Zustandsautomaten noch eine Schleife gebildet, mit dem alle benötigten Beutelpaare abgedeckt werden. So können zyklisch nacheinander die Zustände der einzelnen Beutelpaare ermittelt werden.

Eine Analysatoreinheit zur Analyse der verdünnten Abgase kann mehrere Linien besitzen, die gleichzeitig oder nacheinander während einer Beutelanalyse verwendet werden können. Dadurch müssen in der Steuerung der Beutelpaare die Linien der Analysatoreinheit berücksichtigt werden. Folgendes Beispiel soll das Verhalten aufzeigen:

Die Analysatoreinheit besteht aus zwei Linien (in diesem Beispiel L2 und L5). Die L2 besitzt die Analysatoren CO₂, CO und NO_X, die L5 dagegen den Analysator für THC. Wird beispielsweise ein vierphasiger Fahrzyklus gefahren, bei dem alle Phasen mit einer Beutelbefüllung und anschließender Analyse beauftragt sind, so müssen alle vier Beutelpaare in der Steuerung berücksichtigt werden. Die Zustände Empty, Filling, Filled, PartialFilled und Evacuate können immer auf alle Linien bezogen werden. Die Analyse kann aber unter Umständen linienweise ausgeführt werden, weshalb die Darstellung der Zustandsmaschine um die Linien erweitert werden muss. Tabelle 37 stellt diese Erweiterung dar. Beutelpaar 1 (BP1) wurde bereits befüllt und mit L2 analysiert. Beutelpaar 2 (BP 2) wurde auch fertig befüllt und wird gerade von L2 analysiert. Beutelpaar (BP3) wird währenddessen gerade befüllt, was bedeutet, dass aktuell die dritte Phase des vierphasigen Fahrzyklus abgefahren wird. Beutelpaar 4 (BP4) ist zu diesem Zeitpunkt noch leer, da die vierte Phase noch nicht begonnen hat. Da unter Umständen L5 dazu benötigt wird, um kontinuierlich während des Fahrzyklus die Abgaskonzentration des THC zu messen, muss diese Linie am Ende des Fahrzyklus noch einmal verwendet werden, um die Beutelpaare vollständig auszuwerten.

	BP1	BP2	BP3	BP4
L1	-	-	-	-
L2	Analysed	Analysing	Filling	Empty
L3	-	-	-	-
L4	-	-	-	-
L5	Filled	Filled	Filling	Empty

Tabelle 37: Erweiterte Darstellung der Beutelsteuerung um Linien, Quelle: Eigene Darstellung.

5.2.2.3 Ablauf der Prüfzyklen

Die beiden Zustandsmaschinen steuern das Verhalten während eines Fahrzyklus. Jedoch müssen vor und nach dem Fahrzyklus vorbereitende und abschließende Aktivitäten ausgeführt werden. Mittels eines Aktivitätsdiagramms wird der gesamte Ablauf eines Abgastests modelliert. Das in Abbildung 34 ersichtliche Aktivitätsdiagramm beschreibt die Aktionen, die in der späteren Implementierungsphase als Prüfprogramm in Tornado umgesetzt werden. Der Ablauf ist in drei Bereiche unterteilt (dargestellt als gestrichelte Umrandungen). Der erste Bereich umfasst alle Aktivitäten die zur Vorbereitung des Prüfzyklus bzw. der abzufahrenden Fahrkurve benötigt werden. Folgende Aktivitäten werden dazu definiert:

- **Initialisierung der Variablen**

In dieser Aktivität werden alle Variablen zur Steuerung des Prüfzyklus zurückgesetzt, um einen Initialzustand vor Fahrzyklusstart zu erlangen.

- **Vorbereitung der Prüflaufdaten**

Aus der Beauftragungsstruktur des Fahrzyklus und der Testaufgabe können vorab die Zuordnungen der Phasen und Beutelsteuerung ermittelt werden. Diese vorbereitenden Maßnahmen werden in dieser Aktivität umgesetzt.

- **Messdaten vorbereiten**

In dieser Aktivität werden die Messdaten der phasen- und fahrzyklenbezogenen Ergebnisse zurückgesetzt und somit für den anstehenden Prüfzyklus vorbereitet.

- **Initialisierung und Kalibrierung der Messgeräte**

In dieser Aktivität werden alle Messgeräte in einen initialen Zustand gebracht. Je nach Definition der Geräteeinstellung, wird zusätzlich eine Kalibrierung bestimmter Messgeräte ausgeführt.

- **Zustandsmaschine starten**

In dieser Aktivität werden die Zustandsmaschinen der Fahrzyklensteuerung und der Beutelsteuerung gestartet.

Wie schon bei der Vorstellung des Konfigurationselements ‚Prüfprogramme‘ erwähnt, ist das Fahrerleitsystem in Tornado zwingend mit einem Prüfprogramm anzusteuern. Dadurch müssen der Start und das Ende des Fahrerleitsystems sowie die dahinterliegenden Parametertabellen in diesem Ablauf implementiert werden. Die Zustandsmaschine der Fahrzyklensteuerung dient zur Orientierung in einem Fahrzyklus. Das Prüfprogramm soll lediglich die Anzeige der Fahrkurve steuern. Nach jedem Motorstart (getriggert aus der Zustandsmaschine der Fahrzyklensteuerung) wird somit der Fahrkurvenverlauf im Prüfprogramm neu aufgebaut bzw. das Fahrerleitsystem wird neu initialisiert und gestartet. Bei einem FTP-75, wird z. B. die Schleife im Ablauf zweimal durchlaufen, da dieser Zyklus zwei Motorstarts (zu Beginn des Zyklus und nach der Soakphase) beinhaltet. Im gestrichelten Bereich ‚Fahrkurve‘ werden somit folgende Aktivitäten ausgeführt:

- **Warten auf Motorstart**

In der Zustandsmaschine der Fahrzyklensteuerung wird im Zustand ‚StartEngine‘ auf den Motorstart bzw. der Bestätigung des Motorstarts durch den Fahrer / der Fahrerin gewartet. Ist der Motorstart ausgeführt, so wird ein Trigger aus der Zustandsmaschine der Fahrzyklensteuerung an den Ablauf übermittelt, um das Fahrerleitsystem bzw. die Anzeige der Fahrkurve parallel dazu zu starten.

- **Fahrerleitsystem und Parametertabelle starten**

In dieser Aktivität wird das Fahrerleitsystem sowie die Parametertabelle gestartet.

- **Warten auf Ende des Fahrkurventeils**

Grundsätzlich könnte hier auf einen Trigger der Zustandsmaschine der Fahrzyklensteuerung gewartet werden (z. B. Eintritt in den Zustand ‚EndOfTest‘ oder ‚Soakphase‘), jedoch benötigt das Fahrerleitsystem einen Bezug auf die Dauer der Fahrkurve im Prüfprogramm. Somit wird an dieser Stelle die Dauer der abzufahrenden Fahrkurve abgewartet.

- **Fahrerleitsystem und Parametertabelle stoppen**

In dieser Aktivität wird das Fahrerleitsystem sowie die Parametertabelle gestoppt.

Am Ende einer Fahrkurve wird aus der Zustandsmaschine der Fahrzyklensteuerung entschieden, ob ein weiterer Fahrkurventeil abzufahren ist oder das Ende des Prüfzyklus erreicht wurde. Ist das Ende erreicht oder wird der Ablauf des Prüfzyklus unterbrochen, so wird mit dem Ablauf des Bereichs ‚Abschluss‘ fortgefahren. Dieser Bereich beinhaltet folgende Aktivitäten:

- **Restanalyse abwarten**

Am Ende eines Prüfzyklus wird in dieser Aktivität abgewartet, bis alle befüllten Beutelpaare analysiert wurden. Erst wenn die Zustandsmaschine der Beutelsteuerung erkennt, dass alle Beutel analysiert und evakuiert sind, wird die nächste Aktivität im Ablauf eingeleitet.

- **Messgeräte finalisieren**

In dieser Aktivität werden alle verwendeten Messgeräte in einen finalen Zustand gebracht und die Kommunikation zu den Messgeräten beendet.

- **Messdaten auswerten**

In dieser Aktivität werden die ermittelten Messwerte ausgewertet und die Abgasemissionen und Partikelemissionen berechnet.

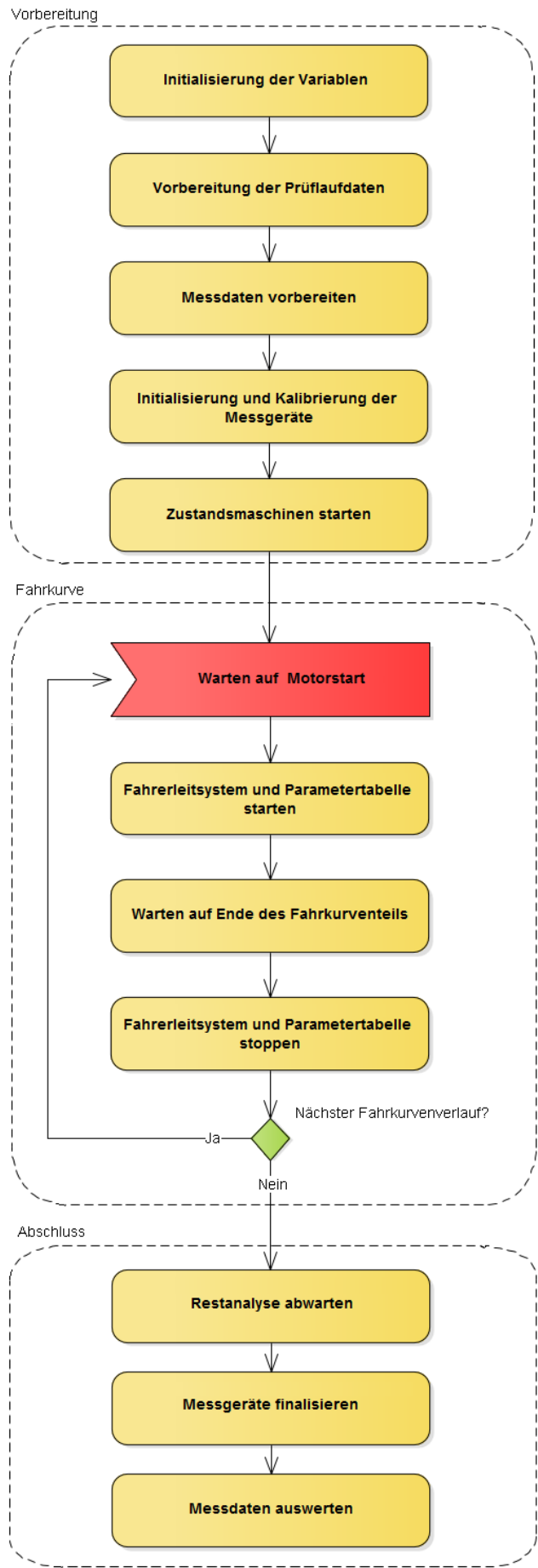


Abbildung 34: Aktivitätsdiagramm des Prüfprogramms, Quelle: Eigene Darstellung.

6 IMPLEMENTIERUNG IN TORNADO

In diesem Kapitel werden die erarbeiteten Informationen und entworfenen Modellen in das Automatisierungssystem Tornado implementiert. Es werden die ersten Ergebnisse der Umsetzung und Überlegungen der Implementierungsphase erläutert.

6.1 Implementierung der Messgeräte

In Kapitel 5.2.1.2 wurden der Entwurf der Gruppierung und die Nomenklatur der Messgerätetypen erstellt. Das Paketdiagramm (siehe Abbildung 19) dient als Vorlage, wie die benutzerdefinierten Gruppen der Geräte im Tornado Designer angelegt werden. Diese Klassendiagramme aus Kapitel 5.2.1.2 lassen sich nun ableiten, sodass die Struktur der Attribute als benutzerdefinierte Datentypstruktur im Tornado Designer angelegt werden kann und die Methoden mittels Befehlsprogrammen erstellt werden können.

In Abbildung 35 ist der Aufbau des benutzerdefinierten Datentyps im Tornado Designer am Beispiel des Messgerätetyps CVS dargestellt. Mit der Einstellung Basistyp kann die Vererbung der Klasse Device bzw. des benutzerdefinierten Datentyps Device in Tornado erzeugt werden. Es wird lediglich die Vererbung bis zur Klasse CVS implementiert, da alle weiteren Subklassen hauptsächlich die Methoden zur Steuerung der Messgerätetypen beinhalten. Alle Attribute, die die Zustände des Messgeräts wiederspielen (Condition_x), müssen in die Klasse CVS übernommen werden. Da es sich um die Zustände eines AK-Protokoll gesteuerten Messgerätes handeln, wird die Bezeichnung AK_Condition_x gewählt.

Nummer	Name	Typ	Dimension	Einheit
1	Pressure_Inlet	Fließkommazahl	Skalar (Einzelwert)	mbar
2	Temp_Inlet	Fließkommazahl	Skalar (Einzelwert)	°C
3	Mainflow_Norm	Fließkommazahl	Skalar (Einzelwert)	m³/min
4	AK_Condition_1	Texttabelle	Skalar (Einzelwert)	-
5	AK_Condition_2	Texttabelle	Skalar (Einzelwert)	-
6	AK_Condition_3	Texttabelle	Skalar (Einzelwert)	-
7	AK_Condition_4	Texttabelle	Skalar (Einzelwert)	-
8	AK_Condition_5	Texttabelle	Skalar (Einzelwert)	-
9	AK_Condition_6	Texttabelle	Skalar (Einzelwert)	-

Datentyp

Name: CVS

Anzeigename (opt.):

Beschreibung:

Basistyp: Device

Externer Editor:

Abbildung 35: Implementierung der Datentypstruktur am Beispiel CVS, Quelle: Eigene Darstellung.

Die Datentypstruktur in Tornado kann bei allen Messgerätetypen nach der Vorlage des Klassendiagramms erstellt werden. In Abbildung 36 ist detailliert ersichtlich, dass sich auch die Unterstruktur (Instanzen der Klassen EMSDIL_Line und EMSDIL_Line_Component) des Messgerätetyps EMSDIL mit den benutzerdefinierten Datentypen ableiten lässt.

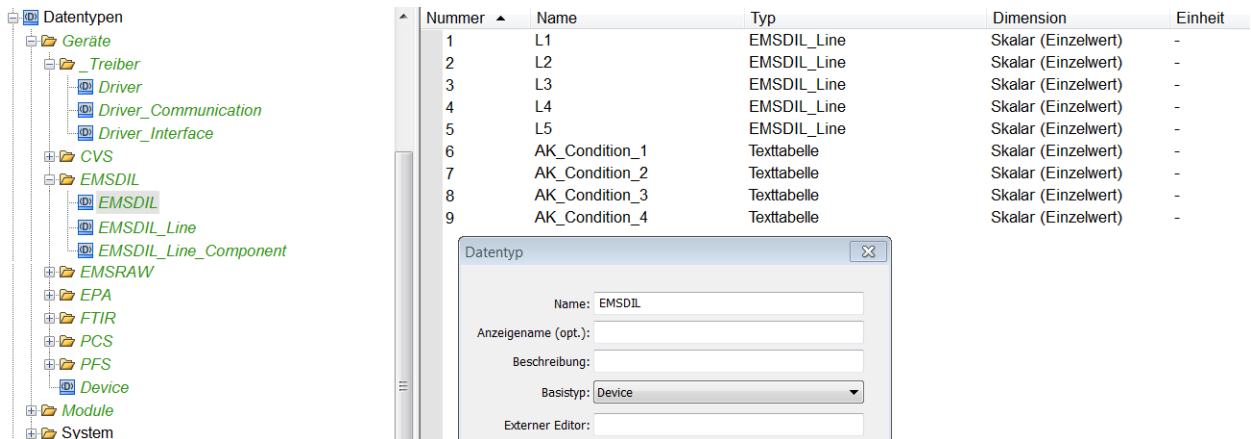


Abbildung 36: Implementierung der Datentypstruktur am Beispiel EMSDIL, Quelle: Eigene Darstellung.

Großer Vorteil bei der Aufbereitung dieser Struktur ist, dass somit entsprechend mehrere Variablen (Instanzen) mit demselben benutzerdefinierten Datentyp erstellt werden können. Werden bei einem Prüfstand beispielsweise zwei Analyseinheiten zur Messung des verdünnten Abgases verwendet, so kann eine Variable EMSDIL_1 und EMSDIL_2 mit dem benutzerdefinierten Datentyp EMSDIL erstellt werden. Etwaige Änderungen des benutzerdefinierten Datentyps werden danach bei beiden Variablen (Instanzen) angewendet.

Die Methoden der Klassendiagramme werden durch Befehlsprogrammen des Tornado Designers realisiert. Es können die objektorientierten Möglichkeiten (Vererbung und Überschreiben von Methoden) in Tornado nicht angewendet werden, da das Automatisierungssystem Tornado dafür nicht ausgelegt wurde. Dennoch werden die Befehlsprogramme der Messgerätetypen eigens strukturiert und in folgende drei Bereiche unterteilt:

- Abläufe
- Befehle
- Befehlsbibliothek

Unter den Abläufen werden die Befehlsprogramme gesammelt, bei denen eine Abfolge von Befehlen an das Messgerät gesendet wird. Damit kann eine Initialisierung, Finalisierung oder auch Kalibrierung des Messgeräts ausgelöst werden. Hinsichtlich des Klassendiagramms spiegeln die Abläufe somit die überschriebenen Methoden Initialisation() und Finalisation() wider. Mit den Befehlsbibliotheken der herstellerspezifischen Messgeräte werden mittels VBS-Funktionen die möglichen Befehle (AK-Befehle, TSI-Befehle etc.) gesammelt. Wie schon in den Klassendiagrammen konzipiert, werden die Befehlsbibliotheken in Antwort-, Steuer- und Einstellungsbefehle unterteilt. Der Bereich der Befehle wird zusätzlich implementiert, da dort einzelne Befehle als Befehlsprogramme aufgerufen werden. Der Grund dafür ist, dass bei Dialogfenstern in Tornado die Buttons mit einem Befehlsprogramm verknüpft werden. Alle aufgerufenen Befehle werden aber aus der Befehlsbibliothek entnommen.

In Abbildung 37 ist die Befehlsprogrammstruktur des Messgerätetyps CVS dargestellt. Alle herstellerspezifischen Geräte besitzen ihre eigene Befehlsprogrammstruktur.

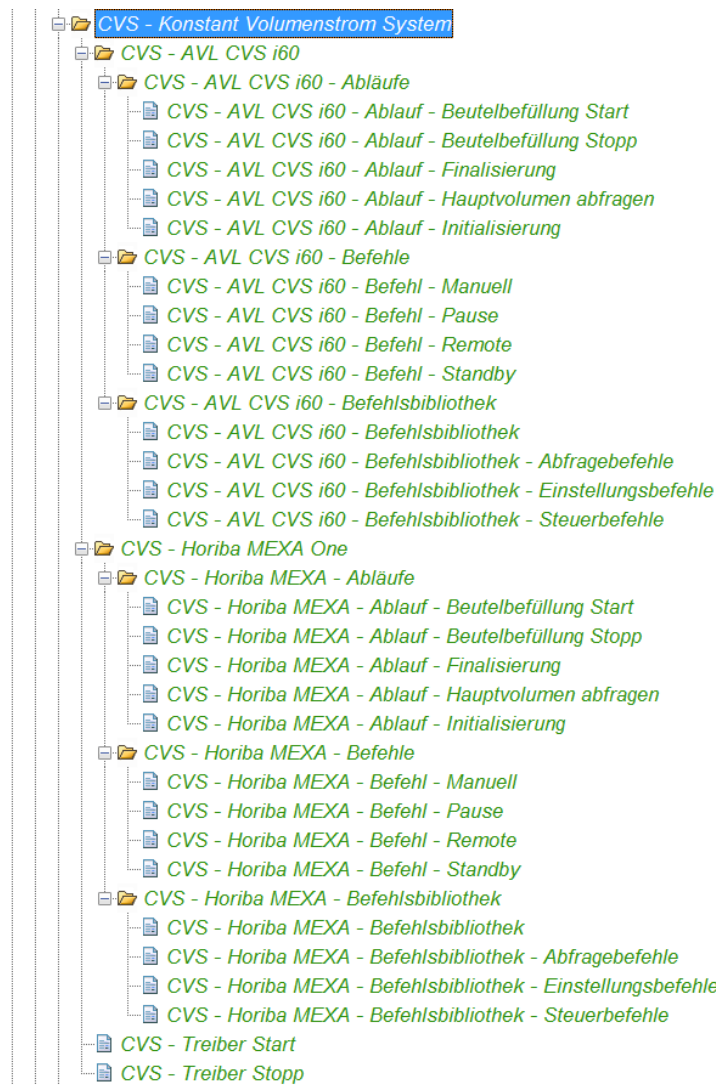


Abbildung 37: Befehlsprogrammstruktur am Beispiel CVS, Quelle: Eigene Darstellung.

Für den Aufbau der Befehlsprogramme wird die ‚Include‘-Funktionalität von Tornado genutzt. Diese Funktionalität ermöglicht es Befehlsprogramme in anderen Befehlsprogrammen bereitzustellen. Somit können alle VBS-Funktionen des inkludierten Befehlsprogramms genutzt werden. In Abbildung 38 ist das Befehlsprogramm des Initialisierungsablaufs der AVL i60 CVS dargestellt (CVS – AVL CVS i60 – Ablauf – Initialisierung). In diesem Befehlsprogramm wird der Ablauf wiederum nur als Funktion (CVS_AVLCVSi60_Initialisation) implementiert, womit dieser Ablauf wieder in weitere Befehlsprogramme inkludiert werden kann. Der Initialisierungsablauf besteht bei diesem Messgerät im Wesentlichen aus drei AK-Befehlen (SREM, STBY, ENOR). Der Syntax dieser Befehle wird mittels Funktionen (z. B. CVS_AVLCVSi60_STBY()) aus der inkludierten Befehlsbibliothek bereitgestellt. Im inkludierten Befehlsprogramm ‚Geräte - Befehl senden‘ ist die Funktion ‚AK_Cmd_Send‘ umgesetzt. Wie schon im Klassendiagramm dargestellt, soll diese Funktion das Senden von AK-Befehlen an den jeweiligen Messgeräten ermöglichen. In dieser Funktion werden auch alle gesendeten Befehle und empfangenen Antworten in ein Ergebnisprotokoll (Engl. Event Log) mitaufgezeichnet. Durch diese Funktion kann somit auch die Kommunikation zwischen Tornado und den Messgeräten nachvollzogen werden.

```

1  Sub main
2  End Sub
3
4  Class CVS_AVLCVSi60_Initialisation_Parameter_Struct
5      Dim TNorm
6      Dim pNorm
7  End Class
8
9  Sub CVS_AVLCVSi60_Initialisation(DrvName, CVS_AVLCVSi60_Init_Parameter)
10     Dim Device
11     Device = GetDeviceName(DrvName)
12     NumericValue(VariableID(CStr(Device&". ErrState"))) = 0
13     NumericValue(VariableID(CStr(Device&". InitState"))) = 1
14
15     Dim AKCmd
16
17     'Aktivieren des Remotebetriebs
18     AKCmd = CVS_AVLCVSi60_SREM()
19     Call AK_Cmd_Send(AKCmd, DrvName)
20
21     'Messgerät in Standby setzen
22     AKCmd = CVS_AVLCVSi60_STBY()
23     Call AK_Cmd_Send(AKCmd, DrvName)
24
25
26     'Einstellen der Normierungstemperatur und des Normierungsdrucks
27     AKCmd = CVS_AVLCVSi60_ENOR(CVS_AVLCVSi60_Init_Parameter.TNorm, CVS_AVLCVSi60_Init_Parameter.pNorm)
28     Call AK_Cmd_Send(AKCmd, DrvName)
29
30     NumericValue(VariableID(CStr(Device&". InitState"))) = 2
31 End Sub
32
33
34 'INCLUDES
35 Include("Geräte - Befehl senden")
681 Include("CVS - AVL CVS i60 - Befehlsbibliothek")
1850

```

Abbildung 38: Programmaufbau des Ablaufs der CVS-Initialisierung, Quelle: Eigene Darstellung.

6.2 Implementierung des Abgasprüfautomatisierungsmoduls

In diesem Abschnitt wird die Umsetzung der Entwürfe des Kapitels 5.2.1.3 näher erläutert. Die Strukturen der modellierten Klassendiagramme können ohne weitere Änderungen mittels der benutzerdefinierten Datentypen in Tornado implementiert werden. Um eine bessere Gliederung der Strukturen zu schaffen, wird die Nomenklatur angepasst. Die benutzerdefinierten Datentypen der Prüfstandsbeschreibung, der Geräteeinstellung, des Fahrzyklus und der Testaufgabe erhalten den Präfix TEMV_PAR, da es sich hierbei um Parametrierstrukturen des Abgasprüfautomatisierungssystems handelt. Die Ergebnisstrukturen der Phasen und Fahrzyklen werden mit dem Präfix TEMV_RSLT versehen.

Im nachfolgenden Kapitel wird hauptsächlich auf die Implementierung des Fahrzyklus eingegangen. Der benutzerdefinierte Datentyp wird entsprechend dem Entwurf aus Abbildung 28 erstellt. In Abbildung 39 ist der benutzerdefinierte Datentyp des Fahrzyklus dargestellt.

Nummer ▲	Name	Typ	Dimension	Einheit
1	Name	Text	Skalar (Einzelwert)	-
2	Dscr	Text	Skalar (Einzelwert)	-
3	Phase1	TEMV_PAR_Drivecycle_Phase	Skalar (Einzelwert)	-
4	SoakPhase1	TEMV_PAR_Drivecycle_Soakphase	Skalar (Einzelwert)	-
5	Phase2	TEMV_PAR_Drivecycle_Phase	Skalar (Einzelwert)	-
6	SoakPhase2	TEMV_PAR_Drivecycle_Soakphase	Skalar (Einzelwert)	-
7	Phase3	TEMV_PAR_Drivecycle_Phase	Skalar (Einzelwert)	-
8	SoakPhase3	TEMV_PAR_Drivecycle_Soakphase	Skalar (Einzelwert)	-
9	Phase4	TEMV_PAR_Drivecycle_Phase	Skalar (Einzelwert)	-
10	SoakPhase4	TEMV_PAR_Drivecycle_Soakphase	Skalar (Einzelwert)	-
11	Phase5	TEMV_PAR_Drivecycle_Phase	Skalar (Einzelwert)	-
12	SoakPhase5	TEMV_PAR_Drivecycle_Soakphase	Skalar (Einzelwert)	-
13	Phase6	TEMV_PAR_Drivecycle_Phase	Skalar (Einzelwert)	-
14	SoakPhase6	TEMV_PAR_Drivecycle_Soakphase	Skalar (Einzelwert)	-
15	Phase7	TEMV_PAR_Drivecycle_Phase	Skalar (Einzelwert)	-
16	SoakPhase7	TEMV_PAR_Drivecycle_Soakphase	Skalar (Einzelwert)	-
17	Phase8	TEMV_PAR_Drivecycle_Phase	Skalar (Einzelwert)	-
18	CycleStartMode	TEMV_PAR_StartMode	Skalar (Einzelwert)	-
19	CycleStopMode	TEMV_PAR_StopMode	Skalar (Einzelwert)	-

Abbildung 39: Aufbau des benutzerdefinierten Datentyps Fahrzyklus in Tornado, Quelle: Eigene Darstellung.

Nachdem die Datentypstruktur in Tornado implementiert ist, werden die Katalogelemente dazu angelegt, wobei jedes von ihnen einen bestimmten Fahrzyklus beschreibt. In Abbildung 40 sind die Implementierung der verschiedenen Fahrzyklen und der detaillierte Aufbau eines WLTC-Klasse-3b-Fahrzyklus zu sehen.

Name	Wert	Einheit
WLTC_Class3b		
Name	WLTC_Class3b	x -
Dscr	Worldwide harmonized Light vehicles Test Cycle (Class 3b)	x -
Phase1		-
Use	Ja (1)	x -
t		s
SpeedTable	WLTC_Class3_Low	x -
ShiftTable		-
SlopeTable		-
BagFillMode	nächstes Beutelpaar (1)	x -
FilterFillMode	nächster Filter (1)	x -
SoakPhase1		-
Phase2		-
SoakPhase2		-
Phase3		-
SoakPhase3		-
Phase4		-
SoakPhase4		-
Phase5		-
SoakPhase5		-
Phase6		-
SoakPhase6		-
Phase7		-
SoakPhase7		-
Phase8		-
CycleStartMode	Startmodus ohne Motorstartbestätigung	x -
CycleStopMode	Stoppmodus ohne Motorstoppbestätigung	x -

Abbildung 40: Fahrzyklenbezogene Katalogelemente in Tornado, Quelle: Eigene Darstellung.

In dieser Implementierung ist ersichtlich, dass die Schaltpunkttabelle vorerst nicht in die Definition der Fahrzyklen integriert wird. Die Entscheidung, welche Schaltpunkttabelle herangezogen werden soll, hängt im Wesentlichen vom zu überprüfenden Fahrzeug bzw. dem darin verbauten Getriebe ab. Je nachdem muss eine Schaltpunkttabelle für Automatikgetriebe oder Manuellgetriebe herangezogen werden. Des Weiteren ist bei Manuellgetrieben zwischen den möglichen Gängen (Fünf-, Sechs- und Sieben-Gang-Getriebe) zu unterscheiden. Diese Zuweisung geschieht über eine weitere Struktur, in der die Zuordnung der Schaltpunkttabelle auf die entsprechende Phase definiert wird. Abbildung 41 zeigt diese Zuordnung am Beispiel eines NEDC. Vor Ablauf des Fahrzyklus werden die Informationen des Getriebes und des Fahrzyklus in einem Befehlsprogramm ermittelt und die Schaltpunkttabellen dem Fahrzyklus zugeordnet.

Name	Wert	Einheit
Mapping_Schaltpunkte - NEDC-M-5-Standard		
ShiftTable_Phase1	ECE_M5_shift	x -
ShiftTable_Phase2	EUDC_M5_shift	x -
ShiftTable_Phase3		-
ShiftTable_Phase4		-
ShiftTable_Phase5		-
ShiftTable_Phase6		-
ShiftTable_Phase7		-
ShiftTable_Phase8		-

Abbildung 41: Implementierung der Schaltpunkttabellenzuweisung, Quelle: Eigene Darstellung.

Nachdem die Struktur der Variablen bzw. der benutzerdefinierten Datentypen und die Implementierung der Katalogelemente festgelegt sind, wird die modellierte Zustandsmaschine der Fahrzyklensteuerung mittels eines VBS-Befehlsprogramms in Tornado implementiert. Dafür wird zunächst eine allgemeine Zustandsmaschine erstellt. In Abbildung 42 sind die Grundfunktionen der programmierten Zustandsmaschine dargestellt.

```

42 Sub main
43   Dim StateVars
44   Set StateVars = New StateVars_Struct
45
46   '----- Init Variable -----
47   StateVars.ActState = State_State1
48   StateVars.FirstEntry = True
49   StateVars.Terminate = False
50
51   '----- Statesmaschine -----
52   Do
53     Select Case StateVars.ActState
54
55       '==== State1 ====
56       Case State_State1 Call Do_State1(StateVars)
57       '==== State2 ====
58       Case State_State2 Call Do_State2(StateVars)
59       '==== State3 ====
60       Case State_State3 Call Do_State3(StateVars)
61       '==== State'4 ====
62       Case State_State4 Call Do_State4(StateVars)
63       '==== State5 ====
64       Case State_State5 Call Do_State5(StateVars)
65
66     End Select
67
68     Call Sleep(10)
69   Loop Until StateVars.Terminate
70 End Sub
71
74 '-----
75 ' Sub: SetState
76 '-----
77 Sub SetState (ByRef StateVars, ReqState)
78   StateVars.ActState = ReqState
79   StateVars.FirstEntry = True
80 End Sub
81
82 '-----
83 ' Sub: EnterState
84 '-----
85 Sub EnterState (ByRef StateVars, ActState)
86   Call Trace("Enter State " & Chr(34) & ActState & Chr(34))
87   StateVars.FirstEntry = False
88 End Sub
89
90 '-----
91 ' Sub: Do_State1
92 '-----
93 Sub Do_State1(ByRef StateVars)
94   '----- Entry -----
95   If StateVars.FirstEntry Then
96     Call EnterState (StateVars, "State1")
97   End If
98
99   '----- Cyclic -----
100
101   '----- Exit -----
102   Call SetState (StateVars, State_State2)
103 End Sub

```

Abbildung 42: Programmausschnitt des allgemeinen Zustandsautomaten, Quelle: Eigene Darstellung.

Die Zustände des Zustandsautomaten sind in drei Bereiche aufgeteilt. Den Eingangsbereich (Entry), den zyklischen Bereich (Cyclic) und den Ausgangsbereich (Exit). Dadurch können Steuerungen und Aktionen bei erstmaligem Eintritt und abschließendem Austritt des Zustandes ausgeführt werden. Sowohl die Zustandsmaschine der Fahrzyklensteuerung als auch jene der Beutelsteuerung können aus dieser allgemein gebildeten Zustandsmaschine abgeleitet werden. Es müssen lediglich die Zustände der entworfenen Modelle (siehe Abbildung 32 und Abbildung 33) eingebunden werden. Für den Ablauf der Zustandsmaschinen muss jedoch noch eine Logik erstellt werden. Die Überlegungen zu diesen Logiken werden in den folgenden Kapiteln erarbeitet.

6.2.1 Logik der Fahrzyklensteuerung

In der Logik der Zustandsmaschine der Fahrzyklensteuerung wird eine Sequenzbeschreibung implementiert, um festzustellen, in welchen Zustand als Nächstes gesprungen werden muss. Diese Beschreibung enthält Informationen über die Abfolge der Phasen und Soakphasen eines Fahrzyklus. Die nachfolgenden Auflistungen stellen die Sequenzabfolgen einiger bekannter Fahrzyklen dar.

NEDC

P1	P2
----	----

WLTC_Class3b

P1	P2	P3	P4
----	----	----	----

FTP-75

P1	P2	S2	P3
----	----	----	----

Die Geschwindigkeits-, Schaltpunkt- und Steigungstabellen werden mittels Parametertabellen phasenweise implementiert. Für den Zusammenbau der Parametertabellen zu einem gesamten fahrzyklenbezogenen Verlauf muss zunächst die Abhängigkeit vom Fahrerleitsystem berücksichtigt werden. Eine Soakphase unterbricht den Verlauf des Fahrerleitsystems, wodurch sich verschiedene Verlaufsteile für das Fahrerleitsystem ergeben. Im oben angeführten Beispiel des FTP-75 unterbricht die Soakphase 2 (S2) den Verlauf des Fahrerleitsystems. Es entstehen zwei Teile: Teil 1 (T1) mit der Phase 1 (P1) und Phase 2 (P2), sowie Teil 2 (T2) mit der Phase 3 (P3). Wird im Extremfall ein Fahrzyklus mit sieben Soakphasen erstellt, müssen somit für das Fahrerleitsystem acht Verlaufsteile vorbelegt werden. Die folgende Darstellung des FTP-75 stellt den Sachverhalt noch einmal dar.

FTP-75

P1	P2	S2	P3
T1			T2

Die Erweiterung des Fahrzyklus zu einem Mehrfachzyklus muss in dieser Überlegung auch mitberücksichtigt werden. Die Varianten des variablen und fix definierten Mehrfachzyklus werden anhand der drei Beispielfahrzyklen erläutert. In der folgenden Auflistung soll zunächst die Sequenzabfolge eines variablen Mehrfachzyklus dargestellt werden. Es werden wieder die drei Fahrzyklen als Beispiel herangezogen, wobei die Mehrfachzyklus-Soakphase mit M und die Fahrzyklen mit F dargestellt werden. Ähnlich zur normalen Soakphase unterbricht diese den Verlauf des Fahrerleitsystems.

NEDC mit zwei Wiederholungen

F1			F2		
P1	P2	M	P1	P2	M
T1			T1		

WLTC_Class3b mit zwei Wiederholungen

F1					F2				
P1	P2	P3	P4	M	P1	P2	P3	P4	M
T1					T1				

FTP-75 mit zwei Wiederholungen

F1					F2				
P1	P2	S2	P3	M	P1	P2	S2	P3	M
T1			T2		T1			T2	

Diese Untersuchung zeigen, dass die Teile sich wiederholen und somit die Anzahl der Teile in der Variante des variablen Mehrfachzyklus unverändert bleiben. Als nächstes werden verschiedene Fahrzyklen mit der Variante des fix definierten Mehrfachzyklus dargestellt.

NEDC mit drei Wiederholungen

F1		F2		F3	
P1	P2	P1	P2	P1	P2
T1					

WLTC_Class3b mit zwei Wiederholungen

F1				F2			
P1	P2	P3	P4	P1	P2	P3	P4
T1							

FTP-75 mit zwei Wiederholungen

F1				F2			
P1	P2	S2	P3	P1	P2	S2	P3
T1			T2				T3

Da bei fixen Mehrfachzyklen ein direkter Übergang der Phasen zwischen zweier Fahrzyklen erfolgt, ist bei Fahrzyklen ohne Soakphase immer nur ein Verlaufsteil notwendig. Bei Fahrzyklen mit einer Soakphase muss der Verlauf dahingegen in mehrere Teile gegliedert werden. Welche Teile bzw. welche Phasenverläufe zusammengefasst werden, wird anhand fiktiv erstellter Fahrzyklen untersucht. Es werden mehrere Varianten analysiert, wobei die folgenden aufgelisteten Sequenzabfolgen den Sachverhalt erklären.

Vierphasiger Fahrzyklus mit einer Soakphase und vier Wiederholungen

F1				F2				F3				F4							
P1	P2	S2	P3	P4	P1	P2	S2	P3	P4	P1	P2	S2	P3	P4	P1	P2	S2	P3	P4
T1			T2				T2				T2				T3				

Vierphasiger Fahrzyklus mit zwei Soakphasen und drei Wiederholungen

F1						F2						F3					
P1	P2	S2	P3	S3	P4	P1	P2	S2	P3	S3	P4	P1	P2	S2	P3	S3	P4
T1			T2		T3				T2		T3				T2		T4

Vierphasiger Fahrzyklus mit drei Soakphasen und zwei Wiederholungen

F1							F2						
P1	S1	P2	S2	P3	S3	P4	P1	S1	P2	S2	P3	S3	P4
T1		T2		T3		T4		T2		T3		T5	

Die Sequenzabfolgen zeigen, dass ein sich wiederholendes Muster erkennbar ist, sobald eine Soakphase im Fahrzyklus festgelegt ist. Bei einem fix definierten Mehrfachzyklus mit einer Soakphase muss ein weiterer Verlaufsteil erstellt werden, der den ersten und letzten Teil zusammenfasst (gelb markiert). Ein Fahrzyklus mit sieben Soakphasen benötigt in der Variante eines fix definierten Fahrzyklus somit neun Teile.

Anhand dieser Erkenntnisse wird entschieden, dass drei Bereiche der Parametertabellengruppen erstellt werden. Der erste Bereich fungiert als Bibliothek der phasenbezogenen Geschwindigkeits-, Schaltpunkt- und Steigungsverläufe, die in die Fahrzyklenstruktur und die Zuordnungsstruktur der Schaltpunkte eingetragen werden. Im zweiten Bereich der Parametertabellengruppe werden die Verlaufsteile des Fahrerleitsystems vorbereitet. Die dritte Parametertabellengruppe fasst die Geschwindigkeits-, Schaltpunkt- und Steigungsverläufe des aktuell laufenden Teils zu einer Parametertabelle zusammen. Abbildung 43 stellt diese Implementierung dar.

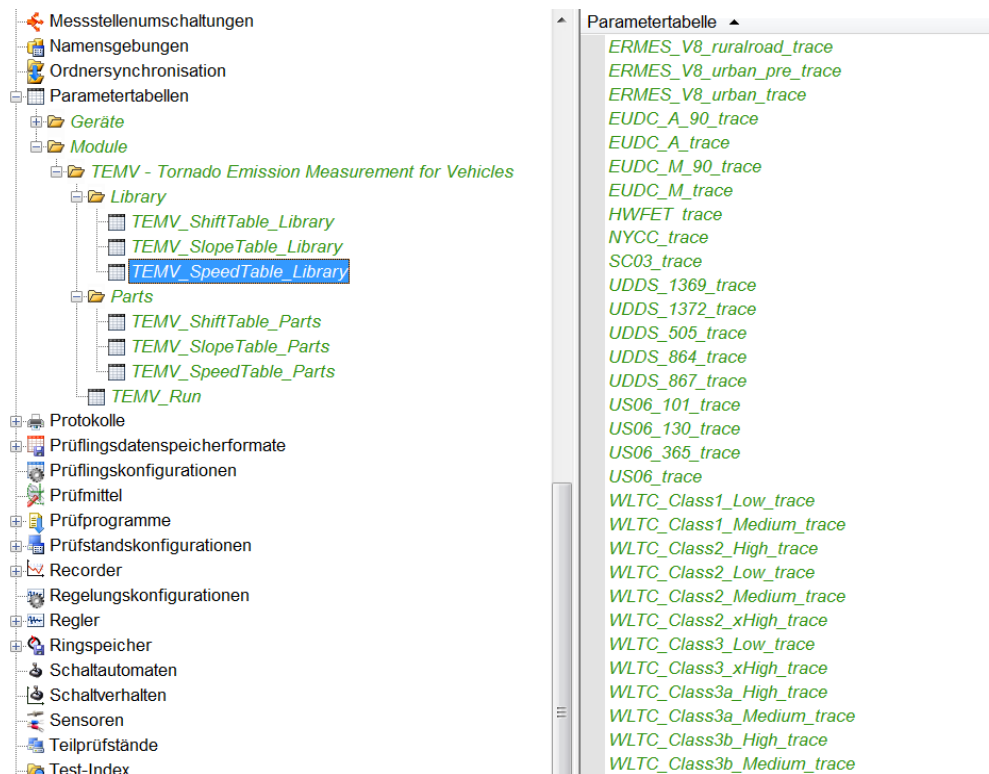


Abbildung 43: Implementierung der Parametertabellen, Quelle: Eigene Darstellung.

6.2.2 Logik der Beutelsteuerung

In der Beutelsteuerung muss der Ablauf eines Mehrfachzyklus auch betrachtet werden. Jeder Beutel ist einer bestimmten Phase zugeordnet, was in der Parametrierung der Phase mit dem Attribut BagFillMode bestimmt wird. Wird die Auswahl ‚nächstes Beutelpaar‘ parametriert, so beginnt die Steuerung mit der Befüllung des ersten Beutelpaares und setzt bei Phasenwechsel mit dem nächsten Beutelpaar bei einem Phasenwechsel fort. Die nachfolgende Auflistung zeigt den Zusammenhang der Phasen (P) und Beutel (B) bei Verwendung eines Mehrfachzyklus (der Fahrzyklus wird mit F abgekürzt).

NEDC mit sechs Wiederholungen

F1		F2		F3		F4		F5		F6	
P1	P2	P1	P2	P1	P2	P1	P2	P1	P2	P1	P2
B1	B2	B3	B4	B1	B2	B3	B4	B1	B2	B3	B4

WLTC_Class3b mit drei Wiederholungen

F1				F2				F3			
P1	P2	P3	P4	P1	P2	P3	P4	P1	P2	P3	P4
B1	B2	B3	B4	B1	B2	B3	B4	B1	B2	B3	B4

FTP-75 mit fünf Wiederholungen

F1				F2				F3				F4				F5			
P1	P2	S2	P3	P1	P2	S2	P3	P1	P2	S2	P3	P1	P2	S2	P3	P1	P2	S2	P3
B1	B2		B3	B4	B1		B2	B3	B4		B1	B2	B3		B4	B1	B2		B3

Die einfachste Variante ist, diese Sequenz bis zu einer maximalen Anzahl an Fahrzyklen vorzubereiten. Jedoch soll die Steuerung ohne eine Obergrenze funktionieren. Daher ist die Idee, die Sequenz so weit vorzudefinieren, bis sich eine Wiederholung der Sequenz ergibt (gelb markierter Bereich). Mit dieser Logik kann die Beutelsteuerung unabhängig von der Anzahl der Fahrzyklen eines Mehrfachzyklus arbeiten.

6.2.3 Messgeräteansteuerung im Abgasprüfautomatisierungsmodul

Es gibt zwei wesentliche Strukturen, die die Ansteuerung der Messgeräte im Abgasprüfautomatisierungsmodul beschreiben: Einerseits die Struktur der Prüfstandsbeschreibung, welche wichtige Informationen über die verbauten Messgeräte des Prüfstands sammelt, andererseits die Struktur der Geräteeinstellung, mit der parametriert wird, welche Messgeräte bei einem beauftragten Testzyklus verwendet werden. Während des gesamten Prozesses muss somit entschieden werden, ob und welches Messgerät angesteuert wird. Anhand der Initialisierung der Messgeräte wird die implementierte Struktur dieser Befehlsprogrammlogik erläutert.

Zunächst wird die Initialisierung aller Messgerätetypen eingeleitet. Durch die Struktur der Geräteeinstellung kann entschieden werden, welche Messgerätetypen angesteuert bzw. initialisiert werden. Abbildung 44 zeigt den Aufbau dieses Befehlsprogramms. Alle Befehlsprogramme der Messgerätetypen werden ohne

Warten auf Beendigung aufgerufen, damit alle Messgeräte ihre Initialisierungen gleichzeitig beginnen. Am Ende des Befehlsprogramms wird gewartet, bis alle Initialisierungen beendet wurden.

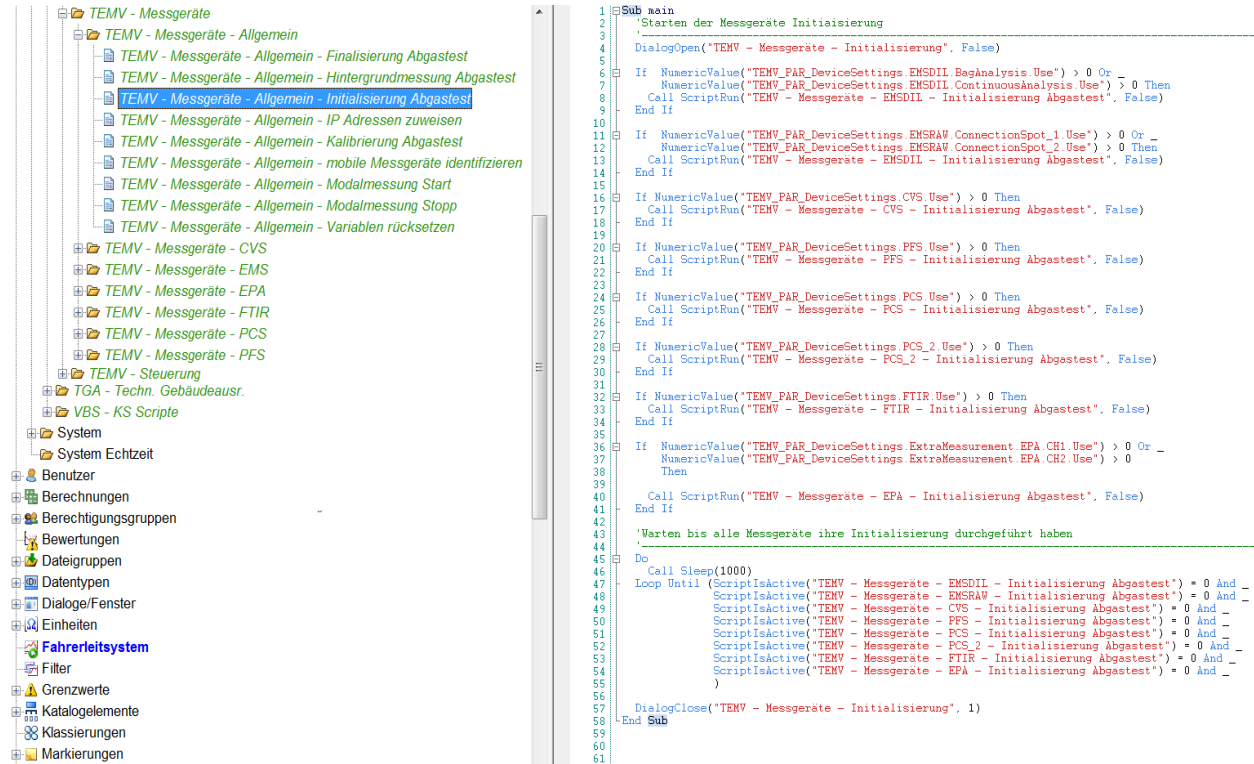


Abbildung 44: Programmaufbau der Messgerätetypensteuerung, Quelle: Eigene Darstellung.

In den allgemeinen Befehlsprogrammen der Messgerätetypen wird entschieden, welches Messgerät angesteuert wird. In Abbildung 45 wird anhand des Messgerätetyps CVS diese Entscheidung dargestellt. Im linken Bereich ist die Befehlsprogrammstruktur des Messgerätetyps CVS ersichtlich, während im rechten Bereich das Befehlsprogramm ‚TEMV – Messgeräte – CVS – Initialisierung Abgastest‘ dargestellt wird.

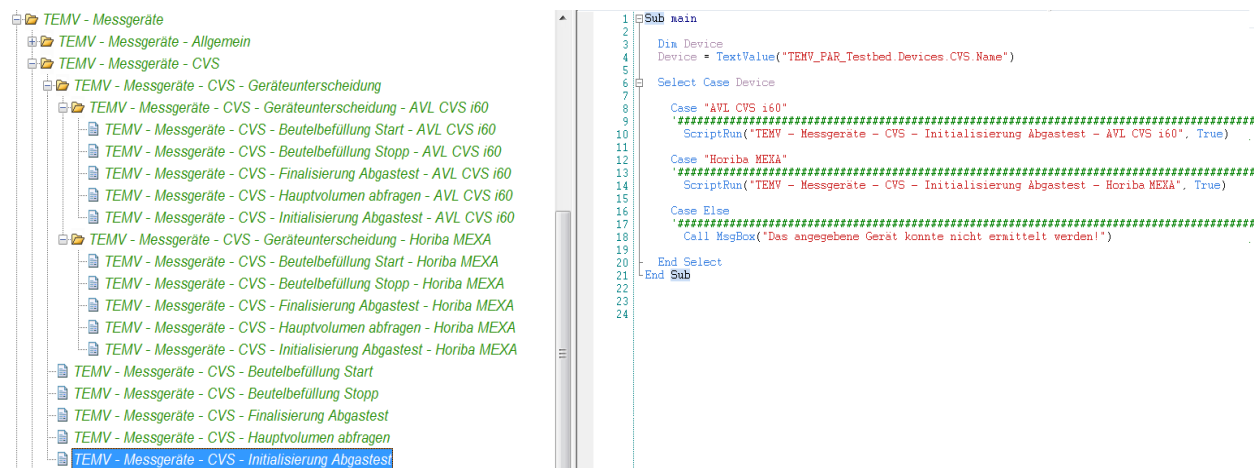


Abbildung 45: Programmaufbau der Messgeräteansteuerung, Quelle: Eigene Darstellung.

In diesem Befehlsprogramm wird das spezifische Messgerät anhand der Prüfstandsbeschreibungsstruktur ausgewählt. In Abbildung 46 ist das spezifische Befehlsprogramm der AVL CVS i60 zu sehen. Hier ist auch das inkludierte Ablauf-Befehlsprogramm aus Kapitel 6.1 wiederzufinden.

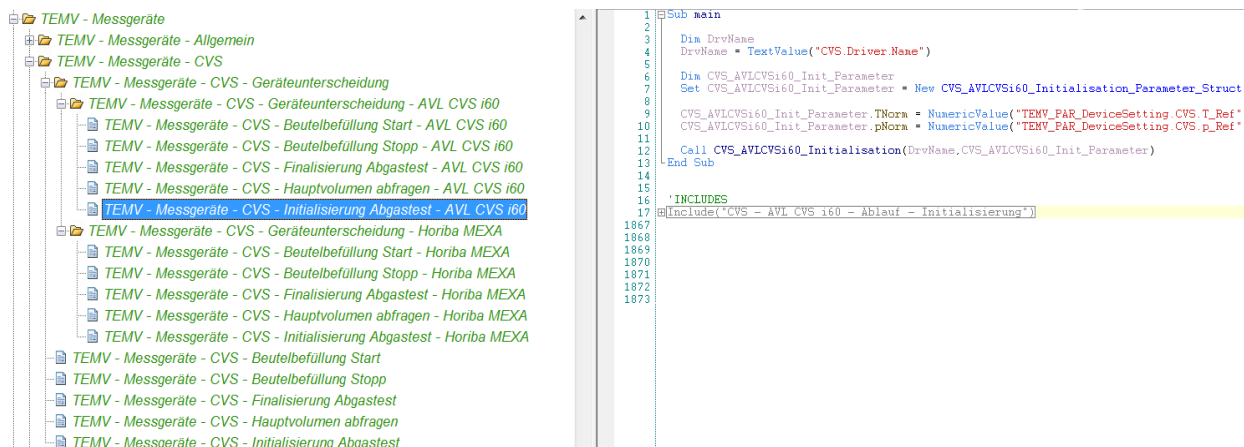


Abbildung 46: Programmablauf der Initialisierung, Quelle: Eigene Darstellung.

Durch den strukturierten Aufbau der Messgeräteeinbindung im Abgasprüfautomatisierungsmodul, ergeben sich folgende Vorteile:

- Ist ein Messgerät bereits in der Befehlsprogrammstruktur des Abgasprüfautomatisierungsmoduls vorhanden, kann dieses durch einfache Parametrierung der Prüfstandsbeschreibung ausgetauscht werden (z. B. wenn an einem Prüfstand das Messgerätesetup vom Hersteller AVL zum Hersteller Horiba wechselt).
- Bei einer Implementierung eines neuen Messgeräts muss lediglich das Befehlsprogramm, an dem die Auswahl des Messgeräts stattfindet (z. B. ‚TEMV – Messgeräte – CVS – Initialisierung Abgastest‘), um die neue Auswahl erweitert werden.
- Durch die Kapselung der Messgeräteauswahl mittels dieses Befehlsprogrammaufbaus muss die eigentliche Steuerung des Abgasprüfautomatisierungsmoduls nicht verändert werden.

6.3 Funktionstest

Die Funktionalität der Struktur und der Steuerung des Abgasprüfautomatisierungsmoduls wird in drei Stufen getestet. Anhand eines FTP-75-Fahrzyklus werden diese Funktionstests im folgenden Kapitel beispielhaft erläutert.

Die erste Stufe des Funktionstests soll verifizieren, ob die entwickelte Logik für die Sequenztafel der Fahrzyklensteuerung und der Beutelsteuerung funktioniert. Wie schon Kapitel 5.2.2.3 definiert wird in der Aktivität ‚Vorbereitung der Prüflaufdaten‘ der Prüfzyklus festgelegt. In dieser Aktivität ist die programmierte Logik der Aufteilung der Fahrprofile aus Kapitel 6.2.1 und die Erstellung der Sequenztafel für die Beutelsteuerung aus Kapitel 6.2.2 enthalten. Mit einem Unittest wird getestet, dass diese Funktionen korrekt implementiert sind. In Abbildung 47 ist zunächst die Aufteilung der Fahrkurve eines FTP-75 in ersichtlich (links Teil 1, bestehend aus Phase 1 und Phase 2; rechts Teil 2, bestehend aus Phase 3).

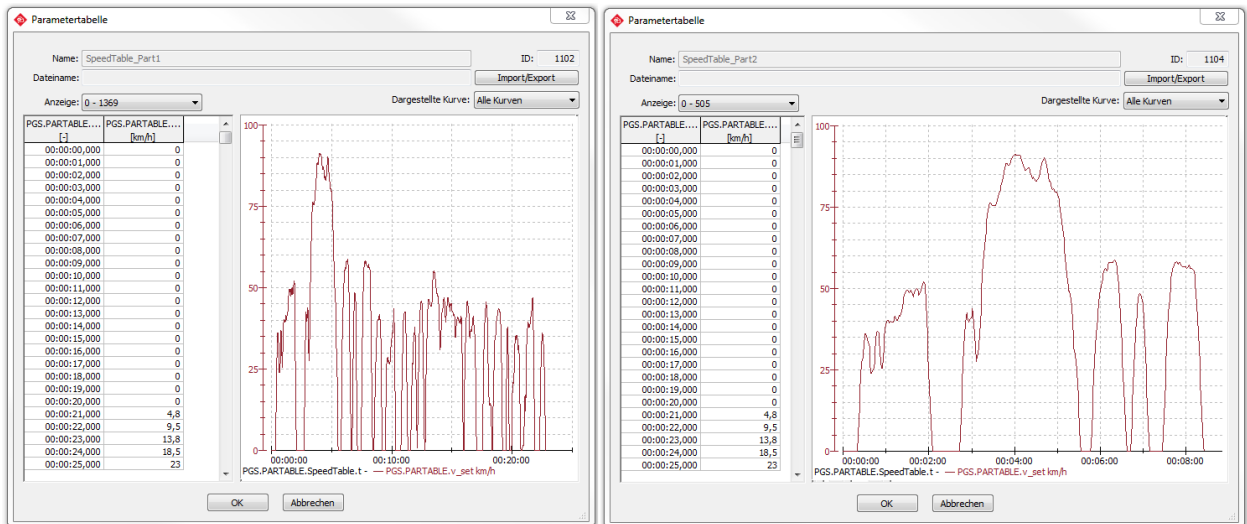


Abbildung 47: Aufteilung des FTP-75-Geschwindigkeitsprofils, Quelle: Eigene Darstellung.

In Abbildung 48 ist die Logik der Beutelsteuerung mittels einer Array-Struktur des FTP-75 ersichtlich. Wie schon Kapitel 6.2.2 dargestellt, sind zwölf Beutelwechsel in vier Fahrzyklen notwendig um ein wiederholendes Muster der Sequenztafel zu erhalten. Mit 'Id' ist die fortlaufende Beutelpaarnummer definiert.

Test_BagData	{...} (4 Elemente)
Bags	{...} (13 Elemente)
(0)	{...} (3 Elemente)
(1)	{...} (3 Elemente)
Cycle	1
Id	1
Phase	1
(2)	{...} (3 Elemente)
Cycle	1
Id	2
Phase	2
(3)	{...} (3 Elemente)
Cycle	1
Id	3
Phase	3
(4)	{...} (3 Elemente)
Cycle	2
Id	4
Phase	1
(5)	{...} (3 Elemente)
Cycle	2
Id	1
Phase	2
(6)	{...} (3 Elemente)
(7)	{...} (3 Elemente)
(8)	{...} (3 Elemente)
(9)	{...} (3 Elemente)
(10)	{...} (3 Elemente)
(11)	{...} (3 Elemente)
(12)	{...} (3 Elemente)

Abbildung 48: Beutelpaarsequenz eines FTP-75, Quelle: Eigene Darstellung.

In der nächsten Stufe des Funktionstests soll nur die Zustandsmaschine der Fahrzyklensteuerung ohne Ansteuerung der Messgeräte getestet werden. Für diesen Test werden in der Geräteeinstellung keine Messgeräte zur Verwendung ausgewählt und die Variablenstruktur des Fahrzyklus mit dem Katalogelement des FTP-75 parametrisiert. Da in der Geräteeinstellung die Messgerädetypen EMSDIL und CVS deaktiviert sind, wird die Zustandsmaschine der Beutelsteuerung nicht aktiviert. In der Zustandsmaschine der Fahrzyklensteuerung sind Logging-Funktionen implementiert, die es ermöglichen den Status des Ablaufs der Zustandsmaschine nachzuvollziehen. In Abbildung 49 ist der Eventlog eines

getestet FTP-75-Fahrzyklus zu sehen. Der Wechsel der Zustände der Fahrzyklensteuerung wird mit ‚Enter State‘ angezeigt. Dieser Test zeigt, dass die zeitliche Abfolge der Zustandsmaschine korrekt abgearbeitet wird.

```

07:55:56,246 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "StartEngine"
07:55:56,248 ### Statuslog ### || ||| Lade Teil 1 der Parametertabellen |||
07:56:03,738 ### Statuslog ### || Startmodus - START drücken, Motor starten und WEITER drücken
07:56:04,939 ### Statuslog ### || Bedieneinheit - "START" wurde gedrückt!
07:56:06,339 ### Statuslog ### || Bedieneinheit - "WEITER" wurde gedrückt!
07:56:06,339 ### Statuslog ### || Taste wurde innerhalb des Zeitbereichs gedrückt!
07:56:06,396 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "Phase"
07:56:06,396 ### Statuslog ### || ||| Fahrzyklus: 1 /// Phase: 1 |||
07:56:06,396 ### Statuslog ### || ||| Fahrzyklus: 1 /// Phase: 1 |||
07:56:06,396 ### Statuslog ### || ||| Fahrzyklus: 1 /// Phase: 1 |||
07:56:06,396 ### Statuslog ### || Start=0 Stop=505 t=505
08:04:31,420 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "Phase"
08:04:31,420 ### Statuslog ### || ||| Fahrzyklus: 1 /// Phase: 2 |||
08:04:31,420 ### Statuslog ### || ||| Fahrzyklus: 1 /// Phase: 2 |||
08:04:31,420 ### Statuslog ### || Start=505 Stop=1369 t=864
08:18:55,415 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "StopEngine"
08:18:56,518 ### Statuslog ### || Bedieneinheit - "STOPP" wurde gedrückt!
08:18:56,521 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "Soakphase"
08:18:56,521 ### Statuslog ### || ||| Fahrzyklus: 1 /// Soakphase: 2 |||
08:18:56,535 ### Statuslog ### || ||| Fahrzyklus: 1 /// Soakphase: 2 |||
08:18:56,535 ### Statuslog ### || ||| Fahrzyklus: 1 /// Soakphase: 2 |||
08:18:56,535 ### Statuslog ### || Dauer: min=540s / max=660s
08:27:56,524 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "StartEngine"
08:27:56,524 ### Statuslog ### || ||| Lade Teil 2 der Parametertabellen |||
08:28:04,961 ### Statuslog ### || Startmodus - Motor starten und START drücken!
08:28:06,062 ### Statuslog ### || Bedieneinheit - "START" wurde gedrückt!
08:28:06,119 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "Phase"
08:28:06,119 ### Statuslog ### || ||| Fahrzyklus: 1 /// Phase: 3 |||
08:28:06,119 ### Statuslog ### || ||| Fahrzyklus: 1 /// Phase: 3 |||
08:28:06,119 ### Statuslog ### || ||| Fahrzyklus: 1 /// Phase: 3 |||
08:28:06,119 ### Statuslog ### || Start=0 Stop=505 t=505
08:36:31,140 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "StopEngine"
08:36:32,331 ### Statuslog ### || Bedieneinheit - "STOPP" wurde gedrückt!
08:36:33,142 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "EndOfTest"
08:36:33,142 ### Statuslog ### || ||| Testende |||
08:36:33,143 ### Statuslog ### || ||| Testende |||
08:36:33,143 ### Statuslog ### || ||| Testende |||

```

Abbildung 49: Ausschnitt des Eventlogs eines FTP-75-Testablaufs ohne Geräteansteuerung, Quelle: Eigene Darstellung.

Im letzten Schritt des Funktionstests wird die Kommunikation der Messgeräte während des Ablaufs getestet. Dazu wird zunächst der Prüfstands Aufbau bzw. die Beschreibung der Messgeräte eines realen Prüfstands in der Variablenstruktur der Prüfstandsbeschreibung implementiert. Als Testobjekt wird der Aufbau und die Beschreibung des Prüfstands der FH Joanneum herangezogen, da dieser die wesentlichsten Messgerätetypen eines Abgasrollenprüfstands (EMSDIL, CVS, PFS, PCS) beinhaltet. Im Anschluss wird in der Variablenstruktur der Geräteeinstellung, die Verwendung dieser Messgerätetypen und die Einstellparameter definiert. Durch die Verwendung der Messgerätetypen EMSDIL und CVS wird auch die Zustandsmaschine der Beutelsteuerung aktiviert. In Abbildung 50 ist das Zusammenspiel der Beutelsteuerung und der Fahrzyklensteuerung im erstellten Eventlog ersichtlich. Für die Simulation der Messgeräte wird ein AK-Simulator herangezogen. Dieser kann auf bestimmte AK-Befehle passende AK-Antworten wiedergeben. Da die Beutelanalyse mehrere aufeinanderfolgende Befehle beinhaltet, ist aus Übersichtlichkeitsgründen nur eine simulierte Wartezeit von 15 Sekunden implementiert. Mit den Zeichen ‚>>‘ wird das Senden eines Befehls und mit ‚<<‘ das Empfangen einer Antwort symbolisiert. Davor steht immer der Name des Messgeräts, mit dem gerade kommuniziert wird. Es ist die erste und zweite Phase des FTP-75 dargestellt, in der der Wechsel der Zustände des ersten und zweiten Beutelpaares der Beutelsteuerung ersichtlich ist.


```

13:24:05,237 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "StartEngine"
13:24:05,239 ### Statuslog ### || ||| Lade Teil 1 der Parametertabellen |||
13:24:05,559 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Empty" / Bag: 1
13:24:05,561 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Empty" / Bag: 2
13:24:05,563 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Empty" / Bag: 3
13:24:05,566 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Empty" / Bag: 4
13:24:12,696 ### Statuslog ### || Startmodus - START drücken, Motor starten und WEITER drücken
13:24:13,797 ### Statuslog ### || Bedieneinheit - "START" wurde gedrückt!
13:24:15,097 ### Statuslog ### || Bedieneinheit - "WEITER" wurde gedrückt!
13:24:15,102 ### Statuslog ### || Taste wurde innerhalb des Zeitbereichs gedrückt!
13:24:15,125 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "Phase"
13:24:15,125 ### Statuslog ### || ||| Fahrzyklus: 1 /// Phase: 1 |||
13:24:15,125 ### Statuslog ### || ||| Start=0 Stop=505 t=505 |||
13:24:15,131 <CVS - AVL CVS i60> >> SDST KO 5
13:24:15,139 <CVS - AVL CVS i60> << SDST 0
13:24:15,140 <PFS - AVL PSS i60> >> SCGO KO 1
13:24:15,143 <PFS - AVL PSS i60> << SCGO 0
13:24:15,143 <PCS - AVL 489 > >> SINTI KO
13:24:15,145 <PCS - AVL 489 > >> SINTI 0
13:24:15,145 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Filling" / Bag: 1
13:24:15,150 <CVS - AVL CVS i60> >> SPVS KO 4
13:24:15,162 <CVS - AVL CVS i60> << SPVS 0
13:24:15,162 <CVS - AVL CVS i60> >> SBPF KO 1
13:24:15,165 <CVS - AVL CVS i60> << SBPF 0
13:32:40,167 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "Phase"
13:32:40,167 ### Statuslog ### || ||| Fahrzyklus: 1 /// Phase: 2 |||
13:32:40,167 ### Statuslog ### || ||| Start=505 Stop=1369 t=864 |||
13:32:40,172 <CVS - AVL CVS i60> >> SDST KO 5
13:32:40,173 <CVS - AVL CVS i60> << SDST 0
13:32:40,174 <PFS - AVL PSS i60> >> SCGO KO 2
13:32:40,178 <PFS - AVL PSS i60> << SCGO 0
13:32:40,178 <PCS - AVL 489 > >> AIKO KO
13:32:40,178 <PCS - AVL 489 > << AIKO 0 38136,2 3,814E+4 75,52 76323313,6 7,632E+7 151135,27 1,511E+5 505 # # # # #
13:32:40,181 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Filled" / Bag: 1
13:32:40,182 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Line State "Analyse" / Bag: 1 / Line: 3
13:32:40,184 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Filling" / Bag: 2
13:32:40,189 <CVS - AVL CVS i60> >> SPVS KO 4
13:32:40,191 <CVS - AVL CVS i60> << SPVS 0
13:32:40,191 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Analyse" / Bag: 1
13:32:40,199 ### Statuslog ### || Simulierte Beutelanalyse - Dauer: 15s
13:32:40,199 <CVS - AVL CVS i60> >> SBPF KO 2
13:32:40,209 <CVS - AVL CVS i60> << SBPF 0
13:32:42,166 <PFS - AVL PSS i60> >> AVAB KO 1
13:32:42,181 <PFS - AVL PSS i60> << AVAB 0 0
13:32:42,185 <CVS - AVL CVS i60> >> AVAI KO 1
13:32:42,209 <CVS - AVL CVS i60> << AVAI 0 0,0000 506
13:32:55,207 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Line State "Analysed" / Bag: 1 / Line: 3
13:32:55,215 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Analysed" / Bag: 1
13:32:55,223 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Evacuate" / Bag: 1
13:32:55,229 <CVS - AVL CVS i60> >> SBEV KO 1
13:32:55,243 <CVS - AVL CVS i60> << SBEV 0
13:32:55,243 <CVS - AVL CVS i60> >> SBEV KO 5
13:32:55,253 <CVS - AVL CVS i60> << SBEV 0
13:33:00,800 ### Statuslog ### || TEMV - Steuerung - Beutel - Zustandsmaschine: Enter State "Empty" / Bag: 1
13:47:04,160 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "StopEngine"
13:47:05,263 ### Statuslog ### || Bedieneinheit - "STOPP" wurde gedrückt!
13:47:05,271 ### Statuslog ### || TEMV - Steuerung - Fahrzyklus - Zustandsmaschine: Enter State "Soakphase"
13:47:05,272 ### Statuslog ### || ||| Fahrzyklus: 1 /// Soakphase: 2 |||
13:47:05,281 ### Statuslog ### || ||| Dauer: min=540s / max=660s |||
13:47:05,281 ### Statuslog ### ||

```

Abbildung 50: Ausschnitt des Eventlogs eines FTP-75-Testablaufs mit Geräteansteuerung, Quelle: Eigene Darstellung.

Der dreistufige Funktionstest hat gezeigt, dass die implementierten Strukturen und die Logik der Abläufe des Abgasprüfautomatisierungsmoduls korrekt umgesetzt sind. Erste Funktionstests an einem realen Abgasrollenprüfstand (Prüfstand der FH Joanneum) haben zusätzliche Erkenntnisse ergeben, die rückwirkend in die Weiterentwicklung des Abgasprüfautomatisierungsmoduls einfließen.

6.4 Erkenntnisse

Das erlangte Wissen im Bereich des Requirements-Engineerings und der Aufbereitung der Softwarearchitektur mittels der UML-Diagrammfamilie führten dazu, dass nicht nur eine objektbasierte Struktur modelliert, sondern auch generische Programmabläufe entwickelt werden konnten. Im Vergleich zu den bestehenden Projekten ergeben sich somit folgende Vorteile:

- Eine allgemeinere, detailliertere Struktur der Fahrzyklen konnte entwickelt werden, mit der durch einfache Parametrierung von Katalogelementen, unterschiedliche Fahrzyklen definiert werden können. Aus dieser Struktur konnte ein generischer Ablauf, mittels eines Zustandsautomaten entwickelt werden, der die Steuerung aller Fahrzyklen abdeckt.
- Der Aufbau der Messgeräteeinstellung ist strukturierter, wodurch der Austausch von verschiedenen Messgeräten unkomplizierter und flexibler durchzuführen ist. Wird ein Messgerät an einem Prüfstand durch ein anderes ausgewechselt, so muss lediglich die Prüfstandsbeschreibung neu parametrieren werden. Wird beispielsweise ein neues Messgerät desselben Messgerätetyps benötigt, so kann dieser durch den strukturierten Befehlsprogrammaufbau deutlich schneller implementiert werden.
- Durch die beiden neu entwickelnden Zustandsautomaten (Fahrzyklensteuerung und Beutelsteuerung), ist der Ablauf der Prüfzyklen und Ansteuerung der Messgeräte hinsichtlich der Wartbarkeit und Testbarkeit einfacher handzuhaben. Beide Zustandsmaschinen beinhalten die wesentlichsten Aktionen während eines Prüfzyklus.
- Die erstellten Variablenstrukturen Fahrzyklus, Prüfstandsbeschreibung, Geräteeinstellung und Testaufgabe ermöglichen eine gekapselte Parametrierung des Abgasprüfautomatisierungsmoduls. Mit der Struktur des Fahrzyklus, können vorab die gesetzeskonformen Fahrzyklen implementiert werden. Jedoch hat der Benutzer immer noch die Möglichkeit spezifische Fahrzyklen mittels der Katalogelemente zu definieren. Die Prüfstandsbeschreibung muss zu Beginn einer Inbetriebnahme einmalig parametrieren werden, um dem Abgasprüfautomatisierungsmodul mitzuteilen welche Messgeräte am Prüfstand vorhanden sind. Die Geräteeinstellung und die Testaufgabe ermöglichen es, mehrere Varianten von Prüfzyklen zu definieren.
- Das Abgasprüfautomatisierungsmodul kann durch seinen gekapselten Aufbau in verschiedene Projekte importiert werden. Durch definierte Aktivitäten im Ablauf des Moduls (z. B. der Messgeräte Initialisierung bzw. Finalisierung) können Ansteuerungen externer Systeme, wie z. B. einer Rolle oder der Abtriebsmaschinen eines Antriebstrangprüfstands, einfach in den Prüfprozess integriert werden.
- Die Inbetriebnahmezeit eines Abgasprüfstands ist durch das neue Abgasprüfautomatisierungsmodul wesentlich geringer. Der Inbetriebnehmer / die Inbetriebnehmerin muss lediglich die Kommunikation der Messgeräte prüfen und die Variablenstruktur der Prüfstandsbeschreibung korrekt parametrieren. Hinsichtlich der Steuerung und des Ablaufs der Prüfzyklen müssen keine Anpassungen während der Inbetriebnahme mehr durchgeführt werden.

7 FAZIT UND AUSBLICK

Zusammenfassung

Am Anfang dieser Arbeit wurden alle Informationen zum Aufbau und der Abarbeitung einer Abgasmessung auf einem Rollenprüfstand für Gesamtfahrzeuge zusammengetragen. Diese Untersuchung veranschaulichte, welche wesentlichen Anforderungen das Abgasprüfautomatisierungsmodul erfüllen muss. Im nächsten Schritt wurden die wichtigsten Merkmale des Requirements-Engineerings und der Entwicklung einer Softwarearchitektur und somit die Anfangsphasen der Softwareentwicklung erarbeitet. Diese Untersuchung gab Aufschluss darauf, wie das Abgasprüfautomatisierungsmodul von Beginn an entwickelt werden soll und mit welchen Hilfsmitteln die Struktur und der Ablauf konzipiert werden können. Im Praxisteil dieser Arbeit wurde die UML-Diagrammfamilie als geeignete Darstellung für diesen Anwendungsfall gewählt und erste Konzepte der Struktur und des Verhaltens des Abgasprüfautomatisierungsmoduls erarbeitet. In der Konzeptionierungsphase ging hervor, dass sowohl der Prüfzyklus als auch die benötigten Messgeräte in eine objektbasierte Struktur überführt werden konnten. Des Weiteren konnte ein generischer Programmablauf entworfen werden, mit dem die Steuerung aller Fahrzyklen abgedeckt werden konnte. Mit Hilfe dieser Masterarbeit konnte somit geklärt werden, wie im Bereich der Abgasautomatisierung allgemeine Strukturen für die Programmierung geschaffen werden können und welche Softwareentwicklungsvorgehensmethodik für diesen Anwendungsfall am geeignetsten ist. Die Zielsetzung ein standardisiertes Softwaremodul im Prüfstandsautomatisierungssystem Tornado für die Abgasprüfung von Gesamtfahrzeugen zu konzipieren und umzusetzen, wurde somit erreicht.

Während der Konzeptionierung und Implementierung des Abgasprüfautomatisierungsmoduls wurden noch weitere Funktionen umgesetzt, die im Rahmen der Masterarbeit nicht vorgesehen waren.

Auswertung der Emissionen und Bereitstellen eines Ergebnisprotokolls

Wie schon in Kapitel 2.3 erwähnt, müssen nach der Messung der Abgaskonzentrationen und der Partikel weitere Auswertungs- und Berechnungsschritte durchgeführt werden, um das endgültige Emissionsergebnis zu erhalten. Dazu wurde ein Befehlsprogramm zur Analyse der phasen- und fahrzyklenbezogenen Ergebnisse konzipiert und implementiert. Dieses Programm durchläuft die zyklische Messdatei eines Prüfzyklus und ermittelt die Mittelwerte und Integrationsergebnisse (z. B. mittlere Konzentration des Abgases in den einzelnen Phasen). Im Anschluss wird dem Benutzer ein Ergebnisprotokoll bereitgestellt.

Bereitstellung einer grafischen Oberfläche

Für die Benutzer des Abgasprüfautomatisierungsmoduls wurde eine grafische Oberfläche erstellt. In Abbildung 51 ist das Haupt- bzw. Übersichtsfenster des Abgasprüfautomatisierungsmoduls veranschaulicht. Am linken Rand können über eine Toolbar die Messgerätefenster geöffnet werden. Im restlichen Fenster kann der Benutzer den Verlauf der gefahrenen Kurve verfolgen und über eine Ablaufmeldungshistorie die aktuellen und vergangenen Aktionen während eines Prüflaufs betrachten. Im zweiten Fenster (siehe Abbildung 52) wurde das Fahrerleitsystem von Tornado eingebunden, das dem Fahrer / der Fahrerin zur Orientierung im Fahrzyklus dient.

Fazit und Ausblick

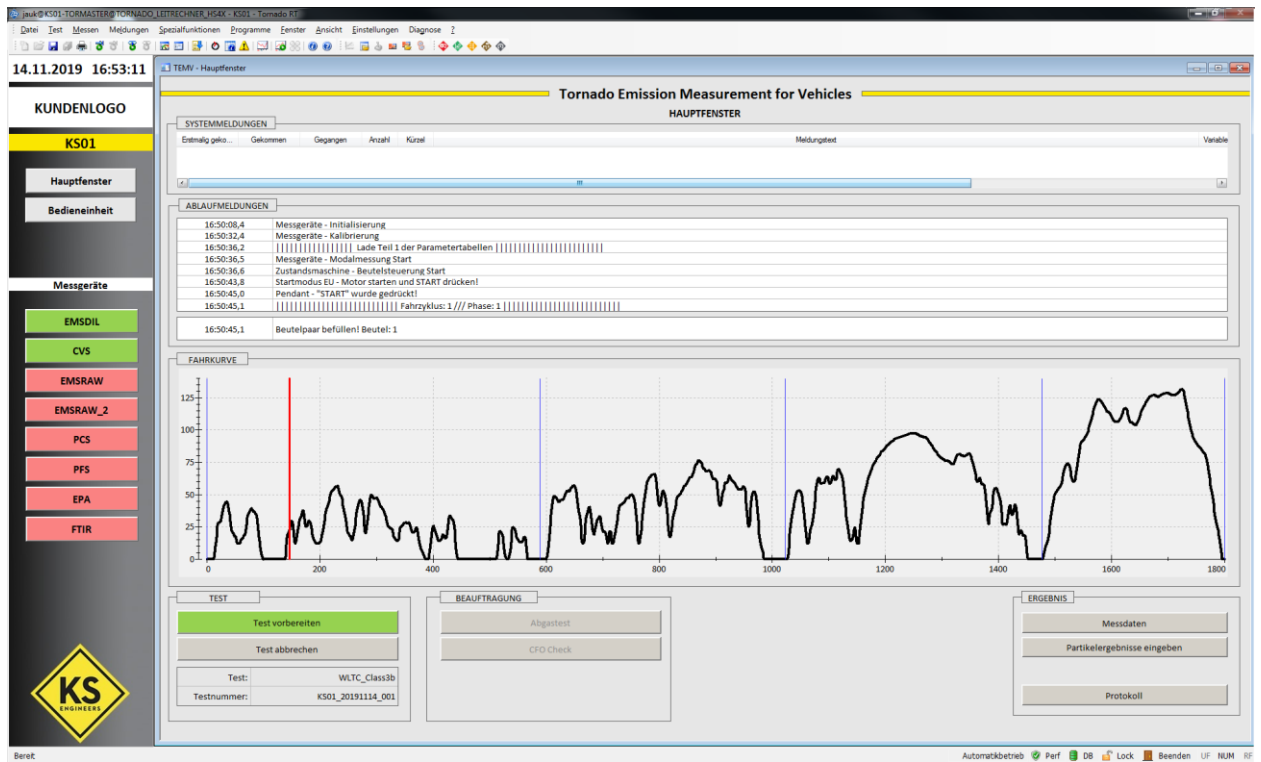


Abbildung 51: Hauptfenster des Abgasprüfautomatisierungsmoduls, Quelle: Eigene Darstellung.

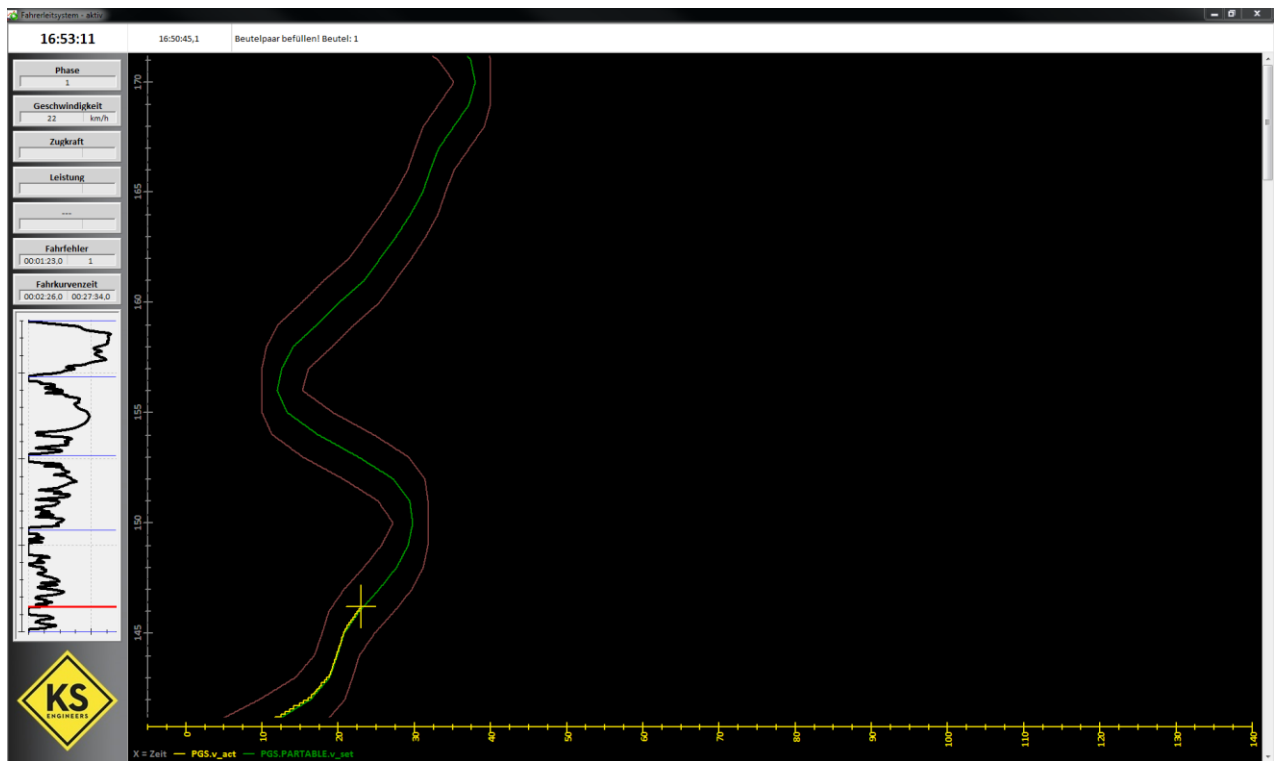


Abbildung 52: Fahrerleitsystemfenster des Abgasprüfautomatisierungsmoduls, Quelle: Eigene Darstellung.

Ausblick

Derzeit wird das entwickelte Abgasprüfautomatisierungsmodul bereits an zwei Prüfständen eingesetzt. Weitere Prüfstände sollen in näherer Zukunft folgen. Die nächsten Schritte in der Weiterentwicklung des Abgasprüfautomatisierungsmoduls sind die Konzeptionierung und Implementierung der restlichen Typprüfungen. Für die geplante Implementierung der Prüfzyklen einer Typ-4-Prüfung (Bestimmung der Verdunstungsemissionen aus Fahrzeugen) wird die im Rahmen dieser Masterarbeit erarbeitete Vorgehensmethodik angewendet. Des Weiteren werden bereits Überlegungen hinsichtlich der Automatisierbarkeit der Funktionstests angestellt, um das Softwaremodul bei etwaigen Änderungen schneller verifizieren und validieren zu können.

Durch Anpassungen der Gesetzgebungen ergeben sich ständig neue Anforderungen im Bereich der Abgasprüfautomatisierung. Dadurch sind fortlaufend Neuerungen sowie Änderungen am entwickelnden Abgasprüfautomatisierungsmodul vorzunehmen. Im Hinblick auf den Wandel zur Elektromobilität werden sich sicherlich laufend Veränderungen im Aufbau der Prüfstände, aber auch in der Abwicklung der Prüfzyklen ergeben. Es bleibt jedoch abzuwarten, inwieweit diese kommenden Prüfverfahren vom jetzigen Stand abweichen werden.

LITERATURVERZEICHNIS

Gedruckte Werke (14)

AVL (Hrsg.) (2012): *AVL iGEM CVS*, 3. Auflage, AVL Emission Test Systems GmbH, Gaggenau

Borgeest, Kai (2017): *Manipulation von Abgaswerten*, 1. Auflage, Springer Vieweg Verlag, Wiesbaden

Braess, Hans-Hermann; Seiffert, Ulrich (2013): *Handbuch Kraftfahrzeugtechnik*, 7. Auflage, Springer Vieweg Verlag, Wiesbaden

Dumke, Reiner (2003): *Software Engineering*, 4. Auflage, Vieweg Verlag, Wiesbaden

Dunkel, Jürgen; Holitschke, Andreas (2003): *Softwarearchitektur für die Praxis*, 1. Auflage, Springer Verlag, Heidelberg

Gadatsch, Andreas (2019): *Datenmodellierung - Einführung in die Entity-Relationship-Modellierung und das Relationsmodell*, 2. Auflage, Springer Vieweg Verlag, Wiesbaden

Kleuker, Stephan (2018): *Grundkurs Software Engineering mit UML*, 4. Auflage, Springer Vieweg Verlag, Wiesbaden

Klingenberg, Horst (1995): *Automobil-Meßtechnik: Band C: Abgasmeßtechnik*, 1. Auflage, Springer Verlag, Heidelberg

Krypczyk, Veikko; Bochkur, Olena (2018): *Handbuch für Softwareentwickler*, 1. Auflage, Rheinwerk Verlag, Bonn

Pischinger, Stefan; Seiffert, Ulrich (2016): *Handbuch Kraftfahrzeugtechnik*, 8. Auflage, Springer Vieweg Verlag, Wiesbaden

Pohl, Klaus; Rupp, Chris (2015): *Basiswissen Requirements Engineering*, 4. Auflage, dpunkt Verlag, Heidelberg

Reif, Konrad (2015): *Abgastechnik für Verbrennungsmotoren*, 1. Auflage, Springer Vieweg Verlag, Wiesbaden

Schatten, Alexander; Demolsky, Markus (2010): *Best Practice Software Engineering*, 1. Auflage, Springer Spektrum Verlag, Heidelberg

Tabeling, Peter (2006): *Softwaresysteme und ihre Modellierung - Grundlagen, Methoden und Techniken*, 1. Auflage, Springer-Verlag, Heidelberg

Online-Quellen (25)

ADAC (2019): *Vergleich Abgasnormen: Moderne Diesel sind sehr sauber*

<https://www.adac.de/rund-ums-fahrzeug/abgas-diesel-fahrverbote/abgasnorm/euro-6d-temp/> [Stand: 01.12.2019]

Bachmann, Charles (o.J.): *Data Structure Diagrams*

<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.172.3370&rep=rep1&type=pdf> [Stand: 01.12.2019]

Continental (2019): *Continental Emission Booklet 2019*

https://www.continental-automotive.com/getattachment/8f2dedad-b510-4672-a005-3156f77d1f85/EMISSIONBOOKLET_2019.pdf [Stand: 01.12.2019]

Delphi (2019): *Delphi Emission Booklet 2019-2020*

<https://d2ou7ivda5raf2.cloudfront.net/sites/default/files/2019-05/2019-2020%20Passenger%20Car%20%26%20Light-Duty%20Vehicles.pdf> [Stand: 01.12.2019]

DemoSys (o.J.): *Requirements-Engineering*

https://www.demosys.de/L_requirements-engineering.html [Stand: 01.12.2019]

DieselNet (o.J.): *Emission Standards - EU: Cars and Light Trucks*

<https://www.dieselnet.com/standards/eu/ld.php> [Stand: 01.12.2019]

DieselNet (o.J.): *Emission Test Cycles*

<https://www.dieselnet.com/standards/cycles/index.php> [Stand: 01.12.2019]

DieselNet (o.J.): *FTP-75*

<https://www.dieselnet.com/standards/cycles/ftp75.php> [Stand: 01.12.2019]

DieselNet (o.J.): *Worldwide Harmonized Light Vehicles Test Cycle*

<https://www.dieselnet.com/standards/cycles/wltp.php> [Stand: 01.12.2019]

Engeljehring, Kurt (2016): *AVL Emission Testing Handbook 2016 V1.0 PA3088E*

<https://www.avl.com/documents/10138/2703362/AVL+Handbook+2016.pdf> [Stand: 01.12.2019]

Engeljehring, Kurt (2017): *AVL - Abgasgesetzgebung RDE/WLTP/EVAP*

<https://www.avl.com/documents/1982862/4664597/01+Engeljehring+-+Abgasgesetzgebung+RDE+-+WLTP+-+EVAP+-+Neuigkeiten+%26+Herausforderungen.pdf> [Stand: 01.12.2019]

Hausenberger, Stefan (2018): *Umweltauswirkungen des Verkehrs Teil I*

https://www.ivt.tugraz.at/assets/files/download/skripten/Skriptum_Umweltauswirkungen_Verkehr_Teill_1_2018.pdf [Stand: 01.12.2019]

Japan Automobile Standard Internationalization Center (o.J.): *Automobile Type Approval Handbook for Japanese Certification ("Blue Book")*

https://www.jasic.org/e/08_publication/bb/20_handbook.htm [Stand: 01.12.2019]

Keller, Mario; Hausberger, Stefan; Wüthrich, Phillip (2017): *HBEFA Version 3.3*

https://www.hbefa.net/d/documents/HBEFA33_Hintergrundbericht.pdf [Stand: 01.12.2019]

Kristl, Seibt & Co GmbH (2019): *Kristl, Seibt & Co*

<http://www.ksengineers.at/> [Stand: 01.12.2019]

Matzer, Klaus; Rexeis, Martin (2016): *Proposal for test description for cars and LCV for chassis dyno*

https://ermes-group.eu/web/system/files/filedepot/10/Test_Description_Cars_and_LCV_V3.pdf [Stand: 01.12.2019]

Österreich GV (o.J.): *Fahrzeugklassen*

https://www.oesterreich.gv.at/themen/freizeit_und_strassenverkehr/kfz/Seite.061800.html [Stand: 01.12.2019]

Rouse, Margaret (2016): *Unified Modeling Language (UML)*

<https://www.computerweekly.com/de/definition/Unified-Modeling-Language-UML> [Stand: 01.12.2019]

SYSML (o.J.): *SysML Open Source Project - What is SysML?*

<https://sysml.org/> [Stand: 01.12.2019]

Transportpolicy (o.J.): *US: Light-Duty: Emissions*

<https://www.transportpolicy.net/standard/us-light-duty-emissions/> [Stand: 01.12.2019]

Transportpolicy (o.J.): *US: Vehicle Definitions*

<https://www.transportpolicy.net/standard/us-vehicle-definitions/> [Stand: 01.12.2019]

TSI (2007): *Model 3772/3771 Condensation Particle Counter - Operation and Service Manual*

https://wmo-gaw-wcc-aerosol-physics.org/files/cpc_3772-3771.pdf [Stand: 01.12.2019]

UNECE (o.J.): *Mission*

<https://www.unece.org/mission.html> [Stand: 01.12.2019]

Van der Hoek, Jasper (2018): *Pursuing a Full Agile Software Development Life Cycle*

<https://www.mendix.com/blog/pursuing-a-full-agile-software-lifecycle/> [Stand: 01.12.2019]

Verband der Automobilindustrie (o.J.): *WLTP - weltweit am Start für realitätsnähere Ergebnisse beim Kraftstoffverbrauch*

<https://www.vda.de/de/themen/umwelt-und-klima/WLTP-realitaetsnaehere-Ergebnisse-beim-Kraftstoffverbrauch/WLTP-Wie-realitaetsnah-ist-der-WLTP.html> [Stand: 01.12.2019]

Normen (2)

UNECE (Hrsg.) (2018): *ECE/TRANS/180/Add.15/Amend.4: United Nations Global Technical Regulation No. 15*

EU (Hrsg.) (2017): *EU 2017/1151: VERORDNUNG (EU) 2017/1151*

ABBILDUNGSVERZEICHNIS

Abbildung 1: Aktuelle weltweite Emissionsgesetzgebungen, Quelle: Continental (2019), Online-Quelle [01.12.2019], S. 18.....	3
Abbildung 2: Entwicklung der Emissionsgrenzwerte, Quelle: Delphi (2019), Online-Quelle [01.12.2019], S. 1 (leicht modifiziert).....	10
Abbildung 3: Phaseinteilung des WLTC Class 3b, Quelle: Pischinger/Seiffert (2016), S. 335.	14
Abbildung 4: Phaseinteilung des FTP-75, Quelle: Pischinger/Seiffert (2016), S. 335.....	15
Abbildung 5: Schema eines Abgasrollenprüfstandes, Quelle: Klingenberg (1995), S. 262.....	16
Abbildung 6: Messtechnik an Abgasrollen, Quelle: Engeljehring (2016), Online-Quelle [01.12.2019], S. 55 (leicht modifiziert).....	17
Abbildung 7: Software-Entwicklungslebenszyklus, Quelle: Van der Hoek (2018), Online-Quelle [01.12.2019].....	21
Abbildung 8: Sprachschablone, Quelle: DemoSys (o.J.), Online-Quelle [01.12.2019].....	24
Abbildung 9: 4+1 Sichtenmodell, Quelle: Schatten/Demolsky (2010), S. 207.	26
Abbildung 10: Überblick der UML 2 Diagrammfamilie, Quelle: Schatten/Demolsky (2010), S. 165.	28
Abbildung 11: Beispiel eines ER-Modells, Quelle: Gadatsch (2019), S. 11.....	30
Abbildung 12: Strukturtypen eines informationellen Systemmodells, Quelle: Tabeling (2006), S. 255.....	30
Abbildung 13: Projekthierarchie der Tornado-Masterprojekte, Quelle: Eigene Darstellung.....	31
Abbildung 14: Oberfläche des Tornado Designer, Quelle: Eigene Darstellung.	32
Abbildung 15: Start des Prüfprogramms eines WLTC im TU-Graz-Projekt, Quelle: Eigene Darstellung...	35
Abbildung 16: ERMES V8 mit Phasenaufteilung, Quelle: Matzer/Rexeis (2016), Online-Quelle [01.12.2019], S. 8 (leicht modifiziert).	36
Abbildung 17: Aufbau der Parametertabelle. Quelle: Eigene Darstellung.	38
Abbildung 18: Allgemeiner Aufbau des TORMASTER-Projekts. Quelle: Eigene Darstellung.	46
Abbildung 19: Paketdiagramm der Geräte, Quelle: Eigene Darstellung.....	47
Abbildung 20: Klassendiagramm Device, Quelle: Eigene Darstellung.	48
Abbildung 21: Erstentwurf der Struktur für den Gerätetyp CVS, Quelle: Eigene Darstellung.....	49
Abbildung 22: Klassendiagramm CVS, Quelle: Eigene Darstellung.....	51
Abbildung 23: Klassendiagramm EMSDIL, Quelle: Eigene Darstellung.....	52
Abbildung 24: Klassendiagramm PCS, Quelle: Eigene Darstellung.....	53
Abbildung 25: Klassendiagramm PFS, Quelle: Eigene Darstellung.	54
Abbildung 26: Klassendiagramm Testbed, Quelle: Eigene Darstellung.	56

Abbildung 27: Klassendiagramm DeviceSetting, Quelle: Eigene Darstellung.	59
Abbildung 28: Klassendiagramm Drivecycle, Quelle: Eigene Darstellung.	60
Abbildung 29: Klassendiagramm Testtask, Quelle: Eigene Darstellung.	63
Abbildung 30: Klassendiagramm PhaseResults, Quelle: Eigene Darstellung.	65
Abbildung 31: Klassendiagramm DrivecycleResults, Quelle: Eigene Darstellung.	66
Abbildung 32: Zustandsautomat der Fahrzyklensteuerung, Quelle: Eigene Darstellung.	68
Abbildung 33: Zustandsmaschine der Beutelsteuerung, Quelle: Eigene Darstellung.	70
Abbildung 34: Aktivitätsdiagramm des Prüfprogramms, Quelle: Eigene Darstellung.	74
Abbildung 35: Implementierung der Datentypstruktur am Beispiel CVS, Quelle: Eigene Darstellung.	75
Abbildung 36: Implementierung der Datentypstruktur am Beispiel EMSDIL, Quelle: Eigene Darstellung.	76
Abbildung 37: Befehlsprogrammstruktur am Beispiel CVS, Quelle: Eigene Darstellung.	77
Abbildung 38: Programmaufbau des Ablaufs der CVS-Initialisierung, Quelle: Eigene Darstellung.	78
Abbildung 39: Aufbau des benutzerdefinierten Datentyps Fahrzyklus in Tornado, Quelle: Eigene Darstellung.	79
Abbildung 40: Fahrzyklenbezogene Katalogelemente in Tornado, Quelle: Eigene Darstellung.	79
Abbildung 41: Implementierung der Schaltpunkttabellenzuweisung, Quelle: Eigene Darstellung.	80
Abbildung 42: Programmausschnitt des allgemeinen Zustandsautomaten, Quelle: Eigene Darstellung.	80
Abbildung 43: Implementierung der Parametertabellen, Quelle: Eigene Darstellung.	83
Abbildung 44: Programmaufbau der Messgerätetypenansteuerung, Quelle: Eigene Darstellung.	85
Abbildung 45: Programmaufbau der Messgeräteeinsteuerung, Quelle: Eigene Darstellung.	85
Abbildung 46: Programmablauf der Initialisierung, Quelle: Eigene Darstellung.	86
Abbildung 47: Aufteilung des FTP-75-Geschwindigkeitsprofils, Quelle: Eigene Darstellung.	87
Abbildung 48: Beutelparsequenz eines FTP-75, Quelle: Eigene Darstellung.	87
Abbildung 49: Ausschnitt des Eventlogs eines FTP-75-Testablaufs ohne Geräteansteuerung, Quelle: Eigene Darstellung.	88
Abbildung 50: Ausschnitt des Eventlogs eines FTP-75-Testablaufs mit Geräteansteuerung, Quelle: Eigene Darstellung.	89
Abbildung 51: Hauptfenster des Abgasprüfautomatisierungsmoduls, Quelle: Eigene Darstellung.	92
Abbildung 52: Fahrerleitsystemfenster des Abgasprüfautomatisierungsmoduls, Quelle: Eigene Darstellung.	92

TABELLENVERZEICHNIS

Tabelle 1: Euro-Grenzwerte der Fahrzeugklasse M, Quelle: Delphi (2019), Online-Quelle [01.12.2019], S. 8 ff.	7
Tabelle 2: Emissionsgrenzwerte nach Tier 3 und LEV III, Quelle: Continental (2019), Online-Quelle [01.12.2019], S. 104.....	8
Tabelle 3: Prüfvorschriften für die Typgenehmigung, Quelle: Continental (2019), Online-Quelle [01.12.2019], S. 55 (leicht modifiziert).	12
Tabelle 4: Fahrzyklen der Typ-1-Prüfung, Quelle: DieselNet (o.J.), Online-Quelle [01.12.2019].	14
Tabelle 5: Auswahlkriterium der WLTC Fahrzyklen, Quelle: DieselNet (o.J.), Online-Quelle [01.12.2019].	15
Tabelle 6: Messverfahren für Schadstoffe, Quelle: Reif (2015), S. 143.	18
Tabelle 7: Neue Fahrzyklen aus dem HBEFA, Quelle: Keller/Hausberger/Wüthrich (2017), Online-Quelle [01.12.2019].....	36
Tabelle 8: Messgeräte der TU Graz, Quelle: Eigene Darstellung.	37
Tabelle 9: Messgeräte der FH Joanneum, Quelle: Eigene Darstellung.....	39
Tabelle 10: Messgeräte des OEM, Quelle: Eigene Darstellung.	41
Tabelle 11: Funktionale Anforderungen, Quelle: Eigene Darstellung.....	44
Tabelle 12: Qualitätsanforderungen, Quelle: Eigene Darstellung.	44
Tabelle 13: Nomenklatur der Messgerätetypen, Quelle: Eigene Darstellung.	47
Tabelle 14: Attribute des Klassendiagramms Device, Quelle: Eigene Darstellung.	48
Tabelle 15: Attribute des Klassendiagramms CVS, Quelle: Eigene Darstellung.....	51
Tabelle 16: Attribute des Klassendiagramms EMSDIL, Quelle: Eigene Darstellung.....	52
Tabelle 17: Attribute der Klasse PCS, Quelle: Eigene Darstellung.	53
Tabelle 18: Attribute der Klasse PFS, Quelle: Eigene Darstellung.....	54
Tabelle 19: Attribute der Klasse Analyser, Quelle: Eigene Darstellung.....	57
Tabelle 20: Attribute der Klasse Analyser_Mearange, Quelle: Eigene Darstellung.	57
Tabelle 21: Attribute der Klasse CVS_BagCfg, Quelle: Eigene Darstellung.....	57
Tabelle 22: Attribute der Klasse CVS_MainFlowStage, Quelle: Eigene Darstellung.	57
Tabelle 23: weitere Attribute der Klasse CVS_SampleFlowStages, Quelle: Eigene Darstellung.	58
Tabelle 24: Attribute der Klasse PFS, Quelle: Eigene Darstellung.....	58
Tabelle 25: Attribute der Klasse PCS, Quelle: Eigene Darstellung.	58
Tabelle 26: Attribute der Klasse EPA, Quelle Eigene Darstellung.	58

Tabelle 27: Attribute der Klasse FTIR_Analyser, Quelle: Eigene Darstellung.	59
Tabelle 28: Attribute der Klasse StartMode, Quelle: Eigene Darstellung	61
Tabelle 29: Attribute der Klasse StopMode, Quelle: Eigene Darstellung	61
Tabelle 30: Attribute der Klasse Phase, Quelle: Eigene Darstellung.....	61
Tabelle 31: Attribute der Klasse Soakphase, Quelle: Eigene Darstellung.	62
Tabelle 32: Attribute der Klasse Law, Quelle: Eigene Darstellung.	63
Tabelle 33: Attribute der Klasse Testcondition, Quelle: Eigene Darstellung.....	63
Tabelle 34: Attribute der Klasse Fuel, Quelle: Eigene Darstellung.....	64
Tabelle 35: Attribute der Klasse Multicycle_Fixed, Quelle: Eigene Darstellung.....	64
Tabelle 36: Attribute der Klasse Multicycle_Variable, Quelle: Eigene Darstellung.	64
Tabelle 37: Erweiterte Darstellung der Beutelsteuerung um Linien, Quelle: Eigene Darstellung.	71

ABKÜRZUNGSVERZEICHNIS

ABNT	Associação Brasileira de Normas Técnicas
CARB	California Air Resource Board
CFV	Critical Flow Venturi
CoC	Certificate of Conformity
CoP	Conformity of Production
EPA	Environmental Protection Agency
EUDC	Extra Urban Driving Cycle
FMC	Fundamental Modeling Concepts
HDPV	Medium Duty Passenger Vehicles
KLEV	Korean Low Emission Vehicles
LDT	Light Duty Trucks
LDV	Light Duty Vehicles
LEV	Low Emission Vehicles
MOE	Ministry of Environment
NEFZ	Neuer europäischer Fahrzyklus
PKW	Personenkraftwagen
RDE	Real Driving Emissions
SULEV	Super Ultra Low Emission Vehicles
SYSMML	Systems Modeling Language
TRIAS	Test Requirements and Instructions for Automobile Standards
UDC	Urban Driving Cycle
ULEV	Ultra Low Emission Vehicles
UML	Unified Modeling Language
UNECE	United Nations Economic Commission for Europe
WLTC	Worldwide harmonized Light-duty vehicles Test cycle
WLTP	Worldwide harmonized Light-duty vehicles Test Procedure
ZEV	Zero Emission Vehicles