

Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Bachelorthesis

Jan Scholz

Ansteuerung von EMV-Prüfgeneratoren und  
automatische Erstellung von Prüfberichten

Jan Scholz

Ansteuerung von EMV-Prüfgeneratoren und  
automatische Erstellung von Prüfberichten

Bachelorthesis eingereicht im Rahmen der Bachelorprüfung  
im Studiengang Informations- und Elektrotechnik  
am Department Informations- und Elektrotechnik  
der Fakultät Technik und Informatik  
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuer Prüfer : Prof. Dr. rer. nat. Henning Dierks  
Zweitgutachter : Dipl.Ing. Julia Voigt

Abgegeben am 29. August 2015

## **Jan Scholz**

### **Thema der Bachelorthesis**

Ansteuerung von EMV-Prüfgeneratoren und automatische Erstellung von Prüfberichten

### **Stichworte**

Java, iText, LabView, Pdf, Steuerung, Elektromagnetische-Verträglichkeit, DESY, Test, Prüfbericht, Objekt-orientierte Programmierung

### **Kurzzusammenfassung**

In dieser Arbeit wird der Entwicklungsprozess einer Software zur vereinfachten Prüfung von Prototypen in Hinblick auf Elektromagnetische Verträglichkeit beschrieben. Die Kernfunktionen der Software sind Ansteuerung der Störgeneratoren, Erstellung von benutzerdefinierten Tests, dynamische Erstellung eines Prüfberichts auf Basis der abgeschlossenen Tests und Laden und Speichern von Test-Datensätzen

## **Jan Scholz**

### **Title of the paper**

Software driven Operation of EMC test generators and automatic generation of test reports

### **Keywords**

Java, iText, LabView, Pdf, control, electromagnetic compatibility, DESY, test, report, object - oriented programming

### **Abstract**

In this study, the development process of a software for simplified testing of prototypes in terms of electromagnetic compatibility is described. The core functions of the software are controlling the disturbance generators , creating custom tests, dynamic creation of an audit report based on the completed tests and loading and saving of test records

## Danksagung

Ich möchte meinem betreuenden Professor Herrn Dr. Dierks danken, dass er sich die Zeit genommen hat, mich im Erstellen dieser Arbeit zu unterstützen.

Vielen Dank an Herrn Zeides, der es mir ermöglicht hat, diese Arbeit bei DESY zu schreiben. Besonderen Dank gilt außerdem Frau Voigt, meiner Betreuerin, die mit ihrer konstruktiven Kritik, maßgeblich zu meiner Bachelor-Arbeit beigetragen hat. Ich möchte außerdem Herrn Eis für die großartige fachliche Unterstützung im Bereich EMV und die sehr kameradschaftliche Zusammenarbeit danken. Jeder Mitarbeiter der Abteilung Prüffeld hat einen Anteil an der erfolgreichen Erstellung dieser Arbeit mit ihren Antworten auf meine Fragen, mit dem Testen der Software oder auch nur mit einem netten Gespräch unter Kollegen.

Zu guter Letzt möchte ich ganz besonders meiner Lebensgefährtin Verena danken, die mir während der letzten Monate den Rücken frei gehalten hat und sich liebevoll um unseren Sohn Enno kümmerte, wenn für die Bachelor-Arbeit gearbeitet habe.

# Inhaltsverzeichnis

<b>Tabellenverzeichnis</b>	<b>7</b>
<b>Abbildungsverzeichnis</b>	<b>8</b>
<b>1 Einführung</b>	<b>10</b>
<b>2 Hardware</b>	<b>12</b>
2.1 Coupling/Decoupling Network (CDN) . . . . .	12
2.2 Combi Coupling/Decoupling CFG-4028 . . . . .	12
2.3 Kombi-Generator CEG-4500 . . . . .	13
2.4 CWS 500N1 . . . . .	13
2.5 Zweileiter-V-Netznachbildung . . . . .	14
2.6 Messtechnik von National Instruments . . . . .	14
<b>3 Software</b>	<b>15</b>
3.1 LabVIEW . . . . .	15
3.1.1 Entwicklungsumgebung . . . . .	15
3.1.2 Grafische Datenflussprogrammierung . . . . .	16
3.1.3 Features . . . . .	16
3.1.4 Programmbeispiel . . . . .	17
3.2 Virtual Instrument Software Architecture (VISA) . . . . .	20
3.3 iText . . . . .	21
<b>4 Elektromagnetische Verträglichkeit</b>	<b>22</b>
4.1 EU-Richtlinie und CE-Kennzeichnung . . . . .	22
4.2 Normen . . . . .	23
4.3 Fachgrundnormen . . . . .	25
4.4 Grundnormen Störaussendung . . . . .	25
4.5 Grundnormen für Störfestigkeit . . . . .	27
4.5.1 EN 61000-4-2 . . . . .	27
4.5.2 EN 61000-4-3 . . . . .	27
4.5.3 EN 61000-4-4 . . . . .	29
4.5.4 EN 61000-4-5 . . . . .	32

4.5.5 EN 61000-4-6 . . . . .	33
4.5.6 EN 61000-4-8 . . . . .	34
4.5.7 EN 61000-4-11 . . . . .	35
<b>5 Analyse</b>	<b>37</b>
5.1 Stakeholder und Ziele . . . . .	37
5.2 Use-Cases . . . . .	42
5.2.1 Report verwalten . . . . .	42
5.2.2 Test durchführen . . . . .	46
<b>6 Entwurf</b>	<b>50</b>
6.1 Datenstruktur . . . . .	50
6.2 Hauptprogramm . . . . .	52
6.2.1 Test-Unterprogramm . . . . .	53
6.2.2 Device-Interface . . . . .	58
6.2.3 Test-Interface . . . . .	61
6.3 PDF-Prüfbericht . . . . .	63
6.3.1 XML-Parsing . . . . .	64
6.3.2 Pdfbuilder . . . . .	65
6.4 Schaltung für die Klimadatenerfassung . . . . .	65
6.4.1 NTC Thermistor TTC05104JSY . . . . .	66
6.4.2 HCZ-J3A HUMIDITY SENSOR . . . . .	69
<b>7 Realisierung</b>	<b>71</b>
7.1 Anwendereingaben . . . . .	71
7.2 CEG-4500 (Hofbauer Electronic) . . . . .	73
7.3 emtest CWS 500N1 . . . . .	74
7.4 Messdatenerfassung . . . . .	75
7.4.1 Raumtemperatur . . . . .	75
7.4.2 Luftfeuchtigkeit . . . . .	76
7.5 Lizenzierung . . . . .	78
7.6 Beta-Testing . . . . .	78
<b>8 Fazit</b>	<b>80</b>
<b>Literaturverzeichnis</b>	<b>82</b>
<b>Index</b>	<b>85</b>

# **Tabellenverzeichnis**

5.1 Stakeholder-Analyse: 1.Ziel . . . . .	39
5.2 Stakeholder-Analyse: 2.Ziel . . . . .	39
5.3 Stakeholder-Analyse: 3.Ziel . . . . .	40
5.4 Stakeholder-Analyse: 4.Ziel . . . . .	40
5.5 Stakeholder-Analyse: 5.Ziel . . . . .	41
5.6 Stakeholder-Analyse: 6.Ziel . . . . .	41
6.1 Widerstandstabelle relative Feuchtigkeit / Temperatur in kΩ [HCZ,S.6] . . . . .	70

# Abbildungsverzeichnis

2.1 CEG-4500 . . . . .	13
2.2 Geräteinformationen . . . . .	13
3.1 Blockdiagramm mit einer einfachen Rechnung . . . . .	16
3.2 LabVIEW's Datentypen . . . . .	17
3.3 Beispiel: VI-Symbol und Interface . . . . .	18
3.4 Beispiel: Frontpanel . . . . .	18
3.5 Beispiel: Fibonacci Sequenz Blockdiagramm 1 . . . . .	19
3.6 Beispiel: Fibonacci Sequenz Blockdiagramm 2 . . . . .	19
4.1 EN 61000-4-3: Versuchsaufbau für feldgebundene Störungen [EMV43 (2010),S.26] . . . . .	29
4.2 EN 61000-4-4: Burst Störpuls [EMV43 (2010),S.21] . . . . .	30
4.3 EN 61000-4-4: Störsignal mit 2kV 5kHz 15ms, verschiedene Auflösungen . . . . .	31
4.4 EN 61000-4-5: gängige Amplituden-/Phasen-Kombinationen . . . . .	32
4.5 EN 61000-4-5: Surge Störpuls [EMV45 (2015),S.15] . . . . .	33
4.6 EN 61000-4-5: Surge Prüfaufbau [EMV45 (2015),S.21] . . . . .	33
4.8 EN 61000-4-6: Prüfaufbau [EMV46 (2014),S.25] . . . . .	34
4.9 EN 61000-4-8: Prüfprozess [EMV48 (2010),S.17] . . . . .	35
4.7 EN 61000-4-6: 80% AM HF-Signal [EMV46 (2014),S.26] . . . . .	35
4.10 Spannungseinbruch . . . . .	36
5.1 Use-Case Diagramm "Report verwalten" . . . . .	43
5.2 Aktivitätsdiagramm "Report verwalten" . . . . .	43
5.3 Use-Case Diagramm "Test durchführen" . . . . .	46
5.4 Aktivitätsdiagramm "Test durchführen" . . . . .	47
6.1 Klassendiagramm der Datenstruktur des LabVIEW-Programms . . . . .	51
6.2 Zustandsdiagramm für das Hauptprogramm . . . . .	53
6.3 Hauptfenster I . . . . .	54
6.4 Hauptfenster II . . . . .	55
6.5 Test-Unterprogramm Statechart . . . . .	57
6.6 Device-Interface Übersicht . . . . .	58

6.7 checksum . . . . .	58
6.8 init Tests . . . . .	58
6.9 Choose standard Test UI . . . . .	59
6.10 open connection . . . . .	59
6.11 Identitiy Check . . . . .	59
6.12 read . . . . .	59
6.13 write . . . . .	60
6.14 Error Catcher . . . . .	60
6.15 close connection . . . . .	60
6.16 Test-Interface . . . . .	61
6.17 manual test UI . . . . .	62
6.18 create Behaviour . . . . .	62
6.19 Check Parameters . . . . .	62
6.20 create Timings . . . . .	62
6.21 create Commandlist . . . . .	63
6.22 eval Command . . . . .	63
6.23 respond to response . . . . .	63
6.24 Klassendiagramm des Java-Programms . . . . .	64
6.25 Schaltplan der Baugruppe . . . . .	66
7.1 manual test UI für den Burst-Test . . . . .	72
7.2 Verbindung PC - CEG-4500 . . . . .	73

# 1 Einführung

Im Lauf der letzten 100 Jahren ist die Elektromagnetische Verträglichkeit (EMV) ein Themengebiet geworden, mit dem sich nicht nur Experten auf dem Gebiet der Kommunikationstechnik auseinandersetzen, sondern jeder Hersteller und Entwickler von Bauteilen und ganzen Baugruppen. Mit immer größerer Anzahl von elektronischen Geräten in Haushalten, steigenden Taktraten und anwachsender Kommunikation per Funk in einem immer breiterem Spektrum<sup>1</sup>, ist es wichtiger denn je, die Störfestigkeit von Geräten zu gewährleisten und eine niedrige Störaussendung zu garantieren.

Beim Deutschen Elektronen-Synchrotron (DESY) werden viele eigene elektronische Baugruppen und Geräte entwickelt und gefertigt. Dies ist notwendig durch die speziellen Anforderungen, die die Forschung an das Equipment stellt. Viele Baugruppen sind sehr sensibel und haben hohe Anforderungen, da Messungen möglichst genau sein sollen und Störungen mit erheblichem zeitlichem und finanziellem Aufwand verbunden sein können. So werden also EMV-Vorprüfungen intern bei DESY durchgeführt, um diesen Anforderungen Rechnung zu tragen. Die Prototypen dieser Baugruppen und Geräte werden getestet und können bereits während der Entwicklung so angepasst werden, dass sich die gewünschte Störfestigkeit und Störaussendung erreichen lässt. Die EMV-Prüfung ist eine recht langwierige Arbeit, die eine Vielzahl von Versuchsaufbauten und Einstellungen an Störgeneratoren erfordern. So kann eine EMV-Prüfung, wenn sie im vollen Umfang angewendet werden soll, durchaus einen bis zwei Arbeitstage beanspruchen.

Im Rahmen dieser Arbeit wird eine Software entwickelt, die die Ansteuerung der Störgeneratoren übernimmt und dem Anwender die Möglichkeit gibt, die Prüfung zu dokumentieren. Im Anschluss wird ein Prüfbericht im PDF-Format ausgegeben.

Die Entwicklung dieser Software wird mit der Entwicklungsumgebung LabVIEW durchgeführt, welche einer grafischen Programmiersprache unterliegt. Der Prüfbericht wird auf Grundlage der gesammelten Daten erstellt und baut auf einem/r Framework/Bibliothek zur programmatischen Erstellung von PDF-Dateien auf. Die Auswahl wird während der Arbeit geschehen. Bei der Auswahl ist darauf zu achten, dass - falls die Umsetzung nicht in LabVIEW stattfinden kann - die Daten zwischen den beiden Applikationen übertragen werden müssen. Wichtig ist ein detailliertes Wissen über die relevanten Prüfnormen und

---

<sup>1</sup>jbw (2015)

über die Ansteuerung der verschiedenen Geräte, die während der Prüfung genutzt werden. Es ist sinnvoll, das Programm so zu gestalten, dass weitere Tests und/oder Geräte einfach hinzugefügt werden können.

Die Arbeit ist in acht Kapitel aufgeteilt. Nach dieser Einführung beginnen die Grundlagen, es werden die Geräte vorgestellt, die zusammen mit dem entwickelten Programm genutzt wurden. Dann wird die Software, die maßgeblich benutzt wurde, um das Programm zu schreiben, dargestellt. Das darauf folgende Kapitel behandelt die Elektromagnetische Verträglichkeit (EMV), hier werden die Normen vorgestellt, welche die Grundlage für die Prüfung von Geräten bilden. Die einzelnen relevanten Normen werden hier kurz vorgestellt. Sie beinhalten die Prüfprozesse und Messaufbauten.

Nachfolgend wird im Theorie-Teil angefangen mit der Entwicklung der Software. Zuerst wird eine Analyse der Stakeholder und der Use-Cases gemacht. Im Anschluss wird der Entwurf für das Hauptprogramm und das Programm zur Erstellung von Prüfberichten im Detail, sowie der Entwurf der Schaltung für die Messung der Klimadaten, geschildert.

Die Umsetzung wird dann im Kapitel Realisierung beschrieben, dort wird der Entwicklungsprozess anhand von wichtigen Meilensteinen erläutert. Es werden außerdem einzelne Problemstellungen ausgeführt, die während der Entwicklung aufraten und wie diese gelöst wurden.

Im letzten Kapitel werden die Ergebnisse nochmals im Fazit zusammengefasst und die neuen Erkenntnisse, besonders in Hinsicht auf Software Entwicklung, erläutert.

## **2 Hardware**

### **2.1 Coupling/Decoupling Network (CDN)**

In der EMV-Technik werden sog. Koppel-/Entkoppel-Netzwerke genutzt, um schmalbandige hochfrequente Störsignale auf den Prüfling einzukoppeln. Es wird außerdem gewährleistet, dass hierbei die Störungen nicht in das Versorgungsnetz mit eingespeist wird. Die CDNs sind auf Grund von unterschiedlichen Messanforderungen in verschiedenen Ausführungen erhältlich. So gibt es z. B. CDNs für das Einkoppeln von Störungen auf Versorgungsleitungen, Telekommunikationsleitungen und unsymmetrischen Leitungen.

### **2.2 Combi Coupling/Decoupling CFG-4028 (Peter Hofbauer Electronic GmbH)**

Dieses Gerät ist ein Koppelnetzwerk zur Ein- bzw. Auskopplung in/aus dem Netz. Das Gerät ist für Prototypen mit Drei-Phasen-Spannungsversorgung obligatorisch, da der Kombi-Generator CEG-4500 nur einen Ausgang für einphasige Spannungsversorgungen hat. Der Vorteil gegenüber CDNs ist, dass an dem Gerät gewählt werden kann, auf welche Signalleitung die Störung eingeprägt werden soll. Bei CDNs kann dies nur mit selbstgebauten Adapters realisiert werden. Das Gerät verfügt auch über eine 9-polige SUB-D-Buchse, die mit RS-485 angesprochen werden kann.

## 2.3 Kombi-Generator CEG-4500 (Peter Hofbauer Electronic GmbH)

Das Gerät hat drei verschiedene Generatoren inne. Mit ihnen können Tests nach den Normen 61000-4-4 (Burst, siehe Kapitel 4.5.3), 61000-4-5 (Surge, siehe Kapitel 4.5.4) und 61000-4-11 (Spannungseinbruch, siehe Kapitel 4.5.7) durchgeführt werden. An der Rückseite des Geräts befindet sich eine 9-polige Sub-D-Buchse, die entweder mit RS-485 oder RS-232 verbunden werden kann. Das Gerät kann direkt oder über CFG-4028 mit dem EUT verbunden werden.



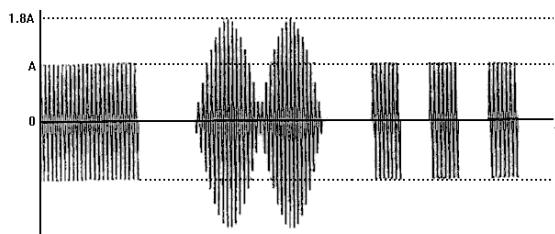
Abbildung 2.1: CEG-4500

## 2.4 CWS 500N1 (EM TEST (Switzerland) GmbH)

Das Prüfgerät CWS 500N1 ist ein kompakter Generator, der Tests für leitungsgebundene Störfestigkeit, z. B. nach IEC 61000-4-6 (siehe Kapitel 4.5.5), durchführen kann. Die Störsignale können in einem Bereich von 100kHz bis 300MHz generiert und mit 2Hz, 400Hz und 1kHz mit 80% amplitudemoduliert werden, diese Signale sind kalibriert. Der Hersteller garantiert nicht für die Integrität anderer händisch eingestellter Modulationsfrequenzen. Kalibriert sind außerdem Phasenmodulationen mit 1Hz und 1kHz. Es können Signale mit einer Amplitude von 0,1V bis 60V ausgegeben werden. Es sind eine USB-Schnittstelle und ein GPIB-Port vorhanden. Es sind 7 Kalibrierungsspeicher vorhanden, auf denen unterschiedliche Kalibrierungen für die notwendigen CDNs abgespeichert werden können. Die Bedienung des Gerätes ist manuell mit Hilfe des LED-Panels möglich oder kann mittels der mitgelieferten Software icd.control realisiert werden.



(a) Gerätefront emtest CWS 500N1



(b) Modulation, keine M., 80% AM, Pulsmoduliert

Abbildung 2.2: Geräteinformationen

werden. Das Gerät verfügt über einen Stromsensor-Eingang, an dem eine Stromzange angeschlossen werden kann. Auf der Rückseite sind außerdem zwei BNC-Buchsen, die, wenn sie auf Masse gezogen werden, den Prüfprozess anhalten oder im anderen Fall den Prüfprozess abbrechen.

## 2.5 Zweileiter-V-Netznachbildung Typ NNB-2/16Z (Telemeter Electronic)

Dieses Gerät hat die Hauptaufgabe, die Netzimpedanz<sup>1</sup> nachzubilden und eine Rückkopplung von Funkstörspannungen in das Netz zu unterbinden. Es arbeitet in einem Frequenzbereich von 9kHz bis 30MHz.

## 2.6 Messtechnik von National Instruments (NI)

Bereits vorhanden waren zwei Module (NI 9221, NI 9264) sowie das Anschlußmodul (NI cDAQ-9178). NI entwickelt zusammen mit seinen Partnern diese Module, um den unterschiedlichen Messanwendungen Rechnung zu tragen. Die Module von NI haben den großen Vorteil, dass sie sehr einfach in LabVIEW eingebunden werden können. Mit Hilfe der NI-DAQmx-Technologie können Signale auf einfache Weise generiert und gemessen werden. Das Modul NI 9221 ist ein Analogeingangsmodul mit 8 Kanälen,  $\pm 60$  V, 800 kS/s und 12 Bit Auflösung. Das Modul NI 9264 kann mit 16 Kanälen Signale im Bereich von  $\pm 10$  V mit 16 Bit Genauigkeit ausgeben.

---

<sup>1</sup>"Die Netzimpedanz ist die Impedanz, die der Strom, der vom Verbraucher aufgenommen wird, beim Energietransport im Niederspannungsnetz durchfließt".(Zach (1990), S. 471). Die Netzimpedanz für hochfrequente Signale ist zunächst für jedes Netz unbekannt. Um wiederholbare und belastbare Ergebnisse zu erhalten, kann dieses Gerät zwei weltweit genormte Netzimpedanzen annehmen.

# 3 Software

## 3.1 LabVIEW

LabVIEW ist der Pionier der strukturierten grafischen Datenflussprogrammierung. Die Entwicklungsumgebung wurde im Jahr 1986 von Jeff Kodosky, einem der Mitbegründer von NI, entwickelt. Die Programmiersprache, in der mit LabVIEW programmiert wird, wird G genannt. Die initiale Idee für LabVIEW war, eine Entwicklungsumgebung zu erschaffen, die es ermöglicht, in kurzer Zeit automatisierte Messanwendungen zu entwickeln. Das Prinzip der grafischen Datenflussprogrammierung ermöglicht es, dem Benutzer Prozesse ohne Weiteres parallel laufen zu lassen und gleichzeitig den Überblick zu behalten. LabVIEW profitiert von der Entwicklung von FPGAs und DSPs, die im Bereich der parallelen Datenverarbeitung neue Maßstäbe setzen. Mit einer prozeduralen Programmiersprache ist es sehr aufwendig, eine massiv parallel laufende Anwendung zu entwickeln. LabVIEW ist durch das Prinzip der Datenflussprogrammierung hierfür wesentlich besser geeignet, da das Paradigma der Datenflussprogrammierung eine parallele Verarbeitung bereits vorsieht.

LabVIEW wurde gewählt, da bereits Komponenten von der Firma NI vorhanden waren und eine Ansprache per LabVIEW sehr leicht realisiert werden kann. Außerdem hat sich LabVIEW als de facto-Standard im Bereich der automatisierten Mess- und Steuerungstechnik etabliert und unzählige Bibliotheken sind bereits im Auslieferungszustand enthalten, die die Ansprache einer Vielzahl von Messgeräten wie beispielsweise Oszilloskopen, Signalgeneratoren oder Spektrumanalysern ermöglichen. Eine große Community und auch Hersteller entwickeln ständig weitere Bibliotheken, so dass bereits ca. 6000 verschiedene Geräte leicht mittels LabVIEW betrieben werden können.

### 3.1.1 Entwicklungsumgebung

Die Entwicklungsumgebung besteht aus zwei Ebenen: dem Frontpanel und dem Blockdiagramm. Das Frontpanel ist die Darstellung aller Bedien- bzw. Anzeigeelemente. Es kann als User-Interface genutzt werden oder lediglich als Definition des Interfaces eines kleineren Programmbausteins. Das Blockdiagramm repräsentiert den eigentlichen Programmcode und kann aus einer Vielzahl von Datenströmen bestehen, die auf kleinere Unterprogramme verteilt werden.

### 3.1.2 Grafische Datenflussprogrammierung

Die grafische Datenflussprogrammierung unterscheidet sich zum einen sehr durch ihre Präsentation, zum anderen durch das zugrunde liegende Konzept der strukturierten Verarbeitung der Befehle. Ein kontinuierlicher Datenstrom kann an den Gates sogenannter VIs (Virtual Instruments) angebunden und dort weiter verarbeitet werden. Der Datenstrom kann dabei beliebig oft verzweigt werden. Wie bereits erwähnt wurde, ist es so auf einfache Weise möglich, parallele Prozesse zu etablieren und zu koordinieren. Die grafische Präsentation verdeutlicht übersichtlich, welche Prozesse zu welchem Zeitpunkt ablaufen. Ein VI ist vergleichbar mit einer Funktion oder Methode in anderen Programmierparadigmen. Auf der linken Seite des VI's werden gewöhnlich die Eingabewerte (Vgl. Abbildung 3.3) angelegt und auf der rechten Seite werden/wird die/der Ausgabewert/e abgetragen. Ein VI wird erst dann mit der Ausführung anfangen, wenn an jedem Eingang ein Signal anliegt.

In Abbildung 3.1 ist die einfache Rechenaufgabe  $(1 + 2) - 3 = 0$  dargestellt. Anhand dieser Aufgabe lässt sich der Charakter der grafischen Datenflussprogrammierung verdeutlichen. Es sind zwei VIs zu sehen: Eines tätigt eine Addition, ein anderes eine Subtraktion, weiter sind drei Konstanten sowie ein Anzeigeelement vorhanden. Zuerst werden die Konstanten 1 und 2 mit Additions-VI verarbeitet. Nachdem das Ergebnis berechnet wurde, kann der zweite Rechenschritt vollzogen werden. Von dem Ergebnis wird im Anschluss die Konstante 3 abgezogen, das Endergebnis wird in einem Anzeigeelement ausgegeben.

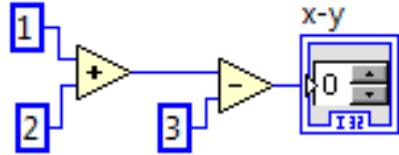


Abbildung 3.1: Blockdiagramm mit einer einfachen Rechnung

### 3.1.3 Features

LabVIEW verfügt über alle Werkzeuge, die für strukturierte Programmierung notwendig sind. Zu ihnen gehören For- und While-Schleifen sowie Case-Strukturen. Die verfügbaren Datentypen sind in Abbildung 3.2 aufgeführt. Hervorzuheben sind **Cluster**, **Refnum** und **LVClass**, die im Folgenden näher erklärt werden sollen:

**Cluster** sind Bündelungen von mehreren Datentypen oder Datenstrukturen. Sie sind sehr hilfreich, um bei der Übertragung vieler verschiedener Signale die Übersicht zu behalten. Außerdem sind sie besonders nützlich, um in einem Zustandsautomaten Daten in Zuständen zu bearbeiten und danach weiterzureichen. Besondere Erwähnung muss der Fehler-Cluster bekommen, da er ein sehr wichtiges Werkzeug ist, um einerseits den Programmablauf zu

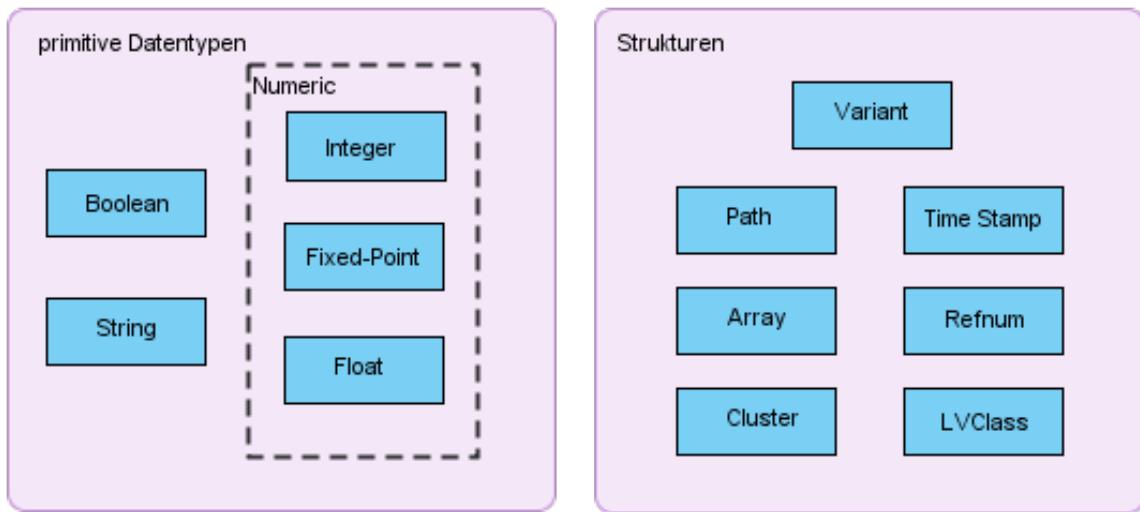


Abbildung 3.2: LabVIEW's Datentypen

steuern und andererseits Fehlerhandling zu betreiben. Viele VIs haben einen Fehler-Cluster-Eingang und -Ausgang. So ist es möglich, mittels des Fehler-Clusters die Ausführungsreihenfolge der VIs zu bestimmen, indem die Verbindungen entsprechend angelegt werden. Der Fehler-Cluster enthält drei Informationen: den Fehler-Code, eine Nachricht und eine boolsche Variable zur einfachen Abfrage, ob ein Fehler aufgetreten ist.

Der Datentyp **Refnum** bezeichnet die Referenz auf ein Objekt. Mit dieser Referenz können mittels eines Eigenschaftsknotens oder eines Methodenknotens diese Objekte verändert werden oder auch der Status eines Parameters erfragt werden.

Objekte sind alle Anzeigeelemente und Bedienelemente sowie LabVIEW Klassen. Diese bilden den Datentypen **LVClass**. Mit Hilfe von Klassen ist in LabVIEW eine objektorientierte Datenverarbeitung möglich. Der Datentyp LVClass beschreibt eine Klasse, in der einzelne Parameter in Form von verschiedenen Datentypen abgespeichert werden können. Weiter kann eine Klasse auch eigene VIs enthalten, hierbei können VIs und Parameter von einer anderen Klasse vererbt werden. VIs können überschrieben und eine andere Implementierung gewählt werden. Dies ermöglicht Polymorphismus und trägt dazu bei, Code leichter instandzuhalten.

### 3.1.4 Programmbeispiel

Um einen Einblick über das Arbeiten mit LabVIEW zu gewinnen, wird an einem Beispiel verdeutlicht, wie Code mit LabVIEW erstellt wird. Das Beispiel berechnet die Fibonacci-Sequenz bis zur 100. Zahl.

In Abbildung 3.4 ist das Frontpanel dargestellt. Wie bereits erwähnt, sind alle Anzeige- und

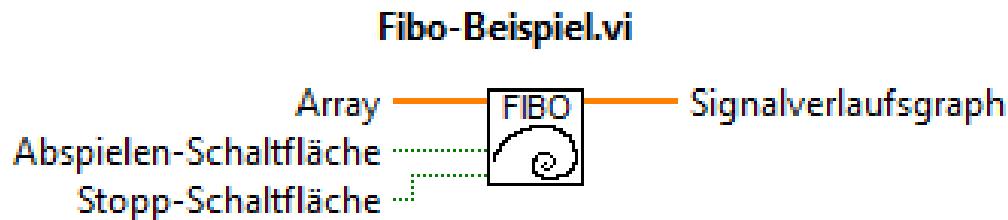


Abbildung 3.3: Beispiel: VI-Symbol und Interface

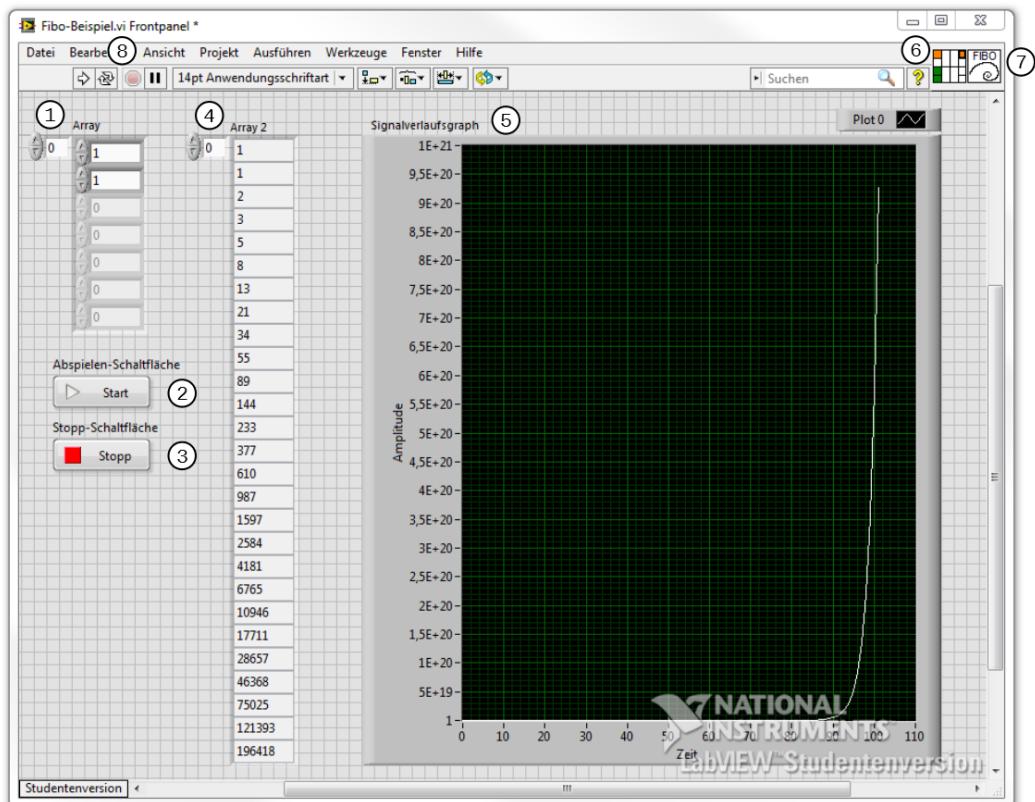


Abbildung 3.4: Beispiel: Frontpanel

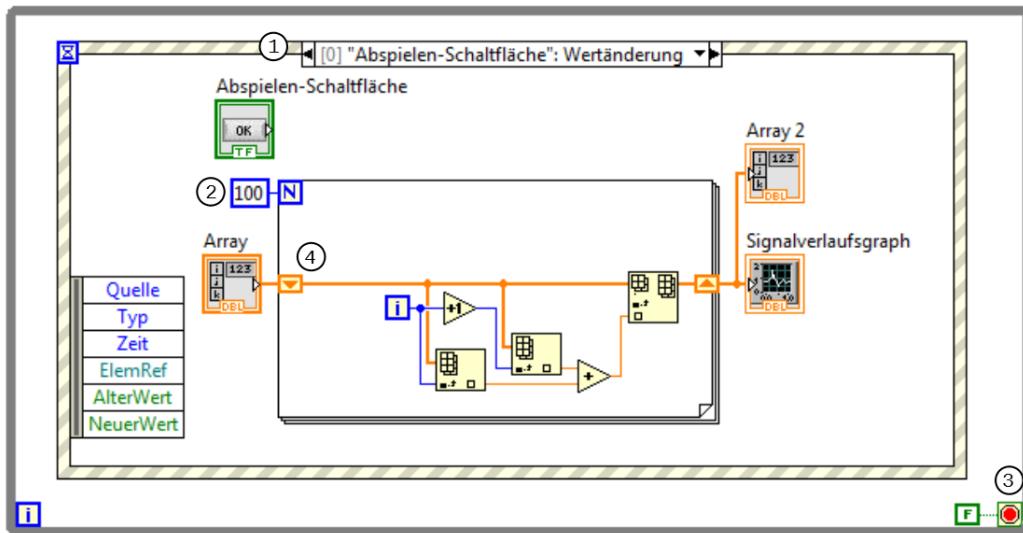


Abbildung 3.5: Beispiel: Fibonacci Sequenz Blockdiagramm 1

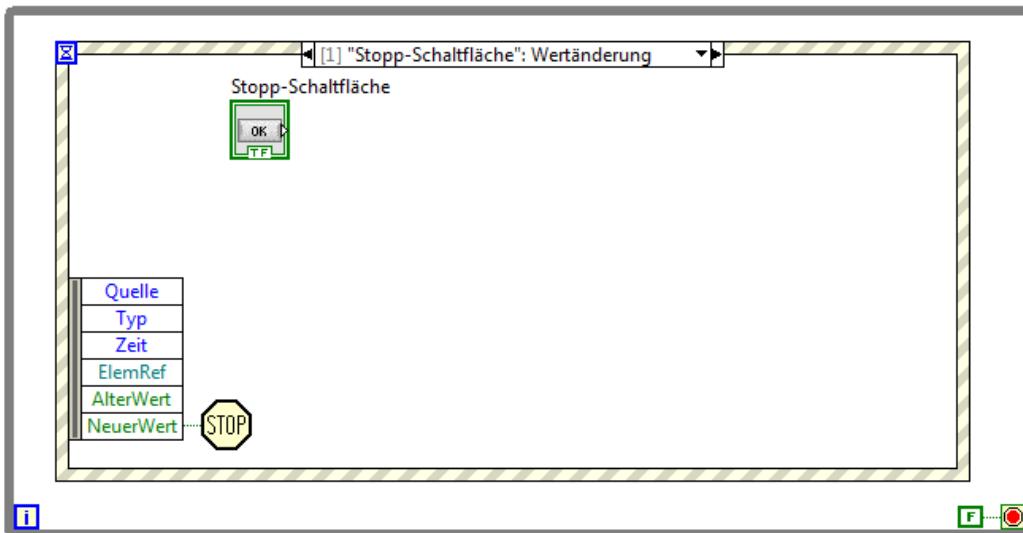


Abbildung 3.6: Beispiel: Fibonacci Sequenz Blockdiagramm 2

Bedienelemente dort zu finden. Unter Punkt 1 im Bild ist das Bedienelemente-Array zu sehen. Hiermit werden die beiden ersten Elemente der Folge an das VI übergeben. Punkt 2 und 3 sind Buttons, die als boolesche Variablen fungieren, um das VI zu starten bzw. zu beenden. Dies ist zwar nicht unbedingt notwendig, da ein VI automatisch anfängt, wenn alle Eingangssignale anliegen und auch dann aufhört, wenn das letzte SubVI abgearbeitet wurde. Allerdings wurden die Buttons hier implementiert, um die Event-Struktur zu veranschaulichen. An Punkt 4 und 5 befinden sich zwei Anzeigeelemente, die die Ausgabe des Algorithmus abbilden. Punkt 6 zeigt die Anschlussmöglichkeiten an dem VI, in Punkt 7 wird das Symbol des VIs dargestellt. Es kann sinnvoll sein, Highlevel-VIs mit benutzerdefinierten Symbolen zu versehen, um die Lesbarkeit des Programmcodes zu verbessern.

In Abbildung 3.3 ist dargestellt, wie das VI mit seinen Anschlüssen aussieht. Das VI ist in diesem Zustand noch eine leere Hülle und stellt lediglich ein Interface dar.

Die übergeordnete Struktur wird von einem Blockdiagramm gebildet. In Abbildung 3.5 ist um alle Bausteine herum eine While-Schleife zu sehen, in ihr befindet sich an Punkt 1 eine Event-Struktur. Diese Event-Struktur ist besonders hilfreich bei Benutzerinteraktionen mit einer Anwendung. Es können unterschiedliche Ereignisse passieren, die dann in der Event-Struktur einen Code ausführen lassen. Diese Event-Struktur hat zwei Trigger: den Button "Abspielen-Schaltfläche" und den Button "Stopp-Schaltfläche" (zu sehen in Abbildung 3.6). Sollte sich nun einer dieser beiden Buttons im Wert verändern, wird der Code innerhalb der Event-Struktur ausgeführt. Im Fall des Start-Buttons ist das Bedienelement "Array" dargestellt, welches in eine For-Schleife mündet.

Bei Punkt 2 wurde mittels einer Konstante die Anzahl der Schleifendurchläufe festgelegt. Punkt 4 zeigt ein sog. Schieberegister (nach unten zeigendes Dreieck), welches mit dem Array initialisiert wird und mit jeder Iteration weiter verändert wird. Innerhalb der Schleife wird auf das Array zugegriffen. Der aktuelle Laufzeitindex kann mittels des blauen i's abgegriffen werden. Mittels zweier VIs können  $\text{Array}[i]$  und  $\text{Array}[i+1]$  addiert und im Anschluss an das Array angehängt werden. Sobald die Schleife abgearbeitet ist, wird das Ergebnis an "Array 2" und den Graphen weitergeleitet. Die While-Schleife läuft so lange, bis am Eingang des Symbols bei Punkt 3 ein boolesches true anliegt. Da dort eine Konstante anliegt, passiert das nie - das Programm muss mit dem Stopp-Button beendet werden.

## 3.2 Virtual Instrument Software Architecture (VISA)

VISA ist ein Standard für die Konfiguration, Programmierung und Fehlersuche für Messsysteme, die GPIB, VXI, PXI, Seriell, Ethernet und/oder USB-Schnittstellen nutzen<sup>1</sup>. VISA bietet die Programmierschnittstelle zwischen der Hardware und LabVIEW. Es gibt VIs, die von vielen Schnittstellen genutzt werden können. So lassen sich mit dem VI "VISA open" dank des

---

<sup>1</sup>Vgl.[NI](#) (2014)

VISA-Interfaces alle Schnittstellen ansprechen und eine Verbindung aufbauen. Der VISA Standard ist eine große Hilfe bei der Erstellung von neuen Gerätetreibern, wenn noch keine für LabVIEW vorhanden sind.

### 3.3 iText

iText ist ein Java-Framework zur Generierung von Pdf- Html- oder RTF-Dateien auf Grundlage von unterschiedlichen Datensätzen. Es ist möglich Vorlagen auszufüllen oder sogar eine Pdf-Datei von Grund auf neu zu erstellen. Das Framework stellt hierfür einige High-Level Objekte zur Verfügung mit denen Pdf-Dokumente erzeugt werden können. Für speziellere Aufgaben kann auch auf Low-Level mittels der Klasse PdfContentByte direkt auf die Pdf-Struktur zugegriffen werden. Die Nutzung von iText bot sich an, um die dynamische Erstellung des Prüfberichts zu ermöglichen. Für iText sprach außerdem, dass es über eine AGPL verfügt, die für unsere Anwendung bedeutet, dass die Benutzung kostenfrei geschehen kann.

# 4 Elektromagnetische Verträglichkeit

Die Untersuchung der EMV eines Prüflings hat verschiedene Gründe. Zum einen ist zu überprüfen, in wieweit ein Gerät sich von außen stören lässt, zum anderen ist festzustellen, in wie weit das Gerät andere Geräte stören kann. Die Ursachen von Störaussendung und Störfestigkeit sind sehr vielfältig. So können z. B. einzelne Leiterbahnen als Antenne wirken (Signale abstrahlen oder empfangen), auf Baugruppen- oder Geräteebene kann elektromagnetische Strahlung durch ungenügende Schirmung austreten. Die Möglichkeiten sind mannigfaltig. Die Verringerung von Störungseinwirkung und Störungsabgabe ist von hoher Priorität für sicherheitsrelevante Technik wie beispielsweise Geräte, die einen Störfall registrieren müssen, um eine Notabschaltung zu veranlassen. Wenn kein Störfall vorlag, ist die Notabschaltung einer Anlage immer mit Kosten verbunden. Im schlimmsten Fall kann die Gesundheit von einzelnen Individuen aus Spiel gesetzt werden, wenn das Gerät einen Störfall nicht registriert. In diesem Kapitel wird ein Überblick über die verschiedenen Tests gegeben, die im Zuge der EMV-Prüfung auf ein Gerät zukommen. Am Beispiel der bei DESY entwickelten Geräte wird gezeigt, welche Normen anzuwenden sind und wie deren Hierarchie ist. Besonders wird auf die Normen eingegangen, die die Störaussendung behandeln, da ein Großteil dieser Prüfungen bei DESY vor Ort durchgeführt werden kann. Alle Prüfungen, die mit der Software möglich sind, fallen auch unter diese Normen.

## 4.1 EU-Richtlinie und CE-Kennzeichnung

Das Europaparlament hat in der Richtlinie 2014/30/EU<sup>1</sup> zur Angleichung der Rechtsvorschriften in der EU im Umgang mit elektromagnetischer Verträglichkeit festgehalten, welche grundlegenden Anforderungen ein Betriebsmittel<sup>2</sup> einhalten muss, das Störungen aussenden oder durch elektromagnetische Störungen beeinträchtigt werden kann.

Wenn das Betriebsmittel den Anforderungen, auferlegt durch die harmonisierte Norm<sup>3</sup>., entspricht, kann eine EU-Konformitätserklärung ausgestellt werden und das Betriebsmittel darf

---

<sup>1</sup>EU (2014)

<sup>2</sup>vormals Gerät, mittlerweile wird mit "Betriebsmittel" von "Geräte" oder "ortsfesten Anlagen" gesprochen.

<sup>3</sup>Harmonisierte Normen sind europäische Normen für Produkte, die den freien Verkehr von Waren und Dienstleistungen innerhalb des europäischen Binnenmarktes regeln. Alle harmonisierten Normen müssen in den einzelnen Ländern in nationale Normen umgesetzt werden [Wikipedia (2015a)]

eine CE-Kennzeichnung erhalten. Betriebsmittel, die keine EU-Konformitätserklärung innehaben, dürfen nicht in Verkehr<sup>4</sup> gebracht werden.

## 4.2 Normen

Es existiert eine Hierarchie der Normen, die sich wie folgt zusammensetzt:

- Übergeordnet sind die **Produktnormen**. Sollte ein Betriebsmittel von diesen Normen erfasst sein, sind diese Normen bindend und anzuwenden. Wenn in anderen zusätzlich zutreffenden Normen höhere Anforderungen gestellt sind, sind diese anzuwenden.
- Es folgen **Produktgruppennormen**, die weiter gefasst sind und geringere Anforderungen haben als Produktnormen.
- Darauf folgen die **Fachgrundnormen**, sie legen die Mindestanforderungen für die Gerätesicherheit fest. In ihnen sind üblicherweise die geforderten Bewertungskriterien für einzelne Tests festgelegt, dies kann allerdings auch in den Grundnormen weiter aufgeschlüsselt werden.
- Zum Schluss folgen die **Grundnormen**. Sie erweitern die Fachgrundnormen um zusätzliche Informationen und spezifizieren Messaufbauten und Prüfabläufe.

Die relevanten Normen für Baugruppen und Geräte, die bei DESY entworfen werden, sind:

- EN 61000-4-2
- EN 61000-4-3
- EN 61000-4-4
- EN 61000-4-5
- EN 61000-4-6
- EN 61000-4-8
- EN 61000-4-11
- EN 61000-3-2
- EN 61000-3-3
- EN 61000-3-11
- EN 61000-3-12

---

<sup>4</sup>die erstmalige Bereitstellung eines Geräts auf dem Unionsmarkt [EU (2014),S.6]

Sie behandeln die Prüf- und Messverfahren für EMV-Prüfungen. Bei DESY können nur leitungsgebundene Störungen eingekoppelt sowie aufgezeichnet werden. So sind die Normen 61000-4-3 und 61000-4-8 nicht anwendbar, da kein spezieller Raum für feldgebundene Störungen zur Verfügung steht. Um die EU-Konformitätserklärung für entwickelte Geräte bei DESY zu erhalten, wird das Gerät bei einem akkreditierten Labor final getestet, da normgerechte Aufbauten am DESY aus Platzgründen nicht immer möglich sind. Um Kosten und Wege zu vermeiden, wird durch die Prüfungen, die hier eher Vorprüfungscharakter haben, sichergestellt, dass alle leitungsgebundenen Prüfverfahren erfolgreich von dem Gerät bestanden wurden.

Wenn die Bewertung der Prüfergebnisse nicht auf Grundlage von vorhandenen Fachgrund-, Produkt- oder Produktfamiliennormen durchgeführt werden kann, sind in den Normen EN 61000-4-2 bis -11 vier standardisierte Bewertungskriterien vorgegeben, die unter Einbeziehung der Spezifikation des Herstellers<sup>5</sup> zur Erarbeitung eigener Bewertungskriterien dienen können. Die Kriterien sind wie folgt:

- a) bestimmungsgemäßes Betriebsverhalten innerhalb der vom Hersteller, Auftraggeber oder Käufer festgelegten Grenzen;
- b) zeitlich begrenzter Ausfall oder zeitlich begrenzte Minderung der Funktion oder des bestimmungsgemäßigen Betriebsverhaltens, der (die) nach dem Abklingen der Störgröße wieder abklingt; ab diesem Zeitpunkt hat das Gerät (die Einrichtung) sein (ihr) bestimmungsgemäßes Betriebsverhalten von selbst, ohne Eingriff der Bedienperson, wiedererlangt;
- c) zeitlich begrenzter Ausfall oder zeitlich begrenzte Minderung der Funktion oder des bestimmungsgemäßigen Betriebsverhaltens, für dessen (deren) Behebung ein Eingriff der Bedienperson erforderlich ist;
- d) Ausfall oder Minderung der Funktion oder des bestimmungsgemäßigen Betriebsverhaltens, die (das) nicht mehr wiederhergestellt werden kann, da das Gerät (Bauteil) oder Betriebsprogramm (Software) zerstört wurde und/oder Daten verloren gingen.

---

<sup>5</sup>Die Spezifikation kann Verhalten wie beispielsweise das autonome Ausschalten des Geräts, wenn das Gerät nachfolgend wieder den Betrieb aufnimmt, als nicht kritisch kategorisieren und so andere Bewertungskriterien zur Folge haben.

## 4.3 Fachgrundnormen

Diese Normen behandelt die Störfestigkeit (EN 61000-6-1,2)<sup>6</sup> und die Störaussendung (EN 61000-6-3,4)<sup>7</sup> von Betriebsmitteln. Die Normen legen fest, welche Bewertungskriterien für die Betriebsmittel mindestens gefordert sind. Hierbei gelten die EN 61000-6-1 und EN 61000-6-3, die für Wohnbereiche, Geschäfts- und Gewerbebereiche und Kleinbetriebe Anwendung finden. Die Normen EN 61000-6-2 und EN 61000-6-4 gelten für Industriebereiche. Die Anforderungen an die Geräte sind in den Normen für Wohnbereiche höher als in denen für Industriebereiche.

Der allgemeine Prüfablauf für die CE-relevanten EMV-Prüfungen ist hier dargestellt. Um genauere Informationen über Messaufbauten und die Prüfverfahren zu erhalten, werden für die Störfestigkeit die Normen 61000-4-X benötigt. Für Prüfungen im Bereich Störaussendung sind die Normen 61000-3-X anzuwenden.

## 4.4 Grundnormen Störaussendung

In den Fachgrundnormen EN 61000-6-3 und 4 sind vier Prüfprozesse beschrieben, die auf unterschiedliche Grundnormen verweisen:

1. Die Störaussendung des Geräts bei geschlossenem Gehäuse, hierbei handelt es sich um feldgebundene Störungen, die vorgeschriebene Maxima im Frequenzbereich von 30MHz bis 6GHz nicht überschreiten dürfen.
2. Weiter sind Grenzwerte für Störungen auf Niederspannungs-Wechselstrom-Netzanschlüsse vorgegeben, die im Bereich von 0Hz bis 30MHz maximal auf das Netz abgegeben werden dürfen. Diese Prüfung kann bei DESY mittels des Spektrumanalyzers Tektronix RSA5100 erfolgen. Die relevanten Grundnormen werden in diesem Unterkapitel anschließend benannt.
3. Es gibt Vorgaben für Gleichstromnetzanschlüsse, dort werden Maxima im Frequenzbereich von 0,15MHz bis 30MHz behandelt.
4. Telekommunikations-/ und Netzanschlüsse<sup>8</sup> sind mit besonderen Anforderungen be-

<sup>6</sup>[EMV61 \(2007\)](#); [EMV62 \(2006\)](#)

<sup>7</sup>[EMV64 \(2011\)](#); [EMV63 \(2001\)](#)

<sup>8</sup>“Anschluss für Sprach-, Daten- und Signalübertragung, der zur Verbindung mit weit verzweigten Netzen mit Hilfe solcher Mittel wie dem direkten Anschluss an Mehr-Nutzer-Telekommunikationsnetze (z.B. öffentliche Telekommunikationsnetze (PSTN), Dienste-integrierende digitale Netze (ISDN), digitale Teilnehmernetze des x-Typs (xDSL) usw.), an lokale Netze (z. B. Ethernet, Token Ring usw.) und an ähnliche Netze vorgesehen ist.” [[EMV63 \(2001\)](#), S.8]

aufschlägt, sie habe niedrigere Grenzwerte und werden im Bereich von 0,15 MHz bis 30 MHz geprüft.

Die Grenzwerte werden mit Spitzenwerten, Quasipitzenwerten und Mittelwerten definiert. Die Maßeinheit hierbei ist dB( $\mu$ V)<sup>9</sup>. Der Quasipitzenwert lässt sich nicht mit einer mathematischen Formel errechnen. Es ist ein Quasi-Spitzenwert-Detektor mit definierten Lade- und Entladekonstanten für verschiedene Frequenzbereiche notwendig. In der CISPR 16-1-1 sind die Anforderungen in Bezug auf Zeitkonstanten, Eingangsimpedanz, Messgenauigkeit und Schirmdämpfung vorgegeben. Der Quasipitzenwert bewertet schmalbandige Sinussignale stärker als breitbandige Impulse oder Rauschen. Die verwendeten Grundnormen für leitungsgebundene Störungen auf Netzanschlüssen werden nun genauer betrachtet.

### 0kHz bis 2kHz

- a) **EN 61000-3-2** Grenzwerte für Oberschwingungsströme<sup>10</sup>, Bemessungsstrom <= 16A / Leiter
- b) **EN 61000-3-3** Begrenzung von Spannungsänderungen, Spannungsschwankungen und Flicker<sup>11</sup> , Bemessungsstrom <= 16A / Leiter
- c) **EN 61000-3-11** Begrenzung von Spannungsänderungen, Spannungsschwankungen und Flicker , 16A < Bemessungsstrom <= 75A / Leiter
- d) **EN 61000-3-12** Grenzwerte für Oberschwingungsströme, 16A < Bemessungsstrom <= 75A / Leiter

### ab 0,15MHz bis 30MHz

- a) **CISPR 16-1-1, 4 und 6** Anforderungen an die Messausrüstung
- b) **CISPR 16-1-2, 4** Anforderungen an Netznachbildung
- c) **CISPR 16-2-1, 7** Messaufbau und Messverfahren
- d) **CISPR 14-1, 4.2** Grenzwerte für diskontinuierliche Störgrößen

<sup>9</sup>0dB $\mu$ V ≡ -120dBV

<sup>10</sup>Oberschwingungen sind ganzzahlige Vielfache der zugrunde gelegten Ausgangsfrequenz, im Fall des Netzanschlusses handelt es sich um 50Hz.

<sup>11</sup>Mit Flicker sind Spannungsschwankungen bezeichnet, die z. B. durch Schaltvorgänge ausgelöst werden können, sie führen zu einem wahrnehmbaren "Flackern" von Beleuchtungselementen.

## 4.5 Grundnormen für Störfestigkeit

Die Grundnormen für Störfestigkeit werden in den nächsten Kapiteln detailliert beschrieben. Der allgemeine Prüfprozess sieht vor, dass Störsignale in das Gerät eingespeist werden und währenddessen geprüft wird, ob das Gerät auf diese Störung reagiert.

### 4.5.1 EN 61000-4-2 Störfestigkeit gegen ESD

ESD (Electrostatic Discharge) wird häufig vom Anwender verursacht. Der menschliche Körper kann durch Reibung eine enorme Ladung aufbauen, welche sich schlagartig an einem Punkt des Gerätes entladen kann. Dies kann dazu führen, dass Geräte ausfallen oder kurzfristig nicht wie vorhergesehen funktionieren. Einzelne ICs (Integrated Circuits) können durch elektrostatische Entladung versagen oder so beschädigt werden, dass die Lebensdauer reduziert wird.

Die Norm 61000-4-2<sup>12</sup> sieht vor, dass mittels eines Generators an verschiedenen Punkten an einem Prüfling Kontaktentladungen sowie Luftentladungen stattfinden. Es existieren sog. Prüfschärfegrade (Prüfpegel)<sup>13</sup> die die Energie, Anstiegszeit, Amplitude und Ströme zu definierten Zeitpunkten festlegen. Jeder Punkt, an dem eine Störung eingekoppelt wird, erfordert einen Prüfschärfegrad, der im Hinblick auf Luftfeuchtigkeit, Material und Kontakt- oder Luftentladung beruht.

Die Auswahl der Prüfpunkte ist so zu wählen, dass Elemente, die der Interaktion zwischen Anwender und Baugruppe dienen, vollständig abgeprüft werden. Ebenfalls müssen metallische Teile, die nicht geerdet sind, mit ESD-Pulsen beaufschlagt werden. Die Auswahl kann auf jeden Punkt erweitert werden, der für den Anwender erreichbar ist. Eine maximale Anzahl der Prüfpunkte ist nicht festgelegt.

### 4.5.2 EN 61000-4-3 Störfestigkeit gegen hochfrequente elektromagnetische Felder

Die beiden Normen EN 61000-4-3<sup>14</sup> und 61000-4-6<sup>15</sup> bauen aufeinander auf. Es wird der Frequenzbereich von 80MHz bis 1GHz oder auch höher von dieser Norm abgedeckt. In

<sup>12</sup>EMV42 (2009)

<sup>13</sup>Prüfschärfegrade beschreiben in einer Norm definierte unterschiedliche Stufen einer Störung, die auf einen Prüfling einwirken. Die zu testende Baugruppe kann während eines Tests unter einer bestimmten Norm verschiedene hohe Prüfschärfegrade erfordern. Meist wird ein maximaler Prüfschärfegrad vorgegeben und alle niedrigeren werden mitgeprüft.

<sup>14</sup>EMV43 (2010)

<sup>15</sup>EMV46 (2014)

dieser Norm werden feldgebundene Störungen auf den Prüfling eingekoppelt. Es gibt vier Prüfschärfegrade mit unterschiedlichen Feldstärken für das Störsignal: 1V/m, 3V/m, 10V/m und 30V/m. Das HF-Feld wird mittels einer Antenne erzeugt, hierbei werden vier verschiedene Arten von Antennen von der Norm empfohlen: Die bikonische Antenne eignet sich gut für Einsätze in kleineren Räumlichkeiten, um Näherungseffekte<sup>16</sup> zu minimieren. Die logarithmisch-periodische Antenne zeichnet sich durch eine verhältnismäßig gute Verstärkung und ein niedriges Stehwellenverhältnis<sup>17</sup> aus. Für Frequenzen größer als 1GHz werden die Hornantenne und die Doppelsteg-Hohlleiterantenne empfohlen. Ein Beispiel für den Messaufbau ist in Abbildung 4.1 zu sehen.

Bevor gemessen werden kann, muss das Feld kalibriert werden. Das bedeutet, dass das Feld über dem Prüfling möglichst homogen ist. Die Feldstärke wird je nach Prüfschärfegrad eingestellt, dann kann die Prüfung beginnen. Es wird nun ein moduliertes HF-Signal mit der Antenne ausgesendet, hierbei gibt es drei verschiedene Modulationsarten: die Amplitudenmodulation<sup>18</sup>, die Rechteck-Amplitudenmodulation und die gepulste HF. Die Rechteck-AM hat einen Betriebszyklus von 1:2 und 100% mit 200Hz Modulationsfrequenz. Die gepulste HF ist anpassbar in Hinsicht auf Betriebszyklus und Modulationsfrequenz, um Betriebsmodi von Telekommunikationsgeräten nachzubilden.

<sup>16</sup>Der Begriff Näherungseffekt (proximity effect) bezeichnet in der Elektrotechnik die Wirkung der Stromein schnürung oder Stromverdrängung zwischen zwei eng benachbarten Leitern unter dem Einfluss von Wechselströmen aufgrund des magnetischen Streuflusses zwischen ihnen, verursacht durch entgegengesetzte Ströme in den Leitern [Wikipedia (2015c)]

<sup>17</sup>Das Stehwellenverhältnis ist definiert durch  $SWR = \frac{V+R}{V-R}$  wobei V die Amplitude des Signal ist und R der Reflektionsfaktor. Ein niedriger SWR ist erwünscht, damit möglichst viel Leistung übertragen und nicht reflektiert wird.

<sup>18</sup>es handelt sich um das gleiche Signal wie in 4.5.5, dort wird die Modulationsart näher beschrieben

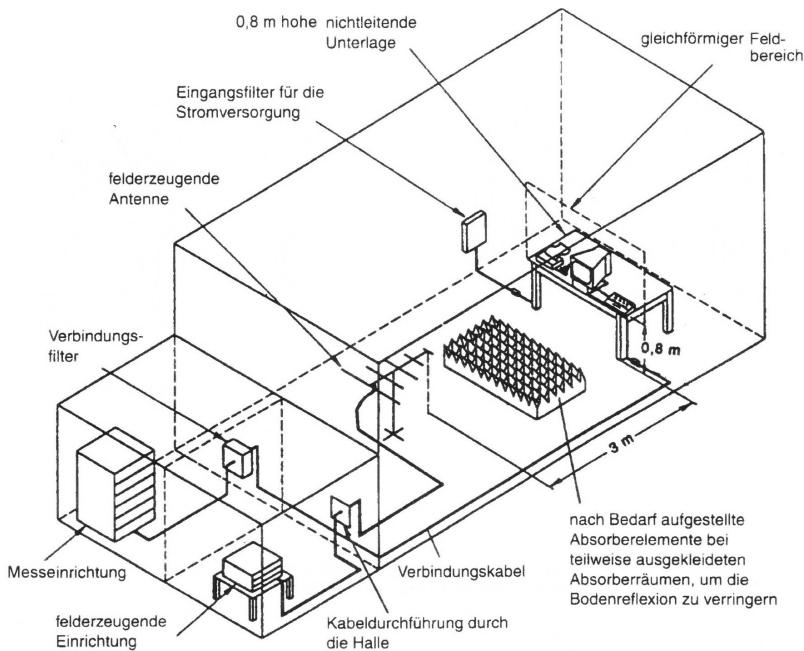


Abbildung 4.1: EN 61000-4-3: Versuchsaufbau für feldgebundene Störungen [[EMV43 \(2010\)](#), S.26]

#### 4.5.3 EN 61000-4-4 Störfestigkeit gegen schnelle Transiente (Burst)

Bursts sind Störungen, die entstehen können, wenn Induktivitäten nach der Öffnung des Stromkreises weiter Strom treiben und so einen abrupten Spannungsanstieg zur Folge haben<sup>19</sup>. Dies kann bei Schaltvorgängen mit hohen Lasten, beispielsweise beim Abschalten eines Motors, passieren. Ein Burst ist dem Surge in Bezug auf die Signalform ähnlich. Sie unterscheiden sich allerdings sehr in der zeitlichen Auflösung und somit auch in der Energie eines solchen Pulses. Ein Burst-Störpuls (siehe Abbildung 4.2) hat eine Anstiegszeit<sup>20</sup> von ca. 5ns und eine Impulsdauer<sup>21</sup> von ca. 50ns, ein Surge-Pulse hat eine Anstiegszeit von 1µs und eine Impulsdauer von ca. 50µs.

Ein Burst besteht aus vielen Pulsen in einem Zeitfenster von 15ms, die Pulse wiederholen sich mit einer Frequenz von 5kHz (oder auch 100kHz je nach Prüfschärfegrad)<sup>22</sup>. Ein Test sieht vor, dass dies, für eine Dauer von mindestens einer Minute, alle 300ms wiederholt wird. Die Amplitude wird einmalig für den erforderlichen Prüfschärfegrad ermittelt und kann

<sup>19</sup> [Stotz \(2013\)](#), Vgl. S.44

<sup>20</sup> Die Anstiegszeit ist die Zeit, die während des initialen Anstiegs vergeht, zwischen einem Pegel von 10% des Maximums bis zu 90% des Maximums

<sup>21</sup> Die Impulsdauer ist die Zeit zwischen dem Erreichen von 50% der maximalen Spannung bis zum Abfall auf 50% der Spannung

<sup>22</sup> [EMV44 \(2012\)](#)

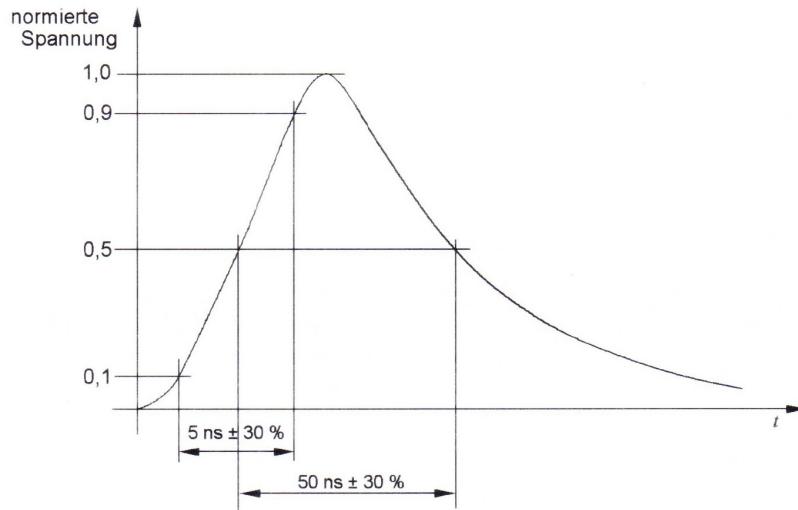


Abbildung 4.2: EN 61000-4-4: Burst Störpuls [EMV43 (2010), S.21]

entweder 0,5kV, 1kV, 2kV oder 4kV betragen. Der Test muss für beide Polaritäten erfolgen. In Abbildung 4.3 ist das Störsignal dargestellt, wie es vor Ort bei DESY gemessen wurde. Bei a) sind die Wiederholungen des Störpulses nicht sehr gut erkennbar aufgezeichnet, der Kontrast des Bildes wurde heruntergesetzt um diese sichtbarer zu machen. Bei genauem Hinsehen ist zu erkennen, dass sich die Störpulse alle 2ms wiederholen (5kHz).

Die Einkopplung der Störung kann auf zwei verschiedene Arten erfolgen. Entweder wird die Störung direkt auf die Spannungsversorgung in alle Leitungen gleichzeitig eingekoppelt oder sie wird mittels einer kapazitiven Koppelzange<sup>23</sup> in eine Kommunikationsleitung eingeprägt. Bei einer Einkopplung in die Spannungsversorgung wird in der Regel eine Amplitude von 2kV verwendet, bei Kommunikationsleitungen 1kV.

<sup>23</sup>Eine kapazitive Koppelzange ist ein ca. 1m langer metallischer Kanal, in den die zu beaufschlagende Leitung hineingelegt wird, anschließend wird er oben mit einer Metallplatte verschlossen. Durch die kapazitive Kopplung kann ein Signal in die Leitung eingeprägt werden, wobei keine galvanische Verbindung vorhanden ist. Die Einkopplung geschieht ausschließlich über die Felder.

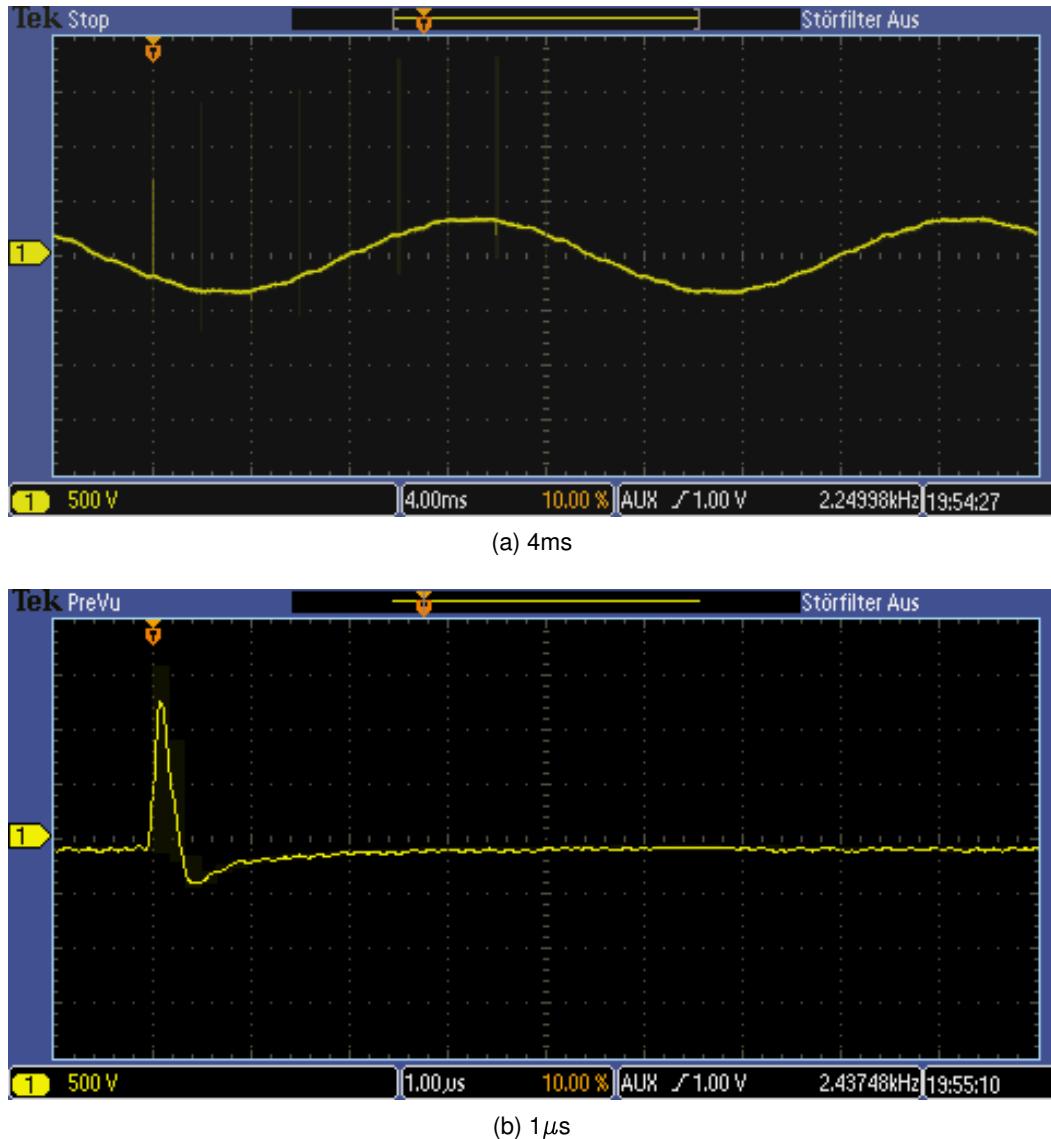


Abbildung 4.3: EN 61000-4-4: Störsignal mit 2kV 5kHz 15ms, verschiedene Auflösungen

#### 4.5.4 EN 61000-4-5 Störfestigkeit gegen Stoßspannungen (Surge)

Ein Surge ist ein hochenergetischer Puls, der unter realen Bedingungen z. B. in Form eines indirekten Blitzschlags auftritt. Er kann auch durch Schaltüberspannungen, die bei Zuschaltung von Kondensatorbatterien auftreten können<sup>24</sup>, entstehen. Die Einkopplung des Surge-Pulses geschieht in der Regel über die Netzeleitung des Prüflings. Seltener werden auch Kommunikationsleitungen geprüft.

Es gibt 4 Prüfschärfegrade (0.5kV, 1kV, 2kV und 4kV), die je nach Anforderung an das Gerät im Test angewandt werden. Der Puls hat eine Anstiegszeit von ca.  $1\mu\text{s}$  und eine Impulsdauer von ca.  $50\mu\text{s}$ , dies gilt für die Leerlaufspannung, der Puls ist in Abbildung 4.5 für die Leerlaufspannung und den Kurzschlussstrom dargestellt. In der Abbildung 4.4 sind drei unterschiedliche Störimpulse zu sehen, die mit dem CEG-4500 erstellt wurden. In Bild a) ist ein Puls bei  $0^\circ$  Phase zu sehen, in Bild b) ist ein Puls bei  $90^\circ$  zu sehen und in c) ein negativer Puls bei  $0^\circ$ . Der Puls in b) ist von den dreien am kritischsten, da er auf das Maximum des Sinus der Netzspannung aufaddiert wird und so eine maximale Spannung von ca. 1,5kV kurzzeitig auftritt (vgl. a) 1,25kV c) -1,27kV). Ein weiterer kritischer Puls ist ein negativer bei  $270^\circ$ , hier wird die Spannung maximal negativ. Besonders diese Pulse sind obligatorisch bei dem Prüfprozess.

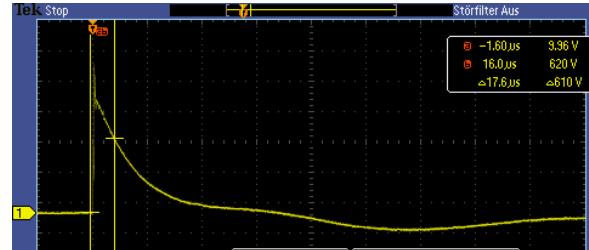
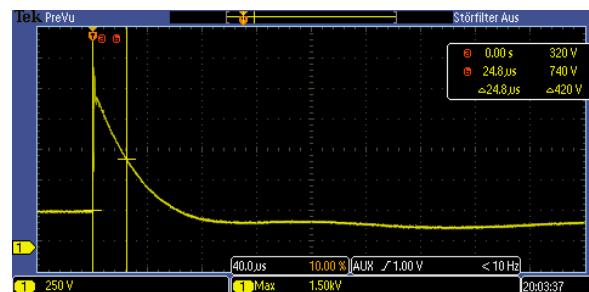
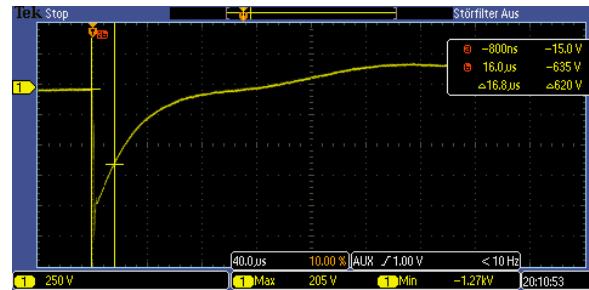
(a) 1kV  $0^\circ$  Phase(b) 1kV  $90^\circ$  Phase(c) -1kV  $0^\circ$  Phase

Abbildung 4.4: EN 61000-4-5: gängige Amplituden-/Phasenkombinationen

<sup>24</sup>Stotz (2013), S.57

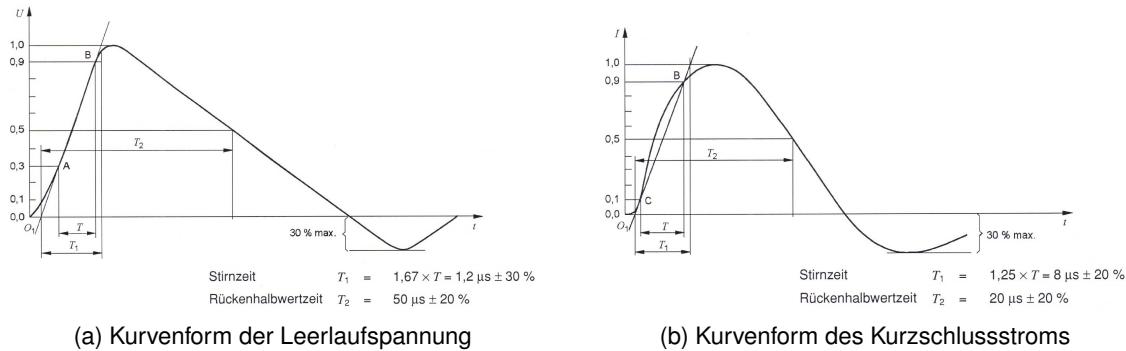


Abbildung 4.5: EN 61000-4-5: Surge Störpuls [EMV45 (2015), S.15]

Die Pulse werden immer gegen zwei Leitungen induziert. Bei einphasigem Netzanschluss mit L,N und PE ergeben sich so 3 verschiedene Möglichkeiten, die abgeprüft werden können. Bei dreiphasigem Netzanschluss mit L<sub>1</sub>,L<sub>2</sub>,L<sub>3</sub>,N und PE ergeben sich bereits 10 Möglichkeiten. In der Norm ist vorgeschrieben, dass mindestens 5 Pulse, jeweils positiv und negativ, mit einem maximalen Abstand von 1min bei 0°, 90°, 180° und 270° vollzogen werden<sup>25</sup>. Dies geschieht für jeden Prüfschärfegrad bis zu dem am höchsten eingeforderten.

Ein Prüfaufbau, wie er mit dem CEG-4500 in Verbindung mit dem CFG-4028 möglich ist, ist in Abbildung 4.6 abgebildet. Diese Abbildung wurde der Norm EN 61000-4-5 entnommen. Es besteht mit diesem Prüfaufbau die Möglichkeit, alle Leitungs-Kombinationen, bis auf die mit der Leitung PE, gegeneinander zu prüfen.

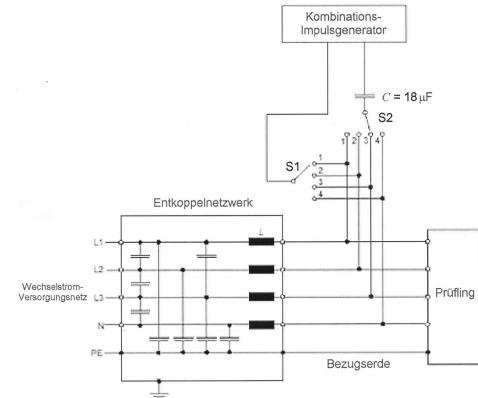


Abbildung 4.6: EN 61000-4-5: Surge Prüfaufbau [EMV45 (2015), S.21]

#### 4.5.5 EN 61000-4-6 Störfestigkeit gegen leitungsgeführte Störgrößen, induziert durch HF-Felder

Diese Norm - wie auch schon die Norm EN 61000-4-3 - behandelt HF-Felder. Hier werden leitungsgebunden induzierte HF-Felder bis 80MHz erzeugt. Es existieren 3 Prüfschärfegrade, die den Störpegel auf 1V, 3V oder 10V festlegen. Das Störsignal ist in Abbildung 4.7 zu sehen. Es handelt sich um ein 80% amplitudemoduliertes HF-Signal mit einer Modulationsfrequenz von 1kHz. Die Einkopplung des Störsignals geschieht entweder über ein Kop-

<sup>25</sup>EMV45 (2015)

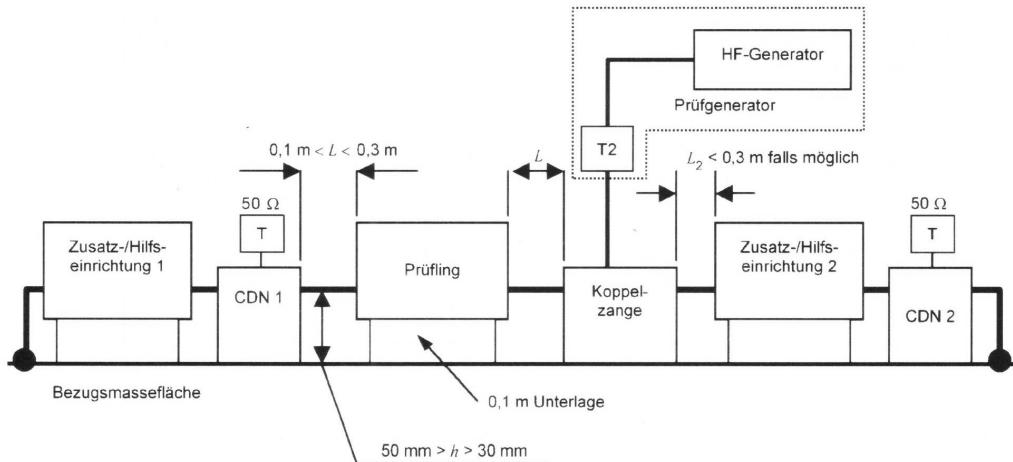


Abbildung 4.8: EN 61000-4-6: Prüfaufbau [EMV46 (2014) S.25]

pelnetzwerk oder eine kapazitive Koppelzange. Es wird auf alle Leitungen gleichzeitig, wie es auch bei feldgebundener Störungseinwirkung wäre, eingekoppelt. In Abbildung 4.8 ist ein schematischer Prüfaufbau für die Einkopplung mittels Koppelzange dargestellt. Die Norm sieht vor, dass ab einer Frequenz von 150kHz in maximal 1% -Schritten die Frequenz erhöht wird und jedes Störsignal mindestens eine Sekunde anliegt.

#### 4.5.6 EN 61000-4-8 Prüfung der Störfestigkeit gegen Magnetfelder mit energietechnischen Frequenzen

Auf den Prüfling werden während dieses Norm-Tests zwei verschiedene Magnetfelder gelegt:

Zum einen ein Dauerfeld mit Prüfschärfegraden von 1,3,10,30 und  $100 \frac{A}{m}$  magnetische Feldstärke und zum anderen ein kurzzeitiges Feld von 300 oder  $100 \frac{A}{m}$  für die Prüfschärfegrade 4 und 5. Bei den niedrigeren Prüfschärfegraden ist kein kurzzeitiges Feld notwendig. Mit einer Induktionsspule wird das Feld auf das Gerät gelegt. In Abbildung 4.9 ist der Prüfvorgang dargestellt. Die Induktionsspule wird über den Prüfling bewegt, so dass das Magnetfeld einmal komplett das Gerät durchdrungen hat. Dies wird dreimal wiederholt, so dass dies von vorn nach hinten, von links nach rechts und von oben nach unten geschieht.

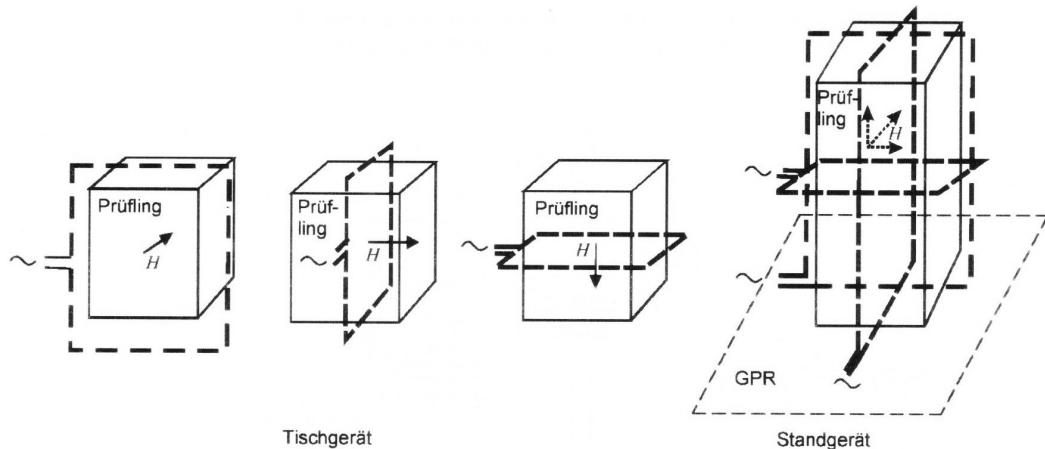


Abbildung 4.9: EN 61000-4-8: Prüfprozess [EMV48 (2010), S.17]

#### 4.5.7 EN 61000-4-11 Störfestigkeit gegen Spannungseinbrüche, Kurzzeitunterbrechungen und Spannungsschwankungen

In dieser Norm wird die Störfestigkeit in Bezug auf kurzzeitige Unterbrechung oder Abfall der Spannungsversorgung behandelt. Die Störung in Form einer Absenkung der Versorgungsspannung wird mittels eines Generators zwischen Spannungsversorgung und Prüfling erzielt.

Es existieren 3 Klassen für den Prüfschärfegrad<sup>26</sup>:

In Klasse 1 sind Geräte üblicherweise durch unterbrechungsfreie Stromversorgungen geschützt. Dies ist etwa sinnvoll für Messgeräte in Laboratorien. Die Anforderungen in dieser Klasse sind an die Geräte und die Bedingungen anzupassen.

Die Klasse 2 behandelt Geräte, die in öffentlichen Netzen sowie in industrieller Umgebung genutzt werden können. In ihr ist vorgesehen, dass zwei Spannungseinbrüche auf 0% mit Dauer 10ms und 20ms und ein Abfall auf 70% für 0.5s stattfinden. Außerdem wird eine Kurzzeitunterbrechung von 5s auf 0% durchgeführt.

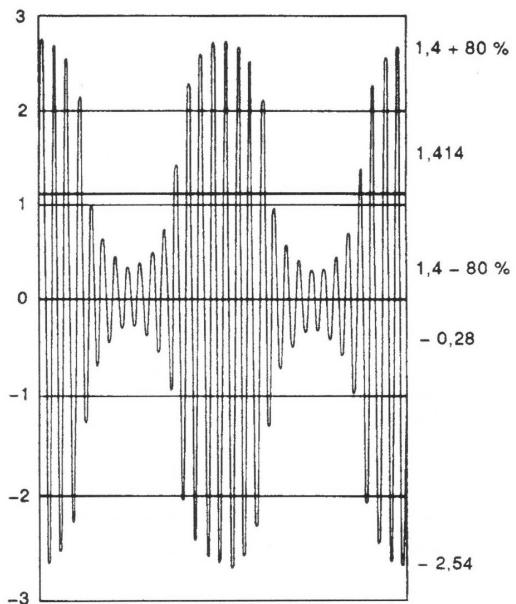


Abbildung 4.7: EN 61000-4-6: 80% AM HF-Signal [EMV46 (2014), S.26]

<sup>26</sup> EMV411 (2004), S.20

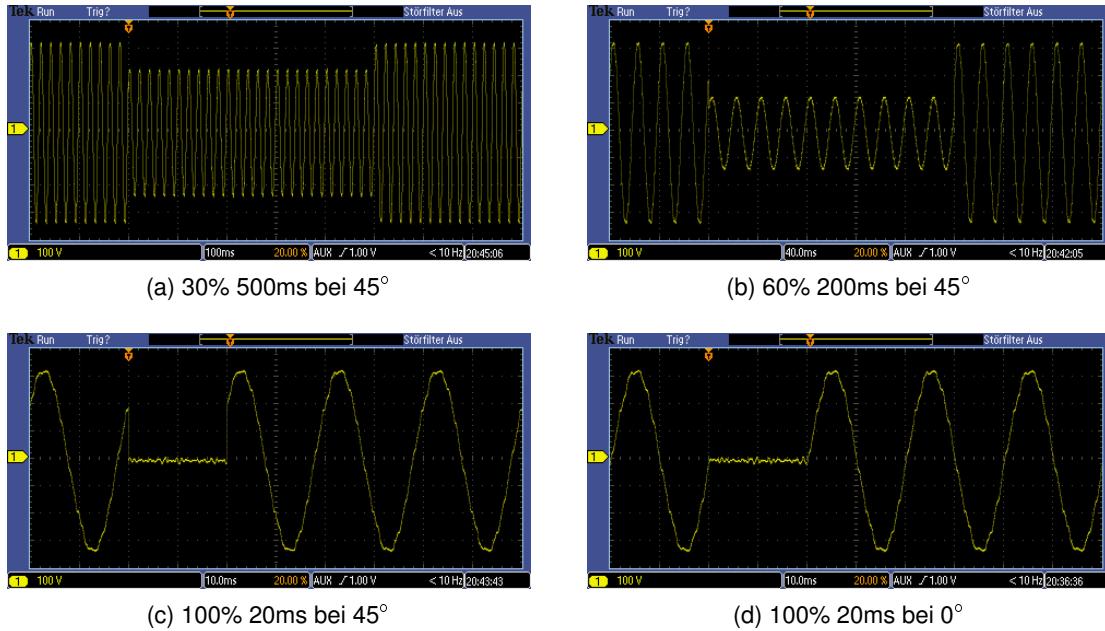


Abbildung 4.10: Spannungseinbruch

Klasse 3 beinhaltet Geräte, die in industrieller Umgebung genutzt werden und eine der folgenden Bedingungen erfüllen: Ein Hauptteil der Last wird durch einen Stromrichter gespeist, Schweißgeräte werden genutzt, große Motoren werden häufig gestartet oder Lasten schwanken schnell. Die Anforderungen sind wie in Klasse 2 mit zwei zusätzlichen Spannungseinbrüchen: 40% für 200ms und 80% für 5s.

Der Prüfablauf sieht vor, dass alle Pegel, außer der Abfall auf 0% für 5s, unter verschiedenen Phasenwinkeln, vorzugsweise  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ ,  $180^\circ$ ,  $225^\circ$ ,  $270^\circ$  und  $315^\circ$ , 3 Mal mit 10s Abstand wiederholt werden. Zur Veranschaulichung sind in Abbildung 4.10 vier verschiedene Spannungseinbrüche dargestellt. Bei Drehstrom wird jeder L-Leiter einzeln dieser Prüfung unterzogen. Die Bewertungskriterien sind wie gewohnt anzuwenden. Es werden bei den verschiedenen Spannungsabfällen unterschiedlich hohe Bewertungskriterien an den Prüfling gestellt. Die Norm erwartet aber beispielsweise nicht, dass ein Gerät der Klasse 2 die Prüfung der Kurzzeitunterbrechung von 5s mit einem Bewertungskriterium, das besser als C ist, abschließen wird.

# 5 Analyse

Es wird eine Analyse der Stakeholder und ihrer Ziele vorgenommen. Die Use-Cases des Hauptprogramms werden erläutert und mit Diagrammen veranschaulicht.

## 5.1 Stakeholder und Ziele

Es wurden folgende Stakeholder ermittelt:

1. **Entwickler** (Jan Scholz)
2. **Anwender und Prüfer** (Mitarbeiter der Fachgruppe ZE-Prüffeld)
3. **Abteilungsleiter**
  - a) Herr Zeides, Leiter der Gruppe ZE
  - b) Frau Voigt, Leiterin der Fachgruppe ZE-Prüffeld
4. **Hersteller**
  - a) EM TEST GmbH ⇒ CWS 500N1
  - b) Peter Hofbauer Electronic GmbH ⇒ CEG-4500, CFG-4028
  - c) Tektronix ⇒ RSA5100
5. **Normen** (siehe Kapitel 4.2 und folgende)
6. **Marketing und Copyright** (Abteilung Technologie-Transfer)
7. **Betreuernder Professor** ( Prof. Dr. Dierks)
8. **Entwickler von Baugruppen** (Diverse Abteilungen bei DESY)

Die Stakeholder und ihre Ziele werden mittels einer Schablone von Stefan Kleuker<sup>1</sup> verdeutlicht. In ihr wird das Ziel, die betroffenen Stakeholder und Auswirkungen für sie im Zusammenhang mit dem Ziel veranschaulicht. Weiter werden Randbedingungen für das Ziel und Abhängigkeiten mit anderen Zielen aufgezeigt. Die Schablone sieht außerdem die Kategorie Sonstiges vor, die der Vollständigkeit halber mit erwähnt wird aber nicht gefüllt wurde. Mit Hilfe dieser Schablone werden die Intentionen der Stakeholder verdeutlicht und gleichzeitig gemeinsame Ziele erkannt.

---

<sup>1</sup>Kleuker (2013), S.61

Tabelle 5.1: Stakeholder-Analyse: 1.Ziel

Ziele	Es soll die Möglichkeit geben, alle leistungsgebundenen Störungs-Tests durch die Software auszuführen. Alle benötigten Geräte sollen von der Software gesteuert werden. Es soll ein dynamisch generierter Prüfbericht als Pdf am Ende der Prüfung generiert werden.
Stakeholder	Anwender, Abteilungsleiter, Entwickler
Auswirkungen auf Stakeholder	Die Prüfung von Prototypen wird immens verkürzt. Des weiteren sind mögliche Fehler durch falsche Eingabe an den Geräten minimiert. Die Dokumentation der Prüfung ist mit dem Abschluss der Prüfung sofort verfügbar.
Randbedingungen	Vorausgesetzt ist, dass der Entwickler genügend Zeit hat, die Software zu entwickeln. Die zur Kommunikation mit den Geräten notwendige Hardware muss zur Verfügung gestellt werden. Die Schnittstellendefinition für die anzusprechenden Geräte muss verfügbar sein.
Abhängigkeiten	
Sonstiges	

Tabelle 5.2: Stakeholder-Analyse: 2.Ziel

Ziele	Die entwickelte Software soll frei zugänglich sein
Stakeholder	Entwickler, Hersteller Störgeneratoren, Anwender
Auswirkungen auf Stakeholder	Der Hersteller der Generatoren profitiert davon, dass bereits Software zur Ansteuerung der Geräte existiert und kann dies als Kaufargument nutzen. Der Anwender profitiert von möglichen Weiterentwicklungen durch Open-Source.
Randbedingungen	Wenn die Software frei zugänglich sein soll, darf keine proprietäre Software integriert werden
Abhängigkeiten	Dieses Ziel ist nicht vereinbar mit dem 3. Ziel
Sonstiges	

Tabelle 5.3: Stakeholder-Analyse: 3.Ziel

Ziele	Die entwickelte Software soll nur bedingt zugänglich sein
Stakeholder	Abteilung Technologie-Transfer
Auswirkungen auf Stakeholder	Wenn die Software nicht frei verfügbar ist, kann das Unternehmen DESY unter Umständen damit Geld verdienen, wenn Personen diese Software oder sogar den Source-Code verwenden möchten.
Randbedingungen	Wenn die Software nicht frei sein soll, darf keine Software mit einer Copyleft-Lizenz integriert werden, da das resultierende Endprodukt dann auch mit einer Copyleft-Lizenz versehen werden muss wie z. B. eine GPL.
Abhängigkeiten	Dieses Ziel ist gegensätzlich zu dem zweiten Ziel und muss mit den Parteien besprochen werden
Sonstiges	

Tabelle 5.4: Stakeholder-Analyse: 4.Ziel

Ziele	Die Software soll einfach erweiterbar sein
Stakeholder	Abteilungsleiter, Anwender, Hersteller
Auswirkungen auf Stakeholder	Wenn in der Zukunft weitere Geräte hinzugekauft oder ausgetauscht werden, ist es auch für Entwickler, die das Programm nicht entworfen haben, möglich, das Programm so anzupassen, dass ein neues Gerät auch genutzt werden kann. Das Programm ist nicht nur für diesen speziellen Fall einsetzbar, sondern kann auch von der Allgemeinheit genutzt werden.
Randbedingungen	Das Programm muss so entwickelt werden, dass die Ansprache von Geräten so verallgemeinert wird, dass jedes Gerät auf diese Weise angesprochen werden kann.
Abhängigkeiten	
Sonstiges	

Tabelle 5.5: Stakeholder-Analyse: 5.Ziel

Ziele	Das Programm soll bis Ende August fertiggestellt sein
Stakeholder	Abteilungsleiter, Anwender, Entwickler, betreuender Professor
Auswirkungen auf Stakeholder	Da das Programm im Rahmen einer Bachelor-Arbeit entwickelt wird und diese zeitlich befristet ist, ist es notwendig, dass das Programm bis zum Ende der Arbeit fertig ist. Sollte dies nicht der Fall sein, kann mit dem Programm nicht gearbeitet werden. Die Bachelor-Arbeit kann bei einem nicht lauffähigem Programm nicht den vollständigen Prozess der Software Entwicklung abbilden.
Randbedingungen	Das Projekt ist sehr umfangreich. So muss unter Umständen etwas an Funktionalität zu Gunsten der Fertigstellung eingebüßt werden
Abhängigkeiten	Die Ziele 1,2 und 4 können in Mitleidenschaft gezogen werden, falls dieses Ziel nicht erreicht wird.
Sonstiges	

Tabelle 5.6: Stakeholder-Analyse: 6.Ziel

Ziele	Die Tests sollen im Rahmen der Normen zur CE-Zertifizierung abgeschlossen werden
Stakeholder	Entwickler, Normen, Anwender, Abteilungsleiter, Entwickler von Baugruppen und Geräten
Auswirkungen auf Stakeholder	Die Tests müssen belastbar sein, damit die getesteten Geräte den Anforderungen der CE-Zertifizierung genügen. Sind die Ergebnisse der Tests nicht wiederholbar, so ist die Software nicht tauglich, um die Geräte in Hinsicht auf die EU-Konformität der EMV zu untersuchen.
Randbedingungen	Die Normen müssen aktuell sein, damit die gültigen Maßstäbe zur Zertifizierung eingehalten werden können.
Abhängigkeiten	
Sonstiges	

## 5.2 Use-Cases

Die Use-Cases wurden in zwei größere Blöcke aufgeteilt: das Hauptmenü mit seinen Funktionen unter “Report verwalten” und das -Unterprogramm mit “Test durchführen”. Die beiden Blöcke werden in Diagrammen dargestellt und im Folgenden näher erläutert.

### 5.2.1 Report verwalten

In diesem Block werden die Features, die im Hauptmenü vorhanden sind, genauer beschrieben. Ein Use-Case-Diagramm ist in Abbildung 5.1 zu sehen. Die einzelnen Use-Cases werden im nachfolgenden Teil genauer aufgeschlüsselt. Auf Basis dieses Blocks wurde außerdem ein Aktivitätsdiagramm entworfen. In Abbildung 5.2 sind die möglichen Abläufe visualisiert.

#### 1. Prüfnorm wählen

Der Anwender kann zwischen sechs verschiedenen Tests im Sinne der Normen aus Kapitel 4.2 auswählen und muss, wenn noch kein Test unter einer dieser Normen vollzogen wurde, einen Prüfaufbau hinterlegen, der aus den genutzten Geräten, Bildern vom Versuchsaufbau und aus einem Kommentar des Anwenders besteht. Im Anschluss hat der Anwender die Möglichkeit, einen Test für die gewählte Norm durchzuführen.

##### a) Leitungsgeführte Störgrößen, induziert durch HF-Felder

Das Gerät emTest CWS 500N1 ist in der Lage, die von der Norm (siehe 4.5.5) geforderten Störimpulse zu generieren. Der Anwender hat die Möglichkeit, mit diesem Gerät einen Test an dem Prüfling durchzuführen.

##### b) schnelle Transiente (Burst)

Der Kombi-Generator CEG-4500 wird für diesen Test genutzt. Die zugrundeliegende Norm wird in Kapitel 4.5.3 genauer beschrieben.

##### c) Stoßspannungen (Surge)

Der Kombi-Generator CEG-4500 wird für diesen Test genutzt. Die zugrundeliegende Norm wird in Kapitel 4.5.4 genauer beschrieben.

##### d) Spannungseinbrüche, Kurzzeitunterbrechungen und Spannungsschwankungen

Der Kombi-Generator CEG-4500 wird für diesen Test genutzt. Die zugrundeliegende Norm wird in Kapitel 4.5.7 genauer beschrieben.

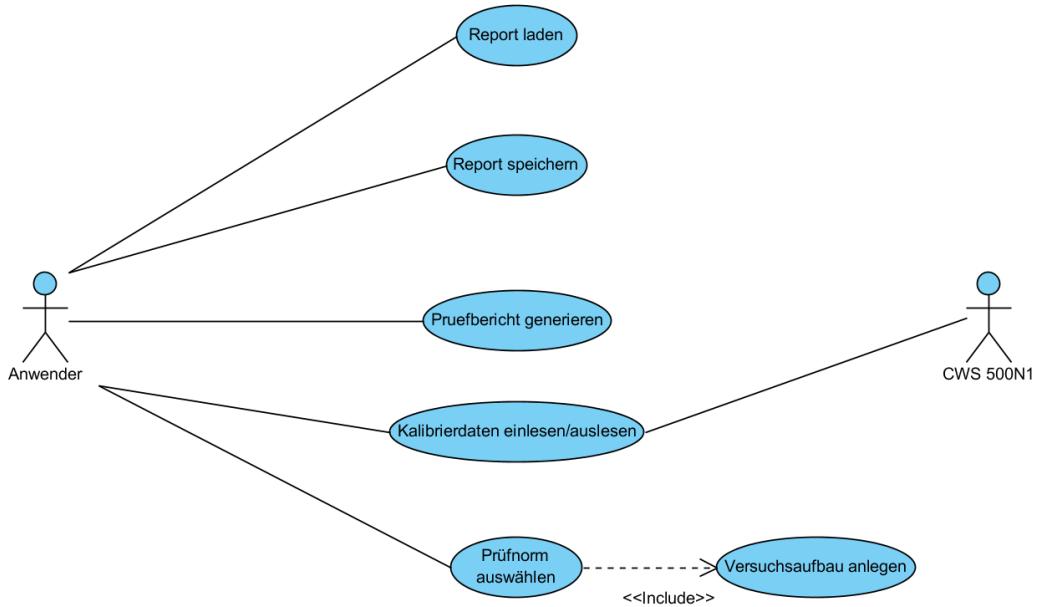


Abbildung 5.1: Use-Case Diagramm “Report verwalten”

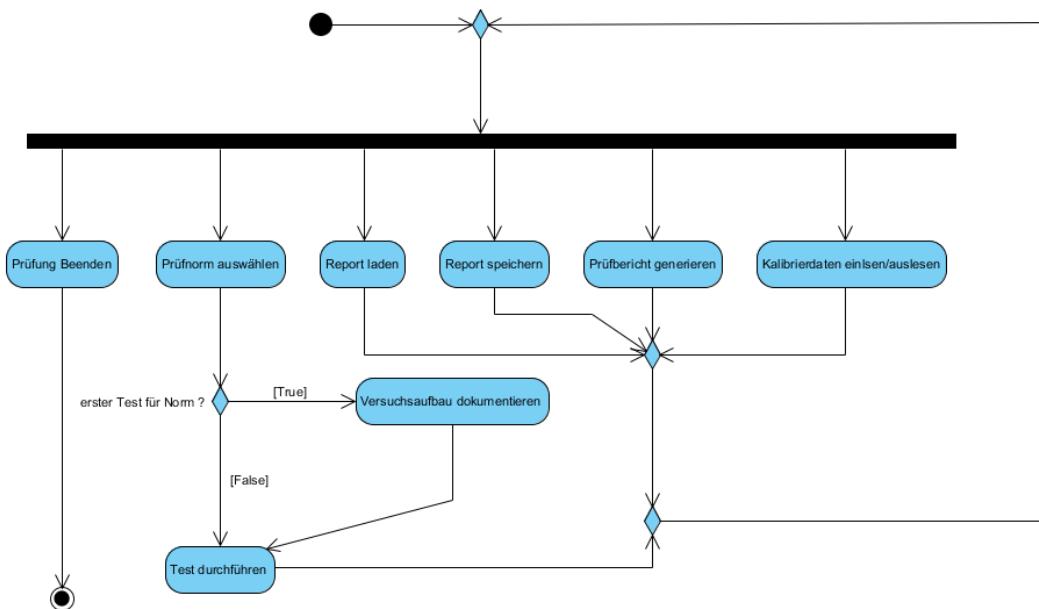


Abbildung 5.2: Aktivitätsdiagramm “Report verwalten”

**e) ESD**

Für diese Norm ist kein automatisierter Test hinterlegt, da mit einer sog. ESD-Pistole manuell an dem Prüfling an vorher festgelegten Punkten Spannungen entladen werden müssen. Die Details der Prüfung sind in Kapitel 4.5.1 erläutert.

**f) hochfrequente elektromagnetische Felder**

Dieser Test behandelt die Störaussendung des EUT. Hierfür wird ein Spektrum-analyzer benötigt. Ursprünglich war vorgesehen, das Gerät Tektronix RSA5100 mittels des Programms anzusprechen und so einen automatisierten Test zu ermöglichen. Dies wurde allerdings dadurch erschwert, dass das Gerät defekt und mehrere Wochen in der Reparatur war. Die relevanten Normen für diesen Test sind in Kapitel 4.4 vorgestellt.

Für die Punkte e) und f) gilt, dass die Versuche manuell durchgeführt werden und im Programm dokumentiert werden können, so dass im Prüfbericht auch diese Tests vorhanden sind.

**2. Report laden**

Im Hauptmenü hat der Anwender die Möglichkeit, über "Report laden" eine zu einem früheren Zeitpunkt angefangene Prüfung wieder aufzurufen. Über einen Datei-Dialog wird eine Datei ausgewählt und die Daten werden in die Datenstruktur des Programms übernommen.

Falls keine Datei ausgewählt wird oder der Dialog vom Anwender geschlossen wird, wird keine Aktion durchgeführt und der Anwender gelangt wieder ins Hauptmenü.

**3. Report speichern**

Im Hauptmenü kann ein Report abgespeichert werden, der Anwender kann mit der Funktion "Report speichern" einen Datei-Dialog öffnen und die Report-Daten in einer Datei mit der Endung XML abspeichern.

Dieser Dialog kann abgebrochen werden, der Anwender gelangt wieder ins Hauptmenü.

**4. Prüfbericht generieren**

Der Anwender kann eine PDF-Datei von den während der Prüfung gesammelten Daten erstellen. Es wird ein Datei-Dialog geöffnet, in dem der Anwender einen Dateinamen für die gewünschte PDF-Datei wählen oder eine existierende Datei auswählen kann. Die Daten werden in einer XML-Datei abgelegt und an das Programm "pdfpruef-bericht" übergeben. Dort werden die Daten eingelesen und es wird eine dynamisch generierte PDF-Datei erstellt.

Sollte der Dialog vom Anwender abgebrochen werden, gelangt er wieder ins Hauptmenü.

## 5. Kalibrierdaten einlesen/auslesen

Das Gerät emTest CWS 500N1 verfügt über mehrere Kalibrierungsspeicher, die notwendig sind, um die verschiedenen CDNs zu kompensieren<sup>2</sup>. Wenn der Anwender im Tab "Einstellungen" den Knopf "Kalibrierung von CDN" drückt, erscheint ein Dialog mit drei Wahlmöglichkeiten:

- a) **Einlesen:** Der Anwender wählt per Datei-Dialog eine Kalibrierdatei aus und die Daten werden an das Gerät übermittelt. Nachdem die Daten vollständig übermittelt sind, sind im Kalibrierungsspeicher K1 die Daten hinterlegt und können im nächsten Test verwendet werden.
- b) **Auslesen:** Das Programm sendet den Befehl zum Auslesen der Kalibrierungsdaten an das Gerät, welche wiederum an das Programm gesendet werden. Nachdem alle Daten empfangen wurden, kann per Datei-Dialog eine Datei erstellt oder ausgewählt werden, in der die Daten abgelegt werden sollen.
- c) **Abbrechen:** Der Dialog wird geschlossen.

---

<sup>2</sup>Die CDNs müssen jedes Jahr einmal neu kalibriert werden, da sich die elektrischen Eigenschaften über die Zeit ändern können. Die Kalibrierung wird an dem Gerät CWS 500N1 direkt vollzogen.

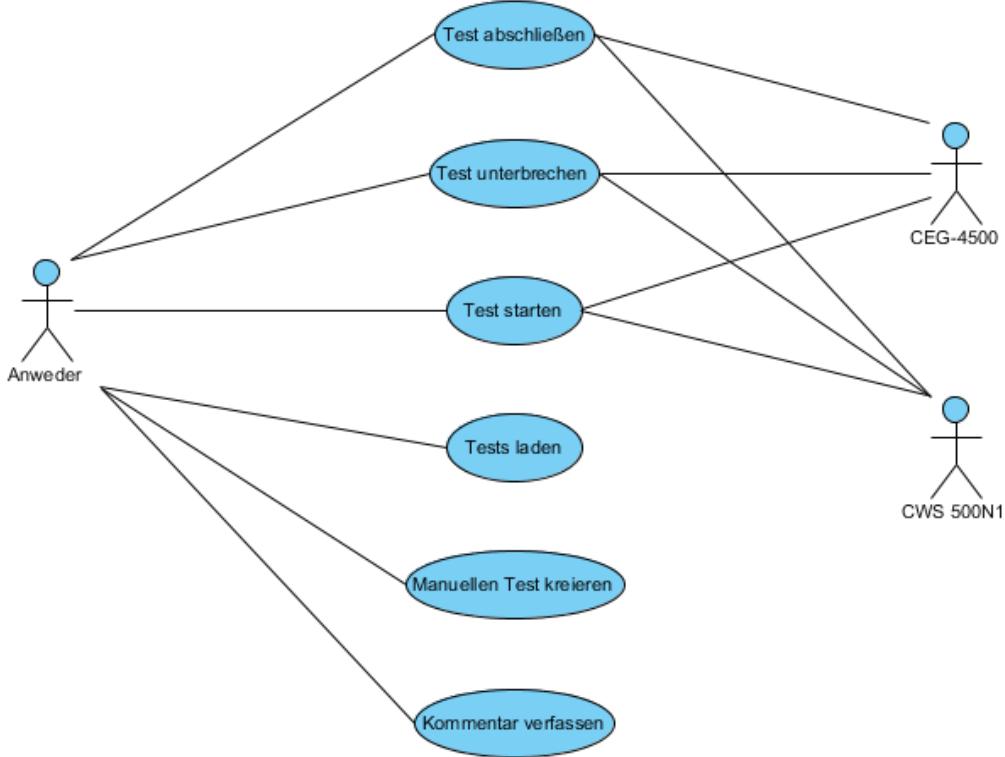


Abbildung 5.3: Use-Case Diagramm "Test durchführen"

### 5.2.2 Test durchführen

Wenn der Anwender sich für eine Prüfnorm entschieden hat und ein Versuchsaufbau dokumentiert wurde, wird er in das Unterprogramm zum Testen überführt. Die Use-Cases sind in Abbildung 5.3 als Use-Case-Diagramm dargestellt. In Abbildung 5.4 wurde ein Aktivitätsdiagramm erstellt, das den Prozess des Tests nochmals besser veranschaulicht. Im Block "Test durchführen" ergeben sich folgende Use-Cases:

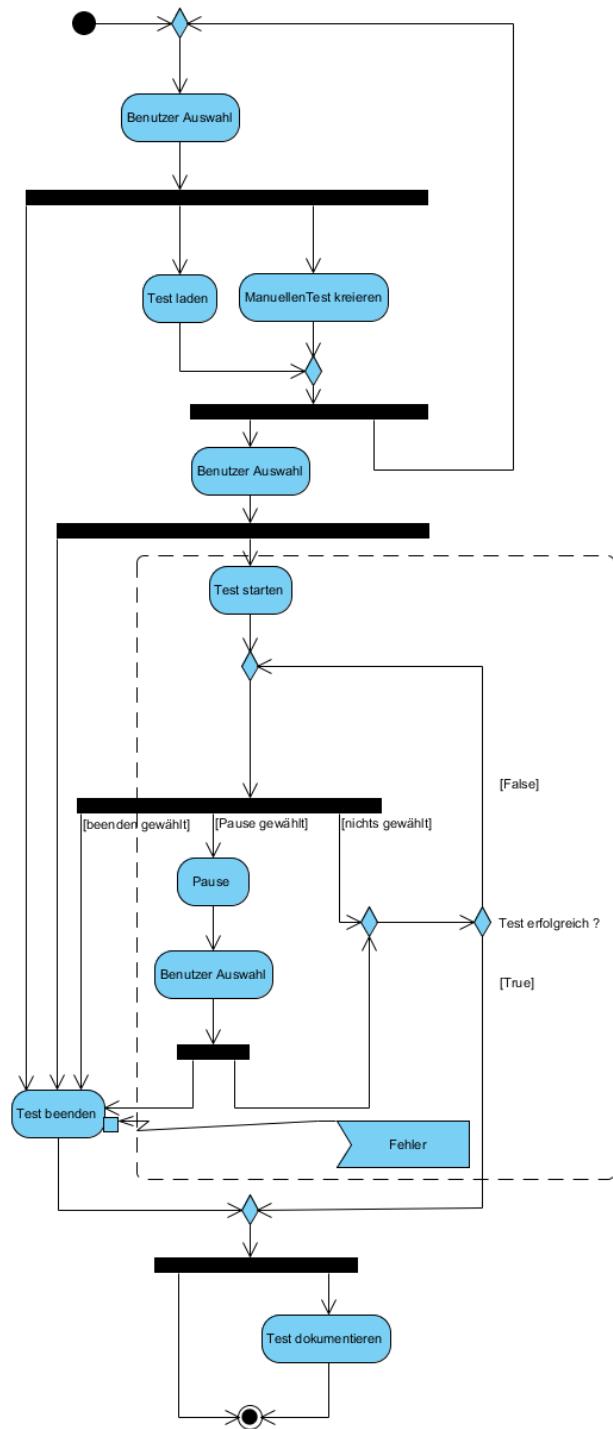


Abbildung 5.4: Aktivitätsdiagramm "Test durchführen"

### 1. Test unterbrechen

Der Anwender kann einen Test pausieren, indem er den Button "Pause" drückt. Im Anschluss kann der Test wieder aufgenommen werden. Die Geräte werden in einen sicheren Zustand gebracht<sup>3</sup> und unterbrechen den Test.

### 2. Test starten

Wenn ein Test ausgewählt ist, kann dieser vom Anwender mit dem Start-Button gestartet werden. Falls noch kein Test ausgewählt ist, wird der Anwender per Dialog aufgefordert, einen Test zu kreieren oder auszuwählen. Es wird überprüft, ob die Geräte angeschlossen sind. Ist dies nicht der Fall, wird eine Fehlermeldung ausgegeben und der Test wird beendet. Wenn das Gerät erkannt wurde, wird der Test gestartet.

### 3. Test laden

Ein Dialog erscheint mit einer Liste der verfügbaren Tests. Der Anwender muss einen Test auswählen und bestätigen, damit der Test übernommen wird.

- a) Es sind je nach Norm mindestens ein oder mehrere standardisierte Tests vorhanden, die genau die Vorgaben der Norm abbilden.
- b) Alle seit Programmaufruf getätigten Tests werden für diese Norm auch zur Auswahl gestellt, damit sie wiederholt werden können.
- c) Der Anwender hat außerdem die Möglichkeit, die Auswahl abzubrechen und kommt so wieder zurück zum Test-Fenster.

### 4. Manuellen Test kreieren

Für Prüfnormen, die mit den Geräten CEG-4500 oder CWS 500N1 getestet werden können, wurden Dialoge erstellt, die dem Anwender ermöglichen, die verfügbaren Parameter nach seinen Wünschen einzustellen. Hierbei wird beachtet, dass die Geräte die gewünschten Parameter auch umsetzen können. Ist dies nicht der Fall, wird der Parameter in den Wert mit der geringsten absoluten Entfernung umgeändert, so dass der Test erfolgen kann.

### 5. Kommentar verfassen

Der Anwender kann während der gesamten Laufzeit des Testablaufs in einer editierbaren Text-Zelle Kommentare verfassen.

### 6. Test abschließen

Der Anwender kann den Test zu jedem Zeitpunkt beenden. Nachdem der Anwender den Test beendet hat, wird mittels eines Dialogs dokumentiert, welche Erkenntnisse

---

<sup>3</sup>Bei einigen Tests werden hohe Spannungen von den Geräten an den Ausgängen angelegt. Um für die Sicherheit des Anwenders Sorge zu tragen, werden die Ausgänge alle stromlos geschaltet. Surge-Pulse können eine Amplitude bis zu 4kV haben und unter Umständen den Anwender schwer verletzen oder sogar töten.

der Test erbracht hat. In diesem Dialog wird vom Nutzer ein Kommentar zu dem Test verfasst, und die Parameter des Tests können hier noch einmal bearbeitet werden. Hier wird auch vermerkt, welches Bewertungskriterium von dem Prüfling in dem entsprechenden Test erreicht wurde. Der Anwender hat nun noch mehrere Optionen:

- a) Der Anwender hat die Möglichkeit, Bilder mit diesem Test zu verknüpfen. Er wird außerdem dazu aufgefordert, eine Bildunterschrift hinzuzufügen. Dies wird auch in einem Dialog abgewickelt, welcher ebenfalls durch den Anwender abgebrochen werden kann.
- b) Wenn der Dialog vom Anwender geschlossen wird, werden die erfassten Daten des Tests nicht in der Datenstruktur festgeschrieben.
- c) Mit der Bestätigung des Dialogs werden die gesammelten Daten abgespeichert und Der Anwender wird wieder ins Hauptmenü weitergeleitet.

# 6 Entwurf

Es werden zwei Programme entwickelt: Das Hauptprogramm in LabVIEW zur Durchführung von Tests sowie ein Programm in Java, um aus den Daten einen Prüfbericht im PDF-Format zu erstellen. Beide Programme haben gemein, dass sie dieselbe Datenstruktur implementiert haben. Der Datentransfer wird über eine XML-Datei vollzogen, welche an das Java-Programm als Eingabeparameter übergeben wird. Vorteilhaft ist, dass durch die Trennung der beiden Programme eine Weiterentwicklung unabhängig voneinander geschehen kann, da die Schnittstelle klar definiert ist. Im Zuge einer Prüfung wird es mitunter notwendig, die Temperatur und die Luftfeuchtigkeit zu messen. Zu diesem Zweck wird eine Schaltung mit zwei Sensoren erstellt.

## 6.1 Datenstruktur

Die Datenstruktur ist objektorientiert organisiert. Es werden mehrere Klassen mit allen notwendigen Parametern erzeugt. In Abbildung 6.24 ist die Datenstruktur im Java-Programm zu sehen. In Abbildung 6.1 ist die Datenstruktur, wie sie in LabVIEW organisiert ist, dargestellt. Die Attribute sind mit dem Symbol eines Ordners und eines Mautschlüssels gekennzeichnet. Die Implementierung unterscheidet sich marginal. Die LabVIEW Implementierung verfügt über einige zusätzliche VIs (äquivalent zu Methoden), die das Anlegen von Daten vereinfacht. Außerdem sind einige Attribute mehr vorhanden, die den Zugriff erleichtern. Das Hauptanliegen der Datenstruktur ist es, alle relevanten Daten abzulegen und zwischen den beiden Programmen austauschen zu können. Im Listing 6.1 ist die XML-Struktur dargestellt. In der Hierarchie ganz oben steht die Report-Klasse. Sie repräsentiert den gesamten Prüfbericht und enthält das Datum der Prüfung, den Namen des Prüfers, die Artikelnummer der Baugruppe<sup>1</sup> und eine Menge Objekte der Klasse Test. Die Klasse Test<sup>2</sup> beinhaltet die vorgenommenen Prüfungen (Segmente) innerhalb einer Norm. In ihr sind die Klimadaten, die Anzahl der Segmente und die ID (nur LV), sowie der Versuchsaufbau (Setup Klasse) und

---

<sup>1</sup>obnr (Orderbase-Nummer), in Anlehnung an das genutzte ERP-System Orderbase

<sup>2</sup>Diese Unterscheidung ist nicht ideal und wird in Zukunft noch abgeändert werden, so dass die jetzige Test-Klasse "Norm" heißen wird und die Klasse Segment dann "Test". So wird es zu weniger Verwirrungen kommen.

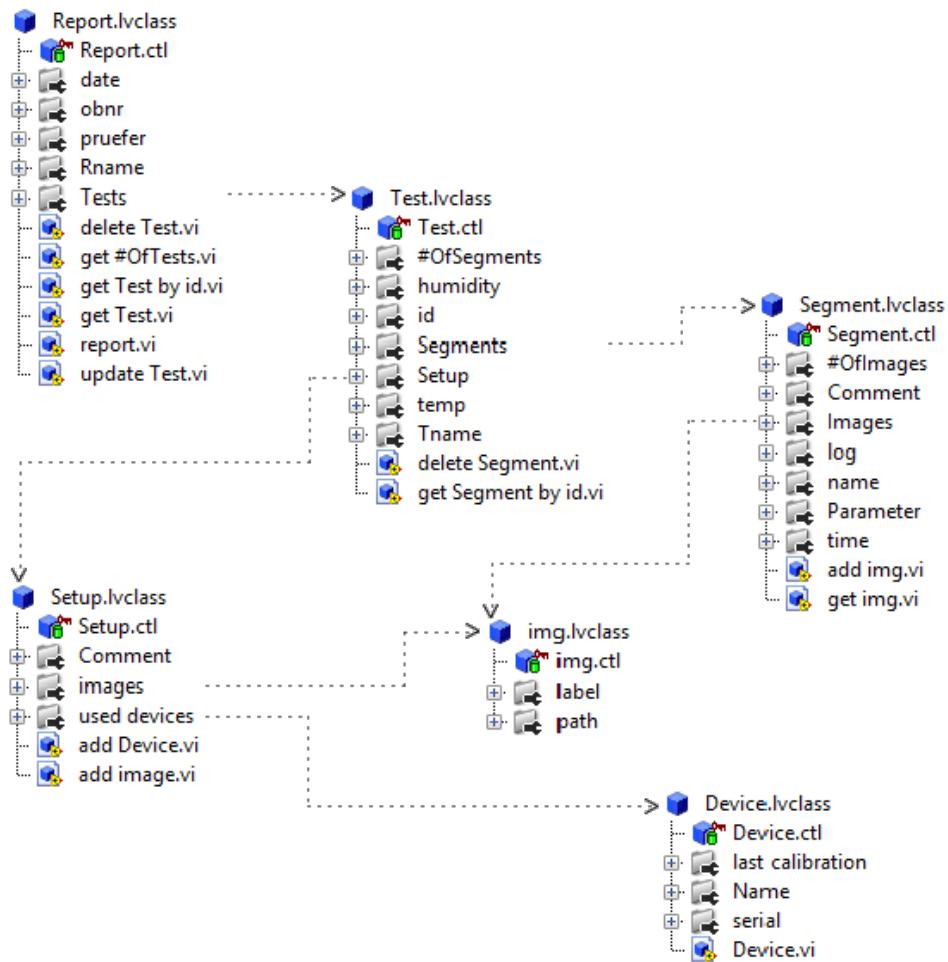


Abbildung 6.1: Klassendiagramm der Datenstruktur des LabVIEW-Programms

der Name des Tests enthalten.

Die Klasse Setup, die den Versuchsaufbau beschreibt, wird für jedes Test-Objekt einmalig angelegt. In ihr sind eine Beschreibung (Comment), Bilder und die genutzten Geräte (Klasse Device) abgelegt. Die Klasse Device stellt ein genutztes Gerät dar. Sie beinhaltet das Datum der letzten Kalibrierung, den Namen des Gerätes und die Seriennummer. Die Klasse Segment enthält die Anzahl der Bilder (nur LV), eine Beschreibung, Bilder, die komplette Kommunikation zwischen PC und Störgenerator als Log, der Name der Test-Sequenz, die Parameter der Test-Abschnitts als Tabelle sowie einen Zeitstempel. Die Klassen Segment und Setup können beide Bilder enthalten. Hierfür wurde die Klasse img kreiert, in welcher der Pfad zum Bild sowie die Bildunterschrift abgelegt wird.

Listing 6.1: Xml-Struktur

```
<?xml version="1.0" encoding="UTF-8" standalone="no" ?>
<Report date="" name="" obnr="" pruefer="">
    <Test humidity="" id="" name="" temp="">
        <Setup>
            <Comment></Comment>
            <img label="" path="" />
            <device lastcal="" name="" serial="" />
        </ Setup>
        <Segment name="" time="">
            <parameter name="" value0="" value1="" .... />
            <parameter name="" value0="" value1="" .... />
            <Comment></Comment>
            <img label="" path="" />
            <log></log>
        </ Segment>
    </ Test>
</ Report>
```

## 6.2 Hauptprogramm

Das Hauptprogramm verwaltet die Daten und ruft verschiedene Features nach Bedarf auf. Prüfdaten werden in einer Baumstruktur (Tree) abgelegt. Diese Repräsentation verdeutlicht die objektorientierte Datenstruktur und ist übersichtlich. Der Programmablauf ist in dem Zustandsautomatendiagramm in Abbildung 6.2 dargestellt.

Nachdem der Anwender Informationen über die zu testenden Baugruppen eingetragen hat (Enter Report Parameters) und die Initialisierung abgeschlossen ist, befindet er sich im Hauptmenü (siehe Abbildung 6.3 a)). Das Programm wartet nun, bis der Anwender eine Aktion macht. Zur Auswahl stehen die sechs verschiedenen Normen, nach denen Tests durchgeführt werden können. Wird eine dieser Normen gewählt, so erscheint, falls noch kein Test unter dieser Norm abgeschlossen wurde, ein Dialog, in dem der Versuchsaufbau eingegeben werden kann. Wird dieser Dialog bestätigt, wird im Subpanel rechts neben den Buttons das Test-Unterprogramm gestartet (siehe Abbildung 6.3 b)). So lange das Test-Unterprogramm im Subpanel läuft, werden die Buttons auf der linken Seite deaktiviert und ausgegraut. Sobald der Test beendet ist, wird der Anwender über einen Dialog aufgefordert, den Test zu dokumentieren (Protokoll). Wird dieser Dialog abgebrochen, werden die gesammelten Daten nicht abgespeichert und das Hauptfenster wird wieder hervorgeholt. Wenn der Dialog bestätigt wird, wird die Baumstruktur auf Grundlage der neuen Daten wiederholt aufgebaut.

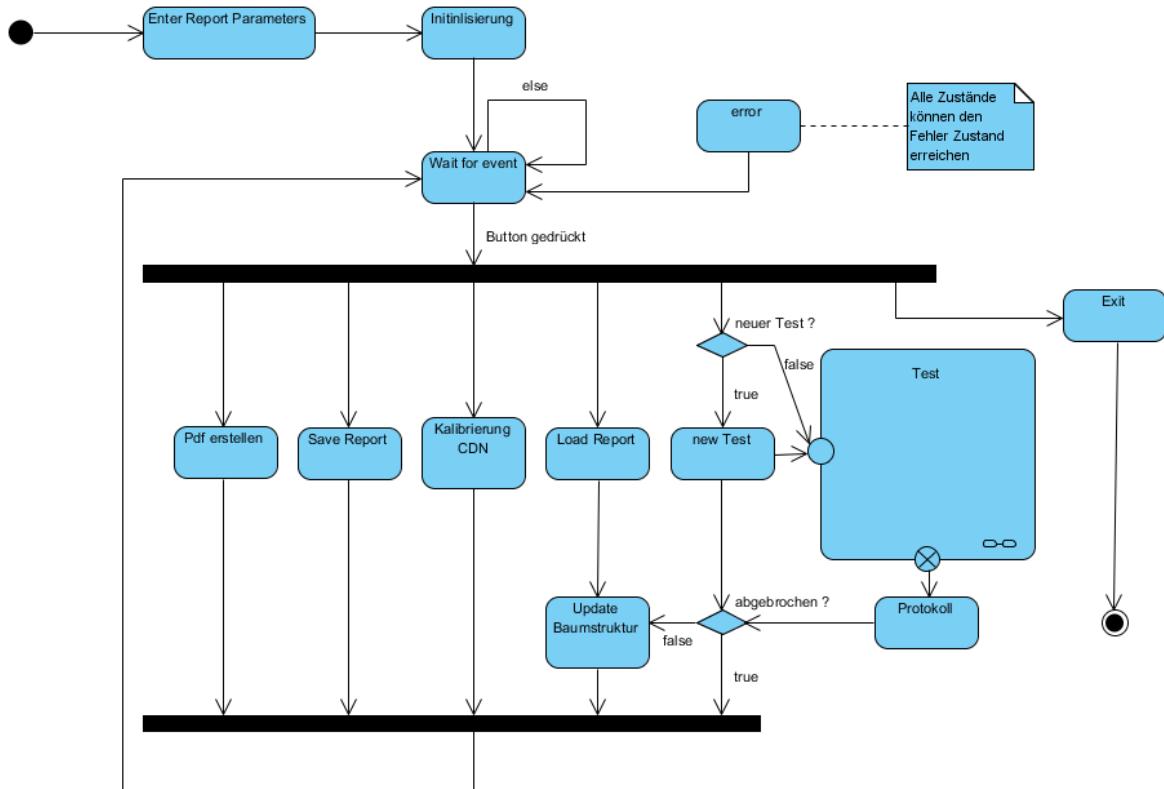


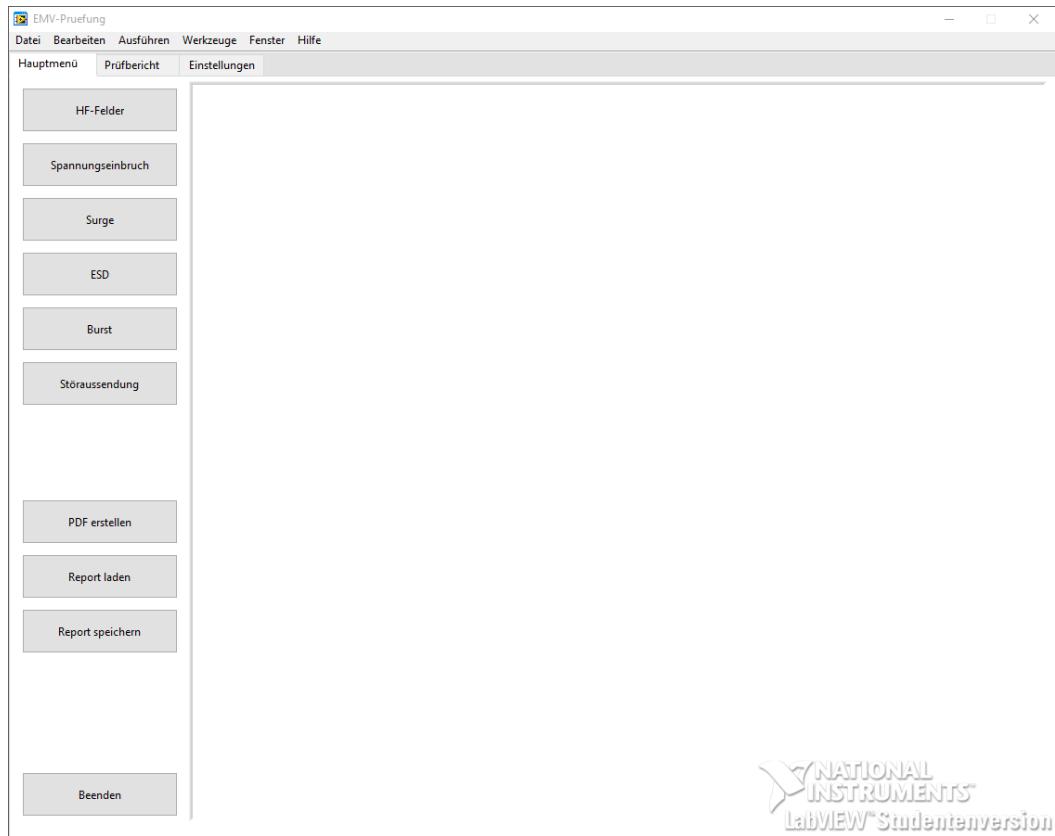
Abbildung 6.2: Zustandsdiagramm für das Hauptprogramm

Die Buttons “PDF erstellen”, “Report speichern” und “Kalibrierung von CDN”<sup>3</sup> wechseln kurz in den Zustand, in der die Funktionalität implementiert wurde, und gehen im Anschluss wieder in den Zustand “wait for event”. Wenn ein Prüfbericht geladen wird, ist es danach notwendig, die Änderung der Daten auch in der Baumstruktur zu berücksichtigen. Alle Zustände können direkt in den Fehler-Zustand “error” übergehen, wenn ein Fehler geschehen ist. Der Anwender bekommt dann eine Fehlermeldung als Dialog und startet wieder im Hauptmenü.

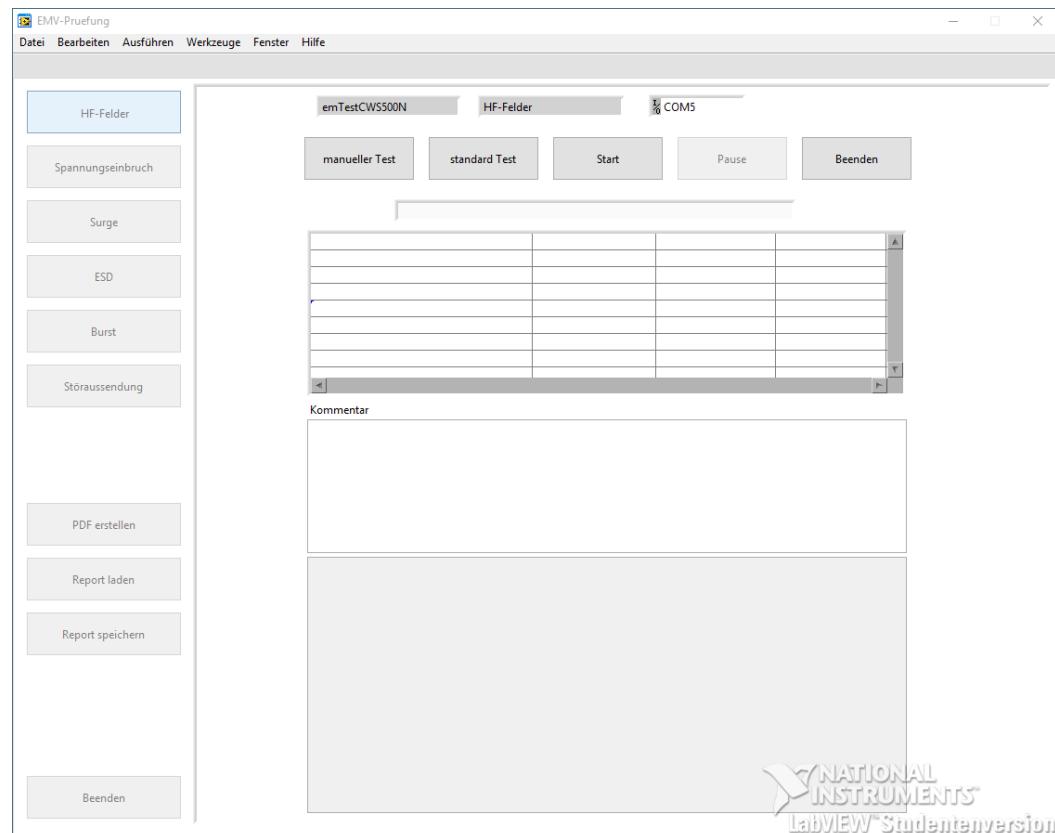
### 6.2.1 Test-Unterprogramm

Alle Tests basieren auf demselben Gerüst eines Zustandsautomaten, der einen standardisierten Ablauf der Kommunikation mit einem Gerät abbildet. In diesem Unterprogramm wird das Device-Interface und das Test-Interface genutzt. Während der Laufzeit wird entschieden, um welchen Test und um welches Gerät es sich handelt. Durch die Polymorphie [Lotz (2011)] können neue Tests und Geräte einfach implementiert werden. So ist lediglich das

<sup>3</sup>unter dem dritten Reiter zu finden, Abbildung 6.4

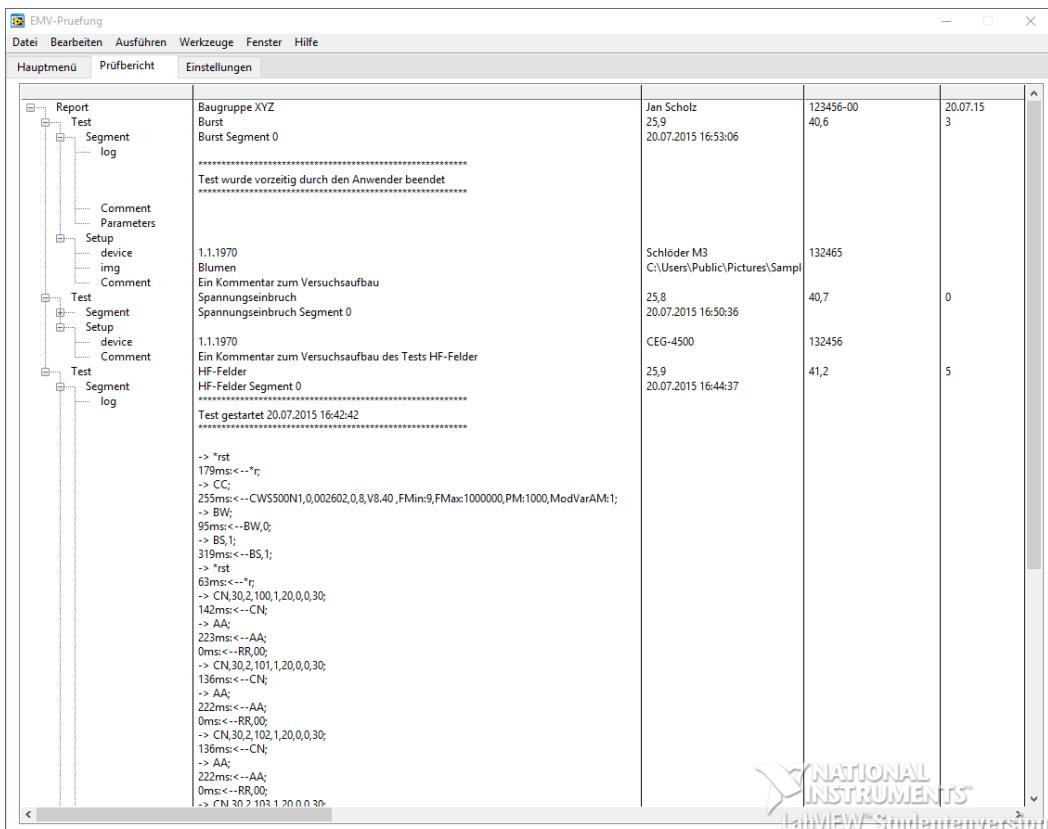


(a) Hauptmenü

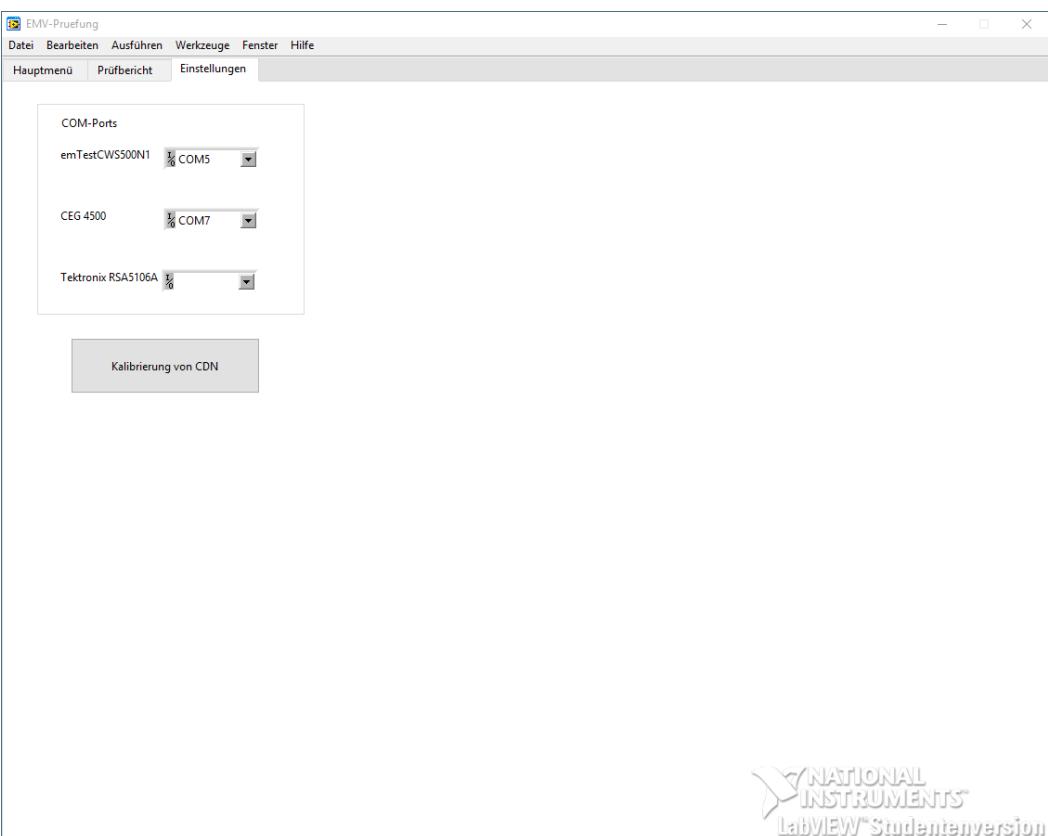


(b) Test-Unterprogramm im Subpanel

Abbildung 6.3: Hauptfenster I



(a) Datenhaltung zweiter Reiter



(b) Einstellungen dritter Reiter

Abbildung 6.4: Hauptfenster II

Device-Interface von der neuen Gerätekasse zu implementieren und das Test-Interface von dem Test, der mit dem Gerät durchgeführt werden soll.

Der Zustandsautomat ist in Abbildung 6.5 dargestellt. Im Zustand "Start" hat der Anwender die Möglichkeit, einen vorgefertigten Test zu laden, einen manuellen Test zu generieren, den Test zu starten (unter der Bedingung, dass ein Test gewählt wurde) und das Programm zu beenden. Nachdem ein Test gewählt und dieser gestartet wurde, werden lokale Variablen im Zustand "initialise" initialisiert. Im Anschluss wird die Verbindung mit dem Gerät aufgebaut und einige Male von dem Gerät gelesen, um sicherzustellen, dass der Puffer geleert ist. Nun wird jede Zeile der Kommandoliste abgearbeitet, wobei sich unterschiedliche Programmabläufe zeigen.

Im Zustand "evaluate line" wird bestimmt, ob der aktuelle Befehl eine Antwort nach sich zieht. Sollte dies nicht der Fall sein, wird - wenn nötig - gewartet und in den Zustand "check for pressed Buttons" gewechselt. Sollte kein Befehl mehr vorhanden sein, wird in den Zustand "close connection" gewechselt. Die andere Möglichkeit, wenn eine Antwort erwartet wird, ist ein Wechsel in den Zustand "wait for Response". Vom Gerät wird gelesen und - wenn nötig - eine Wartezeit nach dem Lesen absolviert. Wurde die Antwort erhalten, wird im Zustand "error catcher" die Antwort des Geräts überprüft. Sollte es sich um eine Fehlermeldung handeln, wird diese in einen LabVIEW-Fehler übersetzt. Ist die Antwort eine Rückgabe der Sollwerte, werden diese mit gesetzten Sollwerten verglichen; sind die Werte nicht identisch, wird ebenfalls ein Fehler generiert. Sollte kein Fehler generiert worden sein, so wird im nächsten Zustand entschieden, ob nochmals gelesen werden soll. Falls eine weitere Nachricht erwartet wird, wird in den Zustand "wait for Response" zurückgesprungen. Sollte keine weitere Nachricht erwartet werden, wird im Zustand "check for pressed Buttons" der Zustand der Buttons erfasst. Sollte kein Button gedrückt worden sein, wird die nächste Zeile der Kommandoliste bearbeitet.

Hat der Anwender Pause gedrückt, wird der "Pause" Zustand so lange gehalten, bis entweder Beenden oder Pause erneut gedrückt wurde. Wird Pause erneut gedrückt, wird die nächste Zeile begutachtet.

Wird Beenden gedrückt, wird wie im Zustand "check for pressed Buttons" in den Zustand "close connection" überführt. Hier wird das Gerät in einen sicheren Zustand versetzt und die Verbindung mit dem Gerät getrennt. Dies ist wichtig, da alle Zustände abgesehen von diesem, wenn ein Fehler auftritt, in den Zustand "error" wechseln und das Gerät möglicherweise noch eine Störspannung am Ausgang anliegen hat. Der "end"-Zustand beendet das Test-Unterprogramm.

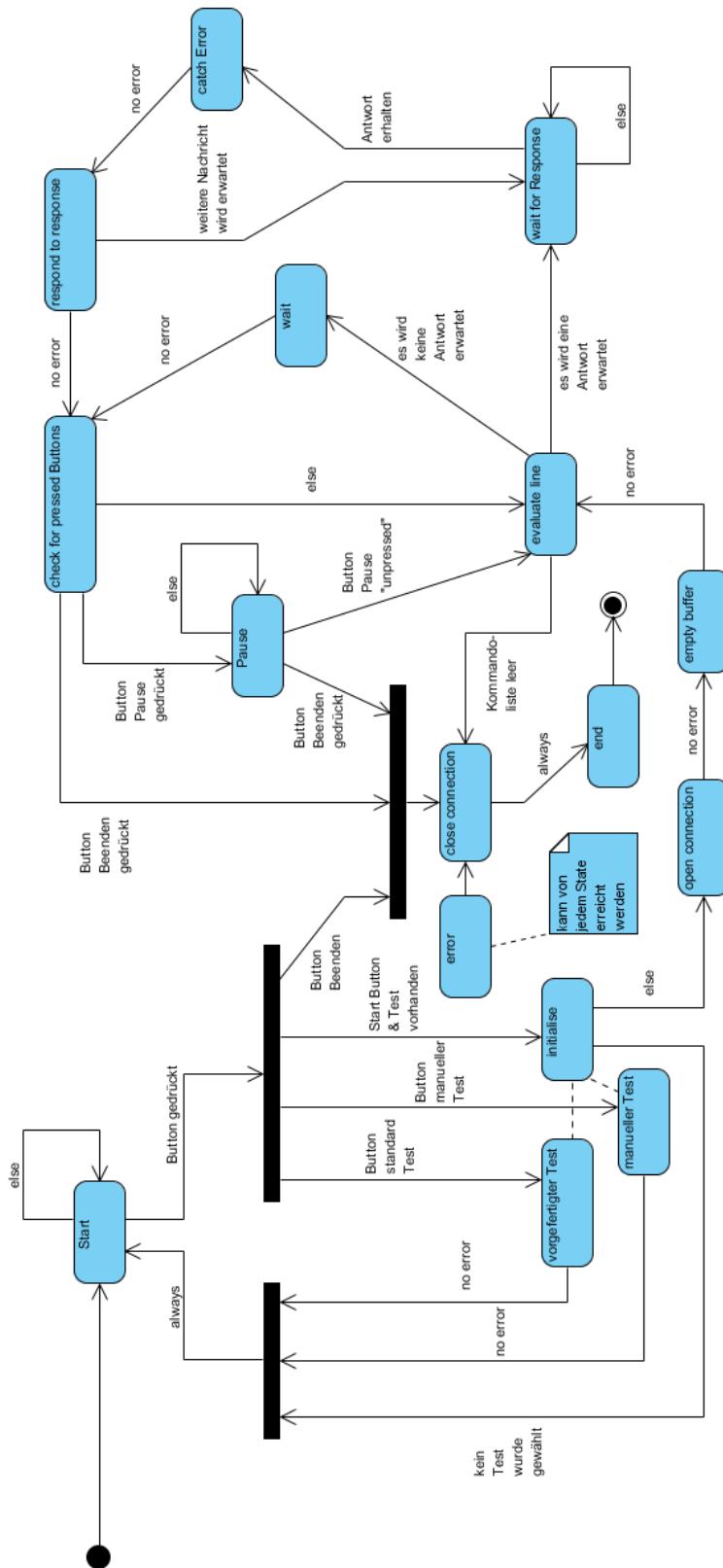
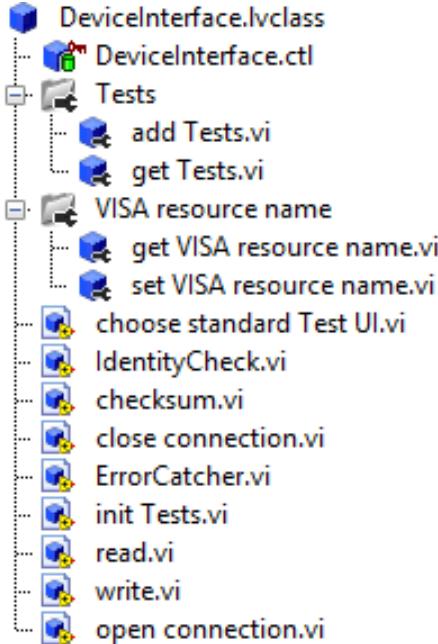


Abbildung 6.5: Test-Unterprogramm Statechart

### 6.2.2 Device-Interface



Es wurde ein Interface für den Einsatz von Geräten entworfen. Dieses Interface standardisiert die Kommunikation mit einem Gerät und deckt alle notwendigen Interaktionen ab, die während eines Tests ablaufen. Ein Device verfügt über einen VISA-Resourcenamen (siehe Kapitel 3.2), der eine Adresse darstellt und die Ansprache des Geräts bereits maßgeblich vereinfacht, da der VISA-Standard eine Palette von einfachen Funktionen bereithält. Ein Device kann außerdem über mehrere Tests verfügen, die im Laufe des Prüfprozesses erstellt werden, sowie über bereits abgelegte Norm-Tests. Alle VIs, die nicht zu den Gettern oder Settern gehören, und "choose standard Test UI" müssen von den erben-den Klassen/Geräten neu implementiert werden.

Abbildung 6.6: Device-Interface  
Übersicht

Beide Geräte nutzen eine Prüfsumme, um die Datenintegrität zu gewährleisten. Sollte ein Gerät keine Prüfsumme benötigen, kann dieses VI den zu sendenden String einfach durchschleifen. In diesem VI wird außerdem das Endzeichen des Kommandos angehängt. Weitere Details sind in Kapitel 7.2 und 7.3 zu finden.

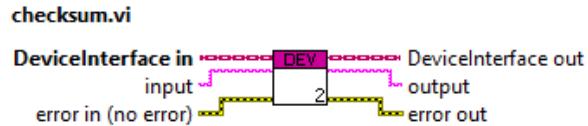
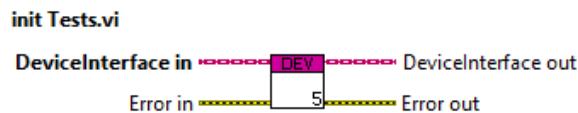


Abbildung 6.7: checksum



In diesem VI sind die Norm-Tests fest codiert.

Abbildung 6.8: init Tests

Dieses VI liest die hinterlegten Tests aus und öffnet einen Dialog, der dem Anwender die Möglichkeit gibt, die Tests auszuwählen. Da bei einem Gerät durchaus auch Tests von verschiedenen Normen hinterlegt werden können, wird hier während der Laufzeit geprüft, welcher Test (bestimmte Norm) anliegt. Lediglich die Tests der gleichen Norm werden angezeigt.

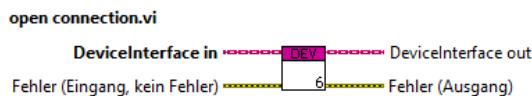


Abbildung 6.10: open connection

Mit diesem VI wird festgestellt, ob es sich um das Gerät handelt, das an der Adresse vermutet wird. Die Geräte sind beide in letzter Instanz auf einem Virtuellen COM-Port erreichbar. Wenn die USB-Kabel abgezogen und an anderen Ports angesteckt werden, ändern sich auch die COM-Ports.

Dem Gerät wird ein Kommando geschickt, auf das es den Gerätetyp und die Seriennummer wiedergibt. Falls so ein Kommando nicht vorhanden ist, wird ein anderes Kommando genutzt, worauf die Antwort des Geräts bekannt ist. Ist die Antwort des Geräts nicht wie erwartet, wird ein Fehler generiert.

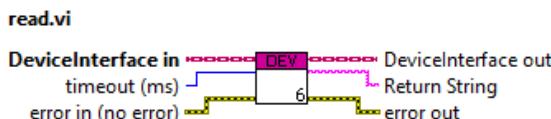


Abbildung 6.12: read

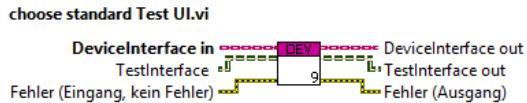


Abbildung 6.9: Choose standard Test UI

Alle notwendigen Einstellungen, wie beispielsweise die Übertragungsrate, Wortbreite, Anzahl der Paritätsbits und andere Parameter, die im Zusammenhang mit der Übertragung stehen, können hier festgeschrieben werden.

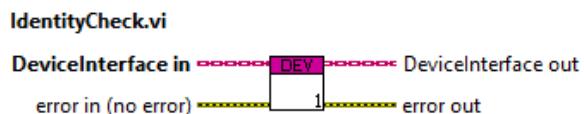


Abbildung 6.11: Identity Check

Dieses VI wird so für jedes Device implementiert, dass gesendete Nachrichten des Geräts vollständig empfangen werden können. Die Geräte arbeiten unterschiedlich. So kann durch unterschiedliche Implementierung Zeit gespart werden. Ebenfalls kann sichergestellt werden, dass wirklich der gesamte Puffer geleert wurde.

Die Implementierung dieses VIs ist trivial und berücksichtigt, dass in Zukunft unter Umständen Geräte hinzukommen, die eine besondere Implementierung zum Senden von Nachrichten bedürfen.

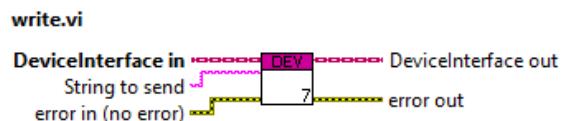


Abbildung 6.13: write

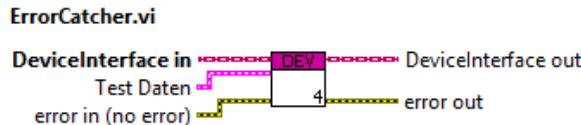


Abbildung 6.14: Error Catcher

Die Geräte geben Fehlermeldungen zurück, diese können mittels dieses VIs in Fehler in LabVIEW umgewandelt werden. Dies ist zum einen wegen der Unfallverhütung sicherheitsrelevant und zum andern notwendig, damit der Informationsfluss weitergeleitet wird und Fehler behoben werden können.

Das VI kann aber noch mehr: Wenn eine Abfrage der gesetzten Sollwerte während des Programmablaufs erfolgt, wird hier verglichen, ob die gesetzten Sollwerte auch tatsächlich vom Gerät übernommen wurden. Ist dies nicht der Fall, wird auch hier ein Fehler generiert.

Wenn der Test abgeschlossen ist, ein Fehler auftritt oder der Anwender den Test vorzeitig beendet, wird die Verbindung mit diesem VI getrennt. Hier ist darauf zu achten, dass das Gerät in einen sicheren Zustand wechselt. Es darf keine Spannung mehr an den Ausgängen anliegen oder zu einem späteren Zeitpunkt automatisch ein Störsignal geschaltet werden.

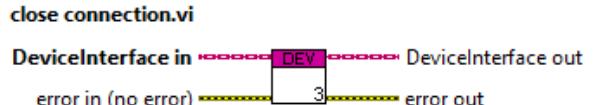
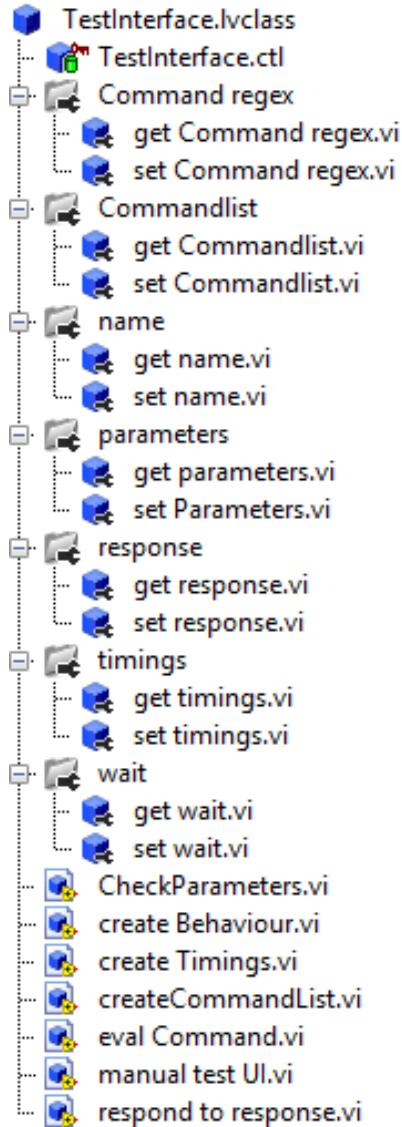


Abbildung 6.15: close connection

### 6.2.3 Test-Interface



Um Tests mit den Störgeneratoren durchzuführen, müssen Kommandos für das Gerät generiert werden und die Art der Kommunikation weiter spezifiziert werden. So können zwei Tests mit dem selben Gerät unterschiedliche Interaktionen zwischen dem PC und dem Gerät zur Folge haben. Die Abbildung 6.16 zeigt den Aufbau des Interfaces, welches an jeden Test vererbt wurde. Es wurden einige Attribute mit Gettern und Settern angelegt und grundlegende Aufgaben wurden mittels VI's hinterlegt.

Command regex ist ein Array aus Strings. Response ist ein boolsches Array. Timings und wait sind Numeric Arrays. Diese Arrays beschreiben das Verhalten bei besonderen Befehlen. So kann ein regulärer Ausdruck hinterlegt werden. Falls dieser Ausdruck bei einem Kommando gefunden wird, werden die Wertigkeiten der anderen Arrays berücksichtigt und es kann ein spezielles Verhalten für diesen Befehl erfolgen. Alle VIs haben gemein, dass sie das Test-Objekt übergeben bekommen und nach der Verarbeitung wieder ausgeben.

Abbildung 6.16: Test-Interface

Dieses VI wird als Dialog genutzt, um einen manuellen Test zu erstellen. Die vom Anwender getätigten Einstellungen werden dann als Parameter in numerischer Form und als zweidimensionales Array (Tabelle) im String-Format erstellt. Es wird außerdem ausgegeben, ob der Anwender den Dialog abgebrochen hat.

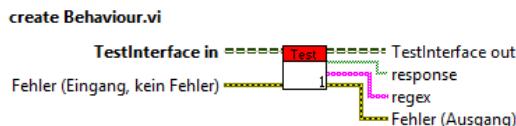


Abbildung 6.18: create Behaviour

Die Parameter aus einem manuell generierten Test werden hier nochmals überprüft. Sollte einer oder mehrere Parameter nicht von dem Gerät verarbeitet werden können, so wird hier ein Fehler generiert. Zur Vermeidung von falschen Nutzereingaben gibt es weitere Details im Kapitel 7.1.

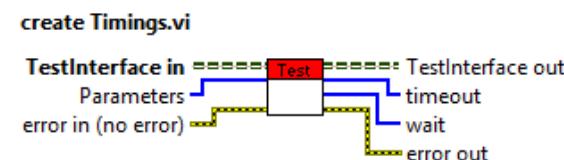


Abbildung 6.20: create Timings

das Gerät erst, nachdem der Befehl vollständig verarbeitet wurde, einen neuen Befehl erhalten soll. Die Timings werden aus den Parametern generiert und sind so den manuellen Tests, die während der Laufzeit generiert werden, angepasst.

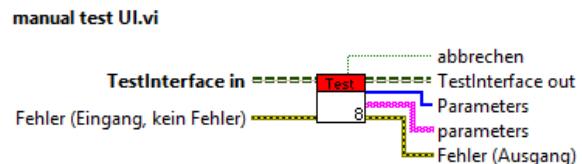


Abbildung 6.17: manual test UI

In diesem VI werden die beiden Arrays `regex` und `response` für das Verhalten festgelegt. Das Array `response` definiert, ob eine Antwort auf einen Befehl erwartet wird oder nicht. Die Erstellung des Verhaltens ist hardcoded und wird nicht während der Laufzeit erstellt.

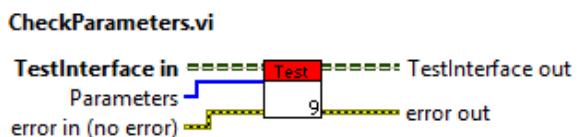


Abbildung 6.19: Check Parameters

Hier wird der zweite Teil des besonderen Verhaltens erstellt. Es handelt sich zum einen darum, wie lange auf eine Antwort gewartet werden soll (wenn eine Antwort erwartet wird), und zum anderen, ob im Anschluss gewartet werden soll. Dieses Verhalten ist notwendig, da teilweise lange Wartezeiten nach einem Befehl zu erwarten sind, in der

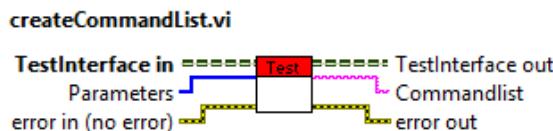


Abbildung 6.21: create Commandlist

In diesem VI wird beurteilt, wie auf den aktuellen Befehl reagiert werden soll. Wenn der aktuelle Befehl auf einen der hinterlegten regulären Ausdrücke passt, wird ein angepasstes Verhalten ausgeführt.

Aus den vom User spezifizierten Parametern wird eine Liste von Befehlen zur Ansprache des Geräts generiert. Der komplette Test wird hier in Form eines Strings (Commandlist) erstellt.

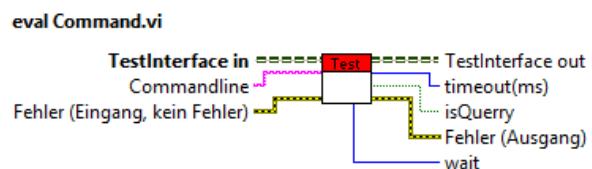


Abbildung 6.22: eval Command

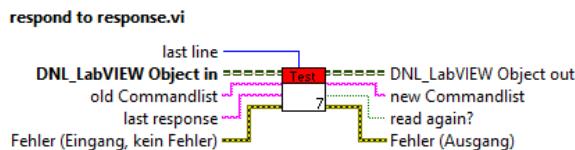


Abbildung 6.23: respond to response

mit dem aktuellen Test fertig ist. Sollte das Gerät dies verneinen, kann nach einer Weile eine neue Anfrage gestellt werden. Dies wird so lange wiederholt, bis der Test abgeschlossen ist.

In diesem VI wird auch der aktuelle Befehl inspiziert. Allerdings wird zusätzlich die empfangene Antwort angeschaut. Sollte die Antwort nicht der Erwartung entsprechen, kann auf den Programmablauf Einfluss genommen werden, indem die Kommandoliste verändert wird. Dies ist besonders sinnvoll, wenn ein Gerät gefragt wird, ob es bereits

## 6.3 PDF-Prüfbericht

Das Java-Programm besteht aus drei größeren Teilen: der Datenstruktur, dem XML-Parsing und der Erstellung der PDF-Datei. Die Datenstruktur ist fast identisch zu der, die im LabVIEW-Programm implementiert ist. Alle notwendigen Informationen, die in den Prüfbericht einfließen müssen, können hier abgelegt werden. In Abbildung 6.24 ist das Klassendiagramm des Programms dargestellt. Es wurde darauf verzichtet, die Klasse ContentParser darzustellen, da das Klassendiagramm sonst sehr unübersichtlich werden würde. Die Klasse besitzt viele Objekte der Datenstruktur, die angelegt werden, während das Report-Objekt

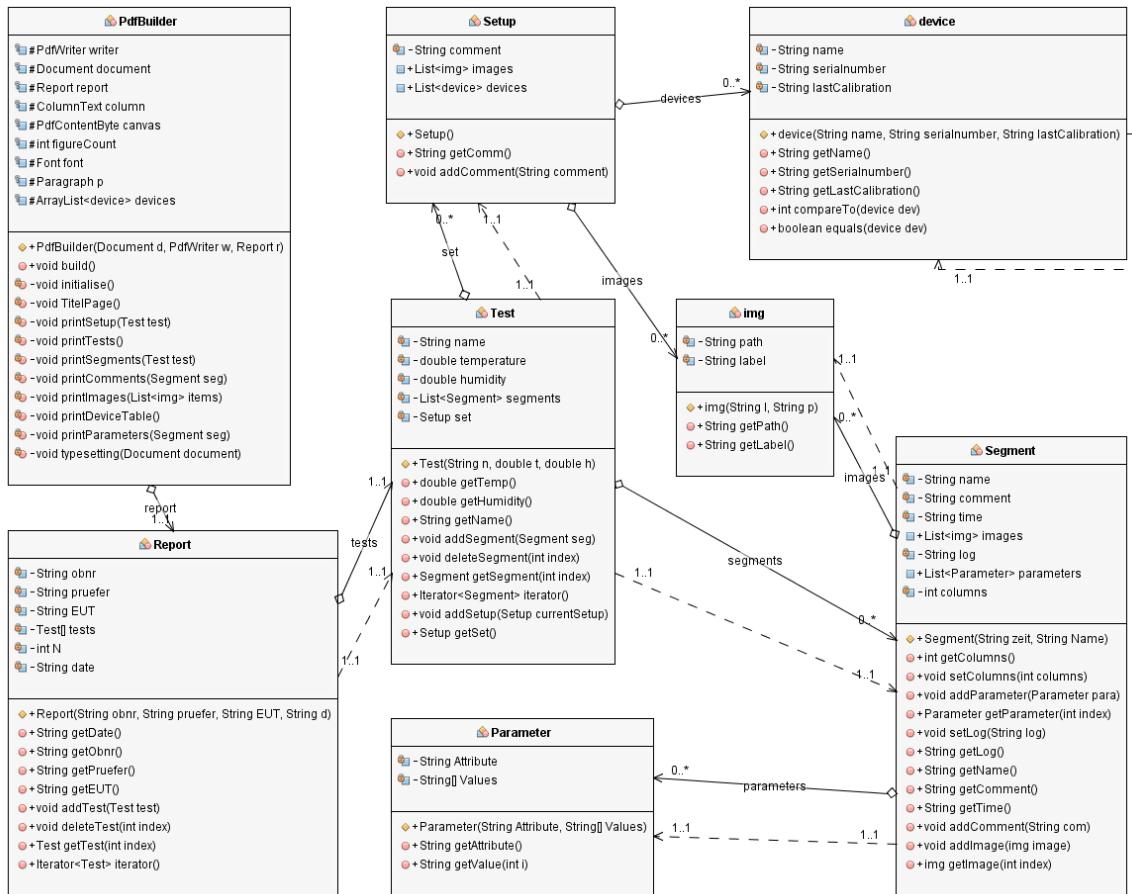


Abbildung 6.24: Klassendiagramm des Java-Programms

zusammengestellt wird. Die Klassen PdfPruefbericht (enthält die Main-Methode), Footer-Header, StructureParser und util wurden ebenso nicht dargestellt, da keine Instanzen von ihnen bei anderen Klassen existieren und anderen Klassen auch keine Instanzen von ihnen innehaben. Das Programm bekommt den Pfad einer XML-Datei und den Pfad für die Ausgabe der PDF-Datei als Parameter übergeben.

### 6.3.1 XML-Parsing

Es wird die SAX (Simple API for XML) benutzt, diese freie Bibliothek ist laut [saxproject.org](http://saxproject.org) der Standard für XML-Parsing. Um die XML-Datei zu lesen, muss ein eigener Handler geschrieben werden, da die Struktur der XML-Datei nicht bekannt ist<sup>4</sup>. Mittels der Methode

<sup>4</sup>Lowagie (2011), Vgl. S.312

startElement() können Attribute eines Elements ausgelesen werden. Die Methode endElement() wird genutzt, um den Inhalt eines Elements zu lesen. Es wurde eine eigene Klasse ContentParser erstellt und die beiden Methoden wurden neu implementiert. In der Methode startElement() wird der Name des Elements angeschaut und die Attribute werden dann in die Datenstruktur eingepflegt. Das gleiche geschieht bei endElement(). Dort wird der Inhalt des Elements festgeschrieben, wenn es sich um einen String handelt. Die Klasse verfügt über ein Report-Objekt, welches die gesamten Daten innehaltet, dieses Objekt wird dann an die Klasse Pdfbuilder übergeben, von der schließlich die PDF-Datei erstellt wird.

### 6.3.2 Pdfbuilder

Ein Titelblatt wird mit dem Namen des Prüfers, dem Datum der Prüfung, dem Namen und der Artikelnummer der Baugruppe erstellt. Für jede Norm, nach der getestet wurde, wird eine neue Seite angefangen. Ein Test beginnt mit der Überschrift, danach werden Temperatur und Luftfeuchtigkeit genannt. Im Anschluss folgen Versuchsaufbau und die einzelnen "Segmente" (Unterabschnitte eines Tests). Segmente verfügen über Parameter, die in Form einer Tabelle dargestellt werden. Das Framework achtet automatisch darauf, dass kein Objekt, wie zum Beispiel Text, ein Bild oder eine Tabelle, die vordefinierten Grenzen überschreitet. Es wird so lange auf das Dokument geschrieben, bis die Seite voll ist. In der Methode typesetting() wird, wenn noch mehr Elemente, die gedruckt werden müssen, vorhanden sind, eine neue Seite öffnet und es können weiter Elemente in das Dokument eingefügt werden. Wenn im Versuchsaufbau oder Segment Bilder vorhanden sind, ist darauf zu achten, dass diese nicht von der Bildunterschrift getrennt werden, es wird also ausgelesen wie viel Platz noch bis zur Grenze vorhanden ist, bevor das Bild gesetzt wird. Wenn die Summe von Text und Bild passt, wird beides platziert. Ist dies nicht der Fall, wird eine neue Seite erstellt. Alle verwendeten Geräte, die bei den Versuchsaufbauten erwähnt wurden, werden zum Schluss in einer Tabelle dargestellt. Im Anhang auf der CD im Ordner "pdf\_gen\_test" ist ein Beispiel vorbereitet, das mit der Datei start.bat gestartet werden kann.

## 6.4 Schaltung für die Klimadatenerfassung

Im Rahmen dieser Arbeit wurde eine Schaltung zur Messung der Klimadaten erstellt. Die Sensor-Bauteile werden kurz beschrieben. Die Bauteile haben beide einen veränderlichen Widerstand und die Messung findet in Form einer Spannungsmessung statt. Aus der gemessenen Spannung kann dann die Temperatur bzw. die Luftfeuchtigkeit berechnet werden. Der Aufbau der Baugruppe ist recht einfach gehalten (siehe Abbildung 6.25). Die beiden Sensorsbauteile sind jeweils mit einem Widerstand in Reihe und es wird jeweils von PAD1 und PAD4 eine Spannung auf PAD3 (Ground) angelegt. Gemessen wird von PAD2 und PAD4

auf PAD3. Die Spannung über dem in Reihe befindlichen Widerstand wird dann von der Gesamtspannung abgezogen. So wird die Spannung über dem Sensorbauteil erhalten.

#### 6.4.1 NTC Thermistor TTC05104JSY

Das Bauteil ist ein NTC-Widerstand. Das bedeutet, dass mit steigender Temperatur der Widerstand des Bauteils kleiner wird. Der Widerstand bei 25°C liegt bei 100kΩ. Der Hersteller garantiert eine maximale Abweichung von 5% für den Widerstandswert bei einer Temperatur von 25°C. Aus der Tabelle im Datenblatt ist zu erkennen, dass das Bauteil bei 10°C einen Widerstand von ca. 110kΩ und bei 30°C ca. 80kΩ hat. So ist also ein Widerstand in Reihe zu diesem zu wählen, der uns durch größere Spannungsschwankungen eine gute Auflösung liefert. Der Spannungsteiler errechnet sich wie folgt:

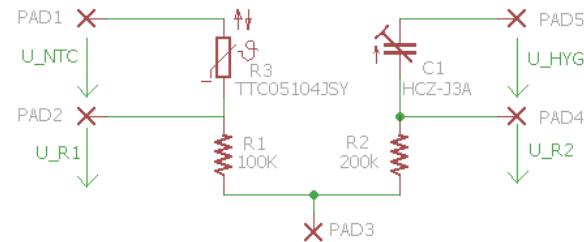


Abbildung 6.25: Schaltplan der Baugruppe

$$\frac{R_{NTC}}{R_1} = \frac{U_{NTC}}{U_1}$$

mit

$$U_1 = U_G - U_{NTC}$$

gilt dann weiter

$$U_{NTC} = \frac{R_{NTC} \cdot U_G}{R_1 + R_{NTC}}$$

Die beste Auflösung ist erreicht, wenn  $\Delta U_{NTC10-30}$  maximal ist für ein gegebenes  $R_1$

$$\Delta U_{NTC10-30} = \frac{R_{NTC10} \cdot U_G}{R_1 + R_{NTC10}} - \frac{R_{NTC30} \cdot U_G}{R_1 + R_{NTC30}}$$

$$\Delta U_{NTC10-30} = \frac{R_{NTC10} \cdot U_G(R_1 + R_{NTC30}) - R_{NTC30} \cdot U_G(R_1 + R_{NTC10})}{(R_1 + R_{NTC10})(R_1 + R_{NTC30})}$$

$$\Delta U_{NTC10-30} = \frac{U_G \cdot R_1(R_{NTC10} - R_{NTC30})}{R_1^2 + R_1 \cdot R_{NTC10} + R_1 \cdot R_{NTC30} + R_{NTC10} \cdot R_{NTC30}}$$

$$\Delta U_{NTC10-30} = \frac{U_G \cdot (R_{NTC10} - R_{NTC30})}{R_1 + R_{NTC10} + R_{NTC30} + \frac{R_{NTC10} \cdot R_{NTC30}}{R_1}}$$

Da  $R_1$  nicht im Zähler steht, wird  $\Delta U_{NTC10-30}$  maximal, wenn der Nenner minimal ist.  
Extremwerte des Terms werden durch das Nullsetzen der ersten Ableitung erhalten:

$$1 + \frac{R_1 - R_{NTC10} \cdot R_{NTC30}}{R_1^2} = 0$$

$$-R_1^2 = R_1 - R_{NTC10} \cdot R_{NTC30}$$

$$R_1^2 + R_1 + \frac{1}{4} = R_{NTC10} \cdot R_{NTC30} + \frac{1}{4}$$

$$R_1 = \sqrt{R_{NTC10} \cdot R_{NTC30} + \frac{1}{4}} - \frac{1}{2}$$

Die Gleichung kann verallgemeinert werden und für große Widerstände fallen die Konstanten weg:

$$R_{ref} = \sqrt{R_{mess,min} \cdot R_{mess,max}} \quad (6.1)$$

$$R_1 = \sqrt{110k\Omega \cdot 80k\Omega} \approx 94k\Omega$$

Um zu belegen, dass es sich auch tatsächlich um ein Minimum handelt, wird die zweite Ableitung errechnet:

$$\frac{R_1 - 2(R_1 - R_{NTC10} \cdot R_{NTC30})}{R_1^2} \quad (6.2)$$

Das Ergebnis wird anschließend eingesetzt.

$$\frac{94k\Omega - 2(94k\Omega - 110k\Omega \cdot 80k\Omega)}{94k\Omega^3} \approx 2$$

Das Ergebnis ist positiv. Somit handelt es sich um ein Minimum.

Der ideale Widerstand für die Messung liegt also bei ca.  $94k\Omega$ . Die Spannung ist so hoch wie möglich zu wählen, damit ein Fehler durch Rauschen so klein wie möglich gehalten wird. Die Begrenzung für die Spannung ist durch die maximale Leistung des Bauteils vorgegeben. Es gilt  $P_{max} = 0,45W$ . Am besten wird als Anhaltspunkt der maximale Strom gewählt, der durch das Bauteil fließen darf. Da vorsichtig kalkuliert wird, wird der Widerstandswert bei einer Temperatur von  $10^\circ\text{C}$  betrachtet, dessen Widerstand dann bei ca.  $110k\Omega$  liegt. Für diesen Extremfall (der wahrscheinlich nicht eintreffen wird) lässt sich nun der maximale Strom berechnen, der sich ergibt zu:

$$\sqrt{\frac{P_{max}}{R_{NTC,max}}} = I_{max} \quad (6.3)$$

$$I_{max} = \sqrt{\frac{0,45W}{110k\Omega}} \approx 2mA$$

Zusammen mit  $R_1$  ergibt sich folgende Rechnung für die maximale Spannung:

$$U_{NTC,max} = R_{NTC,max} \cdot I_{max} = 110k\Omega \cdot 2mA \approx 220V$$

$$U_{G,max} = U_{NTC,max} + U_{R1}$$

$$U_{G,max} = U_{NTC,max} + \frac{R_1 \cdot U_{NTC,max}}{R_{NTC}}$$

$$U_{G,max} = 220V + \frac{100k\Omega \cdot 220V}{110k\Omega} = 420V$$

Das Gerät NI 9264 kann allerdings nur maximal 10V ausgeben, deshalb ist die Messung auf eine Spannung von 10V beschränkt. Es ist aber auch sichergestellt, dass die Bauteile problemlos die an sie angelegte Leistung verarbeiten können.

### 6.4.2 HCZ-J3A HUMIDITY SENSOR

Dieser Luftfeuchtigkeitssensor wurde gewählt, da mit ihm und der bereits vorhandene Hardware von NI (NI 9221 und NI 9264) die kostengünstigste Umsetzung möglich war. Das Bauteil wird mit einer Sinusspannung der Frequenz 1 kHz betrieben und verringert seinen Widerstand bei steigender Luftfeuchtigkeit. Der maximale Widerstand, der erwartet wird, liegt bei ca.  $4,6\text{M}\Omega$  (für  $15^\circ\text{C}$  / 25% Luftfeuchtigkeit). Der minimal zu erwartende Widerstandswert liegt bei  $8\text{k}\Omega$  (für  $30^\circ\text{C}$  / 70% Luftfeuchtigkeit); die Werte sind der Tabelle 6.1 entnommen. In Analogie zum NTC-Widerstand lässt sich auch hier die Gleichung 6.1 anwenden.

$$R_2 = \sqrt{4,6\text{M}\Omega \cdot 8\text{k}\Omega + \frac{1}{4} - \frac{1}{2}} \approx 192\text{k}\Omega$$

Es ist weiter zu belegen, dass es sich um ein Minimum handelt. Gleichung 6.2 kann wieder angewendet werden

$$\frac{192\text{k}\Omega - 2(192\text{k}\Omega - 4,6\text{M}\Omega \cdot 8\text{k}\Omega)}{192\text{k}\Omega^3} \approx 10^{-5}$$

Das Ergebnis ist größer als 0. Somit handelt es sich auch um ein Minimum.

Die Frage nach der anzulegenden Spannung stellt sich bei diesem Bauteil nicht, da laut Datenblatt nur eine Spannung bis maximal 1V Effektivwert zulässig ist, um belastbare Messungen zu erhalten. Da der größte zu erwartende Widerstandswert des Luftfeuchtigkeitssensors bei ca.  $4,6\text{M}\Omega$  liegt, kann eine Messung in diesem Bereich sehr ungenau werden. Es muss geprüft werden, welche klimatischen Bedingungen durchschnittlich in dem Raum herrschen, in dem die Prüfung durchgeführt wird. In diesem Raum gibt es eine Klimaanlage, dadurch ist es möglich, dass die Luftfeuchtigkeit geringer ist als in nicht klimatisierten Räumen. Sollte dies der Fall sein, muss der Widerstand nach oben hin angepasst werden, damit die Auflösung der Messung besser wird.

$\varphi / T$	5°C	10°C	15°C	20°C	25°C	30°C	35°C	40°C	45°C	50°C
25%	9900	6900	4600	3200	2300	1700	1100	920	600	520
30%	4400	3100	2000	1500	920	770	530	430	280	250
35%	1900	1400	900	670	450	360	250	210	140	114
40%	810	600	430	310	220	170	130	96	74	61
45%	420	300	220	160	120	90	71	55	43	35
50%	211	150	110	83	66	51	40	31	25	20
55%	109	83	62	48	37	29	23	19	15	13
60%	63	48	37	29	23	18	15	12	10.3	8.7
65%	37	28	22	18	14	12	10	8.1	6.9	5.9
70%	22	17	14	12	9.6	8.0	6.8	5.8	4.9	4.3
75%	14	12	9.4	7.8	6.5	5.5	4.7	4.1	3.4	3.0
80%	9.0	7.3	6.0	5.0	4.2	3.8	3.3	2.9	2.4	2.0
85%	6.0	4.8	3.9	3.3	2.8	2.5	2.2	2.0	1.7	1.4
90%	4.0	3.2	2.7	2.2	1.9	1.7	1.5	1.4	1.2	1.1

Tabelle 6.1: Widerstandstabelle relative Feuchtigkeit / Temperatur in k $\Omega$  [HCZ,S.6]

# 7 Realisierung

Die Entwicklungsumgebung LabVIEW scheint auf den ersten Blick recht einfach bedienbar. Sie unterscheidet sich jedoch grundlegend von textbasierten Programmiersprachen wie C oder Java dadurch, dass sie auf Arrays aufbaut. Das Erarbeiten einer gewissen Routine im Umgang mit diesen Strukturen ist daher notwendig.

Zuerst wurde die Baugruppe zur Messung der Klimadaten gefertigt, über das NI Compact-Daqmx angebunden und mittels VIs in LabVIEW implementiert.

Die Entwicklung wurde zum Teil Bottom-up und zum Teil Top-down vollzogen. Das Hauptprogramm wurde im ersten Entwurf als leere Hülle mit seinen Zuständen entwickelt. Da die Anforderungen an das Hauptprogramm von Anfang an bekannt waren, konnte bereits zu Beginn ein Konzept entwickelt werden, das sich im Verlauf der Arbeit nicht mehr stark veränderte. Mit der Zeit wurden dann die einzelnen Zustände der Programms mit Funktionalität versehen.

Der Ansatz bei der Entwicklung der Ansprache der Störgeneratoren war anders. Um die Kommunikation zu etablieren, wurden erst essentielle Funktionen geschrieben. Es wurden VIs entwickelt, die als Prototypen die Ansprache der Geräte in Erfahrung bringen sollten. Durch Tests wurden Algorithmen entwickelt, die das Lesen von den Geräten bzw. das Senden an die Geräte möglich machten. Nachdem die Kommunikation erfolgreich funktionierte, wurde ein Prototyp des Test-Unterprogramms entwickelt, von dem die Device- und Test-Interfaces abgeleitet wurden. Diese Strategie bewährte sich und war sinnvoll, da für die Ansprache der Geräte wenig Vorkenntnis bestand und ein Konzept ohne vorherige Einarbeitung in die Schnittstellen nicht möglich gewesen wäre.

## 7.1 Anwendereingaben

Es ist sehr wichtig, das Programm vor Nutzereingaben zu schützen, die nicht vorgesehen sind. Auf diese Weise kann undefiniertes Verhalten auftreten. Das Programm kann unter Umständen auch abstürzen. Es ist zu unterscheiden zwischen Eingaben, die vom Anwender gefordert werden, und Eingaben, mit denen nicht gerechnet wird.

Zuerst werden Eingaben betrachtet, die der Anwender tätigen soll. Alle Eingaben, die unterschiedliche Wertigkeiten haben können, können problematisch sein. So ist zwar ein Button entweder gedrückt oder nicht gedrückt, ein Eingabefeld für eine Integerzahl kann jedoch viele

Werte annehmen. Besonders häufig sind solche Eingabefelder bei den manuellen Tests vorhanden. Dort kann beispielsweise eine Spannung oder eine Frequenz eingegeben werden. LabVIEW bietet die Möglichkeit, Bedienelemente nur für bestimmte Wertebereiche zulässig zu machen. Sollte der Anwender einen Wert wählen, der niedriger ist, wird der nächst höhere valide Wert gewählt. Weiterhin ist auch das Inkrement zu wählen. So kann eingestellt werden, ob jeder Integerwert gültig ist oder nur z. B. jeder zehnte, das Inkrement ist frei wählbar.

Die Aufgabenstellung ist komplexer für den Burst-Test. Das Gerät CEG-4500 gibt für die Parameter Burstperiode, Spike-Frequenz und Burstdauer jeweils eine Liste von 33 validen Werten vor. Diese Werte sind allerdings nicht gleichmäßig voneinander entfernt, so dass es zweckmäßig ist, Arrays für diese Werte anzulegen und falls eine Eingabe getätigt wird, die nicht einen dieser Werte enthält, den nächsten gültigen Wert zu nehmen. Auch wenn diese Prüfung geschehen ist, kann es immer noch zu Fehlern kommen, da manche Parameter sich gegenseitig beeinflussen. So können Wiederholperioden nicht höher sein als eine einzelne Test-Sequenzen. Diese Eingaben sind eingegrenzt auf ein VI. Sollten mehrere VIs gleichzeitig laufen, entstehen neue Probleme.

So ergab es sich anfangs, dass ein Test im Subpanel des Hauptmenüs gestartet wurde und der Anwender immer noch die Möglichkeit hatte, die Buttons des Hauptmenüs zu drücken, was dann zu ungewolltem Verhalten führte. Die einfachste Methode, dem zu entgehen, besteht darin, dem Anwender nicht die Chance zu geben, unerwünschte Eingaben zu tätigen. Die Buttons im Hauptmenü werden ausgegraut und die Tab-Reiter verschwinden, wenn das Test-Unterprogramm geöffnet wird. Wenn sich ein Dialog öffnet, der den Anwender zu einer Eingabe auffordert, kann dieses Fenster in den Vordergrund gestellt werden und das Hauptfenster kann erst wieder angesprochen werden, wenn der Dialog abgebrochen oder bestätigt wurde.

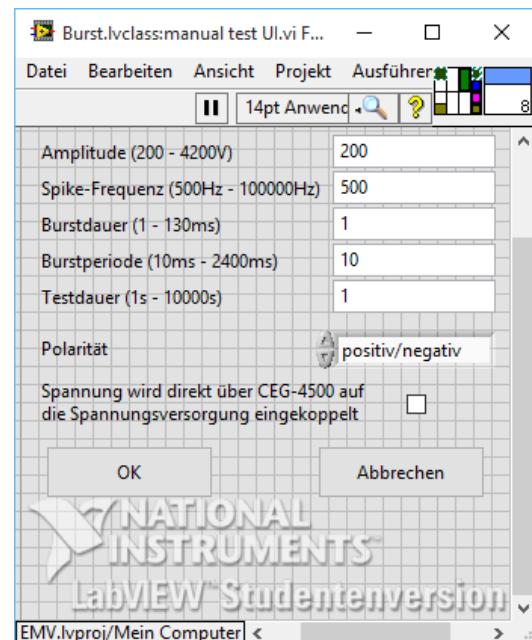


Abbildung 7.1: manual test UI für den Burst-Test

## 7.2 CEG-4500 (Hofbauer Electronic)

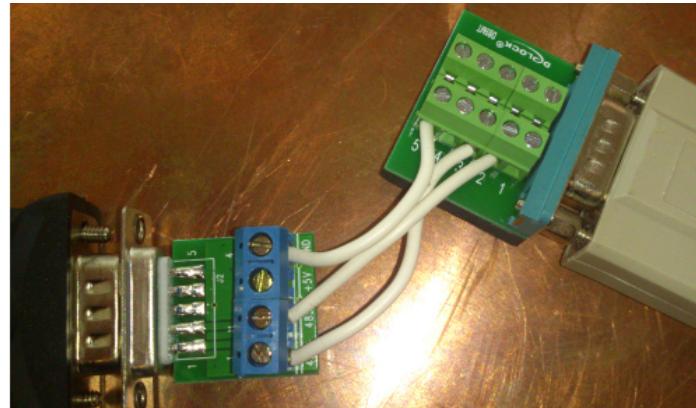
Im Inneren des Gerätes befindet sich ein 8-fach-Schalter, mit dem zwischen den beiden Standards (RS-232/RS-485) gewechselt werden kann. Mittels eines Konverters von USB auf RS-485 kann eine Verbindung mit dem Gerät hergestellt werden. Die Pinbelegung für den RS485-Standard an einem Sub-D-Stecker kann sehr unterschiedlich sein. Nach längerer Recherche wurde in der Anleitung für das CEG-4500 eine schematische Darstellung gefunden, in der die Pins 2,3 und 5 für RS485 benutzt werden. Die Pinbelegung wurde getestet, das Gerät ließ sich ansprechen.

Es fiel die Entscheidung, die Ansprache des Gerätes mittels RS-485 vorzunehmen, da das Gerät CFG-4028 vom selben Hersteller in Kombination mit diesem benutzt werden kann. RS-485 kann als Bus eingesetzt werden. Somit ist es möglich, dass beide Geräte über eine Verbindung angesprochen werden können. Die Ansteuerung des zusätzlichen Geräts wurde nicht implementiert, aber sie ist auf lange Sicht möglich und auch geplant.

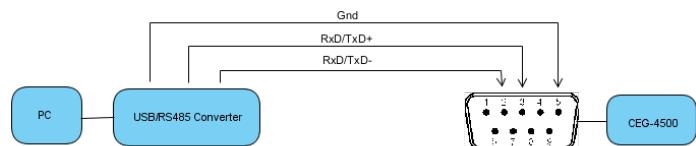
Der Aufbau eines Telegramms ist wie folgt:

**[Geräteadresse][Befehlsbyte][Daten][Prüfsumme][Endzeichen]**

1. **Geräteadresse:** Da das Gerät aus 3 Generatoren besteht, müssen diese adressiert werden. Die Adressen sind:
  - a) 2 = Burstgenerator
  - b) 3 = Netzunterbrechungsgenerator
  - c) 4 = Surgegenerator



(a) Die Verbindung



(b) Symbolische Darstellung

Abbildung 7.2: Verbindung PC - CEG-4500

2. **Befehlsbyte:** Es handelt sich um einen einzelnen Buchstaben. Er signalisiert entweder, dass Daten gesetzt (Großbuchstabe) oder ausgelesen (Kleinbuchstabe) werden sollen. Die Daten, die mit einem Großbuchstaben gesetzt werden, können anschließend mit einem Kleinbuchstaben verifiziert werden.
3. **Daten** Je nach Befehl kann hier eine Folge von Zahlen angeschlossen werden, um Parameter zu setzen.
4. **Prüfsumme** Nach jedem Befehl wird eine Prüfsumme angehängt, um die Datenintegrität sicherzustellen. Sie errechnet sich zu

$$\left( \sum_{i=0}^N x_i \right) \bmod 256 \quad (7.1)$$

wobei N die Anzahl aller vorherigen Zeichen inklusive Adresse ist.

5. **Endzeichen:** Das Telegramm kann mit einem CR/LF/LF+CR oder EOT abgeschlossen werden.

Das Gerät sendet keine Nachricht, wenn ein Telegramm fehlerhaft war. Wenn das Telegramm die richtige Prüfsumme hat, kommt immer eine Antwort zurück. Wenn das Gerät die Daten nicht interpretieren kann oder die Parameter nicht anwendbar sind, wird eine Fehlermeldung in Form eines Fehlercodes von 0 - 9 zurückgegeben.

### 7.3 emtest CWS 500N1

Um einen Befehl erfolgreich zu übertragen, wird eine Prüfsumme von dem Gerät gefordert. Diese Prüfsumme wird wie folgt berechnet: Die Summe der Wertigkeiten der ASCII-Zeichen wird genommen, wobei nur die letzten 8 Bit berücksichtigt werden. Dieses Ergebnis wird von 0x100 abgezogen und am Ende des Befehls vor dem abschließenden Linefeed angehängt. Wenn keine oder eine falsche Prüfsumme übermittelt wird, sendet das Gerät einen Fehler in der Form "RR,14;<LF>" zurück. Für die Errechnung der Prüfsumme wurde ein Programm in C geschrieben, welches per Dynamic Link Library (DLL) in LabVIEW integriert werden konnte (siehe Anhand checksum.c/checksum.h). Es wurde entschieden, diese Funktionalität mittels DLL einzubinden, um die Möglichkeit der Einbindung einer anderen Programmiersprache in LabVIEW zu testen.

## 7.4 Messdatenerfassung

Die Erfassung von Luftfeuchtigkeit und Temperatur wird mittels einer eigens angefertigten Baugruppe ermöglicht. Diese Baugruppe wurde mit THT-Bauteilen diskret aufgebaut.

### 7.4.1 Raumtemperatur

Eine Spannung von 10V wird an Pad5 angelegt. Die Spannung über dem NTC-Widerstand TTC05 wird über den Spannungsabfall an  $R_1$  errechnet. Hiermit kann der Widerstand errechnet werden und aus dem Widerstand wiederum die Temperatur. Für  $R_1$  wurde ein Wert von  $100\text{k}\Omega$  gewählt. Zu beachten ist hier, dass der Eingangswiderstand des NI9221 die Messung verfälscht und die Spannung an  $R_1$  eigentlich höher ist als die gemessene. Die Spannung fällt an  $R'_1$  ab, der sich aus dem Eingangswiderstand von NI9221 parallel zu  $R_1$  ergibt und in der Gleichung 7.2 aufgeschlüsselt ist.

$$R'_1 = \frac{1}{\frac{1}{R_1} + \frac{1}{R_a}} = \frac{1}{\frac{1}{100\text{k}\Omega} + \frac{1}{1M\Omega}} = 90, \overline{90} \text{k}\Omega \quad (7.2)$$

Der Widerstand von  $R_{NTC}$  errechnet sich dann wie folgt:

$$R_{NTC} = \frac{U_{NTC} \cdot R'_1}{U_G - U_{NTC}} \quad (7.3)$$

Der Umstand, dass sich der Referenz-Widerstand verkleinert, ist für die Messung positiv, da der Widerstand sich nun näher am Optimum von  $94\text{k}\Omega$  befindet (siehe Gleichung 6.1). Nachdem der Widerstand des Sensors berechnet wurde, muss nun die Temperatur daraus bestimmt werden. Als Näherung für den Widerstand ist die folgende Gleichung<sup>1</sup> zu finden:

$$R_{NTC} = R_{25} \cdot e^{-B \cdot \left( \frac{1}{T} - \frac{1}{T_{25}} \right)} \quad (7.4)$$

mit der Konstanten  $B$ , die für jeden NTC-Widerstand spezifisch ist zu:

$$B = \frac{E_A}{k_b} \quad (7.5)$$

wobei  $E_A$  für die Aktivierungsenergie steht und  $k_b$  für die Boltzmannkonstante

---

<sup>1</sup>Wikipedia (2015b)

nach T (Temperatur in Kelvin) aufgelöst ergibt dies:

$$T(R) = \frac{1}{\frac{\ln(\frac{R_{NTC}}{R_{25}})}{B} + \frac{1}{T_{25}}} \quad (7.6)$$

Wenn alle Werte laut Datenblatt in die Formel eingesetzt werden, entsteht folgende Rechnung:

$$T(R) = \frac{1}{\frac{\ln(\frac{R_{NTC}}{100k\Omega})}{4400K} + \frac{1}{297,15K}}$$

## 7.4.2 Luftfeuchtigkeit

Die Erfassung der Luftfeuchtigkeit wurde durch einen Spannungsteiler in der Schaltung realisiert (siehe Kapitel 6.4). Der Widerstand  $R_2$  wurde zu  $200k\Omega$  gewählt. Analog zur Erfassung der Raumtemperatur wird auch hier der Widerstandswert von  $R_2$  durch den Eingangswiderstand von NI9221 erniedrigt. Der mit Gleichung 7.2 für  $R_2$  errechnete Widerstand  $R'_2$  für das Hygrometer wird in LabVIEW mittels hygro.vi in die relative Luftfeuchtigkeit ( $\varphi$ ) umgewandelt. In Tabelle 6.1 ist der Zusammenhang zwischen  $\varphi, R_H$  (Widerstand in Abhängigkeit zur Luftfeuchte) und  $T$  (Temperatur) dargestellt. Zuerst wurde versucht, den Einfluss von  $T$  und  $\varphi$  auf den Widerstand mittels einer Formel anzunähern. Es ist zu beobachten, dass mit steigendem  $T$  der Widerstand exponentiell fällt. Dies gilt ebenfalls für  $R_H$ . Weiter ist zu erkennen, dass mit steigendem  $R_H$  der Einfluss auf den Widerstand abnimmt. Somit lässt sich der Zusammenhang nicht mit einer Exponentialgleichung annähern. Formel 7.7 wurde als möglicher Kandidat für eine Näherung ausgewählt. Mittels eines Matlab-Scripts (siehe Anhang ntc.m) wurden numerisch Konstanten ermittelt, die möglichst wenig Abweichung zu den Funktionswerten in Tabelle 6.1 aufweisen. Da die Datenbasis lediglich Widerstandswerte für Paare von Luftfeuchtigkeit und Temperatur zur Verfügung stellt, wurde die Formel mit dem Widerstand als Ergebnis gebildet.

$$R'_2 = \frac{1}{\frac{1}{R_2} + \frac{1}{R_a}} = \frac{1}{\frac{1}{200k\Omega} + \frac{1}{1M\Omega}} = 166, \overline{66} k\Omega$$

$t$  = Temperatur,  $R_H$  = Widerstand in Abhängigkeit zur Luftfeuchte,  
 $\varphi$  = relative Luftfeuchtigkeit

$$R_H = 9900k\Omega \cdot e^{(t-5) \cdot c_1 \cdot e^{(\varphi-25) \cdot c_2}} \cdot e^{(\varphi-25) \cdot c_3 \cdot e^{(\varphi-25) \cdot c_4}} \quad (7.7)$$

Das Matlab-Script basiert auf fünf Schleifen, die jeweils die Werte von  $c_1$  bis  $c_4$  leicht verändern, und im Anschluss die Abweichung mit dem vorgegebenen Fokus (Temperaturbereich) vergleichen. Wurde ein besserer Parametersatz gefunden, werden der Suchradius und die Schrittweite verkleinert und die Werte werden erneut variiert. Die maximale Abweichung über den gesamten Temperaturbereich konnte so auf unter 17% gebracht werden. Die maximale Abweichung im Anwendungsbereich von 15°C bis 30°C lag sogar bei knapp unter 10%. Für die Gleichung wurden folgende Konstanten berechnet:

$$C1 = 0.07395, \quad C2 = 0.0159, \quad C3 = 0.18979, \quad C4 = 0.0069$$

Um mit dieser Gleichung zu arbeiten, muss allerdings nach  $\varphi$  aufgelöst werden. Dies stellte sich als unlösbare Aufgabe heraus. Da der erste Term in Abhängigkeit von der Temperatur steht und der zweite in Abhängigkeit zur relativen Luftfeuchtigkeit, lässt sich zwar von der gesamten Gleichung einmal der natürliche Logarithmus ziehen:

$$e^{(t-5) \cdot c_1 \cdot e^{(\varphi-25) \cdot c_2}}, e^{(\varphi-25) \cdot c_3 \cdot e^{(\varphi-25) \cdot c_4}}$$

Danach aber ist keine weitere Vereinfachung möglich, da der neue Term über eine Addition auf der untersten Ebene verfügt und kein Zusammenfassen mehr möglich ist (siehe Gleichung 7.4.2). Es wurde viel Zeit in die Bestimmung der Konstanten investiert. Dies hätte vermieden werden können, wenn vorher überprüft worden wäre, ob die gefundene Gleichung nach  $\varphi$  aufgelöst werden kann. Die Fragestellung wurde während der Erarbeitung der Konstanten nicht berücksichtigt. Eine bessere Möglichkeit wäre unter Umständen eine polynomiale Anpassung gewesen.

$$\ln\left(\frac{R_H}{9900k\Omega}\right) = (t - 5) \cdot c_2 \cdot e^{(\varphi-25) \cdot c_2} + (\varphi - 25) \cdot c_3 \cdot e^{(\varphi-25) \cdot c_4}$$

Das Problem wurde mittels eines C-Programms (siehe Anhang hygro.c/hygro.h) gelöst, welches in LabVIEW als DLL eingebunden werden konnte. Das Programm hat die Wertetabelle des Sensors HCZ-J3A laut Datenblatt hinterlegt und errechnet mittels linearer Annäherung die Luftfeuchtigkeit. Diese Methode gewährleistet, dass für die im Datenblatt genannten Wertepaare das Ergebnis genau berechnet wird und zwischen den Werten lediglich eine kleine Ungenauigkeit entstehen kann. Diese ist allerdings vernachlässigbar, da das Bauteil eine Abweichung von  $\pm 5\%$  haben kann und die unterschiedlichen Temperaturkurven nicht linear sind. Eine Annäherung auf diese Weise ist am sinnvollsten.

## 7.5 Lizenzierung

Bei der der Frage der Lizenzierung müssen die unterschiedlichen Ziele der Stakeholder in Betracht gezogen werden (siehe Tabelle 5.2 und 5.3). Die Schnittstellendefinition für das Gerät CEG-4500 wurde nur unter der Bedingung freigegeben, dass der/die entsprechende/n Source-Code/Treiber für LabVIEW zur Zertifizierung und unentgeltlichen unbefristeten Nutzung bereitgestellt wurden.

Die Abteilung TT (Technologie-Transfer) hat versucht, einen Lizenzvertrag mit der Firma auszuhandeln, der unter anderem einen geldlichen Ausgleich für finanzielle Vorteile der Firma an DESY vorsah. Die Verhandlungen waren mühsam, da die Verpflichtungen, die der Firma durch den ersten Entwurf des Lizenzvertrags auferlegt wurden, recht umfangreich waren. Eine Einigung erreichte schließlich, dass eine GPL für das gesamte LabVIEW-Programm gewählt wurde, so dass der Hersteller uneingeschränkten Zugriff auf das Programm und den Sourcecode hat. Der Hersteller wird einen Teil des Programms (Test-Unterprogramm inklusive der Implementierung für CEG-4500 und die damit möglichen Tests) erhalten, im Gegenzug wurde die Schnittstellendefinition für diese Entwicklung offen gelegt. Die Lizenzierung für das Programm zur Generierung des Prüfberichts muss eine AGPL sein, da iText diese Lizenz innehaltet und jede Entwicklung, die dieses Framework nutzt, auch automatisch eine AGPL<sup>2</sup> erhalten muss.

## 7.6 Beta-Testing

Das Programm konnte von den zukünftigen Anwendern ausgiebig getestet werden. So wurden einige Fehler sichtbar, die bei den Tests während der Entwicklung nicht erkannt wurden. Da der Ablauf des Programms dem Entwickler geläufig ist, wird eine andere Möglichkeit der Bedienung in vielen Fällen gar nicht erst in Erwägung gezogen. Dies ist bei den Anwendern nicht der Fall. Sie wissen meist nur bedingt, wie das Programm auf der Benutzerebene funktioniert, somit ist hier die Möglichkeit von unerwarteten Eingaben viel höher.

Nachdem zwei Personen das Programm jeweils für mehrere Stunden getestet hatten (dies stellt eine verkürzte Alpha-Phase dar), wurden die gefundenen Fehler berichtigt und es wurde der erste Release des Programms erstellt.

Ab diesem Zeitpunkt war das Programm für die Mitarbeiter des Prüffeldes zugänglich und es konnte damit begonnen werden, mit diesem Programm zu arbeiten. Da je nach Auftragslage auch für einen längeren Zeitraum keine EMV-Prüfungen stattfinden, wird diese Testphase wohl längere Zeit andauern. Wenn während der Arbeit mit diesem Programm Fehler auftreten, wird von den Anwendern ein Bug-Report verfasst mit einer genauen Beschreibung des

---

<sup>2</sup>agplv3,Abs. 5.c

**Punktes im Programmablauf**, bei dem dieser Fehler geschehen ist, der **Schritte, die gemacht wurden**, bevor der Fehler aufgetaucht ist und **wie genau sich der Fehler darstellt**. Diese Bugreports können dann nach und nach abgearbeitet werden. Ab dem Zeitpunkt, an dem die Software ein gewünschtes Mindestmaß an Qualität erreicht hat, wird dann der Live-Release vollzogen.

# 8 Fazit

Die Arbeit deckt mehrere Themenbereiche ab:

**1. Klimadatenerfassung**

- a) Erstellung eines mathematischen Modells
- b) Entwurf und Aufbau einer Schaltung

**2. Software Entwicklung**

- a) Hauptprogramm, Durchführung und Dokumentation von Tests
- b) Java Programm, Erstellung von Prüfberichten

**3. Ansteuerung von Störgeneratoren über Bussysteme**

- a) CEG-4500 (RS-485)
- b) CWS 500N1 (USB)

Der größte Teil der Arbeit behandelt die Software-Entwicklung. Hier wurden zwei Programme entwickelt, um zum einen Tests im Rahmen der EMV-Normen für die EU-Konformitätserklärung durchzuführen und zu dokumentieren und zum anderen die Dokumentation in Form einer PDF-Datei auszugeben. Es wurde eine Schaltung für die Messung von Klimadaten entworfen und eine entsprechende Baugruppe hergestellt. Des Weiteren wurde die Kommunikation zwischen Geräten mittels unterschiedlicher Bussysteme (USB, RS-485) etabliert.

Die Entwicklungsumgebung LabVIEW hat sich als sehr einsteigerfreundlich erwiesen und konnte mit seinen vielen Treibern und Bibliotheken sowie sehr guten Beispielen seine Wahl als Entwicklungsumgebung unterstreichen. Der Vorteil der grafischen Datenflussprogrammierung ist ganz klar, dass modulare Programmierung nahezu erzwungen wird, da große Algorithmen sehr schnell unübersichtlich werden und es notwendig ist, SubVIs zu erstellen, um sich weiter zurecht zu finden. Der Nachteil ist, dass es trotz gutem Programmierstil auch unübersichtlich werden kann, da beispielsweise Anzeige- oder Bedienelemente lediglich mit ihrem Datentyp farbig unterschieden werden. So sind falsch geschlossene Verbindungen möglich. Diese Fehler können bei textbasierten Programmiersprachen nicht in diesem Maße

passieren.

Softwareentwicklung ist ein sehr schwer planbarer Prozess. So kann man sagen, dass Software ab einer bestimmten Dimension eigentlich nie fertig ist, da die Möglichkeit von Fehlern exponentiell zur Größe des Projekts anwächst. Hinzu kommt, dass die Software auch noch Verbesserungen in Hinsicht auf Usability und Performance benötigen könnte, und wenn dies vollbracht ist, gibt es neue Fehler, die korrigiert werden müssen. Es war sehr schwierig, die Weiterentwicklung des Programms zu unterbrechen, da der Programmcode an vielen Stellen noch modularer hätte gestaltet werden können. Das Programm war jedoch in seiner Funktionalität fertig.

Schwer zu kalkulieren war auch, die Zeit die für das Debugging veranschlagt werden musste. Dieser Prozess ist auch mit Abschluss dieser Arbeit nicht zu Ende und wird im Zusammenhang mit den bestehenden Kontakten zum DESY weitergeführt werden.

Für die Entwicklung bot sich ein iterativer Entwicklungsansatz an, damit Entscheidungen wiederholt auf ihre Validität geprüft werden. Es wurden trotzdem Entscheidungen getroffen, die im Nachhinein nicht optimal waren, durch eine längere Analyse wären unter Umständen bessere Lösungen entstanden. Gesichert ist allerdings auch dies nicht, da sich erst während der Entwicklung viele Erkenntnisse und Problemstellungen zeigten. Der iterative Entwicklungsprozess kann die Risiken in dieser Hinsicht minimieren, aber nicht ganz verschwinden lassen.

Einige Ideen, die ursprünglich implementiert werden sollten, mussten ausgesetzt werden, da im Fall des Spektrumanalyzers RSA5100 von Tektronix das Gerät wochenlang nicht zur Verfügung stand, da es in den USA zur Reparatur war. Im Fall des Geräts CFG-4028 wurde entschieden, die Implementierung auszusetzen. Die meisten Baugruppen haben lediglich einen einphasigen Netzanschluss, das Gerät ist erst bei dreiphasigem Netzanschluss notwendig. Der Rahmen der Arbeit wäre gesprengt worden und nur ein Bruchteil der Prototypen hätte daraus Nutzen ziehen können. Die Implementierung beider Geräte ist jedoch auf lange Sicht geplant.

Die Aufgabenstellung wurde erfolgreich beendet, es wurde ein Programm zur Ansteuerung der Störgeneratoren emTest CWS 500N1 und CEG-4500 erstellt. Die klimatischen Bedingungen werden bei jedem Norm-Test erhoben und der Anwender kann die Prüfung mit Hilfe der Software dokumentieren und im Anschluss einen Prüfbericht im PDF-Format erstellen.

# Literaturverzeichnis

[HCZ ] *Datenblatt: HCZ-J3A HUMIDITY SENSOR*

[NTC ] *Datenblatt: NTC Thermistor TTC05 Series 5 mm Lead Type for Temperature Sensing/Compensation*

[agplv3 ] : *GNU Affero General Public License.* – URL <http://www.gnu.org/licenses/agpl-3.0-standalone.html>

[gplv3 ] : *GNU General Public License.* – URL <http://www.gnu.org/licenses/gpl-3.0-standalone.html>

[EMV63 2001] : *Elektromagnetische Verträglichkeit (EMV) Teil 6-3: Fachgrundnormen - Störaussendung für Wohnbereich, Geschäfts- und Gewerbebereiche sowie Kleinbetriebe.* 2001

[EMV411 2004] : *Elektromagnetische Verträglichkeit (EMV) - Teil 4-11: Prüf- und Messverfahren - Prüfungen der Störfestigkeit gegen Spannungseinbrüche, Kurzzeitunterbrechungen und Spannungsschwankungen (IEC 61000-4-11:2004); Deutsche Fassung EN 61000-4-11:2004.* 2004

[EMV62 2006] : *Elektromagnetische Verträglichkeit (EMV) - Teil 6-2: Fachgrundnormen - Störfestigkeit für Industriebereiche (IEC 61000-6-2:2005); Deutsche Fassung EN 61000-6-2:2005.* 2006

[EMV41 2007] : *Elektromagnetische Verträglichkeit (EMV) - Teil 4-1: Prüf- und Messverfahren - Übersicht über die Reihe IEC 61000-4 (IEC 61000-4-1:2006); Deutsche Fassung EN 61000-4-1:2007.* 2007

[EMV61 2007] : *Elektromagnetische Verträglichkeit (EMV) - Teil 6-1: Fachgrundnormen - Störfestigkeit für Wohnbereich, Geschäfts- und Gewerbebereiche sowie Kleinbetriebe (IEC 61000-6-1:2005); Deutsche Fassung EN 61000-6-1:2007.* 2007

[EMV42 2009] : *Elektromagnetische Verträglichkeit (EMV) - Teil 4-2: Prüf- und Messverfahren - Prüfung der Störfestigkeit gegen die Entladung statischer Elektrizität (IEC 61000-4-2:2008); Deutsche Fassung EN 61000-4-2:2009.* 2009

- [EMV43 2010] : *Elektromagnetische Verträglichkeit (EMV) - Teil 4-3: Prüf- und Messverfahren - Prüfung der Störfestigkeit gegen hochfrequente elektromagnetische Felder (IEC 61000-4-3:2006 + A1:2007 + A2:2010); Deutsche Fassung EN 61000-4-3:2006 + A1:2008 + A2:2010.* 2010
- [EMV48 2010] : *Elektromagnetische Verträglichkeit (EMV) - Teil 4-8: Prüf- und Messverfahren - Prüfung der Störfestigkeit gegen Magnetfelder mit energietechnischen Frequenzen (IEC 61000-4-8:2009); Deutsche Fassung EN 61000-4-8:2010.* 2010
- [EMV64 2011] : *Elektromagnetische Verträglichkeit (EMV) - Teil 6-4: Fachgrundnomen - Störaussendung für Industriebereiche (IEC 61000-6-4:2006 + A1:2010); Deutsche Fassung EN 61000-6-4:2007 + A1:2011.* 2011
- [EMV44 2012] : *Elektromagnetische Verträglichkeit (EMV) - Teil 4-4: Prüf- und Messverfahren - Prüfung der Störfestigkeit gegen schnelle transiente elektrische Störgrößen/Burst (IEC 61000-4-4:2012); Deutsche Fassung EN 61000-4-4:2012.* 2012
- [EMV46 2014] : *Elektromagnetische Verträglichkeit (EMV) - Teil 4-6: Prüf- und Messverfahren - Störfestigkeit gegen leitungsgeführte Störgrößen, induziert durch hochfrequente Felder (IEC 61000-4-6:2013); Deutsche Fassung EN 61000-4-6:2014.* 2014
- [EMV45 2015] : *Elektromagnetische Verträglichkeit (EMV) - Teil 4-5: Prüf- und Messverfahren - Prüfung der Störfestigkeit gegen Stoßspannungen (IEC 61000-4-5:2014); Deutsche Fassung EN 61000-4-5:2014.* 2015
- [A.Schatten 2010] A.SCHATTEN, M. Demolsky E. Gostischa-Franta Th. Östereicher D. W.: *Best Practice Software-Engineering.* Spektrum Akademischer Verlag, 2010
- [emtest 2008] emtest (Veranst.): *MANUAL For Remote Control Continuous Wave Simulator Type CWS 500N1.* Oktober 2008
- [emTest 2013] emTest (Veranst.): *Bedienungsanleitung für CWS 500N1, IEC 61000-4-6 / EN 61000-4-6.* 8.14. 12 2013
- [EU 2014] EU: *RICHTLINIE 2014/30/EU DES EUROPÄISCHEN PARLAMENTS UND DES RATES vom 26. Februar 2014 zur Harmonisierung der Rechtsvorschriften der Mitgliedstaaten über die elektromagnetische Verträglichkeit (Neufassung).* 2 2014. – URL <http://www.ce-zeichen.de/templates/ce-zei/richtlinien/emv-2014-30-eg.pdf>. – abgerufen am 13.08.2015, hinterlegt als emv-2014-30-eg.pdf im Anhang
- [jbw 2015] JBW: Netzagentur versteigert Frequenzen für fünf Milliarden Euro. In: *Zeit.de* (2015). – abgerufen 25.08.2015
- [Kleuker 2013] KLEUKER, Stefan: *Grundkurs Software-Engineering.* Springer Vieweg, 2013

- [Lotz 2011] LOTZ, Paul J.: *Strategy Pattern with LabVIEW*. April 2011
- [Lowagie 2011] LOWAGIE, Bruno: *iText in Action Second Edition*. Manning, 2011
- [Lowagie 2015] LOWAGIE, Bruno: *The ABC of PDF with iText*. Lean Publishing, 2015
- [Mütterlein 2009] MÜTTERLEIN, Bernward: *Handbuch für die Programmierung mit LabVIEW*. Spektrum Akademischer Verlag, 2009
- [NI 2014] NI: VISA. (2014). – URL <https://www.ni.com/visa/>. – abgerufen am 28.07.2015, hinterlegt als NI2014.html im Anhang
- [Stotz 2013] STOTZ, Dieter: *Elektromagnetische Verträglichkeit in der Praxis*. Springer Vieweg, 2013
- [unbekannt 2015] UNBEKANNT: *EMV Kombi/Einzel-Generatoren, Übertragungs Protokoll für die Schnittstelle RS232 / RS485*. 2015. – Peter Hofbauer Electronic GmbH
- [Wikipedia 2015a] WIKIPEDIA: Harmonisierte Norm. (2015). – URL [https://de.wikipedia.org/wiki/Harmonisierte\\_Norm](https://de.wikipedia.org/wiki/Harmonisierte_Norm). – abgerufen am 13.08.2015, hinterlegt als harmonisierteNormWikipedia.pdf im Anhang
- [Wikipedia 2015b] WIKIPEDIA: Heißleiter. (2015). – URL <https://de.wikipedia.org/wiki/Heißleiter>. – abgerufen 27.08.2015, hinterlegt als heissleiter.pdf im Anhang
- [Wikipedia 2015c] WIKIPEDIA: Proximity effect. (2015). – abgerufen 21.08.2015, hinterlegt als proximitywiki.pdf im Anhang
- [Zach 1990] ZACH, Franz: *Leistungselektronik*. Springer-Verlag Wien GmbH, 1990

# Index

- Aktivitätsdiagramm, 42
- Amplitudenmodulation, 28
- Anzeigeelement, 16
- Baumstruktur, 52
- Bedienelement, 20
- Bedienelemente, 72
- Befehlsbyte, 73
- Betriebsmittel, 23
- Bewertungskriterien, 23
- Blockdiagramm, 20
- Bug-Report, 78
- Burst, 29
- CDN, 12, 45
- CE-Kennzeichnung, 22
- CE-Zertifizierung, 41
- CEG-4500, 13, 42, 73, 81
- CFG-4028, 12, 73
- Cluster, 16
- COM-Port, 59
- CWS 500N1, 13, 74, 81
- Datenflussprogrammierung, 15
- Datenstruktur, 50
- Datentypen, 17
- Device-Interface, 53, 58
- DLL, 74
- EMV (Elektromagnetische Verträglichkeit), 22
- Endzeichen, 58, 73
- ESD (Electrostatic Discharge), 27
- EU-Konformitätserklärung, 22
- Event-Struktur, 20
- Fachgrundnormen, 23
- Fehler, 53
- Fehlermeldung, 56
- Fibonacci Sequenz, 18
- FPGA, 15
- Framework, 10
- Generator, 13
- Geräteadresse, 73
- Grafische Datenflussprogrammierung, 16
- Grundnorm, 23
- Handler, 64
- harmonisierte Norm, 22
- Hauptmenü, 42, 52
- Hauptprogramm, 50, 52
- HCZ-J3A, 69
- Java, 50, 63
- Kalibrierdaten, 45
- Kalibrierung, 45
- kapazitive Kopplung, 30
- Klassen, 17
- Klassendiagramm, 63
- Klimadatenerfassung, 65
- Kommandoliste, 56, 63
- Kurzzeitunterbrechungen, 35
- LabVIEW, 15, 71
- Lizenzierung, 78
- Luftfeuchtigkeit, 27, 50, 65, 76

- Luftfeuchtigkeitssensor, 69  
LVClass, 17  
Messaufbauten, 25  
Netzimpedanz, 14  
NI 9221, 14, 69  
NI 9264, 14, 68, 69  
NI-DAQmx, 14  
Normen, 23  
NTC, 66  
Oberschwingungsströme, 26  
Objekte, 17  
Pause, 56  
Polymorphie, 53  
Polymorphismus, 17  
Prüfbericht, 39, 50  
Prüfschärfegrade (Prüfpegel), 27  
Prüfsumme, 58  
Prüfverfahren, 25  
Produktgruppennormen, 23  
Produktnormen, 23  
Programmierparadigma, 16  
Quasispitzenwert, 26  
Refnum, 17  
Release, 78  
RS-232, 13  
RS-485, 12, 13, 73  
RSA5100, 44  
SAX, 64  
Schnittstellendefinition, 78  
Segment, 65  
Sensor, 65  
Sollwert, 56  
Spannungseinbrüche, 35  
Spannungsschwankungen, 35  
Spektrumanalyzer, 44  
Störaussendung, 22, 25, 44  
Störfestigkeit, 22  
Störungsabgabe, 22  
Störungseinwirkung, 22  
Stakeholder, 37  
Stehwellenverhältnis, 28  
Subpanel, 52  
Surge, 32  
Temperatur, 50, 66, 75  
Test-Interface, 56, 61  
Test-Unterprogramm, 42, 52  
Thermistor, 66  
USB, 73  
Use-Cases, 42  
Versuchsaufbau, 42  
VI (Virtual Instrument), 16  
Virtual Instrument Software Architecture (VISA), 20  
VISA-Resource, 58  
XML-Parsing, 63  
XML-Struktur, 50  
Ziele, 37  
Zustandsautomat, 53  
Zustandsautomatendiagramm, 52  
Zweileiter-V-Netznachbildung, 14

# **Versicherung über die Selbstständigkeit**

Hiermit versichere ich, dass ich die vorliegende Arbeit im Sinne der Prüfungsordnung nach §16(5) APSO-TI-BM ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen habe ich unter Angabe der Quellen kenntlich gemacht.

Hamburg, 29. August 2015

Ort, Datum

Unterschrift