

Datenverarbeitung und Visualisierung von Umrichter- Testbench-Daten

Jon Feddersen

22122017

Bachelorarbeit im Studiengang Elektrotechnik und Inforationstechnik

bei

Prof. Dr. rer. nat. Kristina Schädler

Semesteranschrift
Ochsendrift 31
25853 Drelsdorf

Studienfach
Elektrotechnik und
Inforationstechnik

Abgabetermin: 01.10.2025

Fachsemesterzahl: 9

Sperrvermerk

Diese Arbeit enthält vertrauliche Daten und Informationen des Unternehmens, in dem die Bachelor-/Masterarbeit angefertigt wurde. Sie darf Dritten deshalb nicht zugänglich gemacht werden.

Die für die Prüfung notwendigen Exemplare verbleiben beim Prüfungsamt und beim betreuenden Hochschullehrer.

Inhaltsverzeichnis

Abbildungsverzeichnis	IV
Tabellenverzeichnis	V
Abkürzungsverzeichnis	VI
1 Einleitung	1
2 Grundlagen	2
2.1 Überblick über den Umrichter-Prüfstand	2
2.2 Verarbeitung von XML-Daten	4
2.2.1 XML-Strukturaufbau	5
2.2.2 Verarbeiten von XML-Dateien mit Python	6
2.3 Datenbankentwurf und Normalisierung	7
2.4 Grundlagen der Datenvisualisierung	7
2.5 Anforderungen an modulare Softwareentwicklung	7
3 Analyse und Konzeption	8
3.1 Definition der Anforderungen	8
3.2 Analyse der bestehenden Strukturen und Prozesse	8
3.3 Datenbankdesign und Strukturkonzeption	8
3.4 Grundkonzept des Benutzeroberflächen-Design	8
3.5 Entwurf der Applikationsarchitektu	8
4 Implementierung	9
4.1 Einlesen und Verarbeiten von XML-Daten	9
4.2 Implementierung der Datenbank	9
4.3 Entwicklung der Benutzeroberfläche	9
4.4 Technische Details zur Visualisierung	9
5 Integration und Test	10
5.1 Einbindung in die bestehende Systemlandschaft	10
5.2 Testmethoden und Durchführung	10
5.3 Ergebnisse der Testmethoden	10
6 Fazit und Ausblick	11
6.1 Zusammenfassung der Ergebnisse	11
6.2 Kritische Bewertung	11
6.3 Möglichkeiten für zukünftige Erweiterungen	11
Literaturverzeichnis	i
Anhangsverzeichnis	ii
Erklärung	iii

Abbildungsverzeichnis

1	Aufbau des Teststandes	3
2	XML Prolog Beispielcode	5
3	XML Elemente Beispielcode	5
4	XML Attribute Beispielcode	5
5	XML Kommentare Beispielcode	5
6	XML Text Beispielcode	6

Tabellenverzeichnis

Abkürzungsverzeichnis

DUTs Devices-Under-Test

1 Einleitung

Das Aufbereiten von Daten, sowohl in Bezug auf die Struktur, in der die Datensätze gespeichert werden, als auch in der visuellen Darstellung einzelner Datenreihen, ist für das Verstehen der dahinterliegenden Prozesse und der effizienten Arbeit mit ihnen unerlässlich. Das Erfassen von Messdaten und ihre Aufbereitung, so wie ihre Interpretation sind aus der modernen Welt nicht mehr wegzudenken. In allen Bereichen der Wissenschaft werden Messdaten erhoben, egal ob bei medizinischen Studien, Wetterdaten oder in der Industrie erhobenen Testdaten, sie müssen alle verständlich aufbereitet werden.

Die Speicherung, Aufbereitung und Visualisierung von größeren Datenmengen wird heutzutage meistens mit Softwaretools. Aus diesem Grund gibt es eine Vielzahl von Anbietern die Lizenzen für solchen Softwaretools vertreiben. Diese Tools sind oft aber allgemein gehalten und bieten daher für fachspezifische Bereiche nicht den passenden Aufbau und Funktionen, welches sich Effizienz, Umgänglichkeit und vor allem in der Genauigkeit widerspiegelt.

Das Ziel dieser Arbeit ist es, eine Web-Applikation zu entwickeln, die als solch ein Softwaretool fungieren soll, welches spezifisch für einen Anwendungsbereich zugeschnitten ist. Die Applikation soll XML-Datensätze, die aus einem Umrichter-Teststand stammen strukturiert speichern und visuell darstellen können. Diese visuellen Daten sollen sowohl firmenintern zur Analyse als auch als in einem Kundenbericht nach außen weitergegeben werden, daher soll die Visualisierung sowohl einfach verständlich also auch professionell gehalten sein, hierbei soll möglichst auf eine intuitive und simple Nutzeroberfläche geachtet werden. Weiterhin soll die Applikation gut erweiterbar sein, um zukünftige Funktionen unkompliziert in die Software einzubetten. Zudem soll eine möglichst einfache Implementierung in die bestehende Softwareumgebung der Firma in dessen Rahmen diese Arbeit durchgeführt wird gewährleistet sein.

Um das gesetzte Ziel und die Anforderungen im Zeitrahmen dieser Arbeit bestmöglich zu erfüllen, wird nach dem Erlangen und Ausformulieren der theoretischen Grundlagen der Erstellung für den eigentlichen Entwicklungsprozess der Web-Applikation eine inkrementelle und iterative Vorgehensweise vorgesehen. Diese soll für Flexibilität und gutes Risikomanagement in Bezug auf die Abgabe sorgen, da selbst bei Komplikation eine funktionelle Version zur Abgabe bereitsteht.

Dieser Ansatz zur Entwicklung der Software wird sich nur geringfügig auf den Aufbau der wissenschaftlichen Arbeit auswirken, sie wird nur in den theoretischen Grundlagen und im Fazit behandelt werden. Der allgemeine Aufbau des Hauptteils der Arbeit beginnt mit den theoretischen Hintergründen zur Strukturierung der Entwicklung der Applikation, sowie die einzelnen zu implementierenden Funktionen. Gefolgt von einer detaillierten Darstellung der verwendeten Methoden im zweiten Kapitel. Das dritte Kapitel beschreibt die Entwicklung der Web-Applikation und das vierte Kapitel analysiert die Ergebnisse und diskutiert die Implementierung. Abschließend werden die wichtigsten Ergebnisse zusammengefasst und ein Ausblick auf mögliche Erweiterungen gegeben.

2 Grundlagen

In diesem Kapitel werden die theoretischen Grundlagen für das Entwickeln der Software, so wie das notwendige Verständnis des Testandes und seine Abläufe behandelt.

2.1 Überblick über den Umrichter-Prüfstand

In diesem Unterkapitel wird der Umrichter-Teststand, von dem die zu verarbeitenden Datensätze stammen beschrieben, da dies für das generelle Verständnis der einzulernenden Datensatzstruktur unerlässlich ist. Die genaue Bezeichnung des Teststandes „USTB DWT Test Bench (XCT0006-1)“ im weiteren als Test-Bench oder Teststand benannt. Diese Art Teststand wird im allgemein für die End-of-Line Prüfung von unterschiedlichen Umrichtern nach ihrer Herstellung genutzt, um die Produktqualität und -funktionalität sicherzustellen.[1]

In dem hier vorliegenden Fall wird der Teststand verwendet, um die aus dem Feld kommenden Umrichter auf ihre weiter Nutzungstauglichkeit zu testen. Die weiter Nutzungstauglichkeit wird ermittelt, indem die Messwerte mit Mittelwerten, die von mehreren fabrikneuen Umrichtern stammenden verglichen werden. Diese Messwerte müssen sich in einen vorher definierten Toleranzbereich befinden, um weiter verwenden zu werden.

Die Test-Bench besteht aus mehreren Hauptkomponenten:

- Das Netzteil, welches mit einer 400V Netzspannung versorgt wird, wandelt diese in eine Gleichspannung. Das Netzteil liefert maximal 8kW mit 1200VDC oder 800VDC, welche Werte verwendet werden kann vor Teststart bestimmt werden. In Abbildung 11 mit PSU bezeichnet, für „Power Supply Unit“.
- Das Elektronik-Rack, auf dem Mess- und Control-Komponenten befestigt sind. Hier befinden sich auch der (XCS2100) System-Controller der das ganze System mit dem PC, auf dem die Test-Bench Software läuft, via Ethernet verbindet. In Abbildung 1 mit ER bezeichnet, für „Electronic Rack“.
- Der Testmatrix-Schrank, in dem die Sammelschienen für den Stromanschluss und den Schützen sitzen. In Abbildung 1 mit TM bezeichnet, für „Test Matrix cabine“.
- Der Schrank mit dem Kühlungssystem, da die Umrichter während des Betriebes Wassergekühlt werden müssen. In Abbildung 1 mit „Cool1“ bezeichnet.
- Dem Carrier, auf dem Umrichter befestigt werden. Dieser wird speziell für bestimmte Umrichter konstruiert. In Abbildung 1 mit Carrier1 bezeichnet.

Neben dem Hauptkomponenten befinden sich außerhalb des Sicherheitsbereiches, der während des Betriebes nicht betreten werden darf, ein PC mit einer Software zum Steuern der Testeinrichtung, sowie eine Betriebsanzeige und ein Notaus. [1]

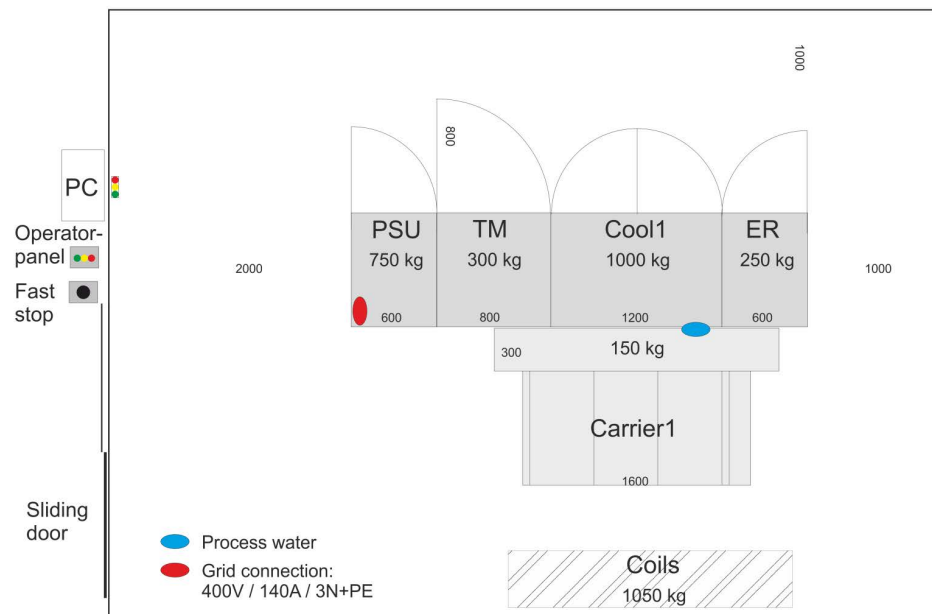


Abbildung 1: Aufbau des Teststandes
Quelle: [1, S. 7]

Ein Test läuft wie folgt ab:

1. Die Umrichter werden auf dem Carrier befestigt, es werden meist 3 Umrichter gleichzeitig getestet, Abweichungen je nach Bauform der Umrichter. Ab diesem Zeitpunkt werden die Umrichter in der gegebenen Fachliteratur als Devices-Under-Test (DUTs) bezeichnet, dies kommt auch in den Testberichten vor, daher hat der Autor diesen Begriff einzuführen und fortan zu übernehmen.
2. Durchführung des Driver Consumption Testes.
3. Durchführung des Pulse Testes.
4. Durchführung des Driver Power Testes.
5. Nach dem Durchlaufen eines Tests wird automatisch ein XML-Datenfile mit den erhobenen Messdaten generiert, die enthält die Messwerte und alle vorher bestimmten Einstellungen.

Es gibt verschiedene Testmodule, die auf dem Teststand laufen. Einige der verschiedenen Funktionen eines DUT können mit dem gleichen Modus eines Testmoduls überprüft werden, indem die entsprechenden Parametersätze ausgewählt werden. Jeder Test ist autonom und kann mehrmals ausgeführt werden, auch mit unterschiedlichen Parametersätzen.

Im Folgenden wird eine kurze Beschreibung der Funktionen der für diese Arbeit relevanten Testmoduls gegeben.

Driver Consumption Test:

Der Driver-Stromverbrauchstest überprüft den Stromverbrauch des Treibers im Leerlauf und während der PWM-Schaltung.

Pulse Test:

Der Impulstest verfügt über drei Funktionsmodi.

- Im Modus „Funktionsschaltung“ (FSW) kann überprüft werden, ob die Halbleiter generell schalten.
- Im Modus „Überstromschutz“ (OCP) kann die Überstromüberwachung („weicher Kurzschluss“) überprüft werden.
- Im Modus „Dynamischer Kurzschlusschutz“ (DSCP) wird das korrekte Verhalten der Treiberstufe in Bezug auf einen harten Kurzschluss überprüft.

Power Test:

Mit diesem Test können zwei verschiedene Funktionen getestet werden:

- Einerseits eignet sich der Burn-In-Test (BIT) dazu, die DUTs zyklisch zu betreiben und so reale Betriebszustände zu simulieren.
- Andererseits kann durch Überprüfung der Kühltemperatur am Ende des Tests die korrekte Wärmeübertragung der Halbleiter überprüft werden.

Während des Übertemperaturschutz-Tests (OTP) wird ein DUT mit reduzierter Kühlung betrieben, bis die maximal zulässige Kühlkörpertemperatur erreicht ist und die Temperaturschutzschaltung auslöst.

2.2 Verarbeitung von XML-Daten

XML, die Abkürzung für Extensible Markup Language, ist eine der beliebtesten Auszeichnungssprachen, die entwickelt wurde, um Informationen in einem maschinenlesbaren und strukturierten Format zu speichern und zu transportieren. Es ist eine textbasierte Auszeichnungssprache, die heute in vielen Anwendungen verwendet wird, wie z.B. Webdiensten, Datenbanken, Konfigurationsdateien und vielen anderen. XML ermöglicht die hierarchische Organisation von Informationen in einer strukturierten Weise, die sowohl für Menschen als auch für Maschinen verständlich ist.

Die Motivation hinter XML war, eine universelle und erweiterbare Sprache zu schaffen, die von verschiedenen Systemen unabhängig von der zugrunde liegenden Technologie genutzt werden kann. Die Fähigkeit, Daten in einem offenen Standard zu speichern und auszutauschen, war entscheidend, um die Interoperabilität zwischen verschiedenen Anwendungen und Plattformen zu fördern.

2.2.1 XML-Strukturaufbau

Ein XML-Dokument besteht aus einer Reihe von Elementen, die durch Tags markiert sind. Die grundlegenden Bestandteile eines XML-Dokuments sind:

Prolog: Ein optionaler Abschnitt, der die XML-Version und die verwendete Zeichencodierung definiert. Ein typischer Prolog sieht so aus:

```
1 <?xml version="1.0" encoding="UTF-8"?>
```

Abbildung 2: XML Prolog Beispielcode

Quelle: eigene Darstellung

Elemente: Die eigentlichen Daten werden in Elementen gespeichert, die mit einem Start-Tag beginnen und mit einem End-Tag abschließen. Ein Element kann weitere Elemente enthalten, was zu einer hierarchischen Struktur führt:

```
1 <buch>
2   <titel>XML-Grundlagen</titel>
3   <autor>Max Mustermann</autor>
4 </buch>
```

Abbildung 3: XML Elemente Beispielcode

Quelle: eigene Darstellung

Attribute: Jedes Element kann Attribute haben, die zusätzliche Informationen enthalten. Attribute werden im Start-Tag eines Elements definiert:

```
1 <buch genre="Lehrbuch">
2   <titel>XML-Grundlagen</titel>
3   <autor>Max Mustermann</autor>
4 </buch>
```

Abbildung 4: XML Attribute Beispielcode

Quelle: eigene Darstellung

Kommentare: Kommentare werden mit den Tags <!-- und --> eingefügt und dienen der Dokumentation oder dem Hinweis auf bestimmte Teile des Codes. Sie werden beim Parsen des Dokuments ignoriert:

```
1 <!-- Dies ist ein Kommentar -->
```

Abbildung 5: XML Kommentare Beispielcode

Quelle: eigene Darstellung

Textinhalt: Zwischen den Start- und End-Tags eines Elements kann Text enthalten sein:

```
1 <titel>XML - Grundlagen</titel>
```

Abbildung 6: XML Text Beispielcode

Quelle: eigene Darstellung

2.2.2 Verarbeiten von XML-Dateien mit Python

2.3 Datenbankentwurf und Normalisierung

2.4 Grundlagen der Datenvisualisierung

2.5 Anforderungen an modulare Softwareentwicklung

3 Analyse und Konzeption

3.1 Defintion der Anforderungen

3.2 Analyse der bestehenden Strukturen und Prozesse

3.3 Datenbankdesign und Strukturkonzeption

3.4 Grundkonzept des Benutzeroberflächen-Design

3.5 Entwurf der Applikationsarchitektu

4 Implementierung

4.1 Einlesen und Verarbeiten von XML-Daten

4.2 Implementierung der Datenbank

4.3 Entwicklung der Benutzeroberfläche

4.4 Technische Details zur Visualisierung

5 Integration und Test

5.1 Einbindung in die bestehende Systemlandschaft

5.2 Testmethoden und Durchführung

5.3 Ergebnisse der Testmethoden

6 Fazit und Ausblick

6.1 Zusammenfassung der Ergebnisse

6.2 Kritische Bewertung

6.3 Möglichkeiten für zukünftige Erweiterungen

Literatur

- [1] A. GmbH, *USTB_DWT(XCT0006 – 1)MainManual*, Aachen: AixControl GmbH, 2018.

i Anhangsverzeichnis

Erklärung

Hiermit erkläre ich, dass ich die von mir eingereichte Bachelor- / Masterarbeit ".....(Titel der Arbeit).....Belbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Ort und Datum

persönliche Unterschrift

(Name des Verfassers)