



**Hochschule für Technik  
und Wirtschaft Berlin**

*University of Applied Sciences*

Fachbereich: Ingenieurwissenschaften –  
Energie und Information  
Studiengang: Computer Engineering

# Bachelorarbeit

Entwicklung eines vollautomatisierten Embedded-Linux-Systems  
zur Ansteuerung und Auswertung eines Langzeittests

*zur Erlangung des akademischen Grades*

**Bachelor of Engineering**

von

Tim Nieter

01. Dezember 2014 – 09. Februar 2015

**- Nicht öffentlich -**

Erstbetreuer: Prof. Dr. F. Bauernöppel

Zweitbetreuer: G. Schoel

Eingereicht am: 09. Februar 2015

# Eidesstattliche Erklärung

Tim Nieter  
Rudower Straße 95  
12351 Berlin

Hiermit versichere ich, dass ich die von mir vorgelegte Arbeit selbstständig verfasst habe, dass ich die verwendeten Quellen und Hilfsmittel vollständig angegeben habe und dass ich die Stellen der Arbeit – einschließlich Tabellen und Abbildungen –, die anderen Werken oder dem Internet im Wortlaut oder dem Sinn nach entnommen sind, auf jeden Fall unter Angabe der Quelle als Entlehnung kenntlich gemacht habe.

Berlin, den 09. Februar 2015

Tim Nieter

---

(Unterschrift)

# Sperrvermerk

Die nachfolgende Bachelorarbeit mit dem Titel „Entwicklung eines vollautomatisierten Embedded-Linux-Systems zur Ansteuerung und Auswertung eines Langzeittests“ enthält vertrauliche Daten der Pepperl+Fuchs GmbH. Veröffentlichungen oder Vervielfältigungen der Arbeit – auch nur auszugsweise – sind ohne ausdrückliche Genehmigung der Pepperl+Fuchs GmbH nicht gestattet. Die Arbeit ist lediglich den Korrektoren sowie den Mitgliedern der Prüfungskommission zugänglich zu machen.

# Vorwort

**D**ie Netze sind in einer vernetzten Welt von netzartiger Bedeutung. Ohne Netze ist

Die vorliegende Arbeit ist auf Basis des Latex-Templates zu [1] erstellt worden.

[1] T. Gockel. Form der wissenschaftlichen Ausarbeitung. Springer-Verlag, Heidelberg, 2008. Begleitende Materialien unter <http://www.formbuch.de>.

# Inhaltsverzeichnis

<b>Abbildungsverzeichnis</b>	<b>I</b>
<b>Tabellenverzeichnis</b>	<b>II</b>
<b>Abkürzungsverzeichnis</b>	<b>III</b>
<b>1. Einleitung</b>	<b>1</b>
<b>2. Problemstellung</b>	<b>2</b>
2.1. Ist-Zustand . . . . .	3
2.2. Soll-Zustand . . . . .	3
<b>3. Konzept</b>	<b>4</b>
3.1. Datenbank . . . . .	4
<b>4. Testen und Validieren</b>	<b>5</b>
<b>I. Literatur und Anhang</b>	<b>i</b>
<b>Literaturverzeichnis</b>	<b>ii</b>

# **Abbildungsverzeichnis**

# **Tabellenverzeichnis**

# Abkürzungsverzeichnis

<b>SQL</b>	Structured Query Language .....	4
<b>MySQL</b>	My Structured Query Language .....	4





# 1. Einleitung

Wann immer Systeme in der realen Welt eingesetzt werden, sollen sie so zuverlässig, fehlerfrei und vorhersehbar wie möglich arbeiten. Gerade wenn diese Systeme an kritischen Punkten zum Einsatz kommen und über lange Zeiträume agieren, sind diese Eigenschaften besonders wichtig.

Um diesen Anforderungen gerecht zu werden, müssen alle Bauelemente eines solchen Systems diese Vorgaben erfüllen, denn die Zuverlässigkeit ist immer abhängig vom schwächsten Glied. Bei der Entwicklung eines Systems ist es somit entscheidend, alle Bauelemente vorher anhand der gegebenen Umstände zu qualifizieren. Dafür müssen die Grenzen ausgelotet werden, in denen sie zuverlässig betrieben werden können.

Eine dieser Grenzen ist die altersbedingte Änderung der Betriebsparameter, die sogenannte Degradation. Um das Degradationsverhalten eines Bauelementes bestimmen zu können, muss es über lange Zeiträume unter erschwerten Bedingungen betrieben und ausgewertet. Nur wenn ein Bauelement ein für die Anwendung akzeptables Degradationsverhalten aufweist, ist es für den Einsatz im Gesamtsystem geeignet.

Da ein hoher Einsatz von Ressourcen nötig ist, um jedes Bauelement individuell zu Prüfen, existieren automatisierte Teststände mit deren Hilfe der Aufwand minimiert werden soll.

Ein solcher existierender Teststand soll im Rahmen dieser Arbeit überarbeitet und optimiert werden.

## 2. Problemstellung

Gegeben ist ein existierender Degradationsteststand. Die Bedienung des bestehenden Systems bringt ein hohes Maß an individuellen Anpassungen, einen großen zeitlichen Aufwand, geringe Benutzerfreundlichkeit und Transparenz mit sich. Aus diesen Gründen soll es hingegen der Benutzerfreundlichkeit, der Effizienz und der räumlichen Ausdehnung optimiert werden. Dazu ist es nötig das vorhandene System von Grund auf zu überarbeiten.

Im folgenden Abschnitt wird auf den gegebenen Ist-Zustand und den erwarteten Soll-Zustand eingegangen.

## 2.1. Ist-Zustand

alles ist doof :(

## 2.2. Soll-Zustand

alles ist toll! :)

## 3. Konzept

Hier ist das Konzept

Mit tollen Bildern und allem was dazugehört.

### 3.1. Datenbank

Für das Speichern der Messdaten und Betriebsparameter wird ein Structured Query Language (SQL) Datenbanksystem verwendet. Dadurch wird eine hohe Performance, Flexibilität und Integrität bei der Auswertung und dem Anlegen der Daten.

Es wird auf das relationale Datenbankverwaltungssystem MySQL zurückgegriffen.

## 4. Testen und Validieren

TEST TEST 123

**Teil I.**

# **Literatur und Anhang**

