



Universität Ulm | 89069 Ulm | Germany

Titel der Arbeit

Masterarbeit

Vorgelegt von:

Heiko Foschum

heiko.foschum@gmx.de

Gutachter:

Prof. Dr. Manfred Reichert

Dr. Rüdiger Pryss

Betreuer:

Dipl.-Ing. Marc Schickler

2016

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation	1
1.2	Aufbau der Arbeit	1
2	Theoretische Grundlagen	3
2.1	(Web)Grundlagen	3
2.2	Mögliche Frameworks / Multi Plattform	3
2.2.1	Desktop	3
2.2.2	Mobile Anwendung	3
3	Implementierung	5
3.1	Entwurf	5
3.1.1	Architekturentwurf	5
3.2	Implementierung	5
3.2.1	Therapeuten Client	5
3.2.2	Patienten App	5
4	Anwendungsfälle	7
4.1	Kontrollmechanismen	7
4.1.1	Weitere denkbare Kontrollen	7
4.2	Feedbackbögen und Auswertung	7
5	Zusammenfassung und Ausblick	9
5.0.1	Zusammenfassung	9
5.0.2	Ausblick	9
	Literaturverzeichnis	11

1 Einleitung

1.1 Motivation

Um ein grundlegendes Verständnis für eine Regelung in einem eingebetteten System zu erlangen, soll im Rahmen des Labors Eingebettete Systeme eine Regelung entworfen werden, die die Drehzahl eines Lüfters auf einen gegebenen Wert konstant halten soll. Hierfür wurden zu Beginn die Grundlagen für eine Implementierung erarbeitet und Teilkomponenten erstellt, welche im späteren Regler erneut verwendet werden. Als Leitfaden und Aufgabenstellung wurde das Skript [3] bereitgestellt.

1.2 Aufbau der Arbeit

Diese Ausarbeitung soll den Ablauf vorstellen, in welchem der Regler implementiert wurde. Im Kapitel 2 werden die theoretischen Grundlagen eines Regelkreises kurz umrissen. Die VHDL-Implementierung und die dazugehörige Software wird im Kapitel 3 vorgestellt. Insbesondere soll auf die Entscheidung eingegangen werden, welche Teilkonzepte in Hardware und welche in Software realisiert wurden. Abschließend wurde die Funktionsweise des Reglers mittels Tests gezeigt, welche im Kapitel ?? beschrieben sind.

2 Theoretische Grundlagen

Im diesem Kapitel werden die Theoretischen Grundlagen der verwendeten Technologien im einzelnen betrachtet. Hierbei wird nach der "was, warum, wie"(eng. what, why, how) Methode vorgegangen.

2.1 (Web)Grundlagen

REST, HTML(GET,SET,...),Client-Server Architektur, JavaScript

2.2 Mögliche Frameworks / Multi Plattform

Eine der Grundvoraussetzungen dieser Arbeit sollte die Verfügbarkeit auf möglichst vielen Plattformen sein. Auch wenn während dieser Arbeit aus verschiedenen Gründen, die später noch aufgezeigt werden, früh klar war das die Wahl auf den sogenannten MEAN-Stack fällt, werden hier dennoch einige andere Frameworks und Plattformen aufgezeigt. Sowohl für die Desktop Anwendung des Therapeuten, also auch für die mobile Anwendung des Patienten.

2.2.1 Möglichkeiten der Anwendungsentwicklung

Plattformspezifische Entwicklung

Was sind die Vor und nachteile von Plattformspezifischer entwicklung.

Cross Plattform Entwicklung

Von Cross Plattform Entwicklung spricht man, wenn der Code der entwickelten Anwendung nicht nur für eine spezifische Plattform verwendbar ist. Es gibt etliche verschiedenen technologischen Ansätze

Die Technologie: Nativ, Web oder Hybrid?

durch verschiedene Technologien

2.2.2 Desktop

Angular, Qt, ...

2.2.3 Mobile Anwendung

Ionic(PhoneGap),Xamarin

(<http://thinkapps.com/blog/development/develop-for-ios-v-android-cross-platform-tools/>)

3 Implementierung

3.1 Entwurf

3.1.1 Architekturentwurf

3.2 Implementierung

3.2.1 Therapeuten Client

3.2.2 Patienten App

4 Anwendungsfälle

Hier sollen die verschiedenen Anwendungsfälle der Anwendung gezeigt werden. Welche Arten von Aufgaben möglich sind / wären. Die Kontrollmechanismen

4.1 Kontrollmechanismen

Welche Arten von Kontrollmechanismen schon vorhanden sind

4.1.1 Weitere denkbare Kontrollen

Ausblick auf mögliche andere

4.2 Feedbackbögen und Auswertung

5 Zusammenfassung und Ausblick

5.0.1 Zusammenfassung

5.0.2 Ausblick

Literaturverzeichnis

- [1] ALTERA (Hrsg.): *Terasic DE2-115 User Manual*. V1.02. 101 Innovation Drive, San Jose, CA 95134: Altera, 2010
- [2] ALTERA (Hrsg.): *Avalon Interface Specifications*. 15.1. 101 Innovation Drive, San Jose, CA 95134: Altera, 2015
- [3] MENHORN, Benjamin ; KEMPF, Kilian ; EHRET, Heiko: Labor Eingebettete Systeme / Institut für Eingebettete Systeme und Echtzeitsysteme, Universität Ulm. 2016 (0). – Forschungsbericht