

MASTER THESIS

Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Engineering at the University of Applied Sciences Technikum Wien - Degree Program Mechatronics/Robotics

Virtualisierung eines Echtzeit-Betriebssystems zur Steuerung eines Roboters mit Schwerpunkt auf die Einhaltung der Echtzeit

By: Halil Pamuk, BSc

Student Number: 51842568

Supervisor: Sebastian Rauh, MSc. BEng

Wien, March 21, 2024

Declaration

“As author and creator of this work to hand, I confirm with my signature knowledge of the relevant copyright regulations governed by higher education acts (see Urheberrechtsgesetz /Austrian copyright law as amended as well as the Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I hereby declare that I completed the present work independently and that any ideas, whether written by others or by myself, have been fully sourced and referenced. I am aware of any consequences I may face on the part of the degree program director if there should be evidence of missing autonomy and independence or evidence of any intent to fraudulently achieve a pass mark for this work (see Statute on Studies Act Provisions / Examination Regulations of the UAS Technikum Wien as amended).

I further declare that up to this date I have not published the work to hand nor have I presented it to another examination board in the same or similar form. I affirm that the version submitted matches the version in the upload tool.“

Wien, March 21, 2024

Signature

Kurzfassung

Erstellung einer Echtzeit-Robotersteuerungsplattform unter Verwendung von Salamander OS, Xenomai, QEMU und PCV-521 in der Yocto-Umgebung. Die Plattform basiert auf Salamander OS und nutzt Xenomai für Echtzeit- Funktionen. Dazu muss im ersten Schritt die Virtualisierungsplattform evaluiert werden. (QEMU, Hyper-V, Virtual Box, etc.) Als weiterer Schritt folgt die Anbindung eines Roboters über eine VARAN-Bus Schnittstelle. Das gesamte System wird in der Yocto-Umgebung erstellt und konfiguriert. Das Hauptziel der Arbeit ist es, herauszufinden, wie die Integration von Echtzeit-Funktionen und effizienten Kommunikationssystemen in eine Robotersteuerungsplattform die Reaktionszeit und Zuverlässigkeit von Roboteranwendungen verbessern kann

Schlagworte: Schlagwort1, Schlagwort2, Schlagwort3, Schlagwort4

Abstract

Abstract

Keywords: Echtzeit, Virtualisierung, Xenomai, VARAN

Contents

1	Introduction	1
1.1	State of the art	2
1.2	Problem and task definition	3
1.3	Objective	4
2	Methodology	5
3	Salamander 4	6
4	Results	7
5	Discussion	8
6	Summary and Outlook	9
	List of Figures	10
	List of Tables	11
	List of Code	12
	List of Abbreviations	13
A	Anhang A	14
B	Anhang B	15

1 Introduction

Diese Masterarbeit wurde in der Firma SIGMATEK GmbH & Co KG erstellt.

Firma SIGMATEK

Yocto

Salamander 4 - harte Echtzeit

Xenomai

QEMU

Trace-cmd

Kernelshark

VARAN-Bus PCV-521

Windows Ubuntu WSL

1.1 State of the art

1.2 Problem and task definition

In robotics, accurate and timely control is crucial to ensure precise movements and reliable interaction with the environment. However, existing robot control systems often have limitations in terms of real-time capabilities and communication efficiency, which can have a detrimental effect on response time and reliability. The challenge is to develop a powerful real-time robot control platform that overcomes these limitations and improves the performance of robotic applications.

1.3 Objective

The main objective of this work is to create a real-time robot control platform that integrates Salamander OS, Xenomai, QEMU and PCV-521 in the Yocto environment.

This integration is expected to lead to a significant improvement in the response time and reliability of robot applications.

2 Methodology

In diesem Abschnitt werden sämtliche theoretischen Konzepte und Randbedingungen sowie praktische Methoden, die zur Erreichung der Zielsetzung dieser Masterarbeit beigetragen haben, detailliert beschrieben.

Zu Beginn der Arbeit wurde eine ausführliche Analyse der einzelnen Virtualisierungsmöglichkeiten von Salamander 4 durchgeführt. Im Besonderen wurde hier die Virtualisierungsperformance von Ubuntu 22.04, Windows 10 und WSL unter QEMU verglichen.

First, a comprehensive literature review is conducted to understand the current trends and challenges in real-time robot control. Based on the literature study, a suitable virtualisation platform is selected.

After the selection of the virtualisation platform, the robot control platform is implemented. This step includes the installation and configuration of Salamander OS, Xenomai, QEMU and PCV-521 in the Yocto environment. Once the platform has been implemented, the robot is connected via a VARAN bus interface. Finally, the platform is evaluated to determine how the integration of real-time functions and efficient communication systems improves the response time and reliability of robot applications.

- Evaluation der Virtualisierungsplattform: Ich werde verschiedene Virtualisierungsplattformen wie QEMU, Hyper-V, Virtual Box usw. evaluieren. Dies ist ein wichtiger Schritt, um die beste Plattform für meine Anforderungen zu finden.

- Erstellung und Konfiguration des Systems in der Yocto-Umgebung: Ich werde das Yocto-Framework verwenden, um mein Embedded Linux System zu erstellen und zu konfigurieren. Yocto bietet viele Tools und Funktionen, die mir bei der Erstellung und Konfiguration meines Systems helfen können.

- Verbesserung der Reaktionszeit und Zuverlässigkeit von Roboteranwendungen: Mein Hauptziel ist es, herauszufinden, wie die Integration von Echtzeitfunktionen und effizienten Kommunikationssystemen die Reaktionszeit und Zuverlässigkeit von Roboteranwendungen verbessern kann. Ich strebe an, die Leistung und Zuverlässigkeit meiner Roboteranwendungen zu verbessern, indem ich ihre Fähigkeit verbessere, in Echtzeit auf Ereignisse zu reagieren.

- Anbindung eines Roboters über eine VARAN-Bus Schnittstelle: Ich plane, einen Roboter in mein System zu integrieren. Ich werde eine VARAN-Bus Schnittstelle verwenden, um eine schnelle und zuverlässige Kommunikation zwischen dem Roboter und dem Steuerungssystem zu gewährleisten.

3 Salamander 4

Salamander 4 ist das proprietäre Betriebssystem. Es basiert auf der Linux-Architektur und integriert zusätzlich Xenomai, eine Echtzeit-Entwicklungsumgebung [**XenomaiXenomai**].

Durch die Kombination dieser Technologien bietet Salamander OS sowohl die Flexibilität und Offenheit von Linux als auch die Echtzeitfähigkeiten von Xenomai, was es zu einer idealen Lösung für unsere spezifischen Anforderungen macht.”

4 Results

5 Discussion

6 Summary and Outlook

I am using [**blackadarHistoricalReviewCauses2016**] for citing.

List of Figures

List of Tables

List of Code

List of Abbreviations

ABC Alphabet

WWW world wide web

ROFL Rolling on floor laughing

A Anhang A

B Anhang B