



**Hochschule  
Augsburg** University of  
Applied Sciences

**Fakultät für  
Informatik**

## Bachelorarbeit

Studienrichtung  
Technische Informatik

### **Konfiguration und Optimierung des Embedded- Linux-Betriebssystem für Automotive Image Processing Unit**

**Betreuer: Mladen Kovacev**

im Fachgebiet In Kooperation mit Firma: EDAG Enginee-  
ring GmbH

Prüfer: Hubert Högl

Verfasser:

Hugues landry Nseupi Nono  
Salomon-Idler-Str 25  
86159 Augsburg  
+49 157 79552970  
landrynono60@yahoo.de  
Matrikelnr.: 2022666

Hochschule für angewandte  
Wissenschaften Augsburg  
An der Hochschule 1  
86161 Augsburg  
Telefon: +49 (0)821-5586-0  
Fax: +49 (0)821-5586-3222  
info@hs-augsburg.de

---

© 2022 Hugues landry Nseupi Nono

Diese Arbeit mit dem Titel

»Konfiguration und Optimierung des Embedded-Linux-Betriebssystem für  
Automotive Image Processing Unit - Betreuer: Mladen Kovacev«

von Hugues landry Nseupi Nono steht unter einer

*Creative Commons Namensnennung-Nicht-kommerziell-Weitergabe unter gleichen  
Bedingungen 3.0 Deutschland Lizenz (CC BY-NC-SA).*

<http://creativecommons.org/licenses/by-nc-sa/3.0/de/>



Sämtliche, in der Arbeit beschriebene und auf dem beigelegten Datenträger vorhandene, Ergebnisse dieser Arbeit in Form von Quelltexten, Software und Konzeptentwürfen stehen unter einer GNU General Public License Version 3.

<http://www.gnu.de/documents/gpl.de.html>

Die LaTeX-Vorlage beruht auf einem Inhalt unter

<http://f.macke.it/MasterarbeitZIP>.

---

## **sperrvermerk**

Die nachfolgende Arbeit enthält vertrauliche Informationen und Daten der Firma EDAG Engineering GmbH. Veröffentlichungen oder Vervielfältigungen - auch nur auszugsweise oder in elektronischer Form sind ohne ausdrückliche schriftliche Genehmigung der Firma EDAG Engineering GmbH nicht gestattet.

---

## Zusammenfassung

Abstract auf Deutsch. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

## Abstract

Abstract in English. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>V</b>
<b>Abkürzungsverzeichnis</b>	<b>VII</b>
<b>Abbildungsverzeichnis</b>	<b>VIII</b>
<b>Verzeichnis der Listings</b>	<b>IX</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation . . . . .	1
1.2 Ziel der Arbeit . . . . .	1
1.3 Überblick über den Aufbau der Arbeit . . . . .	2
<b>2 Technische Grundlagen</b>	<b>3</b>
2.1 Technische Ausgangssituation . . . . .	3
2.2 Can Bus Systeme . . . . .	3
2.2.1 Can Frame . . . . .	3
2.2.2 Can Physical Layer . . . . .	3
2.3 SPI Interface . . . . .	3
2.4 Embedded Linux . . . . .	4
2.5 Komponente eines Embedded Linux Systems . . . . .	4
2.6 Petalinux Tool Flow . . . . .	4
<b>3 Versuch Aufbau</b>	<b>5</b>
3.1 Allgemein über das Projekt . . . . .	5
3.2 Hardware Platform . . . . .	5
3.2.1 MCP251XFD CAN Controller . . . . .	5
3.2.2 Zynq UltraScale + MPSoC ZCU106 Evaluation Board . . . . .	5
3.3 Konfiguration und Bauen des Systems . . . . .	5
<b>4 Fazit und Ausblick</b>	<b>6</b>
4.1 Fazit . . . . .	6
4.2 Ausblick . . . . .	6
<b>Literaturverzeichnis</b>	<b>7</b>

<b>A Anhang</b>	<b>a</b>
A.1 Inhalt des Datenträgers . . . . .	a

## **Abkürzungsverzeichnis**

ACL .....	Agent Communication Language
AMS .....	Agent Management System
Cougaar .....	Cognitive Agent Architecture
CRUD .....	Create, read, update and delete
CT .....	Container Table
DARPA .....	Defense Advanced Research Projects Agency
FIPA .....	Foundation for Intelligent Physical Agents
FPGA .....	Field Programmable Gate Array
GADT .....	Global Agent Descriptor Table
JADE .....	Java Agent Development Framework
JVM .....	Java Virtual Machine
KISS .....	Keep It Small & Simple
LADT .....	Local Agent Descriptor Table
MAS .....	Multi Agent System
P2P .....	Peer-to-Peer
PDF .....	Portable Document Format
SMAS .....	Scala Multi Agent System
VM .....	Virtual Machine
XML .....	Extensible Markup Language

## **Abbildungsverzeichnis**



## **Verzeichnis der Listings**

2.1	Hello World . . . . .	<a href="#">4</a>
-----	-----------------------	-------------------

# 1 Einleitung

## 1.1 Motivation

Aufgrund der Einsätze von immer mehr Geräten, deren Funktionen uns das Leben erreichen. Egal, ob die Waschmaschine zu Hause, der Drucker in unseren Büros, oder die Kaffeemaschine in der Kantine, befindet sich in alle diese Geräte kleine Computer, die ihre Aufgabe erledigen. Aber durch die gestiegene Rechenleistung und die erweiterten Kapazitäten eines Mikroprozessors werden die Aufgaben von kleine Computer immer komplexer. Es besteht dann die Möglichkeit, ein vollwertiges Betriebssystem einzusetzen. Hier hat sich Linux durch die vielseitige Anwendbarkeit und das offene Ökosystem für Embedded Devices besonders bewährt. Am Edag Engineering GmbH, wurde im Rahme des internen Projekts, ein Image Processing Unit (IPU) Hardware Plattform auf Basis des Kria KV260 FPGA(Field Programmable Gate Array) entwickelt. Auf dieser Plattform wird dann aufgrund der Komplexität des Projekts ein Linux Betriebssystem eingesetzt, mit dem die 8 wesentlichen Anwendungen des Projekts verwalten werden.

## 1.2 Ziel der Arbeit

Angesichts des Engpasses an Halbleitern, dem die Welt in den letzten Monaten gegenübersteht, wird es immer schwieriger, hochwertige Bauteile wie das Kria KV260, der für das Projekt verwendet wird, zu finden. Statt auf der einzigen Platine des Unternehmens, musste ich meine Arbeit auf einer alternativen Platine durchführen. Das erste Ziel dieser Arbeit war es, ein in der Firma entwickelte CAN FD Controller (mcp251xfd), der über SPI mit einem Zynq UltraScale + MPSoC ZCU106 Board verbunden ist, in Betrieb zu nehmen, damit verschiedenen Can Node vom Linux angesprochen wird.

Im Anschluss musste ich 3 von den in der Firma entwickelte Applikationen, im Linux bauen, damit das System automatisch mit den Anwendungen bootet. Dafür müsste ich Rezepte schreiben, die sich darum kümmern werden, die Applikationen zu konfigurieren, zu kompilieren und zu installieren.

## 1.3 Überblick über den Aufbau der Arbeit

Diese Arbeit lässt sich in 4 Hauptkapitel aufteilen:

- **Die Einleitung:** In der Einleitung werden, die Motivation, das Ziel der Arbeit und ein gesamter Überblick auf dem Ablauf der Arbeit behandeln.
- **In den technischen Grundlagen** wird zuerst erklärt, wie das System (CAN Controller und die Zynq Mp Plattform) gebaut und funktionieren soll. Des Weiteren werden, der CAN Bus System und die SPI Interface erklärt. Dann wird dem Grundprinzip von Embedded Linux Systemen und deren Komponenten erläutert. Zum Schluss erfolgt, die Beschreibung der Petalinux Tools Flow, welches der Build System, der verwendet wird, um Linux Distribution für Xilinx Bausteinen zu kompilieren.
- **Im Versuch Aufbau** wird das Projekt, in dem ich gearbeitet habe dargestellt, dann folgt eine tiefe beschreibung der Hardware. Anschließen wird detaliert auf verschiedenen Schritte für das Bauen des System eingegangen.
- **Im Kapitel Fazit und Ausblick** werden aufgetretene Probleme und Herausforderungen erläutert, es wird analysiert, wie das Ergebnis von dem Ziel entfernt ist. Und anschließend wird ein Ausblick auf die möglichen Verbesserungen gegeben.

## 2 Technische Grundlagen

Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet. Lorem ipsum dolor sit amet, consetetur sadipscing elitr, sed diam nonumy eirmod tempor invidunt ut labore et dolore magna aliquyam erat, sed diam voluptua. At vero eos et accusam et justo duo dolores et ea rebum. Stet clita kasd gubergren, no sea takimata sanctus est Lorem ipsum dolor sit amet.

### 2.1 Technische Ausgangssituation

### 2.2 Can Bus Systeme

#### 2.2.1 Can Frame

#### 2.2.2 Can Physical Layer

### 2.3 SPI Interface

- vero eos
- accusam
- justo duo dolores
- ea rebum

Beispiel für Quelltexte (Siehe Listing [2.1](#)):

**Listing 2.1:** Hello World [coder blog](#) (2009)

---

```
1 public class HelloWorld
2 {
3     public static void main(String[] args)
4     {
5         System.out.println("HelloWorld");
6     }
7 }
```

---

## 2.4 Embedded Linux

## 2.5 Komponente eines Embedded Linux Systems

## 2.6 Petalinux Tool Flow

## **3 Versuch Aufbau**

### **3.1 Allgemein über das Projekt**

### **3.2 Hardware Platform**

#### **3.2.1 MCP251XFD CAN Controller**

#### **3.2.2 Zynq UltraScale + MPSoC ZCU106 Evaluation Board**

### **3.3 Konfiguration und Bauen des Systems**

## **4 Fazit und Ausblick**

### **4.1 Fazit**

### **4.2 Ausblick**

## Literaturverzeichnis

[coder blog 2009]

BLOG coder: *Java-Tutorial: HelloWorld.* 2009 [http://coder-blog.de/  
java-tutorial-helloworld](http://coder-blog.de/java-tutorial-helloworld) 2.1



Ich, Hugues landry Nseupi Nono, Matrikel-Nr. 2022666, versichere hiermit, dass ich die vorliegende Arbeit mit dem Thema

*Konfiguration und Optimierung des Embedded-Linux-Betriebssystem für  
Automotive Image Processing Unit - Betreuer: Mladen Kovacev*

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Augsburg, den 2. März 2022

---

HUGUES LANDRY NSEUPI NONO

## **A Anhang**

### **A.1 Inhalt des Datenträgers**

Der dieser Arbeit beigelegte Datenträger beinhaltet zusätzliche Materialien. Neben der Arbeit selbst im Portable Document Format (PDF) befinden sich sowohl die Sources der Implementierungen als auch die lauffähigen Pakete.

**`./all-my-packages/`**

Sources der Packages

**`./Architektur/`**

UML-Diagramme der Architektur

**`./Thesis_Vorname_Nachname_123456.pdf`**

PDF Version dieser Arbeit

**`./ThesisVM.ova`**

Virtual Box Image mit lauffähiger Demoumgebung