



# Projekt Zephyr

Echtzeit-OS für das Internet der Dinge

**Pflichtenheft**

Studiengang: Elektro- und Kommunikationstechnik  
Institut: Berner Fachhochschule  
Autoren: Aaron Schmocker, David Wyss  
Betreuer: Martin Aebersold  
Auftraggeber: Martin Aebersold  
Experten: Martin Aebersold  
Datum: 23.10.2016

# Inhaltsverzeichnis

<b>1</b>	<b>Management Summary</b>	<b>1</b>
<b>2</b>	<b>Einleitung</b>	<b>2</b>
<b>3</b>	<b>Ressourcen, Infrastruktur und Betriebsmittel</b>	<b>3</b>
3.1	Räumlichkeiten . . . . .	3
3.2	Software und Betriebssysteme . . . . .	3
3.3	Hardware . . . . .	3
3.4	Entwicklungs und Testwerkzeuge . . . . .	3
3.5	Dokumentation . . . . .	4
<b>4</b>	<b>Definition der Aufgaben</b>	<b>5</b>
4.1	Aufgabenbeschreibung . . . . .	5
4.2	Anforderungen . . . . .	5
<b>5</b>	<b>Bedingungen</b>	<b>6</b>
<b>6</b>	<b>Freigabe</b>	<b>7</b>
	<b>Glossar</b>	<b>8</b>

# Versionen

Version	Datum	Status	Bemerkungen
0.1	15.09.2016	Entwurf	Erster Entwurf
0.2	16.09.2016	Entwurf	Management Summary
0.3	18.09.2016	Entwurf	Einleitung
0.4	19.09.2016	Entwurf	Vorgehen
0.5	20.09.2016	Entwurf	Hardware
0.6	03.10.2016	Entwurf	Software
0.7	04.10.2016	Entwurf	Systemtest
0.8	06.10.2016	Entwurf	Interpretation
0.9	10.10.2016	Entwurf	Schlussfolgerung
1.0	20.10.2016	Final	Korrekturen

# 1 Management Summary

Die Linux Foundation hat mit dem Projekt Zephyr mit der Entwicklung eines Echtzeit-Betriebssystems für das Internet der Dinge (IoT) begonnen. Zephyr ist ein Open-Source-Betriebssystem mit dem Ziel ein solides OS für IoT Geräte mit geringen Ressourcen bereitzustellen. Es nutzt eine echtzeitfähige Kombination aus Nano- und Microkernel. Im Gegensatz zu einem Linux Kernel benötigt Zephyr nur zwischen 8 und 512 KByte an Arbeitsspeicher. Aktuell werden folgenden Plattformen unterstützt: x86, ARM und ARC EM4

## 2 Einleitung

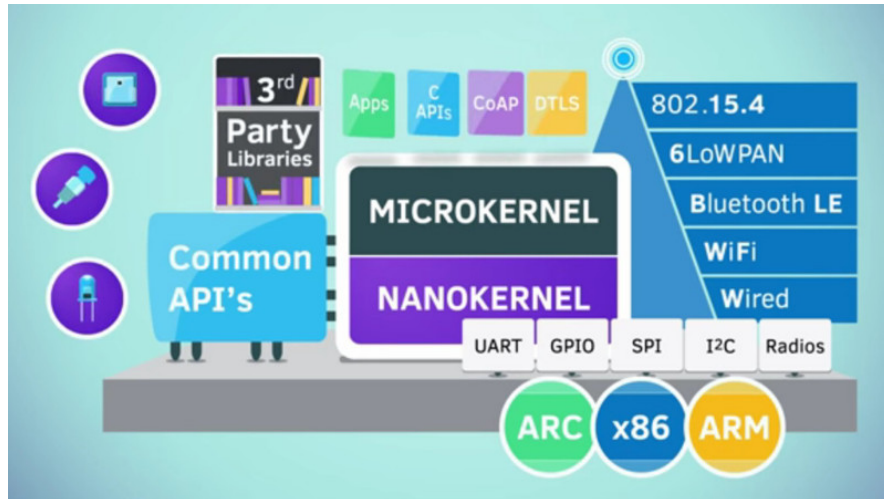


Abbildung 2.1: Komponenten und Übersicht über das Zephyr RTOS

In dieser Projektarbeit soll zuerst ein Vergleich der Eigenschaften ähnlicher Betriebssysteme wie FreeRTOS, RIOT, Kontiki, usw. vor allem bezüglich der Unterstützung der verschiedenen Netzwerkprotokolle gemacht werden. Im Fokus steht dabei das n Zu Demonstrationszwecken soll am Schluss mit dem ausgewählten Board eine Demo-App entwickelt werden.

Die Ziele sind:

- Einarbeitung in das Betriebssystem Zephyr.
- Erstellen einer Vergleichstabelle mit den wichtigsten Eigenschaften der verschiedenen RTOS.
- Evaluation geeigneter Boards für das Zephyr Betriebssystem für eine IoT
- Entwicklung einer Demonstrationsapplikation, basierend auf dem ausgewählten Board

# 3 Ressourcen, Infrastruktur und Betriebsmittel

## 3.1 Räumlichkeiten

Für die Arbeit steht ein Arbeitsplatz im Raum T208 zur Verfügung.

## 3.2 Software und Betriebssysteme

Da es sich bei Zephyr um ein RTOS handelt, wird die Demoapplikation in C geschrieben. Zur Entwicklung und Dokumentation wird ausschliesslich auf offene Software gesetzt.

- Ubuntu und Arch Linux
- Zephyr und Zephyr SDK
- GIT
- Latex
- LibreOffice

## 3.3 Hardware

Folgende Hardware wird für das Projekt zur Verfügung gestellt:

- nRF52 Development Kit von Nordic
- ST-Link/v2
- Hardware und Computer des Raumes T208:

## 3.4 Entwicklungs- und Testwerkzeuge

- Eclipse Neon mit Gnu ARM Plugin
- SDK und Libraries von Nordic Semiconductors für das NRF52 Evalboard
- SEGGER J-Link

## 3.5 Dokumentation

Es wird eine Projektdokumentation erwartet, welche die Entwurfsphase sowie die Realisierung und die gemachten Tests aufzeigt. Alle Dokumente werden auf einem Git-Repository abgelegt. Zur Dokumentation wird auf Latex gesetzt.

# 4 Definition der Aufgaben

## 4.1 Aufgabenbeschreibung

Im Rahmen der Projektarbeit sollen die Möglichkeiten des Zephyr-OS im Hinblick auf Anwendungen im Bereich Internet der Dinge (IoT) abgeklärt werden. Besonderer Fokus gilt hierbei dem Vergleich von Zephyr zu bereits bestehenden Realtime Operating Systems (RTOS). Dieser Vergleich soll tabellarisch ersichtlich gemacht werden. Weiter soll im Rahmen der Arbeit eine Demonstrationsapplikation entwickelt werden, auch mit dem Fokus auf das IoT.

## 4.2 Anforderungen

Minimale Anforderung
<ul style="list-style-type: none"><li>- Aufsetzen der Entwicklungsumgebung<ul style="list-style-type: none"><li>- Aufsetzen einer Ubuntu-VM</li><li>- Einrichten eines Gitrepositories für die Arbeit und Dokumentation</li><li>- Inbetriebnahme der GNU ARM Toolchain unter Eclipse Neon</li><li>- Installation der SDK und Libraries für das Nordic nRF52-DK</li></ul></li><li>- Evaluation der Fähigkeiten von Zephyr als RTOS für das Internet der Dinge<ul style="list-style-type: none"><li>- Vergleich von Kommunikationsprotokollen mit bestehenden RTOS</li><li>- Vergleich der Codebasis mit bestehenden RTOS</li><li>- Vergleich der Leistungsfähigkeit und Binarygrösse mit bestehenden <b>RTOS!</b> (<b>RTOS!</b>)</li></ul></li><li>- Entwicklung einer Demonstrationsapplikation<ul style="list-style-type: none"><li>- Mit Fokus auf Anwendungen rund um das Internet der Dinge</li><li>- Nutzen von Bluetooth Low-Energy (BLE)</li><li>- Nutzen des nRF52-DK</li></ul></li></ul>
Optionale Erweiterungen
<ul style="list-style-type: none"><li>- Portierung des Zephyr-Kernels auf ESP8266</li><li>- Entwicklung einer Demoapplikation im Bereich Wireless-Lan</li></ul>



## 5 Bedingungen

In dieser Projektarbeit sollen die Fähigkeiten des Zephyr-ROTS in bestehender Form ausgetestet werden. Spezielle Beachtung soll den für das Internet der Dinge wichtigen Kommunikationsprotokollen wie BLE und Near Field Communication (NFC) geschenkt werden. Bei der Evaluation und Programmierung der Demoapp sind keine Grenzen gesetzt.

## 6 Freigabe

Datum, Unterschrift:

.....

Martin Aebersold

Datum, Unterschrift

.....

Aaron Schmocker

Datum, Unterschrift

.....

David Wyss

# Glossar

<b>API</b>	Application Programming Interface
<b>CPU</b>	Central Processing Unit
<b>RTOS</b>	Realtime Operating Systems
<b>HTTPS</b>	Hypertext Transfer Protocol Secure
<b>BLE</b>	Bluetooth Low-Energy
<b>NFC</b>	Near Field Communication
<b>IoT</b>	Internet der Dinge