

# Konzeptionierung und prototypische Implementierung einer echtzeitfähigen Restbussimulation

Sven Eisenhauer, Darmstadt, 28.02.2012

# Agenda

- Definitionen
- Motivation
- Ausgangsbasis
- Entwurf
- Prototyp
- Ergebnisse
- Ausblick

# Definitionen

## Restbussimulation

- Software
- Simuliert Steuergeräte
- Test und Entwicklung

## Echtzeitfähig

- „Rechtzeitig“
- Vorhersagbar
- Zeitdomäne

## CAN

- Broadcast-Bus
- Arbitrierung
- Nachrichtenpriorität
- OSI Layer 1,2
- Bit-Stuffing
  - Taktrückgewinnung
  - Nachrichtendauer

# Motivation

- Entwicklung von Steuergeräten
- ca. 70 Steuergeräte im Fahrzeug
- ca. 15.000 Botschaften pro Sekunde
- Abhängigkeiten
- Parallele Entwicklung und Tests
- Analyse über die Zeit

# Motivation

## Restbussimulation

- Desktop
- Simulation
- Analyse
- Test

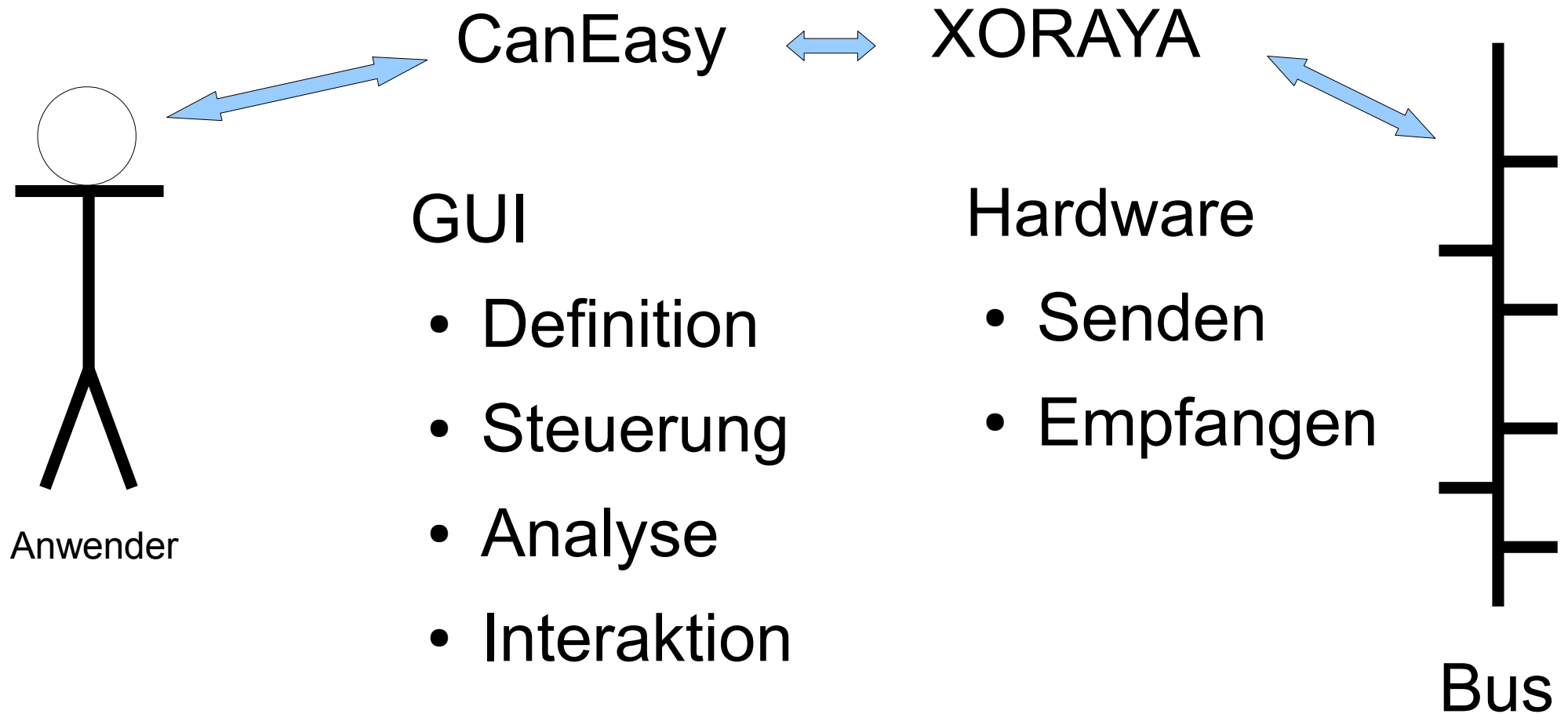
## Echtzeitsystem

- Vorhersagbar
- Zeitlich exakt
- Autark



**Synergie**

# Zielsetzung



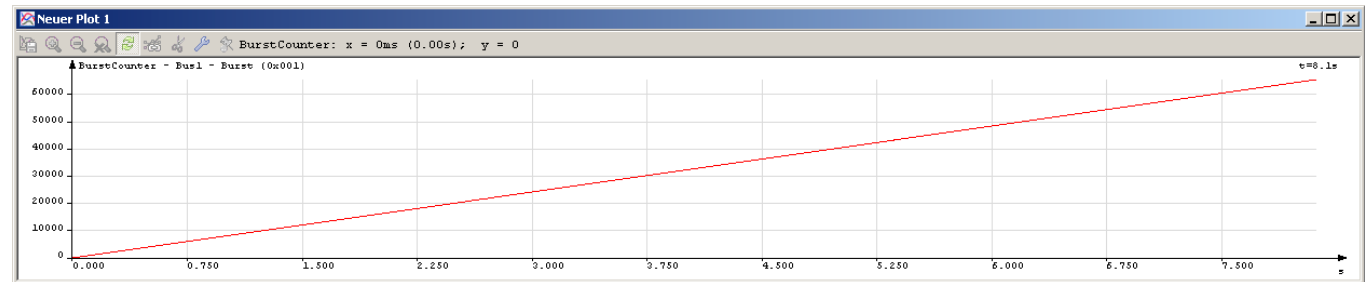
# CanEasy

Software für Microsoft Windows (little endian)

Plugin-Architektur

GUI-Elemente

- Editoren
- Traces
- Plots
- Panels
- ...



# XORAYA CONNECT

## Hardware

PowerPC (big endian)

CAN, FlexRay, MOST

Ethernet

100ns Zeitstempel



## Software

Linux 3.0 mit  
PREEMPT\_RT

C++ - SDK

Eclipse IDE

Targetdebugger

# Entwurf

## **CanEasy**

Code-Generierung

Transfer

Steuerung

Netzwerkclient

## **XORAYA**

Plugin-Architektur

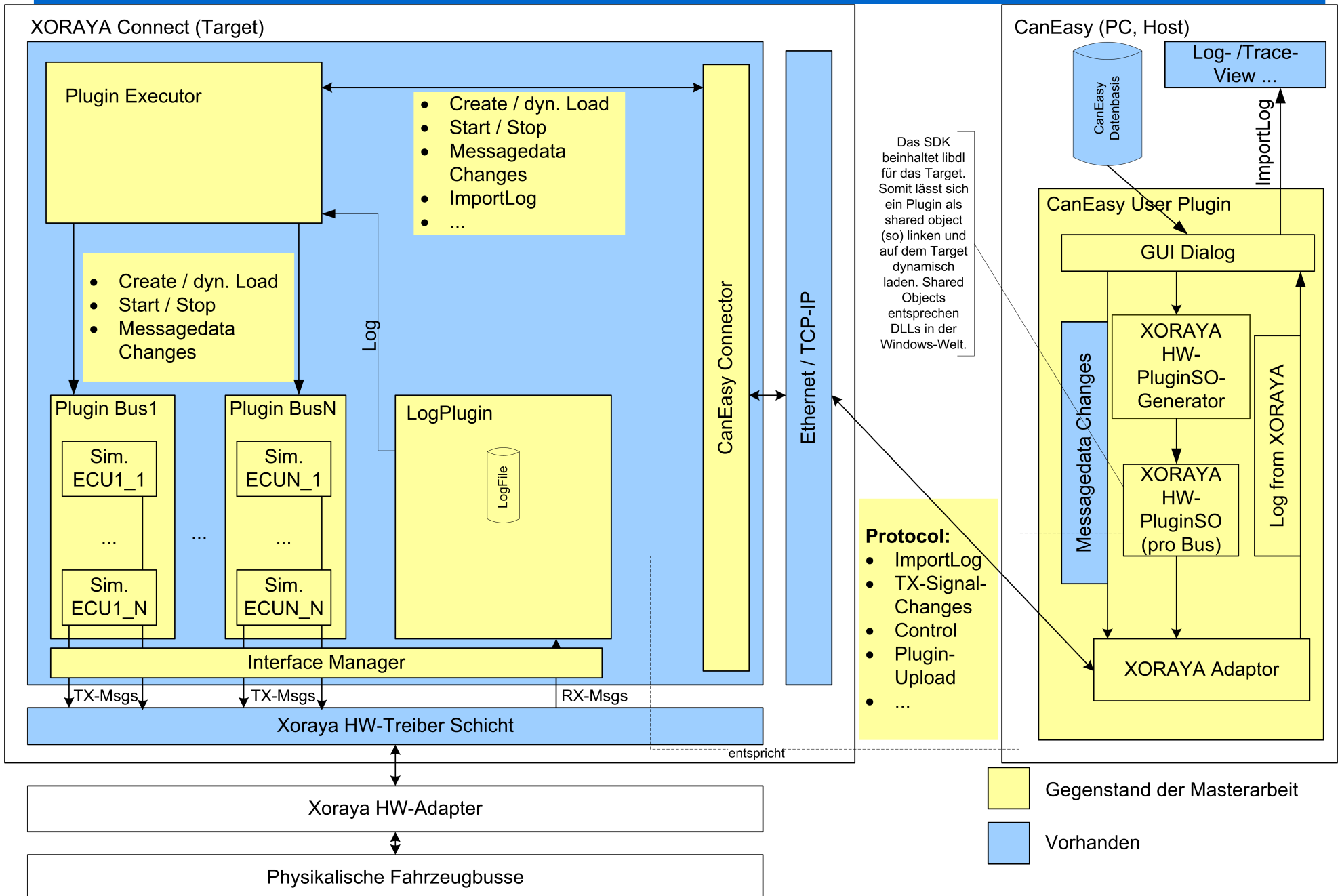
Versand

Aufzeichnung

Transfer

Netzwerkserver

# Architektur



# Schnittstellen

## IPlugin

```
+vRun(): void  
+pxGetCanTxMessage(in u32MsgIndex:const uint32_t): tpstCanTxMessage  
+u32GetNumOfCanTxMessages(): uint32_t  
+i32GetCanInterfaceHandle(): int32_t  
+boAutoload(): bool  
+boIsLogger(): bool  
+enGetLog(in pcRequest:const char*,out pcResponse:char*): tenRetCod  
+vInit(in pvExecutor:void*,in pvIfMan:void*,  
       in i32Interface:int32_t): void  
+vLogMessage(in stLogMessage:const tstLogMessage&): void
```

## tstCanTxMessage

```
+u32TxCycleMilliseconds: uint32_t  
+u32MillisecondCounter: uint32_t  
+u32CanId: uint32_t  
+au8Data[8]: uint8_t  
+u8Dlc: uint8_t
```

# Tasks auf dem Target

## Maintask

- Höchste Priorität
- 1 ms Hardwaretimer
- Auslesen  
Empfangsqueue
- Versenden zyklischer  
Nachrichten

## Netzwerktask

- Mittlere Priorität
- Asynchrone  
Kommunikation

## Logtask

- Niedrigste Priorität
- Schreiben der  
Aufzeichnung

# Netzwerkkommunikation

- FTP für große Dateien
- Binäres Protokoll zur Steuerung
- Asynchron basierend auf Boost.Asio-Bibliothek
- Plattformunabhängig
- Wiederverwendbar auf Applikationsebene durch Datencontainer und -ströme

# Prototyp

- Highspeed CAN (1 Mbit/s)
- Zyklischer Versand
- Aufzeichnung
- Cross-Compile aus Datenbasis
- Shared objects als Transferobjekte
- Kommunikation über Ethernet

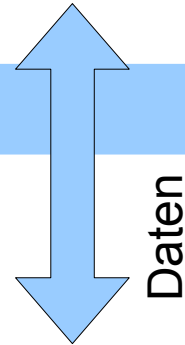
# Netzwerkcommunication

## Applikation ②

Konstante Codier-/Decodier-Objekte

Lesen aus Puffer ①

③ Schreiben schreiben in Puffer



Datencontainer (indiziert)

Netzwerknachrichten  
(Strukturen)

InStream

Empfangspuffer  
(bytes)

OutStream

Sendepuffer  
(bytes)

# Echtzeitlinux

## PREEMPT\_RT Kernel Patches

- Große Teile des Kernels preemptible
- Ersatz von Spinlocks durch Mutexe
- Mutexe mit Priority Inheritance
- Tasks höher als Interrupt-Handler priorisierbar
- Hochauflösende Timer

# Intrusive Container

- Boost Bibliothek
- Multiset mit Nachrichten-ID als Schlüssel
- ✓ Schnittstelle entsprechend STL-Containern
- ✓ Keine Kopien der Elemente
- ✓ Bessere Performance als STL-Container
- x Anpassung der enthaltenen Objekte

# Ergebnisse

- Verifikation
  - Basierend auf Microcontroller
  - Zeitstempelabweichung
  - Burst
- WCRT, Schedulability test
- System unter Last
- Vergleich mit anderen CAN-Adapttern



Atmel AT32UC3C-EK  
(Quelle: Atmel)

# Proof Of Concept

- Shared objects als Transferobjekte
- Einhaltung der Echtzeitanforderungen
- Systemlast ohne Einfluss
- Berechnete maximale Buslast erzeugbar
- Exakte Sendezyklen auf der XORAYA Connect

# Ausblick

- Unterstützung weiterer Schnittstellentypen
- Verlagerung weiterer zeitkritischer Funktionalität auf die XORAYA Connect
- Schedulability Tests
- Transferobjekte inkl. Methoden

**Vielen Dank!**

# Fragen!?