

Konzeptionierung und prototypische Umsetzung einer Echtzeitbasierten Restbussimulation mittels Mikrocontrollerunterstützung

Abstract

Die zu erstellende Arbeit beschreibt ein Konzept zur Simulation eines Fahrzeug-Busses auf einem Mikrocontroller. Als Fahrzeug-Bus wird hierbei exemplarisch der CAN-Bus betrachtet. Dieser Bus verbindet mehrere Steuergeräte in einem Fahrzeug. Für Test- und Entwicklungszwecke stehen aber nicht alle Steuergeräte eines Fahrzeugs physikalisch zur Verfügung. Die benötigten Nachrichten dieser fehlenden Steuergeräte erzeugt eine Software zur Restbussimulation.

Die Firma Schleissheimer GmbH bietet zu diesem Zweck die Software CanEasy an. Hierbei handelt es sich um eine Software für das Betriebssystem Microsoft Windows, mit der der Benutzer Fahrzeugbusses mit allen Steuergeräten und deren Nachrichten definieren kann. Dabei unterscheidet CanEasy zwischen realen und simulierten Steuergeräten. Für simulierte Steuergeräte generiert CanEasy die zugehörigen Nachrichten und versendet diese auf dem Bus. Die Nachrichten realer Steuergeräte zeichnet CanEasy auf. Weiterhin bietet CanEasy Möglichkeiten zur Wiedergabe bestehender Aufzeichnungen und zur Analyse bzw. Auswertung von Aufzeichnungen. Die Konfiguration der Busse mit allen Steuergeräten, Nachrichten und Signalen in den Nachrichten speichert CanEasy in der sog. Datenbasis. Über eine grafische Baumansicht bietet es dem Benutzer komfortable Möglichkeiten zum Editieren der Datenbasis (siehe Abbildung 1: Grafische Oberfläche CanEasy).

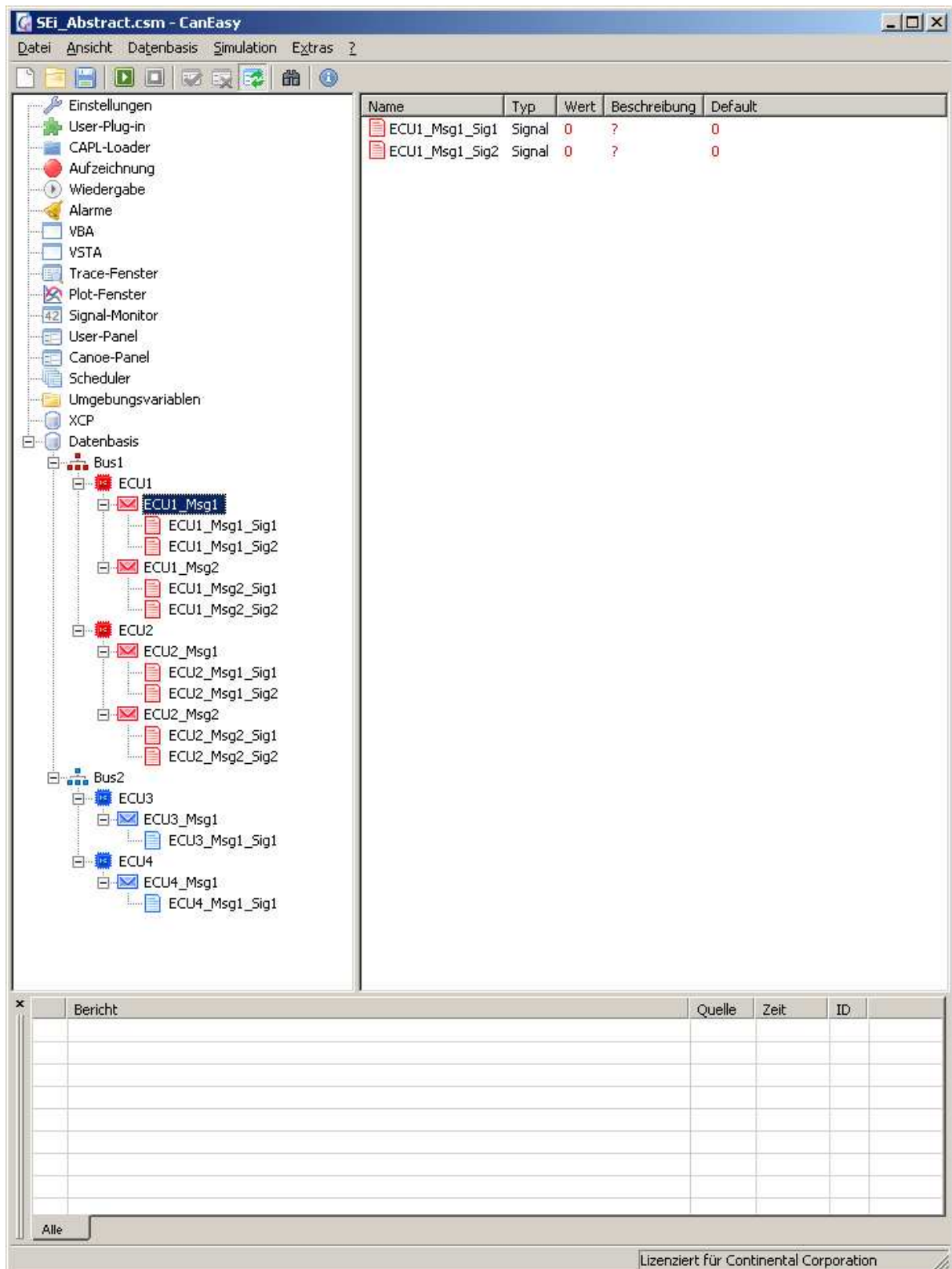


Abbildung 1: Grafische Oberfläche CanEasy

Bisher verwendet CanEasy für die Bus-Kommunikation Adapter, die direkt an den Rechner, auf dem CanEasy ausgeführt wird, angeschlossen sind.

Die Firma X2E bietet mit dem Produkt Xoraya Connect (Abbildung 2: X2E Xoraya Connect und <http://www.x2e.de/en/products/xoraya-connect/xoraya-connect.php>) ein Gerät mit vielfältigen Anschlussmöglichkeiten an Bus- und Netzwerk-Systeme, unter anderem Ethernet und CAN.



Abbildung 2: X2E Xoraya Connect

Weiterhin verfügt das Geräte über eine PowerPC-CPU, auf der ein Linux-Betriebssystem ausgeführt wird. Über ein SDK ist es möglich, eigene Anwendungen auf der Xoraya Connect auszuführen. Die Masterarbeit zeigt ein Konzept zum Versand und Empfang von Bus-Nachrichten durch CanEasy unter Verwendung der Xoraya Connect. Die Umsetzung verfolgt dabei das Ziel, die zeitkritischen Teile von CanEasy als Anwendung auf der Xoraya Connect auszuführen. Dieser Ansatz soll Netzwerk-Latenzen im Ethernet beim Versenden verhindern. Außerdem ermöglicht dieser Ansatz ein zeitlich genaues Versenden von Nachrichten. Empfangene Nachrichten erhalten einen Zeitstempel, der auf 100 Nanosekunden genau ist, was zeitlich sehr präzise Aufzeichnungen ermöglicht.

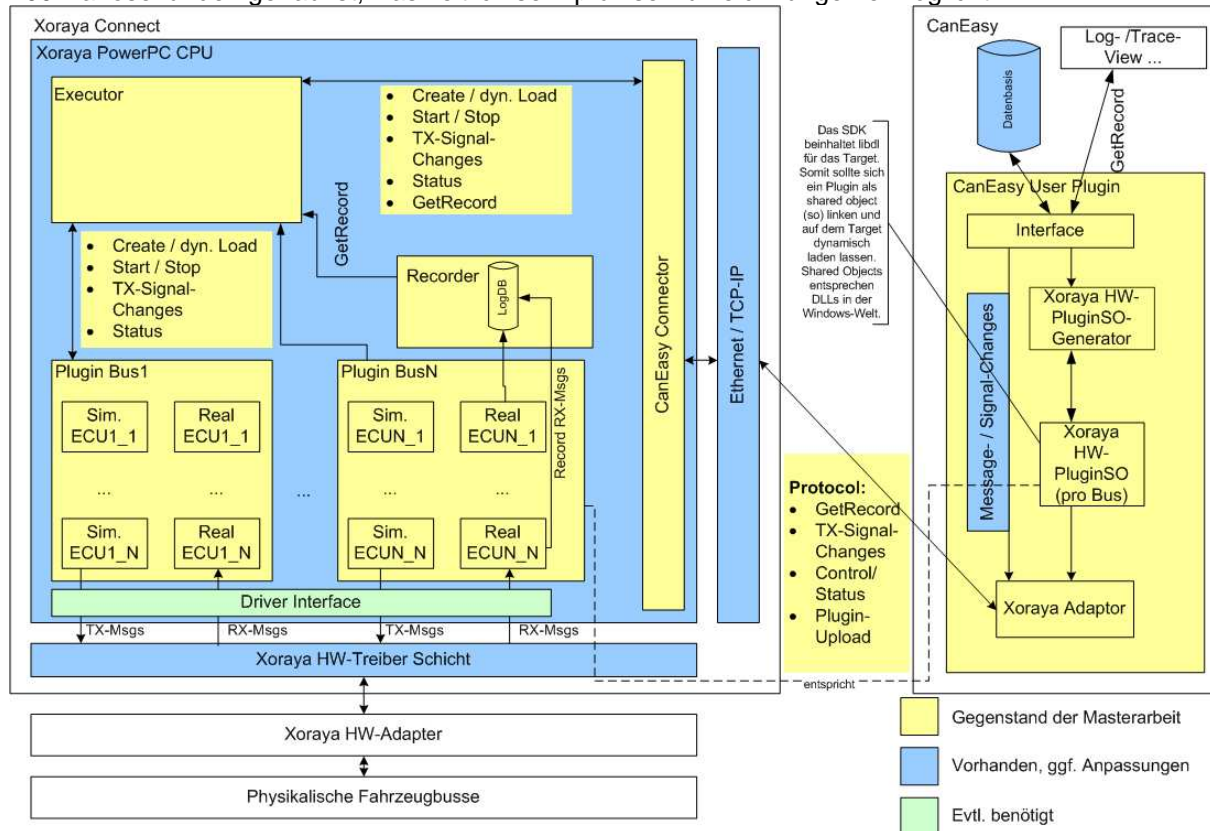


Abbildung 3: Grobkonzept CanEasy Xoraya Integration

Abbildung 3: Grobkonzept CanEasy Xoraya Integration zeigt einen ersten groben Entwurf zur Integration von CanEasy und der Xoraya Connect. Aus diesem Entwurf ergeben sich nachfolgend aufgelistete Artefakte, die im Rahmen der Masterarbeit erstellt werden:

- Netzwerk-Protokoll:
 - Übertragung eines Xoraya-Plugins auf die Xoraya-Hardware.
 - Steuerung der Plugins auf der Xoraya-Hardware von CanEasy aus:
 - Start
 - Stop
 - Status
 - Übertragung von Bus-Aufzeichnungen von der Xoraya-Hardware an CanEasy

- Übermittlung geänderter Sende-Signale von CanEasy aus an das entsprechende Xoraya-Plugin
- Xoraya Plugin Generator:
 - CanEasy Plugin
 - Generiert C/C++ Quellcode aus der CanEasy-Datenbasis, d. h. der Bus-Definition mit Steuergeräten und Messages
 - Übersetzt den Quellcode in ein Shared Object für das Zielsystem (cross-compile)
 - Übergibt shared object an Adaptor
- Xoraya Adaptor:
 - CanEasy Plugin
 - Übernimmt Datenaustausch zwischen CanEasy und Xoraya-Hardware
 - Implementiert Protokoll
 - TCP-IP-Socket-Kommunikation über Ethernet
 - Sendet Plugins, Steuernachrichten, Signaländerungen und Aufzeichnungsanfragen an den Connector
 - Empfängt Status-Meldungen und Aufzeichnungen vom Connector und leitet diese an CanEasy weiter
- CanEasy Connector:
 - Läuft auf Xoraya-Hardware
 - Übernimmt Datenaustausch zwischen Xoraya-Hardware und CanEasy
 - Implementiert Protokoll
 - TCP-IP-Socket-Kommunikation über Ethernet
 - Nutzt X2E embedded SDK
 - Empfängt Plugins, Steuernachrichten, Signaländerungen und Aufzeichnungsanfragen vom Adaptor
 - Sendet Status-Meldungen und Aufzeichnungen an den Adaptor
- Executor
 - Läuft auf Xoraya-Hardware
 - Instanziiert, startet und stoppt Xoraya-Plugins
 - Übergibt geänderte Sende-Signale an das entsprechende Plugin
 - Empfängt Aufzeichnungsanfragen vom Connector
 - Liest angefragte Bus-Aufzeichnungen vom Recorder und übermittelt diese an den Connector
 - Nutzt X2E embedded SDK
- Xoraya-Plugins
 - Zeichnen empfangene Nachrichten vom Bus auf für Steuergeräte, die in der CanEasy-Datenbasis als "real" definiert wurden.
 - Leiten aufgezeichnete Nachrichten an den Recorder weiter
 - Senden Nachrichten auf den Bus für Steuergeräte, die in der CanEasy-Datenbasis als "simuliert" definiert wurden.
 - Nutzt X2E embedded SDK
- Recorder
 - Speichert aufgezeichnete Nachrichten dauerhaft
 - Liefert Aufzeichnungen an den Executor
 - Nutzt X2E embedded SDK
 - evtl. sqlite
- Driver Interface
 - evtl.
 - Läuft auf Xoraya-Hardware
 - Nutzt X2E embedded SDK
 - Einfacher Wrapper für Plugins, entkoppelt Plugins von Xoraya-API
- API
 - Definiert Schnittstelle für Xoraya-Plugins

- start
- stop
- setSendSignal
- getStatus
- Definiert Schnittstelle für Recorder
 - saveRecord
 - getRecords
- Definiert Schnittstelle für Executor
 - createPlugin
 - startPlugin
 - stopPlugin
 - getStatus
- Definiert Schnittstelle für Connector
 - sendStatus
 - sendRecords

Tabelle 1: Zeitplanung

KW	Datum	Phase	Tätigkeit
36	29.08.2011	Prototyp Ecartec	Grobentwurf Gesamtsystem, Inbetriebnahme Hardware, Entwicklungsumgebung X2E, Festlegung Funktionsumfang Ecartec-Prototyp
37	05.09.2011	Prototyp Ecartec	Spezifikation API, Protokoll
38	12.09.2011	Prototyp Ecartec	Spezifikation Connector, Executor, Adaptor
39	19.09.2011	Prototyp Ecartec	Implementierung Prototyp Connector, Adaptor
40	26.09.2011	Prototyp Ecartec	Implementierung Prototyp Recorder, Executor
41	03.10.2011	Prototyp Ecartec	Implementierung Prototyp Log-Plugin
42	10.10.2011	Prototyp Ecartec	Implementierung CanEasy-Anbindung
43	17.10.2011	Prototyp Ecartec	Integration, Test Ecartec-Prototyp
43	18.10.2011		Ecartec
43	19.10.2011		Ecartec
43	20.10.2011		Ecartec
44	24.10.2011	Erweiterte Funktionen	Dokumentation der bisherigen Ergebnisse
45	31.10.2011	Erweiterte Funktionen	Dokumentation der bisherigen Ergebnisse Literaturrecherche, 1. Version zum Review
46	07.11.2011	Erweiterte Funktionen	Implementierung Erweiterung Connector, Adaptor
47	14.11.2011	Erweiterte Funktionen	Implementierung Erweiterung Connector, Adaptor
48	21.11.2011	Erweiterte Funktionen	Implementierung Erweiterung Executor, Plugin-Generator
49	28.11.2011	Erweiterte Funktionen	Implementierung Erweiterung Executor, Plugin-Generator
50	05.12.2011	Erweiterte Funktionen	Implementierung Erweiterung Executor, Plugin-Generator
51	12.12.2011	Erweiterte Funktionen	Implementierung Erweiterung Executor, Plugin-Generator
52	19.12.2011	Erweiterte Funktionen	Implementierung Erweiterung Executor, Plugin-Generator
53	26.12.2011	Erweiterte Funktionen	Implementierung Erweiterung Executor, Plugin-Generator
1	02.01.2012	Erweiterte Funktionen	Implementierung Erweiterung Executor, Plugin-Generator
2	09.01.2012	Erweiterte Funktionen	Schreiben
3	16.01.2012	Erweiterte Funktionen	Schreiben
4	23.01.2012	Erweiterte Funktionen	Schreiben
5	30.01.2012	Erweiterte Funktionen	Schreiben
6	06.02.2012	Erweiterte Funktionen	Korrektur, Drucken, Binden
7	13.02.2012		
7	14.02.2012		Abgabe

Eine erste Zeitplanung für die Umsetzung zeigt Tabelle 1: Zeitplanung.