



Einsatz von Android als Plattform im Fahrzeug: Möglichkeiten und Herausforderungen

Michael Uelschen, Ralph Rakers

Hochschule Osnabrück

17. VDE/ITG Fachtagung Mobilkommunikation

9. - 10. Mai 2012, Osnabrück

Android im Fahrzeug

Übersicht



Hochschule Osnabrück
University of Applied Sciences

- _ Einleitung
- _ Innovationszyklendilemma
- _ Unser Ansatz
- _ Ladeapplikation für Elektrofahrzeuge*
- _ Ergebnisse
- _ Zusammenfassung
- _ Diskussion

* *Unterstützt durch Lemförder Electronic GmbH, Espelkamp (ZF)*



Android im Fahrzeug

EINLEITUNG

Android im Fahrzeug

Verteiltes System



- Ein modernes Fahrzeug der Oberklasse integriert bis zu **100** einzelne **Steuergeräte**.
- Abhängig von der Anwendung werden **verschiedene Netzwerke** mit unterschiedlichen physikalischen Schichten und Protokollen eingesetzt.
- Das **Controller Area Network** ist das zentrale Netzwerk und verbindet Antriebsstrang, Motor, Fahrerinformation etc. miteinander.

Android im Fahrzeug

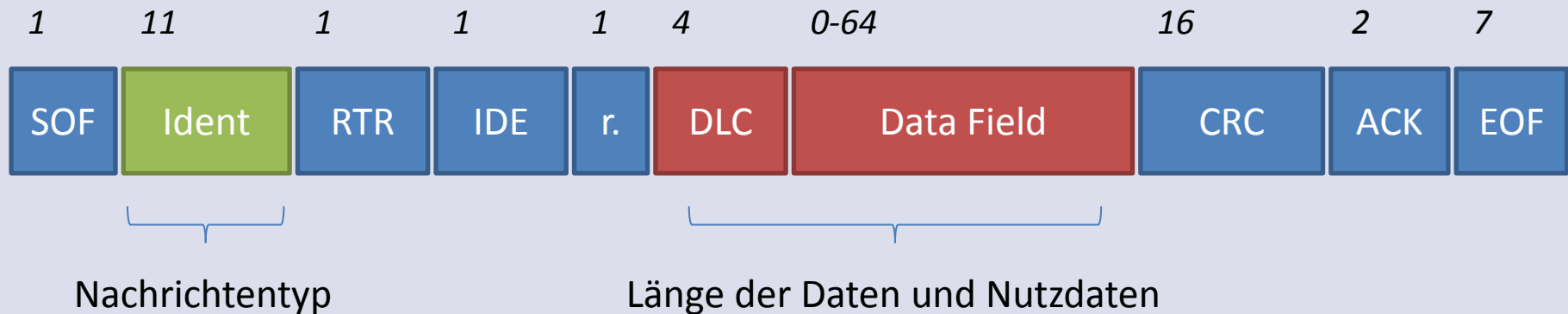
Controller Area Network (CAN)



- Entwicklung durch **BOSCH** und **Intel** ab 1983.
- Die wesentlichen Entwicklungsziele:
 - **Hohe Zuverlässigkeit**
 - **Geringe Stückkosten**
- Ereignisbasierte Kommunikation mit **kurzen Nachrichten** (130 bits) und **hoher Bandbreite** (bis zu 1 Mbit/s).
- CSMA/CD-Verfahren zur Realisierung von **Echtzeitanforderungen**.

Android im Fahrzeug

CAN Message (Data Frame)



- Nachrichten wird von **allen Knoten gelesen** (kein Adressfeld).
- Größe des Datenfeldes abhängig von den Daten.



Android im Fahrzeug

INNOVATIONSZYKLENDILEMMA

Android im Fahrzeug

Innovationszyklen -- Beispiele

1. CAN Bus: 10 Jahre

- Entwicklung durch BOSCH beginnt 1983
- Einführung in die Mercedes S-Klasse 1991
- Einführung in unteres Mittelklasse-Segment 2001



2. Apple iPod: 1 Jahr

- Eingeführt 2001
- 5te Generation vorgestellt 2005
- Innovationszyklen geringer als 1 Jahr



3. Mobiltelefone: < 1 Jahr

- Neue Geräte alle 6 Monate oder weniger



Android im Fahrzeug

Consumer Electronics Beispiel



Hochschule Osnabrück
University of Applied Sciences

TX-32LX70F LCD-FLACHBILDFERSEHER
90 cm hochauflösendes IPS-Alpha LCD Panel, 1.366 x 768 Pixel Bildauflösung, Kontrastverhältnis 8.000:1, Surround Sound, PC-Eingang, Videocast, Abmessungen in cm (ohne Standfuß) BxHxT: 79,1x56,3x11,7, Stromverbrauch Stand by laut Hersteller: 0,7 Watt, Stromverbrauch in Betrieb laut Hersteller: 134 Watt, Art. Nr.: 113 0007

TH-60 PX 600 E PLASMAFERSEHER
127 cm hochauflösendes Plasma Panel, 1.366 x 768 Pixel Bildauflösung, Kontrastverhältnis 10.000:1, Stereocast, Videocast, Abmessungen in cm (ohne Standfuß) BxHxT: 126,6x80,2x13,8, Stromverbrauch Stand by laut Hersteller: 0,3 Watt, Stromverbrauch in Betrieb laut Hersteller: 368 Watt, Art. Nr.: 137 0216

1199,- **1999,-**

video sehr gut



LE37M860D LCD-FLACHBILDFERSEHER
84 cm hochauflösendes LCD Panel, 1.920 x 1.080 Pixel Bildauflösung, dynamisches Kontrastverhältnis 8.000:1, Videocast, Stereo, digitaler Tonchips, Abmessungen in cm (ohne Standfuß) BxHxT: 85,6x55,5x8,7, Stromverbrauch Stand by laut Hersteller: 1 Watt, Stromverbrauch in Betrieb laut Hersteller: 200 Watt, Art. Nr.: 113 1295

TH-42 PZ 70 E PLASMAFERSEHER
106 cm hochauflösendes Plasma Panel, 1.920 x 1.080 Pixel Bildauflösung, Kontrastverhältnis 8.000:1, Videocast, Stereo, eingebauter DVB-T-Tuner, Abmessungen in cm (ohne Standfuß) BxHxT: 113,4x69,6, 1x13,4, Stromverbrauch Stand by laut Hersteller: 0,3 Watt, Stromverbrauch in Betrieb laut Hersteller: 484 Watt, Art. Nr.: 115 0175

1399,- **1999,-**

video sehr gut



3 Monate später

Android im Fahrzeug

Consumer vs. Automotive Domäne

Consumer Electronics Domäne

— “Schnellebige” Technologien

- Kurze Lebenszyklen
- Hohe Anforderungen an
 - Performance
 - Flexibilität
- Große Stückzahlen/Geringe Kosten
- Treiber: IT, Telekom und CE
- Neue Anwendungen
- Anforderungen an Performance stetig wachsend

— Offenes, flexibles System

Automotive Domäne

— Fahrzeugspezif. Technologien

- Lange Lebenszyklen
- Hohe Anforderungen an
 - Qualität
 - Zuverlässigkeit
- Geringe Stückzahlen
- Treiber: Automobilindustrie
- Statische Konfiguration
- Skalierbarkeit über Segmente
- Einfache Bedienbarkeit

— Zuverlässiges System



Android im Fahrzeug

UNSER ANSATZ

Android im Fahrzeug

Linux im Fahrzeug?

— Bisher Linux nicht im Fahrzeug eingesetzt:

- **Entwicklungsmodell:** Community-basiert
- **Lizenzmodell:** Open Source (z.B. GPL)
- **Patentsituation:** Rechteinhaber

— Es gibt inzwischen einige Änderungen:

- **GENIVI Initiative** zur Entwicklung Linux im Auto
- **MeeGo/Tizen** Plattform für Fahrerinformation

— Unser Ansatz: **Android**-basiertes Linux-System

Android im Fahrzeug

Unsere Idee



- Android ist nicht nur eine Technologie, sondern bietet ein komplettes **Eco-System** (z.B. Google Play).
- **Integration** einer Android-basierten Head-Unit in ein **automotives Netzwerk (CAN)**.
- Nachweis der **Interaktion** mit automotiven **Entwicklungswerkzeugen** (hier: CANoe).
- App-Entwicklung: Steuerung und Visualisierung des **Ladevorgangs** für **Elektrofahrzeuge**.

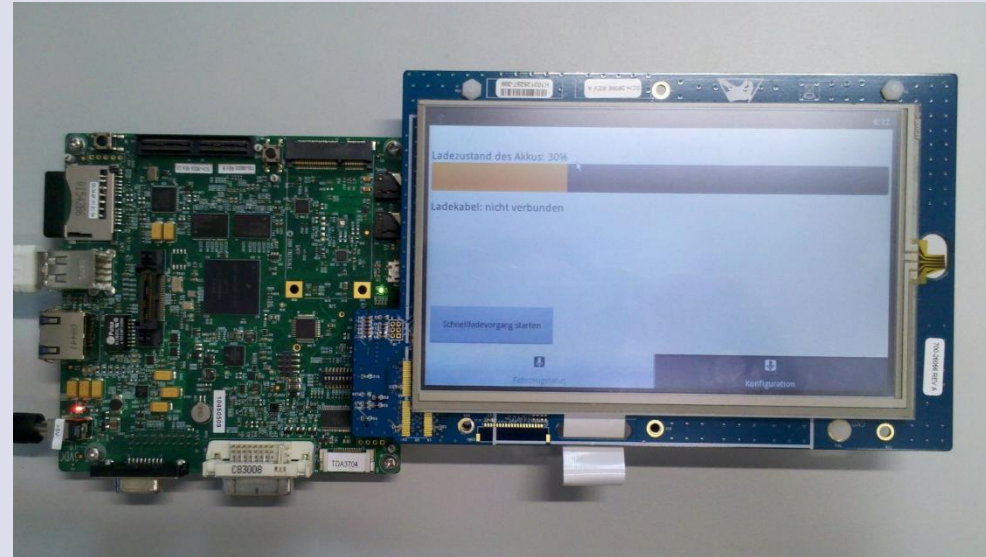
Android im Fahrzeug

Hardware Plattform



Hochschule Osnabrück
University of Applied Sciences

- **Freescale i.MX51 EVK**
- Cortex A8@800 MHz
- 512 MB DDR2
- 7" WVGA Touchscreen
- Standard-Schnittstellen:
USB, Ethernet, SD, ...
- Keine automotiven
Schnittstellen auf dem
Eval-Board integriert.
- Android Patches durch
Freescale.



Android im Fahrzeug

CAN Interface

- **PEAK System PCAN**
- CAN-to-USB Adapter
- USB 1.1 (2.0)
- CAN Spezifikation
2.0A/B
- Linux-Treiber (GPL)
verfügbar für Kernel
2.4-3.0



— CAN Implementierung für Linux-Kernel

- Projekt der Volkswagen Forschung
- Offizieller Bestandteil seit Kernel 2.6.25
- Konfiguration über make menuconfig

— BSD-Sockets API

- `socket()`, `bind()`, `read()`, `write()`, `close()`

— CAN-Frame definiert in `<linux/can.h>`

```
struct can_frame ping;  
ping.can_id = 0x123;  
ping.can_dlc = 1;  
ping.data[0] = 0xFF;
```




— Shared library CANComLib (~ 200 LOC) kapselt die **Socket-Aufrufe** wie

- `socket()`, `bind()`, `read()`, `write()`, `close()`

— Bereitstellung einer einfachen Schnittstelle zum **Senden** und **Empfangen** von CAN-Nachrichten:

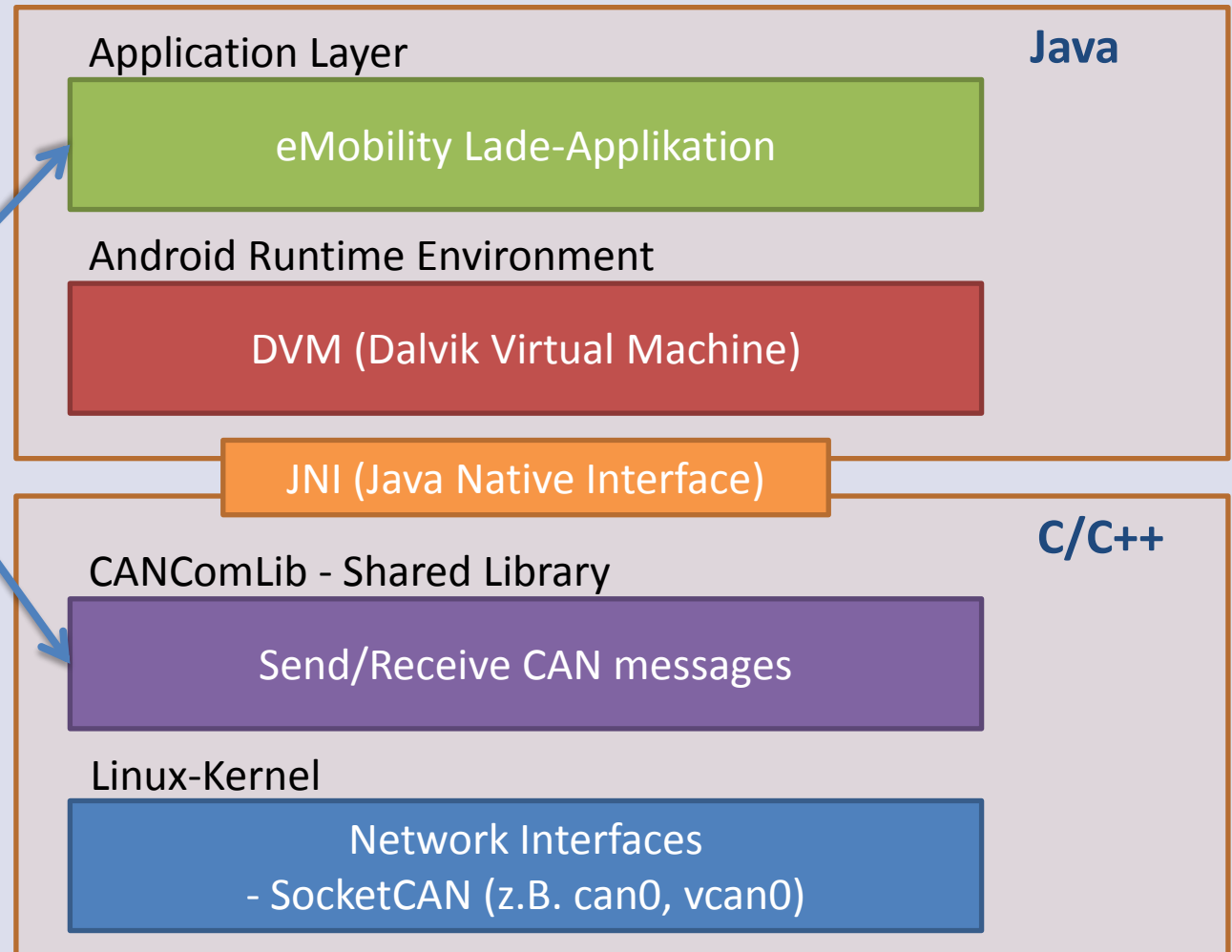
```
int initialize();  
struct can_frame receive_can(struct can_filter filter);  
void send_can(struct can_frame sendFrame);  
int terminate();
```

— Android Applikationen verwenden die native Bibliothek durch das **Android NDK**.

Android im Fahrzeug

Software Komponenten

Bestandteile
im Projekt





Android im Fahrzeug

LADEAPPLIKATION FÜR ELEKTROFAHRZEUGE

Android im Fahrzeug

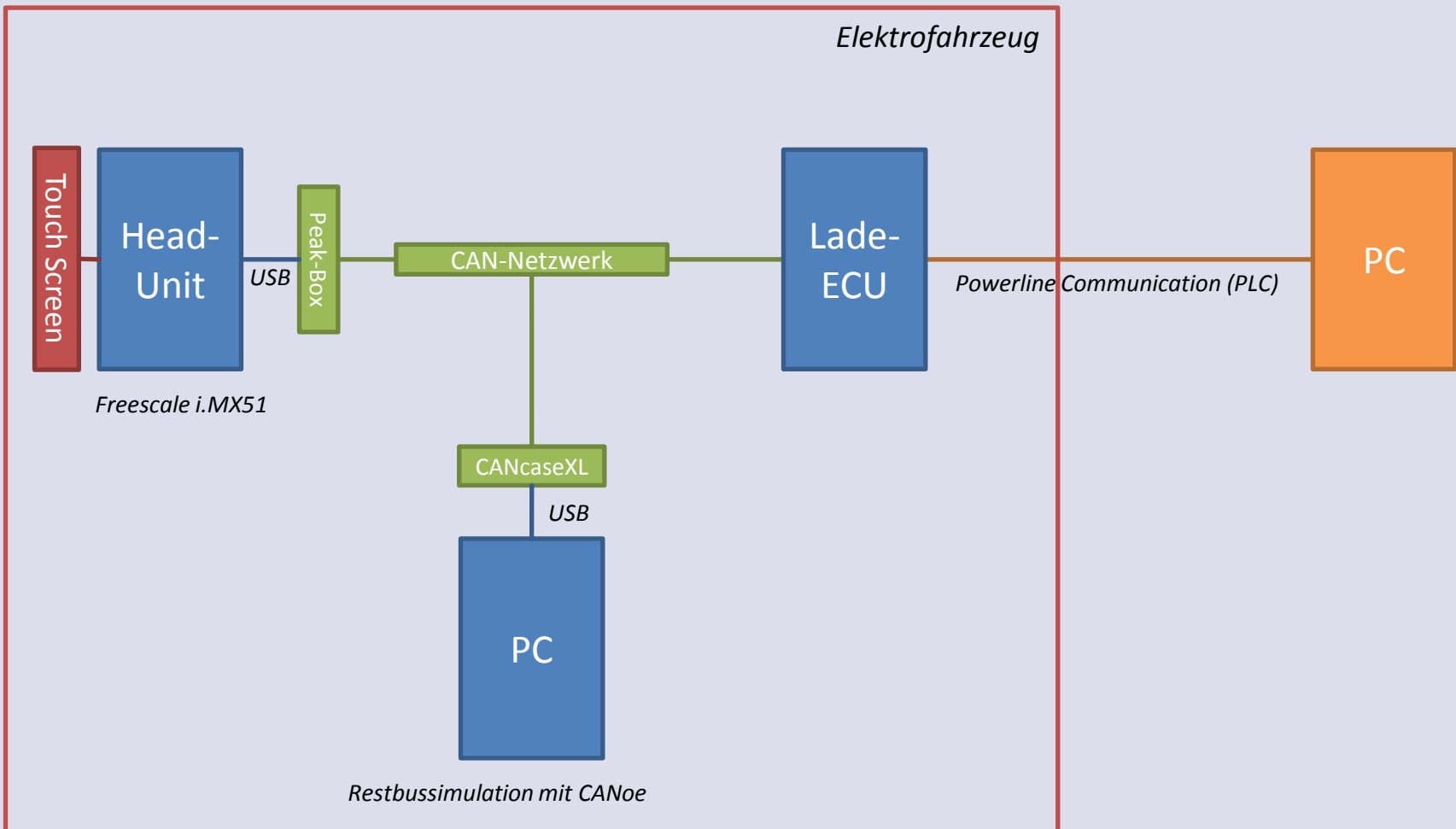
Vehicle-to-Grid Kommunikation



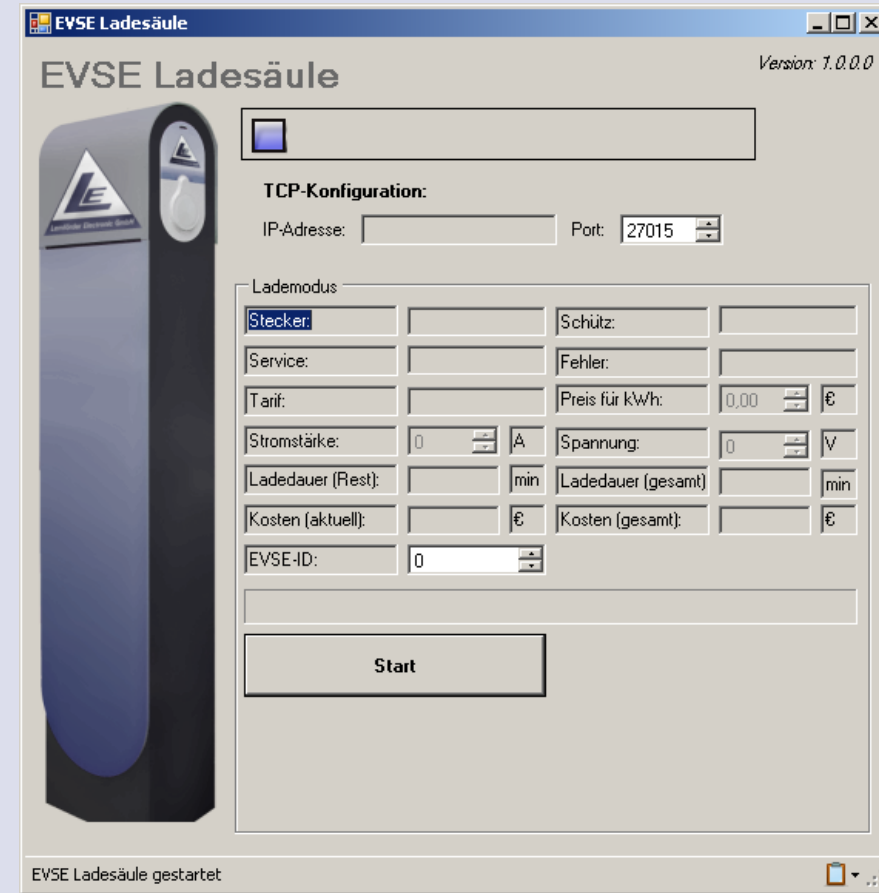
- Die **Kommunikation** zwischen **Elektrofahrzeug** und **Ladesäule** ist definiert durch ISO61851 und ISO15118.
- Unterschiedliche **Dienste** sind definiert
 - Batterie Laden
 - Internet Zugang
 - Benutzerdefinierte Dienste
- App zur Initialisierung, Steuerung und Visualisierung des **Ladevorgangs** basierend ISO Standards.

Android im Fahrzeug

System Übersicht (Entwicklung)



Android im Fahrzeug System Übersicht (Komponenten)



Android im Fahrzeug System Übersicht (Bus Simulation)



Hochschule Osnabrück
University of Applied Sciences

Vector CANoe - Fahrzeug-BA.cfg

Datei Ansicht Start Modus Konfiguration Fenster Hilfe

syn hex Realer Bus

Trace

Time	Chn	ID	Name	Dir	DLC	Data
00:00:00	218.201759	1	7FF ISO61851_MSG	Tx	3	03 20 05
			VAL_PILOT		5	Hilfssignal für das Tastenverhältnis des Pilot-Signals (ISO61851)
			VAL_PROXIMITY		32	Ampere
			ST_PILOT		1	Hilfssignal für Stromtragfähigkeit des Ladekabels, ausgelesen aus dem Ladestecker (ISO61851)
			ST_PROXIMITY		1	Hilfssignal für den Status der Pilotleitung
			ST_PROXIMITY		1	Hilfssignal für den Status der Proximityleitung
218.201573	1	3E9	CHGNG_ST	Tx	2	3C 00
			ST_ACCU		60	percent
			ST_CHGRDI		0	Statusinformation über die Ladebereitschaft des Fahrzeuges.
			ST_CHGNG		0	Statusinformation über den Ladezyklus.
218.201416	1	2FC	STAT_ZV_KLAPPEN	Tx	1	01
			ST_CLSY		1	Dieser Parameter/Signal beschreibt den 'globalen Zustand' der Zentralverriegelung...
187.156954	1	7F8	V2G_MeteringReceipt	Rx	4	72 00 00 00
			V2G_EVSE_EnergyPriceCurrent		114	cent
			V2G_ChargeService...		4	00 00 05 00
218.156758	1	7F6	V2G_ChargeService...	Rx	4	00 00 05 00
			V2G_CPD_RC		0	Charge Parameter Discovery Response Code
			V2G_EVSE_ShutDownTime		5	s
			V2G_EVSE_RCD		0	0
			V2G_EVSE_PowerSwitch		0	0
			V2G_EVSE_ConnectorStatus		0	0
			V2G_EVSE_Status		0	0
			V2G_EVSE_Error		0	0
			V2G_PEV_ConnectorStatus		0	0
			V2G_PEV_ChargerStatus		0	0
			V2G_PWD_RC		0	0
			V2G_MR_RC		0	0
			V2G_MS_RC		0	0
			V2G_LL_RC		0	0
185.157036	1	7F7	V2G_MeteringStatus	Rx	8	F0 55 60 DE 47 16 00 00
			V2G_EVSE_maxPower		22000	W
			V2G_EVSE_MeterValue		5703	Wh
			V2G_EVSE_PCurrent		3680	W
80.131954	1	7F3	V2G_UserChargeServ...	Rx	5	00 00 00 00 00
			V2G_usedTariff		0	0
			V2G_EndOfCharge		0	min
			V2G_EnergyAmount		0	kWh
40.156490	1	7F5	V2G_ChargeService...	Rx	1	03
			V2G_Tariffs		3	Gibt mögliche Tarife der EVSE an. Bei Verfügbarkeit ist das jeweilige Bit gesetzt, d.h. z.
16.156203	1	7F2	V2G_ServiceStatus	Rx	2	00 00
			V2G_SD_RC		0	Service Discovery Response Code
			V2G_SS_RC		0	Session Setup Response Code
			V2G_PD_RC		0	Payment Details Response Code
			V2G_SPS_RC		0	Service and Payment Selection Response Code
14.253105	1	7F4	V2G_PaymentDetails	Rx	4	39 30 00 00
			V2G_ContractID		12345	3039
			V2G_ContractID		12345	Eingegebene Vertragsnummer
12.995274	1	7F1	V2G_UserServiceSetup	Rx	2	00 00
			V2G_usedPayment		0	0
			V2G_usedService		0	0
6.156456	1	7F0	V2G_ServiceSetup	Rx	6	01 01 00 00 00 00
			V2G_EVSE_ID		0	0
			V2G_EVSE_ID		0	Eindeutige ID der Ladesäule

Messaufbau

Real Offline Online

Fahrzeug

Fahrzeugsteuerung
Automatikwahlhebelposition
P R N D

Fahrzeugstatus
Fahrzeugverriegelung
Fahrzeug offen
Fahrzeug geschlossen

Ladestecker
einstecken
ausstecken

Fahrzeugverriegelung
entriegeln
verriegeln

Start-Ladestatus des Akkus
30%

Ladestatus des Akkus
60%

Ladezyklus
Kein Laden
Laden aktiv
Ladeende

Verriegelung Ladestecker
verriegelt
Fehler
ungültig

Netzwerke
CAN Netzwerke
CAN
Knoten
demo-ecu
hmi-ecu
lm-ecu
main-ecu
Generatoren
Interaktive Generatoren
Replay-Blöcke
Datenbasen
LUFU_LIM_CANDB
Umgebungsvariablen
Kanäle

Trace Configuration Analysis

Bereit

0:00:03:38 LUFU_LIM_CANDB;Umgebungsvar... ONLINE REAL NUM HEX

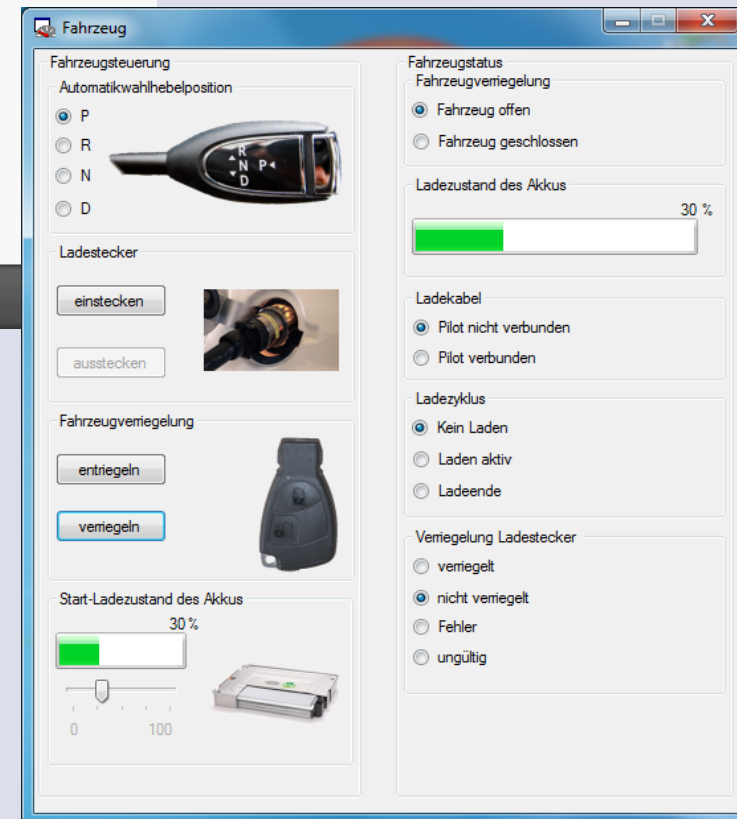
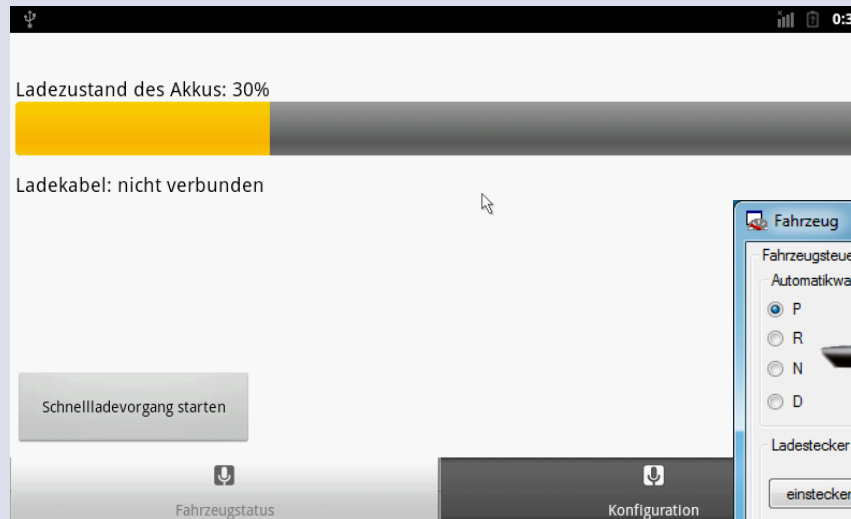
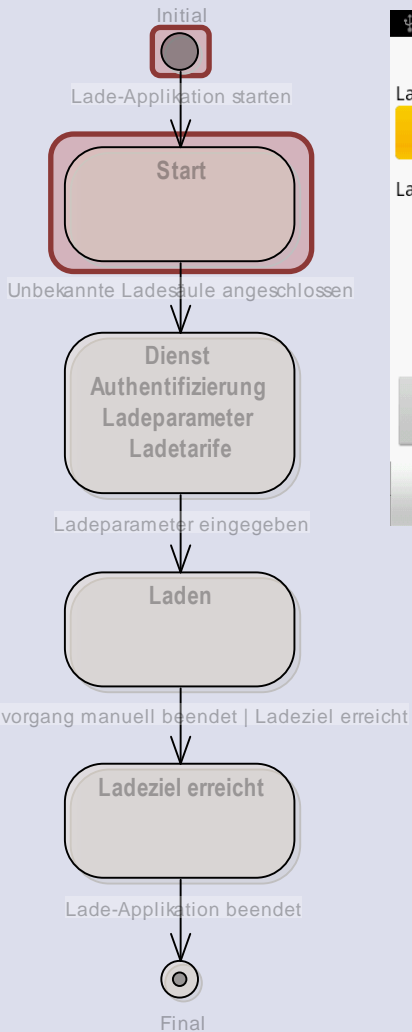


Android im Fahrzeug

ERGEBNISSE

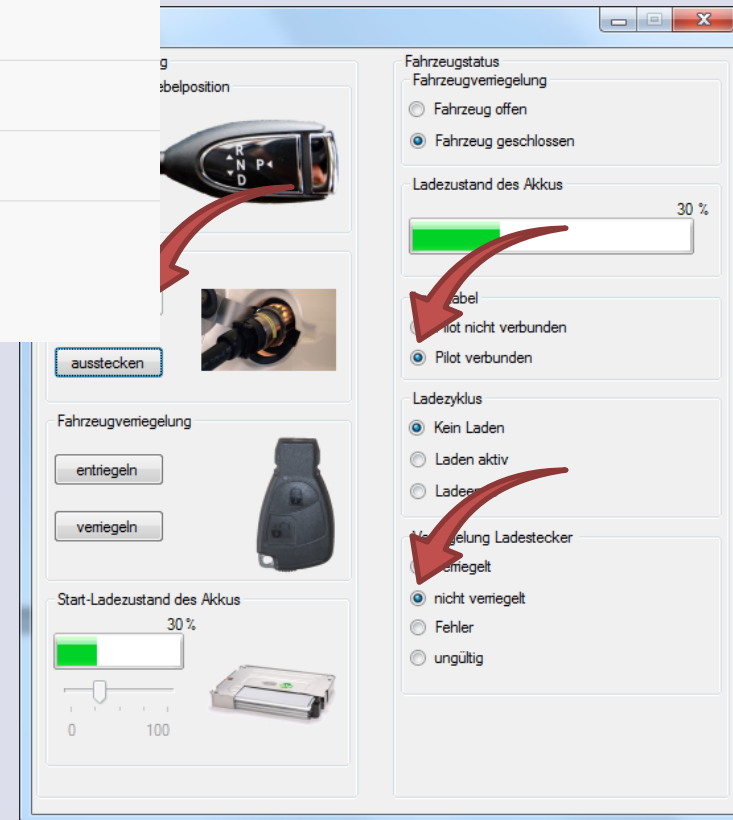
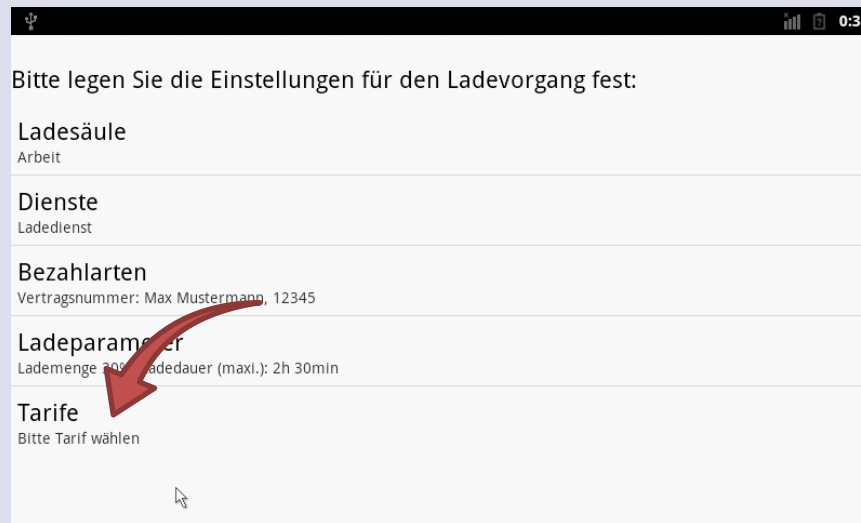
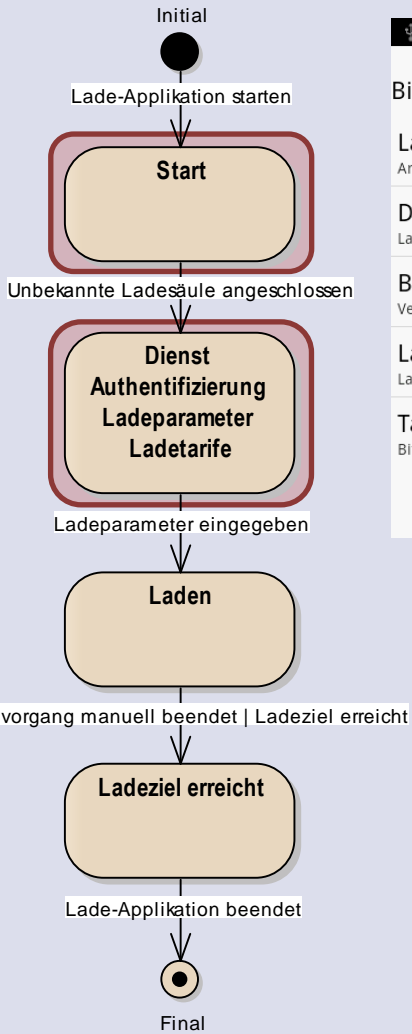
Android im Fahrzeug

Zustandsdiagramm UI -- Willkommen



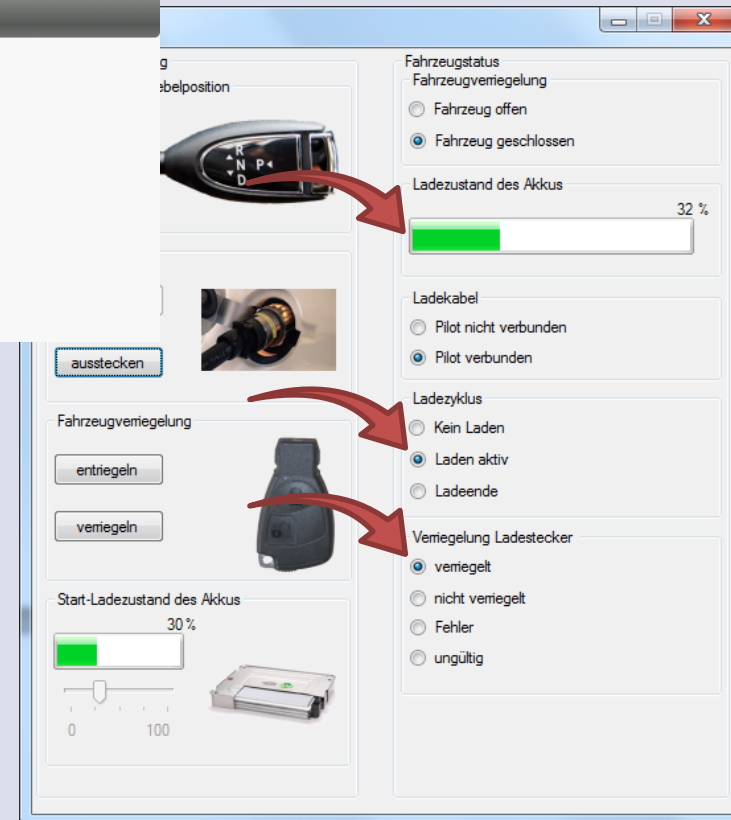
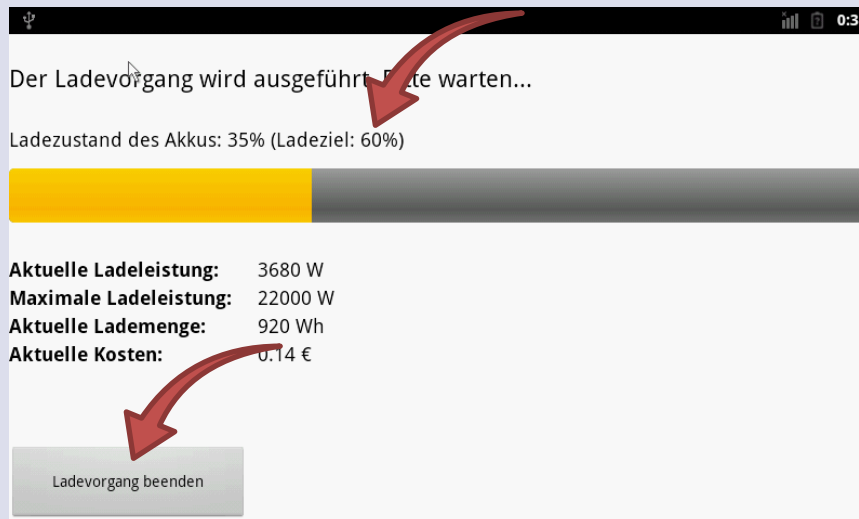
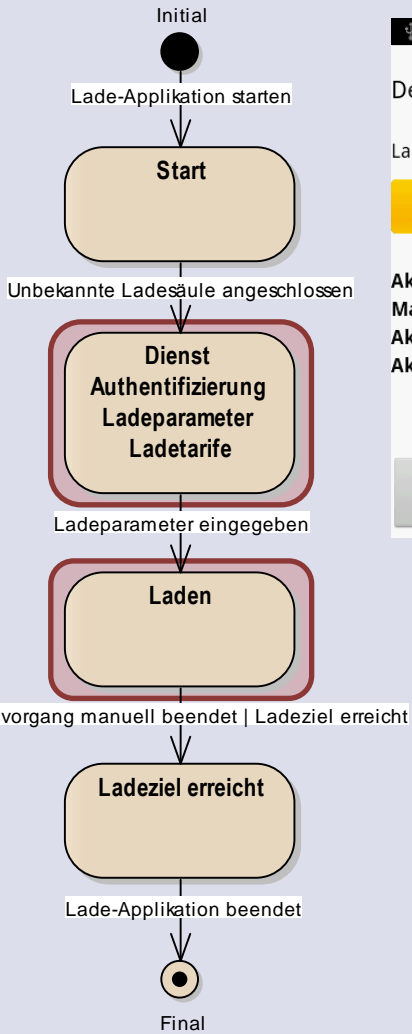
Android im Fahrzeug

Zustandsdiagramm UI -- Parameter



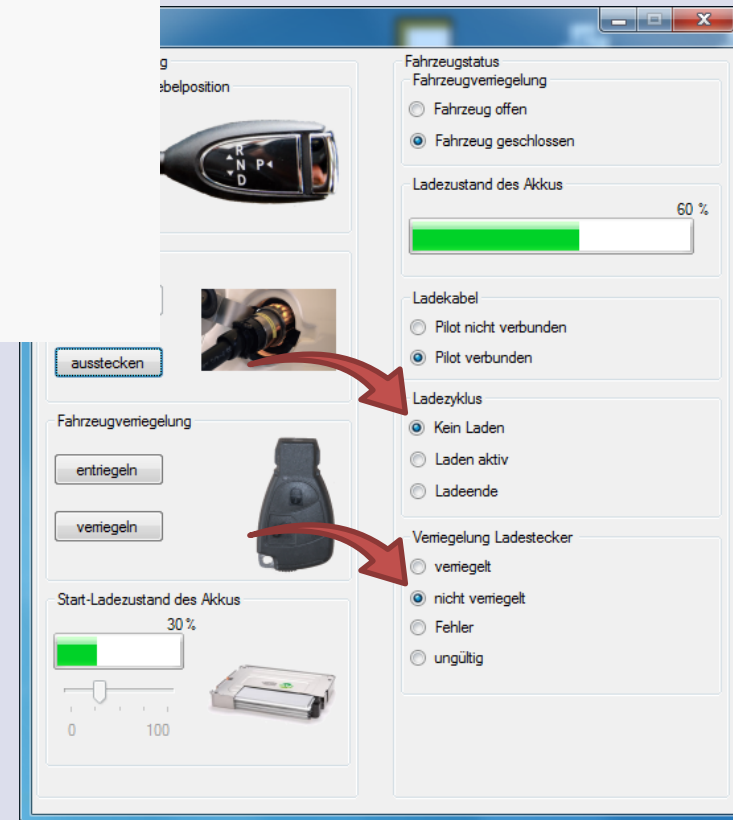
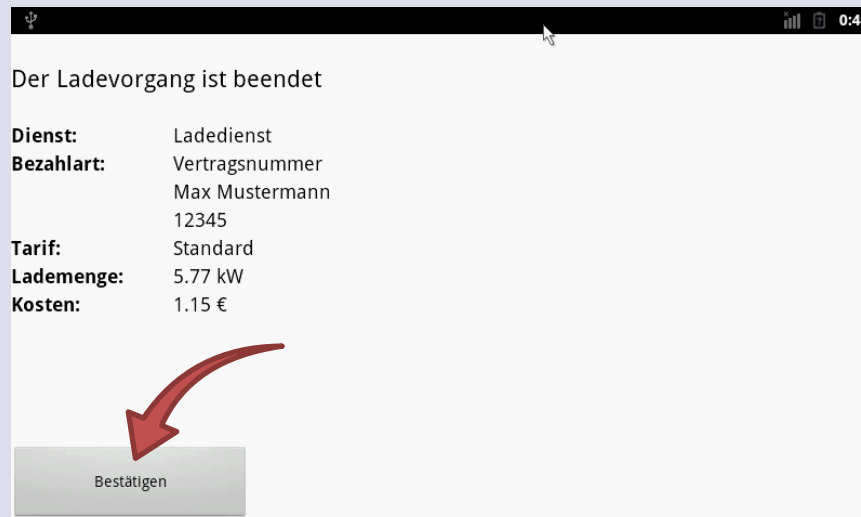
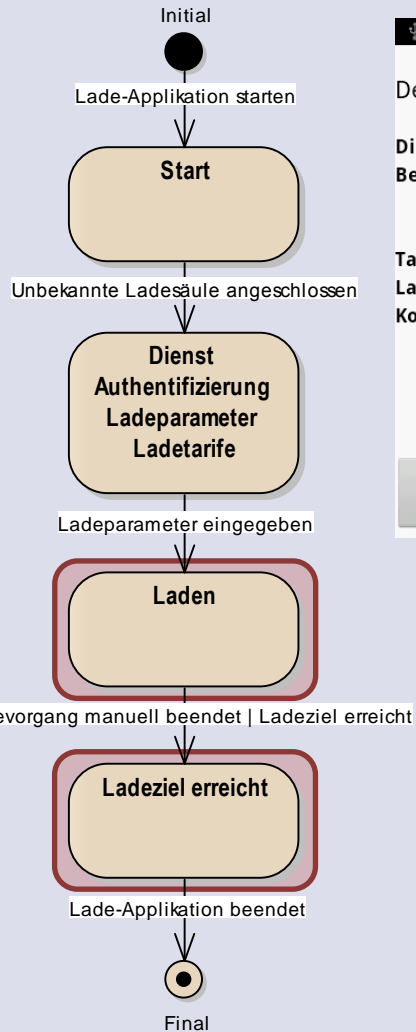
Android im Fahrzeug

Zustandsdiagramm UI -- Laden



Android im Fahrzeug

Zustandsdiagramm UI – Laden beendet





Android im Fahrzeug

ZUSAMMENFASSUNG

Android im Fahrzeug

Zusammenfassung



- Android-System integriert in die **Kommunikation** in einem **CAN-Netzwerk** im Fahrzeug.
- Ansatz lässt sich **übertragen** in andere Bereiche, in denen CAN benutzt wird (z.B. Automatisierung).
- Produktentwicklung erfordert eine **höhere Integration** der Komponenten (insb. CAN Controller).

Android im Fahrzeug

Ausblick



- Einheitliche **Fahrzeug-API** erforderlich, um Android Apps **herstellerübergreifend** entwickeln zu können.
- Anpassung der **Bedienschnittstelle** an automotive Gegebenheiten, insb. **Spracheingabe** und **Drehdrück-Steller**.
- **Entkoppelung** der unterschiedlichen Anforderungen und Innovationszyklen durch **Multicore** und/oder **Virtualisierung**.



- **Anzeige und Steuerung** eines [Android-] Smartphones lässt sich in **Fahrerinformationssystem** einbinden (Mirrorlink, RealVNC, Touch Life von Toyota).
- **Renault** hat angekündigt, integriertes Android-System anzubieten (**R-Link**) mit einem geschützten App-Store (50 Apps).

Android im Fahrzeug

Diskussion



Hochschule Osnabrück
University of Applied Sciences

— Vielen Dank für Ihre Aufmerksamkeit!

Android im Fahrzeug

Kontaktinformation

Prof. Dr.-Ing. Michael Uelschen

Hochschule Osnabrück

Fakultät Ingenieurwissenschaften und Informatik

Laborbereich Technische Informatik

Barbarastraße 16, 49076 Osnabrück

Postalische Adresse: Postfach 1940, 49009 Osnabrück

m.uelschen@hs-osnabrueck.de

Tel.: +49 (0)541 / 969 3885

Raum: SI0212