

Incar

M. Dederer S. Eisenhauer

Fachbereich Informatik
Hochschule Darmstadt

Masterprojekt Systementwicklung SS 2010

Gliederung

- 1 Intel Atom Target
 - Linux Installation
 - Touchscreen
 - Framework
- 2 GPS
 - GPS Komponente
 - SH4 Target
 - NMEA
- 3 OBD
 - OBD Hardware
 - OBD Komponente
 - Demo
- 4 Nächste Schritte

Intel Atom Target

Linux

- Ubuntu 64 bit
- Hardwareunterstützung
- X-Server

Intel Atom Target

Linux

- Ubuntu 64 bit
- Hardwareunterstützung
- X-Server

Intel Atom Target

Linux

- Ubuntu 64 bit
- Hardwareunterstützung
- X-Server

Intel Atom Target

Touchscreen

- Zusätzliche Treiber
- Als Ubuntu-Pakete verfügbar
- Kalibrierung hakelig

Intel Atom Target

Touchscreen

- Zusätzliche Treiber
- Als Ubuntu-Pakete verfügbar
- Kalibrierung hakelig

Intel Atom Target

Touchscreen

- Zusätzliche Treiber
- Als Ubuntu-Pakete verfügbar
- Kalibrierung hakelig

Intel Atom Target

Framework

- Übersetzen auf Laborrechner aus Momentics
- Target LINUX_X86
- Ausführbare Datei auf das Target kopieren
- Ausführbare Datei auf dem Target starten

Intel Atom Target

Framework

- Übersetzen auf Laborrechner aus Momentics
- Target LINUX_X86
- Ausführbare Datei auf das Target kopieren
- Ausführbare Datei auf dem Target starten

Intel Atom Target

Framework

- Übersetzen auf Laborrechner aus Momentics
- Target LINUX_X86
- Ausführbare Datei auf das Target kopieren
- Ausführbare Datei auf dem Target starten

Intel Atom Target

Framework

- Übersetzen auf Laborrechner aus Momentics
- Target LINUX_X86
- Ausführbare Datei auf das Target kopieren
- Ausführbare Datei auf dem Target starten

GPS Komponente

Beschreibung

- Framework-Komponente für alle Targets
- Auswahl zur Compile-Zeit
- Einheitliche Struktur zur Applikation
- Daten im Shared Memory, Nachrichten an andere Komponenten

GPS Komponente

Beschreibung

- Framework-Komponente für alle Targets
- Auswahl zur Compile-Zeit
- Einheitliche Struktur zur Applikation
- Daten im Shared Memory, Nachrichten an andere Komponenten

GPS Komponente

Beschreibung

- Framework-Komponente für alle Targets
- Auswahl zur Compile-Zeit
- Einheitliche Struktur zur Applikation
- Daten im Shared Memory, Nachrichten an andere Komponenten

GPS Komponente

Beschreibung

- Framework-Komponente für alle Targets
- Auswahl zur Compile-Zeit
- Einheitliche Struktur zur Applikation
- Daten im Shared Memory, Nachrichten an andere Komponenten

SH4 Target

GPS Komponente

- Liefert GPS Daten über Filehandle
- GPS Daten binär
- Feste Struktur vorgegeben \Rightarrow kein Datenverlust
- Direktes Einlesen in struct
- Parsen nicht notwendig

SH4 Target

GPS Komponente

- Liefert GPS Daten über Filehandle
- GPS Daten binär
- Feste Struktur vorgegeben \Rightarrow kein Datenverlust
- Direktes Einlesen in struct
- Parsen nicht notwendig

SH4 Target

GPS Komponente

- Liefert GPS Daten über Filehandle
- GPS Daten binär
- Feste Struktur vorgegeben \Rightarrow kein Datenverlust
- Direktes Einlesen in struct
- Parsen nicht notwendig

SH4 Target

GPS Komponente

- Liefert GPS Daten über Filehandle
- GPS Daten binär
- Feste Struktur vorgegeben \Rightarrow kein Datenverlust
- Direktes Einlesen in struct
- Parsen nicht notwendig

SH4 Target

GPS Komponente

- Liefert GPS Daten über Filehandle
- GPS Daten binär
- Feste Struktur vorgegeben \Rightarrow kein Datenverlust
- Direktes Einlesen in struct
- Parsen nicht notwendig

NMEA

NMEA Empfänger

- Targets
 - LINUX_X86
 - QNX_X86
 - Einziger Unterschied: Pfad (/dev/...)
- GPS Daten seriell über USB im NMEA Format
- ASCII Datenstrom
 - Div. Sätze: \$GP... \n (bis zu 82 Zeichen pro Satz)
 - Datenverlust möglich
 - Doppelte Pufferung

NMEA

NMEA Empfänger

- Targets
 - LINUX_X86
 - QNX_X86
 - Einziger Unterschied: Pfad (/dev/...)
- GPS Daten seriell über USB im NMEA Format
- ASCII Datenstrom
 - Div. Sätze: \$GP... \n (bis zu 82 Zeichen pro Satz)
 - Datenverlust möglich
 - Doppelte Pufferung

NMEA

NMEA Empfänger

- Targets
 - LINUX_X86
 - QNX_X86
 - Einziger Unterschied: Pfad (/dev/...)
- GPS Daten seriell über USB im NMEA Format
- ASCII Datenstrom
 - Div. Sätze: \$GP... \n (bis zu 82 Zeichen pro Satz)
 - Datenverlust möglich
 - Doppelte Pufferung

NMEA

NMEA Empfänger

- Targets
 - LINUX_X86
 - QNX_X86
 - Einziger Unterschied: Pfad (/dev/...)
- GPS Daten seriell über USB im NMEA Format
- ASCII Datenstrom
 - Div. Sätze: \$GP... \n (bis zu 82 Zeichen pro Satz)
 - Datenverlust möglich
 - Doppelte Pufferung

Implementierung

Aktueller Stand

- POSIX konform
 - nicht blockierendes Lesen
 - select
- Satzweise Parsen
 - GPRMC implementiert
 - Position, Zeit
 - GPGGA, GPGSA und GPGSV als Stubs vorhanden

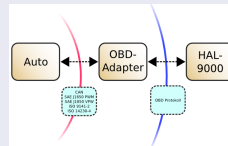
GPRMC Satz

```
$GPRMC,144455.000,A,5012.8939,N,00847.6632,E, \
0.00,204.13,260610,,A*6B
```

OBD Hardware

- Pseudostandard
- Adapter mit ELM Interpreterchip
- AGV4000
 - Ford Fiesta MK6
 - Skoda Roomster
- Open-Source Simulator
 - nicht auf Laborrechnern kompilierbar

Schema



AGV 4000



OBD Komponente

- POSIX konform (non-blocking read, select)
- Daten im Shared Memory, Nachrichten an andere Komponenten
- Aktuell abgefragte Werte
 - Drehzahl
 - Geschwindigkeit
 - Kühlwasser Temperatur
 - Luftmasse
 - Drosselklappenöffnung
 - Motorlast

OBD Komponente

- POSIX konform (non-blocking read, select)
- Daten im Shared Memory, Nachrichten an andere Komponenten
- Aktuell abgefragte Werte
 - Drehzahl
 - Geschwindigkeit
 - Kühlwasser Temperatur
 - Luftmasse
 - Drosselklappenöffnung
 - Motorlast

OBD Komponente

- POSIX konform (non-blocking read, select)
- Daten im Shared Memory, Nachrichten an andere Komponenten
- Aktuell abgefragte Werte
 - Drehzahl
 - Geschwindigkeit
 - Kühlwasser Temperatur
 - Luftmasse
 - Drosselklappenöffnung
 - Motorlast

OB Demo Anwendung

- Video Simulator
- Video Fahrzeug

OBD Demo Anwendung

- Video Simulator
- Video Fahrzeug

Mögliche Fortführungen

GPS

- Kontinuierliche Erweiterung und Verbesserung
- Absprache mit Hr. Ecker

Intel Atom Target

- Touchscreen-Steuerung für Framework
- Remote Debugging

Mögliche Fortführungen

GPS

- Kontinuierliche Erweiterung und Verbesserung
- Absprache mit Hr. Ecker

Intel Atom Target

- Touchscreen-Steuerung für Framework
- Remote Debugging

Mögliche Fortführungen

OBD

- Verbesserung Abtastrate
- Digitales Armaturenbrett
- Kopplung mit anderen Systemen
 - Warnung bei Geschwindigkeitsübertretung
 - Verbrauchsoptimierung

Zusammenfassung

- Framework auf Intel Atom Target ausführbar
- Stabile GPS Komponente, gegen Datenverlust gesichert
- GPS Daten im Framework verfügbar
- OBD Daten im Framework verfügbar
- Einfacher Drehzahlmesser über OBD