

Tabelle1

Projekt Systementwicklung SS2010, Prof. Wietzke

24.03.10

Faktensammlung der Projektgruppe „Portierung ins Auto, Linux und Fahrzeuganbindung“
Martin Dederer, Sven Eisenhauer

1 Target

Stand Atom basierter mini-ITX PC mit nVidia Ion Chipsatz, Ethernet, W-LAN, Line-In, Line-Out, VGA out, USB, 1x seriell, HDD, optisches Laufwerk
+ Faytech 7" Touchscreen (Kalibrierung auf X-Achse z. Zt. fehlerhaft, vermutlich behebbar, Kfz-Spannungsversorgung vorhanden, aber sehr dünne Litzen)
+ USB GPS Maus (ublox 5 basiert)
+ MiniCommander an seriell (Stand Integration in Framework??)
- kein Kfz-Netzteil
- kein MOST

Optionen

a) Beschaffung passendes Kfz-Netzteil für mini ITX Mainboards
+ z. B. M2-ATX (puffert Spannung stabil auf 12V auch bei Motorstart, Entladeschutz der Batterie, schaltet Target über Zündung ein und aus, Einschaltverzögerung für ext. Amplifier)

2 Linux

Stand Eigenes Buildsystem
Kernel
X-Server (nicht lauffähig für target?!)
busybox
qemu und target image
makefile basiert
Miniframework integriert? (Stand, Umfang??)

Optionen

a) Weiter mit eigenem Buildsystem

+ reduzierbar auf absolutes Minimum
- Integration aufwendig
- kann Aufwände generieren, die bei anderen Optionen schon gelöst sind (Warum das Rad neu erfinden?)

b) buildroot

+ Integration weiterer Komponenten über makefiles möglich
- unflexibel
- makefile basiert

c) openembedded

+ flexibel
+ Integration weiterer Komponenten über sog. Recipes
+ verwendet für z. B. Angstrom (embedded linux distribution)
- keine Erfahrung damit

d) T2 project

+ Metadistribution
+ verwendet z. B. für Puppy Linux (embedded linux distribution)

Tabelle1

- + hoch flexibel
- evtl. zu komplex
- keine Erfahrung damit

e) MeeGoo

- + vielversprechender Ansatz (soll Moblin und maemo vereinen)
- + Intel, Nokia
- nicht existent

f) Moblin

- + mobile software plattform
- + IVI branch
- + Plattform für GENIVI (BMW, ARM, Freescale, Delphi, Visteon, Intel...)
- + SDK und API verfügbar
- Ion Unterstützung? (wohl möglich, X-Server recompile mit xinerame enabled und proprietärer nVidia Treiber)
- keine Erfahrung damit

g) Poky

- + Plattform für mobile Geräte (Design, Develop, Build, Test, Debug, and Deploy)
- + Anjuta C++ IDE integriert
- + Ableger von openembedded
- keine Erfahrung damit

h) GENIVI?

- + Soll IVI Referenz Plattform werden
- + viele Hersteller vertreten
- will eigene Frameworks bereitstellen
- Verfügbarkeit?

i) Ubuntu Mobile

- eingestellt?
- (<https://wiki.ubuntu.com/MobileTeam/Mobile?action=show&redirect=MobileAndEmbedded>)

j) LinuxICE / OpenICE

- + Zielsetzung: Freie und offene Linux-Distribution für Incar-Entertainment
- + Basiert auf Ubuntu
- + funktioniert
- kleines Projekt, langfristig verfügbar?
- eigenes Framework, API
- kaum Dokumentation
- basiert auf altem Ubuntu-Release (9.04)

k) Standard Distribution (z. B. Ubuntu)

- + Hardwareunterstützung (bei Ubuntu nvidia ion Grafiktreiber über GUI installierbar)
- + umfangreiche Software
- Desktop-System (nicht für embedded gedacht)
- nicht für geringe Auflösung des 7" Displays gedacht

3 Tuner

Stand nicht vorhanden

Tabelle1

Optionen

a) USB FM Tuner Stick (SiLabs o. ä.)

- + preiswert
- + fertig
- nicht im Auto verwendbar (schlechter Empfang, äußere Einflüsse)

b) USB Radio im Eigenbau

- + basiert auf Incar-Komponenten
- + μ C (Atmel 8-bit ATmega168 o. ä.) steuert Tuner über I²C, RDS über RS232
- + Target-Kommunikation über USB mittels im Radio integriertem FTDI USB
- + Einfaches Protokoll möglich
- + RDS-Dekodierung im μ C
- + Audio-Signal auf Line-In des Targets oder direkt an Amplifier
- community Projekt, früher Prototyp
- Platine ätzen, SMD löten u. ä. nötig

c) Techdesign FM Tuner über IOWarrior

- + Target steuert Tuner über I²C (IOWarrior), alternativ zwischengeschalteter μ C
- + Tuner Modul fertig zu kaufen
- + Incar Tuner Komponenten
- + weitere GPIO über IOWarrior
- + Audio-Signal auf Line-In des Targets oder direkt an Amplifier
- RDS Dekoder muss implementiert werden

d) Harmann/Becker

- vielleicht kann H/B bei der Beschaffung einer tauglichen Tuner-Lösung helfen

4 Amplifier

Stand nicht vorhanden

Optionen

a) Ext. Amplifier über Line-Out des Targets

- + einfacher Anschluss
- + Logisches Device im Framework steuert Mixersoftware der Linux-Distribution
- + Lautstärkeregelung über Soundkartentreiber oder Mixer-Software des Targets
- nur ein Stereoausgang, kein Fader (Rear/Front) möglich

b) Ext. Amplifier über Audio-Outs des Tuners (3b oder 3c)

- + 2 Stereoausgänge
- + CD über Line-Out des Targets an Audio-In des Tuners
- Logisches Device steuert Lautstärke über Befehle an Tuner (Protokoll)
- Logisches Device muss Audio-Ins des Tuners schalten (Protokoll)

5 CD

Stand Optisches Laufwerk im Target

Optionen

a) Integration in Framework

- + Logisches Device im framework steuert vorhandene Systemsoftware für CD-Playback

