The Free Ride

Your guide to automotive Linux

Nathan Willis nate@lwn.net

GENIVI®

AUTOMOTIVE GRADE LINUX

TIZEN™

DIY EFI

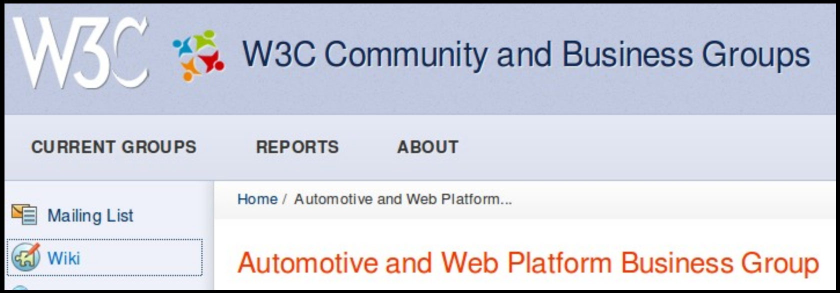
MegaSquirt EFI
www.megasquirt.info

OpenXC

MirrorLink™

AUTOSAR

The Yocto Project IVI Layer



W3C Community and Business Groups

Automotive and Web Platform Business Group

Open Automotive Alliance

Automotive software: two domains

ECM

- “Electronic control modules”
- Under the hood

IVI

- “In-vehicle Infotainment”
- In the cabin

Automotive software: two domains

ECM

- "Electronic control modules"
- Under the hood

IVI

- "In-vehicle Infotainment"
- In the cabin
- "we all die just a little inside whenever we read the word *infotainment*" *(Brian Proffitt)*

Automotive software: two domains

ECM

- “Electronic control modules”
- Under the hood
- Engine timing, braking, safety

IVI

- “In-vehicle Infotainment”
- In the cabin
- Navigation, music, phone

Automotive software: two domains

ECM

- "Electronic control modules"
- Under the hood
- Engine timing, braking, safety

IVI

- "In-vehicle Infotainment"
- In the cabin
- Navigation, music, phone

...and a gray area....

IVI

Typically in-house carmaker projects

GM "CUE"

Renault R-Link

BMW ConnectedDrive

Ford MyTouch

Audi MMI

IVI

Typically in-house carmaker projects

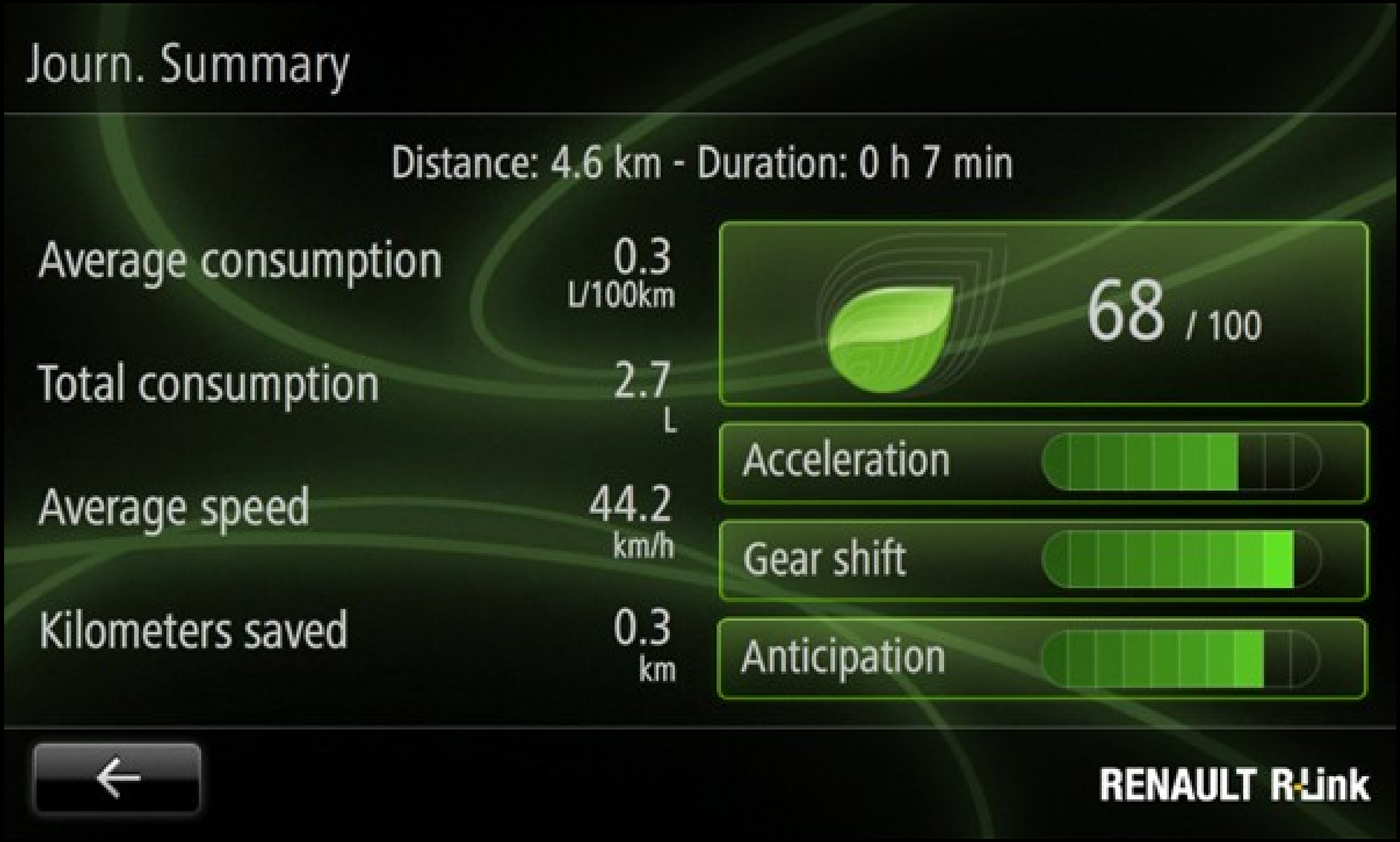
Moving toward consolidation

- Reduce costs
- Long product lifetimes
- Structured supply-chains
- App store goldmine

ECM

Not a customer-facing component, but still cost-sensitive issue.

IVI already exists....



Journ. Summary

Distance: 4.6 km - Duration: 0 h 7 min

Average consumption	0.3 L/100km
Total consumption	2.7 L
Average speed	44.2 km/h
Kilometers saved	0.3 km

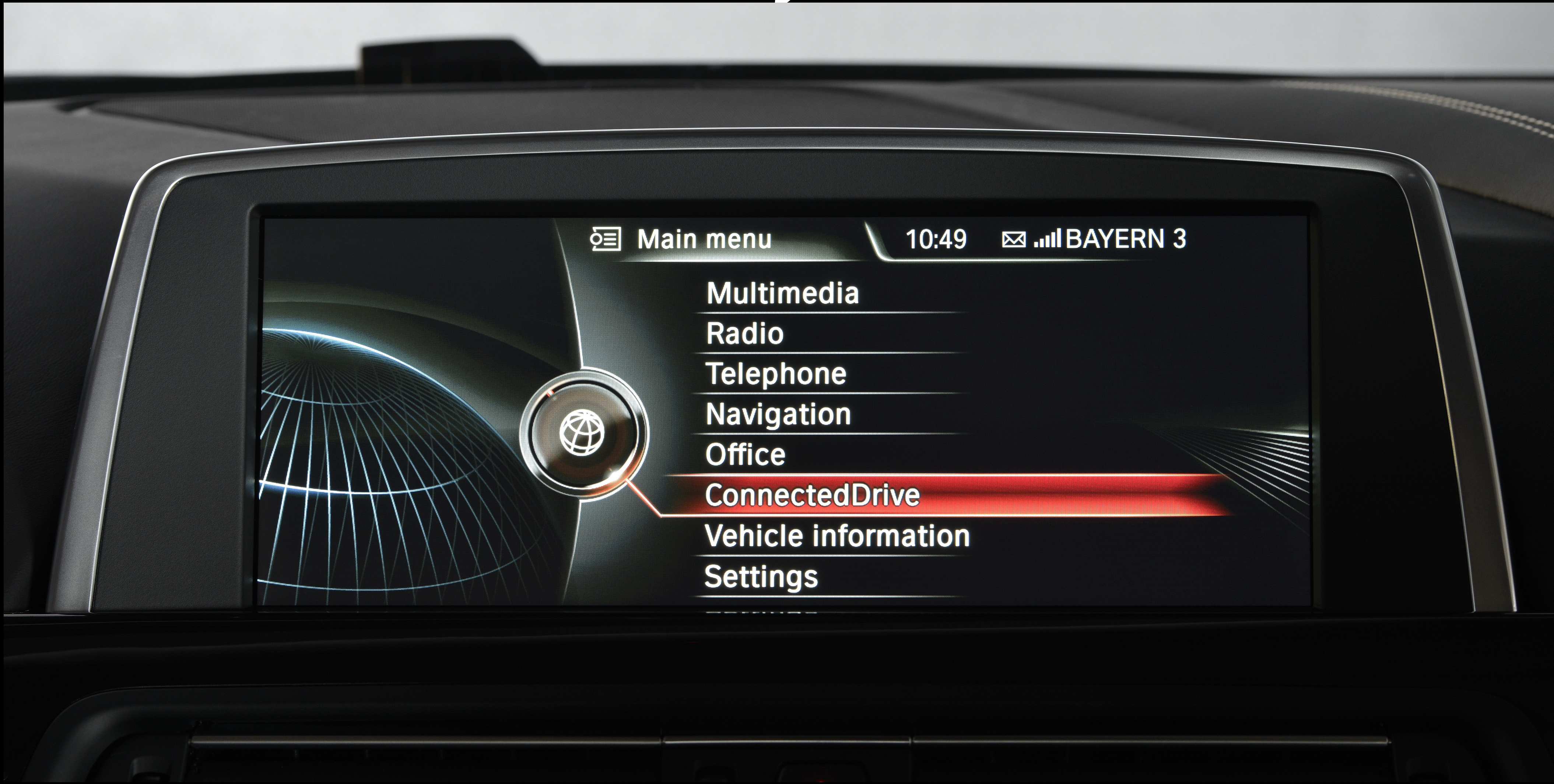
68 / 100

Acceleration

Gear shift

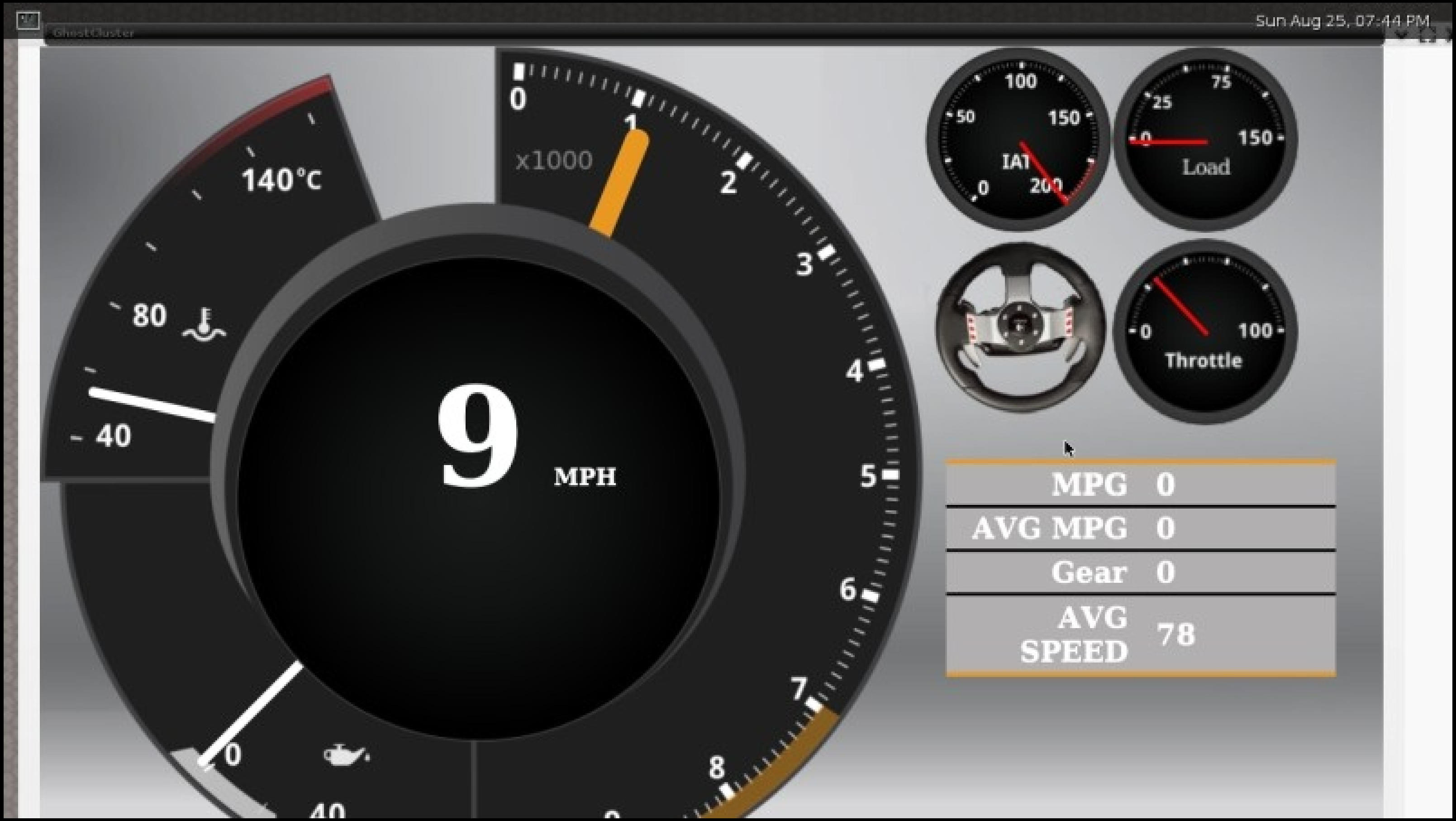
Anticipation

IVI already exists....



IVI already exists....

IVI already exists…



MPG	0
AVG MPG	0
Gear	0
AVG SPEED	78

The Linux / free software field

GENIVI: Car industry consortium for IVI middleware

- OEMs: BMW, GM, JLR, Honda, Nissan, Renault, Peugeot, Volvo
- Tier-1s: Alpine, Bosch, Clarion, Pioneer, etc.
- "makes" a specification and compliance testing program
- target is application developers (including in-house)
- ALSO makes some software components where pieces are missing

The Linux / free software field

Automotive Grade Linux (AGL): LF workgroup

- OEMs: Toyota, JLR, Nissan
- broader focus than GENIVI; goal is adoption of Linux
- "makes" no product; establishes issues and coordinates
- target is OEMs
- ALSO makes some software components as demonstrations

The Linux / free software field

Tizen IVI: IVI flavor of the Tizen distro platform

- Tizen is a base distro for CE devices that tracks mainstream Linux
- different lines require different add-ons (phone, IVI, TV, signage?)
- "makes" installable system images, is AGL demonstration platform
- target is OEMs and Tier-1s
- ALSO makes some software components; bundles GENIVI

The Linux / free software field

Open Automotive Alliance: Google Android workgroup(?)

- Recently-announced (CES 2014)
- Emphasis is improving Android experience (i.e., tethering)
- Deliverables T.B.D.
- Similar to Open Handset Alliance
- At present, essentially a Google "partner program"

The Linux / free software field

AUTOSAR: Automotive industry alliance

- Less well-defined limits
- Targets ECM and IVI and more
- Not as open as GENIVI
- Unclear relationship at present

The Linux / free software field

Car Connectivity Consortium (CCC)

- Sole product is the MirrorLink phone-tethering standard
- Consortium name-inflation
- Not the same as Miracast
- This just in: http://dvdhrm.wordpress.com/?p=326

The Linux / free software field

meta-ivi: Yocto layer

- Used for building embedded distribution
- GENIVI contributes to
- Not really end-user stuff

The Linux / free software field

W3C Automotive and Web Platform BG

- World Wide Web Consortium **business** group ($)
- Working on vehicle-centric APIs for HTML5 and browser
- This just in: https://rawgithub.com/tripzero/automotive-bg/master/vehicle_spec.html

The Linux / free software field

OpenXC: Ford developer program

- Arduino-based hardware in conjuction with Bug Labs
- Target is Android apps
- Read-only APIs
- Released a lot of data publicly

The Linux / free software field

DIYEFI: Homebrew ECM project

- Hardware: FreeEMS and limited MegaSquirt derivatives
- Tuning application MegaTunix
- Very DIY; If you weren’t already making ECM plans, not for you
- Best fit for older cars; others look into chipping community

Current status (what runs)

GENIVI

- Releases based on Baserock and Yocto
- Active application projects at projects.genivi.org
- Key projects: AF_Bus, Audio Manager, Diagnostic Log and Trace (DLT), Layer Manager, App Lifecycle Management, SmartDeviceLink
- Ongoing work on LXC containers

Current status (what runs)

AGL

- Ran 2013 contest for GUI development
- My mirror at https://github.com/n8willis/ivi-demonstrator
- AGL discussion list very open and active
- Ongoing work on features (without rolled releases)
- Excellent series of web lectures

Current status (what runs)

Tizen IVI

- Releases monthly/snapshot images for x86 and ARM
- Git source available at tizen.org
- Wiki and discussion list both very and active
- Incorporates GENIVI code
- Custom software: Automotive Message Bus (AMB), Homescreen, Media Player, HTML5 runtime (W3C Auto), GhostCluster instrumentation, Hands-free dialer, Murphy policy manager
- Ongoing work on speech recognition

Current status (what runs)

Other projects

- Navit, OpenStreetMap, and GeoClue
- Ubuntu IVI and Ubuntu Touch
- Large selection of homebrew and standalone OBD-II connectivity apps

Current status (what doesn't run)

Considering mostly Tizen IVI

- Configuration tools are almost all CLI
- APIs still low-level (e.g., sensors and startup, not Telepathy and Avahi)
- Multi-seat
- App installation

Current status (what doesn’t run)

Unaddressed areas

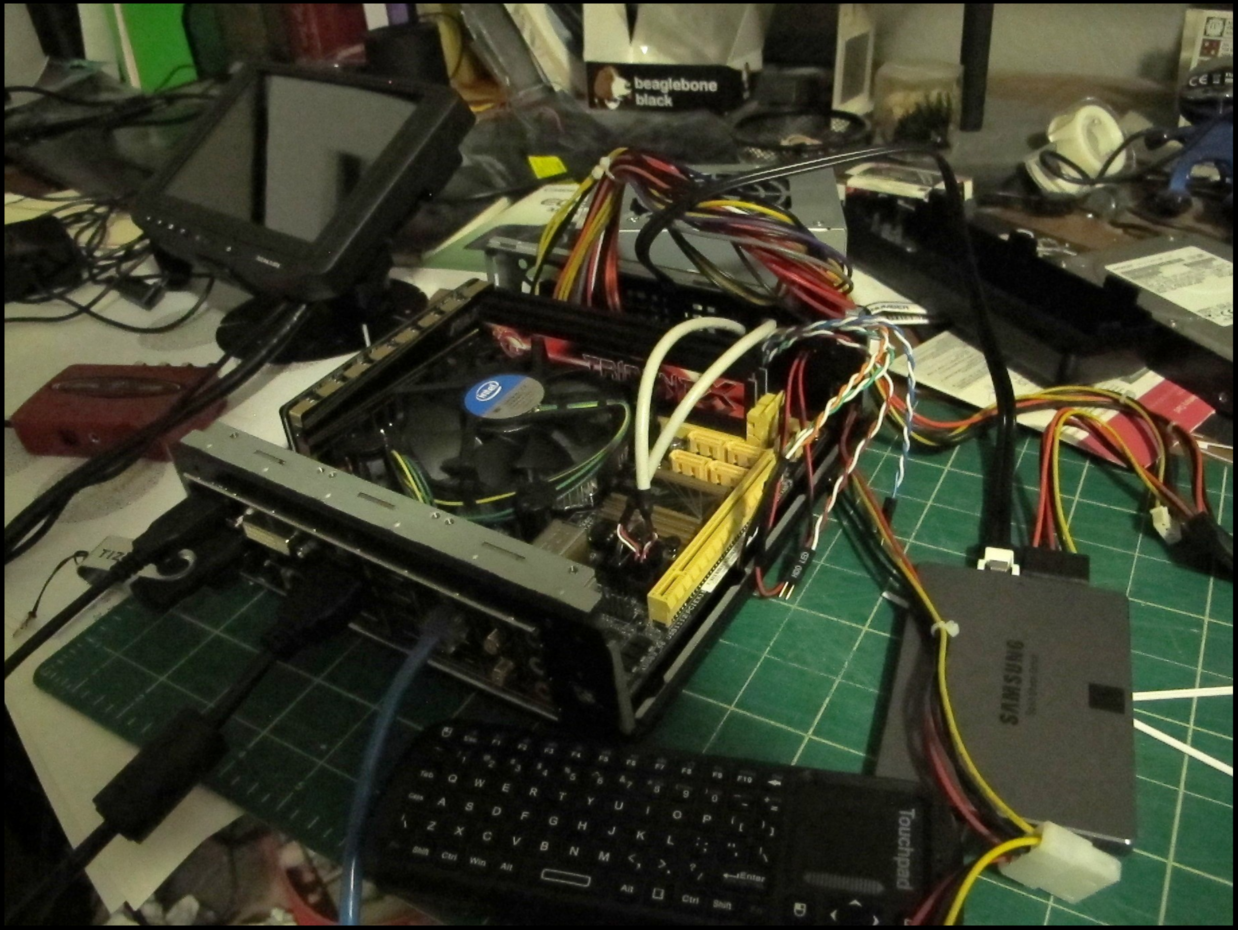
- Traffic interfaces
- How OEMs will do source releases
- V2X networking
- Over-the-air updates

Building your own IVI machine

Tizen IVI is the only full-featured option

- Target hardware is limited: current-generation Intel or PandaBoard
- Hardware installation is entirely DIY
- Specialty products required from mp3car.com or similar
- Difficulty akin to redoing car stereo from scratch; price akin to building mid-range PC from scratch

Building your own IVI machine



Building your own IVI machine

My personal experience

- Micro-ITX board, trunk installed in 2005 Mustang
- 7-in Xenarc display
- Tizen IVI
- M4ATX DC-DC power supply
- Whole mess of cables and wiring crammed in everywhere

Building your own IVI machine

My personal experience

- Current status: hiccup with motherboard (UEFI?)
- Lessons: wiring and power are as big of a challenge as anything else
- You must configure on the bench first
- Daily usage reveals limited package selection
- Updating is a very retro experience
- OpenXC is great, even if you don't use it

Building your own IVI machine

My personal experience

- Other interesting option: Ubuntu Touch
- Gesture support actually works
- HTML5 apps from phone/tablet are ripe for cannibalization
- Updating is much nicer
- No IVI-specific software

Challenges going forward (longer term)

Projects focused on OEM deliverables

- Development hardware non-existent (essentially)
- SDKs slow to appear (Tizen IVI last month; GENIVI soon...)
- Very little interaction with rest of free software community

Challenges going forward (longer term)

GPL3 aversion

- Concern is "liability" - safety and security (which are different)
- My boring slides: http://www.freesoftwhere.org/2013/11/01/linux-and-the-automotive-security-lab/
- Regulation is hard to route around
- No clear answer

Challenges going forward (longer term)

Specifications

- Open versus closed – e.g., MirrorLink versus Miracast
- V2X
- CAN bus replacement protocols

So what should you do?

Build your own hardware!

- Start small: OBD-II
- Tackle audio second
- Tackle navigation third
- All-out IVI last

(and there's one step missing….)

So what should you do?

Compile the IVI software for your Linux machine

- Audio management has **many** other applications
- SmartDeviceLink and Hands-free dialer apply to desktops too (sorta)
- HTML5 is the be-all and end-all
- Bug reports are valuable all the time

So what should you do?

Build your software for an IVI machine

- May be most important idea of all
- We are at the beginning of adoption curve
- If you wait until cars are shipping, the missing components will be filled by proprietary software vendors first (e.g., FourSquare versus pump.io)
- When was the last time FOSS had a head-start and no entrenched monopoly to unseat?

So what should you do?

Do some outreach

- Developer outreach for IVI
- Free software outreach in car forums
- Request some source code!

End of the road?

Ask me more: nate@lwn.net