EVIDENCE®

EMBEDDING TECHNOLOGY

An Open-Source RTOS for the automotive market

Paolo Gai, CEO
Evidence Srl

The company

Founded in 2002, we do custom design and development of software for small embedded devices

~20 qualified people with an average age of 34 years, 30% PhD

Experience in automotive, industrial, white goods

RTOS and MCU skills

- OSEK/VDX, AUTOSAR,
- Automatic code generation

Embedded Linux skills

- 8 Yrs experience in custom BSPs, U-Boot, kernel drivers,
- Initial developers of the SCHED_DEADLINE patch

Erika Enterprise

ERIKA ENTERPRISE

http://erika.tuxfamily.org

- Free RTOS for automotive devices
- Open-source license allowing static linking of closed source code
- ERIKA Enterprise is the first and only OSEK/VDX certified open-source RTOS

- ERIKA Enterprise also implements an API inspired to a subset of the AUTOSAR API
- We are looking forward for an integration with Linux on multi-core devices

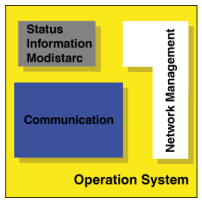
Agenda

- OSEK/VDX RTOS and AUTOSAR for a Linux User…
- Main characteristics of Erika Enterprise
- Success stories
- Setting up and using the development tools
- How the RTOS is tested
- Towards a fully integrated Open-Source solution with Linux and Erika Enterprise
- Summary and Q&A

Motivation

These webinars talk about Linux and open-source tools and how they can be used into automotive products

…but… what was the status of automotive before Linux ?



AUTOSAR

Since 90s, the OSEK/VDX standard (then recently AUTOSAR):

- API specification
- For microcontroller-based automotive systems
- Low footprint (in the order of Kbytes)

Open-source?

OSEK/VDX and AUTOSAR systems are traditionally:

- proprietary systems
- closed source
- dedicated to specific microcontrollers

Moreover, from the open-source point of view,
they lack a common platform aggregating a critical mass
needed to create a first working environment

Why?

Because automotive companies in the past lacked a culture of sharing microcontroller code

- Every company is implementing (or buying) every time the same subsystems
  - RTOS (OSEK/VDX or AUTOSAR)
  - Device Drivers
  - Diagnostic protocols

there is an opportunity to

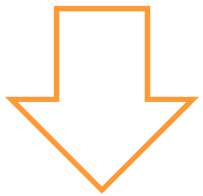
share

software components not in the core business

sharing in automotive

Sharing source code in automotive means:

nobody makes a free gift to competitors



we need a platform
where each company
adds a small part

Example: Linux in infotainment

Many new infotainment systems on cars are based on Linux and Android

Automotive Grade Linux - http://automotive.linuxfoundation.org

Tizen - https://www.tizen.org

Genivi - http://www.genivi.org/

TIZEN

GENIVI

… just have a look at the news on WIRED:

"The Next Battleground for Open vs. Closed: Your Car" (Oct. 2012)

Oct 12th 2012

WIRED

"A luxury automaker recently told me its IVI system contains about 1,900 use cases – "of which we only consider about 3 percent unique to our products; the other 97 percent are common across all car companies."

"But here's the paradox: The automotive industry is going to have to collaborate in order to differentiate."

"Competitors collaborate on the code and requirements to produce a common base, upon which they differentiate and compete with each other."

http://www.wired.com/opinion/2012/10/automakers-become-software-makers-the-next-battle-between-open-and-closed/

What about small ECUs?

Is this happening on small ECUs? Yes!

- OSEK/VDX / AUTOSAR RTOS
- small footprint constraints

http://erika.tuxfamily.org

http://www.arccore.com

http://trampoline.rts-software.org

Main difference: licensing

The three projects differ for the licensing of the target code:

GPLv2+Linking exception

Dual Licensing (GPL + Commercial)

LGPL v2

Only Erika Enterprise allows static linking of proprietary code!

More details on ERIKA Enterprise

Something about ERIKA Enterprise

http://erika.tuxfamily.org

- ERIKA Enterprise is an RTOS OSEK/VDX certified
- ERIKA Enterprise implements an API inspired to a subset of the AUTOSAR API
- With a suitable open-source license allowing static linking of closed source code
- Typical footprint aroun 2-4KB Flash

OSEK/VDX API support

ERIKA Enterprise supports the OSEK/VDX API

Complete implementation of the following components:

- OSEK OS (BCC1, BCC2, ECC1, ECC2)
- OSEK OIL
- OSEK ORTI using Lauterbach Trace32
- OSEK COM (CCCA, CCCB)

- Additional research conformance classes implementing Earliest Deadline First and Resource Reservation (similar to the SCHED_DEADLINE patch for Linux)

ERIKA - OSEK/VDX for a Linux user

What	Linux	ERIKA - OSEK/VDX
Flash Footprint	4-32 MB Flash for a minimal system	2-4 KB Flash
RAM	8-64 MB	Hundreds of Bytes
API	POSIX / pthreads	OSEK/VDX http://www.osek-vdx.org
Static/Dynamic approach	Dynamic	Static, configured with an OIL file or AUTOSAR XML
CPU support	32-64 bit	Down to 8 bit MCUs
Filesystem	Yes	No
MMU support	Yes	No (Yes for AUTOSAR)
Device Drivers	Yes	No (Yes for AUTOSAR, but configured «more statically»)
Execution from flash	No	Yes

ERIKA - OSEK/VDX for a Linux user (2)

What	Linux	ERIKA - OSEK/VDX
Certification suite	No	Yes
Real-time support	Available through patches (RT-PREEMPT, RTAI, Xenomai, SCHED_DEADLINE…)	Native support for Fixed priority, Preemptive and non preemptive stack
Stack sharing	No	Yes
Immediate Priority Ceiling	Yes (with realtime priorities)	Native
Multicore support	Yes, SMP	Yes, Static Partitioning
IRQ handling	in the kernel	in the Application, exposed in the API
Blocking primitives	full support	limited support
Conform. classes	Kernel configurations	Yes, Used to limit footprint

Static Approach

Everything in ERIKA is static

- data structures allocated in Flash
- everything is configured through a text file named OIL

RT DRUID

- Configuration tool
- interprets the OIL file and generates the kernel data structures and makefile
- based on Eclipse
- scriptable

Licensing

ERIKA ENTERPRISE

- License: GPL + Linking Exception
  - http://en.wikipedia.org/wiki/GPL_linking_exception
  - Proprietary applications can be statically linked with the RTOS!

RT DRUID

- License: EPL Eclipse License
  - http://en.wikipedia.org/wiki/Eclipse_Public_License

Industrial usages: Cobra AT

The first one was Cobra AT

with:

2009 – feasibility for a OEM product
based on Freescale S12XS

2012 – integration in an after-market / OES product
based on Freescale S12G

(integration work performed by Massimiliano Carlesso)

Magneti Marelli

Then came Magneti Marelli Powertrain Bologna

With support for:

- PPC MPC5674F (Mamba)
- MPC5668G (Fado)
- Multicore support
- AUTOSAR Memory Protection

Then…

Aprilia Motor Racing on PPC

FAAM on S12XS

esi-RISC port (made by Pebble Bay)

A white goods company

TI Stellaris Cortex M4F, Renesas 2xx and AUTOSAR-like drivers

Demo @ Freescale Automotive seminar

Other automotie and 2-3 wheels companies

PPC Leopard and Infineon AURIX

Hardware supported

ERIKA Enterprise supports the following microcontrollers:

Microchip	PIC24, dsPIC, PIC32
Altera	Nios II
ARM	ARM7, Cortex M0, Cortex M4
Lattice	Mico32
Freescale	S12XS, S12G
Freescale	PPC z0, z4, z6, z7 (Mamba, FADO, Leopard)
Infineon	Tricore AURIX
Atmel	AVR5, Arduino
Ensilica	esi-RISC
TI	MSP430, Stellaris Cortex M4
Renesas	R21x

A Porting guide available on the ERIKA Wiki!

Multi-core support

ERIKA Enterprise supports multi-core designs

- code partitioned in the various cores
- a copy of the RTOS for each core
- interprocessor IRQ and atomic operations on the bus
- data and code sharing
- support for Lauterbach debuggers

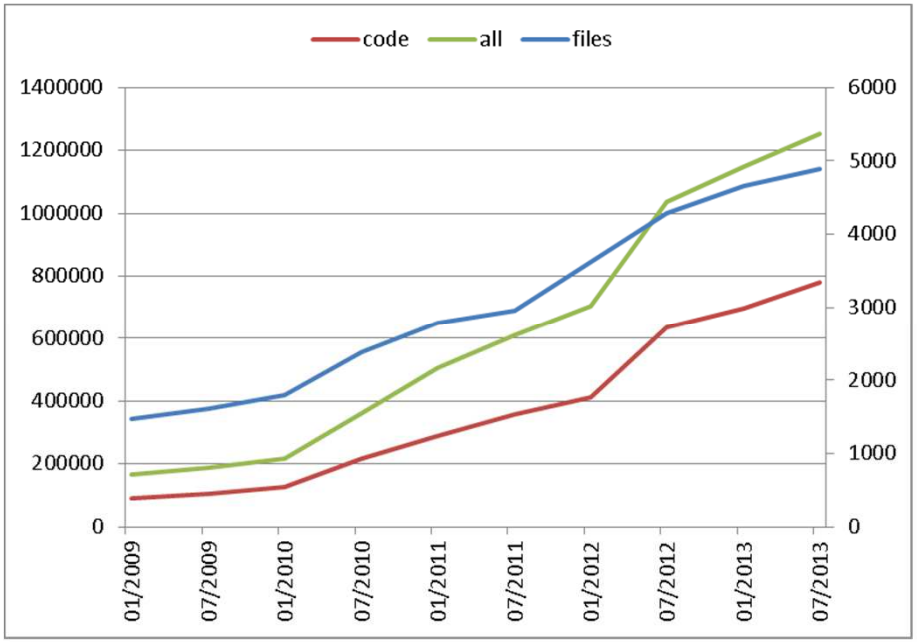
Currently available for

- Altera Nios II 8.1
- Freescale MPC5668 (FADO), cores e200 z6+z0
- Freescale MPC5643L (Leopard), dual z4
- Tricore AURIX

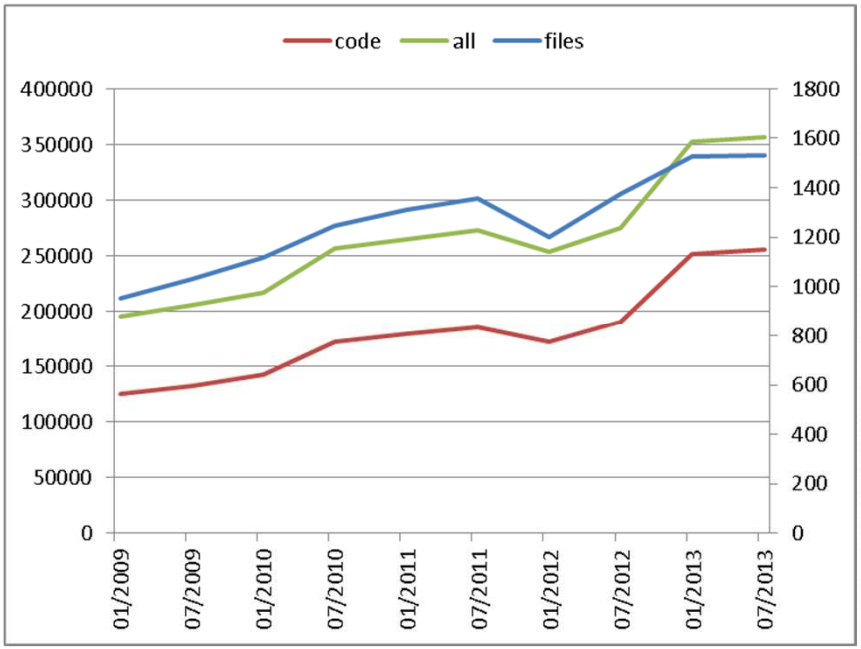
code size

The code base increased 3x from 2009 to 07/2013

ERIKA Enterprise



RT-Druid



OSEK/VDX compliance

OSEK/VDX compliancy done for ARM Cortex M4F
in August 2012

ERIKA Enterprise is the first open-source kernel which has been certified OSEK/VDX compliant

The compliancy is linked to the following:

- RTOS version
- compiler and development environment
- microcontroller

AUTOSAR compliance

- ERIKA is not yet AUTOSAR OS Compliant
- A subset of the AUTOSAR Memory protection API for PPC MPC5674F (Mamba) has been implemented
- RT-Druid is capable of importing AUTOSAR XML produced by SystemDesk
- we are currently implementing a subset of the AUTOSAR requirements for the AUTOSAR OS.
  - we started from the OSEK/VDX implementation available for ERIKA
  - we implemented part of the requirements linked to the currently implemented API, plus a subset of the memory protection API
  - In addition to the basic OSEK/VDX requirements in AUTOSAR OS, at least the following additional AUTOSAR requirements have been implemented:

    OS398, OS242, OS399, OS384, OS304, OS299, OS092, OS054, OS449, OS450, OS239, OS071, OS236, OS112, OS225, OS237, OS246, OS085, OS301, OS383, OS392, OS051, OS088, OS056, OS367, OS052, OS069, OS368, OS369, OS424, OS425, OS093, OS439, OS070

AUTOSAR-like drivers

We developed a set of AUTOSAR-like drivers for various architectures. These drivers include (Sept 2012):

- Cortex M4 Stellaris (DIO, DMA, GPT, MCU, PORT, SCI, SPI, WDG)
- MPC 56xx (MCU, PORT, GPT)

Configuration of the driver is currently done using C templates (not yet from the AUTOSAR XML)

Eclipse-based configurator available on

http://www.evidence.eu.com/products/eforms.html

MISRA C compliancy

A subset of ERIKA Enterprise has been checked for MISRA C compliancy

- tools used: FlexeLint 9.00h
- subset tested
  - OSEK kernel conformance classes, plus FP conformance class
  - CPU: PPC e200 single and multicore core, with/without memory protection
  - Compiler attributes for Diab 5.5 for PPC
- FlexeLint has been configured using Magneti Marelli Lin 7.10, with some additional exceptions which will be documented soon on the ERIKA Enterprise Wiki

Regression tests

Continuous integration test environment based on Jenkins

ERIKA Enterprise and RT-Druid have been tested using:

- Official OSEK/VDX conformance test suite
- Regression tests derived from the MODISTARC tests published on the OSEK/VDX website
  - See http://erika.tuxfamily.org/wiki/index.php?title=Main_Page#Regression_Tests

Benchmarks

Footprint statistics and Benchmarks have been published on the Wiki.

- A typical scenario of 16 tasks + resources + alarms uses 2-4 Kb flash depending on the MCU

- Timings of the primitives are in the range 2-10 usec

They are in line with other commercial offerings

The development community

http://erika.tuxfamily.org

- SVN repository open to the public
- Wiki and forum
- Application notes
  - Template system available in RT-Druid
- libraries for
  - console
  - uWireless (802.15.4 with beaconed mode / GTS support)
  - ScicosLab Libraries
  - Motor control
  - TCP/IP
  - CMOS Cameras, tracking
  - USB
  - various sensors
  - ball & plate, inverted pendulums, robot swarms

What’s next?

Infotainment, Linux, and multicores

- Next generation infotainment systems will be multi-core
- They are likely to host more than one OS

What about

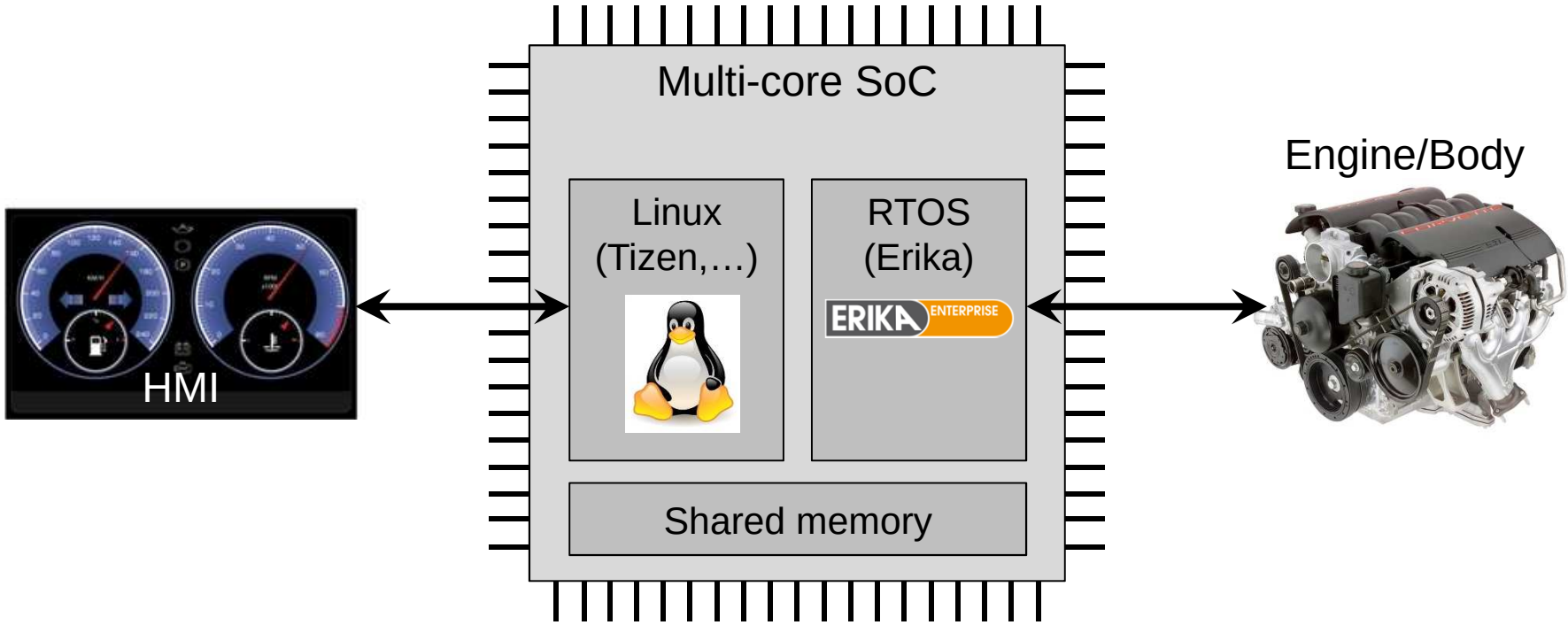
creating a complete open-source environment
for automotive systems integrating

Infotainment + OSEK/VDX/AUTOSAR

on the same chip?

Towards a fully Open-Source platform

We envision the possibility to exploit multi-cores to run Linux and Erika Enterprise complementing each other!



Various approaches…

Various approaches could be possible:

ERIKA boots from Linux

- Linux limited to 1 CPU
- Linux controls the ERIKA boot

ERIKA boots from U-Boot

- modified U-Boot to boot both ERIKA and Linux

Hypervisor-like approach

- both ERIKA and Linux as hypervisors

References

- Website: http://erika.tuxfamily.org
- Wiki: http://erika.tuxfamily.org/wiki/
- Forum: http://erika.tuxfamily.org/forum/
- Subversion repositories:
  - Erika Enterprise: svn://svn.tuxfamily.org/svnroot/erika/erikae/repos/ee/trunk/ee
  - RT-Druid: svn://svn.tuxfamily.org/svnroot/erika/erikae/repos/rtdruid/trunk/rtdruid

Questions ?

Contacts

Evidence Srl

http://www.evidence.eu.com

info@evidence.eu.com