



EVIDENCE[®]

EMBEDDING TECHNOLOGY

Summary

- About me
- About Evidence Srl
 - Who we are: company profile
 - What we provide: products and services
 - Open-Source business model
 - Relationships with Open Source communities
 - How we work

About me

About me

Claudio Scordino: claudio@evidence.eu.com

Academic curriculum:

2003: Master Degree in Computer Engineer (Pisa)

2007: PhD in Computer Science (Pisa)

2006-2008: Software Engineer at Evidence Srl

- Porting of Linux to embedded boards
- Development of Linux drivers

2009-Today: Project Manager at Evidence Srl

Company profile

the company

founded in 2002, headquarters in Pisa, Italy



spin-off of the ReTiS Lab.
at the Scuola S. Anna - Pisa

we provide **innovative software solutions**
for the design and the development of
real-time embedded systems,
with a special focus on **multi-core hardware
platforms**

background



strong collaboration with the
ReTiS Lab of the Scuola Superiore S. Anna

15+ years of of experience in real-time research
strong commitment in the embedded systems area

collaborations with research centers in Europe and USA

(some) customers and partners

OSEK,
microcontrollers,
schedulability analysis



Linux,
SW devel.



Partnerships



StartCUP 2005



Evidence won the first prize at Start Cup Pisa 2005

most innovative young Italian entrepreneur

March 12, 2007

Evidence founder **Paolo Gai** selected by “**Corriere della Sera**” as one of the most innovative Italian young entrepreneurs

Sondaggio Un panel di dieci esperti elegge per il Corriere Economia il Gotha degli under 50. Che saranno famosi

Imprenditori emergenti: ecco i 30 più innovativi

Hanno una forte propensione internazionale. Sanno usare gli strumenti finanziari, dal private equity alla Borsa. Nuovi prodotti, nuovi processi, old economy reinventata

EU Projects (FP6)

- **ARTIST2**

NoE for spreading out the knowledge on real-time systems

- **FRESCOR**

Framework for Real-time Embedded Systems based on COntRacts



- Evidence will lead a Task that aims “[...] to provide support for the contract model [...] on reconfigurable multiprocessor platforms, and specifically for the *Nios II processor on ALTERA FPGA's boards*. [...] This task will also validate the results by providing a set of demo programs able to use all the support contract model features and dynamically reconfigure themselves.”

- **INTEREST**

INTEgrating euRopean Embedded Systems Tools



- mainly integration with third party tools coming from ETAS (ASCET), Esterel (SCADE), and others european tools

EU Projects (FP7)

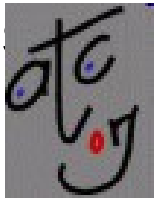
- **ARTIST Design**

NoE for spreading out the knowledge on real-time systems

- **ACTORS**

Adaptivity and Control of Resources in Embedded System

- Other partners: Ericsson Res. Labs, Scuola S. Anna, TU Kaiserslautern, Ec. Polytechnique Féd. de Lausanne, Lund University, AKAtch SA



- **INTERESTED**

INTER-operable Embedded Systems
Toolchain for Enhanced rapid Design,
prototyping and code generation

- Other partners: ESTEREL, AbsInt, TTTech, Syntavision, UNIS, Artisan Software, Sysgo, Airbus, Magneti Marelli, CEA, Thales, Siemens



products and services

Small embedded systems

- ERIKA Enterprise – RTOS
- RT-Druid – development environment
- Flex – embedded board

Larger embedded systems

- Evelin SDK – development environment
- Evelin Linux – operating systems
- consulting services – consulting and training

ERIKA Enterprise and RT-Druid

ERIKA Enterprise

- open source
- OSEK-like RTOS for minimal embedded systems
- 1-4 Kb ROM footprint
- enhanced scheduling algorithms
- support for Lauterbach Trace32
- available for dsPIC, Nios II, ARM7, Tricore, PPC, H8, ST10, AVR, and others



RT-Druid

- RTOS configuration using OSEK OIL
- schedulability analysis
- integrated in eclipse.org



multicore support on Nios II

ERIKA Enterprise has been successfully ported to



- the first multicore RTOS based on the Nios II soft-core on Altera FPGA
- advanced multicore scheduling techniques
- code placement and multicore synchronization
- listed on the Altera Web Site and on the Nios II Forum

success story: Magneti Marelli Powertrain

- the RT-Druid tool has been funded and used by Magneti Marelli
- integrated in the MM development flow
- used for:
 - specification of the mapping of the various functionalities
 - schedulability analysis
 - sensitivity analysis
 - integration into the AUTOSAR build flow



FLEX embedded boards



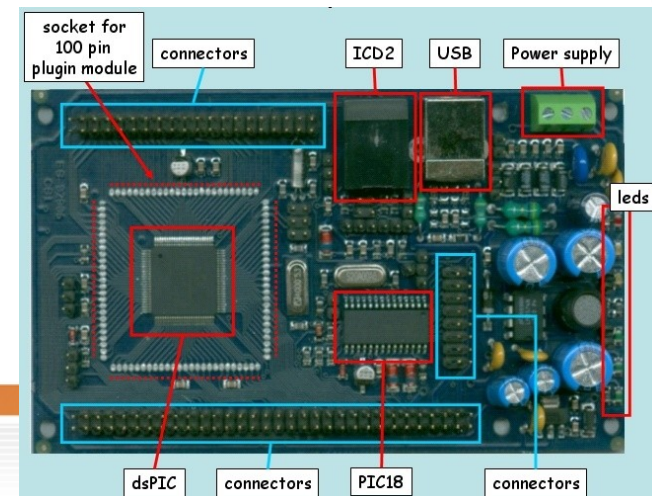
produced by Evidence & Embedded Solutions
cheap, small, easy-to-use evaluation boards



Typical applications:

- industrial sensing and control
- small robots
- wireless sensor networks
- demo boards for university labs

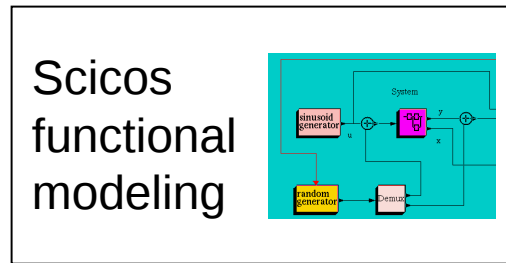
all rights reserved



fast prototyping using Scilab/Scicos

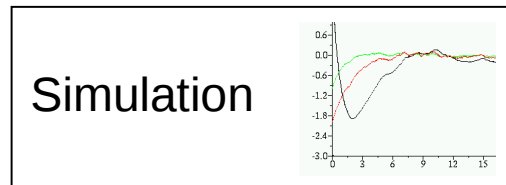


in collaboration with INRIA (FR) and Supsi Lugano (CH)



INRIA/SUPSI Code generator

USB Connection



Same Behavior!

HW + Erika Enterprise
FLEX

embedded Linux SDK



- Environment for embedded Linux development
- Allows to develop, build and debug your application as it was running on the target platform!
- Cross-compilation toolchain based on gcc
- C and C++ languages support
- **Automatic target emulation**
- GUI based on the well-known **Eclipse IDE**
- Graphical debugger integrated with GDB server



Evelin SDK - Supported targets



Supported targets:

- ARM7
- ARM9 little-endian and big-endian
- ARM11
- Renesas SH4

Evelin SDK – Eclipse IDE



The screenshot shows the Eclipse IDE interface during a debug session. The top toolbar contains various icons for debugging, including a play button (run) and a pause button (suspend). The 'Debug Console' shows the current task status. The 'Variables' view displays the current value of variables, with a table showing a variable 'i' with a value of 1074412984. The 'Registers' view shows the current values of registers. The 'Breakpoints' view shows the current position in the program. The 'Code Editor' shows the source code of the program, with a breakpoint set on the line `unsigned long int i = 0;`. The 'Console' view shows the output of the program.

Task status

Pause and step-by-step buttons

Current value of variables, breakpoints and registers

Breakpoints

Current position in the program

Evelin BSP

Evelin Linux BSPs

- Build scripts for automatic generation of initial firmware images
- U-Boot bootloader
- Startup scripts
- Linux kernel sources with drivers for internal and external hardware
- Busybox
- Package management system and repository for target maintenance based on the apt-get Debian standard
- Graphical libraries based on GTK-FB

services

Consulting

- Real-time and embedded systems
- Planning the right move to Linux
- Evaluation of real-time and timing requirements
- Selection of the most appropriate real-time solution
- Installation of development tools

Development

- Kernel drivers for custom hardware
- Custom Linux BSPs

Training about the usage of our products

success story: Linux on ConPro9

Evidence developed the Linux distribution for ConPro9 produced by Sistemi Avanzati Elettronici and MPL



Sistemi
Avanzati
Elettronici

- HW: ARM9, touchscreen, digital/analog I/O, CAN
- Linux distribution: scratchbox, Debian packaging, GTK/FB, applications, Lauterbach Trace32 support

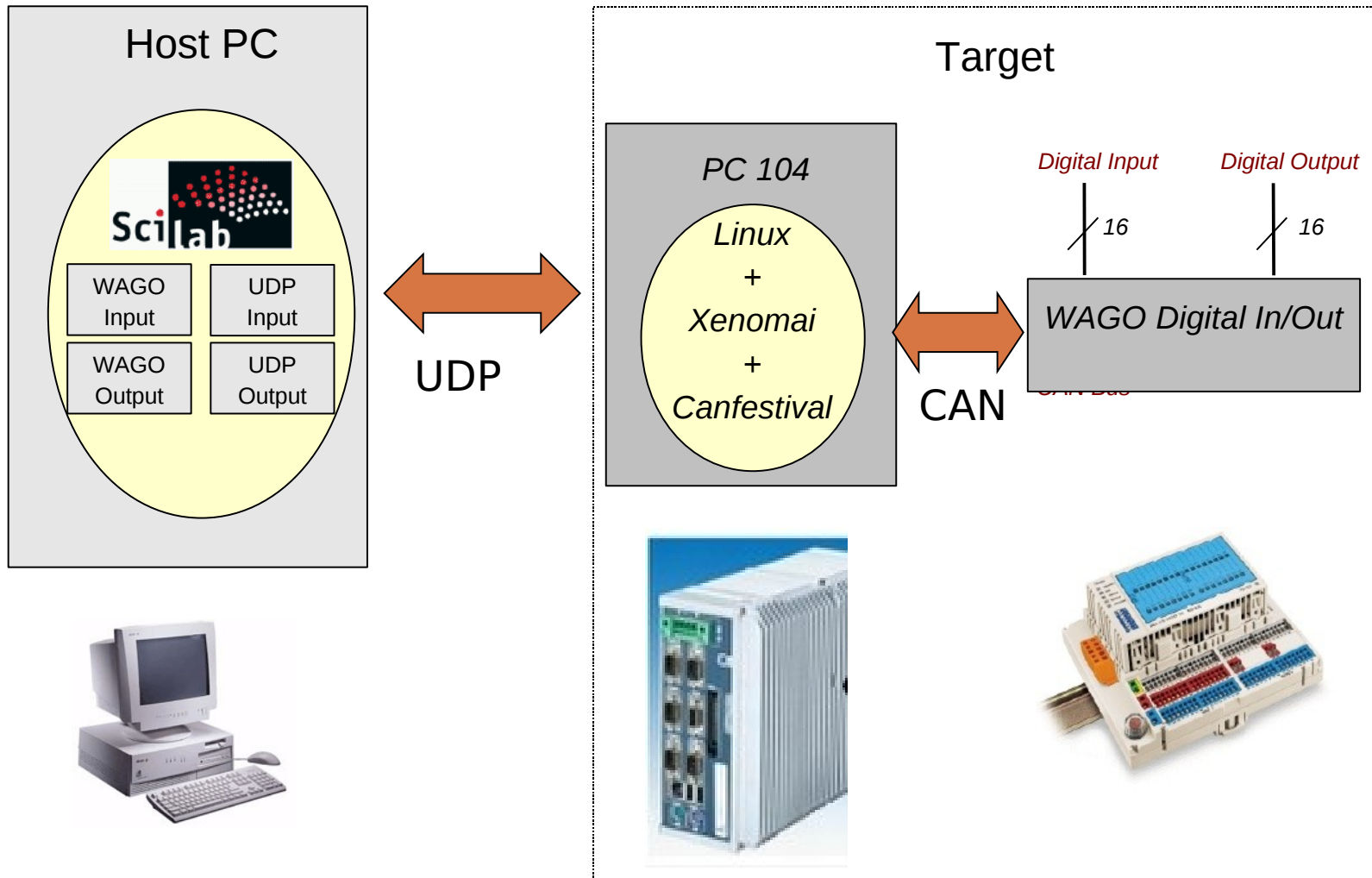
success story: navigation systems



- Support for a complete embedded linux solution
 - Implementation of custom drivers
 - Design of a proper boot sequence
 - Optimization of the system footprint
- Support for application design
 - Design and integration of custom application inside an embedded Linux hardware



success story: CANOpen, code generation



who we are

- Evidence is a spin-off company of the ReTiS Lab of the Scuola Superiore S. Anna
- we provide solutions for firmware development
- especially for
 - small microcontrollers
 - multicore SoC
 - embedded linux systems

conclusions: what we provide

- RTOS for small microcontrollers
- Flex embedded boards with microcontrollers
- OS and Firmware for Linux-based embedded devices
- tools for the design of embedded control systems
- training

Business model

Open Source myths

- Open Source doesn't mean "no money"
 - People (Linus included) and companies (e.g. RedHat, Google) make money with Linux
 - We do too :)
- But...
 - It's not trivial
 - A good business model is needed
 - Otherwise, the risk is bankruptcy (see Ubuntu)
- Several Open-Source licenses exist

GNU General Public License (GPL)

- Published by the Free Software Foundation
- Used for Linux
- Allows to modify and redistribute (even sell) the code as long as the recipient has access to the source and maintains the same rights
- No warranty
- Modifications are and remain GPL
- The source code must be released to the end-user

GNU Lesser General Public License (LGPL)

- Published by the Free Software Foundation
- More permissive than GPL
- Proprietary programs can be linked to LGPL libraries
- No warranty

BSD Licenses

- Family of licenses
- Used for BSD by the Berkley university
- Modification and redistribution in binary form is permitted
- It just acknowledges original developers
- No warranty
- Most permissive license

Advantages of Open-Source code

- Code might not be written from scratch
 - Reuse of code from similar applications
- Free help and support from community
- Positive image of the company
 - Easier to hire talented developers
- Better code
 - Reviewed and maintained by others for free

Our business model

Part of our business model is based on Open-Source...

Services:

- GPL: the customer pays for development of open-source code
 - Enforced by a lower cost than a proprietary solution
 - Examples: porting of Linux, implementation of drivers
 - Hard in Italy, where customers don't know open-source
- !GPL: we use LGPL code already existing (if any)
 - LGPL libraries (e.g. GTK+), or BSD code

Our business model

Part of our business model is based on Open-Source...

Services:

- GPL: the customer pays for development of open-source code
 - Enforced by a lower cost than a proprietary solution
 - Examples: porting of Linux, implementation of drivers
 - Hard in Italy, where customers don't know open-source
- !GPL: we use LGPL code already existing (if any)
 - LGPL libraries (e.g. GTK+), or BSD code

Working with community

- Linux kernel:
 - Scheduler (Peter Zijlstra, Ingo Molnar, Thomas Gleixner)
 - Drivers (Andrew Morton, Alan Cox)
 - Porting to new architectures: ARM, SH, Cris
- U-Boot bootloader:
 - Drivers and porting
- Anjuta IDE:
 - Plugins
- Scratchbox dev.environment:
 - Bug fixing

Working with community (cont'd)

- Qemu emulator:
 - Bug fixing
- Scilab/Scicos environments:
 - Automatic code generation
- Xenomai:
 - Porting and bug fixing

Tools

- Operating systems:
 - Ubuntu and Debian
- Software development:
 - Versioning: CVS, SVN, GIT
 - Gcc and make
 - Vim and Eclipse
- Documentation and slides:
 - LaTeX
 - OpenOffice

Our employees

- **Technical background**
 - Computer Science or Computer Engineering
 - Better from University of Pisa
 - PhD or Master degree

- **Passion for**
 - Programming
 - Learning
 - Open Source software
 - Linux :)

How we work

- Informal working environment
 - No tie required ;)
- Possibility of working from home
 - Depending on the kind of task
- Very flexible working time
 - Important thing: 40 hours/week
- Learn new things every day
- Training courses
 - About several topics

the end

Questions ?

