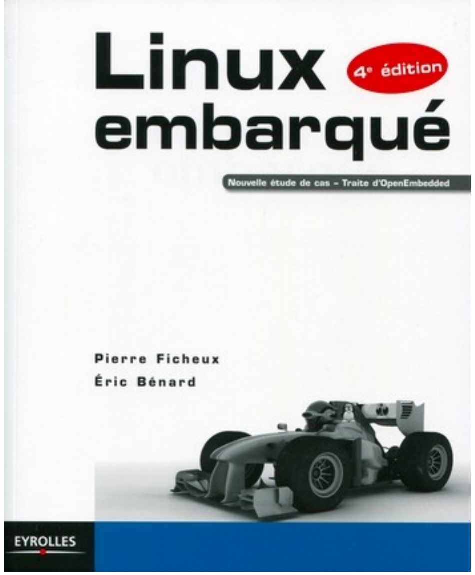
openPOWERLINK over Xenomai

Pierre Ficheux (pierre.ficheux@openwide.fr)

October 2015

$ whoami

- Free software enthusiast since 1989
- Linux user since 1992
- Author of 4 editions of « Linux embarqué » a french book about embedded Linux
- Managing editor of Open Silicium
- CTO @ Open Wide Ingénierie, a french software service company (Paris, Lyon, Toulouse, Grenoble)
- teacher @ EPITA (french computer science school)



Linux embarqué

4e édition

Nouvelle étude de cas - Traite d'OpenEmbedded

Pierre Ficheux

Éric Bénard

EYROLLES

N°144 DÉCEMBRE 2011

GNU LINUX MAGAZINE / FRANCE

Administration et développement sur systèmes UNIX

CODE / C / NOYAU / RT

JOUONS AVEC LE KERNEL !

- B.A.BA : POURQUOI ET COMMENT COMPILER UN NOYAU PERSONNALISÉ ?
- ÉCRIRE UN PILOTE RÉSEAU : UN EXERCICE TRÈS PARTICULIER
- TROIS EXEMPLES CONCRETS POUR ÉCRIRE VOS PILOTES TEMPS RÉEL AVEC RTDM

VITESSE / HTTP

Compactez votre site web pour le rendre léger et rapide :
1 – avec les outils UNIX
2 – en utilisant JavaScript

FS / DISTRIBUÉ

Haute disponibilité et analyse des performances avec un système de fichiers distribué Gluster 3.1.2

PARTAGE / FICHIERS

LinShare, une solution de partage de fichiers avec une interface web simple et jolie

KERNEL / NEWS

Explorez les nouveautés du noyau 3.1 : virtualisation Xen/KVM, debugging ptrace, réseau, FS, affichage, etc.

SERVEUR / HTTP

Installation et configuration de votre premier serveur HTTP Apache 2

DATAMINING / R

Réalisez un réseau de neurones et prévoyez le temps qu'il fera demain avec Rattle

ÉLECTRONIQUE / AVR

Découvrez la programmation avancée des microcontrôleurs 8 bits Atmel AVR

MARS / AVRIL / MAI 2013 N°6

Open Silicium MAGAZINE

INFORMATIQUE
OPEN SOURCE
EMBARQUÉ
INDUSTRIEL ET R&D

LE MAGAZINE DE L'EMBARQUÉ ET DE L'ÉLECTRONIQUE NUMÉRIQUE !

OPENRISC / FPGA

Introduction aux soft-cores avec OpenRISC et Linux sur Altera Cyclone Terasic DE0-nano p.6

SAM9 / SOM

Découverte, prise en main et mise en œuvre du SoM low cost Aria G25 d'ACME System p.18

C / RASPBERRY PI

La RPi pour les grands : création d'un serveur HTTP minimal tout en C permettant le contrôle des GPIO p.46

ARM / RT

TECHNOLOGIE AÉROSPATIALE : EXPÉRIMENTEZ UN RTOS RICHE ET MATURE !

INTRODUCTION AU TEMPS RÉEL AVEC RTEMS SUR MINI2440

- Mise en place du compilateur croisé x86/ARM
- Compilation de RTEMS et test en API classique et POSIX
- Débogage sous Qemu et avec sonde JTAG
- Test d'applications réseau
- Réalisation d'un pilote de périphérique

ANALYSE / PYTHON

Que faire lorsqu'un outil propriétaire de flashage refuse de fonctionner sur un système récent ? p.56

HACK / IR

Améliorez les capacités de vision nocturne/infrarouge et la discrétion de votre netcam p.74

À SAVOIR / SCIENCE

Petit détour dans l'infiniment petit pour comprendre le phénomène de (semi) conduction p.70

L 18310 - 6 - F: 9,00 € - RD

France Métro : 9 € / Belgique - Luxembourg : 9,50 € / Suisse : 14 CHF / DOM : 9,90 € / CAN : 15,50 $CA / N. CAL/S : 1200 CFP / POL/S : 1300 CFP

Agenda

- Industrial bus
- (open) POWERLINK introduction
- Linux and RT (PREEMPT-RT, Xenomai)
- OpenPOWERLINK over Xenomai architecture
- Problems, tests and results
- Future work

POWERLINK

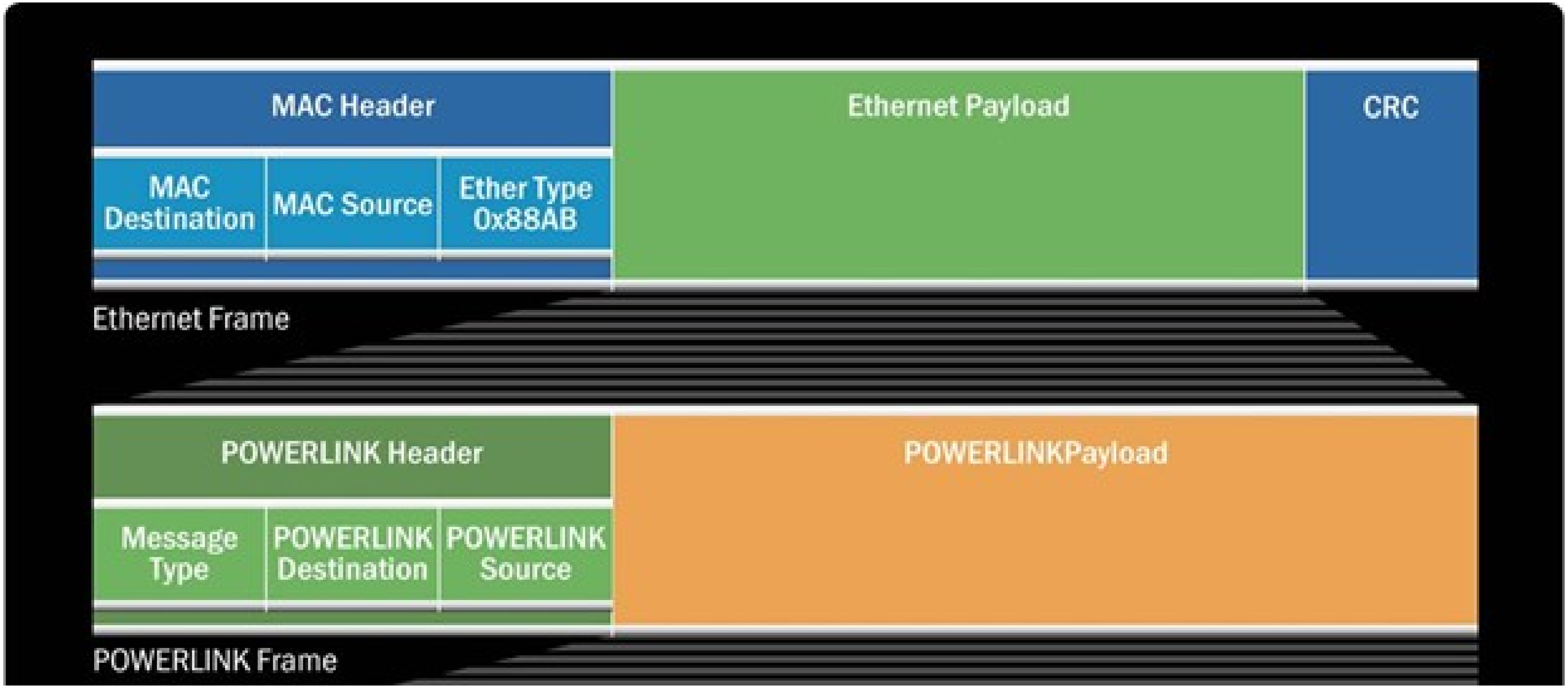
Industrial bus

- Used to connect industrial devices in real time mode
- Main standards are both “serial” and/or Ethernet
  - CAN
  - MODBUS (-TCP)
  - Profinet
  - EtherCAT
  - EtherNet/IP (IP for *Industrial Protocol*)
  - **POWERLINK** !
- Ethernet is a standard
  - Easy to integrate, cheap hardware and good performances (CAN is 1 Mbps)
  - Homogeneous networking (routing, etc.)
  - No RT because of CSMA/CD (collision detection)

POWERLINK

- Deterministic Ethernet based industrial bus
- Originally invented by B&R automation (Austria) in 2001
- Managed since 2003 by open organization EPSG (Ethernet POWERLINK Standardization Group)
- Leverage advantages of Ethernet for RT networking systems
- 1.1 M systems installed (#1 industrial Ethernet)
- Min cycle time is 100 µs, 240 nodes on a single network
- Works on standard NIC (software only) 802.3 compliant
- Avoid collisions thanks to a dedicated protocol :-)
- Open-source version (2.2.1) *openPOWERLINK* available from SourceForge

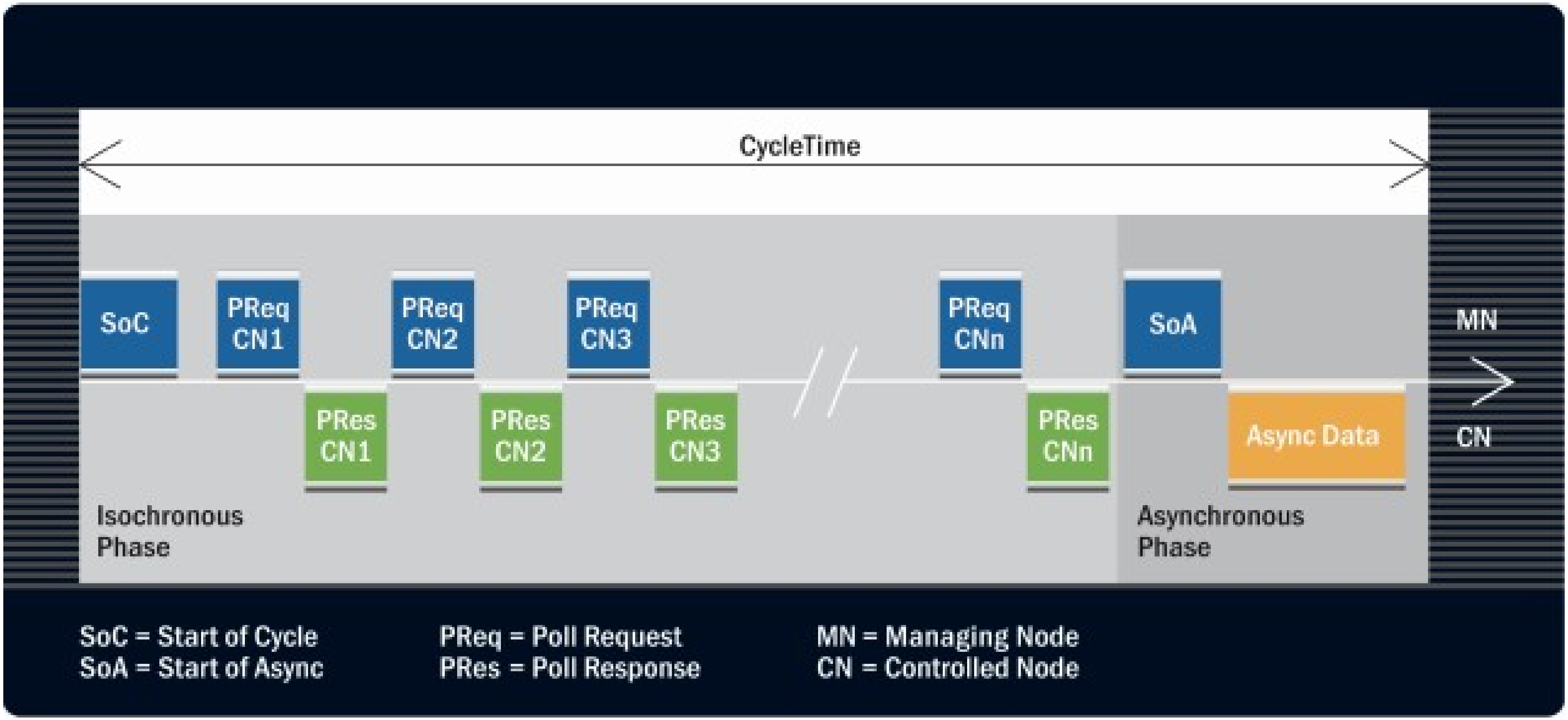
POWERLINK frame



POWERLINK protocol

- One “manager” node (MN) and X “controlled” nodes (CN)
- Cycle divided in 3 steps
  - MN synchronizes CNs with a *SoC* (Start of Cycle) frame which starts “isochronous phase” (RT)
  - CN receives *PReq* (Poll Request) from MN, and replies with *PRes* (Poll Response) and data
  - Last step is “asynchronous phase” (no RT) started with *SoA.* Addressed node should answer *ASnd*
- Standard IP-based protocols and addressing can be used during the asynchronous phase

POWERLINK protocol



SoC = Start of Cycle
SoA = Start of Async

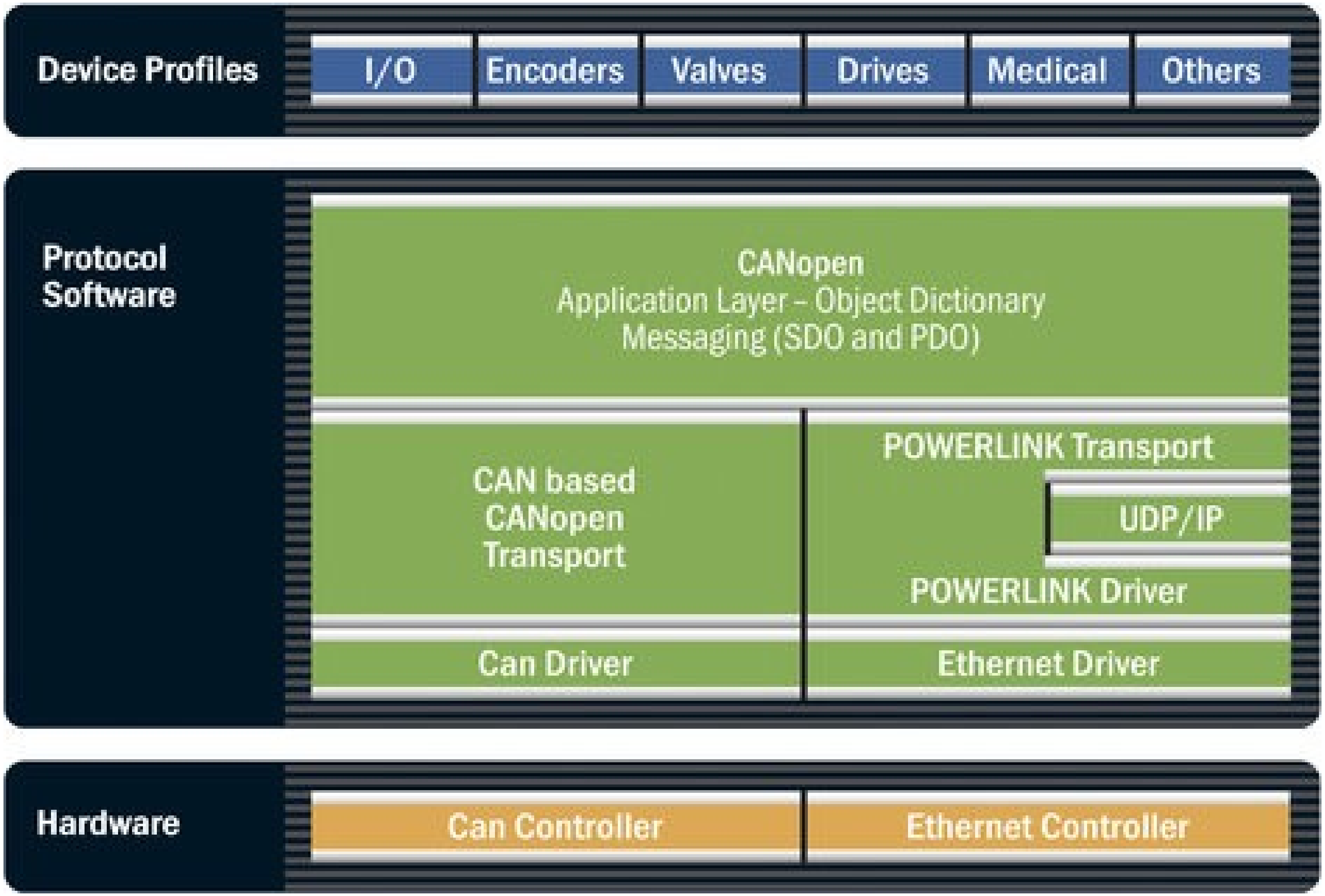
PReq = Poll Request
PRes = Poll Response

MN = Managing Node
CN = Controlled Node

POWERLINK / CANopen

- CANopen is one of the most widely used application protocols today
- Standardized device description files
- POWERLINK defines a CANopen-based Application Layer
- Same device description files as CANopen
- Same object dictionaries and communication mechanisms
  - process data objects (PDO)
  - service data objects (SDO)
  - network management (NMT)
- POWERLINK = “CANopen over Ethernet"

Powerlink / CANopen

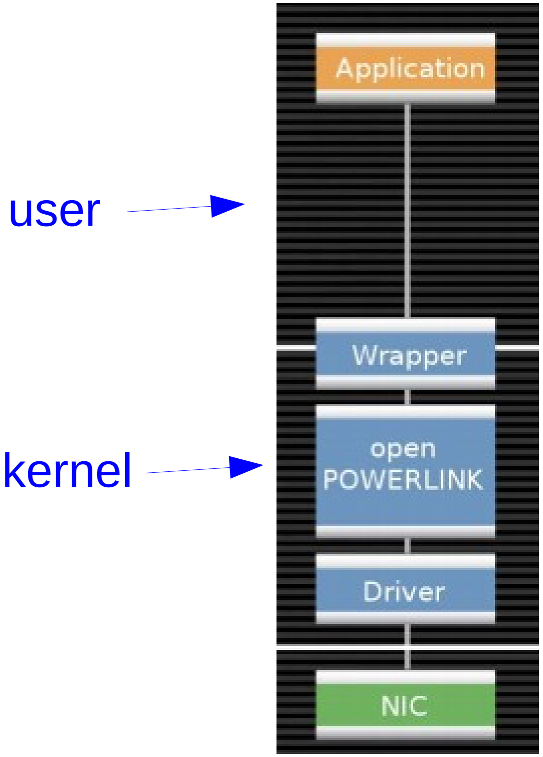


openPOWERLINK

- BSD license
- Support for Linux, Windows, Xilinx/Altera FPGAs
- Official support for x86, ARM (Zynq)
- CMake based → `CMAKE_TOOLCHAIN_FILE` for cross-compilation
- Buildroot packaging (version 1.08.5)
- Building process :
  - Stack
  - Drivers (if necessary)
  - Demo applications MN/CN (console, Qt)

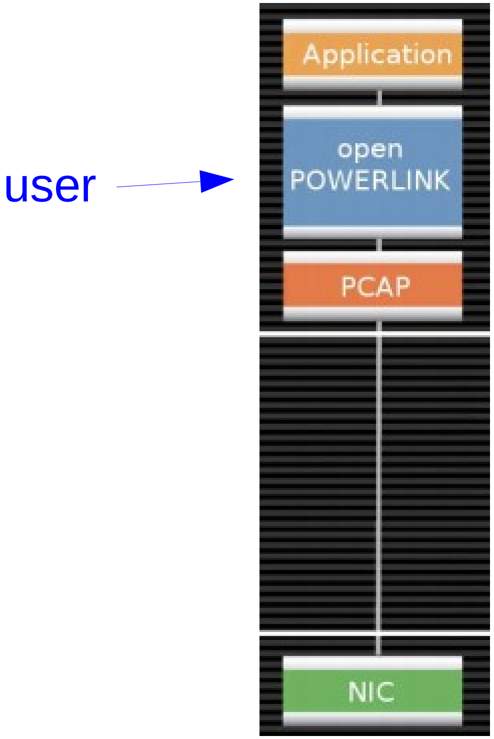
Architecture 1 (kernel)

- Application in user space
- Stack and drivers in kernel space
- High performance and precision
- Specific drivers (*Edrv* for Ethernet drivers)
  - About 10 supported controllers
  - No Linux “mainlining”
- Hard to debug (kernel)



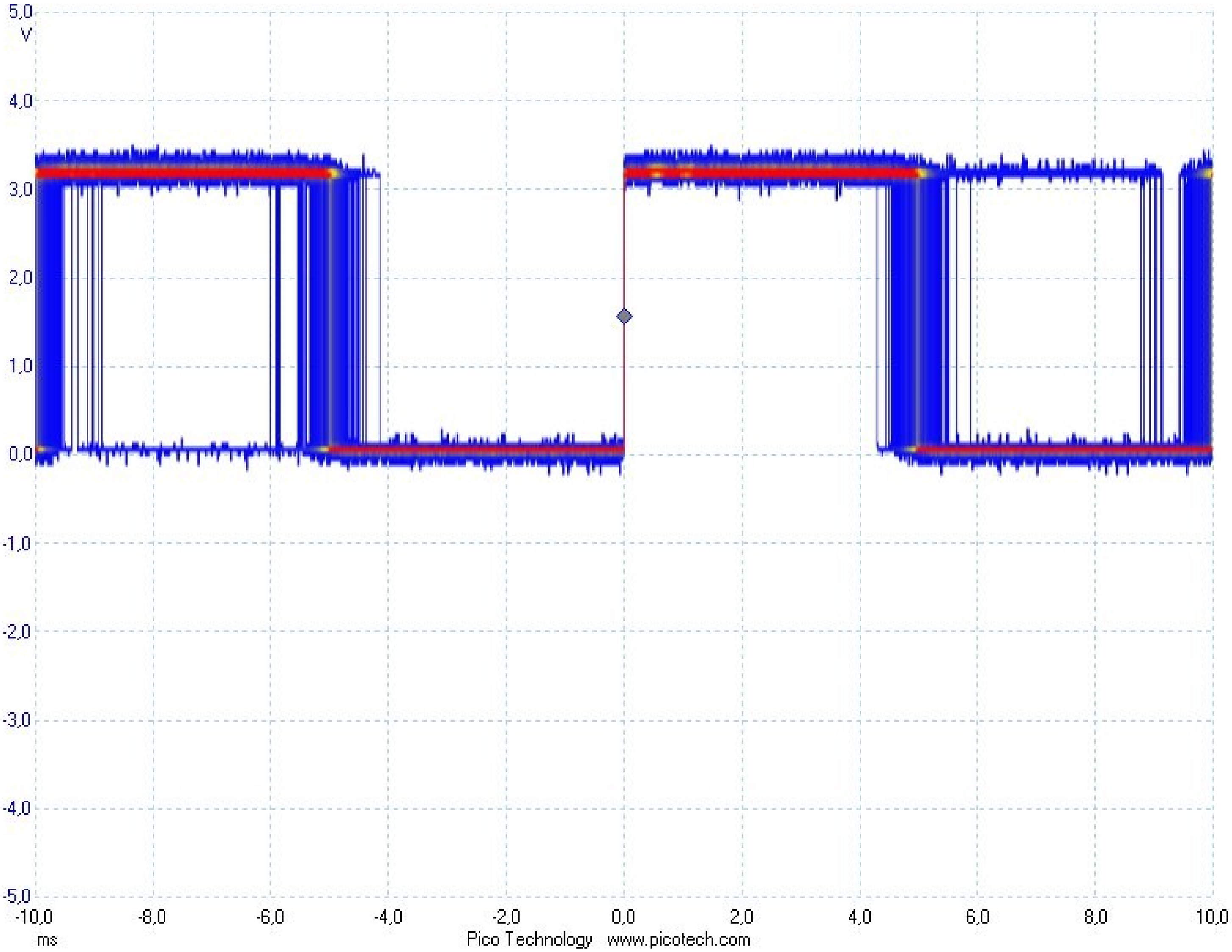
Architecture 2 (user)

- Moving stack to user space
- Using *libpCAP* to talk with standard Linux driver
- Proven solution
- Much easier to debug
- Works with PREEMPT-RT patch
- 100 µs jitter (only 40 µs in kernel)

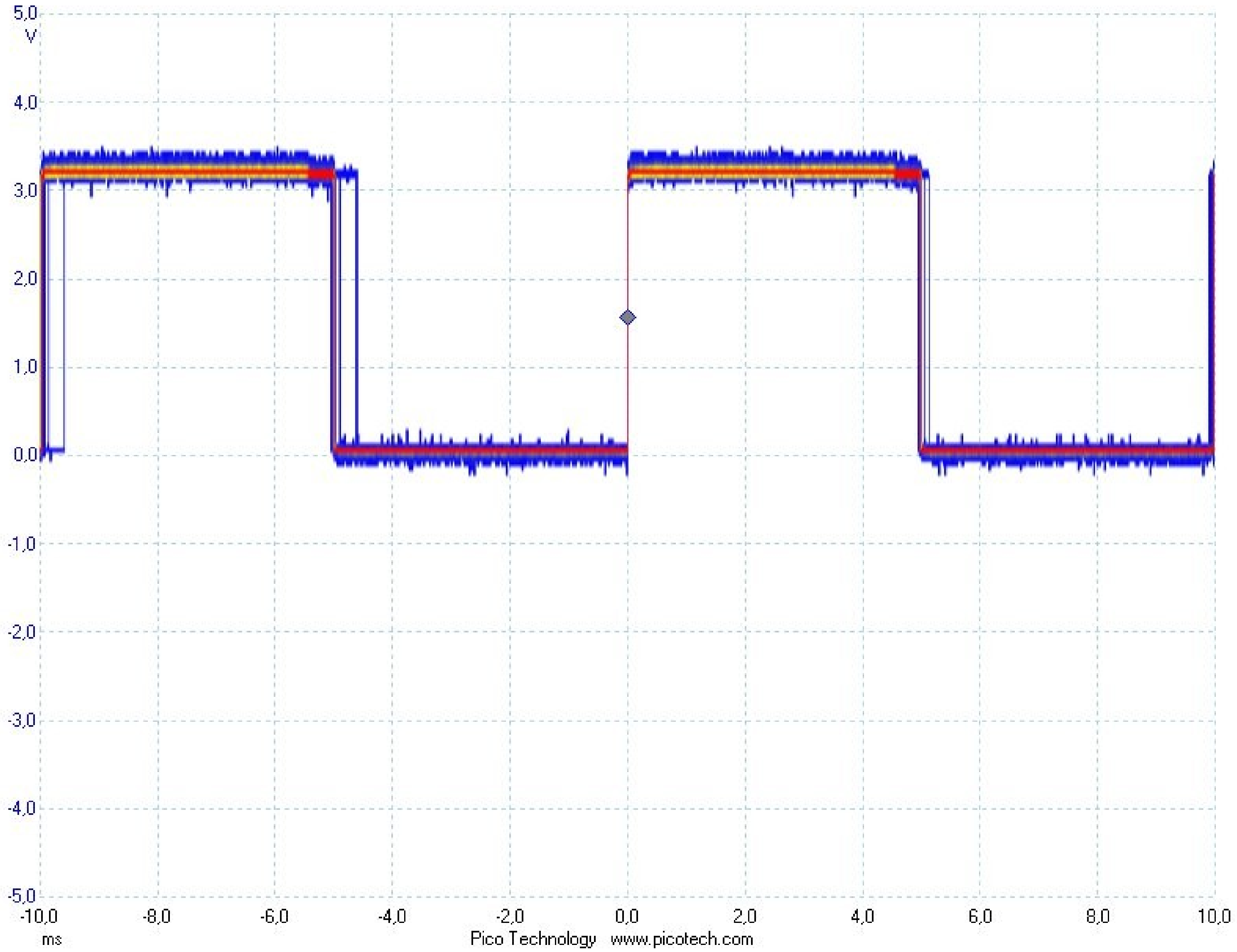


Xenomai

Standard Linux & RT :-(



“Extended” Linux & RT :-)



Linux & RT

- Using Linux as “RTOS” is very interesting
  - POSIX
  - Hybrid approach with some RT tasks
  - Usable as a standard UNIX
- 2 solutions :
  - Upgrading Linux kernel RT performance (PREEMPT-RT)
  - Adding a RT “co-kernel” sharing hardware with Linux (RTLinux, RTAI, Xenomai)

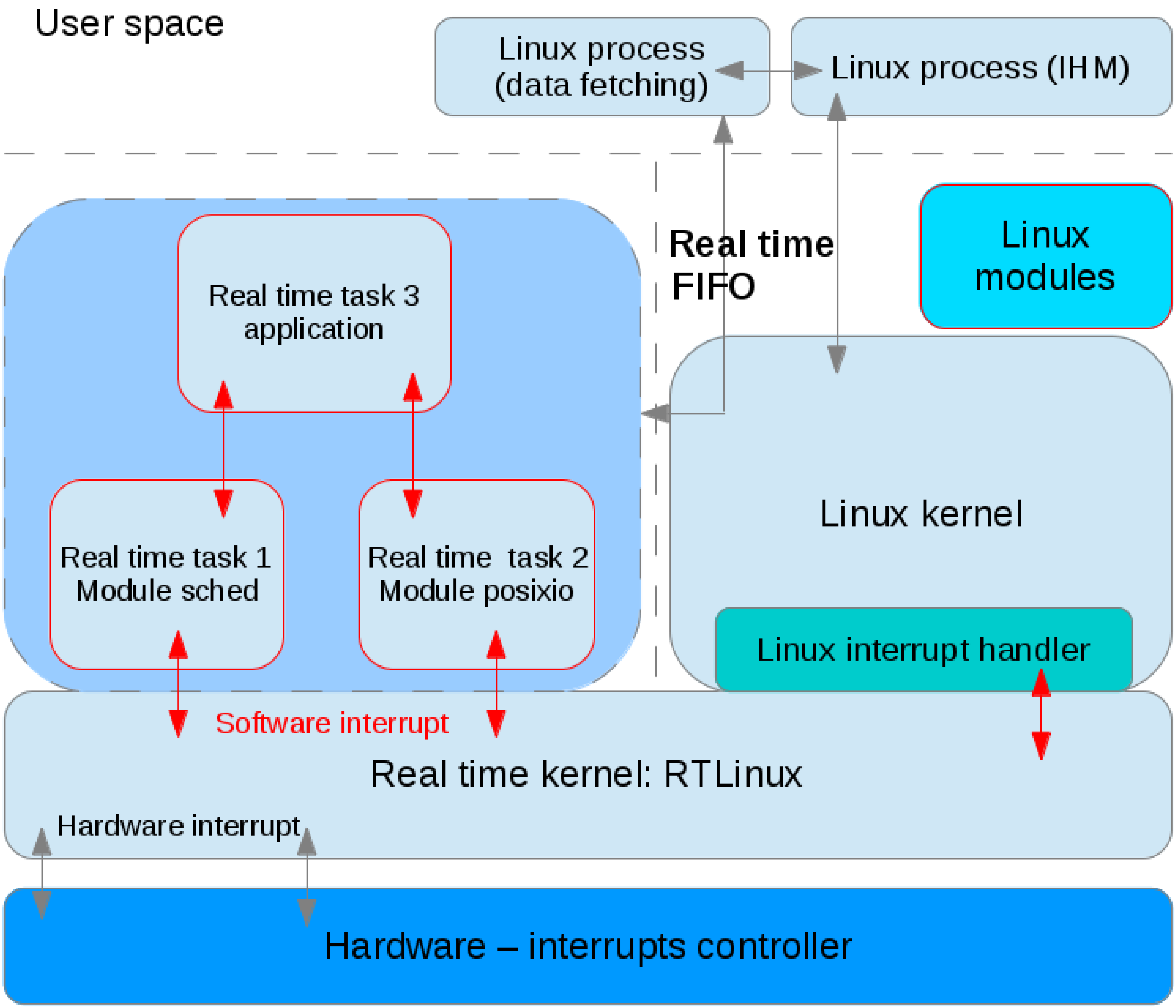
PREEMPT-RT

- Maintained by Thomas Gleixner
- Mostly used on x86 (but runs on recent ARM, Nios2, Microblaze)
- Needs a mainline kernel (or something like)
- Very easy to install (just a kernel patch)
- Same programming APIs as standard kernel (user and kernel space)
- 50 µs jitter (x86/Atom), 150 µs on Raspberry Pi B+
- Currently usable with openPOWERLINK

Co-kernel

- Adding co-kernel for RT tasks
  - RT subsystem inside kernel module(s)
  - Needs kernel patch for hardware resource (IRQ) virtualization
- Main projects
  - Kernel only (RTLinux, 1996) → “dead”
  - Kernel & (partially) user space (RTAI, 1998)
  - Full user space integration (Xenomai, 2001)
- 10 µs jitter on tom/x86, 50 µs on Raspberry Pi B+

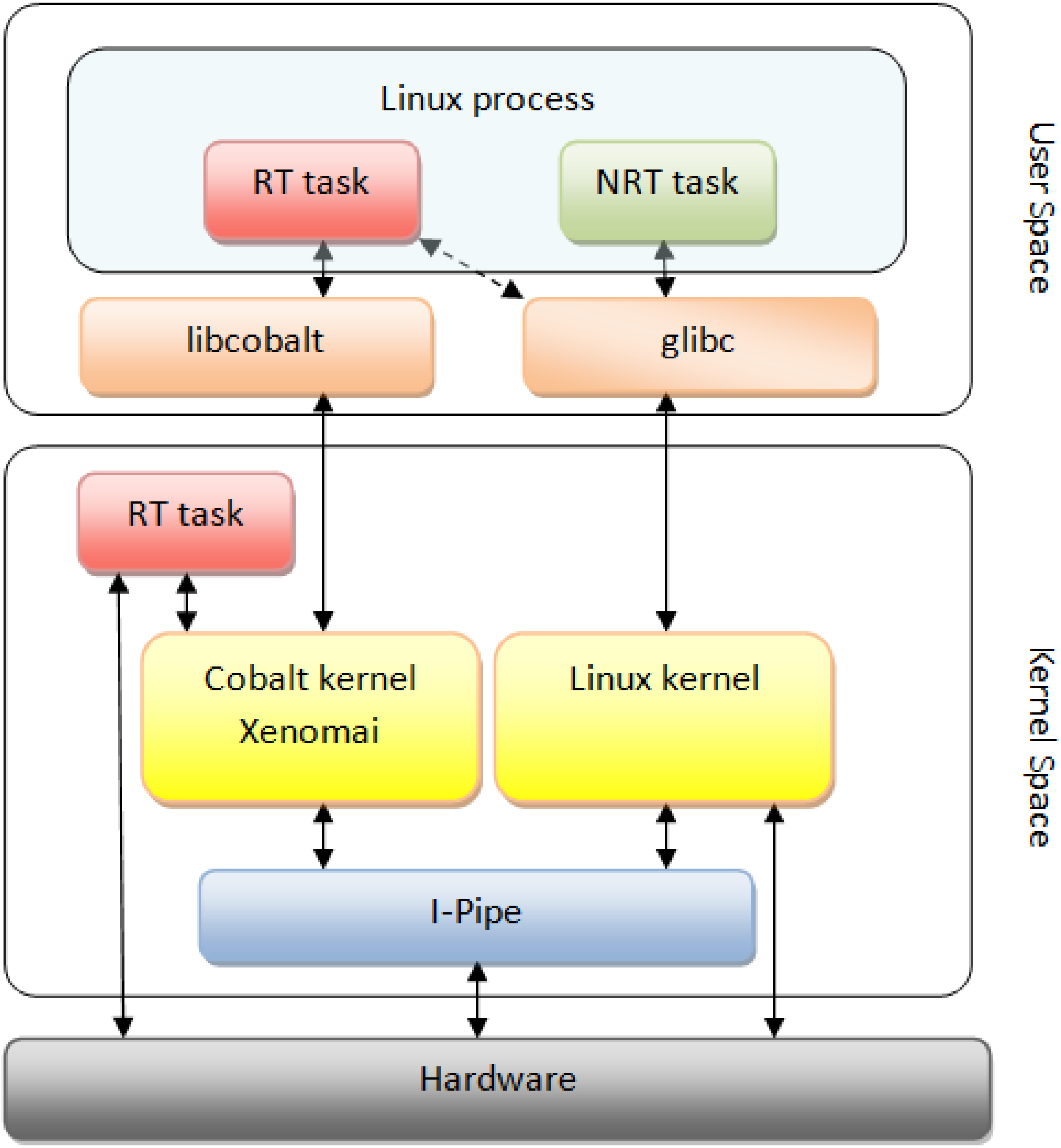
RTLinux architecture (kernel only)


User space
Linux process (data fetching)
Linux process (IHM)
Real time FIFO
Linux modules
Real time task 3 application
Real time task 1 Module sched
Real time task 2 Module posixio
Linux kernel
Linux interrupt handler
Software interrupt
Real time kernel: RTLinux
Hardware interrupt
Hardware – interrupts controller

Xenomai

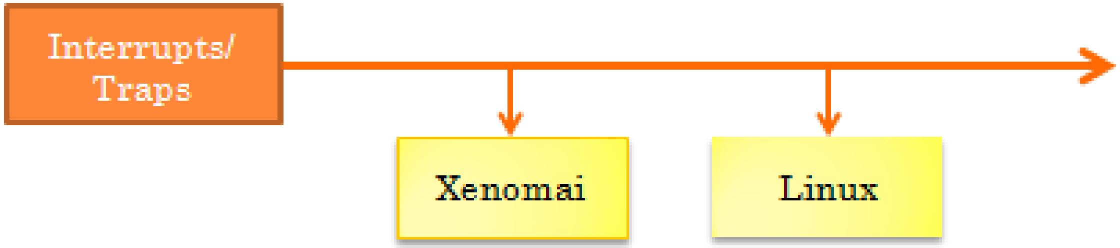
- Maintained by Philippe Gerum
- Xenomai = realtime Linux subsystem
  - RT tasks in user space
  - RT driver API = RTDM for “Real Time Driver Model”
  - RT network stack = RTnet !
- Include “skins” for POSIX, VxWorks, VRTX, uITRON, pSOS, …
- Runs on top of I-pipe/Adeos (Interrupt pipeline)
  - Xenomai domain (RT)
  - Linux domain (No RT)
- v3 can run on top of PREEMPT-RT
- Currently v2.6.4 et v3.0-rc7
- GPL license (kernel), LGPL (user)

Xenomai 3 architecture



I-pipe

- I-pipe = interrupt source for domains (Xenomai, Linux)
- Highest priority to Xenomai (RT)



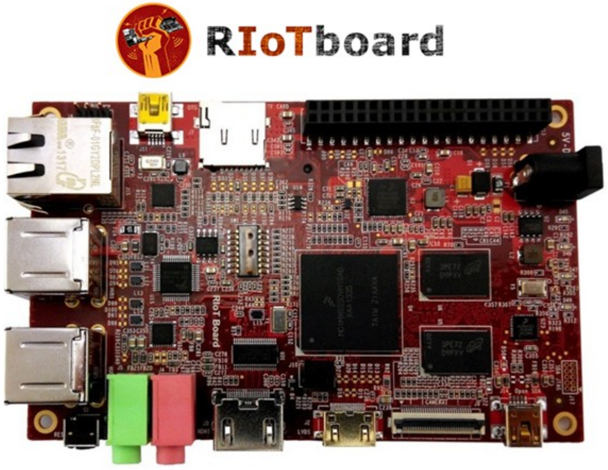
Xenomai in industry

- CANFestival (CANopen stack)
- PEAK System CAN boards drivers
- EtherCAT master
- RT SPI driver (i.MX28)
- BEREMIZ, integrated development environment for machine automation

POWERLINK over Xenomai

POWERLINK and Xenomai

- Started as 6 months internship with Damien Lagneux from ECE Paris
- Currently a “proof of concept” by OWI
- ARM/i.MX6 target (Armadeus APF6, RIOTboard)
- Xenomai is often used by our customers



RTnet

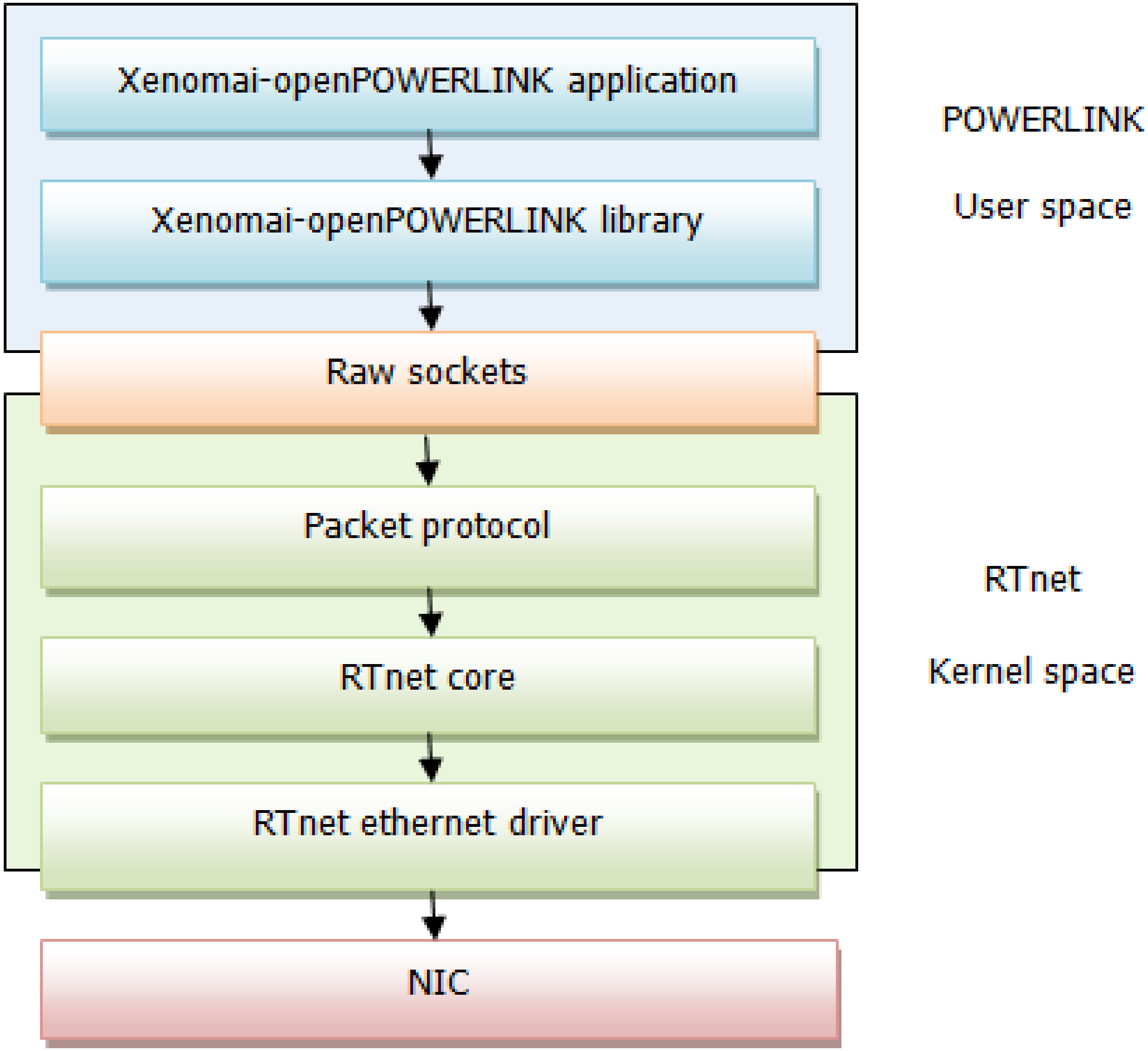
- Xenomai v2 contribution, merged with v3
- Based on RTDM (protocol device)
- Limited hardware support (dedicated driver API)
  - Fec, AT91, AM335x (BB Black)
  - RTL8139, Natsemi, PCnet32, ...
  - MPC8xxx, …
- Example session (BB Black)

```
# insmod rtnet .ko
# insmod rt_smsc.ko
# insmod rt_davinci_mdio.ko
# insmod rt_ticpsw.ko
# insmod rtpacket.ko
# insmod rtipv4.ko
# rtifconfig rteth0 up 192.168.1.1
# rtroute add 192.168.1.2 00:22:15:80:D5:88 dev rteth0
# rtping 192.168.1.2
```

Architecture

- Current openPOWERLINK architecture is based on libPCAP
- libPCAP is based on “packet socket” (Linux)
- Xenomai RTnet stack includes packet socket support (rtpacket module)
- Porting libPCAP to Xenomai is too long for internship
- Hardware is limited by RTnet drivers but just a POC...

Architecture



Architecture

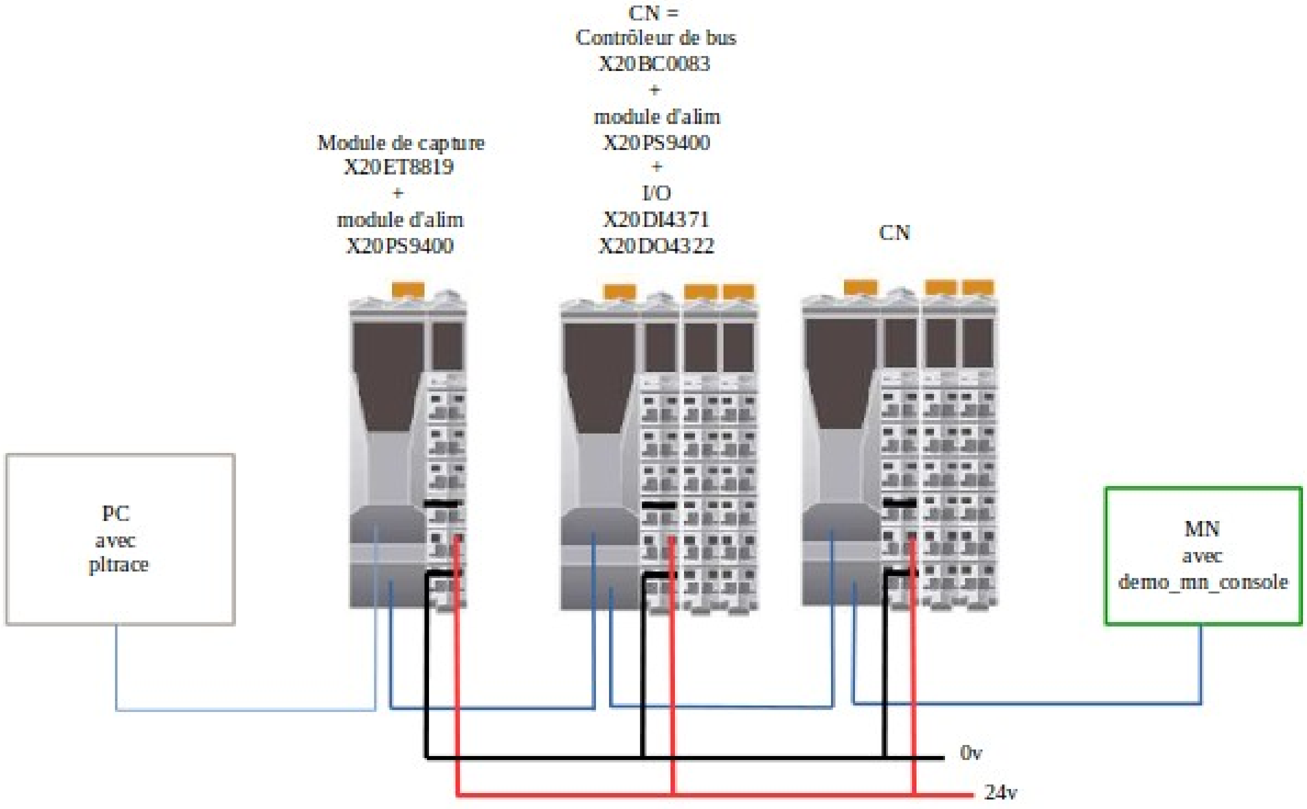
- PCAP layer removed
- Sending / receiving packet (through packet socket) directly from/to the openPOWERLINK stack
- Modified RT network interfaces searching
- RTnet architecture is close to POWERLINK “kernel” architecture

Problems

- Problem 1
  - Only the first SoA frame of the POWERLINK cycle is emitted
  - We have to implement a new packet handler since RTnet stack cannot capture packets it sent
- Problem 2
  - POWERLINK cycle stops due to an unsent PollResponse frame
  - We have increase the Ethernet driver buffer pool

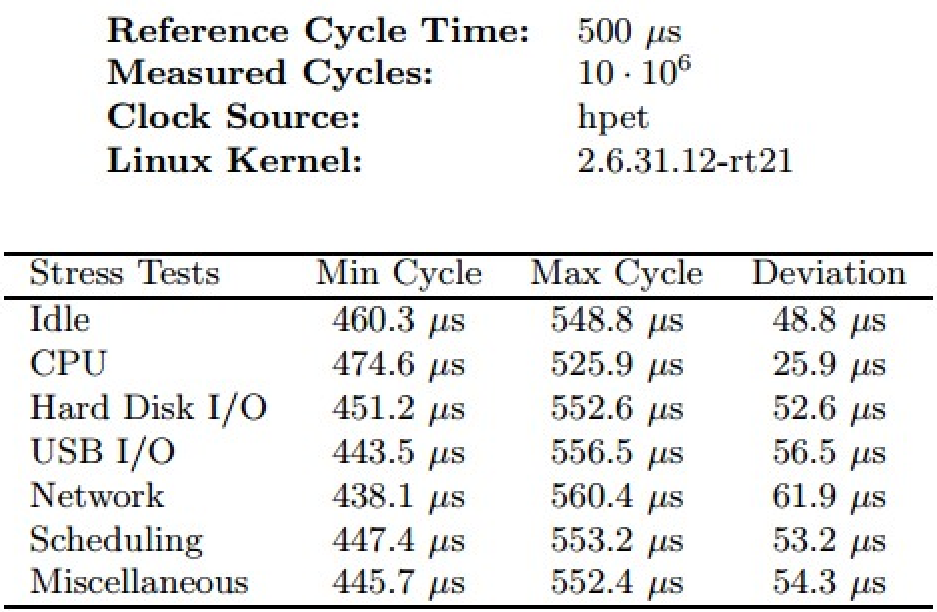
Test configuration

- 1 i.MX6 board as MN
- 2 B&R modules as CN
- 1 B&R capture module (timestamping)
- 1 PC for saving frames



Test and results

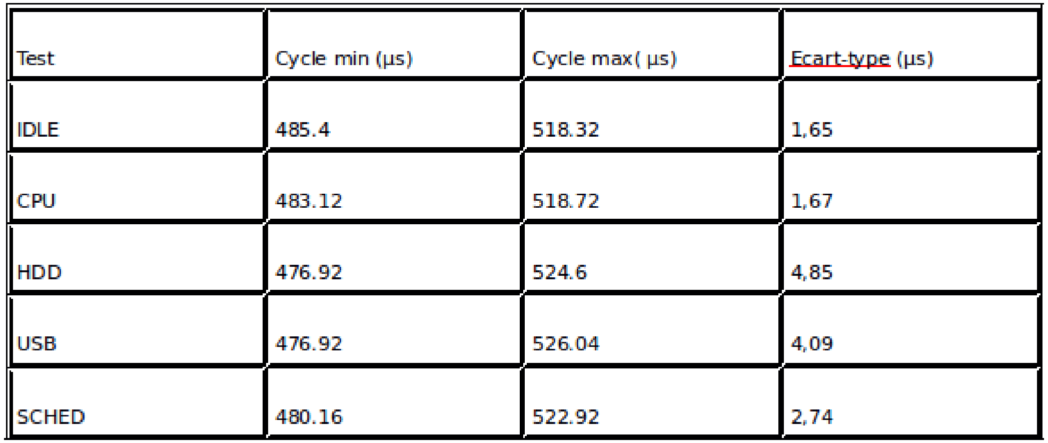
- Xenomai solution is close to Linux “kernel” version of openPOWERLINK (architecture 1)
- Comparison with Baumgartner/Schoenegger paper (B&R)
- Workload with `dd`, `hackbench`, “flood ping”



Reference Cycle Time:	500 μs
Measured Cycles:	$10 \cdot 10^6$
Clock Source:	hpet
Linux Kernel:	2.6.31.12-rt21

Stress Tests	Min Cycle	Max Cycle	Deviation
Idle	460.3 μs	548.8 μs	48.8 μs
CPU	474.6 μs	525.9 μs	25.9 μs
Hard Disk I/O	451.2 μs	552.6 μs	52.6 μs
USB I/O	443.5 μs	556.5 μs	56.5 μs
Network	438.1 μs	560.4 μs	61.9 μs
Scheduling	447.4 μs	553.2 μs	53.2 μs
Miscellaneous	445.7 μs	552.4 μs	54.3 μs

Xenomai (i.MX6, 3.x kernel)



Test	Cycle min (µs)	Cycle max(µs)	Ecart-type (µs)
IDLE	485.4	518.32	1,65
CPU	483.12	518.72	1,67
HDD	476.92	524.6	4,85
USB	476.92	526.04	4,09
SCHED	480.16	522.92	2,74

Conclusion + future work

- Good job as Damien didn't know anything about Xenomai (and POWERLINK) when he arrived !
- Currently not stable enough for industrial use → stack debug and optimization
- Work with EPSG and B&R
  - mainlining in openPOWERLINK project
  - more test with available POWERLINK devices

Bibliography

- http://www.ethernet-powerlink.org
- http://openpowerlink.sourceforge.net/web
- http://www.automationworld.com/networking-amp-connectivity/fieldbus-industrial-ethernet
- https://lwn.net/Articles/572740
- http://www.beremiz.org
- https://www.osadl.org/fileadmin/dam/rtlws/12/Baumgartner.pdf
- https://lwn.net/images/conf/rtlws-2011/proc/Baumgartner.pdf
- “Introduction à RTnet”, P. FICHEUX Open Silicium #15 (french)
- Internship report “openPOWERLINK over Xenomai” D. LAGNEUX (french)
- http://www.armadeus.com/francais/produits-cartes_microprocesseur-apf6.html
- http://www.embest-tech.com/shop/star/riotboard.html