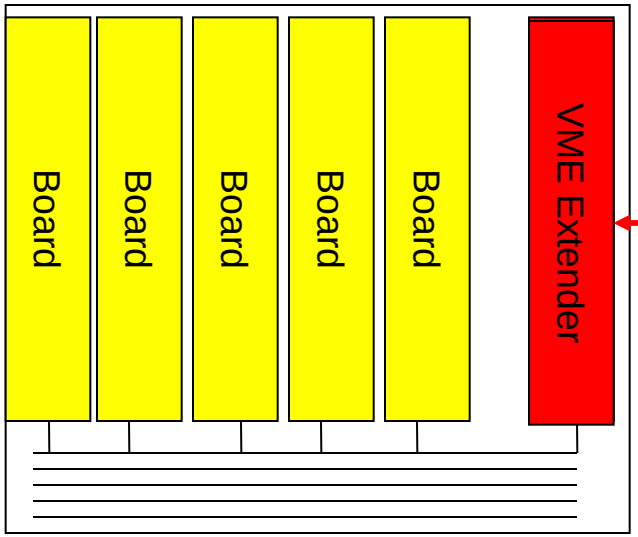
Real-time control using embedded micro-controllers

Flavio Fontanelli, Giuseppe Mini, Mario Sannino, INFN Genoa

Zbigniew Guzik, Institute for Nuclear Research

Richard Jacobsson, Beat Jost, Niko Neufeld CERN, PH

Traditional board-control



Ethernet LAN

(NFS) Server

e.g. USB

VME library

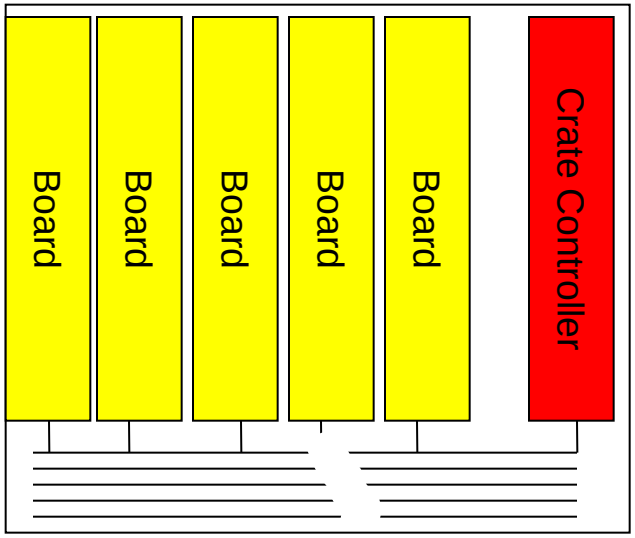
3, 6 or 9U crate with backplane bus (VME, CompactPCI) and Crate Controller

Problems with Crate-based Board Control

Scalability / Throughput

- Each board shares the bus (bandwidth) and resources in Controller
- Need crate-controller + crate for a single board

Robustness

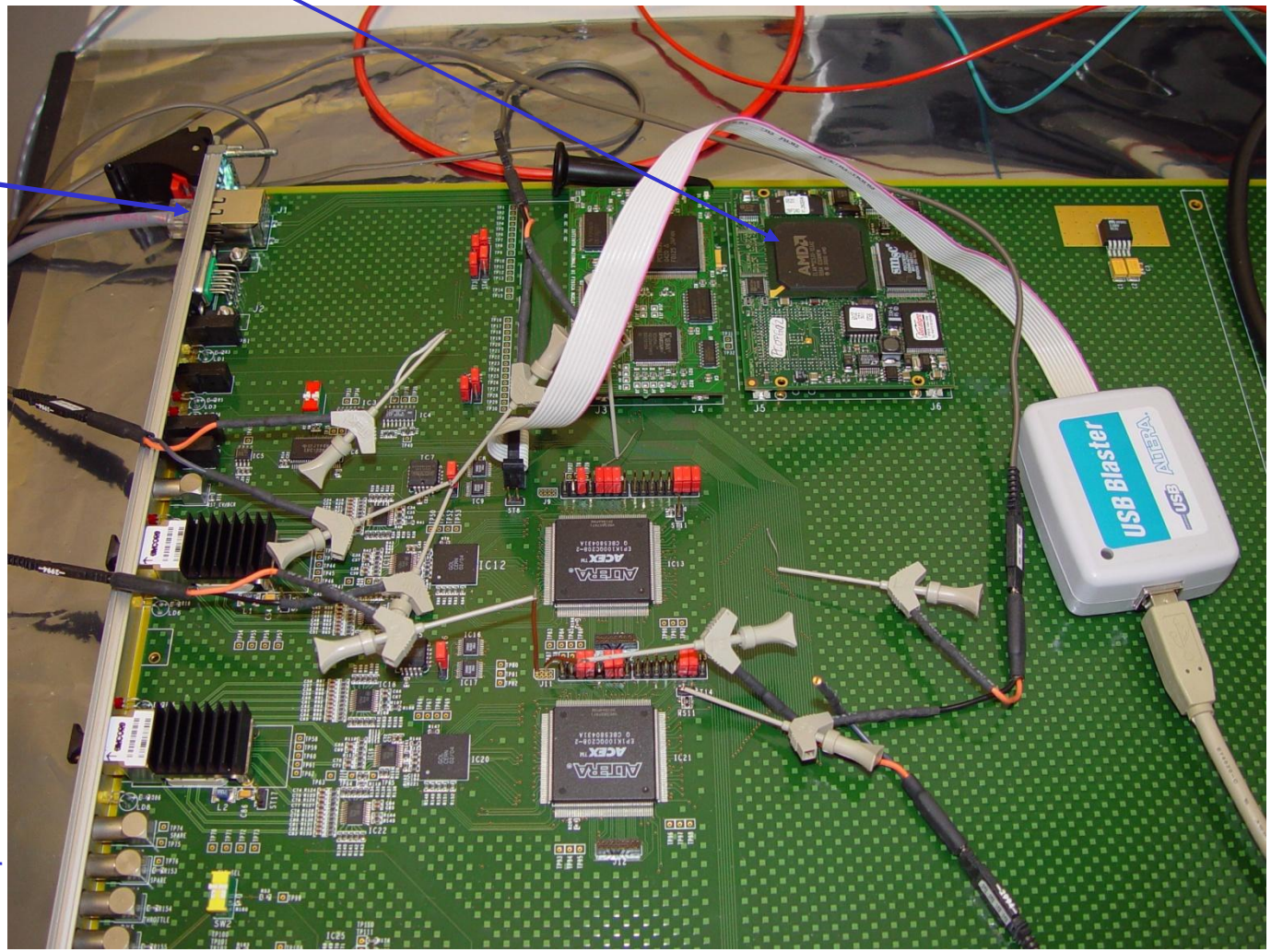


Access to all boards blocked!

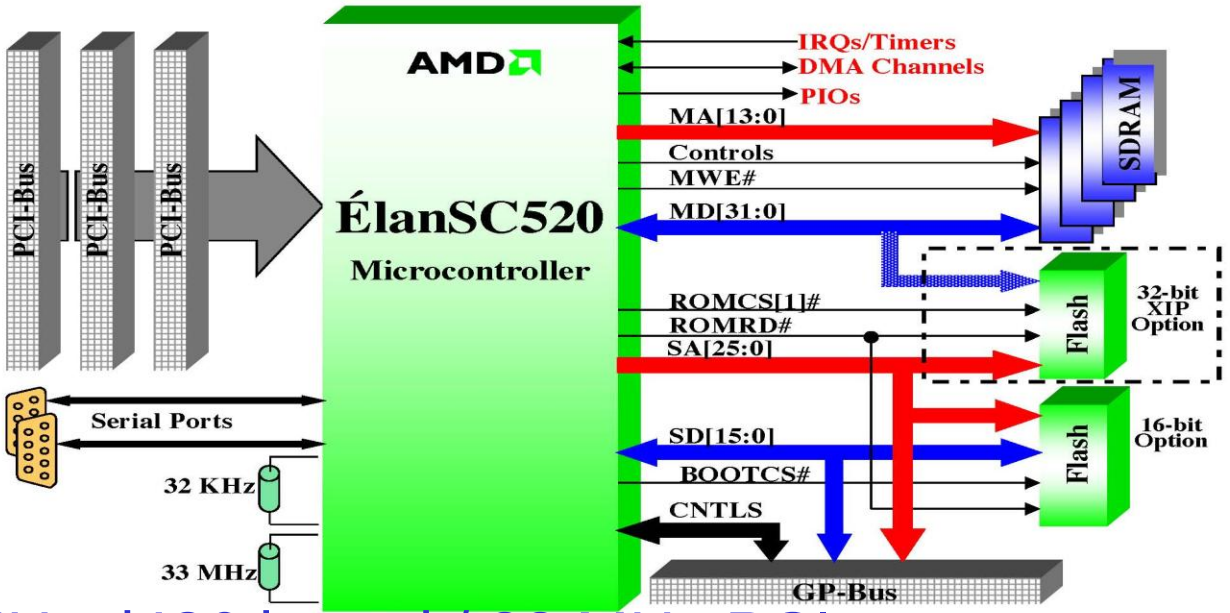
Microcontroller for Board-control

- Accessed via cheap ubiquitous LAN infrastructure (Ethernet)
- Scales well (you pay what you need)
- Nice for table setups *(Debugging!)*

(thanks to M. Mücke (Poster S10-2) for the photo)



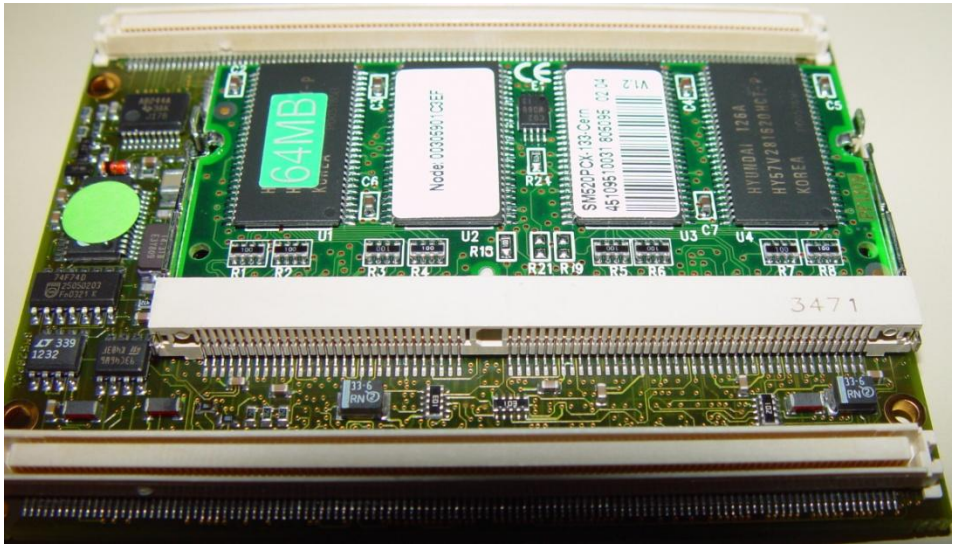
An example: the AMD ELAN 520



- 32-bit, 133 MHz, i486 based / 33 MHz PCI,
- Lots of useful features: hardware watch-dog, high-resolution timers, GPIOs
- Low-power (1.4 W typical, 2 W max), no active cooling necessary
- Enormous software base of i386 (GNU/Linux, MS-Windows)
- Other examples: **P2-3, P2-8**

A Microcontroller is not (yet) a board-controller

- Add a “few” things (Ethernet, USB, Flash) to make an Embedded PC from a Microcontroller
- Use a commercially available device SM520PC from Digital Logic AG Switzerland, http://www.digitallogic.ch

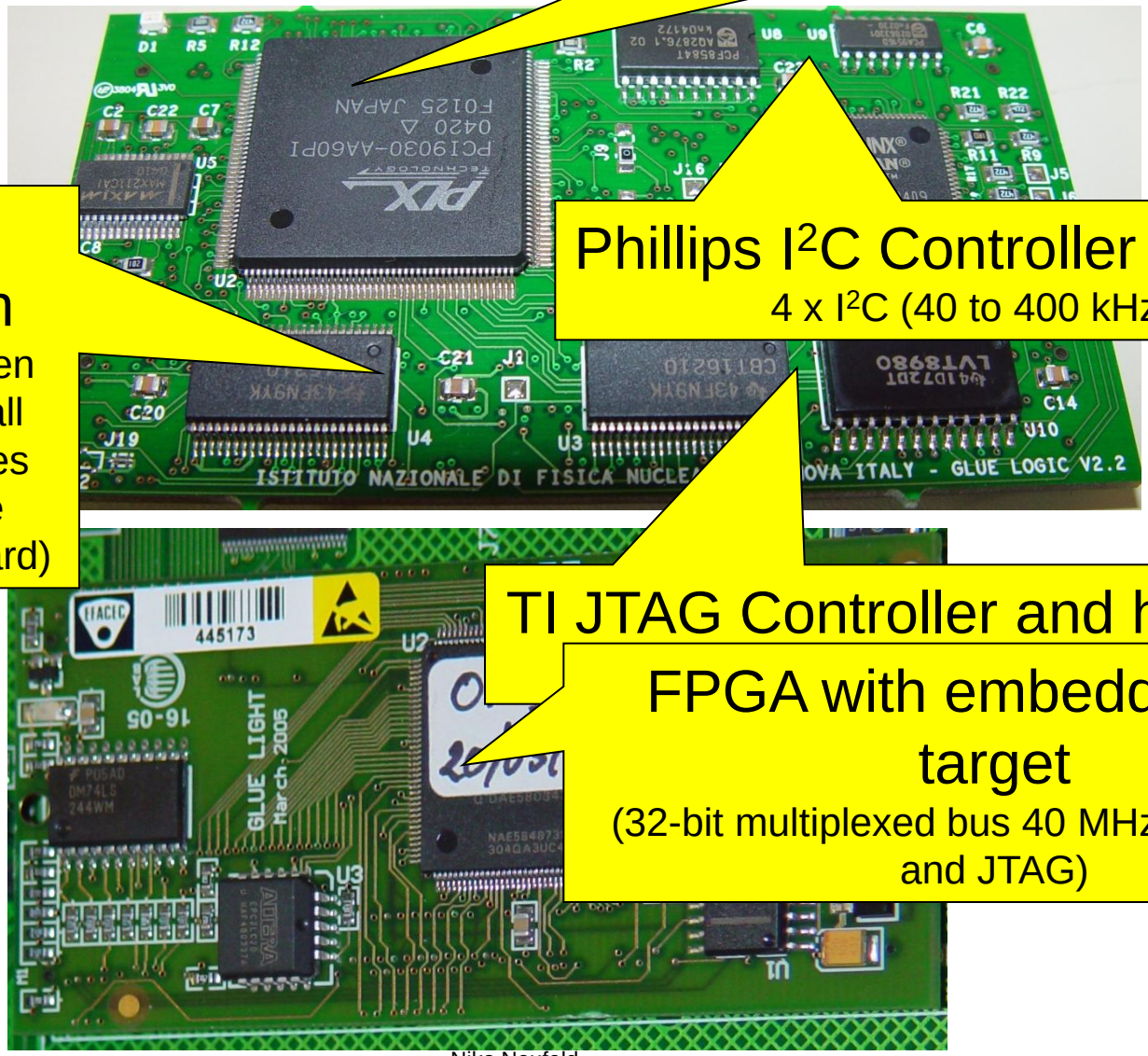


An embedded PC is still not a board-controller

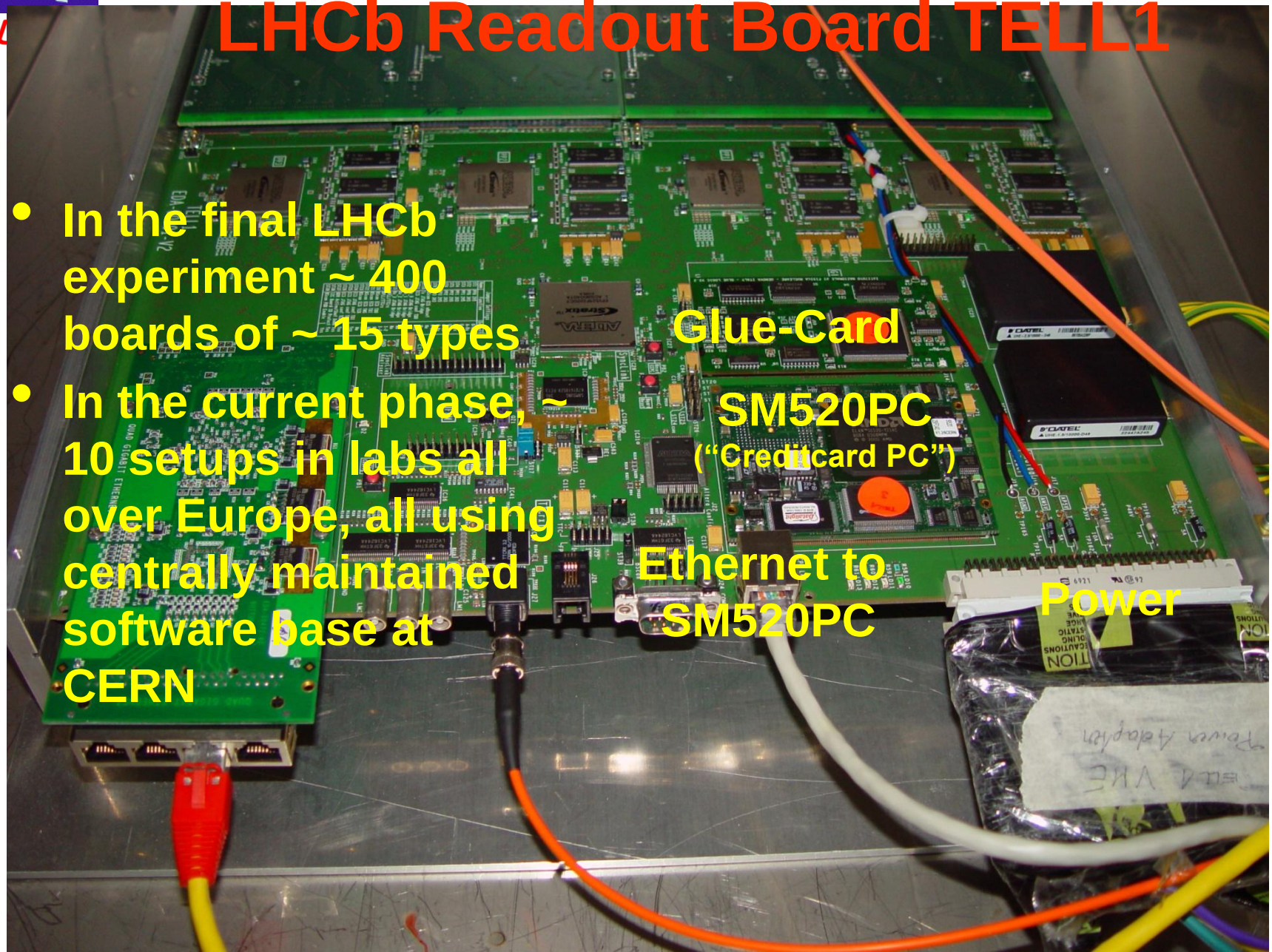
- Traditional PC interfaces: PCI, ISA, parallel port, USB
- For controlling, configuring and monitoring FPGAs and ASICS need rather I^2C, JTAG and a high-speed, “simple”, long distance parallel bus
- Need some small adapter or “glue-”logic

Glue-logic fo[illegible]s

- ASIC Based
- FPGA Based

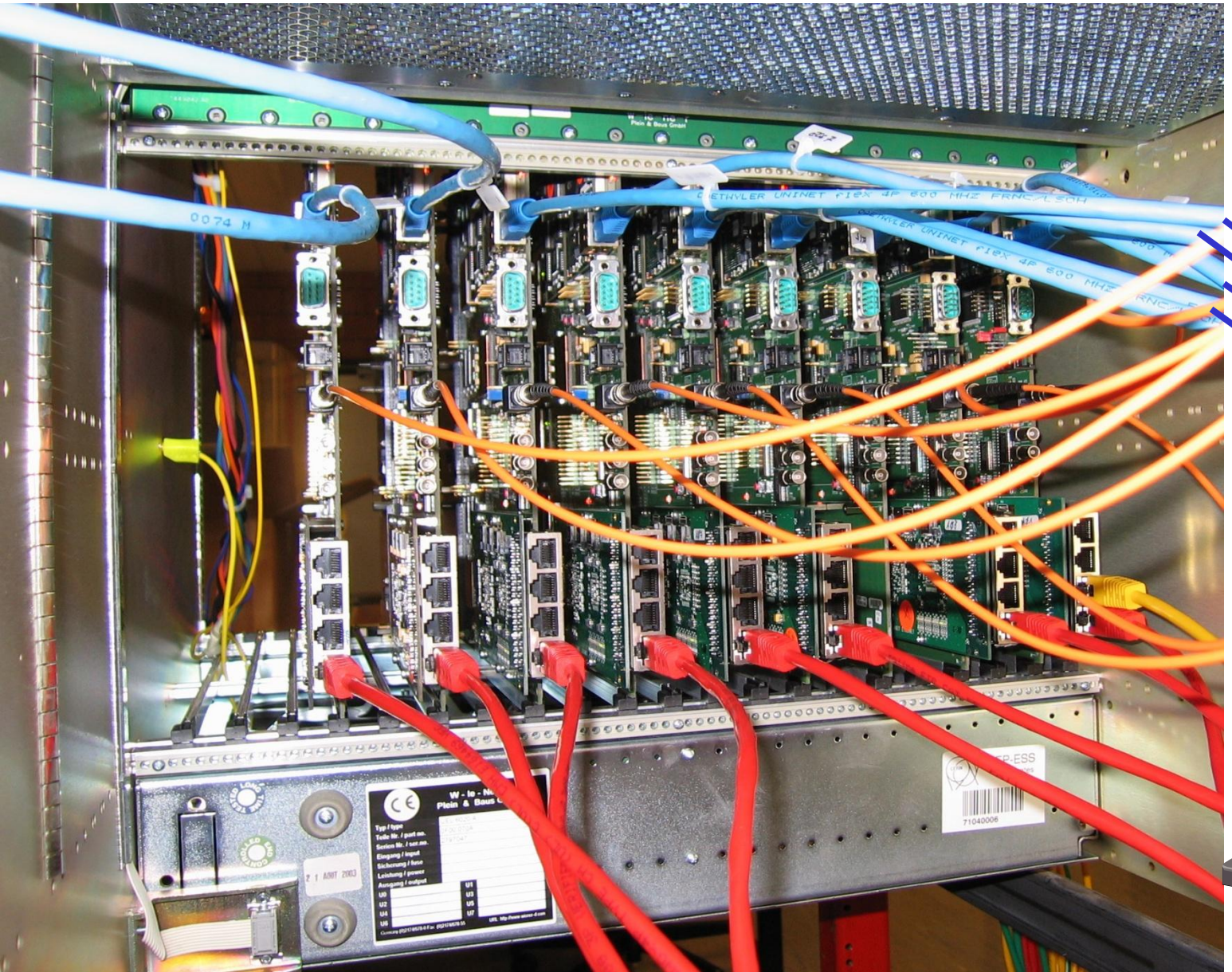


Full System in Action: the LHCb Readout Board TELL1



- In the final LHCb experiment ~ 400 boards of ~ 15 types
- In the current phase, ~ 10 setups in labs all over Europe, all using centrally maintained software base at CERN

A complete crate



One PC
for ~ 20 to 40 boards acts as Boot and NFS server

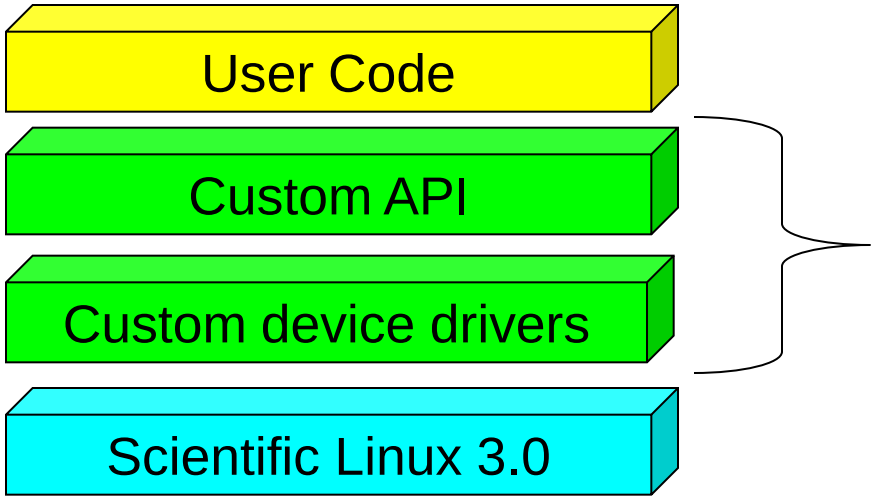
Ethernet Switch

Capabilities and Performances

- 4 I2C, 3 JTAG, PLX 9030 bus, 8 GPIOs, busswitch to isolate all signal lines of the SM520PC/Gluecard (manually or automatically when reset-line of SM520PC is pulled low), attached via 100 MBit Ethernet
- Running Linux and standard software we achieve: > 20 MB/s throughput on 9030 bus (read) from memory and FPGA registers, > 1.5 MBit/s on JTAG
- Program configuration devices and FPGAs via JTAG
- 64 MB Ram, 2 MB Flash, 133 MHz CPU (66 “Bogomips”), can do quite some processing

And what about software?

- We use Scientific Linux, a freely available Linux distribution made by Fermilab and CERN, fully compatible with Redhat Enterprise Linux (https://www.scientificlinux.org)
- We provide a software package for automatic setup of the boot and NFS server for the embedded PCs a
- User applications can developed/compiled on any (Linux) PC. Practically any available software (java, apache, X11 etc…) can be used together with a common set of custom libraries to access the board-resources



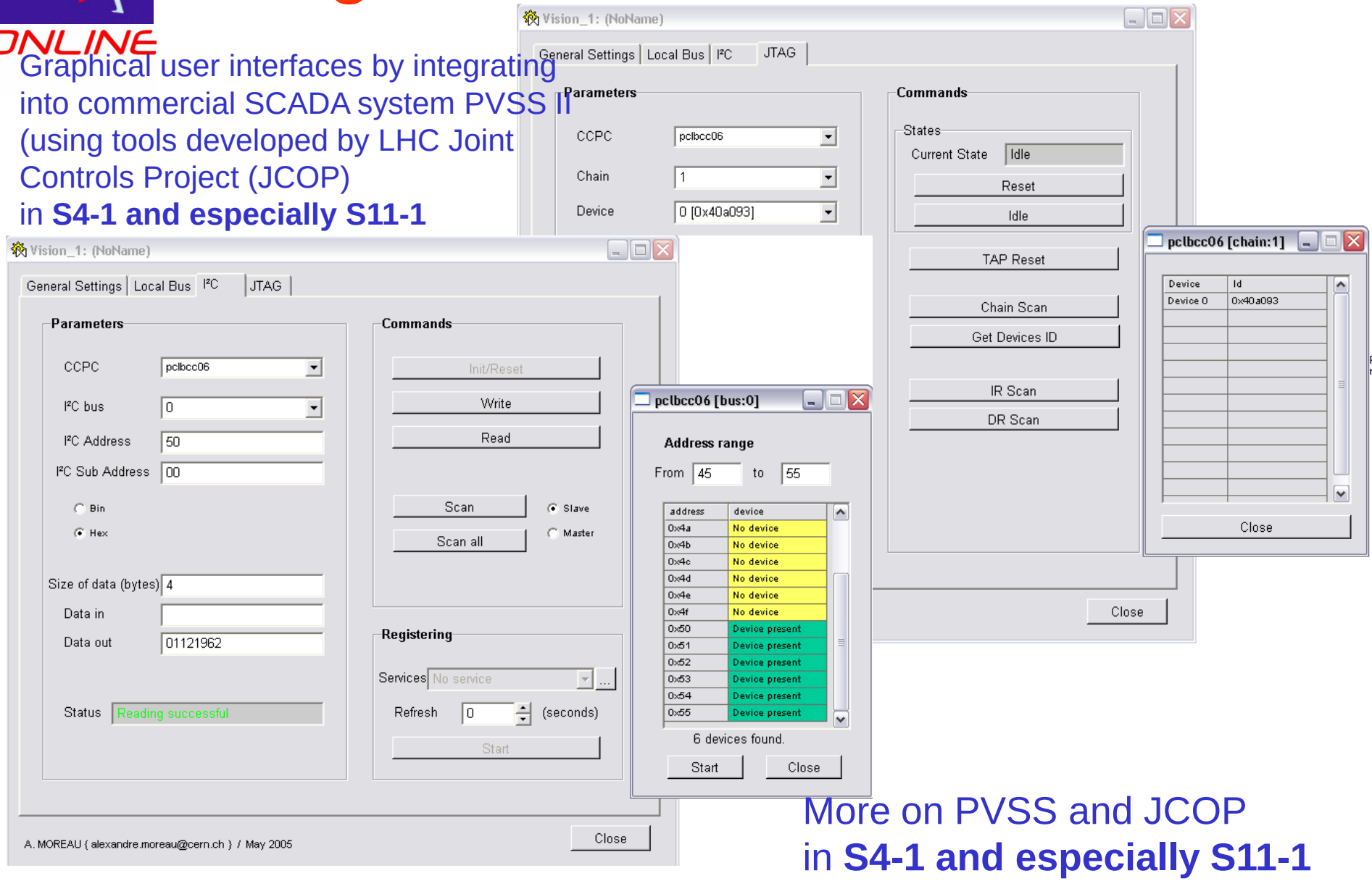
~ 15 Packages to access all resources on board

Central web-repository Automatic updates in all collaborating labs

Special small footprint kernel diskless boot

High Level Software: SCADA

Graphical user interfaces by integrating into commercial SCADA system PVSS II (using tools developed by LHC Joint Controls Project (JCOP)
in **S4-1 and especially S11-1**



More on PVSS and JCOP
in **S4-1 and especially S11-1**

These screenshots thanks to Alexandre Moreau, ISIMA

A word about “Real-time”

- Do you need it for board-control? Really?
- Standard Linux has no hard real-time support (i.e. guaranteed maximum time to handle an interrupt/syscall). It has however support for soft or maybe better quasi real-time (i.e. processes can be strictly prioritised)
- In LHCb we have not (yet) found an application for even soft real-time. Polling critical registers if necessary gives a very good response time.
- There exist of course a host of true real-time OS or Linux variants (RTLinux), which would run fine on the Elan520

Lessons / What we could do better

- Ideally fully integrated device “surrounded” by FPGA
- Exists: see **P2-3, P2-8** (but not PC/i386 compatible)
- i386 and Linux are a good thing (PPC and Linux are probably also good)
- and for free (total software effort probably o1.5 man-years)
- Central software repository and automated distribution of packages allows keeping disconnected sites up-to-date easing (remote) trouble-shooting
- Using commercial hardware is great (Microcontroller change) but need still to verify *everything…*