

Embedded Controllers for Local Board-Control

Flavio Fontanelli, Giuseppe Mini, Mario Sannino, Zbigniew Guzik, Richard Jacobsson, Beat Jost, and Niko Neufeld

Abstract—The LHCb experiment at CERN has a large number of custom electronics boards performing high-speed data-processing. As in any large experiment the control and monitoring of these crate-mounted boards must be integrated into the overall control-system. Traditionally this has been done by using buses on the back-plane of the crates, such as VME. LHCb has chosen to equip every board with an embedded micro-controller and connecting them in a large Local Area Network. The intelligence of these devices allows complex (soft) real-time control and monitoring, required for modern field programmable gate array (FPGA) driven electronics. Moreover, each board has its own, isolated control access path, which increases the robustness of the entire system. The system is now in pre-production at several sites and will go into full production during next year. The hardware and software will be discussed and experience from the R&D and pre-production will be reviewed, with an emphasis on the advantages and difficulties of this approach to board-control.

Index Terms—Control electronics, embedded processors.

I. INTRODUCTION

TRADITIONAL board-control uses shared buses like VME or Compact PCI to control custom or commercial electronics-boards. This approach has proven to work well, however, it has also shown to suffer from at least two drawbacks.

- Intelligence is centralised on the crate-controller, which, depending on the task of the controlling entity, this can be a bottle-neck.
- The control-paths of a comparatively large number of boards are shared and linked together. A misconfigured or badly behaving device can block the entire chain.

LHCb has therefore chosen a different approach and equips every electronics board with a micro-controller, which is accessed via a dedicated Local Area Network (LAN). In the final experiment, LHCb will have some 400 boards of 15 different types, with ≈ 100 registers to monitor continuously. In addition, there are large memories of several 100 megabytes on the boards for look-up tables and similar functions, which need to be pre-loaded in a short time interval.

After a brief overview of micro-controllers, the embedded PC which has been chosen for LHCb, is presented. The next section discusses the hardware of the embedded PC and how board-control is achieved. In section 4 the software framework, which has been developed to operate a large number of micro-controllers in a distributed system is described. Finally, in the

conclusions, the advantages and drawbacks of this approach in the light of our five years of experience with this system which is now widely used in pre-production environments are reviewed.

II. MICRO-CONTROLLERS

Micro-controllers are basically microprocessor systems on a chip (SOC), and consist of a microprocessor and all, or most, associated electronics integrated on a single chip. To create a full computer usually only a few discrete components (connectors) must be added, and, if needed in larger amounts, memory, which becomes considerably cheaper when standard DRAM chips are used as opposed to built-in, on-chip memory.

Micro-controllers are ubiquitous in modern electronics, particularly in consumer electronics such as cell-phones, MP3 players and many other devices. They are based on a wide variety of CPU architectures such as PowerPC, ARM and also i386 (Intel) compatible.

The wide availability of software for the i386 architecture resulted in LHCb using it.¹

Rather than building our own micro-controller a complete embedded PC with a standardised interface, the SM520PCX from Digitallogic [1] was used. It is based on the AMD ELAN 520 micro-controller [2] and comprises all necessary hardware on a plugin board of $85 \times 66 \text{ mm}^2$, shown in Fig. 1. The ELAN520 is a i486 compatible processor running at up to 133 MHz, with all the standard functionality of a PC. In addition extra precise hardware timers, a hardware watch-dog for automatic reboot and general purpose input/output (GPIO) lines are available. The SM520PCX is only one of a family of pin-compatible embedded PCs from Digitallogic.

III. INTERFACING TO CUSTOM ELECTRONICS BOARDS

A PC is not “naturally” suited to directly control resources on an electronics board. Most custom chips in LHCb are accessed via the I²C or JTAG (IEEE 1149) interfaces. The latter is also widely used for in-situ programming (ISP) of configuration devices such as EEPROMs and FlashRAMs. For high-speed data-transfer and register access to FPGAs or memories these two methods are unsuitable and a parallel bus is needed. PCs offer two choices, the aged ISA bus and the more modern PCI. ISA is unpopular with electronics engineers, for a number of reasons, not the least of which being it is a non-multiplexed bus.

On the other hand PCI, due to its signaling, which relies on reflective coupling, has impractical trace length restrictions given LHCb’s electronics boards are typically $9U \times 400 \text{ mm}$. A

¹The LHCb experiment will be operating well into the next decade. The operating system and application software of the microcontroller will need changes and maintenance, which is greatly eased when using an architecture, which is certain to remain in widespread use for the foreseeable future.

Manuscript received June 19, 2005; revised March 30, 2006.
F. Fontanelli, G. Mini, and M. Sannino are with INFN Genoa, Genoa, Italy.
Z. Guzik is with the Institute for Nuclear Research, Warsaw, Poland.
R. Jacobsson, B. Jost, and N. Neufeld are with CERN, CH-1211 Geneva 23, Switzerland (e-mail: niko.neufeld@cern.ch).

Digital Object Identifier 10.1109/TNS.2006.874801



Fig. 1. Top-side of the SM520PCX embedded PC. The micro-controller together with some external components like the FLASH-RAM for the BIOS can be seen. A standard SO-DIMM memory-module, normally used in portable computers, is on the back.

simple local,² parallel bus such as is generated by a PLX PCI bridge [3] has therefore been chosen.

Neither of these can be found on PC based micro-controllers. On other micro-controllers, there are usually only one or two I²C chains and only few and/or slow JTAG chains, while ISP in an environment where frequent reprogramming is necessary, definitely requires a high-speed JTAG chain.

A small glue-board has been created with the required interfaces connected via PCI. Two approaches for the glue-card have been followed by two groups in LHCb - an ASIC based and a purely FPGA based one.

In Fig. 2 the ASIC-based glue-card is shown. It is built-around a PLX PCI9030 PCI bridge, JTAG controllers from Texas Instruments and I²C controllers from Philips. It also includes a bus-switch to automatically electrically isolate the glue-card from the carrier-board, when the PC is rebooted. This board provides 4 I²C chains of 40 or 400 kHz and 3 JTAG chains of up to 2 MHz.³

The same functionality can also be implemented in an FPGA, which in this case acts directly as the PCI target and internally generates the I²C and JTAG required. This board is shown in Fig. 3.

When reading from registers or memories attached to the local bus from PCI, transfer performances better than 20 MB/s were achieved.⁴ This exceeds the bandwidth of a 100 Mbit/s

Ethernet connection, and is likely to be more than a 133 MHz processor can handle.

IV. SOFTWARE FOR BUILDING A CONTROL-SYSTEM

The micro-controller and glue-card solution presented above will be used to control, configure and monitor over 400 boards of approximately 15 different types. The R&D for these boards is being done in many laboratories across Europe. We have therefore tried to provide a simple generic software distribution mechanism, which allows small, easy installations for a few boards, as well as larger multi-server setups. The system guarantees that updates of software are centrally transmitted to all installations, so that once software problems have been corrected, the necessary patches are automatically applied at all sites.

A. Embedded Software

Linux has been chosen as the operating system (OS) of the micro-controller. The main reason for this is the easy customisability of the kernel for embedded, disk-less operation, in some R&D projects however Windows has also been used allowing using commercial software packages. All the required functionality has subsequently been ported to Linux.

On top of a slightly customized kernel we run the standard CERN Linux distribution, which is derived from the joint Fermilab and CERN Scientific Linux project [4].

The custom software, drivers and user-libraries to access the resources on the system are packaged and distributed in the same manner as the base-system, using the standard Redhat package

²Local here means not going out of the board.

³We have bench-marked the programming time of a 16-MBit ALTERA EPC16 programming device to be better than 10 s.

⁴For FPGA registers this obviously depends of course strongly on the implementation on the carrier board.

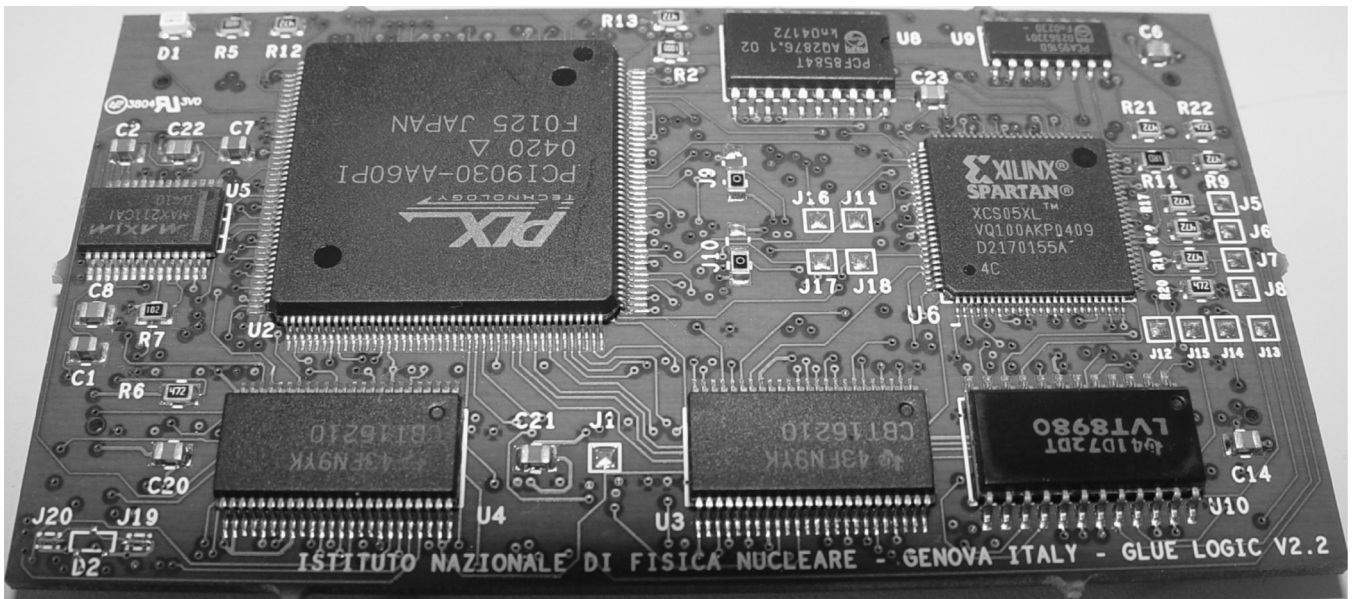


Fig. 2. ASIC based glue-card. In the top left corner the PLX PCI bridge is visible. It creates the local bus, is interfaced to the tiny Xilinx FPGA on the right, which in turn controls the I2C and JTAG hubs in the lower part of the card. In the bottom-left part a level-adapter for the serial line has been added for standard console access.

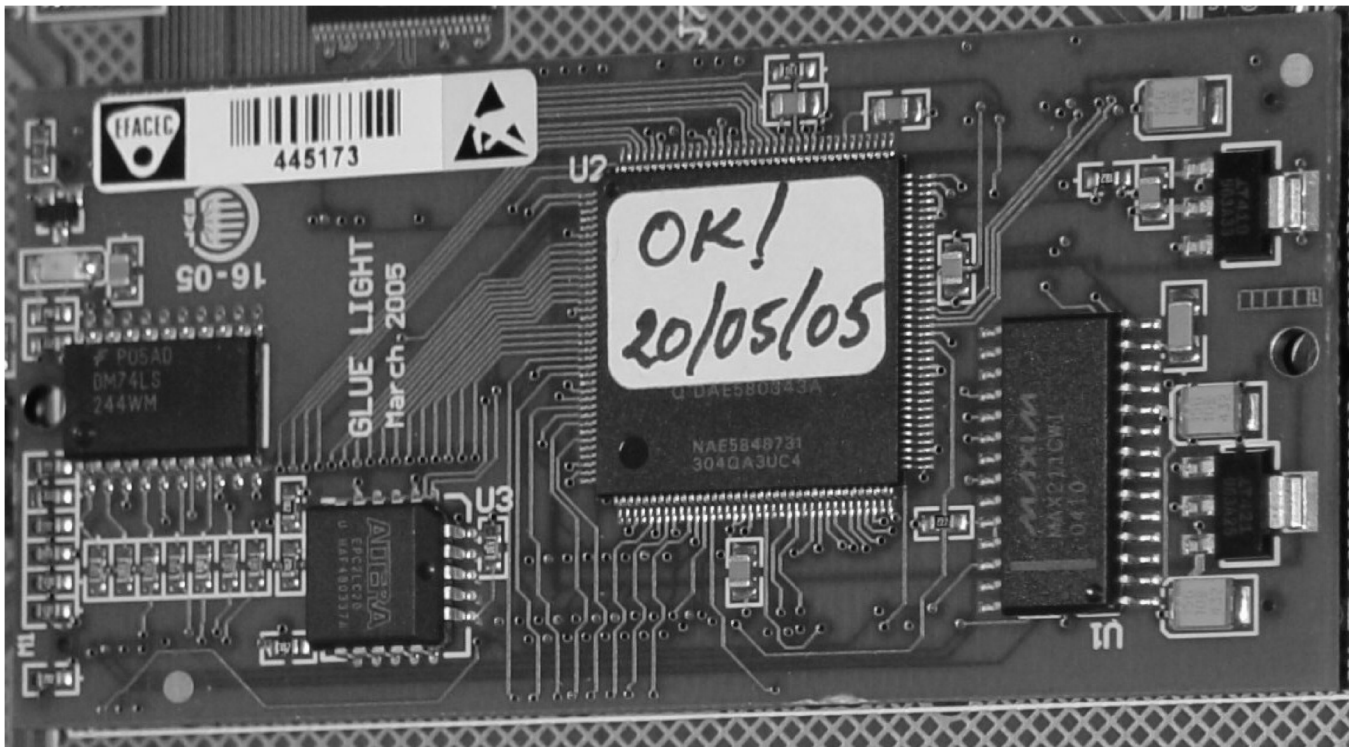


Fig. 3. FPGA based glue-card. A simplified local bus in the style of the PCI PLX 9030 is created by a small ALTERA FPGA (shown in the middle) which also creates I2C and JTAG interfaces. The board is slightly more integrated than the ASIC based card, at the expense of a somewhat more complex and more expensive FPGA.

management system (RPM) [5]. This is described in the next section.

Except for the device driver no cross-compiler or other special software is required. Editing, compiling and, to the extent of not accessing the actual board hardware, debugging can be done on any Linux PC. Most developers run all development

tools on a server, which shares the files with the embedded PC via the network file system (NFS).

B. Management Software

In a large collaboration it is necessary to disseminate the software quickly and transparently to all teams. The resulting

common software base allows effective peer-help. In the present case this task is made more difficult owing to the wide geographical spread of the teams. There is also no common intranet, as in some large companies. For example, a shared NFS server for the software is possible from a performance point of view, but is not an option in the age of firewalls, net-filters, etc. The software distribution is therefore based on HTTP and a simple automatic update mechanism. This requires a minimal administration effort by the users and can be fully automated.

In this manner all security updates and software patches which concern packages not directly managed by us, the vast majority, benefit from the effort of the respective support teams. The embedded controllers can therefore be put on the LAN or internet with no more risk than a standard desktop PC running Scientific Linux. This is an issue mostly during the R&D phase, where test-setups are operated in various laboratories, and the overhead of a private LAN, such as will be operated in the LHCb experiment is not justified. Since the embedded PCs are no less secure than standard desktops, users can connect them to their campus LANs. This approach allows *direct* remote accessibility of individual machines, a fact that greatly eases remote, rapid debugging and support.

The system relies only on a (central) web-server and a few simple scripts, which are documented and available at our webserver [7]. The package-management of the RPM packages is done by the powerful tool YUM [6] which, in particular, adds automatic resolving of package dependencies to the RPM functionality.

C. High Level Software

At a higher level all control functionality in LHCb is integrated in the Experiment Control System (ECS), which is based on a commercial SCADA system. This system provides alarm-handling, finite-state machine modeling, graphical user interfaces etc. More details on this layer and its capabilities can be found in references [8] and [9].

V. CONCLUSION

A. Commercial Solution

Selecting a commercial micro-controller board allowed profiting from the extensive design work done by the PC industry. Moreover, the design against a standardized connector pin-out has proven to be invaluable as it effectively shielded the system from problems with component obsolescence. For instance, when the producer of the original micro-controller went out of business, Digitallogic redesigned a board with the identical connector and mechanical dimensions around a different micro-controller from AMD, the ELAN520, which is presently being used. This was transparent to LHCb, both in terms of design effort and costwise.⁵

Unfortunately, commercially available components can also have unexpected problems. During final testing it turned out that due to the specific connector used on the SM520PCX module,

⁵The only real difference is a slightly higher power-consumption of the ELAN520, which uses on average 1.4 Ws, a value perfectly acceptable for our purposes.

it was not possible to drive Ethernet from the card over more than 20 m at 100 MBit/s. Apparently verification had only been performed at 10 MBit/s or short cables, where the problem does not occur. Since the only solution to this problem is to exchange the connectors and 600 modules had already been produced, it was decided to place the network switches required to connect the PCs to the control-system LAN, within a distance of 10 m. This is easily achievable within LHCb.

Another disadvantage is the need of a separate glue-board, which necessitates additional connectors and routing of many signals. This makes integration on carrier-boards somewhat cumbersome. A micro-controller, which already has several of the interfaces available on chip (I2C, JTAG) and is combined with an FPGA part, would allow everything to be on a single chip. Such chips exist already, unfortunately not yet based around an i386 compatible core, which is a key advantage for long-term, low-effort maintenance of LHCb's large distributed system.

There is currently a lot of interest in these systems, as can be seen in these proceedings, for example see references [10] and [11].

B. Using Standard Linux as the OS

The use of Linux has proven to be an excellent choice. Users were immediately at ease and practically any software available on normal desktop machines ran easily. The unfortunate exceptions were some of some proprietary FPGA programming tools, where extensive reverse-engineering was required to provide the functionality from Linux. In our domain of application we never found the need for "hard" real-time behaviour of the operating system. High-speed time-critical control is anyhow better done on the carrier-board itself in state-machines than on FPGAs. Since each PC controls only one board, time-critical tasks can always resort to polling. This allows for an excellent response time, if a low rate of small but unpredictable extra delays is acceptable. Critical tasks, which should not be interrupted could be added to the kernel. Linux can be made very-light weight and still be kept up-to-date with whatever is in fashion in the "real world". This is very important in the conception of a system which will be required for some 15 years.

C. Comparison With Crate-Controllers

The use of crate-control with embedded processors has at least three advantages over using crate-controllers. The first advantage is scalability. The system grows with each board added and each board gets its own, unshared 100 MBit connection. When using a controlling PC which also acts as an NFS server, connected via a Gigabit link to the network, no degradation of performance could be observed for up to 15 boards. For additional security and robustness a private LAN between the control-PC and the boards will be used in LHCb. The second advantage is the individual, isolated, direct access to each board even from remote sites. Finally, the increased robustness of the system against a faulty board is a distinct advantage. Such a faulty board can not affect the accessibility of other boards in the system, since there is no shared access from the controlling unit.

ACKNOWLEDGMENT

The authors would like to thank the whole LHCb Online team and the many users of this system for their help and feedback.

REFERENCES

- [1] *Digitallogic, Switzerland*, [Online]. Available: <http://www.digitallogic.ch>
- [2] *Advanced Micro Devices*, [Online]. Available: <http://www.amd.com> [Online]. Available: http://www.amd.com/us-en/ConnectivitySolutions/ProductInformation/0,50_2330_8634_8635,00.html
- [3] *PLX Technology Inc.*, [Online]. Available: <http://www.plxtech.com>
- [4] *Scientific Linux*, [Online]. Available: <http://www.scientificlinux.org>
- [5] *Redhat Package Manager, RPM*, [Online]. Available: <http://www.rpm.org>
- [6] S. Vidal, *Yellowdog Updater Modified, Yum* [Online]. Available: <http://linux.duke.edu/projects/yum/> [Online]
- [7] *LHCb Online webpages*, [Online]. Available: <http://cern.ch/lhcb-daq>
- [8] S. Schmeling on behalf of the JCOP Project Team, "Common tools for large experiment controls—A common approach for deployment, maintenance and support," *IEEE Trans. Nucl. Sci.*, vol. 53, no. 3, pp. 970–973, Jun. 2006.
- [9] C. Gaspar and B. Franek, "Tools for the automation of large physics experiments," in *Realtime Conf.*, Stockholm, Sweden, 2005.
- [10] J. Alme, "The control system for the ALICE TPC detector," in *Realtime Conf.*, Stockholm, Sweden, 2005.
- [11] S. Silverstein, "A Simple Linux-Based Platform for Rapid Prototyping of Experimental Control Systems," in *Realtime Conf.*, Stockholm, Sweden, 2005.