SPS/LEP Beam Transfer Equipment Control Using Industrial Automation Components

A. Aimar, G. Bérard, J. -L. Bretin, E. Carlier,
J. H. Dieperink, M. Laffin, V. Mertens, H. Verhagen
European Organization for Nuclear Research (CERN) - SL Division,
CH - 1211 Geneva 23, Switzerland

Abstract

Several control systems for SPS and LEP beam transfer equipment have to be commissioned in the near future. Tools for fast software development, easy maintenance and modifications, compliance with industrial standards, and independence of specific suppliers are considered to be essential. A large fraction of the systems can be realized using off-the-shelf industrial automation components like industrial I/O systems, programmable logic controllers, or diskless PCs. Specific electronics built up in G-64 can be integrated. Diskless systems running UNIX and X Windows are foreseen as process controllers and local access media.

I. INTRODUCTION

Within the SPS and LEP Beam Transfer sector at CERN, there are currently several control systems being prepared for the application in both accelerators. Among those are, in different phases of progress, the control for the LEP beam dump and for the LEP Pretzel beam separators.

Whereas the size and the complexity between our different systems varies considerably - from some 10 I/0 channels to nearly 1000 - all systems can commonly be characterized as 'slow controls', i.e. the required response times as seen from the main control room are only in the order of seconds for most actions. Specific fast responses, e.g. for beam dumping, are supported by special hardware. The size of data exchanged between the main control room and the equipment is relatively small. However, the significance of the equipment for the machine operation requires a continuous monitoring of its performance and efficient diagnostic tools for debugging in case of failure.

To facilitate the running of an increasing number of systems with different functionality and different composition with the available manpower we are looking for rationalization opportunities. Inspired by investigations in the field of industrial process automation several areas have been identified. In this article we try to demonstrate how we plan to apply our findings in the coming generation of control systems.

In the following we shall first discuss the criteria which we consider essential for the selection of the appropriate components. Afterwards, the characteristics of the preferred field bus will be given. Then, the full concept will be presented complemented by a brief status report.

II. SELECTION CRITERIA

A. Hardware Interface

While the prices for equipment electronics in general are decreasing, any cabling, either inside crates and racks or between racks and equipment, is becoming a dominant cost factor, besides being error prone. Therefore all data should be picked up by remote I/O systems where they arise, whenever possible and if not totally inconvenient. All I/O systems belonging to the same logical system should be hooked to a field bus connected to a process controller.

A second factor is a clean and simple rack cabling scheme including appropriate connection systems permitting a fast and cost saving installation and modifications and an easy maintenance of the electronics.

Other important criteria concern the economic use of rack space and the cost, reliability, and the technological lifetime of the selected I/O systems.

B. Local and Remote Access Facilities

Since the equipment to be controlled is, for technical reasons, sometimes far away from the process control computer or even the I/O system, one has to provide means to influence or monitor the process locally (i.e. without the need to communicate to somebody in front of the computer), e.g. for debugging or calibrating an equipment. This can be done using local hardware, e.g. panels with DVMs, switches, and potentiometers, hand-held terminals in the case of programmable logic controllers, or portable PCs. It is essential to enable the person in charge of maintenance to get access to all important parameters of the system in a user-friendly form and to permit to develop or to modify test procedures rapidly.

Adequate facilities have to be added on the process controller level and on other systems from where people have to access the equipment, which should be as complete as possible, preferentially without much adaption work for different platforms.

C. Software

The equipment software can be a particularly problematic area in that here the user requests have to be translated into specific instructions by comparatively few specialists. The consequences have intensified within the past few years with the trend towards a higher degree of distributed processing, made possible through the application of modern technology.

In fact, if reaction time is virtually no problem as in our applications, distributed processing can largely be replaced by central processing (data 'harvesting'), using sub-systems mainly as remote I/O stations. Local processing can be restricted to those tasks requiring a really tight control over the process thus reducing the communication traffic and freeing the central system, or to mask hardware peculiarities. This reduces the amount of code to be written for the communication between the sub-systems. Software development is also speeded up due to the usually superior facilities available on the central system.

The facilities for the development of the remaining sub-system code should still be adapted to the amount of code to be delivered (e.g by using cross software on PCs).

With the distances (and sometimes also the number of systems) involved the possibility to download code rather than to change EPROMs has also become mandatory.

D. Field Bus

The proper choice of a field bus is a key item in that the field bus constitutes the same for the equipment electronics as a network for the process controllers and other computers. To a certain extent it predetermines the type of equipment to be used. Generally spoken, the adherence to industrial standards widens the choice for off-the-shelf acquisition of material.

Besides the obvious criteria like reliability (recognition of transmission errors), sufficient speed and extensibility, and reasonable cost we consider a good multi-vendor support essential. The applied field bus should be able to host equally different types of industrial equipment as well as existing home-made equipment.

III. THE BITBUS

A. General Remarks

We find that the Bitbus fulfills our criteria for a field bus in a nearly ideal way and devote therefore some room for its discussion.

The Bitbus is a high speed serial control bus which was developed in the early eighties mainly by Intel [1] and which has meanwhile found a broad worldwide distribution. In fact, it has become one of the de-facto standards for industrial control applications. It is already being applied in other physics research institutions, e.g. in [2, 3, 4].

B. Basic Characteristics

Electrically the Bitbus is based on the RS-485 standard. It uses twisted pair wires for transmission with termination resistors on both ends forming a segment. Such a segment can have a maximum length of 0.3 km hosting up to 28 'nodes' when operated at a speed of 375 kbit/s. If a speed of 62.5 kbit/s is used, and with 10 repeaters, a maximum length of 13.2 km can be reached. The maximum number of slave nodes is 250. The communication is based on messages (order-reply) exchanged between nodes. The node connectivity is master-slave-slave-... with the master node usually residing in the host interface.

C. Data Integrity

A subset of the SDLC protocol (= Synchronous Data Link Control) by IBM is used as transmission protocol describing the data exchange and the access rights of the different nodes. For assuring the correctness of delivered and received data a CRC (= Cyclic Redundancy Check) method is employed.

D. Node Structure

Each node contains a 8044 microcontroller built up around the 8051 microprocessor. These controllers run the multi-tasking operating system DCX-51 contained in firmware. The communication between the master node and the slave nodes is performed at the task level, i.e. the controllers take care of the communication between nodes using Bitbus messages. In each node the task 0, called RAC (= Remote Access and Control) task mediates the accesses from the applications to the I/O port of the nodes. The DCX-51 permits to run up to 7 user tasks in addition which can be used to perform local processing if desired. This is a particularly attracting feature since one is not obliged to write any node software to read or write I/O ports, but provides a proper framework in case one needs to do something more.

E. Software Tools

The software development for the nodes can be done 'cross' on PCs under DOS (or e.g. on a workstation running a DOS emulation) either in PLM/51, in C, or in ASM/51 (assembly language). Afterwards, the code is downloaded. The allowed code size is typically about 32 kbytes.

For the communication between DOS-based hosts and the nodes exist high-level interfaces to the usual languages. Monitor programs of several suppliers allow to perform all basic functions on the nodes or to check the actual hardware configuration on the bus.

A driver to use the Bitbus on PCs running the real time operating system LynxOS as proposed for the future use in all process controllers at the CERN accelerators [5] is currently under development.

F. Multi-vendor Support

The widespread support the Bitbus has found is reflected in the number of interfaces available to different host systems: There exist links to VME, PC, Q-Bus (μVAX), Multibus-II, and the SMP-Bus, partly also from several suppliers. A SCSI interface developed recently [6] allows to connect Bitbus based systems also to workstations, for instance.

On the node level, there is also a great number of possible choices with different characteristics and form factors: I/O modules with fixed functionality, modular I/O systems consisting of a crate with a node controller and various I/O modules, programmable logic controllers (automata), or PCs used as slaves nodes. The recent development of a Bitbus node controller with G-64 interface [7] permits now to integrate also electronics based on this standard. Last but not least, piggy-back nodes are available for direct implantation into special equipment.

This multi-vendor situation is advantageous when selecting the appropriate components due to the competition between suppliers.

The recent implementation of the Bitbus protocol on the newer microcontroller 80C152 (offering enhanced possibilities, but remaining backwards compatible) indicates that this concept will retain its importance over the next years.

G. Experience

To test the feasibility of the Bitbus in our environment we installed a PC together with a Bitbus node in an equipment hall and connected them through a non-shielded twisted pair cable of 15 m length, with the Bitbus loosely traversing an area heavily perturbed by fast high current discharges from one of the SPS injection systems. The system run over several days without halting or producing transmission errors.

For test setups in the laboratory we simply use flat cable. The use under DOS is greatly simplified due to the available good software support in form of high-level language interfaces, libraries, and monitor programs.

IV. ARCHITECTURE

The full controls concept actually pursued comprises the following ingredients (please refer to figure 1).

A. General layout

On the level nearest to the equipment two major cases can be distinguished:

a) In case of dominant specific requirements (as e.g in the LEP beam dump) the I/O hardware will mainly consist out of G-64 or Eurocard form factor modules, as practised in the past for similar applications. Inter-crate wiring is to a large extent done using coaxial cables. Built-in front panel elements are used for basic local control. Diskless industrial PCs running DOS are foreseen for certain applications which require greater local processing power and better display and access facilities, e.g. fast data logging and fault detection. The use of the Bitbus as field bus is envisaged for all sub-systems needing few or no specific processing, for others the connection can be made using RS-232 links between a processor in the G-64 crate and the process controller via a Terminal server (this would have the advantage that no additional driver software is required). Necessary industrial measurement devices like digital storage oscilloscopes will be branched using a GPIB link.

b) In case of more standard requirements (e.g. for the LEP Pretzel beam separators) the I/O hardware will predominantly been built up using industrial components, preferentially in

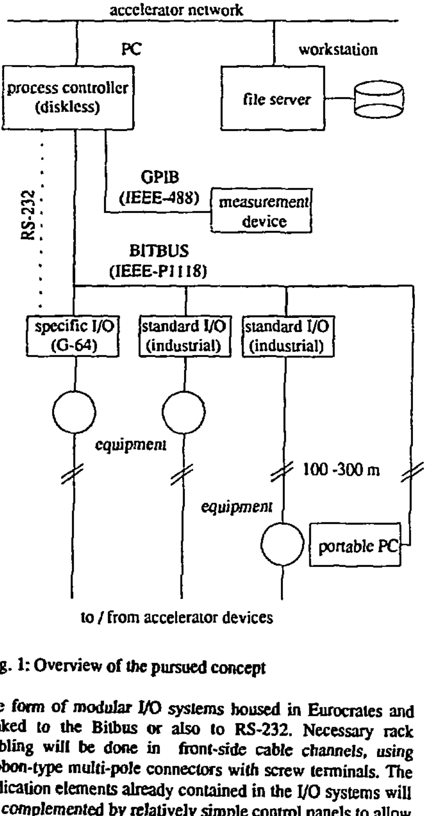

Fig. 1: Overview of the pursued concept

the form of modular I/O systems housed in Eurocrates and linked to the Bitbus or also to RS-232. Necessary rack cabling will be done in front-side cable channels, using ribbon-type multi-pole connectors with screw terminals. The indication elements already contained in the I/O systems will be complemented by relatively simple control panels to allow local control in case of malfunctioning of the higher levels.

To provoke actions when servicing in remote areas (e.g in the accelerator tunnel) portable PCs will be used. The desired actions will be provoked by communication with the process controller via the Bitbus.

B. Node Software

The software to run on this level will be restricted to the minimum. Typical examples are routines taking care of hardware particularities or the transformation of measurements into data directly useable by the higher levels. The software for the Bitbus nodes will be cross-developed in PLM/51 on PCs running under DOS (or alternatively on workstations with a DOS emulation) and then stored on the file server for downloading. The software for diskless PCs on the Bitbus can be developed using standard PC tools like Turbo Pascal or Quick C and will stored in non-volatile memory.

C. Process Controller

All equipment specific electronics will be connected to the process controller put to our disposal by the SPS/LEP controls group as described in [5]. This device, an industrial PC running under the real time operating system LynxOS, is directly connected to the accelerator network. This PC will be diskless, possessing sufficient memory to run, besides the general software, the equipment specific control software without the need for swapping. The required code will be loaded at boot time from a file server which will also be used to store historical data for later analysis.

D. Local Control and Diagnostic Software

The implementation of X Windows on these systems lays the ground for the implementation of the required good equipment specific control and diagnostics software including graphics, to be used simultaneously for local and remote access with interaction possible through workstations, X terminals, or PCs under LynxOS or under DOS.

Such a software, at the same time aimed at grouping our various needs into one configurable package, is currently being developed and described in [8].

E. Test Setup and Status

The key ingredients of the presented architecture have been set up in the laboratory using a diskless DECstation 5000/125 to simulate the final process controller, with another workstation acting as remote file server. The Bitbus has been attached using a SCSI-Bitbus gateway. Since this device is conceived to simulate an already existing SCSI device there was no need for writing special driver software thus permitting to advance more rapidly. PCs under DOS are used in some smaller test applications and for development of node software.

V. CONCLUSION

Due to the continuous progress in automation industry, many different types of I/O systems have become available which can directly be used for slow controls applications at accelerators. Home-made equipment can easily be integrated.

The choice of the communication link to these systems is important. The use of an open, non-proprietary link like Bitbus, allows, for many of our applications, off-the-shelf acquisition of a variety of systems from different suppliers.

This permits to concentrate our efforts on system integration and on the more specific accelerator control electronics which remains to be designed in our laboratory.

VI. ACKNOWLEDGEMENTS

The continuous support of the SL/CO group and their efforts for providing the necessary general hardware and software infrastructure are gratefully acknowledged.

Many thanks go also to the DEC Joint Project Office at CERN for the generous and informal help.

The authors would like to thank the organizers of this conference for their efforts.

VII. REFERENCES

[1] INTEL, The Bitbus Interconnect Serial Control Bus *Specification*, Ref. 280645-001, 1988.
[2] INFN, Milano, Italy, D. Giove, private communication.
[3] EPFL, Lausanne, Switzerland, J. Lister, private communication.
[4] KVI, Groningen, The Netherlands, P. Kroon, private communication.
[5] R. Rausch and Ch. Serre, "Common Control System for the CERN PS and SL Accelerators", this conference.
[6] INCAA Computers, Apeldoorn, The Netherlands.
[7] INCAA Computers, Apeldoorn, The Netherlands, and J. H. Dieperink, CERN, private communication.
[8] A. Aimar, E. Carlier, V. Mertens, "An Open Software System Based on X Windows for Process Control and Equipment Monitoring", this conference.