A NEW CONTROL SYSTEM FOR THE S-DALINAC[‡]

M. Platz[§], A. Araz, U. Bonnes, M. Brunken, H.-D. Gräf, M. Hertling,
A. Karnaukhov, O. Patalakha, A. Richter, O. Titze, B. Truckses,
Institut für Kernphysik, Technische Universität Darmstadt,
Schlossgartenstrasse 9, 64289 Darmstadt, Germany
W. Ackermann, W. F. O. Müller, B. Steiner, T. Weiland,
Institut Theorie Elektromagnetischer Felder, Technische Universität Darmstadt,
Schlossgartenstrasse 8, 64289 Darmstadt, Germany

Abstract

Recent results with respect to the development of a new control system for the superconducting cw electron accelerator S–DALINAC are presented. The system is based on common industrial standards. Due to the exceptionally large number of devices necessary to control the beam at the S–DALINAC, a simple and inexpensive communication interface is required to replace the current proprietary bus topology. The existing devices will be extended by a microcontroller based CAN-Bus interface as a communication path to one or more control servers. These servers utilize a TCP/IP connection to give application programs (clients) access to the device parameters. The protocol for this communication is composed of a special binary protocol and a text protocol based on XML.

INTRODUCTION

The superconducting recirculating electron accelerator S–DALINAC [1] is designed to deliver a continuous wave (cw) electron beam for nuclear and radiation physics experiments and has commenced operations in 1991. It provides an electron beam with an average current of up to 60 μA at energies of up to 130 MeV. A variety of nuclear physics experiments require the development of a new photoinjector electron gun as well as the increase of energy and intensity of the electron beam. For all experiments a reliable operation of the accelerator based on a modern control system is needed.

The accelerator and its beam transport system shown in fig. 1 consists of about 200 magnets, 80 targets and cameras and several special setups [2], e.g. Faraday cups for beam current measurements, beam position monitors or Compton diodes, to be controlled. The possibility and the demand of controlling additional devices, e.g. new diagnostic elements or parameters of the Helium liquifier, require the development of a new control system which will be realized in three steps.

The human device interface (HDI) used to control the S–DALINAC is a so-called knob board. A knob board is a thumb device emulating an analog tuning of the accelerator. A counter adds up the increments of a rotary encoder and sends the result to a knob server. Button switches are used for digital inputs, e.g. for switching power on/off or toggling between the dial mode for selecting a device from a list, which is defined for a special mode of operation, and the set mode for changing the set value of the selected device. Depending on the mode of operation the server recalculates an internal value, changes the device parameters and displays the result in the LCD elements.

The control system currently used [3] consists of a remote part based on VMS and PSOS including a knob server with an MC68020 processor, giving the possibility of selecting and turning devices, and a VAX 3800 acting as client. As control computers MC68020 microcomputers running under PSOS as operating system (OS) and two AXPvme boards using VxWorks as OS are used. While the MC68020 systems accessing the devices through a local VMEbus, e.g. the HF Control System, the gross of the devices were connected to the AXPvme via a proprietary bus topology. For the communication between the clients and servers, the Linac Control Protocol (LCP) based on Ethernet/DECnet was developed.

Although a reliable operation of the accelerator is possible, this solution has several drawbacks:

- Due to the proprietary bus, an implementation of new types of devices is rather difficult.
- The support for the AXPvme Single Board Computer (SBC) is discontinued.
- The AXPvme board architecture is no longer supported by new versions of VxWorks.
- Therefore, no hardware manufacturer supports the combination of SBC and OS.
- The clients are using VMS as OS.

THE NEW CONTROL SYSTEM

The aim of the new control system is to use a standard transport protocol like TCP/IP instead of the unique LCP used before, which will allow the definition of an application layer protocol independent of operating systems.

[‡] Supported by the Sonderforschungsbereich 634 ”Nuclear Structure, Nuclear Astrophysics and Fundamental Experiments at low Momentum Transfer at the Superconducting Darmstadt Electron Accelerator (S–DALINAC)” and by the Graduiertenkolleg 410 ”Physics and Technology of Accelerators”

[§] platz@ikp.tu-darmstadt.de

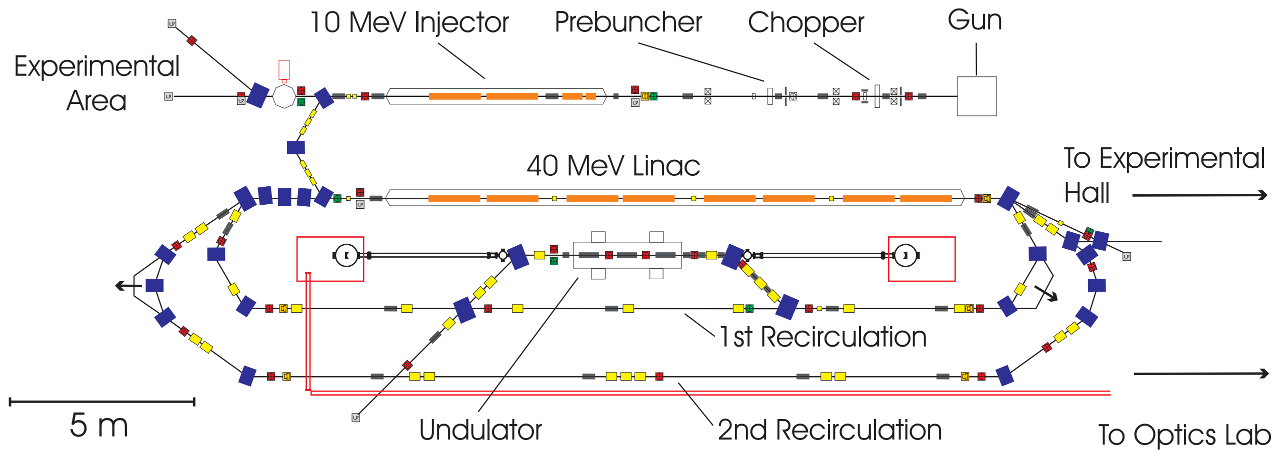

Figure 1: Layout of the S–DALINAC.

Requirements

From the experience gained in the last years several requirements for the new system could be defined:

- Low cost and reliable.
- Knob boards are to be retained.
- The system should allow easy maintenance.
- Simple integration of new elements in soft- and hardware.
- No real time beam alignment necessary.

Since the machine and personal interlock is completely hardwired, the control program should be able to retrieve and record the status. Furthermore, only this control program is going to be used for adjusting the machine parameters whereas an automatical beam alignment will not be implemented in the first step. Since the manual adjustment of the beam alignment is considered to be a rather slow process, neither a fast interaction between devices nor a common timescale (trigger) is required. Since the devices are distributed over the entire area around the accelerator, the rewiring and positioning of the control computers have to be considered carefully.

Layout

A schematic layout of the new control system is shown in fig. 2. As mentioned above, the first aim is to control all devices by a control computer, there will be no need for an implementation of a direct interaction of two devices. If a fast control operation is required, e.g for the HF control, this can be realized by a special hardware bypassing the control program. In case of major beam losses the machine and personal interlock system will stop the beam. By implementing dynamically selectable devices used for beam diagnostics into the control system, slow drifts of the beam might be compensated automatically. This will be developed in the future.

The control system is based on several control servers. The server functionality is implemented in the core of the application running on the control computer, allowing clients to request a TCP/IP connection. The functionality necessary to handle the particular devices is loaded to the application program by initialization in form of plugins. The information about the plugins needed and the devices to be controlled is inquired by the server from a global database. This database is also used by clients to find the addresses of the servers handling the devices to control.

Clients and servers exchange new device settings as follows. The client application sends a packet over a TCP/IP connection to the control server. The first byte is a protocol ID defining the structure of the following data, e.g. the ASCII value of ”<” in the special case of an XML based protocol. Usually a packet is dedicated to a specific device and is thus handled by a specific driver, i.e. only this driver is able to understand the request. Therefore a dispatcher task receives the packet from the TCP/IP socket, extracts the command for the specified driver and calls the driver entry point with the command as argument. After the driver has executed the requested operation (synchronous or asynchronous) it notifies the dispatcher task in order to

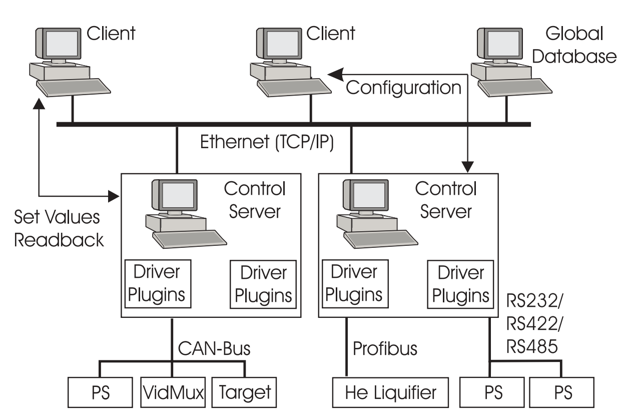

Figure 2: Layout of the control system, where PS denote a power supply for the magnets and VidMux denotes the Video Multiplexer.

initiate the reply to the client application. It turns out that this leads to software interfaces between drivers and server applications which are device independent, but they depend on the protocol. Therefore these interfaces have to be defined carefully to allow further extensions of the system.

All noncommercial devices will have the same bus interface. The device specific part of the hardware is controlled by the microcontroller attached to the bus controller, leading to a distributed system of ”intelligent” devices. While the software for the bus communication is planned to be the same for each device, the software controlling the device specific part has to be developed in conjunction with the new hardware.

As common bus interface the CAN-Bus was chosen, because it is a simple, inexpensive and reliable multimaster field bus, defined by an ISO standard for the automotive industry. Compared to other commonly used buses, like Ethernet, the speed is low, the packet size is small and it therefore suits the demands of the devices equipped with a low cost microcontroller, namely a small memory and a low computing speed. The ISO standard defines the bus on the OSI Layer 2 (datalink layer) as an object oriented bus, i.e. an object ID is defining the type of the packet. Each destination programmed to recognize this object ID will then receive the packet. The priority of the packet is determined by the object ID; the higher the object ID, the lower the priority This is to overcome conflicts in case of different controllers trying to get access to the bus simultaneously. Because of the necessity to address special devices instead of a common set of devices, the datalink layer has to be extended. This is done by redefining the object ID of a packet in that way that it contains the address and the type of service definition of the device. Thus it is possible to create reliable connections between two peers, allowing the transmission of packets as well as setting up streams.

Client Applications

Among the set of different client applications the tuning of the accelerator by using the knob boards is one of the most important. In this scope a Window based application for driving targets and selecting video multiplexer channels, giving information about the beam position and shape, is essential. By omitting the proprietary LCP, clients may be allowed to easily access the control servers automatically. In the simplest approach this could be used for beam simulation calculations, e.g. based on the V-Code [4], or will in a more complex way lead to the possibility of automatic beam emittance measurements, which is not possible presently. Finally, specific clients may be used to control slow drifts, depending on operator defined machine premises.

Current Status

The implementation of a new control system into a running accelerator is a rather difficult and complex task. So far, prototypes for the required hardware, e.g. improved knob boards and the CAN-Bus interfaces, have been designed and built. Currently this hardware is tested. The second major issue is the development of new software. As far as the application clients are concerned, a preliminary application has been accomplished and tests are currently been carried out. The global database for the storage of the data for devices and the monitored parameters have been successfully tested. The computer program for the microcontrollers is almost finished and will be tested accurately thereafter.

SUMMARY AND OUTLOOK

While designing the new control system special attention was payed to the flexibility, reliability and simplicity. Regarding the hardware the extension of the control system is achieved by a simple field bus system sufficient for slow control operation of the S–DALINAC. By using the same interface for all uncommercial devices a well-defined set of functions merged in a library could also be used to implement new types of devices.

By using TCP/IP as transport layer the development of clients is possible employing all well established operating systems. The lynchpin is the application layer protocol used to exchange the information between clients and servers. Utilizing XML as an object oriented markup language a set of common device requests could be defined. From that new device request classes can easily be derived.

By extending the server application with plugins, new drivers could be easily added after off line development and testing. The software interface between server application and driver plugin is an entry point receiving the device request that is defined by the application layer protocol. After the first device driver able to execute the common request has been developed, the others could be derived from it. The realization of the control system described in this paper will give the possibility of a flexible and extremely convenient operation of of S–DALINAC.

REFERENCES

[1] A. Richter, Proc. EPAC96, ed. by S. Myers et al., IOP Publishing, Bristol, Barcelona (1996) 110.

[2] H.-D. Gräf, J. Herrmann, M. Hüning, U. Laier, P. Mohr, M. Platz, A. Richter, B. Schweizer, S. Watzlawik, H. Weise, NIM A **512** 3 (2003) 453.

[3] H. Genz, H.-D. Gräf, J. Horn, V. Huck, K.D. Hummel, M. Knirsch, C. Lüttge, A. Richter, T. Rietdorf, K. Rühl, P. Schardt, E. Spamer, A. Stiller, F. Thomas, O. Titze, J. Töpper, H. Weise, Proc. EPAC92, ed. by H. Henke, M. Homeyer and Ch. Petit-Jean-Gomez, Edition-Frontières, Gif-sur-Yevette Cedex (1992) 53.

[4] W. Beinhauer, R. Cee, W. Koch, M. Krassilnikov, A. Novokhatski, S. Ratschow, T. Weiland, P. Castro, S. Schreiber, Proc. PAC2001, Chicago, Illinois, USA, 06/18-06/22/2001.