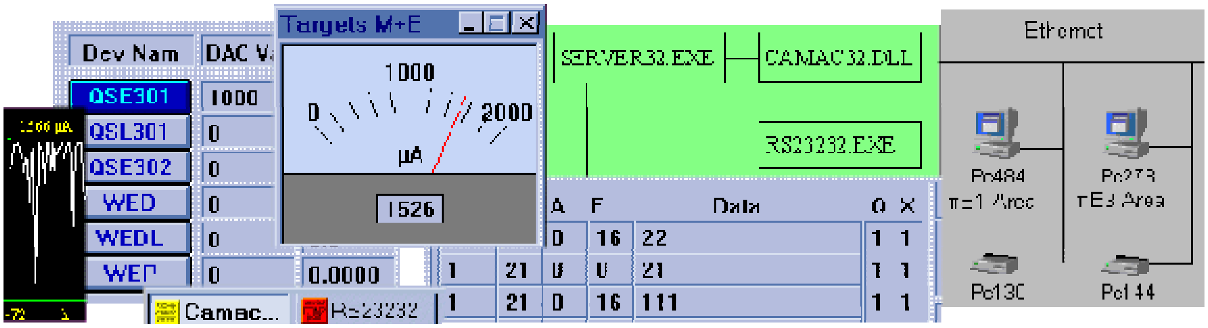
PAUL SCHERRER INSTITUT

Secondary Beam Line Control System



The current **PSI Secondary Beam Line Control System** has evolved from the former system, which was based on the usage of Atari ST computers [1]. The basic principle behind the current system for the secondary beam line control is still the same: All seven secondary beam line areas in the experimental hall must be independent of each other, but should use the same software (e.g. Operating System and Programs) and hardware components (e.g. Camac and Road-C). If one secondary beam line area is down, the others shouldn't be affected by this event. But if resources like power supplies have to be shared, then there should be the possibility to connect the components of two (or more) secondary beam line areas via networking. Furthermore, the system has to be small, simple and easily maintainable. This has been realized during the years 1995-96 by using Intel-CPU based PCs as hardware platforms, running MS Windows 95 (upgraded later to Windows 98 and now to XP-SP2) as OS and utilizing a **TCP-IP Client/Server architecture** (see Fig.1 (*14 kB*) and Fig.2 (*16 kB*)). Each secondary beam line area has its own CAMAC crate, which is connected via RS-232 with the server PC and contains interfaces and modules according to the needs of the specific secondary beam line area. In Fig.3 (*66 kB*) a photograph of a typically equipped camac crate (the one of the former µE4 secondary beam line area) is shown. All secondary beam line area client and server PCs are inter-connected via Ethernet and included in 3 Microsoft Network groups (AEA_NT, WEHAS and WEHAC) for easy maintenance (backups, downloading new programs, upgrades etc.) (see Fig.4 (*10 kB*)). A résumé about the features of this simple **Secondary Beam Line Control System** has been published in the **PSI Scientific and Technical Report 1998 Volume VI** (Large Research Facilities) on pages 18 - 21. More detailed information about some important aspects and 2 Java Applets are selectable via the following buttons:

Last updated by **Urs Rohrer** on 3-Aug-2006

Table containing some additional useful information:

[1]	U. Rohrer, Secondary Beam Line Computer Control at PSI. Annex I of PSI Annual Report 1988, p.7-8 (*21 kB*)
UPDATED	All 7 area server PCs have been replaced by new PCs running Windows XP-SP2 Professional as Operating System (in a FAT32 partition). The PC cases are black or light-gray colored rack-mountable 19" units. For more info press 'Used Computers' button. *(No hardware watchdog units are used anymore.)*
UPDATED	On all 7 area client PCs the Operating System has been upgraded from Windows 98 SE to Windows XP-SP2 Professional running in a FAT32 (4 GB) partition. The RAM has been upgraded from 128 to 256 MB and the Ethernet interface has been replaced with a modern 3Com 3C905CX-TX-M PCI card.
	Each of the 7 secondary beam line 's **device settings** are logged every 20 secs. These beam line data are kept in separate directories divided in up to 10 files of 1 MB size each. Thus, depending on the number of devices, data for the last 3 to 6 days are stored. You may access these logging data .
	2 spare PCs (server and client) are available for replacing temporarily a defective area or server PC. Here some instructions on how to proceed.

	A flaw in the 'Rate Optimizing Program for a Secondary Beam Line' (Optima) has been corrected. Until now the 6-fold scaler (S-500) has not been gated, which led occasionally to bad normalized experimental rates and which therefore disturbed the automatic search for the maxima. Now all 7 areas are equipped with an IO-50x unit (must be located next to the scaler unit on the right side) which controls the inhibit input of the scaler.
	All 7 area-server PCs have been equipped with a second CAMAC server module based on CORBA (**C**ommon **O**bject **R**equest **B**roker **A**rchitecture), which is fully multi-threaded. But in order to avoid conflicts inside the single-threaded RS232 module between this new server and the standard TCP/IP CAMAC server (both embedded in a Windows shell) a Critical Section Object (a component of the shared memory structure "Cambuf") acting as a Mutex inside the multi-threaded Camac-DLL code queues the accesses to the shared RS232 interface code (see Fig.5 (*7 kB*)). **CORBA** allows telecommunicating with remote objects (servers) on a relatively high level of abstraction. As CORBA implementation the freely available omniORB2 by AT&T (programmed in C++) has been chosen. In order to get information about the number of connected hosts and their names and how often they timed-out the omniORB2 API had to be expanded (by adding some code and rebuilding the libraries). First tests with this new CAMAC server (Server99.exe) show very encouraging results (stable operation and no side effects so far). Code for a simple CORBA CAMAC test client is presented here for C++ (AT&T's omniORB2) and for Java (Sun's JDK/IDL). The Seondary Beam Line Programs "Set Point", "Combi-Control", "Optima", "Magnet Cycling" and "Inter Active Camac Command" are now also available in the "C++ Corba client" version.
	For the 4 Secondary Beam Line areas πE1, πE5, πM1 and µE4 it is now possible to open and close the beam shutter(s) remotely. This is very useful for automatic data taking in case you have to check there from time to time the background with a closed beam shutter. The 3 other areas may also be connected to this feature if required by the experimentalists.
UPDATED	The 4 Secondary Beam Line programs 'Set Point', 'Optima', 'Combi Control' and 'Magnet Cycling' may now be connected to up to 4 server PCs. Because of the added complexity, for each of the 4 programs the indication of the server to which the page, group or device in use relates has been improved.
NEW	At the 3 Secondary Beam Line areas µE4, πM1 and πM3 the ROAD-bus has been replaced by the more reliable **CAN-bus**. The 'Firmware' of the new CAMAC-to-CAN interface handles CAMAC-I/Os 100%-compatible with those of the old CAMAC-to-ROAD interfaces. Therefore no program modifications had to be made. Only the local addresses of the different power-supplies may eventually had to be changed (new device.lis). The remaining 4 areas will also be equipped with the new bus during the next shutdowns (probably one per year).

NEW	In order to get a 16 times better resolution for some critical secondary beam line magnets, the "16/12 bits signed combi controller" (type = 10) has been implemented. (This controller supports polarity switching together with 16 bits DAC and 12 bits ADC resolution.) The first magnets equipped with this controller type are the bending magnets ASK71 and ASK72 at the πE3 area.

Secondary Beam Line Control System: Used Computers

The bandwidth of the hardware outfit of the used PCs may be seen from Table 1. As operating system **Windows XP-SP2 Professional** for servers and clients is used. If a system has to be rebooted or if it does it automatically after a power failure, then most of the time no user interference is required in order to bring the system up again. This is important for the **unattended servers**, which have no video monitor and no keyboard. Login is avoided by the enabled Autologon feature. The need for rebooting is relatively rare (less than once per 2 weeks).

The standard software outfit of the area client PCs can be seen from table 2. On all server PCs the *CAMAC server*, the *CORBA server*, the *pcAnywhere* client-server connectivity program by **Symantec** (in host mode), the *VNCserver* (Virtual Network Computing founded by **AT&T Laboratories Cambridge**, allows logon from Netscape via a Java applet), a *Watchdog* (observing the Camac server) and *McAfee VShield* are started up automatically at the end of each boot procedure.

Table 1: Hardware configuration of used Noname PCs

Mother Board	Shuttle AV42 for clients or Asus P4PE-X for servers. Cases for servers are of 19" rack mountable type.
CPU	Intel Celeron 2.0 GHz for servers or Intel Celeron 1.8 GHz (5) & 2.0 GHz (2) for clients.
RAM	512 MB (servers & clients)
Hard disk	EIDE, 8 GB (client PCs), 40 GB (server PCs), FAT32 file system

Ethernet adapter	on-board ASUSTek/Broadcom 440x for servers or 3Com 3C905CX-TX-M (PCI) for clients
Graphic adapter	MGA Matrox Millennium G200 AGP with 8 MB video memory (clients) or MGA Matrox Millennium 2 (PCI) with 4 MB video memory (servers)
Monitor	Acer AL1715 17" TFT-LCD monitors (client PCs only)
Keyboard	US ASCII (client PCs only)
Mouse	PS/2 (clients + servers)
COM2: port	Serial connection to CAMAC interface (server PCs only)

Table 2: Standard software outfit of area PCs

Program	Purpose
Optima et al.	Beam line programs
Netscape 4.7 or higher and MS Internet Explorer 6.0	WWW and mail client access
Atomic Clock Sync	automatic time adjustment
McAfee VShield	Virus protection and scanning

- **Secondary Beam Line Control System: *Home Page***
- Last updated by ***Urs Rohrer*** on 5-Jan-2007

Secondary Beam Line Control System: Available Programs

In List 1 follows an enumeration and short descriptions of the programs (Optima et al. for Windows) available on all 7 area PCs. They are all programmed in C++ using the WIN32-API and an own static class library [MyWPP.Lib, subset of Blaise Computing's WIN++ for MS Windows 3.1 (1991) ported to WIN32]. In order to add some additionally needed functionality to the WIN32-API, a few DLLs (Direct Link Libraries) have been created **(see List 2)**. All presented programs and libraries run currently under Windows XP Professional.

List 1:

- **Set Point Program** (DevoIP32.Exe). This program is the basic I/O control program, which allows the user to control the settings of all devices. Each device may be set individually via keyboard input or all devices together (or only a group of it) may be set from disk file. The display of the device parameters (name, DAC value and ADC value) is done via a page per page view **(see Fig.1, *23 kB*).** One page contains up to 16 devices. The actual beam line settings may be stored on disk file or compared with similar settings saved on disk file previously.
- **Optima Program** (OptaIP32.Exe). This program allows the user to bring up an arbitrary experimental rate provided by the user to a maximum by changing the settings of magnetic elements, slits and/or other elements of a secondary beam line according to a script file given to the program as input. When running, some output is displayed in three different windows of which one is graphical with auto scaling capabilities **(see Fig.2, *30 kB*).** The input script file can be created or modified either with a simple **text editor** or with a special GUI editor **(see Fig.3, *21 kB*).**
- **Combi Control Program** (CoCoIP32.Exe). This program is specially designed for collecting more information about the current status of power supplies controlled via ROAD-C *combi devices* (DAC, ADC, remote switch and some diagnostics are in the same plug-in unit). Besides switching on and off the main power for the devices one has also the possibility to set a value, to increment or decrement it or to inspect each meaningful bit of the status word of a selected power supply. All devices of a secondary beam line having a combi are listed in a so called combo-box and may be selected by double-clicking at the desired name **(see Fig.4, *20 kB*)**.
- **Magnet Cycling Program** (MaCyIP32.Exe). Up to 20 magnets may be simultaneously passed through a programmable magnetic field cycling process. The input for the program is a selectable script file, which contains 1 line for each of the chosen magnets. Each input line may contain a series of up to 10 pairs of time and DAC-setting-values placed behind the name of the respective device. Once started, the progress of the cycling process - displayed as current settings and trend plots - is shown in 2 windows (a text and a graphic one). A typical

output for a single magnet cycling process is shown in **Fig.10 (*17 kB*)**.

- **Interactive CAMAC Command Program** (ICCTCP32.Exe). This program represents a debugging tool for CAMAC experts. It allows to perform interactively all possible Camac commands (single or in a loop). The most general commands like Dataway Z, C, I etc. may be done directly by clicking at a reserved button **(see Fig.5, *22 kB*)**. A list of up to 10 commands may be typed in, stored or loaded from disk file. When executed, besides the data input/output also Q and X are displayed for each Camac transfer. The data representation may be chosen to be binary, octal, decimal or hexa-decimal. All commands are recorded and may be reviewed on demand via a special window. An initialization of the Kinetics KS-3989 Camac Serial Crate Controller may also be performed (with some diagnostic output).
- **2-Dim Scan Program** (2DScIP32.Exe). This program is capable of measuring and displaying experimental rates as a function of 2 varied device settings. During the data taking scans the rate profiles for each setting of the second device are plotted in a graphic window (similar to Optima). At the end, the result may be viewed via a contour, lego or surface plot. The input of the program is given via a script file which looks similar to one for the program Optima with only 2 devices to optimize. A picture of a test run is shown in **Fig.11, (*34 kB*)**.
- **Marquardt Gradient Search Program** (MarsIP32.Exe).
- **Proton Accelerator Status** (AccSta32.Exe). This program receives the UDP/IP packages broadcasted at a rate of 1 Hz by a computer in the PSI proton accelerator domain (ACS). The information contained in these packages is analyzed, stored and displayed in different manner like analog or as history plot (**see Fig.6, *31 kB***). The user may choose himself which beam current(s) and which history time span(s) he wants to have displayed. Beam drop-outs, availability and charge are also summed up and displayed upon request. All operator's messages and all HE-beam dropouts are collected in the Ascii files AccStat.txt and AccStat.off and may be inspected on request. Every 5 minutes or when exiting, the history data collected so far are saved in a file (AccStat.dat) as backup information for a desired or forced restart of the program. This program has also proven to be a useful ad-hoc beam monitor during the commissioning of the neutron guides of the spallation neutron source (SINQ). There exists also a modified version of this program designed as a Java Applet startable from your Web Browser (no previous installation required). In order to try out the applet, please **click here** . The applet consists of about 66 kB of code and 70 kB of GIF files.
- **Editor** (Editor32.Exe). This editor resembles to the well known Notepad editor. (Under Windows 95/98 it can only handle files of up to around 50 kB in size.) But besides the menu bar it has also a tool bar and a status bar displaying the current line number **(see Fig.7, *14 kB*).** The keypad may be used like with the DEC EDT editor well known to former or current VAX-VMS users.
- **Serial Port I/O-Handler** (RS23232.Exe). This program runs only on the server PC and is called by CAMAC32.DLL in order to execute the requested serial I/Os for the Kinetics 3989 RS-232 Crate Controller **(see Fig.8, *14 kB*)**. When RS23232.Exe is started it shows up in the task bar as a red icon **(see Fig.9, *9 kB*)** which has an extended system menu for requesting some extra information. No special driver is required to perform a RS232-I/O, because the serial port is supported by all Windows APIs. With a baud rate of 19200, up to 50 Camac cycles per second can be executed, which is sufficient for all available programs.
- **Test Program for Camac Server** (Client32.Exe). This program which runs in a DOS box allows the user to test all features of the Camac Server interactively. To learn more about the commands this program understands see **Table 1 in the Client/Server dialog section**. If a command is typed in, which the server does not understand, then the typed in command is

just prompted.

- **Camac Server** (Server32.Exe). This program runs only on the server PC and is started automatically at boot time (short cut in the startup folder). It can handle up to 10 clients at the same time (queuing) and calls routines in CAMAC32.DLL for performing the requested Camac-I/Os **(see Fig. 8, *14 kB*)**. When Server32.Exe is started it shows up in the task bar as a yellow icon **(see Fig.9, *9 kB*)** which has an extended system menu for requesting some extra information.
- **Watchdog** (Watchdog.exe).Also this program runs only on the server PC and is started automatically at boot time (short cut in the startup folder).

List 2:

- **Camac Library** (Camac32.Dll). This library is used by **Server32.Exe** and contains all routines for performing the Camac-I/Os of the above described programs. When initialized, it reads in the **database (DEVICE.LIS)** in order to learn about the addresses, mode bits etc. of all devices. A new initialization (over the network) is triggered each time when the **Set Point Program** is exited and restarted. This is important to know when changing the DEVICE.LIS on the server in order to make the newly typed in data effective. If a reservation unit (REMOD) is present in the crate and included in the DEVICE.LIS, then an arbitration between the different programs and a second computer connected via a second crate controller is performed.
- **User Drawn Button Library** (Usrbtn32.Dll). This library contains some bitmaps and routines for drawing buttons and patterns to give the dialog boxes of all above described programs the look similar to programs built with the old 16bit Borland bwcc-library independent of the brand name of the newly used 32bit C/C++ compiler.
- **Analog Meter** (Anamtr32.Dll). The source code of this DLL has been extracted form "Windows Programmierbuch" (Sybex 1992) by Marcellus Bucheit and ported to the WIN32 API. This DLL is used by the application program **Proton Accelerator Status** to display the different beam currents with the help of an analog meter.
- **Colored Button** (Colrbt32.Dll). This DLL is needed for producing buttons with arbitrarily colored faces, shades, highlights and fonts. It's utilized with the **Setpoint Program** for enhancing in different blues the device which has the input focus and to show in red the name of the currently loaded disk file.

Optional Add-On:

- **On-Line Transport** (Tpmenu32.Exe) In order to debug a new or a modified secondary beam line it is advantageous to have a version of **Graphic Transport** on some of the area PCs. This is the case for the πM3 and πE5 areas, where some development is still going on. If the Camac Server is present and Transport is started up with the second parameter on the command line as Server_name$Port_ID, then DAC values of magnets may be read or written directly from Transport. Of course at least one of the 2 files ***QUADS.CAL* and *BENDS.CAL*** with conversion data for DAC to magnetic field values for the magnets in use has to be present in the Transport directory.

Secondary Beam Line Control System: Used Database

Each line of the (ASCII) data base file DEVICE.LIS contains up to 17 parameters. The meaning of these parameters is given in table 1. The CAMAC and ROAD addresses are assigned by the Control System Group and are filed in their address book. (Contact Francois Kreis or Alfred Beck for more info). Additionally, there are 5 special types of lines (see also sample DEVICE.LIS below):

- **Line has only a star (*):** This line is only meaningful for the "Set Point" program , where it means, that the devices on the following lines should appear on a new page of the display. The maximum number of devices per page is 16. If more than 16 consecutive device lines are not interrupted by a star (*), then a new page on the display will be created automatically for the surplus devices.
- **Line is empty:** This is only for the "Set Point" program where it means, that an empty space after the preceding device should appear on the display.
- **Line starts with a minus sign (-):** This is for commenting out a line. The purpose is to keep the data in the DEVICE.LIS file for eventual later usage.
- **Device name = RESUNI:** This line has only to be present, if a reservation unit is used which is plugged in at the station (slot) N of the CAMAC crate. N must be set as the 4th parameter of this device line. All other parameters are dummy (e.g. 0).
- **Alias name = name:** This is for temporarily changing the name of a device without loosing the default name of the device. Different magnets of different experiments have sometimes to share the same power supply. The name of the power supply usually keeps the name of the original (default) magnet.

Caution: The file DEVICE.LIS must reside on the corresponding area server PC in the directory C:\OPTIMA. There are 2 methods to modify this file. Either you edit it remotely via the Microsoft network or you edit it locally on the server PC after a login via VNCviewer.

Table 1: Description of the 17 device list parameters.

Entry #	Group	Meaning
1		Device Name
2	DAC parameters	Special bit (0 or 1)
3		ROAD address (0 to 15)
4		Camac station Number (1 to 23)
5		Lower limit (in DAC units)
6		Upper limit (in DAC units)
7		Device index (see table 2)
8	ADC parameters	Special bit (0 or 1)
9		ROAD address (0 to 15)
10		Camac station Number (1 to 23)
11		Channel (0 to 31) (*
12		Range (see table 3) (**
13		Device index (see table 2)
14	Mixed parameters	Scale (+
15		Precision (+
16		Full scale (Amps etc.) (++
17		I/O-Flag (see table 4)

Table 1b: More details for table 1.

*	For SMK-DACs (Device index = 9) this parameter means a DAC offset. Example: If this parameter is -500 instead of 0, then the DAC range goes from -500 to +500 instead of 0 to +1000.
**	For SMK-ADCs (Device index = 9) this parameter means an ADC offset given in units of 0.001. Example: If this paramter is -500 instead of 0, then the ADC range goes from -0.500 to +0.500 instead of 0.0 to +1.000 .
+	**FAQ:** What do the 2 parameters 'Scale' and 'Precision' mean ? **Answer:** They are only needed for the **"Set Point" program**. If DAC and ADC values of a device do not correspond within a certain limit (= Precision), then a star is placed in front of the ADC-value and a sound signal is produced. Scale is the value which is taken by the ADC (in units of 1.0) if the DAC value is set to DacMax (= 2047, 4095 or 65535). (e.g.: Scale = 0.2 if a power supply runs in the '20 % mode'.) **Formula:** if [| (1.0 - (DacMax * |AdcValue|) / (Scale * |DacValue|)) | > Precision] then { print star and do sound }
++	**FAQ:** What does the parameter 'Full scale' mean ? **Answer:** It is only needed for the **"Set Point" program**. If it is not present or if it is zero, then the ADC value is displayed in the range between 0.0 and +- 1.0 . If a value other than 0 is given, then the ADC value is multiplied with this 'Full scale' value before displaying it.

Table 2: currently supported devices.

Index	DAC type	ADC type
0	12 bit with polarity (12SW1)	2DV8, 3DV32 (3 decades)
1	16 bit unsigned (16SW..)	4DV8, 4DV32 (4 decades)
2	12 bit Combi	12 bit Combi
3	DCN32 pointer	DCN32 (unipolar 1V)
4	16 bit signed (16SW1A)	like 12SW1A (12 bits only)
5	12 bit bipolar Combi	12 bit bipolar Combi

6	-	12 bit unipolar SADC (CSA)
7	12SW1A	12SW1A
8	12SWBA (bipolar)	12SWBA (bipolar)
9	SMK (12 bit)	SMK (12 bit)
10 NEW	16 bit signed Combi	12 bit signed Combi

Table 3: only applicable for multiplexed ADCs of type 0 and 1.

Value	Range
0	1 Volt
1	10 Volt

Table 4: possible I/O-Flag parameters

Value	Meaning
N	No Save on file
R	Remote access only
X	No Save on File, no automatic switch-on of Combi

Sample DEVICE.LIS (πE3 secondary beam line):

```
 RESUNI   0  0 12        0        0  0 0   0   0   0   0   0 0.0    0.0
 QTD71    0  1  3    -4095     4095  2 1   1   3   0   0   2 0.200 0.100 500.0
 QTD72    0  2  3    -4095     4095  2 1   2   3   0   0   2 0.200 0.200 500.0
 QTB71    0  3  3    -4095     4095  2 1   3   3   0   0   2 0.200 0.100 500.0
 ASK71    0  9  5    -4095     4095  2 1   9   5   0   0   2 1.000 0.100
 HSA71    0  1  5     -750      750  5 1   1   5   0   0   5 1.000 0.100
 QSK71    0  9  3    -4095     4095  2 1   9   3   0   0   2 0.200 0.100
 QSK72    0 10  3    -4095     4095  2 1  10   3   0   0   2 0.200 0.100
 QSK73    0 11  3    -4095     4095  2 1  11   3   0   0   2 0.200 0.100
 QSK74    0 12  3    -4095     4095  2 1  12   3   0   0   2 0.200 0.100
 HSA72    0  4  5    -4095     4095  2 1   4   5   0   0   2 1.000 0.100
 ASK72    0 10  5    -4095     4095  2 1  10   5   0   0   2 1.000 0.100
 HSD71    0  3  5    -4095     4095  2 1   3   5   0   0   2 1.000 0.100
 QSB71    0  4  3    -4095     4095  2 1   4   3   0   0   2 0.200 0.100
 QSB72    0  5  3    -4095     4095  2 1   5   3   0   0   2 0.200 0.100
 QSD01    0  8  5    -4095     4095  2 1   8   5   0   0   2 1.000 0.100
- QSB74    0  7  5    -4095     4095  2 1   7   5   0   0   2 1.000 0.100
*
 FS71-0   0  0  6        0     1000  9 1   0   6   0   0   9 4.095 0.100 0.0 N
 FS71-U   0  1  6        0     1000  9 1   1   6   0   0   9 4.095 0.100 0.0 N
 FS71-L   0  0  8        0     1000  9 1   0   8   0   0   9 4.095 0.100 0.0 N
 FS71-R   0  1  8        0     1000  9 1   1   8   0   0   9 4.095 0.100 0.0 N
 FS72-0   0  0  9        0     1000  9 1   0   9   0   0   9 4.095 0.100 0.0 N
 FS72-U   0  1  9        0     1000  9 1   1   9   0   0   9 4.095 0.100 0.0 N
 FS72-L   0  0 10        0     1000  9 1   0  10   0   0   9 4.095 0.100 0.0 N
 FS72-R   0  1 10        0     1000  9 1   1  10   0   0   9 4.095 0.100 0.0 N
*
 QSE43    0 11  5    -2047     2047  5 1  11   5   0   0   5 1.000 0.100 0.0 R
 QSE44    0  2  5    -2047     2047  5 1   2   5   0   0   5 1.000 0.100 0.0 R
 SINDRUM  0  8  3    -1500     1500  2 1   8   3   0   0   2 1.000 0.100 0.0 R
 WEK      0  6  3    -2047     2047  2 1   6   3   0   0   2 1.000 0.100 0.0 R
 WEN      0  7  3    -4047     4047  2 1   7   3   0   0   2 1.000 0.100 0.0 R
*
SOL01 = QSD01
```

- **Secondary Beam Line Control System: *Home Page***
- Last updated by ***Urs Rohrer*** on 10-Feb-2006

Secondary Beam Line Control System: Client/Server Dialog

TCP/IP programming interface between client and server: A list of all currently existing commands is given in table 1. They are heavily used by the programs Optima, Setpoint etc. **(See available programs.)** The most useful commands for an eventual user are at the beginning of the table. They allow to read from or write to devices of the secondary beam line from a data acquisition system. A **sample program in C** and a **sample program in Fortran** for test purposes are available. The last 3 commands represent a "mini ftp" facilitating backups under Windows 3.1x. They are not needed anymore under Windows XP, because Microsoft network supports all these functions.

```
      In table 1 below the following variable names are used:

      XYZ is a device name existing in file DEVICE.LIS
      nnn, mmm ... are integer numeric values (encoded as string)
      n.m represents a floating numeric value (encoded as string)
      [list] is a list of device names and numerics and looks
like:

            XYZ nnn n.m LF
            XYZ nnn n.m LF
            ..
            XYZ nnn n.m

      LF        means:  line feed character ( \n )
      [3par]    means:  low= nnn hi= nnn scale= n.m
      [4par]    means:  XYZ nnn n.m nnn
      [7par]    means:  XYZ nnn mmm a.b c.d e.f ddd
      [17par]   means:  list of all 17 parameters
      [6rates]  means:  all six counting rates (as strings)
      f         means:  camac F parameter
      N,A,F     means:  standard camac N, A and F parameters
      sta       means:  returned status of camac transfer
      sn, n     means:  station number of 6-fold scaler
      s1, s2    means:  numbers of scaler channel
```

Table 1:

Action	Send	Receive (ok)	on Error
SockTimeOut (*	TOUT nnn	*TOUT* nnn	-
ReadAllDevices	RALL	*RALL* [list]	*RALL* 0
ReadAllDacs	ALLD	*ALLD* [list]	*ALLD* 0
WriteDAC	WDAC XYZ nnn	*WDAC* XYZ= nnn	*WDAC* error
WriteDAC&Wait (**	WDAW XYZ nnn	*WDAW* XYZ= nnn	*WDAW* error
ReadDAC	RDAC XYZ	*RDAC* XYZ= nnn	*RDAC* error
ReadADC	RADC XYZ	*RADC* XYZ= nnn	*RADC* error
DeviceName	DEVN nnn	*DEVN* nnn= XYZ	*DEVN* error
ReadCamacDac	RCAD XYZ f	*RCAD* nnn	*RCAD* error
WriteCamacDac	WCAD XYZ f nnn	*WCAD* nnn	*WCAD* error
DoReadCamac	RCAM N A F	*RCAM* sta nnn	*RCAM* error
DoWriteCamac	WCAM N A F nnn	*WCAM* sta	*WCAM* error
DoControlCamac	CCAM N A F	*CCAM* sta	*CCAM* error
DeviceParameter	DEVP nnn	*DEVP* [3par]	*DEVP* error

DeviceParameter3	DEPB nnn	*DEPB* [4par]	*DEPB* error
ReserveCrate	RESC	*RESC* 1	*RESC* 0
FreeCrate	FREE	*FREE* 1	-
NewDeviceList	NEWL	*NEWL* 1	*NEWL* 0
GetPageIndex	PIND nnn	*PIND* mmm	*PIND* error
GetNumberOfPages	NPAG	*NPAG* nnn	-
ReadOnePage	RPAG nnn	*RPAG* nnn [list]	*RPAG* 0
SwitchOnCombis	SWON	*SWON* nnn	-
SwitchOnCombi	SWCO XYZ	*SWCO* XYZ 1	*SWCO* XYZ 0
SwitchOffCombi	SWOF XYZ	*SWOF* XYZ 1	*SWOF* XYZ 0
ClearScalers	CLSC sn s1 s2	*CLSC*	*CLSC* error
ReadScalers	RDSC sn s1 s2	*RDSC* nnn mmm	*RDSC* error
ReadClearAllScalers	RCAS n	*RCAS* [6rates]	*RCAS* error
GetDirectory	FDIR [mask[>f]]	[dirlist]	-
ChangeDirectory	CDIR [S][C]XYZ	*CDIR* XYZ	*CDIR* error
GetFile	FGET [@]XYZ	file(s) in defdir	*FGET* error

Table 1a: more details for Table 1.

*	If a client program looses its TCP-IP connection to the server (because of a crash or some network-overload) the used server socket (1 out of 25) remains busy and has to be recycled after some time. Thus, low time-out values around 5 min are preferred in order to keep the amount of busy sockets low. So do not change the time-out value (given in minutes) without an urgent need. If your client program has to wait for a longer time and you wish to keep the connection to the server alive then you should send every 10 - 30 secs a dummy command (e.g. "test") which will reset the used server socket's time-out counter each time like any of the commands given in table 1.
**	Waits after polarity changement until power supply is switched on again (only for Combis and Combi-likes). Time-out after 30 sec.

In order to connect your computer to the provided server functions, you have to know the name of the server and the port identification. The server names as well as the names of the clients of the different areas are given in table 2. For the currently used port identification inform yourself at the respective area. The programs which need the CAMAC server have two parameters on the command line (1st: Server name, 2nd: Port ID, see properties of the shortcuts to these programs).

Table 2: List of the client and server PC names of the 2 Target Station M and the 5 Target Station E secondary beam line areas.

Section	Area	Client	Server
Target Station M	πM1	PC299	PC235
	πM3	PC125	PC231
Target Station E	μE1	PC284	PC202
	πE1	PC484	PC130
	πE3	PC278	PC144
	μE4	PC565	PC451
	πE5	PC239	PC98

Secondary Beam Line Control System: Connection with up to 4 area servers

In some areas the number of needed magnets may exceed the number of currently installed power supplies in order to run a complex experiment. Because of budgetary restrictions and space problems on some supply galleries no additional power supplies are easily available. But it is usually possible to borrow some 3 to 5 fix-installed power supplies temporarily from a neighboring area by connecting the additional magnets via DC power cables laid over the shielding walls, and leaving the computer control still hooked to the area's default server PC. But in order to access all magnets from a single area host PC, TCP/IP connections to up to 4 area server PCs have to be established by the controlling programs on the host computer. The following PC host programs have been modified to maintain a TCP/IP dialog with up to 4 area server PCs:

- Set Point Program
- Combi Control Program
- Optima Program
- Magnet Cycling Program

The version number of these upgraded programs is 2.06 or higher. In order to tell to these programs the server name(s) the first command line parameter is either **server1+server2+server3+server4** for a 4 server connection or only **server** for a normal single server connection. (The second command line parameter is either **port1+port2+port3+port4** or simply **port** if all are the same.) The 2 expressions may be reduced to 2 or 3 servers.

Note: Of course not all devices of the second server should be visible and accessible from your host PC. In order to hide all but the borrowed devices, the Device.Dum file on the working directory (e.g. c:\optima) of your host PC has to be modified with a text editor. This file serves as a template when the programs on the host PC download the Device.Lis from the server PC(s). If the nth DUM is altered to DUMx then the information of the nth device contained in the Device.Lis of the corresponding server is

blanked out on your host. If more than one server is used, then the downloaded lists of the other servers are appended to the downloaded list of the first server. Thus, if the first list contains 30 devices and the first 20 devices of the second list should be made invisible, then the DUM entries number 31 to 50 should be modified to DUMx. (Lines in the Device.Lis files starting with '*', '-' or 'RESUNI' or alias definitions {A=B} are not counted as device definition lines. But blank lines are counted as such ones.) Of course, the Device.Dum list in the working directory of the host PC of the area which borrowed you some power supplies should also be modified in order to make devices belonging now temporarily to your area invisible for the users of the other area.
When using 2 or more servers notice also the following facts:
For the **Optima program** it is assumed, that the rate scalers on the first area server have to be used.
The page structure for the **Set Point program** is maintained: The pages of the additinal servers are appended to the pages of the first server. (Some pages may appear as being empty when enough devices are made invisible.)

Hint: Instead of connecting to a 2nd area server it is also possible to connect to a second server in the same experimental area. For example this feature is used in order to tune a µSR-apparatus via some PSI-SMK Camac motor drives connected to an own transportable PC (to be used either as a host for local tests or as a server of the corresponding area PC for remote tuning of the apparatus while the muon-beam is on).

Second, more elegant method: Instead of connecting to the same Server32.exe program via the same TCP-IP port nnn as the main user, a second instance of the Server32.exe program must be run with a different TCP-IP port mmm on the server PCxyz to which the borrowed devices are attached. But in order to have a separate device.lis (containing only the borrowed devices) being used by the parasitic user, it has to be in a subdirectory of C:\optima on the server PCxyz (e.g. C:\optima\PiE5). Important: This directory has to be declared as the startup directory in the second instance's shortcut of Server32.exe (via properties -> Shortcut -> Start in:). On the client PC of the parasitic user, in addition to the 1st command line parameter Server1+Server2+.. the second command line parameter for the 4 programs mentioned above has to be port1+port2+.. (instead of only 'port'). (The device.dum files on both machines [server and client] do not have to be edited anymore). The version numbers of the 4 programs mentioned above have to be 2.06 or higher. In order to have the second Server32.exe instance restarted automatically for the case the server PCxyz is rebooted, a shortcut with the correct parameters mentioned above has to be created in the server PCxyz's startup directory (get a hint from the corresponding shortcut of the main user's Server32.exe).

Secondary Beam Line Control System: Spare area servers and clients.

PC397 has been foreseen as a replacement PC and therefore has a multi-boot feature installed (realized with System Commander), which allows it to boot the PC with different Windows OS. It is normally used in my office as a development machine.

- Windows XP-SP2 is necessary for the operation as an area server PC. The names of the server PCs to be replaced are PC98, PC130, PC144, PC202, PC231, PC 235 and PC451. The screen resolution is 1024x768. Several programs (servers etc.) are started automatically at boot time.
- Windows XP-SP2 is used for the operation as a user client PC. The names of the user PCs to be replaced are PC125, PC239, PC278, PC284, PC299, PC484 and PC565. The screen resolution is set to 1280x1024. The most important user programs have desktop icons.

While booting the PC, you have to choose between server PC or client PC option. This selection will be remembered the next time the PC reboots or will be rebooted. Besides the already pre-installed features the following things have to be done if one of the above mentioned PCs has to be replaced. The necessary steps depend on whether the PC to replace is a server or client PC:

1. **User Client replacement:**

    - The 3 Desktop icons for “Set-Point”, “Combi-Control” and “Optima” programs have to get the proper link to your corresponding area server PC. This is done by right-clicking with the mouse the desktop icons -> Properties -> Target: Replace pc397 by the corresponding server PC name, which can be taken from table 1.
    - The saved files (status: ?) with the stored settings (*.set) may be copied from the directory C:\beamline\xyz to the directory C:\optima, where xyz means MuE1, MuE4, PiE1, PiE3, PiE5, PiM1 or PiM3.
    - The name of the PC does not have to be changed for this case. For accessing your server PC you may use the pcAnywhere program (pre-configured for all 7 server PCs, no password required).

2. **Server replacement:**

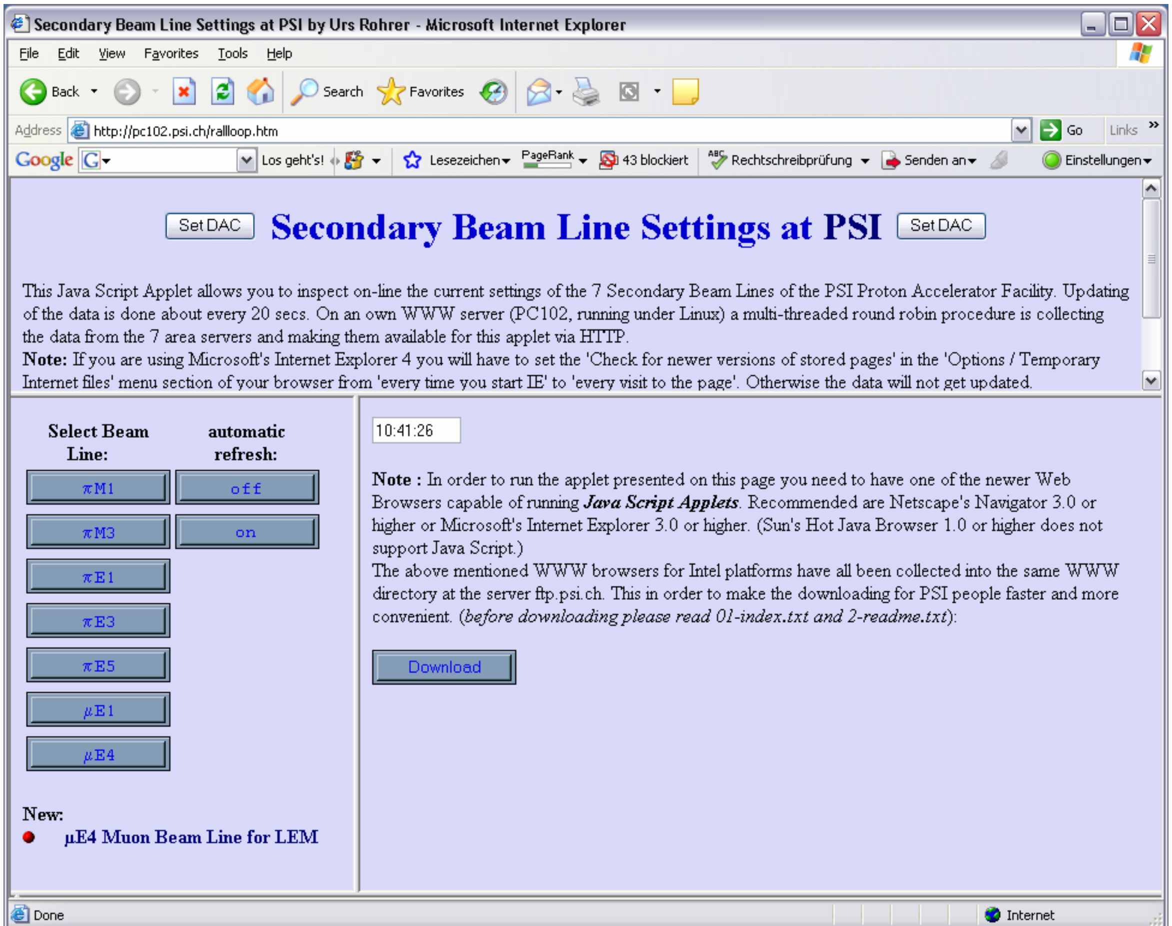
- A server PC is usually run without screen and keyboard but must be run with a connected serial mouse in order to allow remote access via pcAnywhere or VNC. All 7 area servers are mounted inside a closed cabinet together with the needed CAMAC hardware. For a short period of 1 week or so there is no need to have a replacement PC, which fits into this cabinet. So the spare PC may be placed on the floor behind or besides the cabinet with the door open for the needed cable connections. The cables to re-connect with the replacement PC are:
    1. AC 220 V cable.
    2. Ethernet RJ45 cable.
    3. RS-232 (25 poles) cable including the null-modem unit to the COM2: port.
    4. PS/2 mouse.

    5. Eventually a keyboard.
    6. Eventually a video screen.
       The last 2 items are only needed for debugging the cases where a remote login with pcAnywhere or VNC is not possible, because the boot procedure stalls before the network is up.
- The PC name and the IP address have to be changed to the name and address of the replaced server PC. The name modification is done by right-clicking the Network desktop icon -> Properties -> Identification -> Computer name: replace pc397 by the PC name of your area PC server, which can be taken from table 1. The IP address modification is done by right-clicking the Network desktop icon -> Configuration -> double-clicking TCP/IP … In the IP-Address field modify the last 2 numbers according to the corresponding IP address given in table 1.
- The proper device.lis file, which represents the magnet/slit-configuration of your experimental setup has to be put into the C:\optima directory. The different possible files may be found in C:\beamline\xyz\Server, where xyz means MuE1, MuE4, PiE1, PiE3, PiE5, PiM1 or PiM3. Eventually it has to be edited, because you may have changed it since the last backup (date ?). The newly loaded device.lis becomes only activated after a reboot of the server PC or after stopping and restarting the Set-Point program on your area PC.
- Some additional hints: Do the above mentioned software modifications in my office, where the spare PC is running (hooked-up to a screen, a mouse and a keyboard). After a reboot do not exit the Accstat program but iconize it. Do not have 2 PCs running simultaneously with the same name or IP address.

Table 1:

Area	Server name	IP-address
µE1	PC202	129.129.140.46
µE4	PC451	129.129.140.44
πE1	PC130	129.129.141.27
πE3	PC144	129.129.140.45
πE5	PC98	129.129.140.41
πM1	PC235	129.129.140.43
πM3	PC231	129.129.140.42

- **Secondary Beam Line Control System: *Home Page***
- Last updated by ***Urs Rohrer*** on 11-Oct-2006


Secondary Beam Line Settings at PSI by Urs Rohrer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://pc102.psi.ch/ralloop.htm
Go Links
Set DAC
Secondary Beam Line Settings at PSI
Set DAC
This Java Script Applet allows you to inspect on-line the current settings of the 7 Secondary Beam Lines of the PSI Proton Accelerator Facility. Updating of the data is done about every 20 secs. On an own WWW server (PC102, running under Linux) a multi-threaded round robin procedure is collecting the data from the 7 area servers and making them available for this applet via HTTP.
Note: If you are using Microsoft's Internet Explorer 4 you will have to set the 'Check for newer versions of stored pages' in the 'Options / Temporary Internet files' menu section of your browser from 'every time you start IE' to 'every visit to the page'. Otherwise the data will not get updated.
Select Beam Line:
automatic refresh:
πM1
off
πM3
on
πE1
πE3
πE5
µE1
µE4
New:
µE4 Muon Beam Line for LEM
10:41:26
Note : In order to run the applet presented on this page you need to have one of the newer Web Browsers capable of running Java Script Applets. Recommended are Netscape's Navigator 3.0 or higher or Microsoft's Internet Explorer 3.0 or higher. (Sun's Hot Java Browser 1.0 or higher does not support Java Script.)
The above mentioned WWW browsers for Intel platforms have all been collected into the same WWW directory at the server ftp.psi.ch. This in order to make the downloading for PSI people faster and more convenient. (before downloading please read 01-index.txt and 2-readme.txt):
Download
Done
Internet

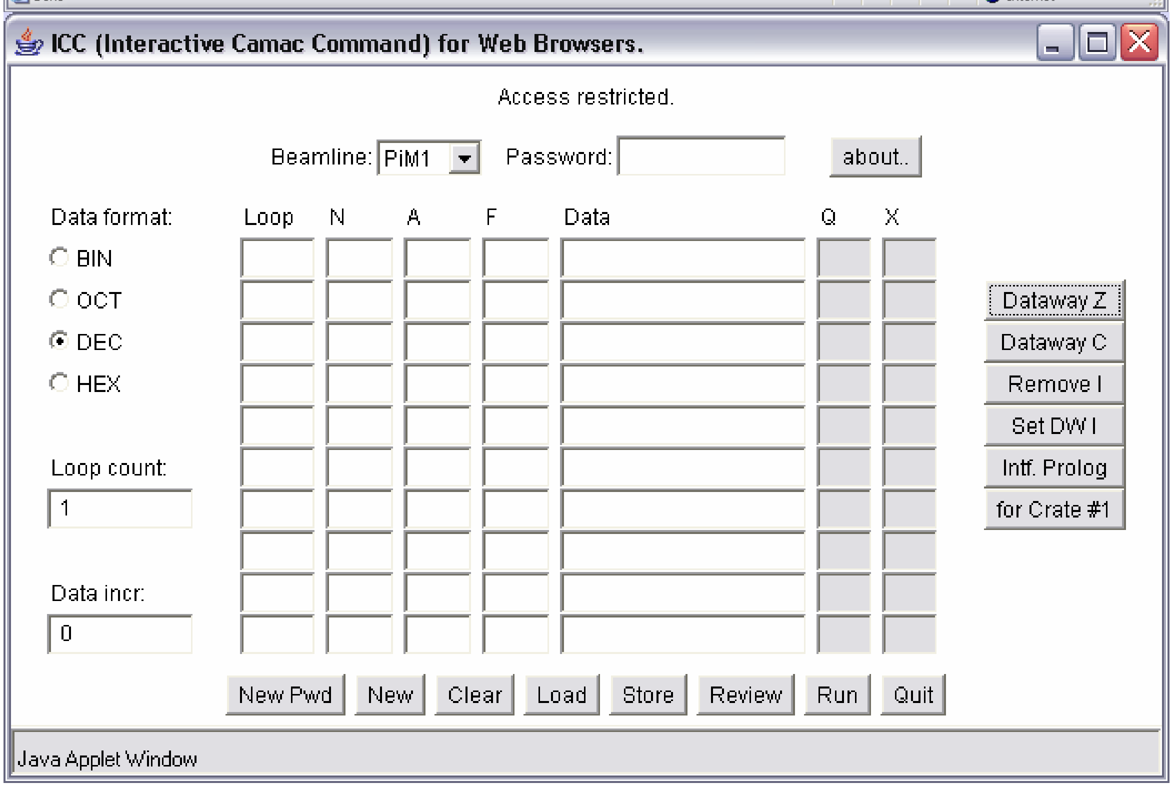

ICC (Interactive Camac Command) for Web Browsers.
Access restricted.
Beamline: PiM1
Password:
about..
Data format:
Loop
N
A
F
Data
Q
X
BIN
OCT
DEC
HEX
Loop count:
1
Data incr:
0
Dataway Z
Dataway C
Remove I
Set DW I
Intf. Prolog
for Crate #1
New Pwd
New
Clear
Load
Store
Review
Run
Quit
Java Applet Window

Fig. 1:

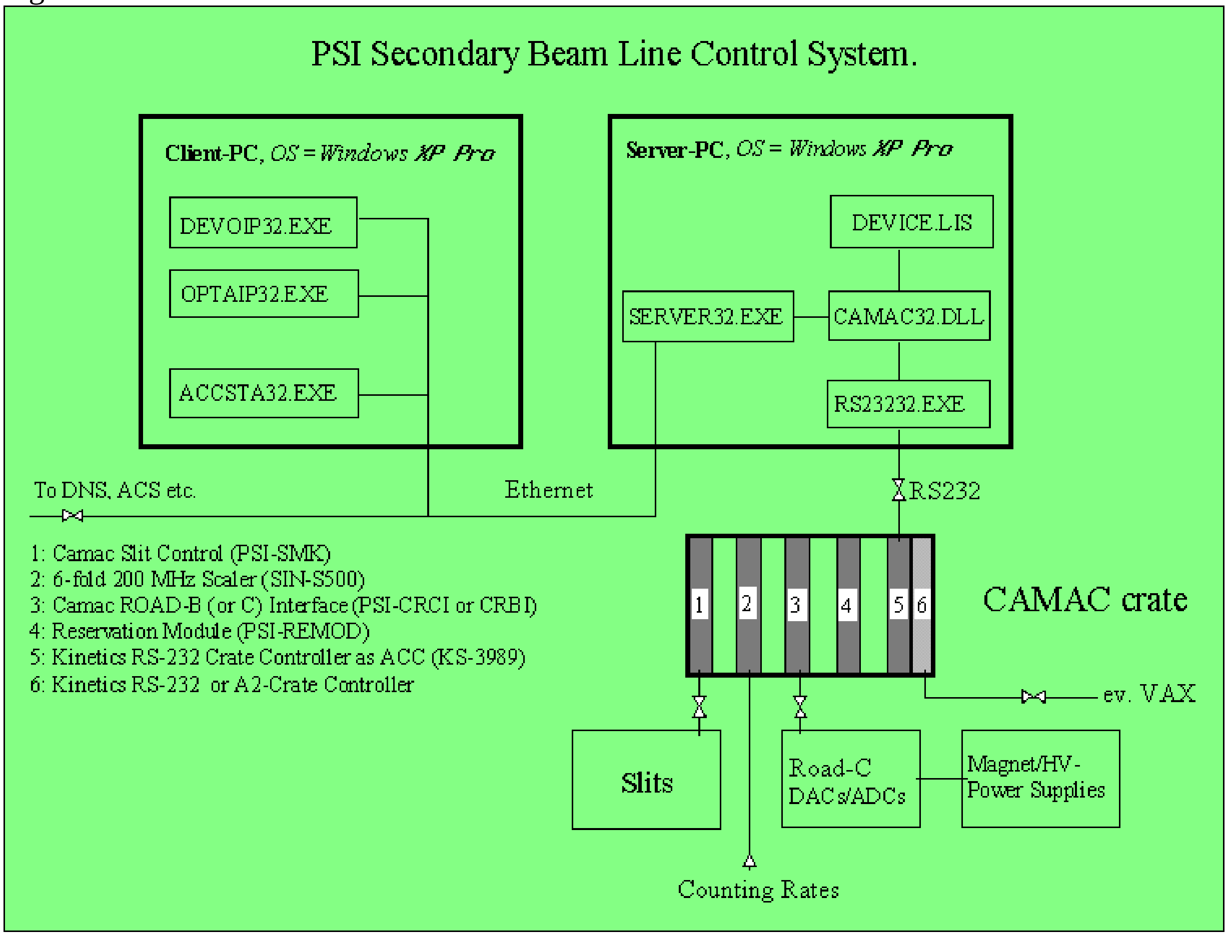
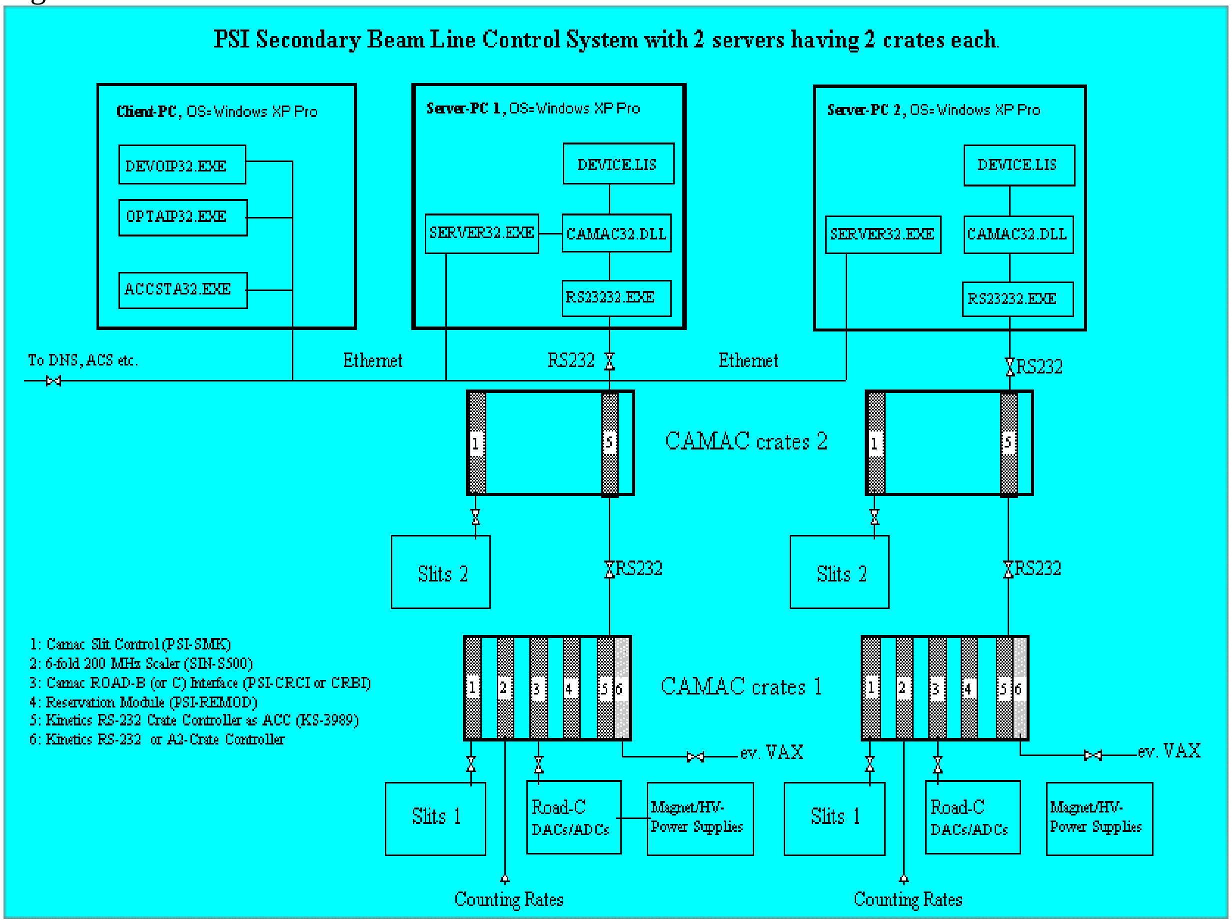

PSI Secondary Beam Line Control System.
Client-PC, OS = Windows XP Pro
DEVOIP32.EXE
OPTAIP32.EXE
ACCSTA32.EXE
Server-PC, OS = Windows XP Pro
DEVICE.LIS
SERVER32.EXE
CAMAC32.DLL
RS23232.EXE
To DNS, ACS etc.
Ethernet
RS232
1: Camac Slit Control (PSI-SMK)
2: 6-fold 200 MHz Scaler (SIN-S500)
3: Camac ROAD-B (or C) Interface (PSI-CRCI or CRBI)
4: Reservation Module (PSI-REMOD)
5: Kinetics RS-232 Crate Controller as ACC (KS-3989)
6: Kinetics RS-232 or A2-Crate Controller
1
2
3
4
5
6
CAMAC crate
ev. VAX
Slits
Road-C
DACs/ADCs
Magnet/HV-
Power Supplies
Counting Rates

Fig. 2:



PSI Secondary Beam Line Control System with 2 servers having 2 crates each.
Client-PC, OS=Windows XP Pro
DEVOIP32.EXE
OPTAIP32.EXE
ACCSTA32.EXE
Server-PC 1, OS=Windows XP Pro
DEVICE.LIS
SERVER32.EXE
CAMAC32.DLL
RS23232.EXE
Server-PC 2, OS=Windows XP Pro
DEVICE.LIS
SERVER32.EXE
CAMAC32.DLL
RS23232.EXE
To DNS, ACS etc.
Ethernet
RS232
Ethernet
RS232
CAMAC crates 2
Slits 2
RS232
Slits 2
RS232
1: Camac Slit Control (PSI-SMK)
2: 6-fold 200 MHz Scaler (SIN-S500)
3: Camac ROAD-B (or C) Interface (PSI-CRCI or CRBI)
4: Reservation Module (PSI-REMOD)
5: Kinetics RS-232 Crate Controller as ACC (KS-3989)
6: Kinetics RS-232 or A2-Crate Controller
CAMAC crates 1
ev. VAX
ev. VAX
Slits 1
Road-C DACs/ADCs
Magnet/HV-Power Supplies
Slits 1
Road-C DACs/ADCs
Magnet/HV-Power Supplies
Counting Rates
Counting Rates

Fig.3: photograph of a typically equipped camac crate
(the one of the former µE4 secondary beam line area)

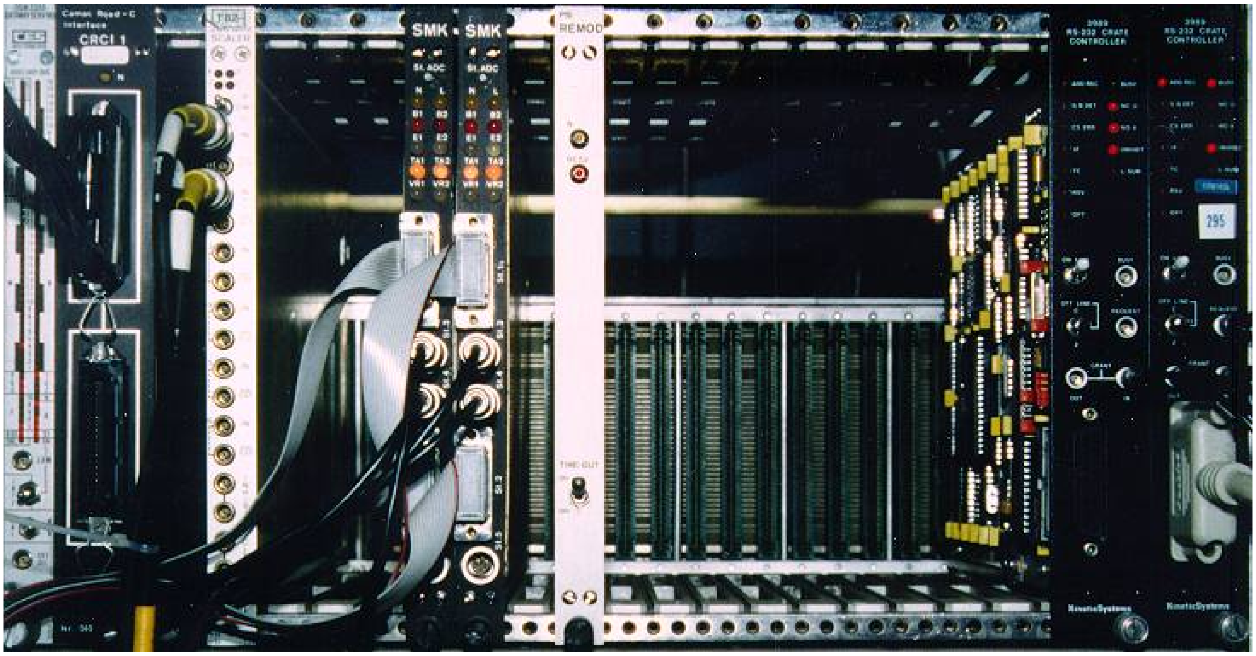
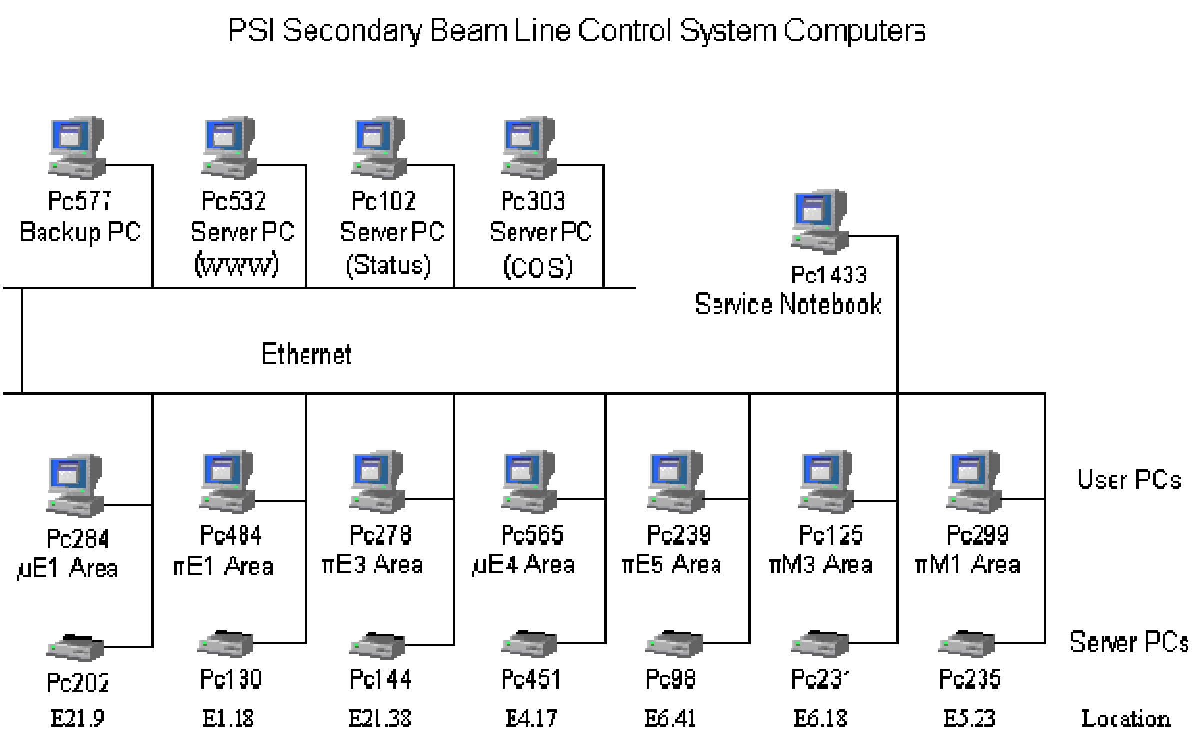


Fig 4:



Next pages contain pictures of Available Programs in List 1:

Fig.1:

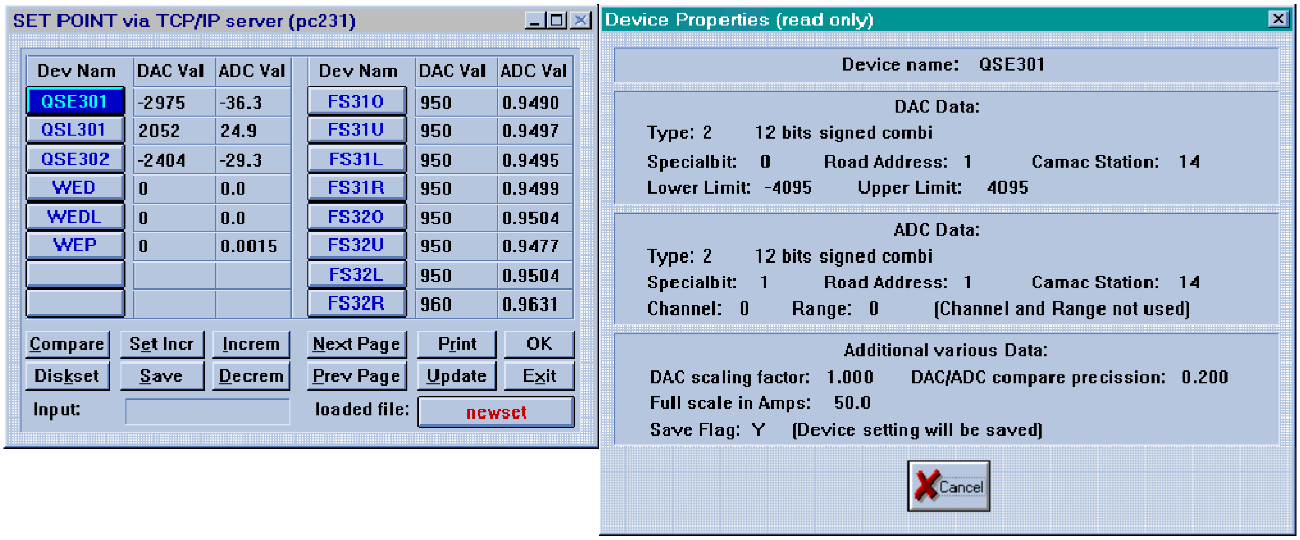

SET POINT via TCP/IP server (pc231)
Dev Nam DAC Val ADC Val Dev Nam DAC Val ADC Val
QSE301 -2975 -36.3 FS31O 950 0.9490
QSL301 2052 24.9 FS31U 950 0.9497
QSE302 -2404 -29.3 FS31L 950 0.9495
WED 0 0.0 FS31R 950 0.9499
WEDL 0 0.0 FS32O 950 0.9504
WEP 0 0.0015 FS32U 950 0.9477
FS32L 950 0.9504
FS32R 960 0.9631
Compare Set Incr Increm Next Page Print OK
Diskset Save Decrem Prev Page Update Exit
Input: loaded file: newset
Device Properties (read only)
Device name: QSE301
DAC Data:
Type: 2 12 bits signed combi
Specialbit: 0 Road Address: 1 Camac Station: 14
Lower Limit: -4095 Upper Limit: 4095
ADC Data:
Type: 2 12 bits signed combi
Specialbit: 1 Road Address: 1 Camac Station: 14
Channel: 0 Range: 0 [Channel and Range not used]
Additional various Data:
DAC scaling factor: 1.000 DAC/ADC compare precission: 0.200
Full scale in Amps: 50.0
Save Flag: Y [Device setting will be saved]
Cancel

Fig. 2:

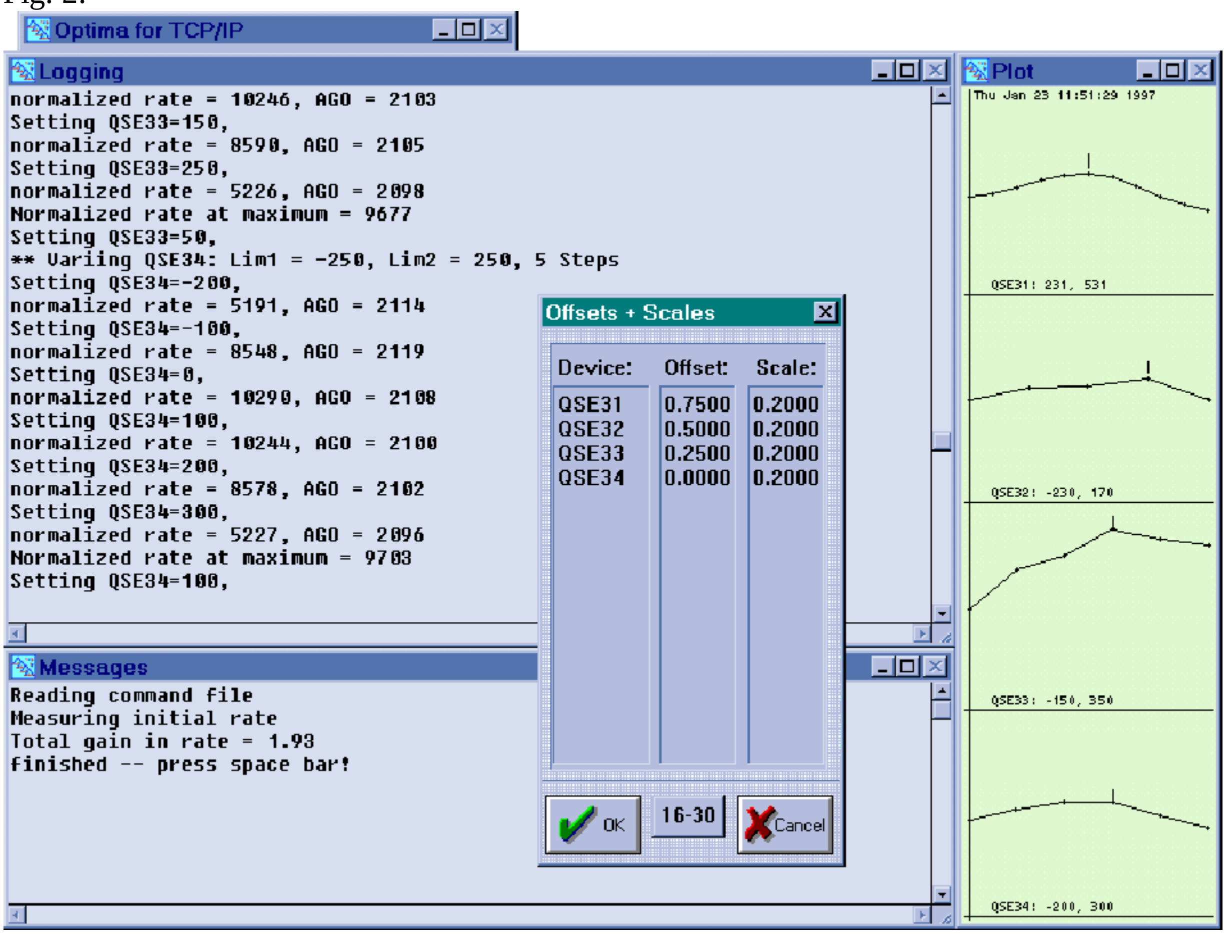
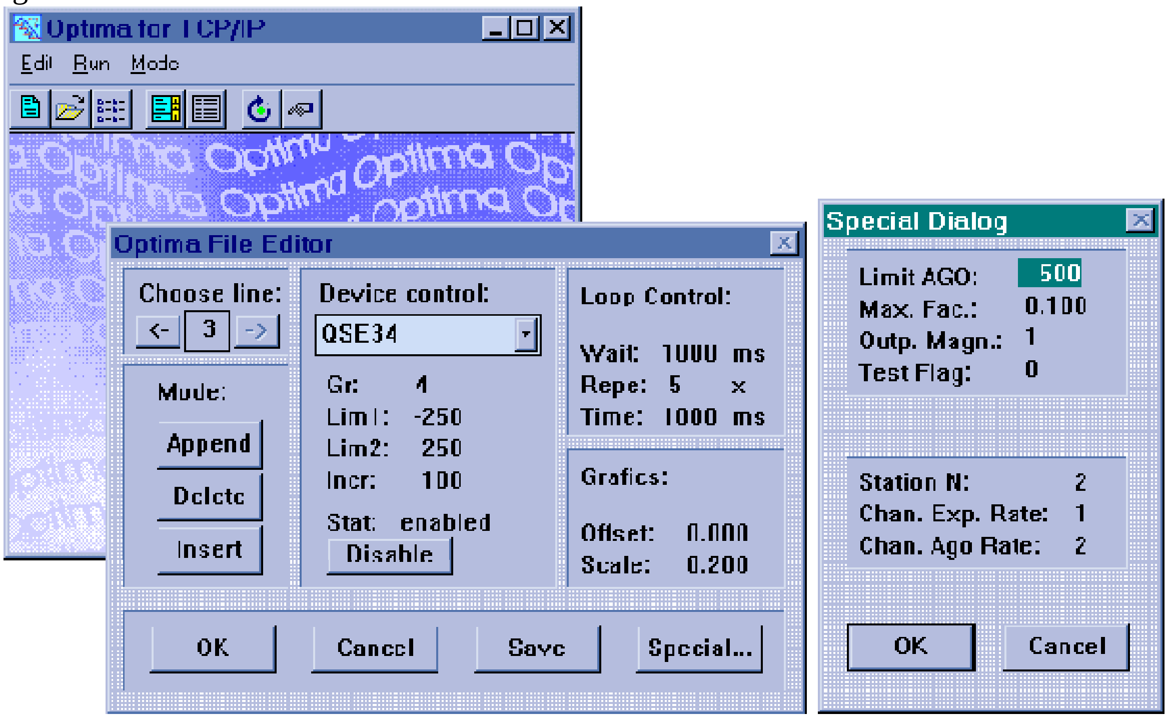
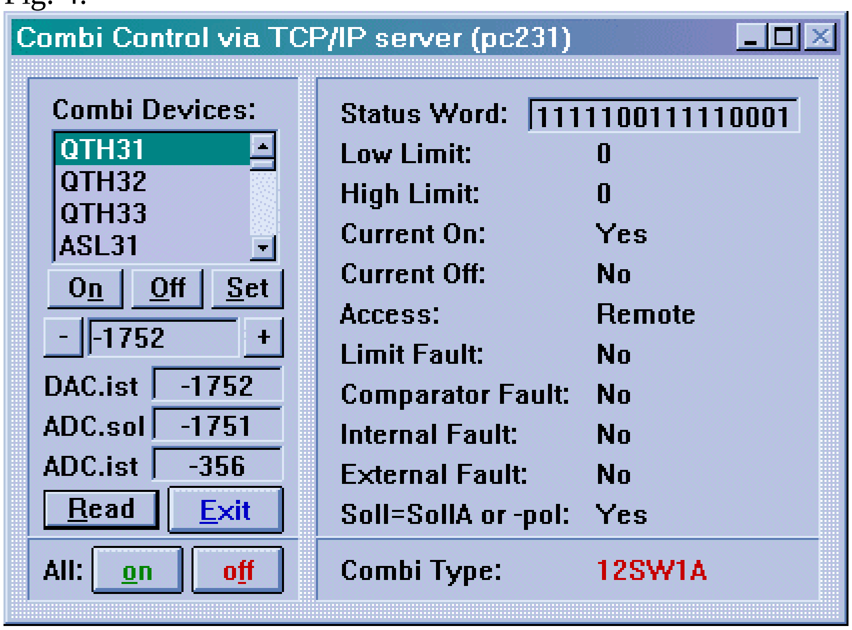

Optima for TCP/IP
Logging
normalized rate = 10246, AGO = 2103
Setting QSE33=150,
normalized rate = 8590, AGO = 2105
Setting QSE33=250,
normalized rate = 5226, AGO = 2098
Normalized rate at maximum = 9677
Setting QSE33=50,
** Variing QSE34: Lim1 = -250, Lim2 = 250, 5 Steps
Setting QSE34=-200,
normalized rate = 5191, AGO = 2114
Setting QSE34=-100,
normalized rate = 8548, AGO = 2119
Setting QSE34=0,
normalized rate = 10290, AGO = 2108
Setting QSE34=100,
normalized rate = 10244, AGO = 2100
Setting QSE34=200,
normalized rate = 8578, AGO = 2102
Setting QSE34=300,
normalized rate = 5227, AGO = 2096
Normalized rate at maximum = 9703
Setting QSE34=100,
Messages
Reading command file
Measuring initial rate
Total gain in rate = 1.93
finished -- press space bar!
Offsets + Scales
Device: Offset: Scale:
QSE31 0.7500 0.2000
QSE32 0.5000 0.2000
QSE33 0.2500 0.2000
QSE34 0.0000 0.2000
OK 16-30 Cancel
Plot
Thu Jan 23 11:51:29 1997
QSE31: 231, 531
QSE32: -230, 170
QSE33: -150, 350
QSE34: -200, 300

Fig. 3:



Optima for TCP/IP
Edit Run Mode
Optima File Editor
Choose line:
<- 3 ->
Device control:
QSE34
Loop Control:
Wait: 1000 ms
Repe: 5 x
Time: 1000 ms
Mode:
Append
Delete
Insert
Gr: 4
Lim1: -250
Lim2: 250
Incr: 100
Stat: enabled
Disable
Grafics:
Offset: 0.000
Scale: 0.200
OK
Cancel
Save
Special...
Special Dialog
Limit AGO: 500
Max. Fac.: 0.100
Outp. Magn.: 1
Test Flag: 0
Station N: 2
Chan. Exp. Rate: 1
Chan. Ago Rate: 2
OK
Cancel

Fig. 4:



Combi Control via TCP/IP server (pc231)
Combi Devices:
QTH31
QTH32
QTH33
ASL31
On
Off
Set
-1752
DAC.ist -1752
ADC.sol -1751
ADC.ist -356
Read
Exit
All: on off
Status Word: 1111100111110001
Low Limit: 0
High Limit: 0
Current On: Yes
Current Off: No
Access: Remote
Limit Fault: No
Comparator Fault: No
Internal Fault: No
External Fault: No
Soll=SollA or -pol: Yes
Combi Type: 12SW1A

Fig. 5:

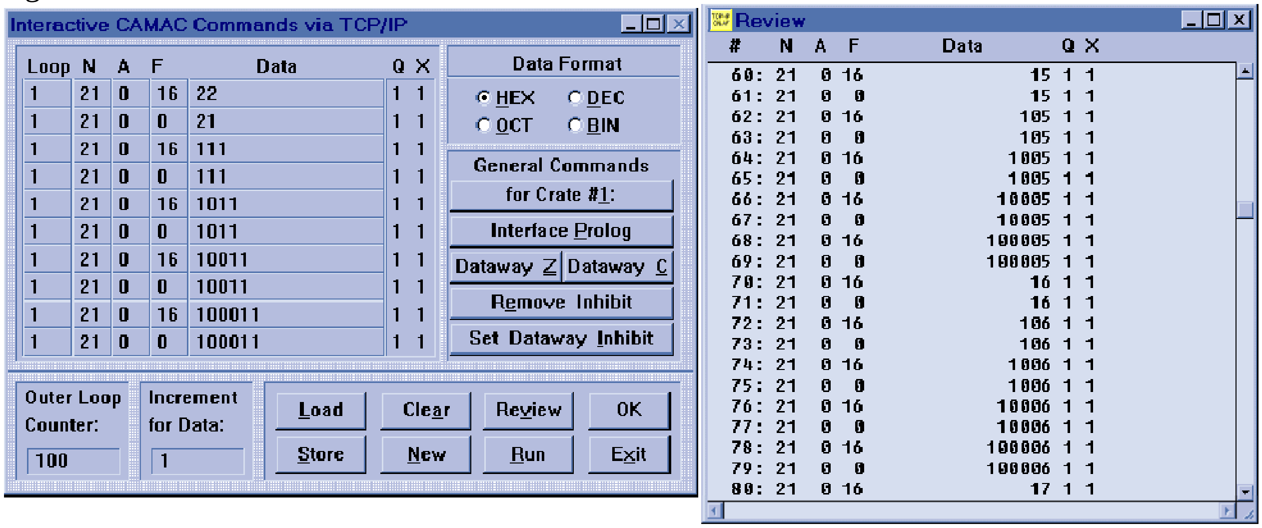

Interactive CAMAC Commands via TCP/IP
Loop N A F Data Q X
1 21 0 16 22 1 1
1 21 0 0 21 1 1
1 21 0 16 111 1 1
1 21 0 0 111 1 1
1 21 0 16 1011 1 1
1 21 0 0 1011 1 1
1 21 0 16 10011 1 1
1 21 0 0 10011 1 1
1 21 0 16 100011 1 1
1 21 0 0 100011 1 1
Data Format
HEX DEC
OCT BIN
General Commands
for Crate #1:
Interface Prolog
Dataway Z Dataway C
Remove Inhibit
Set Dataway Inhibit
Outer Loop Counter:
100
Increment for Data:
1
Load Clear Review OK
Store New Run Exit
Review
N A F Data Q X
60: 21 0 16 15 1 1
61: 21 0 0 15 1 1
62: 21 0 16 105 1 1
63: 21 0 0 105 1 1
64: 21 0 16 1005 1 1
65: 21 0 0 1005 1 1
66: 21 0 16 10005 1 1
67: 21 0 0 10005 1 1
68: 21 0 16 100005 1 1
69: 21 0 0 100005 1 1
70: 21 0 16 16 1 1
71: 21 0 0 16 1 1
72: 21 0 16 106 1 1
73: 21 0 0 106 1 1
74: 21 0 16 1006 1 1
75: 21 0 0 1006 1 1
76: 21 0 16 10006 1 1
77: 21 0 0 10006 1 1
78: 21 0 16 100006 1 1
79: 21 0 0 100006 1 1
80: 21 0 16 17 1 1

Fig. 6:

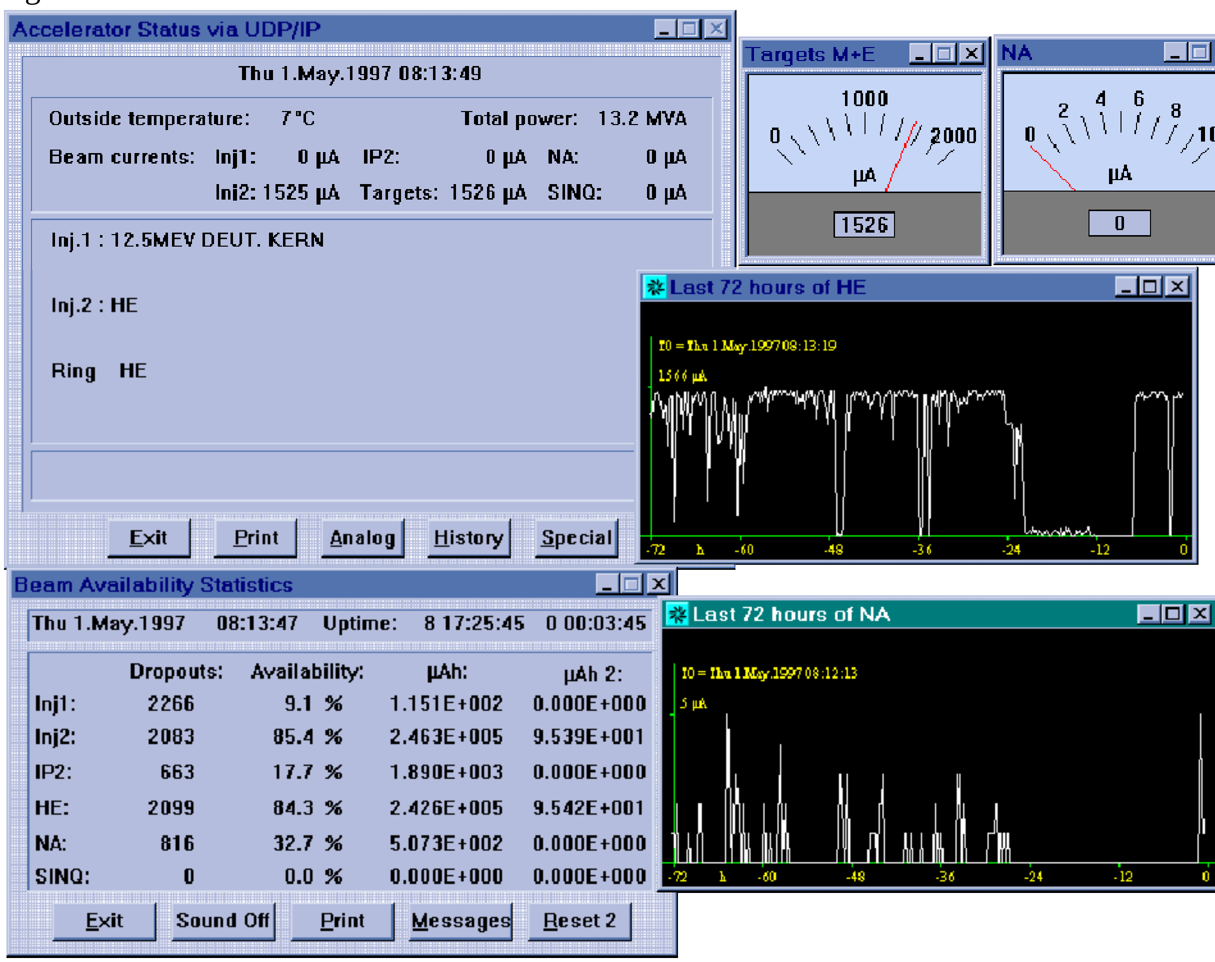

Accelerator Status via UDP/IP
Thu 1.May.1997 08:13:49
Outside temperature: 7°C Total power: 13.2 MVA
Beam currents: Inj1: 0 µA IP2: 0 µA NA: 0 µA
Inj2: 1525 µA Targets: 1526 µA SINQ: 0 µA
Inj.1 : 12.5MEV DEUT. KERN
Inj.2 : HE
Ring HE
Exit Print Analog History Special
Targets M+E
1000
0
2000
µA
1526
NA
2 4 6 8
0
10
µA
0
Last 72 hours of HE
T0 = Thu 1 May 1997 08:13:19
1566 µA
-72 h -60 -48 -36 -24 -12 0
Beam Availability Statistics
Thu 1.May.1997 08:13:47 Uptime: 8 17:25:45 0 00:03:45
Dropouts: Availability: µAh: µAh 2:
Inj1: 2266 9.1 % 1.151E+002 0.000E+000
Inj2: 2083 85.4 % 2.463E+005 9.539E+001
IP2: 663 17.7 % 1.890E+003 0.000E+000
HE: 2099 84.3 % 2.426E+005 9.542E+001
NA: 816 32.7 % 5.073E+002 0.000E+000
SINQ: 0 0.0 % 0.000E+000 0.000E+000
Exit Sound Off Print Messages Reset 2
Last 72 hours of NA
T0 = Thu 1 May 1997 08:12:13
5 µA
-72 h -60 -48 -36 -24 -12 0

Fig. 7:

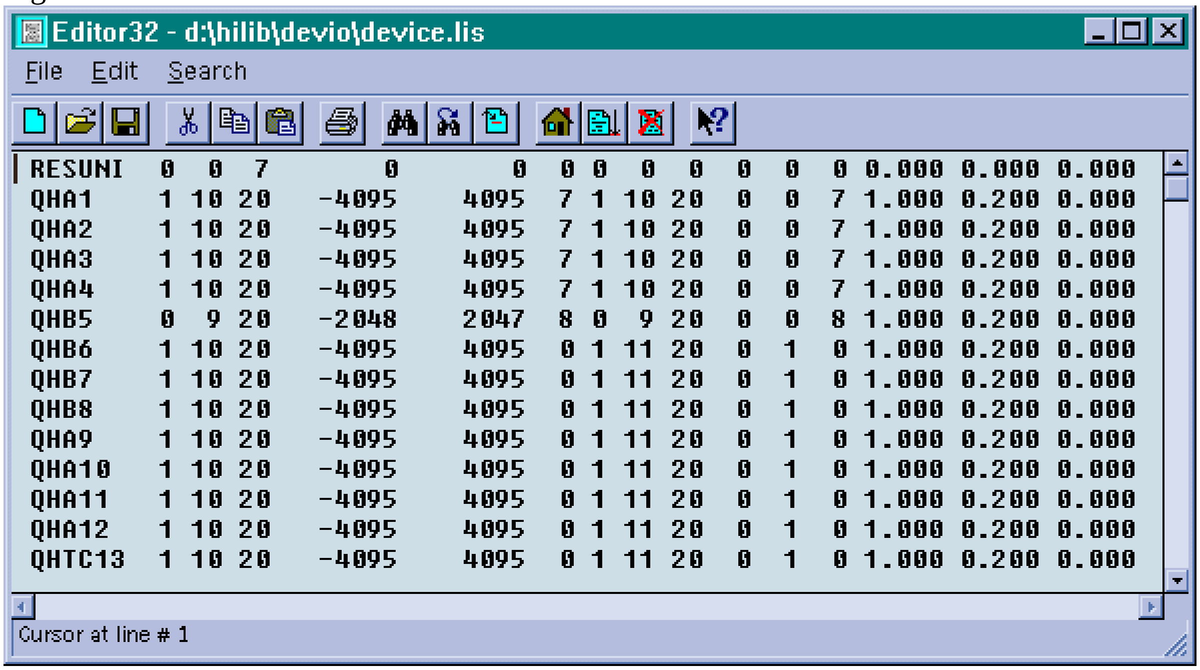


Fig. 8: equal to Fig.1 from above

Fig. 9:

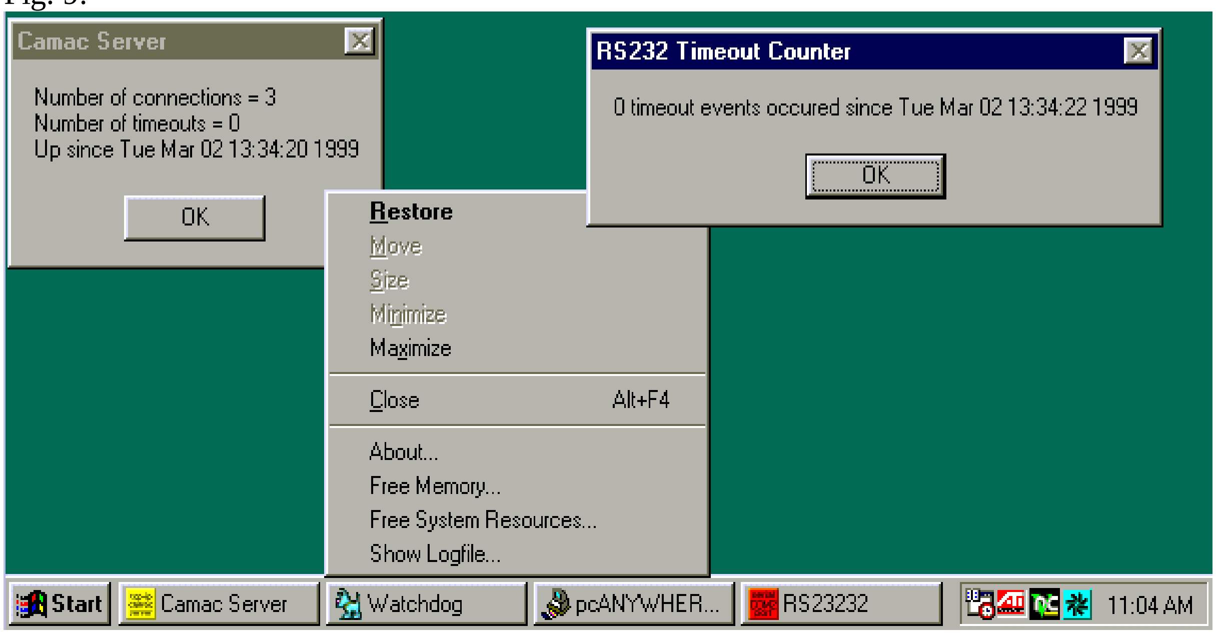
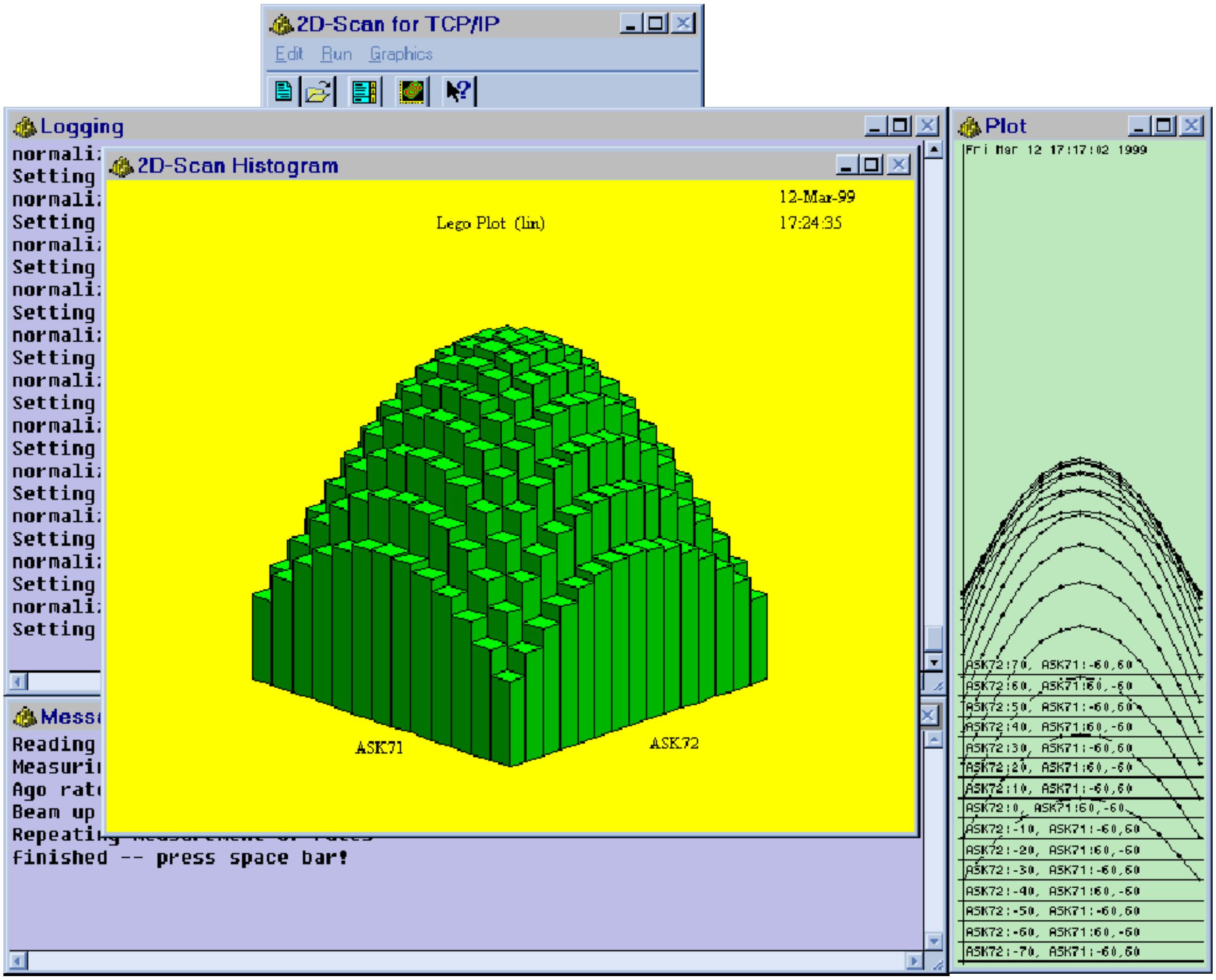


Fig. 10:

Macy via TCP/IP server (pc144)
Edit Run
Magnet Cycling Report
140 sec: setting ASK72 to 2348
140 sec: ASK72= 0.9998
145 sec: ASK72= 0.9231
150 sec: ASK72= 0.8418
155 sec: ASK72= 0.7592
160 sec: ASK72= 0.6762
165 sec: ASK72= 0.5934
170 sec: ASK72= 0.5739
175 sec: ASK72= 0.5734
180 sec: setting ASK72 to 3131
180 sec: ASK72= 0.5722
185 sec: ASK72= 0.6488
190 sec: ASK72= 0.7321
195 sec: ASK72= 0.7641
200 sec: setting ASK72 to 2585
200 sec: ASK72= 0.7643
205 sec: ASK72= 0.6891
210 sec: ASK72= 0.6137
215 sec: ASK72= 0.6117
220 sec: setting ASK72 to 2609
220 sec: ASK72= 0.6117
finished -- press space bar!
Time Charts
ASK72

Fig. 11:



The next 3 pages show the first Control System from 1988

Secondary Beam Line Computer Control at PSI

Urs Rohrer, Paul Scherrer Institut, CH-5232 Villigen PSI, 1988

In 1987 almost every secondary beam line at PSI was still controlled by a hardwired ROAD-B command station with no built-in intelligence. During the year of 1988 these difficult to maintain command stations have been successively replaced by Atari ST computers (MC68000 processor, 8 MHz, 1 MB RAM, ROM-TOS, 720 KB floppy drive) interfaced via the modem port (19200 baud) to some new camac hardware (see Fig. 1). As interface between the computer and the CAMAC dataway serves a CES-2180 ACC (Starburst) with CATY on ROM as driving software. The ROAD-C DACs and ADCs of each cluster of magnet power supplies are interfaced via a subgroup master (SGM) and a ROAD-B highway cable with a special interface (CRBI) to a CAMAC crate. All ROAD hardware has been developed by the Accelerator Control Group.

Whereas most of the user software for the Atari ST has been coded in Fortran, some routines and utilities were written in Assembler, Basic or C. All programs fit well onto one single micro floppy disk. During the cold start procedure, the most important programs are copied onto a RAM disk of 500 KB size. Therefore, there is no need for a hard-disk which makes the whole computer control very handy and easy to transport into the barrack of the physics experimentalist.

All programs are data file driven, so changes to the hardware configuration are quickly transferred to the software by just editing files without recompilation. The programs, which also make full usage of the user friendly GEM software layers (WIMP human interface), may be divided up into 3 different groups:

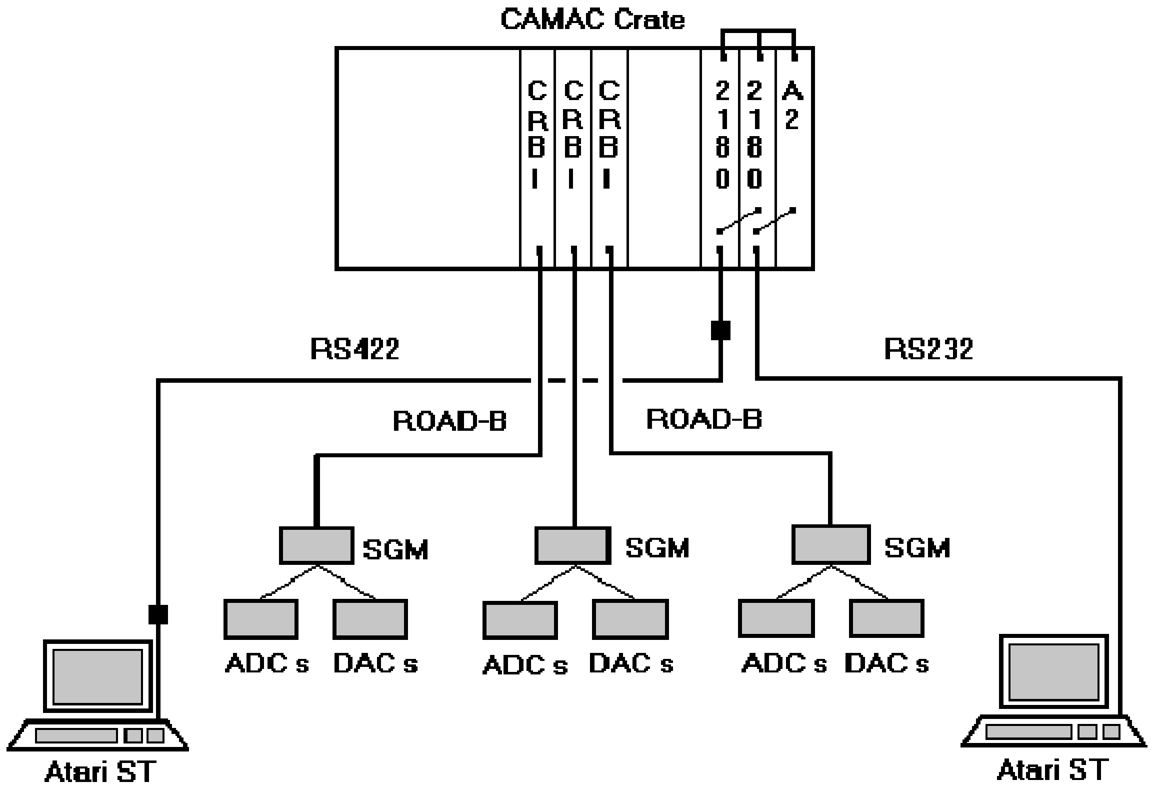


Figure 1: Block diagram of the new computer control for the 2 coupled areas µE1 and π E1. Each area has its own Atari ST computer. For distances longer than 30 m between computer and CAMAC crate, RS232-to-RS422 boosters are used. The ROAD bus system could be taken over almost unmodified from the former command stations.

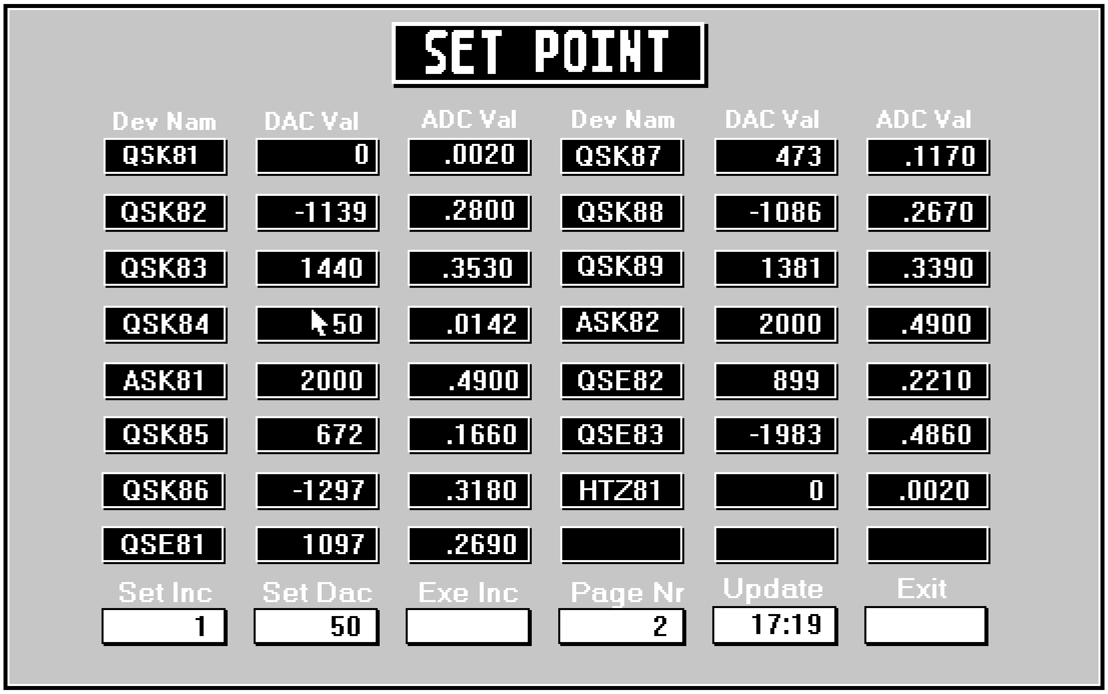


Figure 2: A typical screen output (monochrome, 640x400 pixels) produced by the Set Point program shows some actual settings of the µE1 magnets. The usage of the handy mouse allows the physicist to access the program's whole functionality in an easy way.

- Set-point program for individual control of all devices including saving current settings on or restoring old values from disk files. (see figure 2)
- Rate optimizing program which allows the user to bring up his experimental rate automatically to a maximum by executing a command file containing a list of the magnet elements the settings of which have to be adjusted.
- Transport program package (CERN/SLAC/FERMILAB version) which allows the experienced beam physicist to study or to debug online new optical modes or new beam lines. The DAC settings are readable or settable directly from the programs and converted via tables into physical units.

Up to now this computer control has been tested and installed on 4 different beam lines (πM3, πE3, µE1 and πE1) and was found to be working reliably. (Because µE1 and πE1 beam lines have 4 common magnets, they are controlled from a single common CAMAC crate containing 2 daisy-chained CES-2180 ACCs with 2 separate Atari ST computers, one for each area.) In case of a rare system crash or hang-up, pressing one button boots the computer within 30 seconds. This computer control helps to bring up a beam line in a much shorter period of time than previously, and makes it easier to control the magnets during a data taking run, because among other things it alerts the user if a device becomes faulty.