BRAHMS High Voltage

Claus Jørgensen[*]
Niels Bohr Institute
University of Copenhagen

July 30, 1999

This note describes the high voltage system for the BRAHMS experiment at RHIC. The first chapter is a short manual to the preliminary control system BRAHVO. In the second chapter the hardware setup is described, and the third chapter contains a short discussion on how a more complete and elaborate system could be build.

---

[*]E–mail: <ekman@nbi.dk>

Contents

1 Controlling High Voltage in BRAHMS

The high voltage is controlled by running the `brahvo.perl` script on the O-PUS machine (`/home/slowc/hv/pro/brahvo.perl`), just type `perl brahvo.perl`. BRAHVO is an acronym for BRAhms High VOltage.

1.1 The main window

When the program is executed following window will appear:
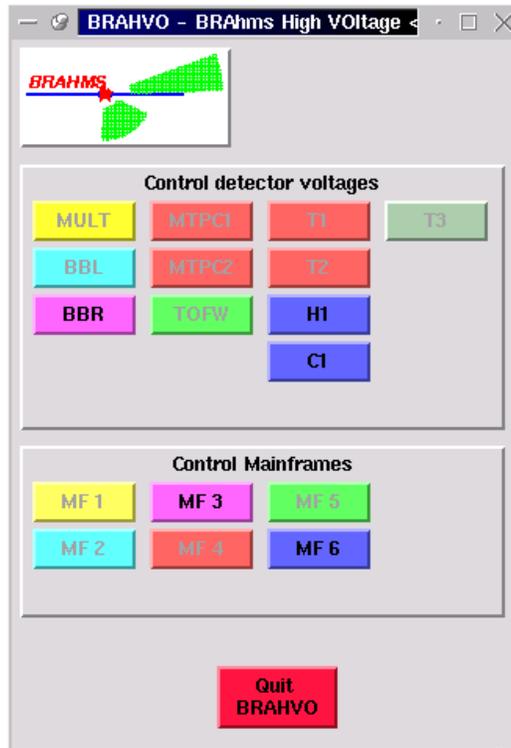
From this main window it is possible to open a control window either for a detector or for a high voltage mainframe (Notice that not all detectors and mainframes can be controlled in the present version of BRAHVO. Currently July 30, 1999the following detectors are implemented: MULT, BBR, BBL, H1, C1). When a control window is opened a connection to the mainframe is established. Until the window is closed it is not possible to open any of the other control windows. The colors of the buttons indicate which detectors are supplied by which mainframes.

1.2 The detector control window

The figure shows an example of a control window for a detector (H1 has 40 scintillator slats, each with two PM tubes).
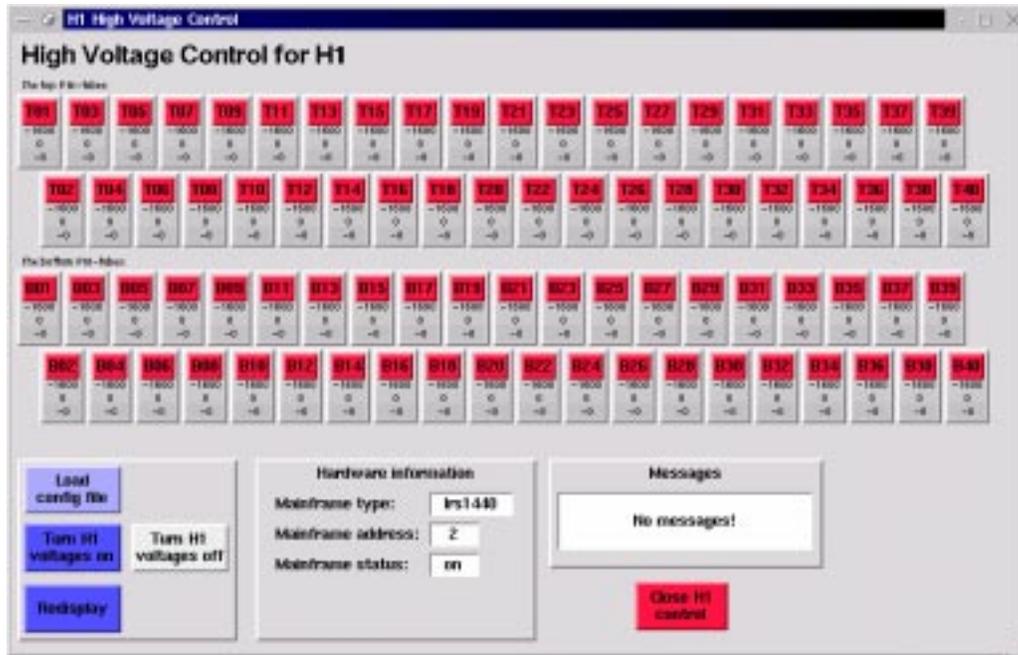
Each high voltage channel is shown in a little frame containing a button with a label and three voltages. The function of the label button will be described later. The first of the three voltages is the default voltage, which is read from a configuration file. The middle one is the mainframe demand voltage, i.e. the voltage that the mainframe will put on the specific channel (if the high voltage is on). Notice that the demand is the same whether the mainframe is on or off. The last of the three displayed voltages is the actual voltage, that is measured by the mainframe. The default voltage is therefore a software value, written in the configuration file, while the demand voltage and the actual voltage are values in the memory of the mainframe.

When the red button in the right bottom of the detector control window is pressed both the connection to the mainframe and the control window is closed.

2

1.2.1 The control buttons

In the right bottom corner of the detector control window is four control buttons:

Load config file When the Load button is pressed the configuration file will be re-read and the default voltages will be displayed. A new functionality will perhaps added later, allowing the user to load default voltages from different configuration files.

Turn HV on When the Turn on button is pressed the demand voltages will be slowly turned up to the default voltage. This should only be done when the mainframe is turned on, and unfortunately it is a relatively slow process (approximately 3 minutes for H1). Turning the mainframe on when the demand voltage is high will make the actual voltage go up with a very high ramp speed, which could damage the PM tubes.

Turn HV off When the Turn off button is pressed the demand voltages will be turned down slowly.

Redisplay When the Redisplay button is pressed the configuration file is re-read, the demand and actual voltages are read from the mainframe and the new values are displayed on the screen. Notice that a change in the voltages is not automaticly shown on the screen.

Perhaps a save button will be added at some point. The function of such a button should be to save the current demand voltages in a new configuration file. It should then be possible to load these voltages (as default voltages) at a later point. How the demand voltages is changed is described in the next section.

1.2.2 The label button

As previously mentioned each high voltage channel has a label. For example the channel supplying the top PM tube on the first slat of H1 has the label T01 (For the MULT and the BB detectors the label runs from 1 to 32 corresponding to the mainframe channels from 0 to 31). This label is displayed on the label button. The label button can in different states: grey, green or red. It is grey when the mainframe is turned off, green when the mainframe is on

and the difference between the default and the actual voltage on the channel does not exceeds 10 V, and red indicating that the the difference between the actual voltage and the default voltage is greater than 10 V. When this label button is pressed a window with further information on the channel will appear.

In this window it is also possible to write a new demand voltage to the mainframe.
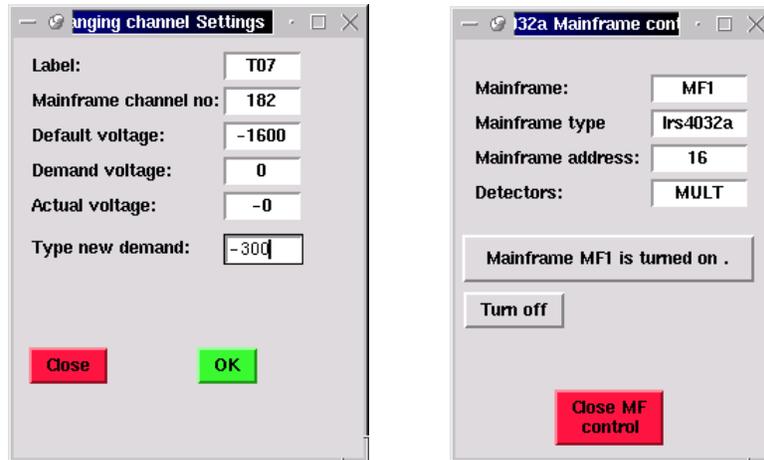
Figure 1: When the label button is pressed, a window with information on the corresponding HV channel appears (left). The mainframes can be turned on and off by pressing the on/off button in the mainframe control window (right).

1.2.3 Hardware information and messages

The hardware information frame in the bottom of the detector window gives brief information on the hardware setup for the detector, and the message line tells the user what process in running.

1.3 The mainframe control window

The mainframe control window has only two functions. To show which detectors the mainframe supplies (which is already shown by the colors of the buttons in the main window) and to make it possible to turn on and off the mainframe.

4

Remember that some mainframes supply more than one detector. Before tuning on or off a mainframe, check that all the demand voltages on every detector supplied by the mainframe is zero.

1.4   The configuration files

There should be one configuration file for all the mainframes and one for each detector. The configuration files is in `/home/operator/hv/config`. When the script reads the configuration file it looks for lines that start with the first two letters in the detectors name (three for the beam-beam counters as an extra precaution). For example in the H1 configuration file only lines that start with H1 will be read. This is an example of a line in the H1 configuration file:

```
H1T05                   4                -1600
```

First field is the detector component label (H1T05 is the top PM tube on the fifth slat of H1), then the mainframe high voltage channel (the first channel is channel 0), and then the voltage, that is default voltage. The user can thus write what he or she want on all the other lines, but only the label and the two numbers on the lines that should be read. The program does not check if a high voltage channel or a label is displayed twice.

Control of the Lecroy 1458 mainframe is not included in BRAHVO yet, but controlling it can be done using its control/display program vt100. Just telnet to `tsir 3016` and write `vt100`.

The BRAHVO programs are not completely debugged and some changes will probably occur in the nearest future.

2 The hardware setup

The detector elements in BRAHMS are supplied with high voltage from three different Lecroy HV mainframe types: the 1458, the 1440 and the 4032a. The 1458 and 1440 mainframes are connected individually to the terminal server and can be controlled with commands written directly to the terminal server ports. The 4032a's are connected in a daisy chain and are controlled via a Lecroy 2132 CAMAC interface.
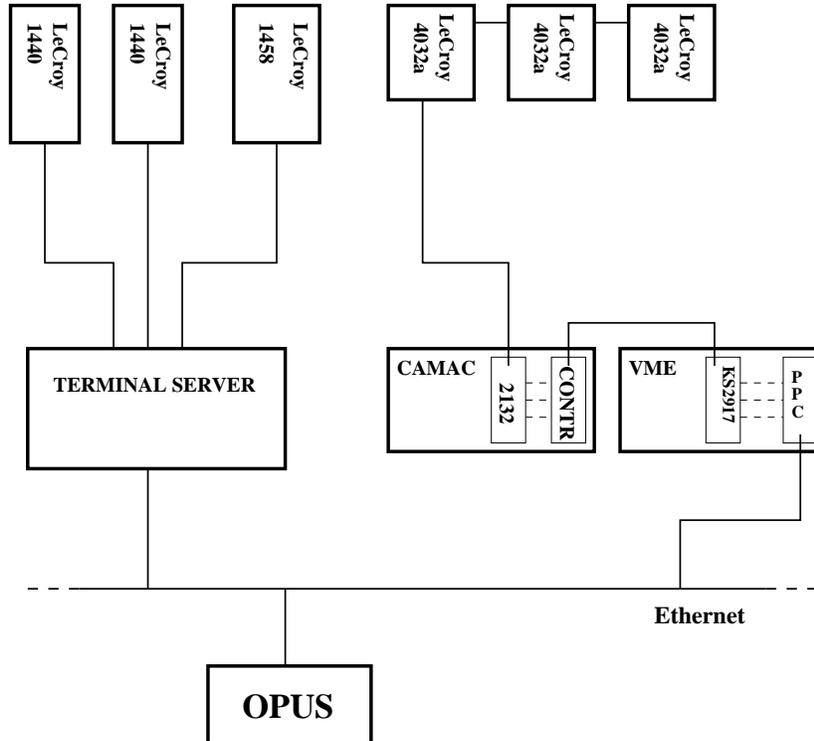
Figure 2: The 1440 and the 1458 mainframes are controlled via the terminal server, while the 4032a mainframes are controlled via CAMAC/VME.

Lecroy 1440 The 1440 mainframe has room for 16 cards with 16 HV channels each (output voltage from 0 to -2500 V). The channels are numbered 0 to 255 counting vertically from the upper right corner (empty slots are counted!).

6

Lecroy 1458   The 1458 mainframe has room for 14 cards in different configurations. At present we have 1461N, 1471N and 14771P's.

Lecroy 4032a   The 4032a mainframe has room for 8 pods with 4 HV channels each (output voltage from 0 to 3.3 kV).

In the table below information on the high voltage control for each detector is displayed.

Detector	MF	type	TS-port/Addr.	config file	contact
MULT	MF1	4032a	- / 16	`MULTsettings`	Y. K. Lee
BB Right	MF3	4032a	- / 1	`BBRsettings`	B. Budick
BB Left	MF2	4032a	- / 2	`BBLsettings`	B. Budick
TPC1,TPC2	MF4	1458	3016 / -	-	FV / RD
TOFW	MF5	1440	3015 / 1	`TOFWsettings`	D. Beavis
T1,T2	MF4	1458	3016 / -	-	FV / RD
H1	MF6	1440	3014 / 2	`H1settings`	I. Bearden
C1	MF6	1440	3014 / 2	`C1settings`	R. Debbe

3 Things to consider

First of all we have to discuss what we require from a high voltage system, what we would like and how this could be established within the limits of the present hardware and our human resources. We need a system that has the following features:

- Remote control. We need to be able to control each high voltage channel separately, and to keep track of which high voltage channel supplies which component in which detector.

- Logging. We need to read out all the voltages to a log file a not-yet-specified number of times during each hour when the experiment is running.

- Alarm. We need to monitor all the channels continuously and to be alarmed if something is not how it's supposed to be.

It would be nice to have these three features in one smooth, easy to use GUI. The first problem is whether we want to use EPICS or not.

The problem with using EPICS is that another major software package is entering the BRAHMS universe. EPICS is able to do a lot more than we need, and is thus very complicated. A good high voltage system (based on the existing, preliminary version of BRAHVO) could probably be build in less time than it would take to make EPICS run.

The present version of BRAHVO is written in Perl/Tk and it has only the first of three features we want. It is possible to control some of the detectors (BB, MULT, H1, C1), and routines that can control the rest of the detectors can be added without problems. The high voltage control for the TPC's requires some work because the Lecroy 1458 mainframe is used to supply these detectors. The 1458 can set individual ramp speed and trip current limits for each channel which we should be able to control with our GUI. It could also be discussed whether we want to connect to the 1458 via the terminal server or if we want to invest in the faster (1450-ET) Ethernet interface.

The logging and the alarm features are somewhat more difficult to implement in a Perl script because they - or at least the alarm feature - require a process running continuously in the background. It could be done by running an independent script with no connection to the GUI. The only consideration

8

is how to prevent the two scripts from connecting at the same time, and how to make sure that the logging/alarm script is running all the time when the experiment is running.

A third scheme is to build the system in C++ using the Qt graphics package. This would require new coding, but the control, the logging and the alarm features could be included in the same GUI. I think the C++ solution will give the best result: one program that has all the features. The disadvantage of C++ compared to Perl is that code is harder to maintain.

If you have any comments or ideas how to improve the control system please don't hesitate to contact me.

Claus Jørgensen
`<ekman@nbi.dk>`

July 30, 1999
Brookhaven National Laboratories

9