High voltage control software for Miniball

Contents

[1] Nigel Warr March 2007

1 Introduction

The original Miniball high voltage software was written by Heiko Scheit then of MPI Heidelberg to run on a LeCroy HV mainframe. This mainframe proved unreliable so we bought a CAEN mainframe and Heiko adapted his software. This worked fine for several years.

However, in 2006, the CERN network people started a new kind of network scan, supposedly to enhance security by breaking insecure systems before some cracker did it. Our system proved vulnerable and the CERN network people ruled that using it was a violation of CERN network security rules. As a result the software was redesigned from scratch using a different philosophy.

The old software used a server and a client communicating via IP sockets. This had the advantage that the server and client didn't necessarily have to be on the same computer. In practise, however, we only ever used this at GSI, since at CERN both the server and the client always ran on the autofill computer and people would log into that computer to start the client, even if they were working from another computer. The problem was that there was no mechanism for logging in and all connections on the appropriate port were accepted without authorisation. This is not permitted by CERN rules. It also meant that anyone with access to any computer within the CERN firewall (including people who have cracked other CERN systems, or viruses on CERN computers) had full access to the HV control. It would not have been possible to change this without some significant modification.

The new software, described in this document, also uses a server/client philosophy, but the communication is done via a unix pipe from the client to the server and via files written to disk in the other direction. No ports are open, so a cracker must actively break into the computer hosting the software using some other security hole, before he or she can access the HV software.

2 Overview of the HV control software

There are two parts to the HV control system:

- A server (*/usr/bin/hv_server*) which communicates with the CAEN HV mainframe. This should be started up at system start time and should run under the "hv" account. The "hv" account also owns a master configuration file, which the server must be able to read, but which should be hidden from non-privileged users (as it contains passwords) and the channel limit configuration file. There is also a control pipe, which should be writable by any account, which should be allowed to operate the system. The server generates a data file, giving the current status of the system. This should be world readable.
- A client (*/usr/bin/hv_control*) which provides a graphical user interface (GUI) for the system. This can run on any account which has write access to the control pipe and read access to the data file.

3 Security and authorisation

The authorisation philosophy is that only experts should use the "hv" account, in order to:

- change the IP address or username/password for the HV system by changing the master configuration file.
- change the name of a channel or the limits for voltages and leakage currents in the limits configuration file.

The server should run under the "hv" account.

Other users should use a non-privileged account other than "hv" and are then constrained by the limits set by the experts.

All logging is written to the system log file.

The server and client both have to run on the same computer. No ports are opened, so there is no possible impact on the security of the host computer. In order to access the software, a user must first log onto the computer hosting the server e.g. via ssh.

4 The HV server

The server reads the file */var/lib/hv/hv_master_config.dat* which contains the IP address of the HV mainframe, the username and password to use on the mainframe and the logging level. This file should only be owned by "hv" and should be readable and writable by "hv" and "root" but nobody else (i.e. permissions 600).

It then tries to connect to the mainframe. If it succeeds, it reads the current values. If not, it keeps trying until it succeeds. Note that if the connection is lost, it will again keep trying until it can be re-established. It is not necessary to restart the server if the HV mainframe is rebooted as the server will reconnect and as soon as the mainframe is available again.

Next the system reads the file */var/lib/hv/hv_channel_limits.dat* which contains one line for each HV channel being used. e.g.:

```
!       Number  Name          Vmax     Rup      Rdown  Curlim Curtime
CHANNEL 0       "12A"         4000     1        5      0.8    10
CHANNEL 1       "12B"         3500     1        5      0.8    10
CHANNEL 2       "12C"         4000     1        5      0.8    10
```

where the first line is a comment. This file defines the name of each channel, the maximum voltage, the ramp up and ramp down speeds, the current limit and the time the channel must exceed this current limit for a trip to occur. This file should be world readable, but only "hv" or "root" should be able to change it. It should be owned by "hv" and have permissions 644.

After that, the server checks for commands on the control pipe */var/lib/hv/hv_control*. This pipe should be world writable (i.e. permissions 666).

All these steps are performed over and over, the files being checked for changes on each iteration and re-read if they have been modified. This means, that if you change, for example, the name of a channel, the server will see that the file has changed, re-read it and tell the mainframe to change the name. If you change the IP address, the server will disconnect from the mainframe and try to connect to the new address. There is no need to restart the server for this.

> It is the server, that enforces the policy of preventing the user from ramping up in too large steps. Below 1500 Volts, you may not ramp up by more than 500 Volts in a single request. Above 1500 Volts you may not ramp up by more than 250 Volts in a single request.

Of course, there is nothing to stop you from continually increasing the slider, so that the ramping effectively goes from zero to maximum without a break. However, you should not do this unless you are capable of fixing broken detectors!

If a warning status occurs, the script */var/lib/hv/hv_warn_script.sh* is called and should a trip occurs */var/lib/hv/hv_warn_trip.sh* is called. These scripts should belong to the "hv" user and can be made to perform any action the user wishes, such as sending an e-mail or SMS.

5 The graphical control program

> The graphical control program is started by the command *hv_control*.

It does not communicate directly with the high voltage mainframe, but instead sends commands via the named pipe */var/lib/hv/hv_control* to the server and reads the file *hv_channel_data.dat* which is periodically

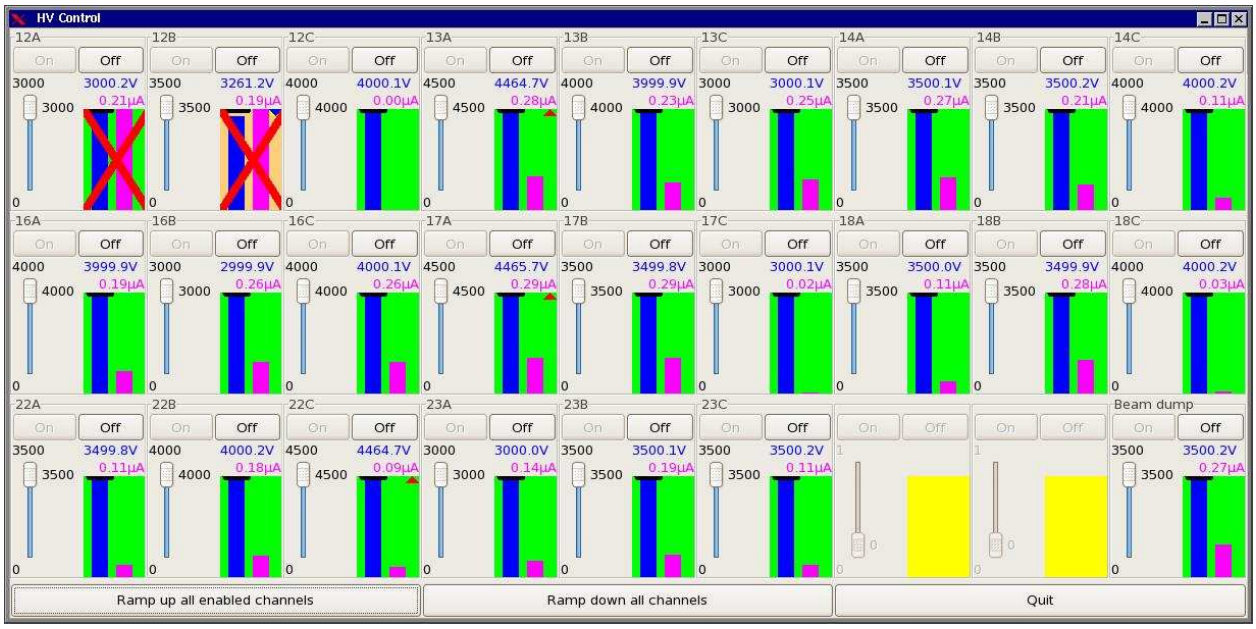

Figure 1: The graphical control program.

written by the server.

The program has one box for each channel of the mainframe (this depends on the number of cards installed, but is normally 36).

Each box has the following items:

- The name of the channel. This cannot be changed in the GUI (it can only be set from the "hv" account by editing */var/lib/hv/hv_channel_limits.dat*.
- A field with a coloured background which is yellow if the channel is off, green if it is on or pale orange if the channel has tripped.
- A button marked "On" to turn on the high voltage, which is disabled if the voltage is already on.
- A button marked "Off" to turn on the high voltage, which is disabled if the voltage is already off. If you click on "Off" for a channel that is above 50 Volts and not tripped, the program will ask you to confirm. If you say yes, it will ramp down safely.
- A slider, to allow the user to set the demand voltage. Note, that the range of this slider is automatically changed by the program, to prevent the user from ramping up too fast. The maximum range is determined from the measured voltage and the maximum voltage for the channel. If the measured voltage is below 1500 Volts the range is from zero to 500 Volts above the measured voltage. If the measured voltage is above 1500 Volts, the range is from zero to 500 Volts. If, however, this is more than the maximum voltage, then the range is from zero to the maximum voltage. In this way, it is not possible for the normal user to apply more than 500 Volts in one step at low voltages or 250 Volts in one step at higher ones. Changing the slider causes the program to ask the server to make that change. This will happen at the appropriate ramp up or ramp down rates.
- Next to the slider is the demand voltage, which changes when the slider is moved.
- Below the slider is the lower limit of the range (always zero).
- Above the slider is the upper limit of the range (which changes depending on the measured voltage).
- Above the coloured background is the measured voltage in blue and the measured current in magenta. These values are also displayed as bars in the coloured field. If the blue bar goes right to the top of the coloured field, it indicates the voltage is at its maximum allowed value for the channel.
- A black horizontal line indicates the demand voltage. If this is right at the top of the coloured field, it means that channel is set to ramp to full voltage.

At the bottom, there are three buttons:

- A button to ramp up all enabled channels. Pressing this button, will set the demand voltage on every channel that is on, to its greatest allowed value, as determined by the measured voltage. i.e. it is exactly the same thing as sliding the slider all the way up on each of these channels. It will not ramp up more than 500 Volts when below 1500 Volts or 250 Volts when above.
- A button to ramp down all channels. This is the fast but safe way to turn everything off. It ramps down each channel at the appropriate ramp down speed.
- A quit button, which leaves the program. This doesn't do anything to the server. Channels which were on, remain on. Channels that were ramping up or down, continue to ramp.

There are certain tips, which pop up to explain the meaning of a control, when you move the mouse pointer over it.

If a warning occurs for a channel, a red cross will be drawn over that channel, which will disappear if the warning goes away (e.g. if the channel's measured current goes above the current limit for a time shorter than the current time). If, however, the channel trips, the background colour changes to pale orange. In that case, the channel can only be used again by turning it off and then back on again.

In the example in figure 1, both the first two channels are in an overcurrent state, as can be seen from the fact that the magenta bar reaches the top of the box and there is a red cross over the box. However, in the case of channel 12A the measured current only briefly exceeded the current limit, so this is a warning, not a trip. Channel 12B, however, has tripped, as can be seen from the pale orange background. It is also ramping down, as can be seen from the blue downward pointing arrow. This state was produced by reducing the current limit on both channels, and reducing the current time on 12B.

Both channels 12C and 13A have the demand voltage turned up to their respective maxima (4000 Volts for 12C and 4500 for 13A), but while 12C is already at full voltage, 13A is still ramping up, as can be seen from the red upward pointing arrow.

Two of the channels at the bottom are unused. There is no entry for them in */var/lib/hv/hv_channel_limits.dat*, so they do not have a name and both on and off buttons are disabled. They are off, so they have a yellow background rather than a green one.

6 Logging

The GUI logs to windows that it opens as needed.

The server logs to the system logger, so messages should end up in somewhere like */var/log/messages* or */var/log/syslog* (depending on the configuration of the system logger. They should automatically be rotated by the system logrotate program.

7 Files

7.1 /var/lib/hv/hv_master_config.dat

This file has one keyword and value pair on each line. Valid keywords are:

- LOGLEVEL - sets the level of logging. 1 is normal, a higher number means more things are logged to the system log file.
- SYSNAME - this is an arbitrary string enclosed in double quotes. It is sent to the mainframe and used to identify it, but that is about all.
- IP - this is the IP address (not the name) of the HV mainframe.
- USERNAME - this is the username to use on the mainframe.
- PASSWORD - this is the password to use on the mainframe.

This file should belong to "hv" and have permissions 600. This ensures that nobody except "hv" and "root" can change the IP address of the HV mainframe and normal users cannot even read the username and password. Note, however, that at time or writing, the username and password for the CAEN HV mainframe

are given in the manual and cannot be changed!

Blank lines are ignored and an exclamation mark introduces a comment which continues until the end of the line.

Here is an example, with fake entries for the IP address, username and password:

```
LOGLEVEL            1                    ! Logging level
SYSNAME             "Miniball HV system" ! This is just an arbitrary string
IP                  192.168.1.1          ! IP address of the mainframe as a system
USERNAME            someusername         ! Username on mainframe
PASSWORD            somepassword         ! Password on mainframe
```

7.2 /var/lib/hv/hv_channel_limits

This file sets the limits for each channel. There should be one line per channel starting with the keyword "CHANNEL" and followed by the number, name, maximum voltage, ramp up rate, ramp down rate, current limit and current time. Note that the name should be in double quotes.

The fields are:

- The keyword "CHANNEL".
- The name of the channel in double quotes.
- The maximum voltage that a normal user is allowed to apply to that channel in volt.
- The ramp up rate in volts/second.
- The ramp down rate in volts/second.
- The current limit in μA,
- The current time in seconds. If the measured current exceeds the current limit for this time, a trip will occur (this is enforced by the mainframe itself).

The channel number corresponds to the channel on the mainframe. The name is arbitrary. The maximum voltage is the highest voltage that a normal user is allowed to apply to that channel. The ramp up rate is the increase in volts per second when ramping up. The ramp down rate is the corresponding decrease when ramping down. The current limit and current time are for trips. If the measured current exceeds the current limit (in μA) for the current time (in seconds) the channel will trip (this is enforced by the mainframe itself).

Blank lines are ignored and an exclamation mark introduces a comment which continues until the end of the line.

```
CHANNEL 0       "12A"         3000     1        5      0.8    10 ! First cluster
CHANNEL 1       "12B"         3500     1        5      0.8    10
CHANNEL 2       "12C"         4000     1        5      0.8    10
```

This file should be owned by "hv" and have permissions 644. In this way, "root" or "hv" can change these limits, but normal users can only read them.

7.3 /var/lib/hv/hv_channel_data.dat

This file is created and periodically updated by the server. It should be owned by "hv" and be world readable. You can read it at any time to see what the status of the mainframe is. The fields are:

- Channel number.
- Channel name.
- Enable/disable flag (0 means channel is off, 1 means channel is on).
- Demand voltage in volts.
- Measured voltage in volts.

- Maximum voltage in volts.
- Ramp up rate in volts/second.
- Ramp down rate in volts/second.
- Measured current in μA.
- Maximum current in μA for trip purposes.
- Maximum time, measurement is allowed to exceed maximum current before a trip occurs (in seconds).
- Channel status flag - a bit flag (see below).
- The temperature of the board in °C.
- Board status flag - another bit flag (see the manual of the HV card).
- The maximum voltage that the HV card can supply.

The bits of the channel status flag have the following meanings (taken from page 9 of the *CAENHVWrapper-2.7.doc* file). Values of type "Information" are just to let the user know what is happening. Values with "Warning" are an indication that something is not normal, but it is not yet considered to be an error. Values of type "Error" are deemed to be serious and the HV will be shut down for that channel. For example, if the measured current goes above the limit for a time shorter than the current time, this will give the over-current warning, but if it stays above that limit for longer than the current time the internal trip error will be triggered and the HV shut down.

Note that some of these values have hard coded limits. It is not possible, even for experts, to increase the ramp up rate above 20 Volts/s or the ramp down rate above 50 Volts/s. Both these values have a minimum of 1 Volt/s as well (since they are integers and ramping down with 0 Volts/s doesn't do anything useful).

Bit	Meaning	Type
Bit 0	Channel is on	Information
Bit 1	Channel is ramping up	Information
Bit 2	Channel is ramping down	Information
Bit 3	Channel is over-current	Warning
Bit 4	Channel is over-voltage	Warning
Bit 5	Channel is under-voltage	Warning
Bit 6	Channel is in external trip	Error
Bit 7	Channel is in max V	Error
Bit 8	Channel is in external disable	Error
Bit 9	Channel is in internal trip	Error
Bit 10	Channel is in calibration error	Error
Bit 11	Channel is unplugged	Error
Bits 12...31	Reserved, forced to 0	Error

The bits of the board status flag have the following meanings (also taken from page 9 of the *CAENHVWrapper-2.7.doc* file). If any bit is set, this is an error:

Bit	Meaning
Bit 0	Board is in power-fail status.
Bit 1	Board has a firmware checksum error.
Bit 2	Board has a calibration error on HV.
Bit 3	Board has a calibration error on temperature.
Bit 4	Board is in under-temperature status.
Bit 5	Board is in over-temperature status.
Bit 6...31	Reserved, forced to 0.

7.4 /var/lib/hv/hv_warn_script.sh

This is a shell script which should be owned by "hv" and have permissions 755. It is called if a warning occurs. e.g. if the measured current goes above the current limit for a time shorter than the current limit. Such events are not necessarily a problem, and the high voltage remains up, but they could be the precursor to a problem. In particular, if the measured current goes above threshold for long enough to cause a trip, it will first trigger a warning then a trip.

You might want to record all the warnings in a file and e-mail it to someone once a day, for example.

7.5 /var/lib/hv/hv_trip_script.sh

This is a shell script which should be owned by "hv" and have permissions 755. It is called if a trip occurs. e.g. if the measured current goes above the current limit for a time longer than the current limit.

A trip indicates a problem, which is probably urgent, so you might want to send an SMS when this happens.

8 Control from command line

As well as using the graphical user interface, it is possible to control the high voltage directly from the command line. The following scripts are available:

- hv_kill - ramps down all channels safely to zero.
- hv_unkill - cancels a kill, by turning back on any channel over 100 Volts and setting the demand voltage equal to the measured voltage.
- hv_enable - enable a specific channel.
- hv_disable - disable a specific channel, causing it to ramp down safely to zero.
- hv_set_voltage - ramp up or down a specified channel to a specified voltage.
- hv_ramp_up - ramp up the specified channel (or all enabled channels) to the highest safe voltage.

When using these commands it is useful to monitor things using:

```
cat /var/lib/hv/hv_channel_data.dat
```

8.1 hv_kill

This ramps down all the channels to zero at the configured ramp down speed. Any arguments are logged as the reason for the kill, but do not influence the way the command works.

8.2 hv_unkill

This command doesn't take any parameters. It enables any channel with a measured voltage more than 100 Volts and sets the demand voltage equal to the measured voltage. In other words, it cancels a kill, stopping further ramping down, but doesn't ramp back up.

8.3 hv_enable

This command takes a single argument, the channel number to be enabled. It turns on the channel, but does not change the demand voltage.

8.4 hv_disable

This command takes a single argument, the channel number to be disabled. It turns off the channel, causing it to ramp down at the configured ramp down speed.

8.5 hv_set_voltage

This command takes two parameters. The first is the channel number and the second is the voltage to set. The usual rules restricting changes of voltage apply (since they are enforced by the server).

8.6 hv_ramp_up

This command takes a single argument, the channel number to be ramped up. It ramps up the channel (as long as it is enabled) and sets it to the highest voltage allowed by the policy. This takes into account both the maximum voltage for the channel and the maximum step allowed based on the measured voltage.

Alternately, if "-a" is given as the single argument instead of a channel number, all enabled channels are ramped.

9 Controlling via the pipe

It is possible to send commands directly to the server by writing to the control pipe. The commands are: enable, disable, voltage, kill and ramp_up.

To turn on channel 12:

```
echo enable 12 > /var/lib/hv/hv_control
```

To turn off channel 12:

```
echo disable 12 > /var/lib/hv/hv_control
```

To set the voltage for channel 12 to 4000 Volts:

```
echo voltage 12 4000 > /var/lib/hv/hv_control
```

Note, however, that unless the voltage is already over 3750 Volts, the server will not let you set the voltage that high. Instead, it will set the voltage to 500 Volts above the present measured voltage for low voltages or 250 Volts above it for higher voltages. It will also not let you set voltages above the maximum set in the limits file.

If you request too much voltage, this is logged, but the system will simply ignore the request and chose the closest voltage to the one you selected, which is allowed based on the present measured voltage.

Once the voltage is set, the mainframe will ramp up or down as appropriate at the rate set in the limits file.

To turn off all channels and ramp them down to zero Volts:

```
echo kill > /var/lib/hv/hv_control
```

To tell channel 12 to ramp up to the highest safe voltage (i.e. not exceeding the maximum voltage, nor going in too large a step based on the measured voltage):

```
echo ramp_up 12 > /var/lib/hv/hv_control
```