HADES RICH700

High Voltage Control

Documentation

Adrian Amatus Weber - AG Höhne

JUSTUS-LIEBIG-
UNIVERSITÄT
GIESSEN

1 General informations

This document describes the usage of the EPICS based RICH700 high voltage slow control. The EPICS database files are written for a ISEG crate with a CC24 Master. This Master allows to upload up to 5 `.db` files and one `.substitution` file. The EPICS IOC is running on the CC24 Master. To get access to the crate, the computer has to be connected to the crate via Ethernet (LAN). Therefore both devices have to be in the same network. If the computer is connected to the crate, one can open the browser and enter the IP-address of the crate (Default: 192.168.2.1). After a login it is possible to upload the database and substitution files under `iCScontrol`→`EPICS`. After this step a restart of the IOC is necessary. Due to the limited access to the EPICS files on the crate, it is not possible to run a sequencer or different EPICS modules (Planed for the future: ssh access).

2 EPICS Code

The substitution file calls two different database files. Each database file is called with different parameters.
The standard naming schema from ISEG always starts with `ISEG:` . This was changed to the HADES-naming schema: `HADES:RICH:HV:CR1:` which is called with `DEV`. To keep the ability to use the browser interface of the crate, aliases to the `ISEG` naming schema are implemented. The next parts of the PV naming schema are

- CAN_LINE : ID of the CAN line (here: 0)
- MODULE_ID : ID of the module in the crate (0 . . . 5)
- CHANNEL_ID : ID of the channel of a module (0 . . . 15)
- DEVICE_ID: Every crate has a own device ID (here: 1000)

These naming parts are used to call the database files and create the process variables (PV) for the proper crate. For example: The voltage of a channel of a module can be called via

```
${DEV}:${CAN_LINE}:${MODULE_ID}:${CHANNEL_ID}:VoltageSet
```

This schema is more or less self-explanatory and so it wont be explained in more detail.
The EPICS database is splitted into `two different files`. The first database file is called `iseg_epics_1.db`. This file contains the basic EPICS code from ISEG. This *basic* code is a standard code from delivery and gives the possibility to access the single channels of every module but also the possibility to get status informations of every module or the whole crate, firmware informations and so on. This basic code is used as the ground layer of the database. Every group behaviour of the crate will access this layer of the code (see Fig. 5). The standard code allows to access single channels and change their voltage, current, voltage bounds, . . . and also to measure the voltage, current and so on. There is also the possibility to switch on and off the different channels.

2.1 Group Behaviour

If the system is used with up to 16 channels per module and up to 6 modules per crate, it is not suitable to change all the values of each channel of each module separately. In order to get this more comfortable the database gets a new layer (see Fig. 5). This layer, which is called *Group Behaviour*, is added to every module of the crate and gives the possibility to set many process variables of a module in one step. Every PV that belongs to this layer is called with ...:${MODULE_ID}:Group.... The group behaviour is defined for all the necessary set-values and has (nearly) always the same inner structure. The set-values are

- GroupSetVoltage
- GroupSwitchOn
- GroupSetCurrent
- GroupSetDelayedTripTime
- GroupSetVoltageBounds
- GroupSetCurrentBounds
- GroupSetEmergency

The inner structure of these PVs is always like the structure of GroupSetVoltage (see Fig. 1). The GroupSetVoltage PV is a `dfanout` record and writes its value to the `.VAL` fields of the two PVs `FanGroupSetVoltage1_` and `FanGroupSetVoltage2_`. These PVs are used to fan out the voltage that should be set to all (selected) channels of the module. For this step we need 2 PVs because the dfanout record has only up to 8 output fields. It is necessary to use the `dfanout` field `SELM` in the `Mask`-mode to have the possibility to select different channels of the module.
All of the `GroupSetVoltage`, `GroupSetCurrent`, ... channels are implemented the same way, except the PV GroupSwitchOn. The PV `GroupSwitchOn` is a `bo` record. It's output is written to a `dfanout` record called `FanGroupSwitchOn_` which behaves like the `GroupSetVoltage`, `GroupSetCurrent`, ... process variables in all other cases.
Now it is possible to set values to many different PVs of a module at the same time, but it is not possible to select the channels individually. Therefore the new PV `GroupSelection` is introduced (see Fig.2 and List. 3). The PV `GroupSelection` is a `mbboDirect` record and can store a 16 bit value. This value contains the information of the channels which are selected.
For example, 1 means, that channel 0 should be set. 3 has the meaning, that channel 0 and 1 should be set, and so on.
First, the `GroupSelection` PV writes its value to `FanGroupSelection1_`. Second, the PV `GroupSelection2_` is processed via a Forward Link.
The PV `GroupSelection2_` is very important. Due to the fact, that a dfanout record has only 8 output channels, it can only read the first 8 bits of a mbboDirect record.

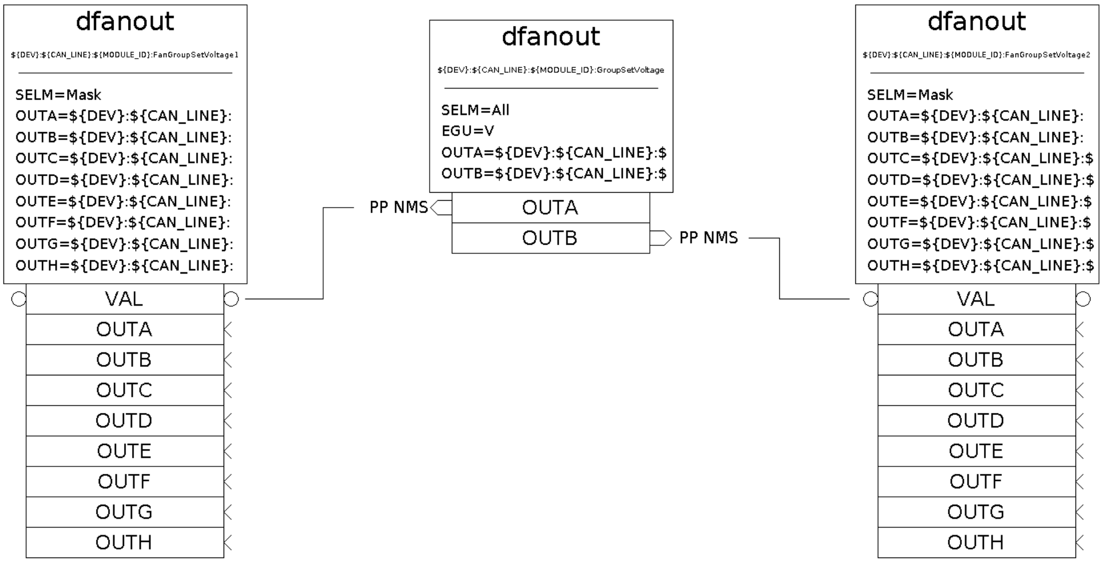


Figure 1: VDCT picture of the EPICS code for GroupSetVoltage from iseg_epics_1.db. The OUT fields without a outgoing line process the data to the corresponding PV `VoltageSet` of every channel of the module. The EPICS source code can be found in Listing 1 in the section 4.

But the information of the selection of the channels 8-15 is stored in the upper 8 bits of a mbboDirect. So it is necessary to shift the mbboDirect record 8 bits to the right. The problem is that a mbboDirect record has only the possibility to shift to the left. Therefore one needs the `GroupSelection2_` PV. `GroupSelection2_` is a mbbiDirect record which can shift to the right. So the `GroupSelection` value is send to the GroupSelection2_ PV, shifted to the right, then stored as a mbboDirect record in `GroupSelection3_` and then written to the PV `FanGroupSelection2_`. The PVs `FanGroupSelection1_` and `FanGroupSelection2` set the mask of all seven set-values like GroupSetVoltage. All the outputs are `NPP` which means they do not process the corresponding PV. This is also very important because the mask has to be set before the PV is processed and it should be selectable which value is set and processed. Therefore the PV `StartBehaviour_` is implemented. This value is processed after all `SELN` fields are set. `StartBehaviour_` is a `dfanout` record and sets the values to the GroupSetVoltage, GroupSetCurrent, ... PVs. The `SELN` field is again used as a selector for the value one wants to set. The StartBehaviour_ PV is the access point to the group behaviour and is called from other layers above in the EPICS Code (The layers are *Variable Group Behaviour* and *Grouping*).
The group behaviour code allows to set different values like voltage, current, ... for selectable channels at one time and completely individually.

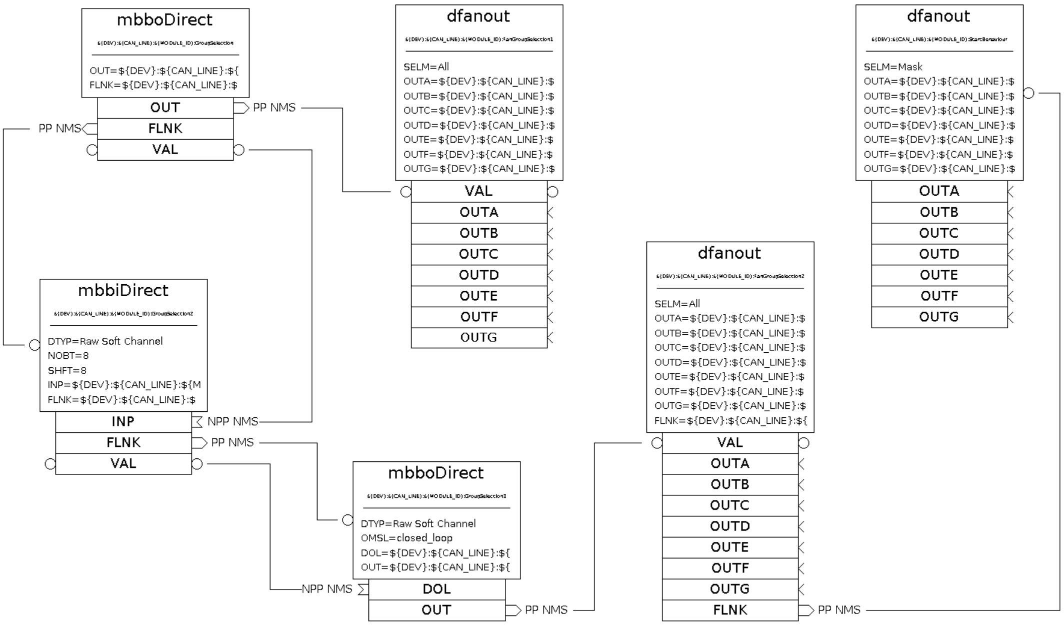

mbboDirect
${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection
OUT=${DEV}:${CAN_LINE}:${
FLNK=${DEV}:${CAN_LINE}:$
OUT
FLNK
VAL
PP NMS
PP NMS
dfanout
${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSelection1
SELM=All
OUTA=${DEV}:${CAN_LINE}:$
OUTB=${DEV}:${CAN_LINE}:$
OUTC=${DEV}:${CAN_LINE}:$
OUTD=${DEV}:${CAN_LINE}:$
OUTE=${DEV}:${CAN_LINE}:$
OUTF=${DEV}:${CAN_LINE}:$
OUTG=${DEV}:${CAN_LINE}:$
VAL
OUTA
OUTB
OUTC
OUTD
OUTE
OUTF
OUTG
dfanout
${DEV}:${CAN_LINE}:${MODULE_ID}:StartBehaviour
SELM=Mask
OUTA=${DEV}:${CAN_LINE}:$
OUTB=${DEV}:${CAN_LINE}:$
OUTC=${DEV}:${CAN_LINE}:$
OUTD=${DEV}:${CAN_LINE}:$
OUTE=${DEV}:${CAN_LINE}:$
OUTF=${DEV}:${CAN_LINE}:$
OUTG=${DEV}:${CAN_LINE}:$
OUTA
OUTB
OUTC
OUTD
OUTE
OUTF
OUTG
mbbiDirect
${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection2
DTYP=Raw Soft Channel
NOBT=8
SHFT=8
INP=${DEV}:${CAN_LINE}:${M
FLNK=${DEV}:${CAN_LINE}:$
INP
FLNK
VAL
NPP NMS
PP NMS
dfanout
${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSelection2
SELM=All
OUTA=${DEV}:${CAN_LINE}:$
OUTB=${DEV}:${CAN_LINE}:$
OUTC=${DEV}:${CAN_LINE}:$
OUTD=${DEV}:${CAN_LINE}:$
OUTE=${DEV}:${CAN_LINE}:$
OUTF=${DEV}:${CAN_LINE}:$
OUTG=${DEV}:${CAN_LINE}:$
FLNK=${DEV}:${CAN_LINE}:${
VAL
OUTA
OUTB
OUTC
OUTD
OUTE
OUTF
OUTG
FLNK
PP NMS
mbboDirect
${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection3
DTYP=Raw Soft Channel
OMSL=closed_loop
DOL=${DEV}:${CAN_LINE}:${
OUT=${DEV}:${CAN_LINE}:${
DOL
OUT
NPP NMS
PP NMS

Figure 2: VDCT picture of the EPICS code for GroupSelection from iseg_epics_1.db. The OUT fields without a outgoing line process the data to the corresponding PVs. The PVs can be found in the EPICS source code in listing 3 in section 4.

2.2 Variable Group Behaviour (VarGB)

The group behaviour code gives the possibility to set values to selected channels. The new layer in the EPICS code is called `Variable Group Behaviour (VarGB)` and is implemented to select every channel of a module individually via a `checkBox` and set the value of these selected channels from one PV.

In order to select the channels the new PV `VarGBGroupSelection` is created. This PV is a `mbboDirect` record and stores a 16 bit value. This record was chosen because every module has up to 16 channels. So every channel represents one bit of the VarGroupSelection PV. Each bit will be set from a `checkBox` in the CSS GUI (see section 3). The VarGBGroupSelection PV writes its value to a `dfanout`. This dfanout is called `FanVarGBGroupSelection` and fans out the selected channels to all seven set-values, which are called

- VarGBVoltageSet
- VarGBSwitchOn
- VarGBCurrentSet
- VarGBDelayedTripTimeSet
- VarGBVoltageBoundsSet
- VarGBCurrentBoundsSet
- VarGBEmergencySet

All these PVs have the same internal structure. Here `VarGBVoltageSet` is chosen as an example (see Fig. 3 and List. 2). The PV `VarGBVoltageSet` gets the value of the voltage from the CSS GUI and writes this value to the `DO3` field of the new process variable `VarGBSeqVoltageSet_`. `VarGBSeqVoltageSet_` writes the voltage information, which is stored in `.DO3`, to the `VAL` field of `StartBehaviour_` from the EPICS layer *Group Behaviour* which is described as the access point in the previous subsection.
`DO1` gets the binary information about the channels which should be set. `DO2` is the field which selects which set-value is set. The coding of `DO2` can be seen in table 1.

Value	Process Variable
1	sets the Voltage
2	sets the SwitchOn
4	sets the Current
8	sets the DelayedTripTime
16	sets the VoltageBounds
32	sets the CurrentBounds
64	sets the Emergency

Table 1: Coding of the DO2 field of the VarGBSeq... PVs.

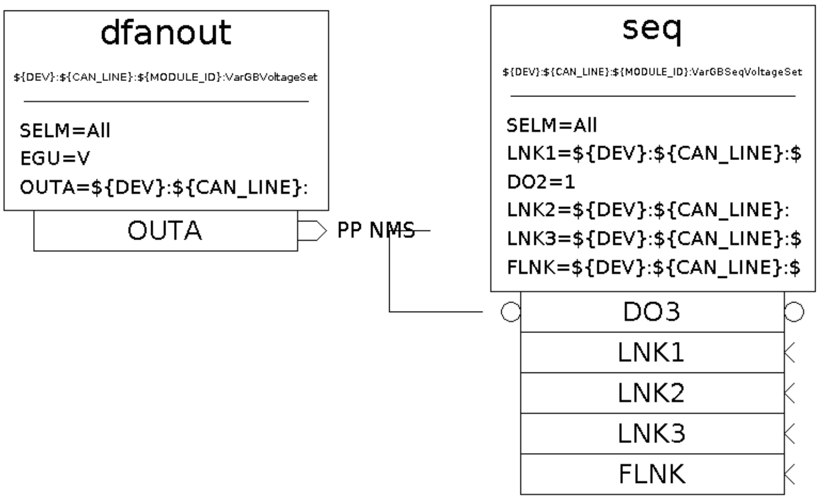

dfanout
${DEV}:${CAN_LINE}:${MODULE_ID}:VarGBVoltageSet
SELM=All
EGU=V
OUTA=${DEV}:${CAN_LINE}:
OUTA
PP NMS
seq
${DEV}:${CAN_LINE}:${MODULE_ID}:VarGBSeqVoltageSet
SELM=All
LNK1=${DEV}:${CAN_LINE}:$
DO2=1
LNK2=${DEV}:${CAN_LINE}:
LNK3=${DEV}:${CAN_LINE}:$
FLNK=${DEV}:${CAN_LINE}:$
DO3
LNK1
LNK2
LNK3
FLNK

Figure 3: VDCT picture of the EPICS code for VarGBVoltageSet from iseg_epics_1.db. The LNK fields without a outgoing line process the corresponding PVs StartBehaviour_.VAL and .SELN and GroupSelection.VAL of the module with NPP. The FLNK processes the PV GroupSelection. The EPICS source code can be found in listing 2 in section 4.

All the LNK fields are again set as `NPP`. This is necessary because of the structure of StartBehaviour_. StartBehaviour_ will set the values of the channels as soon as it is processed. But the channels have to be selected first and the selection itself is done with `GroupSelection`. So `StartBehaviour_` has to be started from the PV `GroupSelection` and not from any other PV!

2.3 Static Grouping

Another important possibility is to define groups with static channels. This means that one has access to all 6 modules and each module has a static pre-selection of channels. So the selected channels are not changeable via a *checkBox*.
The corresponding EPICS Code is written in the second file which is called from .substitutions, `iseg_epics_2.db`.
The naming schema is now a bit different. It is just

`${DEV}:${CAN_LINE}:${GROUP}:`

Every static Grouping G1, G2, ... is created via the `.substitutions` file and gets the values for the selected channels of every module from this file. The selection of a channel is still done via a binary coding. If channel 0 should be selected, the number for GroupSelection is 2^0=1. If channel 2, 5 and 13 should be selected, the number for GroupSelection is $2^2 + 2^5 + 2^{13} = 8228$ and so on. With GroupSelection=0 no channel is selected and the module stays untouched.

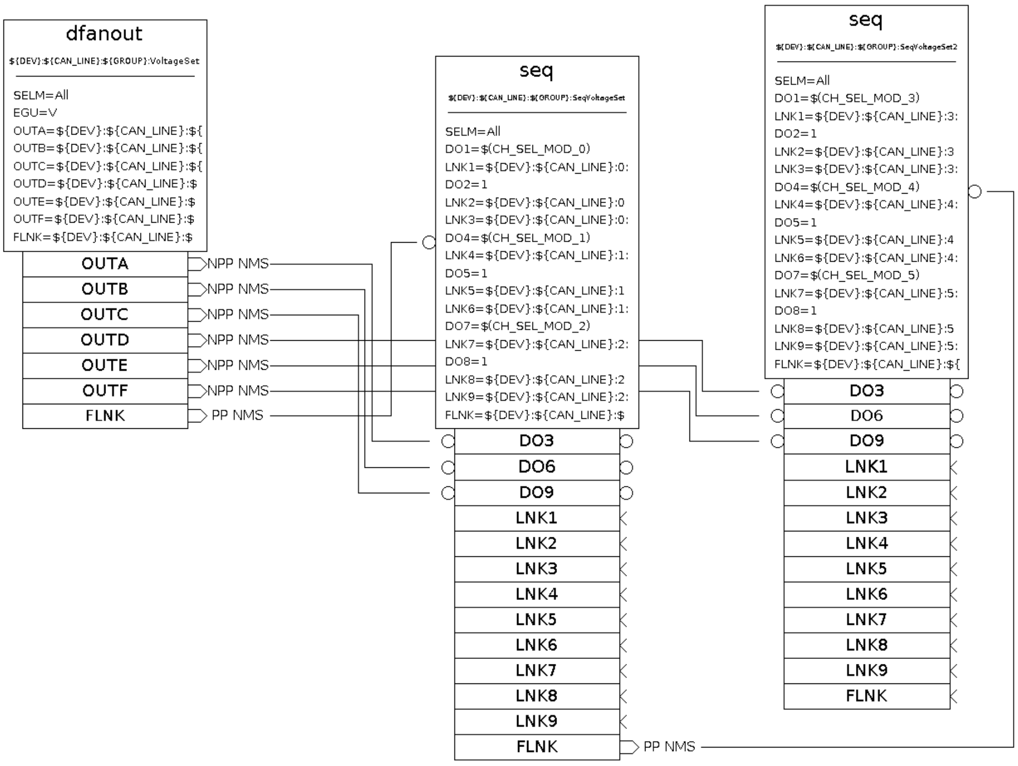

dfanout
${DEV}:${CAN_LINE}:${GROUP}:VoltageSet
SELM=All
EGU=V
OUTA=${DEV}:${CAN_LINE}:${
OUTB=${DEV}:${CAN_LINE}:${
OUTC=${DEV}:${CAN_LINE}:${
OUTD=${DEV}:${CAN_LINE}:$
OUTE=${DEV}:${CAN_LINE}:$
OUTF=${DEV}:${CAN_LINE}:$
FLNK=${DEV}:${CAN_LINE}:$
OUTA
OUTB
OUTC
OUTD
OUTE
OUTF
FLNK
NPP NMS
NPP NMS
NPP NMS
NPP NMS
NPP NMS
NPP NMS
PP NMS
seq
${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet
SELM=All
DO1=$(CH_SEL_MOD_0)
LNK1=${DEV}:${CAN_LINE}:0:
DO2=1
LNK2=${DEV}:${CAN_LINE}:0
LNK3=${DEV}:${CAN_LINE}:0:
DO4=$(CH_SEL_MOD_1)
LNK4=${DEV}:${CAN_LINE}:1:
DO5=1
LNK5=${DEV}:${CAN_LINE}:1
LNK6=${DEV}:${CAN_LINE}:1:
DO7=$(CH_SEL_MOD_2)
LNK7=${DEV}:${CAN_LINE}:2:
DO8=1
LNK8=${DEV}:${CAN_LINE}:2
LNK9=${DEV}:${CAN_LINE}:2:
FLNK=${DEV}:${CAN_LINE}:$
DO3
DO6
DO9
LNK1
LNK2
LNK3
LNK4
LNK5
LNK6
LNK7
LNK8
LNK9
FLNK
PP NMS
seq
${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet2
SELM=All
DO1=$(CH_SEL_MOD_3)
LNK1=${DEV}:${CAN_LINE}:3:
DO2=1
LNK2=${DEV}:${CAN_LINE}:3
LNK3=${DEV}:${CAN_LINE}:3:
DO4=$(CH_SEL_MOD_4)
LNK4=${DEV}:${CAN_LINE}:4:
DO5=1
LNK5=${DEV}:${CAN_LINE}:4
LNK6=${DEV}:${CAN_LINE}:4:
DO7=$(CH_SEL_MOD_5)
LNK7=${DEV}:${CAN_LINE}:5:
DO8=1
LNK8=${DEV}:${CAN_LINE}:5
LNK9=${DEV}:${CAN_LINE}:5:
FLNK=${DEV}:${CAN_LINE}:${
DO3
DO6
DO9
LNK1
LNK2
LNK3
LNK4
LNK5
LNK6
LNK7
LNK8
LNK9
FLNK

Figure 4: VDCT picture of the EPICS code for VoltageSet from `Grouping` in iseg_epics_2.db. The LNK fields without a outgoing line process the data to the corresponding PVs which can be found in listing 4 in section 4.

The EPICS Code for the static grouping is related to the code of variable group behaviour (see Fig. 4 and List. 4). The PV `VoltageSet` is again a dfanout record and is used to set the voltage value to all possible modules. The field FLNK is necessary to process the new record `SeqVoltageSet_`. `SeqVoltageSet_` is used to set the StartBehaviour_ values and the GroupSelection (the selected channels) for the *Grouping Behaviour* of every module. Due to the fact that a seq record has up to 12 LNK fields and the crate at HADES RICH700 has 6 modules (18 LNK fields), one seq record is not enough. So a second seq record `SeqVoltageSet2_` is implemented. Every seq record controls 3 modules of the crate. It is also possible to control more modules. At the end of the last seq field the record `Start` is processed. `Start` is a dfanout record and processes the `GroupSelection` PV of all 6 modules. This is again necessary due to the processing order of the PVs.

The selected channels of the modules can be changed in the `.substitutions` file or via the fields `DO1`, `DO4` and `DO7` of the two PVs `SeqVoltageSet1_` and `SeqVoltageSet2_` (and also for Current, SwitchOn, . . .).

The whole structure of the EPICS code is shown in Fig. 5. This figure shows that the *basic* EPICS code from ISEG is part of every channel. The group behaviour is also part of every module and every module has 16 channels. The third layer is the code of variable Ggroup behaviour (VarGB). It is connected to every module separately and accesses the code from group behaviour. The last layer is the static grouping G1, G2, ... This grouping accesses the group behaviour code and is directly connected to all modules of the crate.

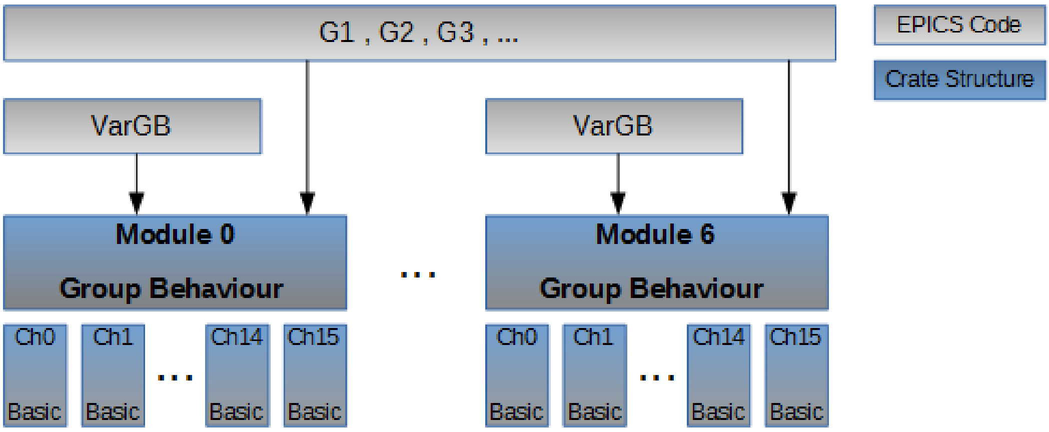


Figure 5: Scratch of the structure of the EPICS code together with the crate structure.

2.4 Save and Load

It is very useful to save and load the settings of all single channels of the crate. This allows to try different voltages, currents and so on, without the danger of loosing the old, working values.

This possibility is implemented in the file `iseg_epics_1.db`. Every channel of the

module has an own save and load PV for every value which can be set. All this `save-PVs` contain the saved value. The load PV uses this value and writes it back to the set values of the channel.
If the PV `HADES:RICH:HV:CR1:0:SaveAll` is processed, many fanout records are processed such that all `save-PVs` which are saved will be processed. If the PV `HADES:RICH:HV:CR1:0:LoadAll` is processed, the same structure of fanout records is processed. The only difference is, that all `Load-PVs` are processed.
The save and load fields will save all single channel set-values. The values of `VarGB` and `Group Behaviour` are not saved due to the fact, that they would overwrite the single channel values if the values are loaded.

3 CSS GUI

The graphical user interface (GUI) for the ISEG crate is created with the software `Control System Studio` (CSS). CSS is used in version 4.11. The GUI is build as a modular GUI. Different .OPI files are linked to the Main OPI file with Linking Containers. This modularisation is shown in Fig. 6. Due to this modularisation a problem appears. If the GUI is started the first time, the performance is very slow. This happens due to the loading process of all the OPI files. But if the OPI files are loaded and the GUI is running, the performance is normal.

The GUI is splitted in 2 different sections:
The upper section 1 (see Fig. 7) includes the main switch of the crate and the fan speed of the crate. The button *Main Switch* switches the crate `on` or `off`. The LED on the right side shows the current status of the crate. The buttons `Save` and `Load` can be used to store and load the values of all single channels of all modules. The button on the right side can be used to switch `on/off` all channels of all modules.
The lower section 2 (see Fig. 7) includes the whole regulation of the modules and the channels. This part contains a tabbed container with the tabs for all modules and one tab that is used for `static Grouping`.

The tabs for the modules have all the same structure. They are divided in 4 parts (see Fig. 8). The first part contains all single channels of a module. There it is possible to set the voltage, current, trip time, voltage bounds and the current bounds. It is also possible to switch on every single channel. The button *emergency* can be used to shut down a single channel immediately. The LEDs in *Status* show the current status of the channels. For example, one sees, if a channel ramps up, is switched on, and so on. In front of every channel is a checkBox. This checkBox is used for channel/Group selection. You can use it to select all the channels of a module you want to operate at once.
The second part of the tab is used to get and set properties of the module. It shows the current temperature, event status , voltage-, current limits, sample rate, the nominal voltage and the nominal current of the module.

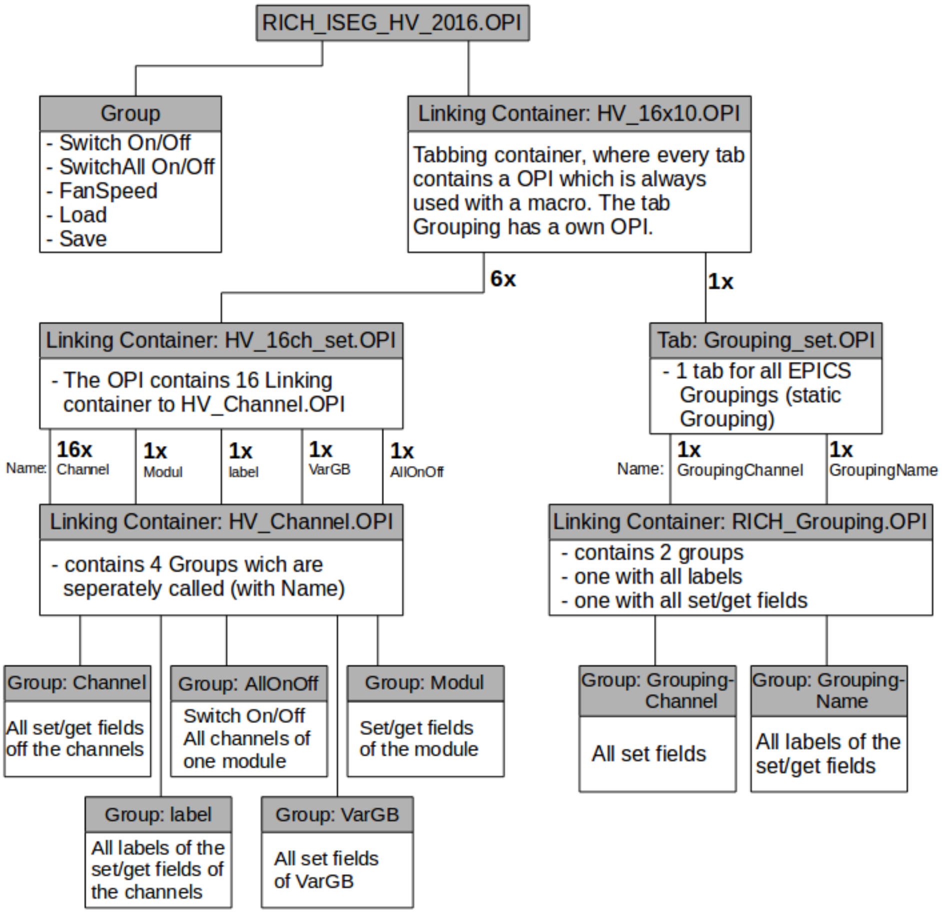
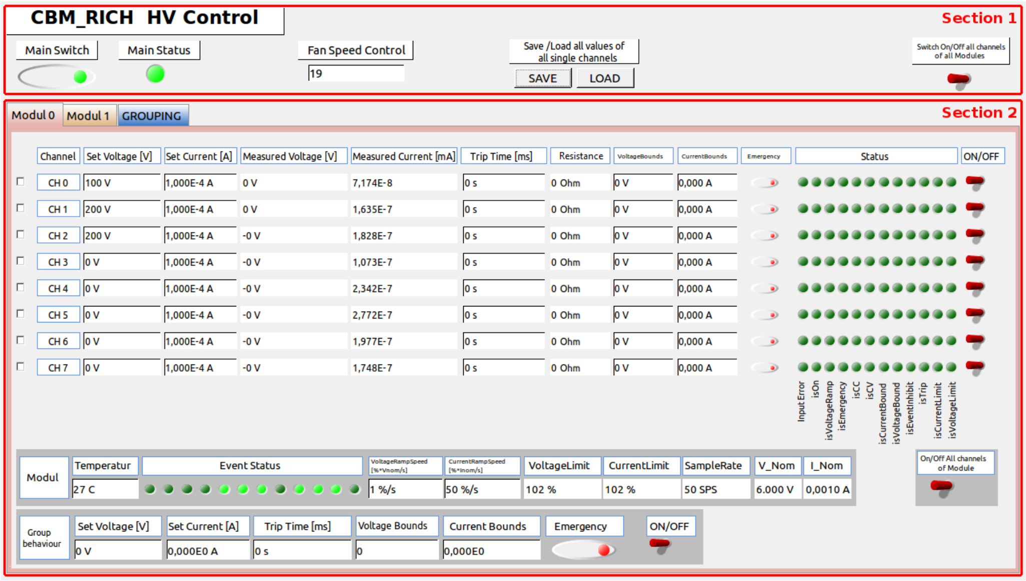

RICH_ISEG_HV_2016.OPI
Group
- Switch On/Off
- SwitchAll On/Off
- FanSpeed
- Load
- Save
Linking Container: HV_16x10.OPI
Tabbing container, where every tab contains a OPI which is always used with a macro. The tab Grouping has a own OPI.
6x
1x
Linking Container: HV_16ch_set.OPI
- The OPI contains 16 Linking container to HV_Channel.OPI
Tab: Grouping_set.OPI
- 1 tab for all EPICS Groupings (static Grouping)
Name:
16x
Channel
1x
Modul
1x
label
1x
VarGB
1x
AllOnOff
Name:
1x
GroupingChannel
1x
GroupingName
Linking Container: HV_Channel.OPI
- contains 4 Groups wich are seperately called (with Name)
Linking Container: RICH_Grouping.OPI
- contains 2 groups
- one with all labels
- one with all set/get fields
Group: Channel
All set/get fields off the channels
Group: AllOnOff
Switch On/Off All channels of one module
Group: Modul
Set/get fields of the module
Group: Grouping-Channel
All set fields
Group: Grouping-Name
All labels of the set/get fields
Group: label
All labels of the set/get fields of the channels
Group: VarGB
All set fields of VarGB

Figure 6: The modularisation of the GUI.



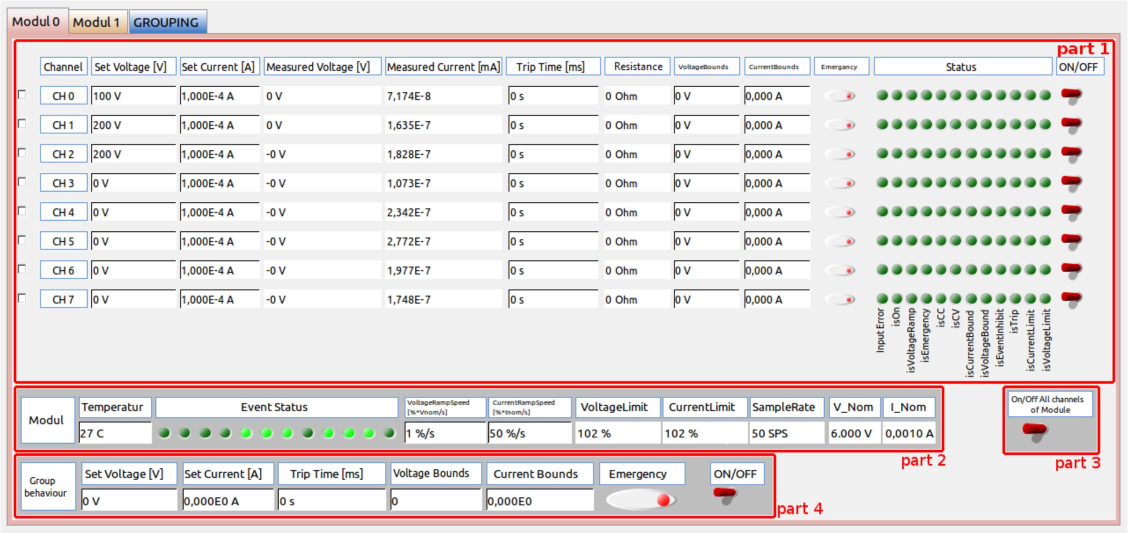
Fan Speed Control: 19

	Channel	Set Voltage [V]	Set Current [A]	Measured Voltage [V]	Measured Current [mA]	Trip Time [ms]	Resistance	VoltageBounds	CurrentBounds	Emergency	Status	ON/OFF
□	CH 0	100 V	1,000E-4 A	0 V	7,174E-8	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 1	200 V	1,000E-4 A	0 V	1,635E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 2	200 V	1,000E-4 A	-0 V	1,828E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 3	0 V	1,000E-4 A	-0 V	1,073E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 4	0 V	1,000E-4 A	-0 V	2,342E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 5	0 V	1,000E-4 A	-0 V	2,772E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 6	0 V	1,000E-4 A	-0 V	1,977E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 7	0 V	1,000E-4 A	-0 V	1,748E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	

Modul	Temperatur	Event Status	VoltageRampSpeed [%*Vnom/s]	CurrentRampSpeed [%*Inom/s]	VoltageLimit	CurrentLimit	SampleRate	V_Nom	I_Nom
	27 C	●●●●●●●●●●●●	1 %/s	50 %/s	102 %	102 %	50 SPS	6.000 V	0,0010 A

Group behaviour	Set Voltage [V]	Set Current [A]	Trip Time [ms]	Voltage Bounds	Current Bounds	Emergency	ON/OFF
	0 V	0,000E0 A	0 s	0	0,000E0		

Figure 7: The full GUI of the Hades Rich700 HV crate.



	Channel	Set Voltage [V]	Set Current [A]	Measured Voltage [V]	Measured Current [mA]	Trip Time [ms]	Resistance	VoltageBounds	CurrentBounds	Emergancy	Status	ON/OFF
□	CH 0	100 V	1,000E-4 A	0 V	7,174E-8	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 1	200 V	1,000E-4 A	0 V	1,635E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 2	200 V	1,000E-4 A	-0 V	1,828E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 3	0 V	1,000E-4 A	-0 V	1,073E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 4	0 V	1,000E-4 A	-0 V	2,342E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 5	0 V	1,000E-4 A	-0 V	2,772E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 6	0 V	1,000E-4 A	-0 V	1,977E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	
□	CH 7	0 V	1,000E-4 A	-0 V	1,748E-7	0 s	0 Ohm	0 V	0,000 A		●●●●●●●●●●●●	

Modul	Temperatur	Event Status	VoltageRampSpeed [%*Vnom/s]	CurrentRampSpeed [%*Inom/s]	VoltageLimit	CurrentLimit	SampleRate	V_Nom	I_Nom
	27 C	●●●●●●●●●●●●	1 %/s	50 %/s	102 %	102 %	50 SPS	6.000 V	0,0010 A

On/Off All channels of Module

Group behaviour	Set Voltage [V]	Set Current [A]	Trip Time [ms]	Voltage Bounds	Current Bounds	Emergency	ON/OFF
	0 V	0,000E0 A	0 s	0	0,000E0		

Figure 8: The GUI of the Hades Rich700 HV crate. The picture shows the tabbed container with all parts of a module.

Here it is important that the fields *nominal Voltage* and *nominal Current* are no real module variables. These variables are taken from a single channel. These fields are listed here because there is no difference between the values for all channels. The fields *VoltageRampSpeed* and *CurrentRampSpeed* are used to change the ramp speed. The value is a percent value. The final voltage (current) that is ramped up, is calculated with

$$V_{Ramp} = V_{RampSpeed} \cdot V_{Nom} \qquad I_{Ramp} = I_{RampSpeed} \cdot I_{Nom}$$

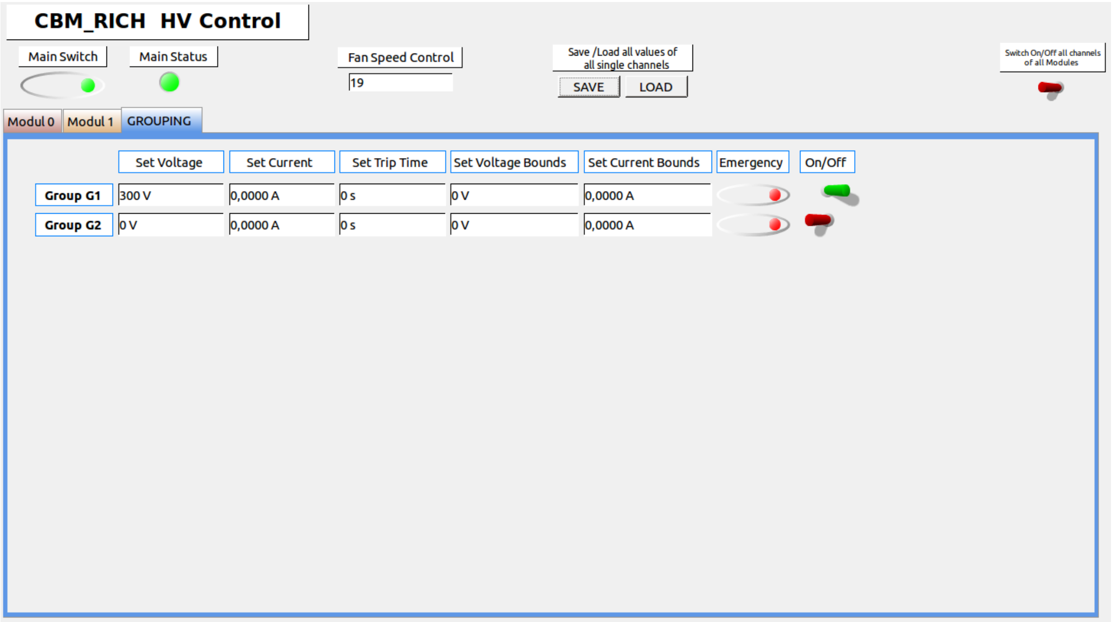
The third part is just one switch button. This button can be used to switch on (or off) all the single channels of a module at once.

The fourth part is used to set the values for *variable Group Behaviour* (VarGB). All the settings which are done there will affect all channels which are selected via the checkBox of the particular channel (see Fig. 10). The tab `Grouping` contains the fields which allows to set new values to a fixed selection of channels from all modules (see Fig. 9). For both, static Grouping and variable group behaviour, all values have the same meaning as in part one.

The change of a value in variable Group Behaviour or static Grouping will change the set values of the corresponding (selected) channels. A change of a value is always independent from other values. For example, it is possible to set the voltage of a channel while it is a selected channel of the variable Group Behaviour. It is also possible to switch on all selected channels at once, change their voltage, and so on. While you do this, it is still possible to change all the single channel values individually. So every selection and setting works completely independent of the other settings.

It is important to mention that a small problem can appear: If the IOC is restarted and one wants to change a value/status of the module via Variable Group Behaviour the first time after restart, the setting or status change wont be recognized. One has to do it twice. The same problem appears for the static Grouping Behaviour and stamps from the record type *mbboDirect*. Unfortunately it is not possible to use another record type and get rid of the problem.

The value of setCurrent is limited to 1mA. If the value of setCurrent is smaller than the current that would result from setVoltage and the load at the HV output, the channel switches in the constant current mode (CC). With a minimum value of 0.1 mA the channel (normaly) stays in the constant voltage mode (CV).


CBM_RICH HV Control

Main Switch ●

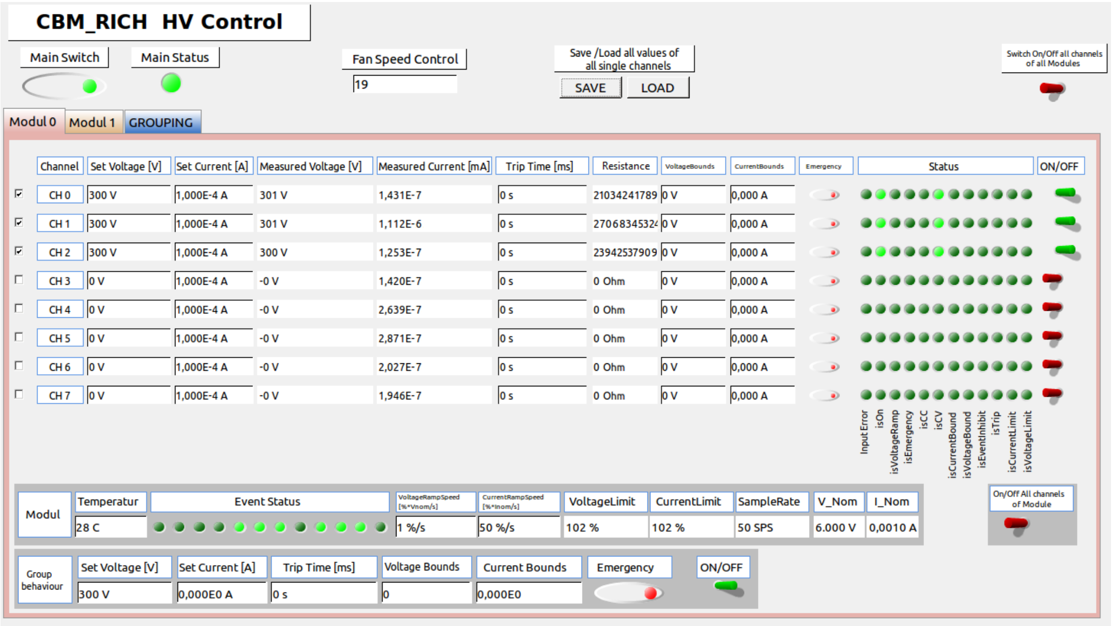
Main Status ●

Fan Speed Control: 19

Save /Load all values of all single channels

	Set Voltage	Set Current	Set Trip Time	Set Voltage Bounds	Set Current Bounds	Emergency	On/Off
Group G1	300 V	0,0000 A	0 s	0 V	0,0000 A	●	🟩
Group G2	0 V	0,0000 A	0 s	0 V	0,0000 A	●	🟥

Figure 9: The full GUI of the Hades Rich700 HV crate with the tab GROUPING.



CBM_RICH HV Control

Main Switch | Main Status | Fan Speed Control: 19 | Save /Load all values of all single channels: SAVE LOAD | Switch On/Off all channels of all Modules

Modul 0 | Modul 1 | GROUPING

	Channel	Set Voltage [V]	Set Current [A]	Measured Voltage [V]	Measured Current [mA]	Trip Time [ms]	Resistance	VoltageBounds	CurrentBounds	Emergency	Status	ON/OFF
☑	CH 0	300 V	1,000E-4 A	301 V	1,431E-7	0 s	21034241789	0 V	0,000 A			
☑	CH 1	300 V	1,000E-4 A	301 V	1,112E-6	0 s	27068345324	0 V	0,000 A			
☑	CH 2	300 V	1,000E-4 A	300 V	1,253E-7	0 s	23942537909	0 V	0,000 A			
☐	CH 3	0 V	1,000E-4 A	-0 V	1,420E-7	0 s	0 Ohm	0 V	0,000 A			
☐	CH 4	0 V	1,000E-4 A	-0 V	2,639E-7	0 s	0 Ohm	0 V	0,000 A			
☐	CH 5	0 V	1,000E-4 A	-0 V	2,871E-7	0 s	0 Ohm	0 V	0,000 A			
☐	CH 6	0 V	1,000E-4 A	-0 V	2,027E-7	0 s	0 Ohm	0 V	0,000 A			
☐	CH 7	0 V	1,000E-4 A	-0 V	1,946E-7	0 s	0 Ohm	0 V	0,000 A			

Modul	Temperatur	Event Status	VoltageRampSpeed [%*Vnom/s]	CurrentRampSpeed [%*Inom/s]	VoltageLimit	CurrentLimit	SampleRate	V_Nom	I_Nom
	28 C		1 %/s	50 %/s	102 %	102 %	50 SPS	6.000 V	0,0010 A

On/Off All channels of Module

Group behaviour	Set Voltage [V]	Set Current [A]	Trip Time [ms]	Voltage Bounds	Current Bounds	Emergency	ON/OFF
	300 V	0,000E0 A	0 s	0	0,000E0		

Figure 10: The full GUI of the Hades Rich700 HV crate with the tab Group 1. This figure shows the variable Group Behaviour of module 1 with the selected channels 0, 1 and 2.

4 Appendix

```
record( dfanout, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSetVoltage" ) {
  field( SELM, "All")
  field( EGU, "V")
  field( OUTA, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetVoltage1.VAL PP" )
  field( OUTB, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetVoltage2.VAL PP" )
}

record( dfanout, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetVoltage1")
{
  field( SELM, "Mask" )
  field( OUTA, "${DEV}:${CAN_LINE}:${MODULE_ID}:0:VoltageSet.VAL PP" )
  field( OUTB, "${DEV}:${CAN_LINE}:${MODULE_ID}:1:VoltageSet.VAL PP" )
  field( OUTC, "${DEV}:${CAN_LINE}:${MODULE_ID}:2:VoltageSet.VAL PP" )
  field( OUTD, "${DEV}:${CAN_LINE}:${MODULE_ID}:3:VoltageSet.VAL PP" )
  field( OUTE, "${DEV}:${CAN_LINE}:${MODULE_ID}:4:VoltageSet.VAL PP" )
  field( OUTF, "${DEV}:${CAN_LINE}:${MODULE_ID}:5:VoltageSet.VAL PP" )
  field( OUTG, "${DEV}:${CAN_LINE}:${MODULE_ID}:6:VoltageSet.VAL PP" )
  field( OUTH, "${DEV}:${CAN_LINE}:${MODULE_ID}:7:VoltageSet.VAL PP" )
}

record( dfanout, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetVoltage2")
{
  field( SELM, "Mask" )
  field( OUTA, "${DEV}:${CAN_LINE}:${MODULE_ID}:8:VoltageSet.VAL PP" )
  field( OUTB, "${DEV}:${CAN_LINE}:${MODULE_ID}:9:VoltageSet.VAL PP" )
  field( OUTC, "${DEV}:${CAN_LINE}:${MODULE_ID}:10:VoltageSet.VAL PP" )
  field( OUTD, "${DEV}:${CAN_LINE}:${MODULE_ID}:11:VoltageSet.VAL PP" )
  field( OUTE, "${DEV}:${CAN_LINE}:${MODULE_ID}:12:VoltageSet.VAL PP" )
  field( OUTF, "${DEV}:${CAN_LINE}:${MODULE_ID}:13:VoltageSet.VAL PP" )
  field( OUTG, "${DEV}:${CAN_LINE}:${MODULE_ID}:14:VoltageSet.VAL PP" )
  field( OUTH, "${DEV}:${CAN_LINE}:${MODULE_ID}:15:VoltageSet.VAL PP" )
}
```

Listing 1: EPICS code of GroupSetVoltage from iseg_epics_1.db .

```
record(dfanout, "${DEV}:${CAN_LINE}:${MODULE_ID}:VarGBVoltageSet") {
  field( SELM, "All" )
  field( EGU, "V")
  field( OUTA, "${DEV}:${CAN_LINE}:${MODULE_ID}:VarGBSeqVoltageSet.DO3 PP")
}

record(seq, "${DEV}:${CAN_LINE}:${MODULE_ID}:VarGBSeqVoltageSet") {
  field( SELM, "All" )
  field( DO1, "" )                     # Channels to execute (binary)
  field( LNK1, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection.VAL NPP" )
  field( DO2, "1" ) # 1 is equal to Voltage
  field( LNK2, "${DEV}:${CAN_LINE}:${MODULE_ID}:StartBehaviour.SELN NPP" )
 #field( DO3, "")                       # Value of Voltage
  field( LNK3, "${DEV}:${CAN_LINE}:${MODULE_ID}:StartBehaviour.VAL NPP")
  field( FLNK, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection" )
}
```

Listing 2: EPICS code of VarGBVoltageSet from iseg_epics_1.db .

```
record( mbboDirect, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection") {
  #field( DESC, "Set this PV to activate Group behavior of this channel.")
  field( OUT, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSelection1.VAL PP")
  field( FLNK, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection2 PP")
}

record( mbbiDirect, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection2") { # right shift
  field( DTYP, "Raw Soft Channel" )
  field( NOBT, "8")
  field( SHFT, "8")
  field( INP, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection.VAL")
  field( FLNK, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection3 PP")
}

record( mbboDirect, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection3") {
  #field( DESC, "Set this PV to activate Group behavior of this channel.")
  field( DTYP, "Raw Soft Channel" )
  field( OMSL, "closed_loop")
  field( DOL, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSelection2.VAL")
  field( OUT, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSelection2.VAL PP")

}

record( dfanout, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSelection1")
{
  field( SELM, "All")
  field( OUTA, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSwitchOn1.SELN")
  field( OUTB, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetVoltage1.SELN")
  field( OUTC, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetCurrent1.SELN")
  field( OUTD, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetDelayedTripTime1.SELN")
  field( OUTE, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetVoltageBounds1.SELN")
  field( OUTF, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetCurrentBounds1.SELN")
  field( OUTG, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetEmergancy1.SELN")
}

record( dfanout, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSelection2")
{
  field( SELM, "All")
  field( OUTA, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSwitchOn2.SELN")
  field( OUTB, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetVoltage2.SELN")
  field( OUTC, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetCurrent2.SELN")
  field( OUTD, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetDelayedTripTime2.SELN")
  field( OUTE, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetVoltageBounds2.SELN")
  field( OUTF, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetCurrentBounds2.SELN")
  field( OUTG, "${DEV}:${CAN_LINE}:${MODULE_ID}:FanGroupSetEmergancy2.SELN")
  field( FLNK, "${DEV}:${CAN_LINE}:${MODULE_ID}:StartBehaviour PP")
}

record(dfanout, "${DEV}:${CAN_LINE}:${MODULE_ID}:StartBehaviour")
{
    field( SELM, "Mask")
    field( OUTA, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSetVoltage.VAL PP")
    field( OUTB, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSwitchOn.VAL PP")
    field( OUTC, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSetCurrent.VAL PP")
    field( OUTD, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSetDelayedTripTime.VAL PP")
    field( OUTE, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSetVoltageBounds.VAL PP")
    field( OUTF, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSetCurrentBounds.VAL PP")
    field( OUTG, "${DEV}:${CAN_LINE}:${MODULE_ID}:GroupSetEmergancy.VAL PP")
}
```

Listing 3: EPICS code of GroupSelection from iseg_epics_1.db .

```
record(dfanout, "${DEV}:${CAN_LINE}:${GROUP}:VoltageSet") {
  field( SELM, "All" )
  field( EGU, "V")
  field( OUTA, "${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet.DO3 NPP")
  field( OUTB, "${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet.DO6 NPP")
  field( OUTC, "${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet.DO9 NPP")
  field( OUTD, "${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet2.DO3 NPP")
  field( OUTE, "${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet2.DO6 NPP")
  field( OUTF, "${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet2.DO9 NPP")
  field( FLNK, "${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet PP")
}

record(seq, "${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet") {
  field( SELM, "All" )

  field( DO1, "$(CH_SEL_MOD_0)" )
  field( LNK1, "${DEV}:${CAN_LINE}:0:GroupSelection.VAL NPP")
  field( DO2, "1" ) # 1 is equal to Voltage
  field( LNK2, "${DEV}:${CAN_LINE}:0:StartBehaviour.SELN NPP")
 #field( DO3, "")
  field( LNK3, "${DEV}:${CAN_LINE}:0:StartBehaviour.VAL NPP")

  field( DO4, "$(CH_SEL_MOD_1)" )
  field( LNK4, "${DEV}:${CAN_LINE}:1:GroupSelection.VAL NPP" )
  field( DO5, "1" ) # 1 is equal to Voltage
  field( LNK5, "${DEV}:${CAN_LINE}:1:StartBehaviour.SELN NPP" )
 #field( DO6, "" )
  field( LNK6, "${DEV}:${CAN_LINE}:1:StartBehaviour.VAL NPP" )

  field( DO7, "$(CH_SEL_MOD_2)" )
  field( LNK7, "${DEV}:${CAN_LINE}:2:GroupSelection.VAL NPP" )
  field( DO8, "1" ) # 1 is equal to Voltage
  field( LNK8, "${DEV}:${CAN_LINE}:2:StartBehaviour.SELN NPP" )
 #field( DO9, "" )
  field( LNK9, "${DEV}:${CAN_LINE}:2:StartBehaviour.VAL NPP" )

  field( FLNK, "${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet2" )
}

record(seq, "${DEV}:${CAN_LINE}:${GROUP}:SeqVoltageSet2") {
  field( SELM, "All" )

  field( DO1, "$(CH_SEL_MOD_3)" )
  field( LNK1, "${DEV}:${CAN_LINE}:3:GroupSelection.VAL NPP")
  field( DO2, "1" ) # 1 is equal to Voltage
  field( LNK2, "${DEV}:${CAN_LINE}:3:StartBehaviour.SELN NPP")
 #field( DO3, "")
  field( LNK3, "${DEV}:${CAN_LINE}:3:StartBehaviour.VAL NPP")

  field( DO4, "$(CH_SEL_MOD_4)" )
  field( LNK4, "${DEV}:${CAN_LINE}:4:GroupSelection.VAL NPP" )
  field( DO5, "1" ) # 1 is equal to Voltage
  field( LNK5, "${DEV}:${CAN_LINE}:4:StartBehaviour.SELN NPP" )
 #field( DO6, "" )
  field( LNK6, "${DEV}:${CAN_LINE}:4:StartBehaviour.VAL NPP" )

  field( DO7, "$(CH_SEL_MOD_5)" )
  field( LNK7, "${DEV}:${CAN_LINE}:5:GroupSelection.VAL NPP" )
  field( DO8, "1" ) # 1 is equal to Voltage
  field( LNK8, "${DEV}:${CAN_LINE}:5:StartBehaviour.SELN NPP" )
 #field( DO9, "" )
  field( LNK9, "${DEV}:${CAN_LINE}:5:StartBehaviour.VAL NPP" )

  field( FLNK, "${DEV}:${CAN_LINE}:${GROUP}:Start" )
}
```

Listing 4: EPICS code of static Grouping VoltageSet from iseg_epics_2.db .